iscte

INSTITUTO
UNIVERSITÁRIO
DE LISBOA

# UWB System and Algorithms for Indoor Positioning

Yuankang Gao

Master in Telecommunication and Computer Science

Supervisor: Dr. Octavian Postolache, Associate Professor with Habilitation
Iscte- Instituto Universitario de Lisboa

Co- Supervisor: Dr. Bin Yang, Assistant Professor
Shanghai Maritime University

November, 2020

## Acknowledgments

As an exchange student, I am very lucky to come to Portugal to study in ISCTE-IUL for one year and to receive important theoretical and practical knowledge for my future. I am happy to mention that I made so many friends in Portugal and I hope to keep our friendship forever.

I would like to express my sincere thanks to my supervisors, Professor Octavian Postolache and Prof. Bin Yang , for their continuous help during the realization of this research. I am very grateful to Prof. Postolache for providing a lot of support for my project and constantly providing me with relevant and useful opinions.

Special thanks go to Instituto de Telecomunicacoes and Iscte-Instituto Universiario de Lisboa that provide me the resources that are required for this MSc dissertation.

I thank to my Portuguese colleagues in the Smart Sensing International Laboratory for their help under a new circumstance that I'm not familiar with.

I can't mention all the people who helped me once during this period, but I sincerely thank to them.

## Resumo

Este trabalho de pesquisa apresenta um estudo de posicionamento de banda ultra-larga (UWB) em ambientes internos considerando diferentes tipos de obstáculos que podem afetar a precisão de localização. No armazém real, uma variedade de obstáculos incluindo metal, placa, trabalhador e outros obstáculos terão impacto NLOS (não linha de visão) no posicionamento do pacote logístico, o que influencia a medição da distância entre o pacote logístico e a âncora, afetando assim a precisão do posicionamento. Um novo método desenvolvido tenta melhorar a precisão do posicionamento interno UWB, através de um algoritmo de posicionamento e algoritmo de filtragem aprimorados. Neste projeto, para simular o ambiente de warehouse em laboratório, diversas simulações comprovam que o algoritmo de filtro de Kalman e o algoritmo de Markov usados podem efetivamente reduzir o erro de NLOS. A validação experimental é realizada considerando um tag móvel montado em uma plataforma de robô.

**Palavras-Chave:** UWB; Posicionamento interno; Algoritmo de filtro Kalman; Algoritmo de Markov, plataformas moveis de tipo robo

# Abstract

This research work presents of study of ultra-wide band (UWB) indoor positioning considering different type of obstacles that can affect the localization accuracy. In the actual warehouse, a variety of obstacles including metal, board, worker and other obstacles will have NLOS (non-line-of-sight) impact on the positioning of the logistics package, which influence the measurement of the distance between the logistics package and the anchor , thereby affecting positioning accuracy. A new developed method attempts to improve the accuracy of UWB indoor positioning, through and improved positioning algorithm and filtering algorithm. In this project, simulate the warehouse environment in the laboratory, several simulation proves that the used Kalman filter algorithm and Markov algorithm can effectively reduce the error of NLOS. Experimental validation is carried out considering a mobile tag mounted on a robot platform.

**Keywords:** UWB, indoor Positioning, Kalman filter algorithm, Markov algorithm.

# Índice

## Tables Index

# Figure Index

## List of Abbreviations

5G – 5th Generation

GPS – Global Positioning System

RFID – Radio Frequency Identification

UWB – Ultra-wide Band

WLAN – Wireless Local Area Network

FCC – Federal Communications Commission

EIRP – Equivalent Isotropically Radiated Power

NLOS – Non-line-of-sight

HMM – Hidden Markov Model

IMM – Interactive Multiple Model

INS – Inertial Navigation System

LOS – Line-of-sight

OFDM – Orthogonal Frequency Division Multiplexing

MB – Multi-band

DS-CDMA – Direct Sequence - Code Division Multiple Access

ICI – Inter-carrier Interference

FEC – Forward Error Correction

TFI – Time-frequency Interleaved

WPAN – Wireless Personal Area Network

PLC – Power Line Communication

HVAC – Heating Ventilation and Air Conditioning

RTLS – Real-time Location System

WSN – Wireless Sensor Network

MCU – Micro Control Unit

TOA – Time of Arrival

AOA – Angle of Arrival

TDOA – Time Difference of Arrival

TOF – Time of Flight

ROS – Robot Operating System

# Chapter 1 – Introduction

## 1.1. Motivation and overview

Now when the 5G Era became a reality, the daily life and various related industries known a spectacular evolution. The positioning technology is one of the most important technologies associated to smart environments, thus indoor positioning technology is used in many industries.

Indoor positioning refers to the realization of indoor positioning. It mainly uses wireless communication, inertial navigation positioning and other technologies that are integrated into a set of indoor positioning system. The position monitoring of people and objects in indoor space is carried out.

Several types of wireless technologies are used for indoor location, which include RFID, Cellular-Based, UWB, WLAN, ZigBee, Bluetooth and so on [1]. These technologies have their own characteristics and advantages. Among them, UWB technology has been widely used in indoor positioning in recent years.

UWB technology is a wireless carrier communication technology that uses a frequency bandwidth above 1 GHz. It does not use a sinusoidal carrier, but uses nanosecond-level non-sinusoidal narrow pulses to transmit data. Therefore, it occupies a large spectrum. Although wireless communication is used, its data transmission rate can reach hundreds of megabits per second. UWB technology can transmit signals over a very wide bandwidth. The Federal Communications Commission of the United States stipulates that UWB technology: occupies a bandwidth of more than 500MHz in the 3.1~10.6GHz frequency band [2].

There are several advantages in UWB technology, compared to traditional wireless technologies. Due to its low equivalent isotropically radiated power emission limit, an UWB signal “behaves as noise” to other radio systems, which results in a low probability of interception and detection. Besides, an UWB signal has excellent multipath immunity and less susceptibility to interferences from other radios, due to its wide bandwidth nature. Furthermore, the UWB technology offers the possibility to simplify the transceivers architectures [3].

For positioning technology, positioning accuracy is particularly important. In recent years, many studies on UWB have focused on the positioning accuracy of UWB. In the past, the accuracy of UWB positioning research mainly focused on the improvement of hardware and UWB localization algorithms. Many classic algorithms or new algorithms were used in UWB indoor localization field , which  were carried out slightly to improve positioning accuracy.

New positioning algorithms were proposed recently reported in the literature. In view of the influence of NLOS on UWB positioning technology, the author proposes [4] to perform particle filtering twice on the original data collected by UWB to achieve precise dynamic target positioning. The first particle filter algorithm is used to store the relationship model between the measured distance and error of each anchor. The second particle filter adjusts the weight according to the distance error model in the update stage, and further constrains the estimated target particles to achieve more accurate dynamic target positioning. It had been reported [5] the usage of Time of Flight (TOF) principle combined with the spatial 4-point positioning method to determine the position of the target and adopted STM32 as master chip to control the DW1000 which is a module that transmits and receives pulse signals to realize the UWB signal transceiving on the hardware;

There are also some studies that build models for calibration. In one study, [6], a method for error correction based on modeling with dynamic features is proposed, taking into account that UWB ranges suffers systematic errors and NLOS error in the form of bias. Divide the area covered by the Indoor pedestrian track, and divide the pedestrian track into multiple nodes. The dynamic character of the kinematic data can be used to correct each trajectory node that is divided. Due to the rigor of kinematic data, each trajectory node to be corrected has good accuracy. Finally, integrate all the divided trajectory areas to get the final calibrated trajectory.

## 1.2. Objetives and research questions

For the future smart logistics technology, the identification and positioning of the logistics packages in the warehouse is a key link. Theoretically speaking, it is easy to realize the goal of carrying tags through UWB technology. But in practical applications such as warehouse, the problem of NLOS (non-line-of-sight) will affect the accuracy of UWB positioning. Heavy metallic machinery often blocks the line of sight path, metal walls and metal surfaces increase reflections and multi-path effects.

Under the existing UWB hardware technology, the objective of this research is to study the UWB localization accuracy in industrial environments and improve the accuracy of UWB indoor positioning through improved positioning algorithms and filtering algorithm. The interface with high usability will be considered. An important objective is to eliminate the positioning errors caused by NLOS through simulation and experiments. New methods will be studied in order to eliminate the positioning errors caused by various interfering objects in the actual warehouse.

Set up in the laboratory through the existing UWB mini3 kit which include three anchors to locate a mobile tag mounted on a robot as well as use host computer software for related data acquisition and processing. Several algorithms such Kalman filter algorithms, Markov algorithm were implemented and tested for accurate indoor localization.

The final positioning result is presented through the corresponding algorithm program that was developed in Matlab. Because the purpose of the research is only to figure out the positioning accuracy of the tag in a fixed position, the positioning result does not need to be real-time.

This project aims to obtain an answer to certain questions, such as:

- What are the factors affecting UWB indoor positioning?

- For the existing UWB kit, are there any minor improvements that can be made？

- Allow for no major changes to the hardware system, which methods can be used to improve positioning accuracy?

- How to realize the robot's line tracking function?

- How to perform laboratory simulation experiments to implement Markov algorithm?

### 1.3. Structure of the Dissertation

This dissertation is divided in seven chapters:

♣ Chapter 2 includes some existing projects regarding UWB, UWB technical protocol and some practical application background.

♣ Chapter 3 contains all the hardware used in the project, including UWB system, microcontrollers, robot.

♣ Chapter 4 describes the software used and some related positioning algorithms. The software part includes the basic functions of the robot and real-time positioning software and simulation software.

♣ Chapter 5 describes the calibration work before the simulation experiment and  the theoretical assumptions on how to conduct the simulation experiment.

♣ Chapter 6 considers the experimental results and simulation results.

♣ Chapter 7 includes the conclusions and future work.

♣ In the annex it is included the submitted article to Conftele2020 that it is still under revision process.

# Chapter 2 – State of the Art

## 2.1. Existing project of algorithm

Nowadays, many researches have focused on filtering algorithms and positioning algorithms to improve the positioning accuracy of UWB systems.

This paper [7] studies the ultra-wideband precise positioning of power field operations in typical application scenarios, and proposes a UWB positioning algorithm based on BP neural network compensation Kalman filter. First, use the positioning system to measure the initial observation solutions at all sampling moments within a certain period of time to obtain the initial observation solution set; secondly, according to the state equation of the Kalman filter, the initial observation solution set is input to the Kalman filter; finally, the BP neural network is used Compensate Kalman filter to obtain the estimated value of the two-dimensional state vector of the positioning system under the minimum mean square error.

A new adaptive UKF algorithm is proposed [8] to achieve high-precision indoor positioning. This method takes into account the error characteristics of UWB in the actual measurement environment. By constructing a state error compensation function, fusing the UKF algorithm, using the previous state estimation value as a reference value, the state compensation amount at that time is calculated, and the tag can be arbitrarily determined in the positioning area. The position is automatically compensated in real time to correct the deviation of the UWB positioning results in the actual measurement environment and improve the positioning accuracy.

A combined positioning method [9] of UWB and pedestrian dead reckoning (PDR) with NLOS detection is proposed. The combined positioning algorithm not only overcomes the positioning error problem of the UWB system under NLOS conditions, but also avoids the cumulative error of pedestrian track estimation. This method filters the harmful data caused by the UWB measurement due to NLOS, and uses the Kalman filter of the residual state to fuse the effective data of UWB and PDR, avoiding the error caused by system nonlinearity.

## 2.2. UWB technical protocol

IEEE802.15.3a has standardized UWB's technical solutions since 2003. In the implementation of UWB physical layer technology, there are two mainstream technical solutions [10]: multi-band OFDM (MB-OFDM) based on orthogonal frequency division multiplexing (OFDM) technology, and direct sequence CDMA (DS-CDMA) based on CDMA technology Program. CDMA technology is widely used in 2G and 3G mobile communication systems.

There is no essential difference between the CDMA technology used in the UWB system and the CDMA technology used in the traditional communication system, but a very high chip rate is used to obtain compliance with UWB technology Standard ultra-wide bandwidth. OFDM is the core technology applied to E3G and B3G, and has the advantages of high spectrum efficiency, anti-multipath interference and strong anti-narrowband interference capabilities.

In [11] two standards have been discussed in fierce controversy for a long time. It is useless to evaluate whether DS-CDMA and MB-OFDM are technically superior or inferior to the final compromise of the scheme. Although within IEEE802.15.3a, after many votes, one of the standards cannot be eliminated and unified. Finally, at the IEEE802 meeting held in January 2006, 802.15.3a disbanded the task group after voting, and the standardization process of UWB in IEEE was terminated.

## 2.3. Application

In the past, UWB technology was actually a communication protocol. Many years ago, UWB technology was still a short-distance protocol focusing on high-speed data transmission. After more than ten years of evolution, today’s UWB technology is based on the characteristics of 2ns pulses and can be used for precise positioning applications. The technology can achieve precise positioning of about 10cm and can measure angle and distance. Whether it is Wi-Fi positioning or Bluetooth positioning, or even any other existing technology, this cannot be achieved. With the rapid development of the Internet of Things, UWB has many effective applications [12]:

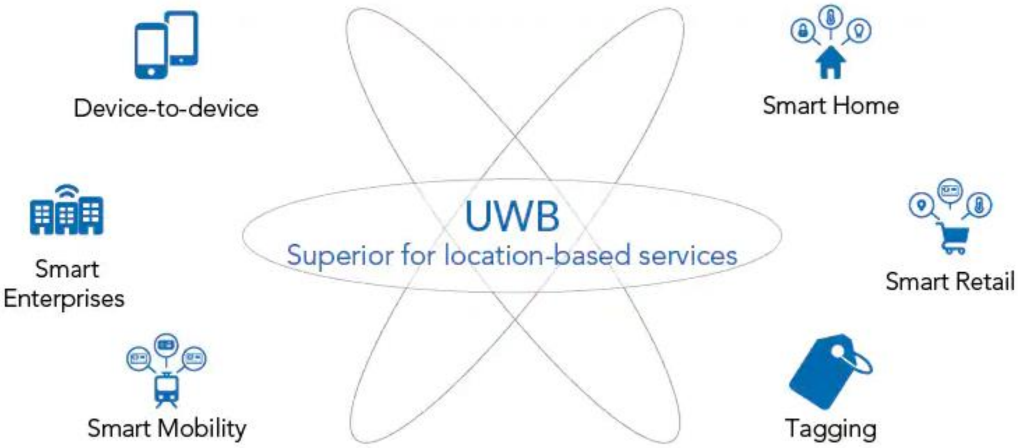


*Figure 1 – UWB applications*

1、Smart factory:

In response to the business needs of personnel, objects or equipment in a mobile state, it improves production efficiency while ensuring the personal safety of operators. Carry out real-time management and information feedback on the production process, provide real-time location data support for the informatization and standardization of management and operations in the factory, and improve the intelligent level of the manufacturing industry.

2、Judicial Prison:

UWB real-time positioning technology is used to liberate police force, improve work efficiency, make up for supervision loopholes, reduce supervision and law enforcement risks, and carry out 24-hour position monitoring and trajectory tracking of prisoners; anti-dismantling positioning bracelet can monitor the vital signs of detainees in real time; it also has electronic fences With the video linkage function, the forbidden borders are arranged at the key entrances and exits and the perimeter, and the detainees will automatically alert when they approach or enter illegally, reduce the risk of supervision and law enforcement, make supervision work intelligent, improve the three-dimensional prevention and control capabilities, and quickly respond to emergencies.

3、Tunnel gallery:

UWB personnel positioning system is used to establish a set of personnel positioning management system, combined with the existing access control system, to realize the positioning display of the personnel in the tunnel/pipe corridor, real-time automatic tracking of the construction personnel, and the camera to perform detection without ID card. Comprehensively grasp the construction personnel's movement trajectory and location distribution in the tunnel, and can trigger an alarm in an emergency. The most important thing is that when encountering an emergency in a tunnel, you can quickly find the location information of the trapped person, ensuring the safety of the personnel, and effectively improving the management efficiency.

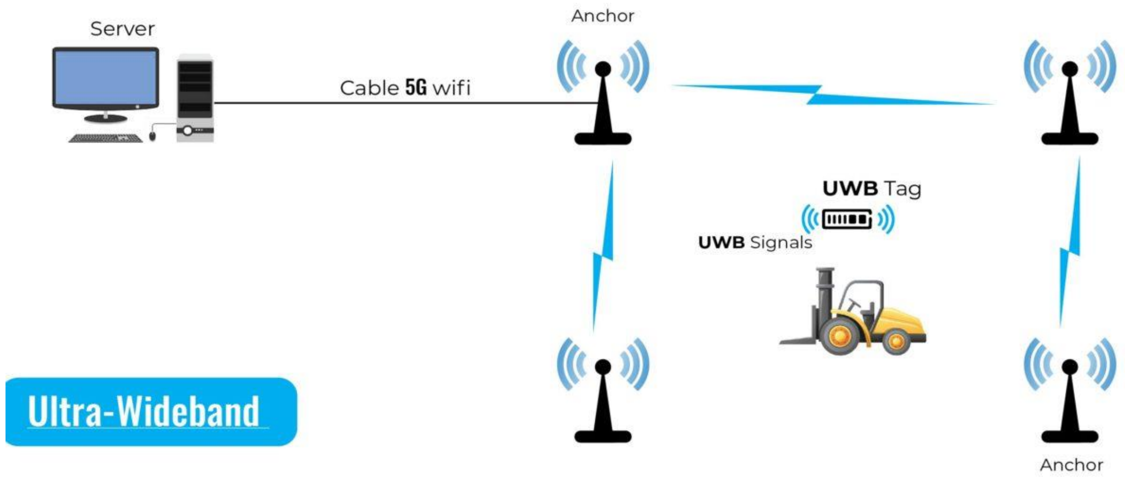


*Figure 2 – UWB transportation application*

# Chapter 3 – Hardware Description

In this project, the UWB kit is main tool to be used to studied accuracy of positioning. Meanwhile, an unmanned detecting vehicle (UDV) robot is used to cooperate with UWB system to complete experiment. Besides, ST-link V2 is responsible for recompiling the calibrated positioning program.

## 3.1. UWB system

The indoor positioning is carried out with a UWB indoor localization kit that consists of 4 UWB Mini3 which integrates UWB positioning system (Figure 3).

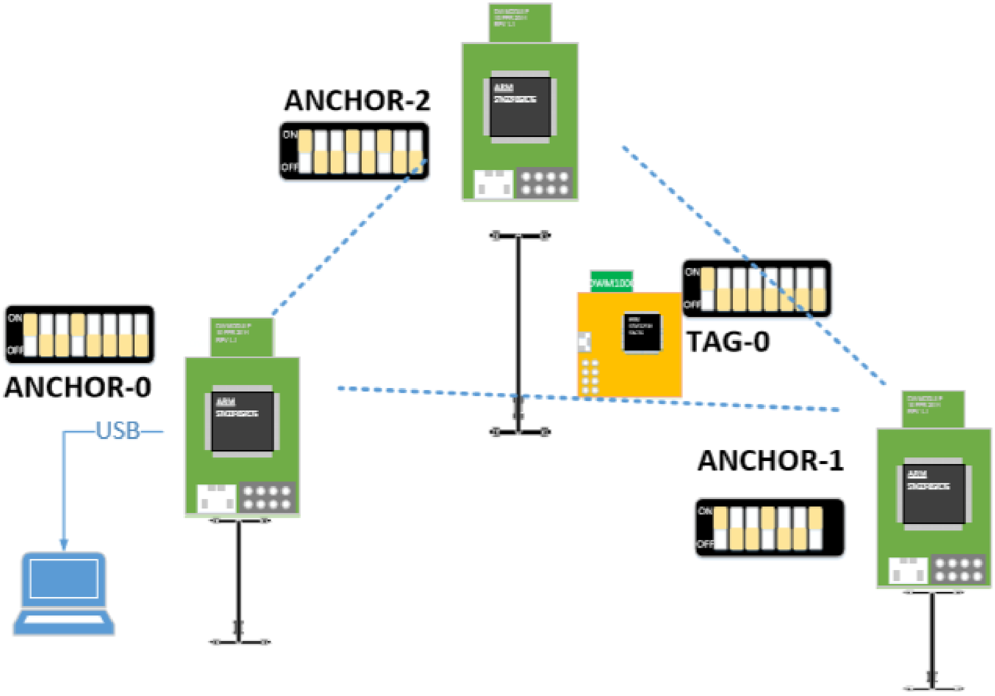


*Figure 3 – UWB system*

UWB Mini 3 (Table 1) module are based on STM32F105 microcontroller as the main control and computation unit. Peripheral circuits include: DWM1000 module, power supply module, LED indicator module, DIP switch, Reset circuit etc. The module can be used as a base station or as a tag, which can be switched through a DIP switch. The implementation diagram of UWB mini3 is shown in Figure 4 .

| Basic parameters | | Wireless parameters | |
|---|---|---|---|
| PCB material | 4-layer board-epoxy resin | Communication rate | 110 kbit/s, 850 kbit/s, 6.8 Mbit/s |
| Power supply interface | micro-USB(5.0V) / terminal | working frequency | 3.5 GHz ~ 6.5 GHz |
| Communication Interface | micro-USB (5.0V) / serial port (3.3V TTL) | Work channel | 6 |
| Download interface | SWD (VCC SDIO SCK GND) | Transmit power | -35dbm/MHZ ~ -62dbm/MHZ |
| main controller | STM32F105RCT6(64 pin) | Maximum length | 1023 bytes |
| External crystal | 12Mhz | Communication distance | Indoor 30m, outdoor 50m |
| PCB size | 35mm * 24mm | Data fluctuation | Typical ±10cm, general shielding ±30cm |

*Table 1 – UWB mini3 overview*

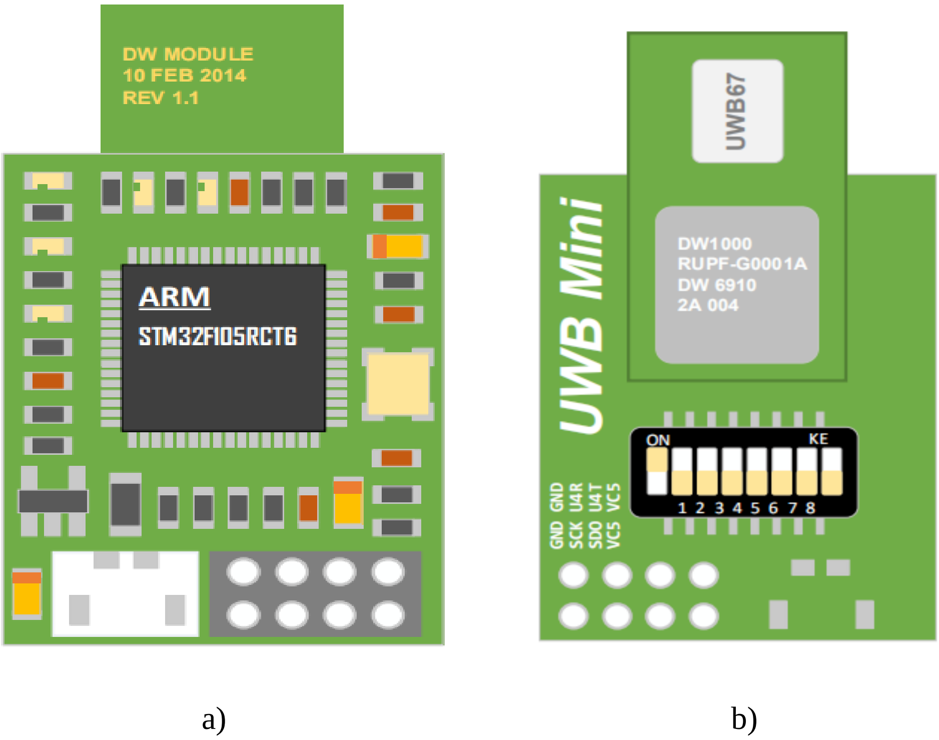


*Figure 4 – UWB Mini 3 a) front view; b) back view*

### 3.1.1 STM32F105

STM32F105, as the main control chip of UWB mini 3, has many features. The STM32F105xx [13] connectivity line family operates in the –40 to +105 °C temperature range, from a 2.0 to 3.6 V power supply, it ensured that we can use it normally in extremely hot or cold weather. A comprehensive set of power-saving mode allows the design of low-power applications. STM32F105 embbed in UWB mini3 is powered by mobile power.

| Power consumption | Tag | Anchor |
|---|---|---|
| Sending and receiving data | 41-54mHA | 170-190mHA |
| Do not send or receive data | 56.1mHA | 192.0-192.3mHA |

*Table 2 – power consumption*

In Figure 5 are presented main blocks associated to the microcontroller. The presented characteristics  make the STM32F105xx connectivity line microcontroller family suitable for a wide range of applications such as motor drives and application control, medical and handheld equipment, industrial applications, PLCs, inverters, printers, and scanners, alarm systems, video intercom, HVAC and home audio equipment.

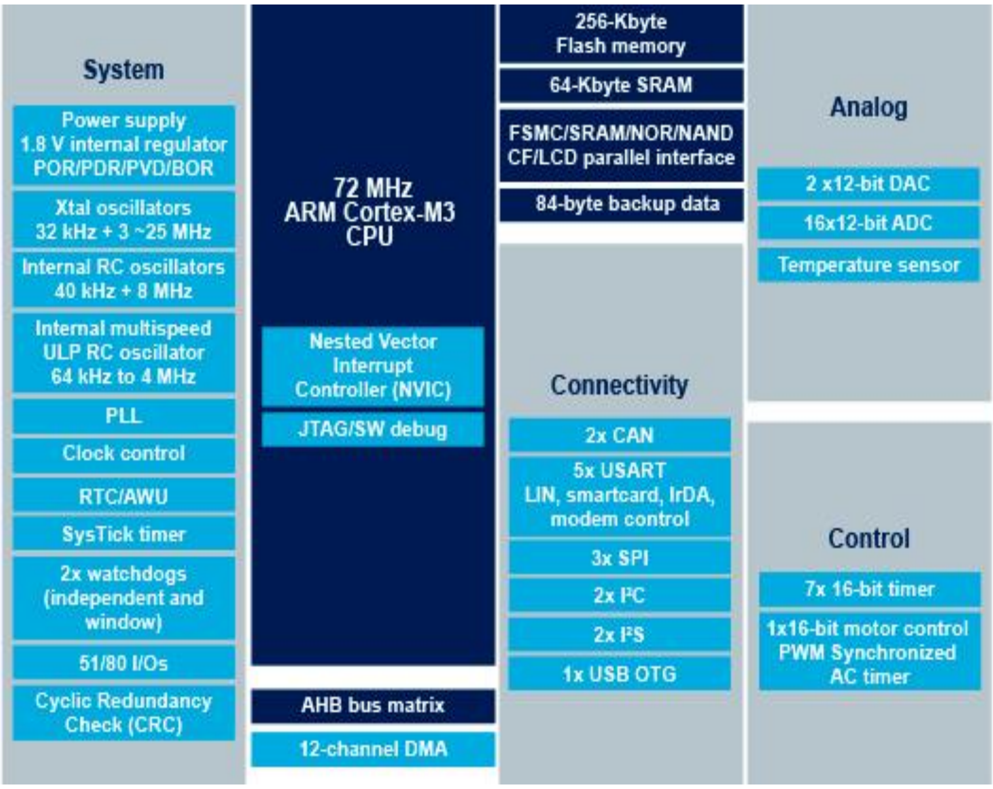


*Figure 5 – STM32F105 overview*

### 3.1.2 DWM1000

Based on the DWM1000 [14] chip developed by DecaWave, the ultra-wideband wireless transceiver chip compatible with the IEEE802.15.4-2011 protocol is used for object positioning in real-time positioning systems, with an accuracy of up to 10 cm, a data transmission rate of up to 6.8Mb/s, and a communication distance About 300 meters.

With short packet communication, the density of tags is as high as 11,000 within a 20-meter radius. It has stronger anti-interference ability, and reliable communication can be carried out in high-fading environment. It has low power consumption characteristics so that reduces the frequency of battery replacement and reduces system life cycle costs. The DWM1000 physical size is small and it is very easy to integrate into real-time location system (RTLS) and wireless sensor network (WSN). The module diagram of DWM1000 is shown in the figure 6.

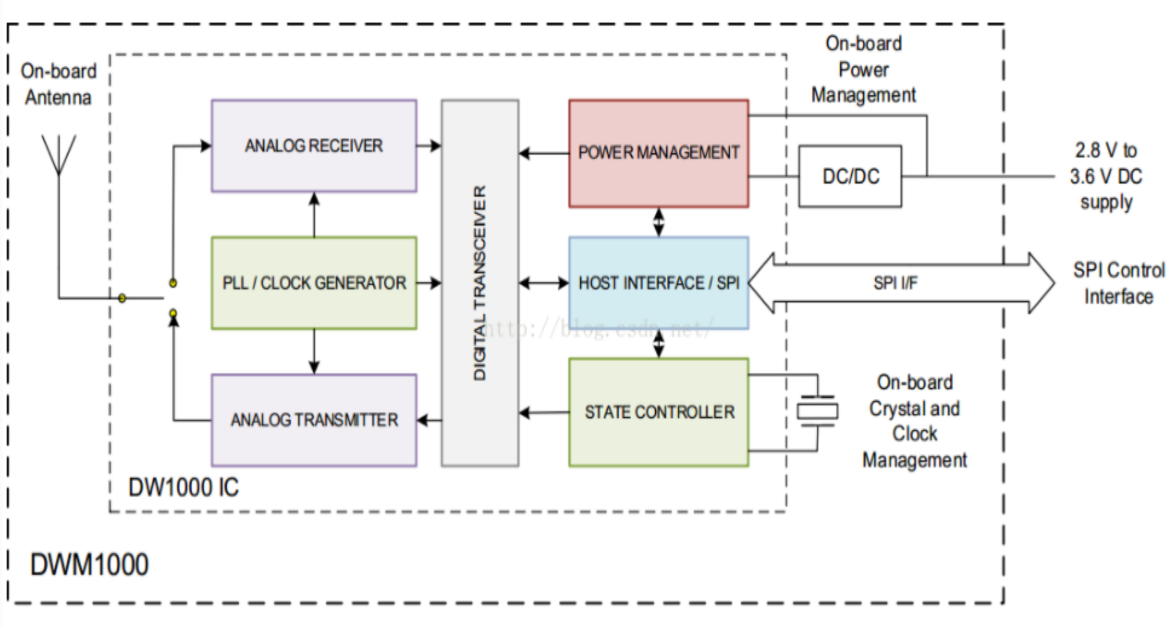


*Figure 6 – DWM1000 module block diagram*

### 3.1.3 UWB mini3 firmware update

In the following chapters, we will mention the need to perform experimental calibration on the existing UWB mini3 firmware before the experiment to ensure that the measurement accuracy is high for the experimental environment at that time. In order to modify its original positioning program, ST-Link V2 (Figure 7) was used to update its firmware.

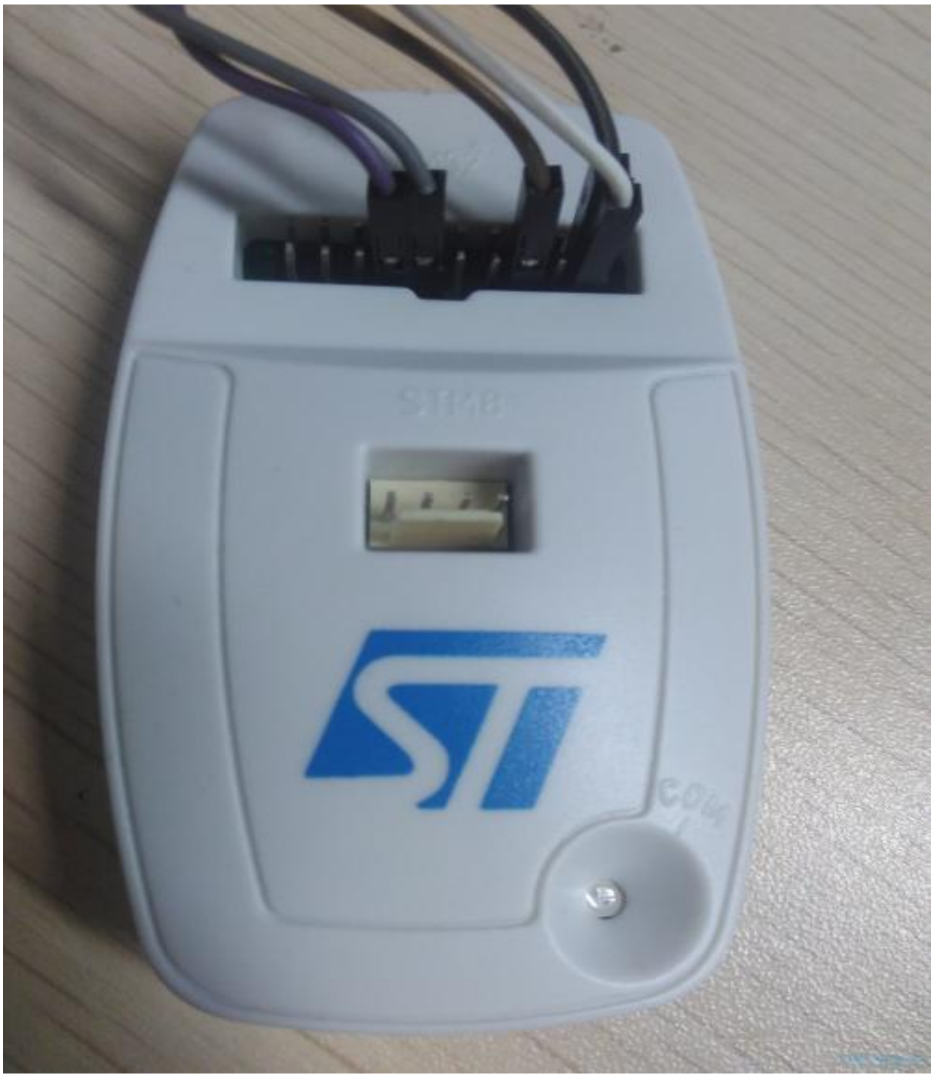
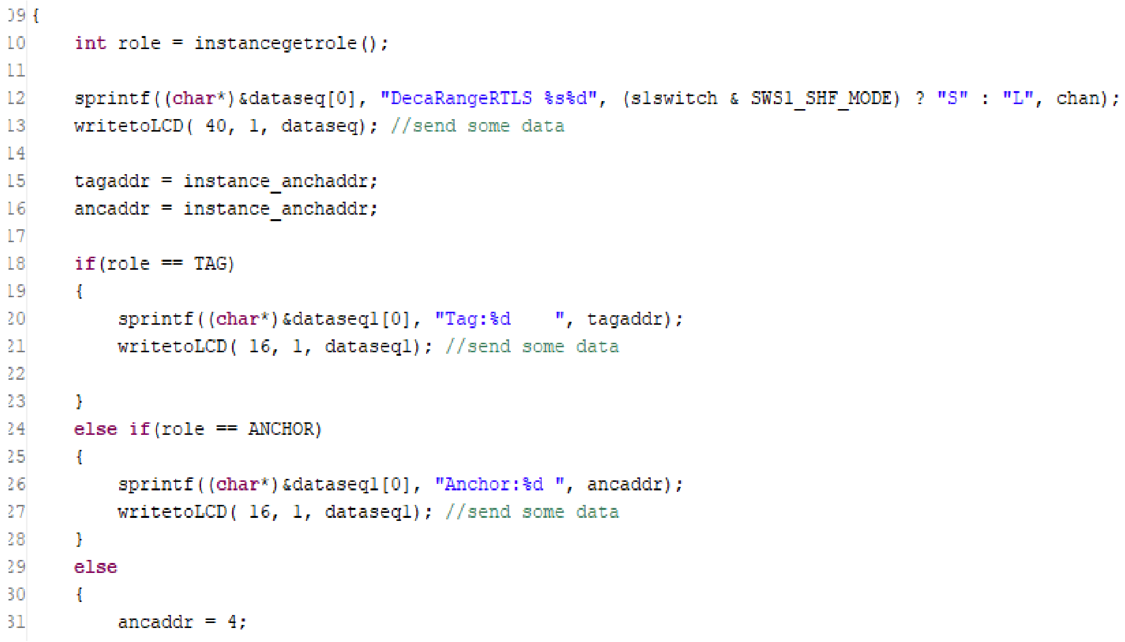


*Figure 7 – ST-Link V2*

ST-Link V2 [15] is ST's second-generation emulator, suitable for ST's 8-bit microcontroller. When we are engaged in embedded project, after writing it, download it to the MCU and run it, if there are errors, it is difficult to find the problem. For example, in Figure 8, we need to download the positioning program to the UWB mini3 microcontroller.



```
{
    int role = instancegetrole();

    sprintf((char*)&dataseq[0], "DecaRangeRTLS %s%d", (s1switch & SWS1_SHF_MODE) ? "S" : "L", chan);
    writetoLCD( 40, 1, dataseq); //send some data

    tagaddr = instance_anchaddr;
    ancaddr = instance_anchaddr;

    if(role == TAG)
    {
        sprintf((char*)&dataseq1[0], "Tag:%d    ", tagaddr);
        writetoLCD( 16, 1, dataseq1); //send some data

    }
    else if(role == ANCHOR)
    {
        sprintf((char*)&dataseq1[0], "Anchor:%d ", ancaddr);
        writetoLCD( 16, 1, dataseq1); //send some data
    }
    else
    {
        ancaddr = 4;
```

*Figure 8 – Positioning program downloaded to MCU*

So when debugging the single-chip computer program, it is necessary to use the simulator to simulate and debug online in order to find out the error of the program. When debugging, the JTAG or SWIM/SWD interface of ST-LINK V2 is connected to the microcontroller, and you can debug by setting breakpoints, observing registers, and single chip operation. ST-Link V2 can be used for single-chip program programming besides simulation debugging. The hardware connection is shown in the figure 9 below.

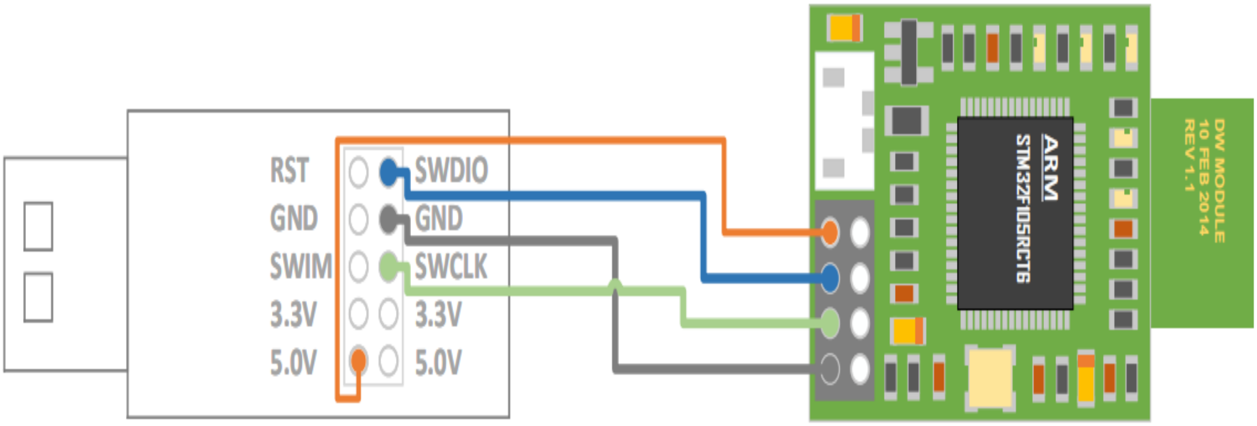


*Figure 9 – ST-Link V2 hardware connection*

### 3.2. UDV Robot

In order to simulate the actual warehouse scenario, we need an unmanned detecting vehicle robot that can be moved by manipulation, the tag is carried by the UDV robot as a carrier so that the moved tag can simulate a mobile logistics package in the warehouse. The UDV robot (Figure 10) is controlled by a PC through Wi-Fi PC, thus the WiFi network includes the PC and Raspberry Pi computation platform mounted on the mechanical part of the UDV robot.

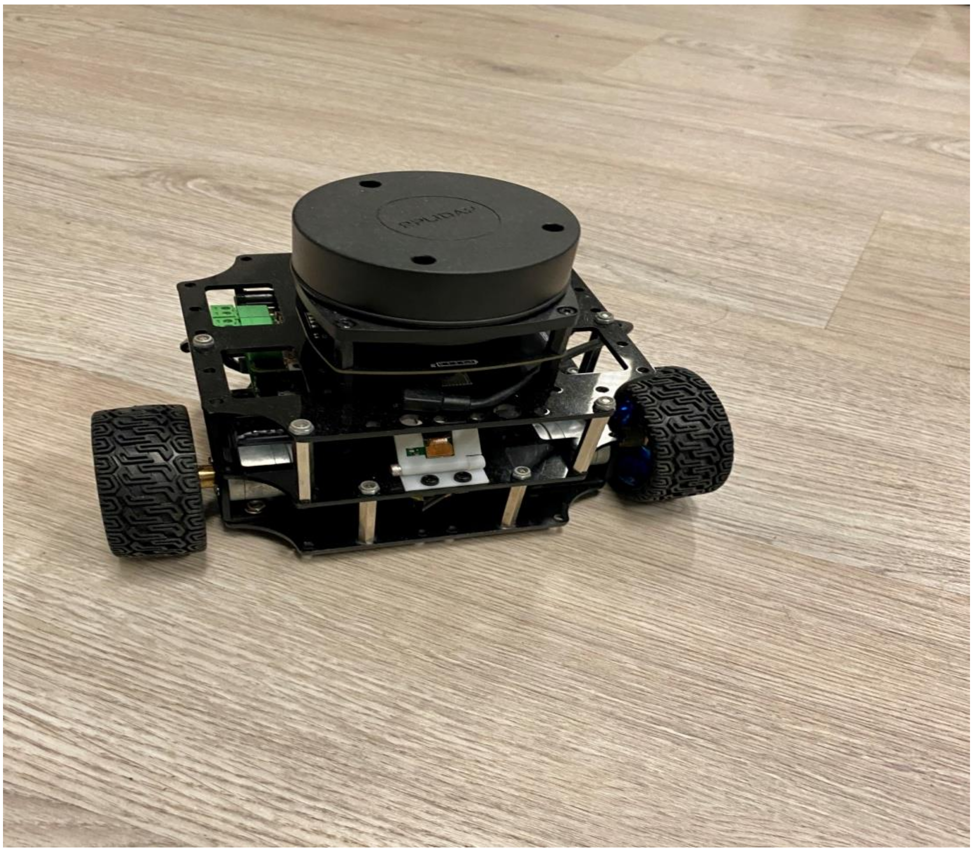


*Figure 10 – UDV Robot*

#### 3.2.1 Raspberry Pi robot computation platform

Raspberry Pi [16] is a miniature card type computer, which can run Linux system and windows IOT system, and then can run applications on these systems. It can be used in the embedded and Internet of Things field, and can also be used as a small server to complete some specific Features. Compared with embedded microcontrollers (common 51 single-chip microcomputers and STM32), in addition to completing the same IO pin control, because it runs the corresponding operating system, it can complete more complex task management and scheduling, and can support higher levels Application development provides a broader application space for developers.

For example, the choice of development language is not limited to the C language. It connects the underlying hardware and the upper application to realize the cloud control and cloud management of the Internet of Things. You can also ignore the IO control of the Raspberry Pi (Figure 11) and use the Raspberry Pi to build a small network server. Do some small test development and services. Compared with the general PC platform, the IO pins provided by the Raspberry Pi can directly control other underlying hardware, which is impossible for a general PC. At the same time, it has a smaller size and a low cost. It can also complete some PC tasks and application.

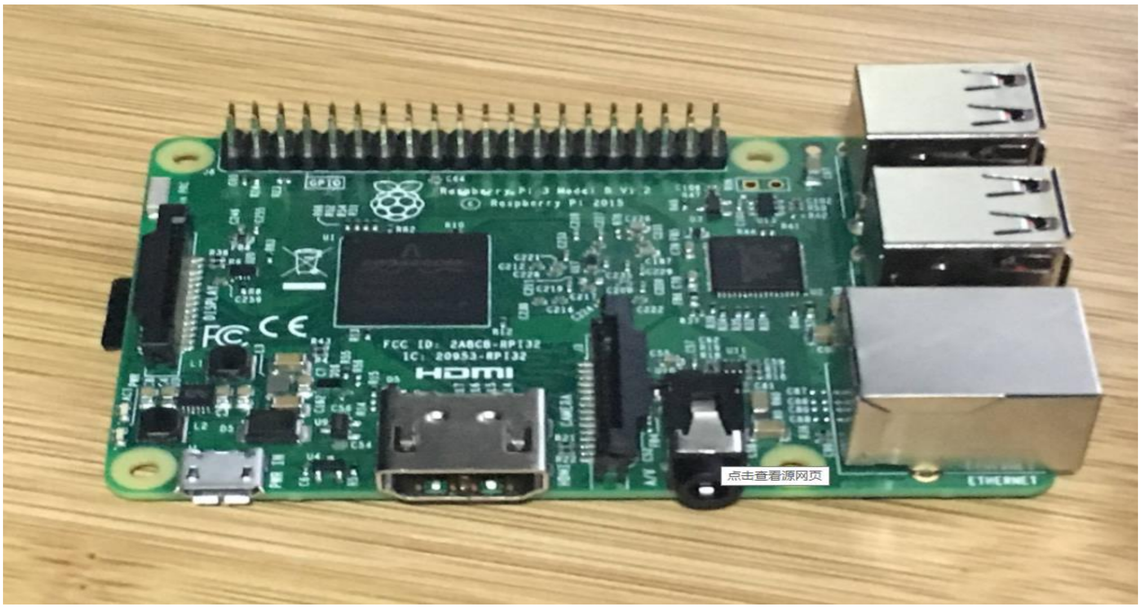


*Figure 11 – Raspberry Pi 3B*

In this project, we choose Raspberry Pi 3B. We need to judge our program and changes through the movement of the robot car, so Raspberry Pi is relatively convenient. Through the preparation of the corresponding program, it is used to control the forward, backward, left and right steering, speed adjustment, program debugging, mobile phone remote control, obstacle avoidance, distance measurement, positioning, vision, tracking, program control, etc.

### 3.3. Hardware Setup

In the present work, three anchors which were mounted on tripod (Figure 12) were considered. The tag was mounted on the level of a robot to assure the static and dynamic experiments (Figure 13). The specifications regarding efficient range of indoor positioning is 30 meters and the efficient range of outdoor positioning is 50 meters. Through manual measurement, the ground length of the laboratory is about 6 meters, and the ground width is about 4.5 meters. According to UWB's indoor effective ranging can indicate that our experience in laboratory scenario can be considered appropriate.

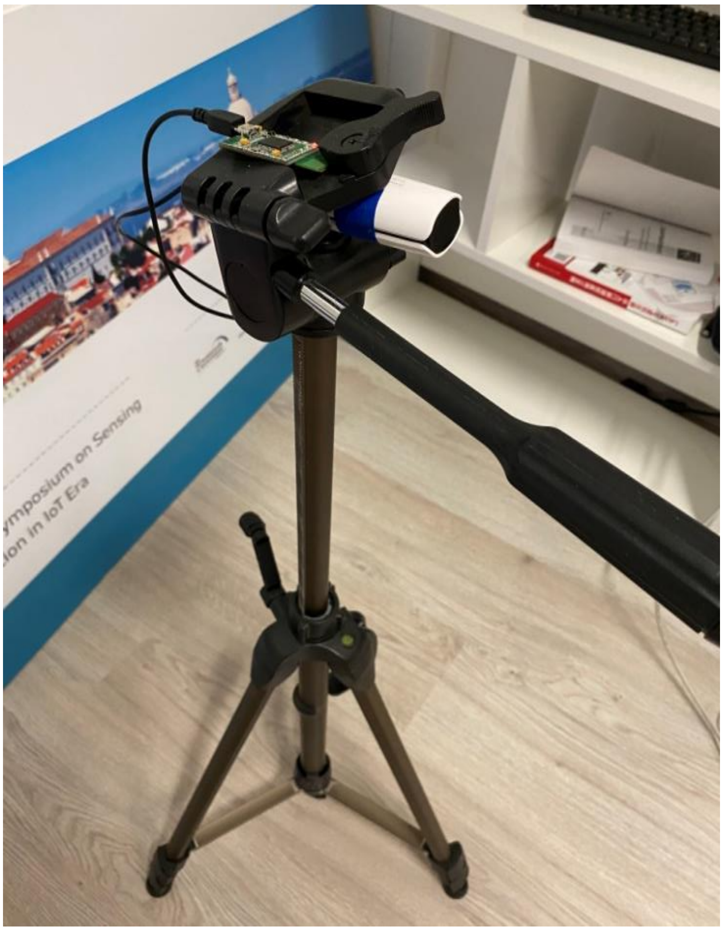


*Figure 12 – UWB mini3 anchor supported by tripod*

The tag is mounted on UDV robot which is operated by PC and the tag is powered by UDV’s own stored power. The implemented setup can be used to determine the trajectory of the moving tag and study the positioning error of UWB.

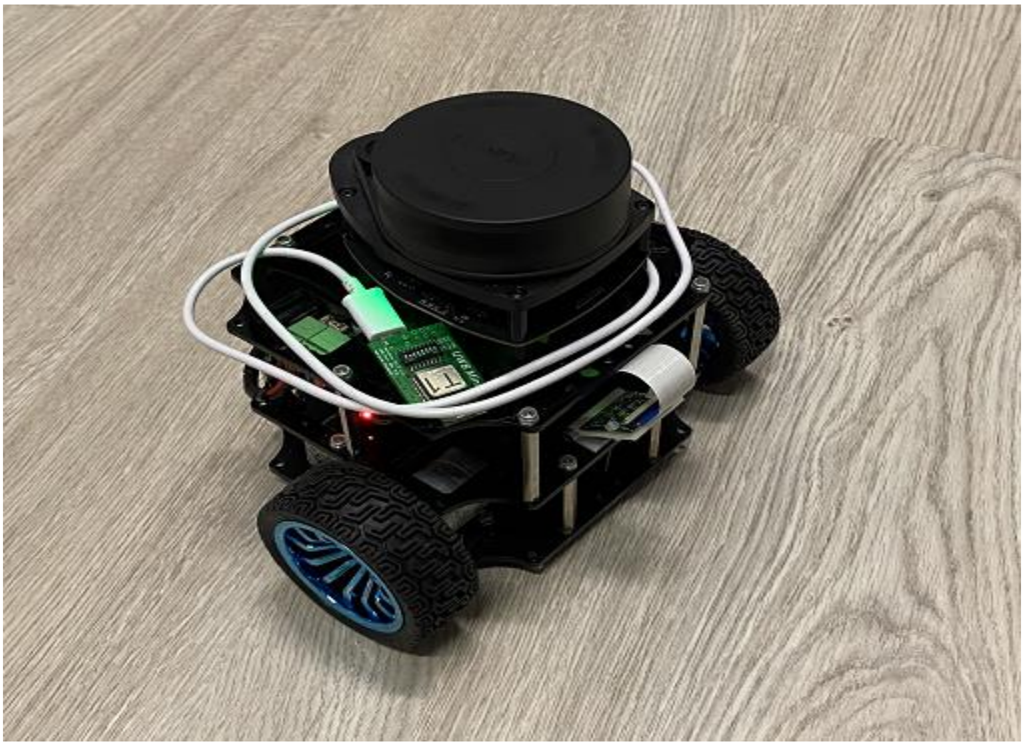


*Figure 13 – UWB mini3 mounded on the robot*

There are no official strict restrictions on the placement of three anchors and one tag. Theoretically speaking, as long as the tag is active in the area covered by the three anchors, the position of the tag can be located. In this project, in order to make the experiment best, we place the three anchors on the three sides of the laboratory, and let the UDV robot equipped with TAG move within the positioning range covered by the three anchors. So far, the UWB positioning system is set up (Figure 14) .

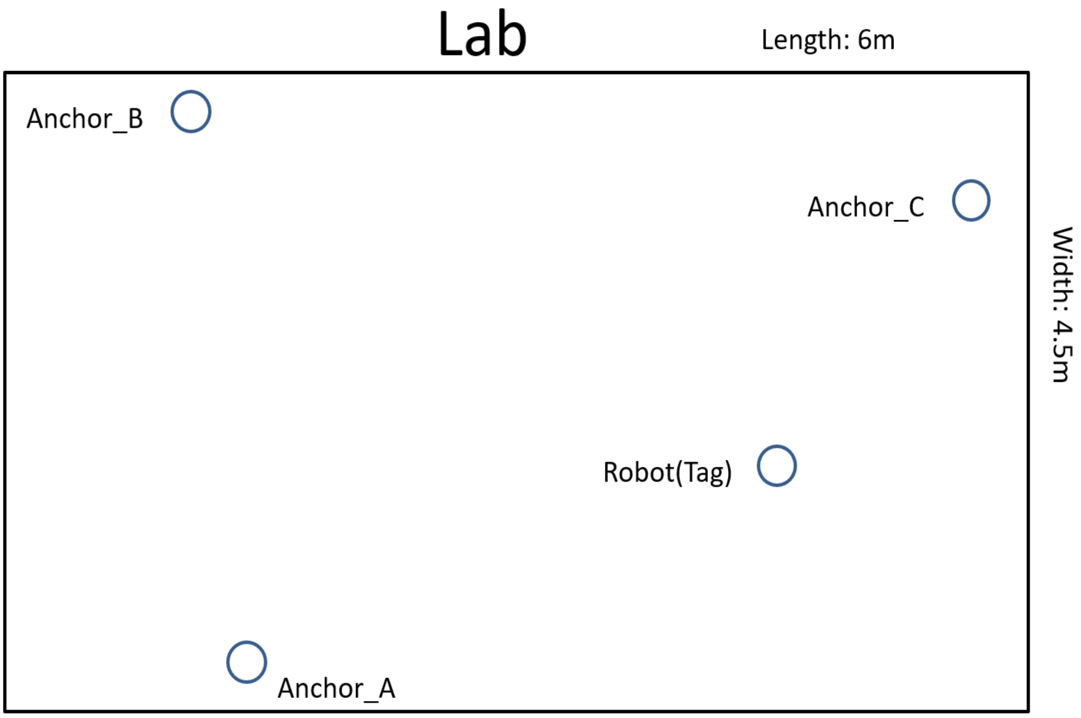


*Figure 14 – Hardware setup environment*

# Chapter 4 – Software: algorithms and implementation

Today's existing UWB technology is based on several traditional positioning algorithms. In my research, I want to explore the feasibility of Kalman filtering algorithm and Markov algorithm for UWB positioning,  and use Matlab to complete the simulation on the basis of a large number of existing theoretical studies.

## 4.1. Original algorithm

There are four classic positioning methods which include: Time of Arrival (TOA) [17], Angle of Arrival(AOA) [18],Time Difference of Arrival (TDOA) [19], Time of Flight(TOF) [20].

### 4.1.1 TOA algorithm

Among classical algorithms, TOA algorithm [21] is the most classic and the most widely used algorithm. In this project, the original UWB mini3 that was used is worked under TOA algorithm. The TOA algorithm is used to calculate the distance between the tag and the anchor, so that the position coordinates of the tag can be obtained through the three-sided distance.

It is assumed that under perfect conditions, tag and anchor have synchronized time standards. At a certain moment, (Figure 15) anchor_C sends a signal to tag, records the time when tag receives the signal, and calculates the time difference based on this. Multiply the time difference by the speed of light to get the distance from anchor_C to tag, and use the same method to get the distances from three anchors to tag respectively. Use the three distance values as the radius to draw a circle, and you can get the intersection of the three circles, that is the coordinate position of the tag.

①：distance=speed of light ×time/ 2

②：On the XY level, 3 circles can determine a point

③：In XYZ space, 4 circles can determine a space point

$$r_A = \sqrt{(x - x_1)^2 + (y - y_1)^2} \tag{4-1}$$

$$r_B = \sqrt{(x - x_2)^2 + (y - y_2)^2} \tag{4-2}$$

$$r_C = \sqrt{(x - x_3)^2 + (y - y_3)^2} \tag{4-3}$$

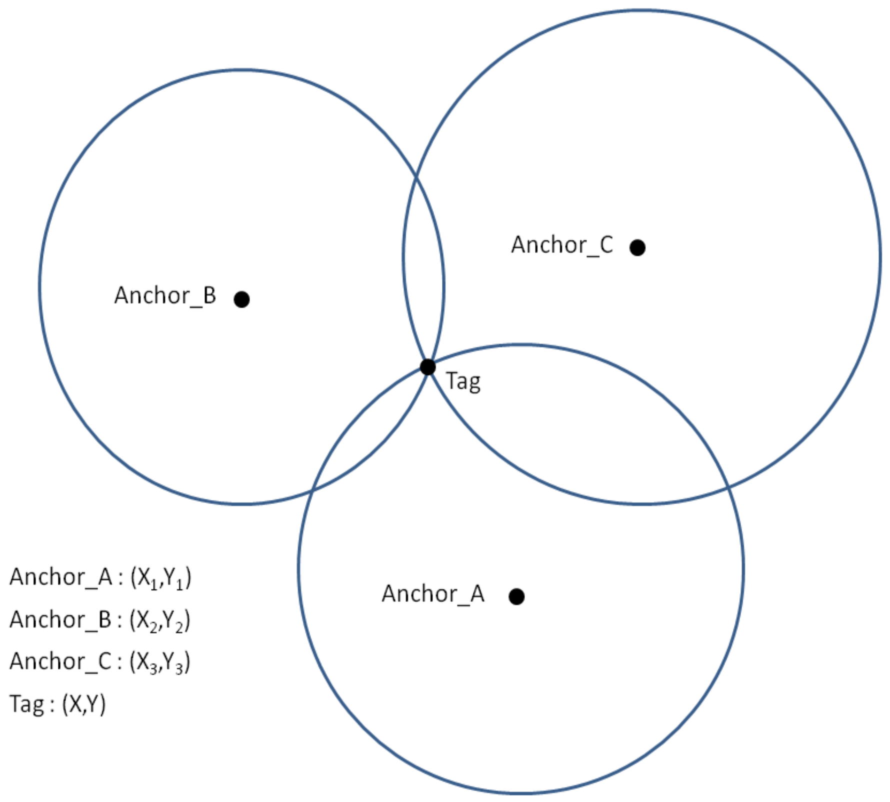


*Figure 15 – TOA algorithm theory*

In the case of $r_A$ ,$r_B$ , $r_C$ obtained by calculation(4-1,4-2,4-3), draw a circle with each anchor as the center and $r_A$ ,$r_B$ , $r_C$ as the radius to determine the position of the tag.

However， the problems of NLOS existing in warehouse environment will greatly affect UWB indoor positioning accuracy. The main factors that affect the positioning accuracy are conductive objects such as the logistics machines, cement walls, metal shelves, or workers. These obstacles may block the transmission of pulse signals and interfere with positioning accuracy. To increase the accuracy different algorithms were considered and following discussed.

### 4.1.2 TOF algorithm

TOF algorithm [22] is a two-way ranging technology, which calculates the distance by measuring the flight time of the UWB signal between the anchor and the tag. According to the mathematical relationship, the distance from a point to a known point is a constant, then this point must be on a circle with the known point as the center and the constant as the radius. If there are two known points, there will be two intersection points. Make three circles with three known points and distances, and they intersect at the same point, which is the location of the tag. The TOF positioning method requires round-trip communication between the anchor and the tag, so the TOF power consumption is greatly increased, and the battery life is relatively short.

TOF is divided into two types according to [23] : Single-sided Two-way Ranging and Double-sided Two-way Ranging.

① Single-sided Two-way Ranging:

Single-sided Two-way Ranging is a simple measurement of the time of a single round-trip message. Device A actively sends data to device B, and device B returns data to device A in response. The propagation time of the data is $T_{prop.}$ As shown below:

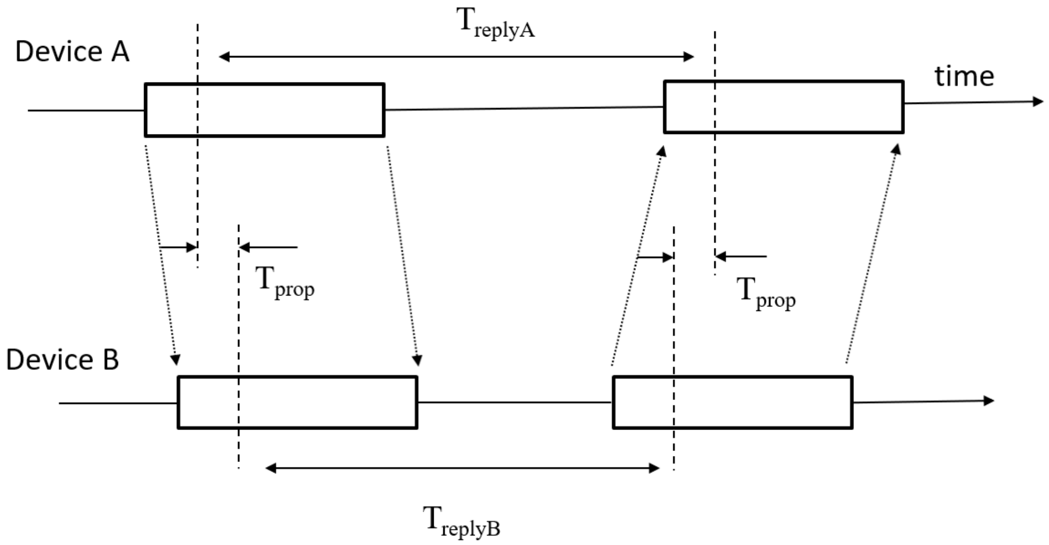


*Figure 16 – Single-sided Two-way Ranging theory*

The process of unilateral two-way ranging is as follows: device A actively sends data while recording the sending timestamp, and device B records the receiving timestamp after receiving it; Delay $T_{replyB}$ , device B sends data and records the sending timestamp at the same time, and device A receives data and records the receiving timestamp at the same time.

So we get two time difference data, the time difference $T_{replyA}$ of device A and the time difference $T_{replyB}$ of device B, and finally the propagation time of the data $T_{prop}$ of the wireless signal can be obtained.

$$T_{prop} = \frac{1}{2}(T_{replyA} - T_{replyB}) \qquad (4-4)$$

The two difference times are calculated based on the local clock. The local clock error can be offset, but there will be slight clock offsets between different devices. Assuming that the clock offsets of devices A and B are $e_A$ and $e_B$ respectively, Then the flight time measurement value is:

$$\widehat{T}_{prop} = \frac{1}{2}(T_{replyA}（1+ e_A） - T_{replyB}（1+ e_B）) \qquad (4-5)$$

So the ranging error is as follows:

$$Error = \widehat{T}_{prop} - T_{replyB} = \frac{1}{2}(T_{replyA} \times e_A - T_{replyB} \times e_B) = T_{prop} \times e_A + \frac{1}{2}T_{replyB}（e_A - e_B） \qquad (4-6)$$

Because $T_{replyB}$》 $T_{prop,}$ can be simplified to get:

$$Error = \widehat{T}_{prop} - T_{replyB} \approx \frac{1}{2}T_{replyB}（e_A - e_B） \qquad (4-7)$$

It can be seen that with the increase of T and clock offset, the error of flight time will increase, which will make the ranging inaccurate. Therefore, Single-sided Two-way Ranging is not commonly used, but for specific applications, if the accuracy requirements are not very high, but a shorter ranging time is required. T is not only the time it takes for device B to receive the transmission, but also the time it takes to load and send data (in addition to supporting positioning, UWB can also transmit data. The standard can load 128 bytes, and the extended mode can load 1024 bytes. data).

② Double-sided Two-way Ranging:

Double-sided Two-way Ranging is an extended ranging method of unilateral two-way ranging as well as reduce ranging error.

Double-sided Two-way Ranging is divided into two ranging. Device A actively initiates the first ranging message, and device B responds. When device A receives the data, it returns the data to device B. The propagation time of the data is $T_{prop}$ , the time difference between device A and device B sending and receiving signal is defined as $T_{reply.}$ Finally, the following four time differences can be obtained: $T_{replyA1,}$ $T_{replyA2,}$ $T_{replyB1,}$ $T_{replyB2.}$ As shown below:

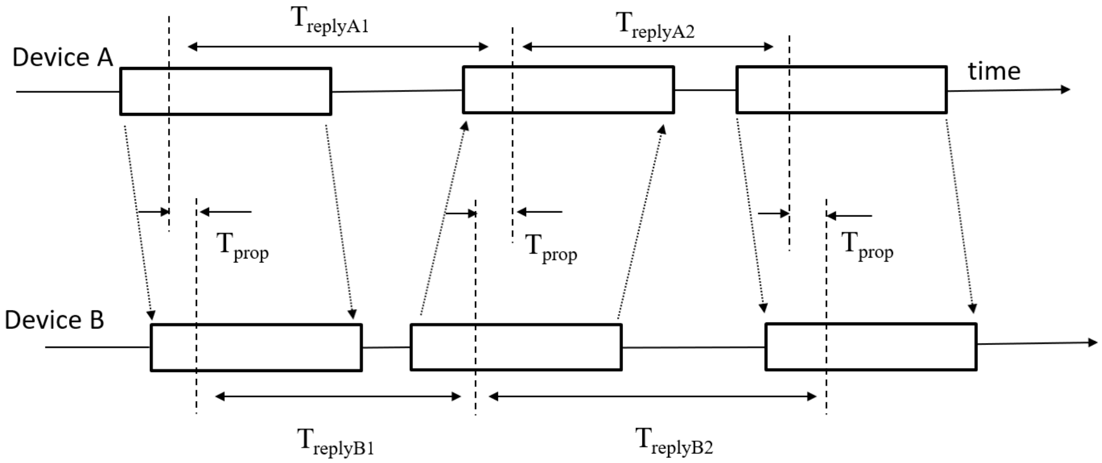


*Figure 17 – Double-sided Two-way Ranging theory*

Consistent with (4-4) calculation principle:

$$T_{prop} = \frac{1}{2}(T_{replyA1} - T_{replyB1}) \tag{4-8}$$

$$T_{prop} = \frac{1}{2}(T_{replyA2} - T_{replyB2}) \tag{4-9}$$

Combine (4-8) and (4-9):

$$T_{replyA1} \times T_{replyB2} = (2T_{prop} + T_{replyB1})(2T_{prop} + T_{replyA2}) \tag{4-10}$$

$$T_{prop} = \frac{TreplyA1 \times TreplyB2 - TreplyB1 \times TreplyA2}{TreplyA1 + TreplyA2 + TreplyB1 + TreplyB2} \tag{4-11}$$

The error of $T_{prop}$ is consistent with (4-5) calculation principle, finally deduced as:

$$Error \approx \frac{eA+eB}{2}\hat{T}_{prop} \tag{4-12}$$

It can be concluded that the error is related to clock drift and flight time.

### 4.1.3 TDOA algorithm

One of the most commonly used positioning methods in UWB positioning technology is TDOA. The UWB positioning tag sends a UWB signal to the outside, and all anchors within the wireless coverage of the tag will receive the wireless signal. If any two anchors with known coordinate points receive the signal, and the distance between the tag and the two anchors is different, then the time interval when these two anchors receive the signal is different.

The TDOA  positioning algorithm described in [24],  uses the time difference of the positioning signals of the same tag received by different anchors to calculate the distance difference d from the tag to different anchors, and thus the hyperbola between the anchors can be obtained. The intersection of the hyperbola is the location of the tag which shown in Figure 18.

Suppose that the time from the measured tag to the nth anchor receiving the UWB signal from the tag is $t_i$ (i=1,2,3,4...n), and suppose the distance $r_i$ from the tag to the nth anchor ( i=1,2,3,4...n)

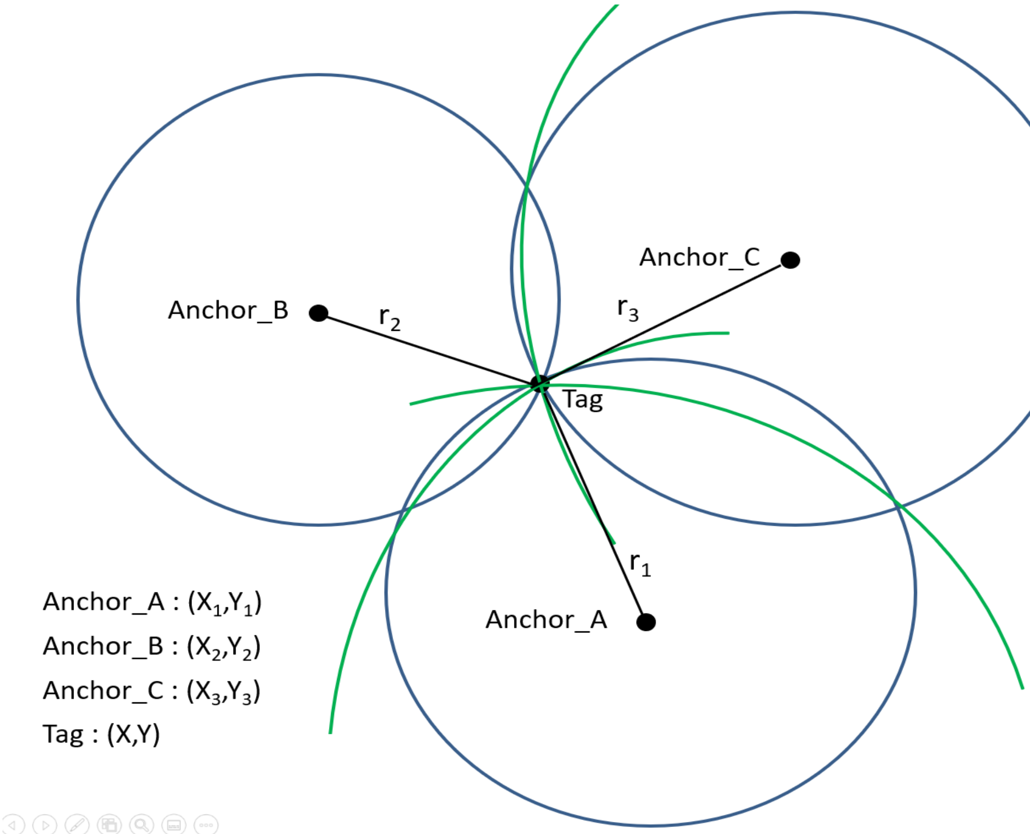


*Figure 18 –Tag localization based on TDOA algorithm diagram*

Define the distance between anchor_A and tag is $r_{1,}$ the distance between anchor_B and tag is $r_{2,}$ the distance between anchor_C and tag is $r_{3.}$ In the case of complete synchronization between anchors, define the distance differences $d_{12}$ , $d_{13}$, $d_{23}$ of the positioning tag relative to the three positioning anchors is obtained as:

$$d_{12} = r_1 - r_2 = (t_1 - t_2) \times c \tag{4-13}$$

$$d_{13} = r_1 - r_3 = (t_1 - t_3) \times c \tag{4-14}$$

$$d_{23} = r_2 - r_3 = (t_2 - t_3) \times c \tag{4-15}$$

$t_1$, $t_2$, $t_3$ are respectively the time interval when the anchor_A, anchor_B, anchor_C receives the signal from the tag. And c is speed of light.

Multiple TDOA measurement values can be used to form a hyperbolic equation set about the tag position, and as shown in figure 18 , the tag coordinates can be obtained by solving this equation set.

$$d_{12} = \sqrt{(x - x_1)^2 + (y - y_1)^2} - \sqrt{(x - x_2)^2 + (y - y_2)^2} \tag{4-16}$$

$$d_{13} = \sqrt{(x - x_2)^2 + (y - y_2)^2} - \sqrt{(x - x_3)^2 + (y - y_3)^2} \tag{4-17}$$

$$d_{23} = \sqrt{(x - x_2)^2 + (y - y_2)^2} - \sqrt{(x - x_3)^2 + (y - y_3)^2} \tag{4-18}$$

The TDOA algorithm is based on the improvement of the TOA algorithm, but it requires strict synchronization of the time of each reference node. In order to improve the positioning accuracy, the signals received by different reference nodes can also be used to perform correlation operations to obtain the TDOA value, and then perform positioning calculations. However, TDOA does not require reciprocating communication between the UWB positioning tag and the positioning anchor. It only needs the positioning tag to transmit the UWB signal once. The working time is shortened and the power consumption is greatly reduced, so it can achieve higher positioning dynamics and positioning capacity.

### 4.1.4 AOA algorithm

The AOA algorithm described in [25] is a positioning algorithm based on the angle of arrival of the signal. It is a typical positioning algorithm based on ranging. It senses the direction of arrival of the signal from the transmitting node through some hardware devices (such as DWM1000) , and calculates the relative position or position between the receiving node and the anchor node.

AOA positioning is generally based on the phase difference method to calculate the angle of arrival, generally not used alone, because AOA involves the problem of angular resolution, if the AOA positioning alone, if the farther away from the base station, the positioning accuracy will be worse.

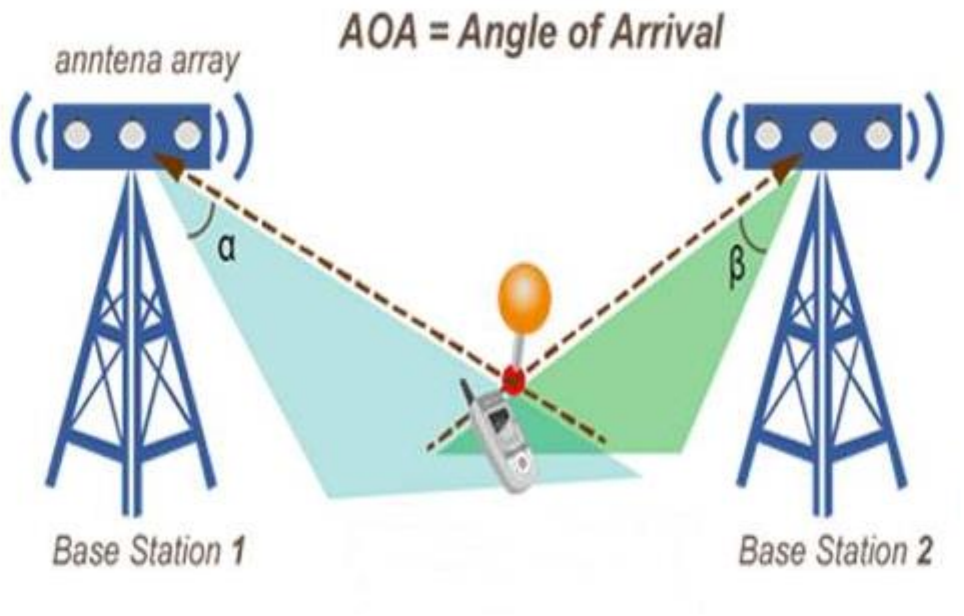


*Figure 19 – The AOA algorithm diagram*

### 4.2.Kalman filter algorithm

The signal transmission between tag and anchor is easily affected by NLOS so that the obtained distance value has an error. The original UWB system will get fluctuating and unstable distance values. If the existing TOA algorithm is used to draw a circle, it will make the three circles intersect into an area instead of a point.

Therefore, the algorithm that can filter interference and optimize the data set is very suitable for the study of reducing errors. In this project, Kalman filter algorithm is deeply researched.

Kalman filtering has been widely used for more than 30 years, including robot navigation, control, sensor data fusion and even military radar systems and missile tracking. In recent years, it has been applied to computer image processing, such as face recognition, image segmentation, image edge detection and so on.

Kalman filtering [26] is suitable for estimating the optimal state of a dynamic system. Even if the observed system state parameters contain noise and the observed values are not accurate, Kalman filtering can complete the optimal estimation of the true state values.

In actual application, we often obtain individual observations discretely (either intermittent measurement or offline data sampled by AD), then first establish a measurement system model and use the initial value (current value) to estimate the next State/value, while comparing the actual measured value of the next state, we know that the estimated value may be inaccurate (the state of the measured object cannot be immutable), and the measured value may not have perfect accuracy (noise exists), so which one is more reliable? More, is it estimated or measured? Then the Kalman filter is to solve the weight problem (uncertainty problem) of the two, calculate a result to make it closer to the true state of the measured object, and recurse in turn.

Kalman estimation consists of two processes: prediction and correction. In the prediction stage, the filter uses the estimation of the previous state to make a prediction of the current state. In the correction stage, the filter uses the observed value of the current state to correct the predicted value obtained in the prediction stage to obtain a new estimate that is more in line with the true value.

Kalman filtering algorithm can be divided into three parts: prediction, calibration, update the covariance estimate. The specific calculation method [27] is following description:

- prediction:

$$\widehat{X}_k' = A\,\widehat{X}_{k-1}+Bu_k \tag{4-19}$$

$$P_k' = A\,P_{k-1}A^T+Q \tag{4-20}$$

- calibration:

$$\widehat{Z}_k = Z_k-H\,\widehat{X}_k' \tag{4-21}$$

$$K_k = P_k'H^T(H\,P_k'H^T+R)^{-1} \tag{4-22}$$

$$\widehat{X}_k = \widehat{X}_k'+ K_k\,\widehat{Z}_k \tag{4-23}$$

Update the covariance estimate:

$$P_k = (I- K_kH)\, P_k' \tag{4-24}$$

| | |
|---|---|
| $X_k$ | Real value |
| $\hat{X}_k$ | Kalman estimated value |
| $P_k$ | Kalman estimation error covariance matrix |
| $\hat{X}_k'$ | Predictive value |
| $P_k'$ | Prediction error covariance matrix |
| $K_k$ | Kalman gain |
| $\hat{Z}_k$ | Measurement margin |
| H | H is the parameter of the measurement system, for multi-measurement system, H is the matrix |
| Q | White Gaussian Noise (System interference) |
| A | System parameters, for multi-model systems, it is a matrix |
| B | System parameters, for multi-model systems, it is a matrix |
| I | Identity matrix |
| R | White Gaussian Noise (System interference) |

*Table 3 – Definition*

For the application of the above Kalman filter principle, it will be shown in Figure 23 in the matlab program part.

Simplify the complicated mathematic (4-19)-(4-24), the core idea of Kalman filtering is to continuously correct errors. For instance, define observation value is Z, and the estimated value is X, then the new estimated value should be

$$Xnew = X + K\,(Z-X) \tag{4-25}$$

This will ensure that the new estimate must be more accurate than the current one. Once it calculates again, it will become more and more accurate. The difficulty is how to determine the value of Kalman gain K, which needs to be continuously tested.

### 4.3.Markov algorithm

Markov chain is a random process in a state space. This process requires a “no memory” property: the probability distribution of the next state can only be determined by the current state. There have been many studies based on Markov chains in recent years [28].

It is used to indicate the random probability of ranging from state i to state j after N times, and is used to indicate the probability that the ranging situation of the Nth cycle is exactly at state i.

$$\pi^{N}= \pi^{N-1}.\ P= \pi^{0}.\ P^{N} \tag{4-26}$$

If we know the initial state vector and probability matrix of the predicted state, we can get the vector of the next state. It is assumed that a Markovian chain as Figure 20 .

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \\ P_{41} & P_{42} & P_{43} & P_{44} \end{bmatrix}$$

*Figure 20 – Random probability*

According to the nature of the Markov chain state transition matrix, Pij (i,j=1,2,3,4) meaning can be explained intuitively. And there are known argumentations [29] to get the following conclusions:

$$Pij > 0, \sum_{j=1}^{4} Pij = 1 \tag{4-27}$$

Taking into account the common interferences in the actual warehouse: metal, board, worker, we selected the above obstacles to simulate the warehouse environment in the laboratory.

In order to effectively eliminate NLOS interference, was used the reliable blocking probability to "predict" possible interference medium in advance and perform data compensation. This kind of fuzzy control can be applied to experiments.

As shown Figure 20 , i represents the current state of robot, j represents the new state of robot, when i=1,2,3,4, the current state respectively indicate that the UDV robot affected by nothing, metal, board and worker, when j=1,2,3,4, the new state respectively indicate that the robot affected by nothing, metal, board and worker (Table 4).

So for instance: P11 indicates the probability that when the current state of the robot is without any interference, the next new state is also without any interference; P12 represents the possibility that when the current state of the robot is without any interference, the next new state is affected by metal; P13 represents the possibility that when the current state of the robot is without any interference, the next new state is affected by board; P14 represents the possibility that when the current state of the robot is without any interference, the next new state is affected by worker; P21 represents the possibility that when the current state of the robot is affected by metal, the next new state is without any interference; P22 represents the possibility that when the current state of the robot is affected by metal, the next new state is affected by metal; P23 represents the possibility that when the current state of the robot is affected by metal, the next new state is affected by board; P24 represents the possibility that when the current state of the robot is affected by metal, the next new state is affected by worker; P31represents the possibility that when the current state of the robot is affected by board, the next new state is without any interference; P32 represents the possibility that when the current state of the robot is affected by board, the next new state is affected by metal; P33 represents the possibility that when the current state of the robot is affected by board, the next new state is affected by board; P34 represents the possibility that when the current state of the robot is affected by board, the next new state is affected by worker; P41 represents the possibility that when the current state of the robot is affected by worker, the next new state is without any interference; P42 represents the possibility that when the current state of the robot is affected by worker, the next new state is affected by metal; P43 represents the possibility that when the current state of the robot is affected by worker, the next new state is affected by board; P44 represents the possibility that when the current state of the robot is affected by worker, the next new state is affected by worker.

| new / current | Blocked by nothing | Blocked by metal | Blocked by board | Blocked by worker |
|---|---|---|---|---|
| Blocked by nothing | $P_{11}$ | $P_{12}$ | $P_{13}$ | $P_{14}$ |
| Blocked by metal | $P_{21}$ | $P_{22}$ | $P_{23}$ | $P_{24}$ |
| Blocked by board | $P_{31}$ | $P_{32}$ | $P_{33}$ | $P_{34}$ |
| Blocked by worker | $P_{41}$ | $P_{42}$ | $P_{43}$ | $P_{44}$ |

*Table 4 – State transition*

### 4.4. Software description

The software components for indoor localization system include the firmware deployed on the level of UWB mini3, the robot control software module and data analysis software module. The interaction between different software modules is presented in the Software flowchart (Figure 21)， real-time positioning software-decarangeRTLS under TOA algorithm is responsible for providing the distance value of each anchor and tag and the coordinates of the tag under the TOA algorithm and MATLAB is responsible for storing data above and using Kalman filter algorithm and Markov algorithm to process data.

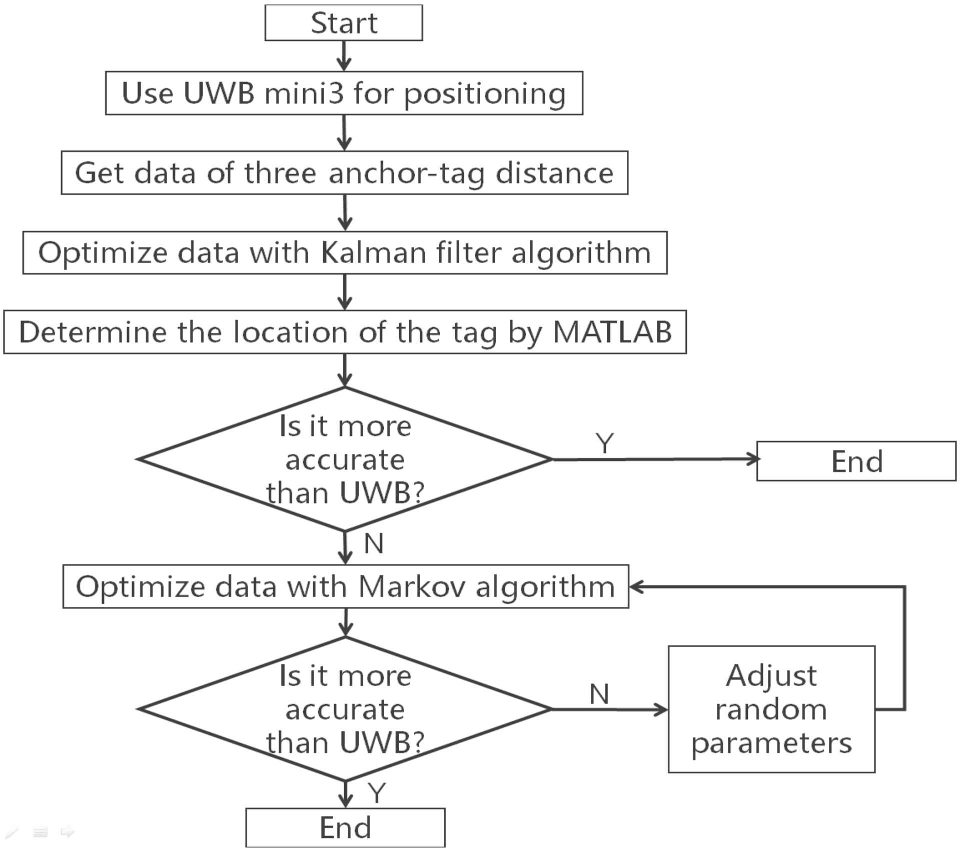
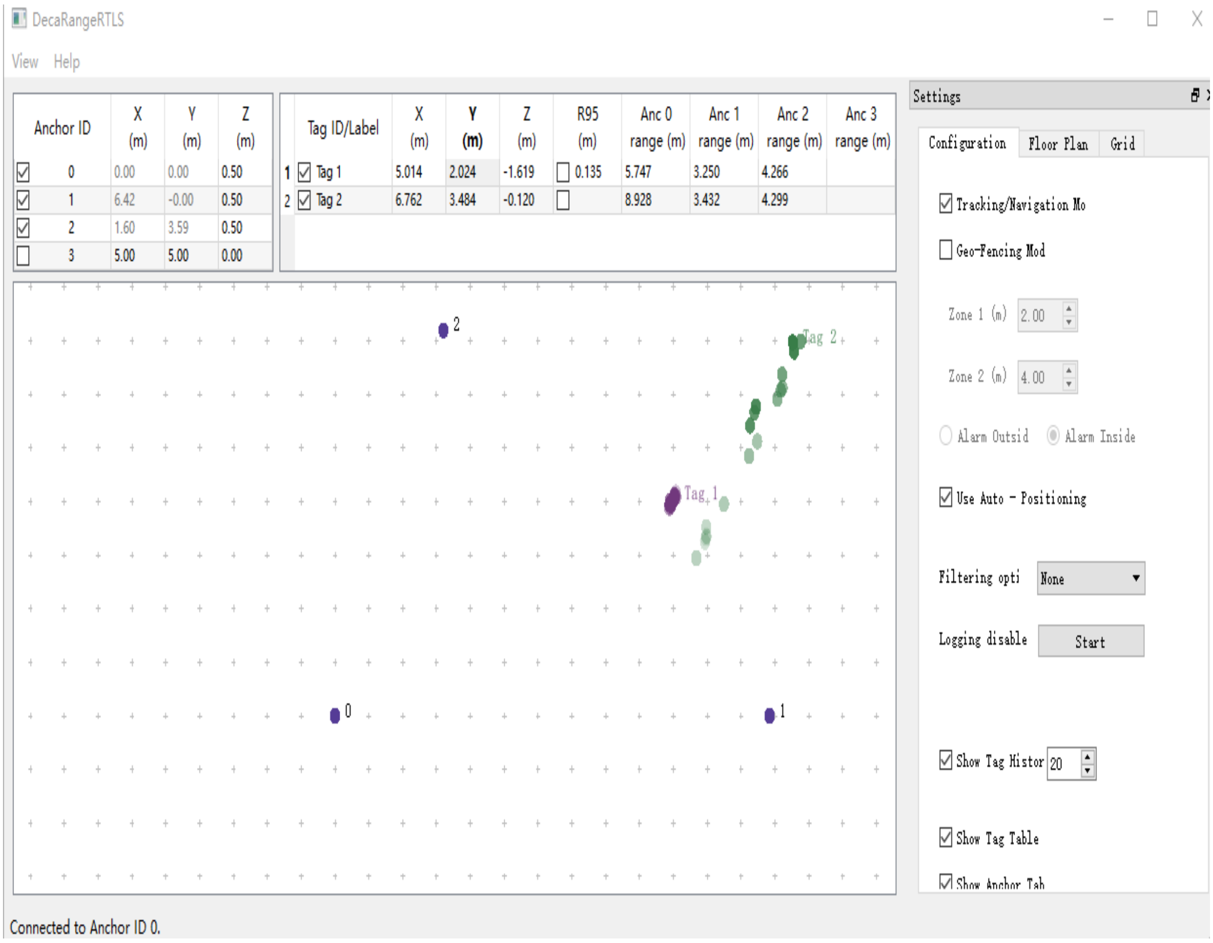

Start
Use UWB mini3 for positioning
Get data of three anchor-tag distance
Optimize data with Kalman filter algorithm
Determine the location of the tag by MATLAB
Is it more accurate than UWB?
Y
End
N
Optimize data with Markov algorithm
Is it more accurate than UWB?
N
Adjust random parameters
Y
End

*Figure 21 – Software modules flowchart*

### 4.4.1 DecarangeRTLS

DecarangeRTLS (Figure 22) is the host computer software, which is connected to the PC through the anchor, which can display the distance value and coordinate position of the anchor and tag in real time.



DecaRangeRTLS

| | Anchor ID | X (m) | Y (m) | Z (m) |
|---|---|---|---|---|
| ☑ | 0 | 0.00 | 0.00 | 0.50 |
| ☑ | 1 | 6.42 | -0.00 | 0.50 |
| ☑ | 2 | 1.60 | 3.59 | 0.50 |
| □ | 3 | 5.00 | 5.00 | 0.00 |

| | Tag ID/Label | X (m) | Y (m) | Z (m) | R95 (m) | Anc 0 range (m) | Anc 1 range (m) | Anc 2 range (m) | Anc 3 range (m) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | ☑ Tag 1 | 5.014 | 2.024 | -1.619 | □ 0.135 | 5.747 | 3.250 | 4.266 | |
| 2 | ☑ Tag 2 | 6.762 | 3.484 | -0.120 | □ | 8.928 | 3.432 | 4.299 | |

☑ Tracking/Navigation Mo

□ Geo-Fencing Mod

Zone 1 (m) 2.00

Zone 2 (m) 4.00

○ Alarm Outsid ● Alarm Inside

☑ Use Auto - Positioning

Filtering opti None

Logging disable Start

☑ Show Tag Histor 20

☑ Show Tag Table

☑ Show Anchor Tab

Connected to Anchor ID 0.

*Figure 22 – DecarangeRTLS*

As shown in Figure 22, the unit is meters. when the coordinates of the three fixed anchors are known to be (0, 0), (6.42,0), (1.6,3.59) respectively, tag1 and tag2 are positioned and displayed in the software decarangeRTLS in real time. The coordinates of tag1 is (5.014,2.024), the coordinates of tag2 is (6.762,3.484), and the distance between tag1 and the three anchors is respectively 5.747m, 3.250m, 4.266m, and the distance between tag2 and the three anchors is respectively 8.928m,3.432m,4.299m.

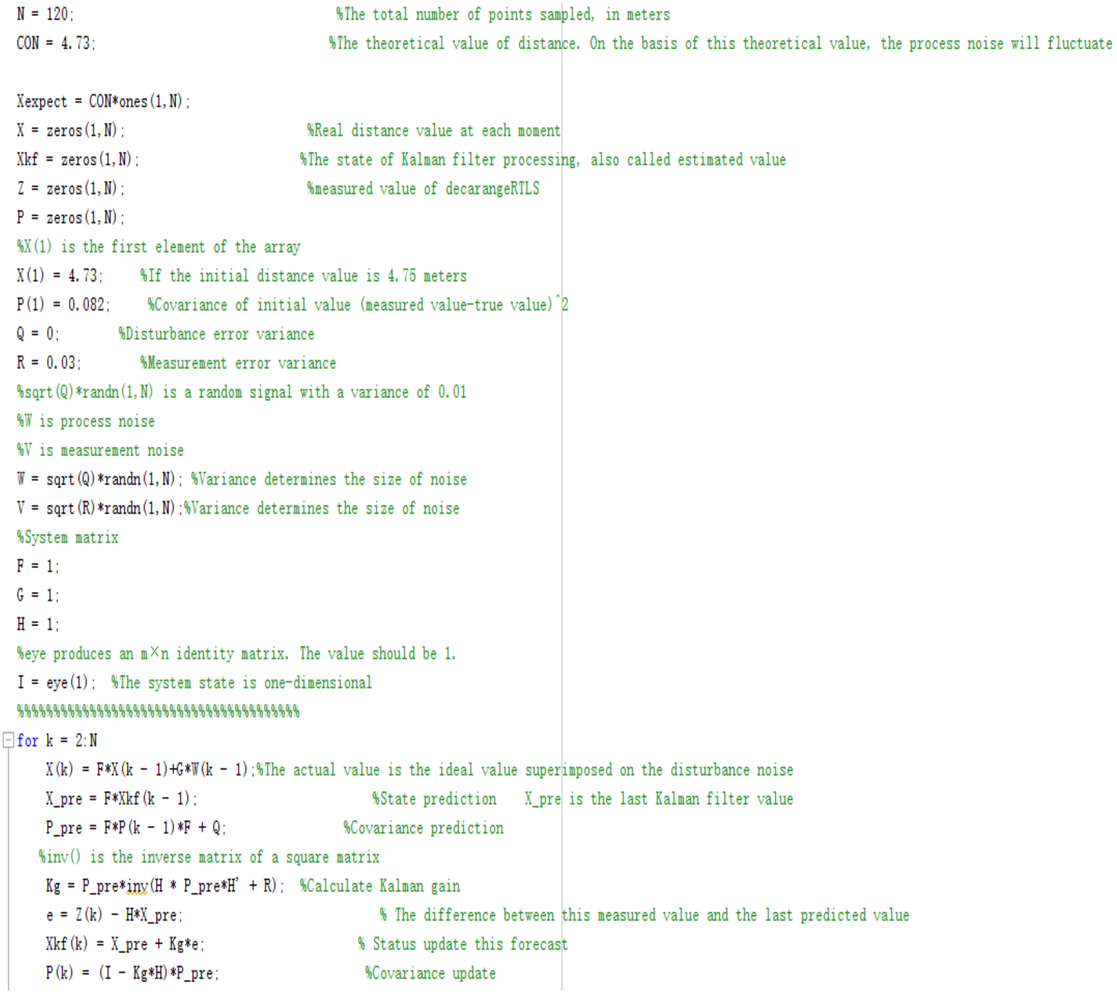
But the big disadvantage of this software is that because the UWB kit used is embedded with the TOA positioning algorithm, the positioning of the tag is very inaccurate, and the coordinates of the tag will fluctuate greatly and cannot be positioned at a fixed point. We will collect the data of distance between tag and three anchors that decarangeRTLS provide, and use developed algorithm to improve the original algorithm—TOA.

### 4.4.2 Matlab Software

Matlab [30] is a software produced by MathWorks in the United States. It is used in data analysis, wireless communication, deep learning, image processing and computer vision, signal processing, quantitative finance and risk management, robotics, control systems and other fields. MATLAB is a combination of the two words matrix&laboratory, which means matrix factory (matrix laboratory).

The software mainly faces the high-tech computing environment of scientific computing, visualization and interactive programming. It integrates many powerful functions such as numerical analysis, matrix calculation, scientific data visualization, and nonlinear dynamic system modeling and simulation in an easy-to-use window environment. It is used for scientific research, engineering design, and many sciences that require effective numerical calculations. The field provides a comprehensive solution, and to a large extent get rid of the editing mode of traditional non-interactive programming languages (such as C, Fortran).

In this project，the simulation of Kalman filter algorithm (Figure 23) and Markov algorithm (Figure 24 and Figure25) was developed in Matlab. The Kalman filter algorithm optimizes the distance values of each anchor and tag under the TOA algorithm collected by DecarangeRTLS.

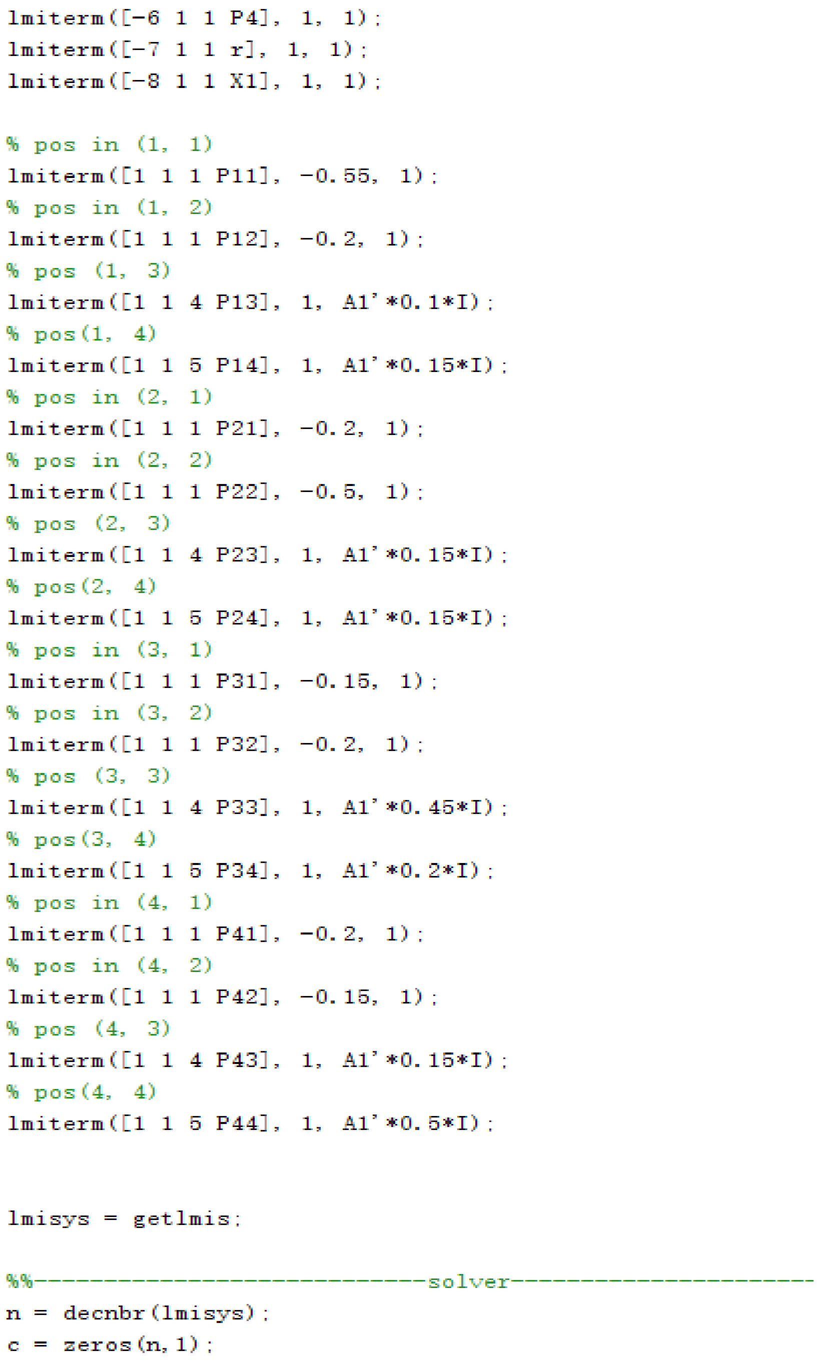


```
N = 120;                                    %The total number of points sampled, in meters
CON = 4.73;                                 %The theoretical value of distance. On the basis of this theoretical value, the process noise will fluctuate

Xexpect = CON*ones(1,N);
X = zeros(1,N);                             %Real distance value at each moment
Xkf = zeros(1,N);                           %The state of Kalman filter processing, also called estimated value
Z = zeros(1,N);                             %measured value of decarangeRTLS
P = zeros(1,N);
%X(1) is the first element of the array
X(1) = 4.73;    %If the initial distance value is 4.75 meters
P(1) = 0.082;    %Covariance of initial value (measured value-true value)^2
Q = 0;        %Disturbance error variance
R = 0.03;       %Measurement error variance
%sqrt(Q)*randn(1,N) is a random signal with a variance of 0.01
%W is process noise
%V is measurement noise
W = sqrt(Q)*randn(1,N); %Variance determines the size of noise
V = sqrt(R)*randn(1,N);%Variance determines the size of noise
%System matrix
F = 1;
G = 1;
H = 1;
%eye produces an m×n identity matrix. The value should be 1.
I = eye(1);  %The system state is one-dimensional
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for k = 2:N
    X(k) = F*X(k - 1)+G*W(k - 1);%The actual value is the ideal value superimposed on the disturbance noise
    X_pre = F*Xkf(k - 1);                       %State prediction    X_pre is the last Kalman filter value
    P_pre = F*P(k - 1)*F + Q;                 %Covariance prediction
   %inv() is the inverse matrix of a square matrix
    Kg = P_pre*inv(H * P_pre*H' + R);  %Calculate Kalman gain
    e = Z(k) - H*X_pre;                           % The difference between this measured value and the last predicted value
    Xkf(k) = X_pre + Kg*e;                      % Status update this forecast
    P(k) = (I - Kg*H)*P_pre;                     %Covariance update
```

*Figure 23 – Kalman filter algorithm program developed in Matlab*

On the basis of Kalman filter optimization, the Markov algorithm uses a random process to “determine” the largest possible interference factors and relocate.

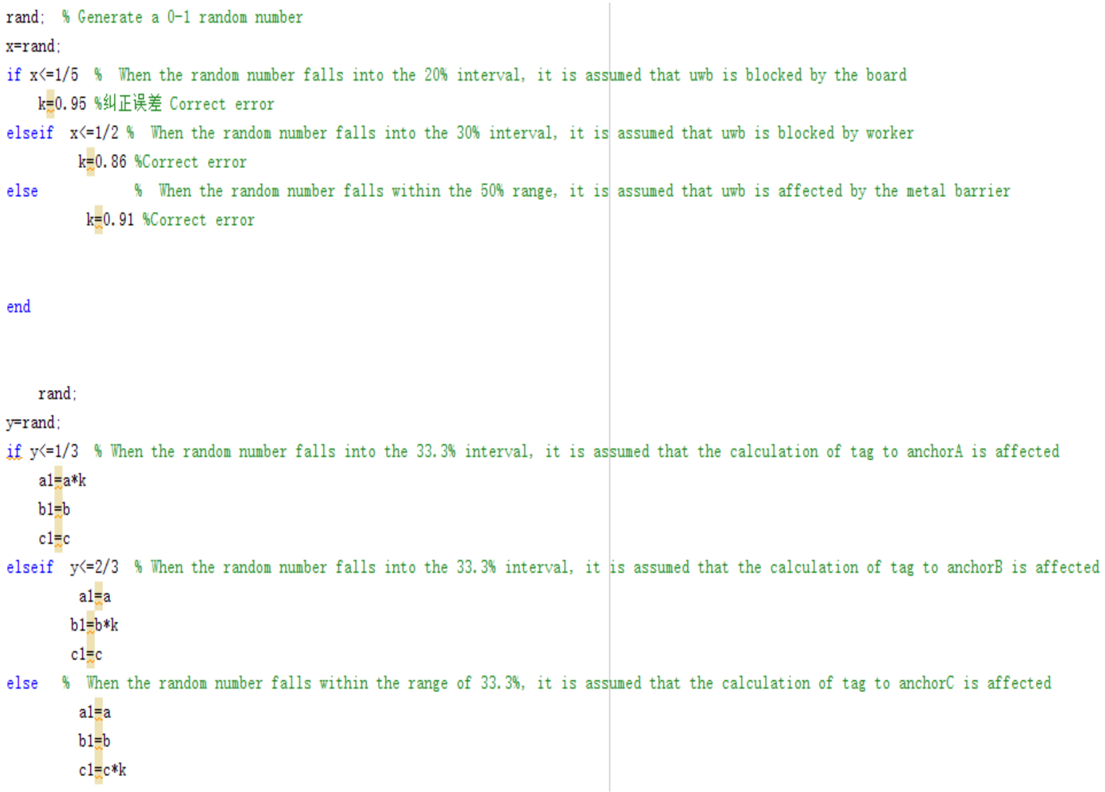
First, take the probability transition matrix ( Figure 35) obtained after the experimental observation and summary into the program for linearization processing ( Figure 24), and determine the maximum possible transition state of the current UDV robot. (That is, what medium is the current state of UDV robot affected, and what medium is the next new state?)



```
lmiterm([-6 1 1 P4], 1, 1);
lmiterm([-7 1 1 r], 1, 1);
lmiterm([-8 1 1 X1], 1, 1);

% pos in (1, 1)
lmiterm([1 1 1 P11], -0.55, 1);
% pos in (1, 2)
lmiterm([1 1 1 P12], -0.2, 1);
% pos (1, 3)
lmiterm([1 1 4 P13], 1, A1'*0.1*I);
% pos(1, 4)
lmiterm([1 1 5 P14], 1, A1'*0.15*I);
% pos in (2, 1)
lmiterm([1 1 1 P21], -0.2, 1);
% pos in (2, 2)
lmiterm([1 1 1 P22], -0.5, 1);
% pos (2, 3)
lmiterm([1 1 4 P23], 1, A1'*0.15*I);
% pos(2, 4)
lmiterm([1 1 5 P24], 1, A1'*0.15*I);
% pos in (3, 1)
lmiterm([1 1 1 P31], -0.15, 1);
% pos in (3, 2)
lmiterm([1 1 1 P32], -0.2, 1);
% pos (3, 3)
lmiterm([1 1 4 P33], 1, A1'*0.45*I);
% pos(3, 4)
lmiterm([1 1 5 P34], 1, A1'*0.2*I);
% pos in (4, 1)
lmiterm([1 1 1 P41], -0.2, 1);
% pos in (4, 2)
lmiterm([1 1 1 P42], -0.15, 1);
% pos (4, 3)
lmiterm([1 1 4 P43], 1, A1'*0.15*I);
% pos(4, 4)
lmiterm([1 1 5 P44], 1, A1'*0.5*I);


lmisys = getlmis;

%%----------------------------solver----------------------
n = decnbr(lmisys);
c = zeros(n,1);
```

*Figure 24 – Linear processing of state transition matrix in Matlab*

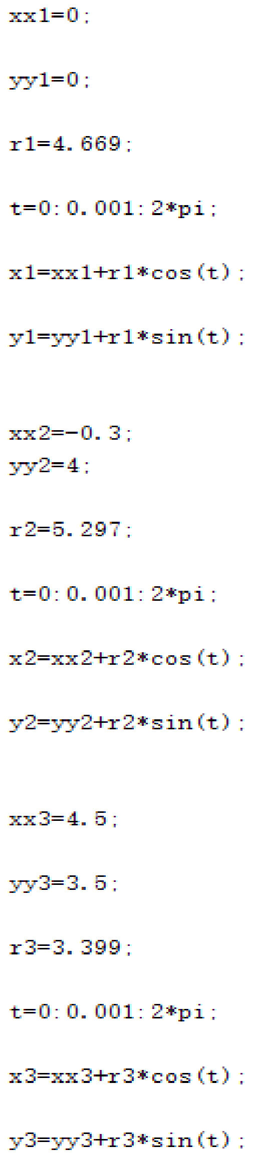
According to the new probability parameter and the interference coefficient obtained in the experimental part, the distance value is compensated ( Figure 25).



```
rand;  % Generate a 0-1 random number
x=rand;
if x<=1/5  %  When the random number falls into the 20% interval, it is assumed that uwb is blocked by the board
    k=0.95 %纠正误差 Correct error
elseif  x<=1/2 %  When the random number falls into the 30% interval, it is assumed that uwb is blocked by worker
         k=0.86 %Correct error
else              %  When the random number falls within the 50% range, it is assumed that uwb is affected by the metal barrier
          k=0.91 %Correct error


end


    rand;
y=rand;
if y<=1/3  % When the random number falls into the 33.3% interval, it is assumed that the calculation of tag to anchorA is affected
    a1=a*k
    b1=b
    c1=c
elseif  y<=2/3  % When the random number falls into the 33.3% interval, it is assumed that the calculation of tag to anchorB is affected
         a1=a
        b1=b*k
        c1=c
else   %  When the random number falls within the range of 33.3%, it is assumed that the calculation of tag to anchorC is affected
         a1=a
         b1=b
         c1=c*k
```

*Figure 25 – Distance value compensation in MATLAB*

Take the new compensated distance value between anchor and the tag as the radius of the circle, and draw a circle from the coordinates of the three known anchors to determine the new coordinates of the tag. The result is shown in Figure 43 of Chapter 6.



```
xx1=0;

yy1=0;

r1=4.669;

t=0:0.001:2*pi;

x1=xx1+r1*cos(t);

y1=yy1+r1*sin(t);


xx2=-0.3;
yy2=4;

r2=5.297;

t=0:0.001:2*pi;

x2=xx2+r2*cos(t);

y2=yy2+r2*sin(t);


xx3=4.5;

yy3=3.5;

r3=3.399;

t=0:0.001:2*pi;

x3=xx3+r3*cos(t);

y3=yy3+r3*sin(t);
```

*Figure 26 – Draw a circle to get new tag coordinates in MATLAB*

### 4.4.3 ROS

ROS is Robot Operating System. It was originally a robotics project of Stanford University. It was later developed by Willow Garage and is currently an open source project maintained by OSRF. ROS is system software that manages computer hardware and software resources and provides common services for computer programs.

ROS [31] uses the concept of Node to represent an application. Different nodes are connected through (Topic), (Service), and (Action) in a pre-defined format. Based on this modular communication mechanism, developers can easily replace and update certain modules in the system; they can also replace individual modules of ROS with their own nodes, which is very suitable for algorithm development. In addition, ROS can be cross-platform and used on different computers, different operating systems, no programming languages, and different robots. Ros features diagram is shown below：

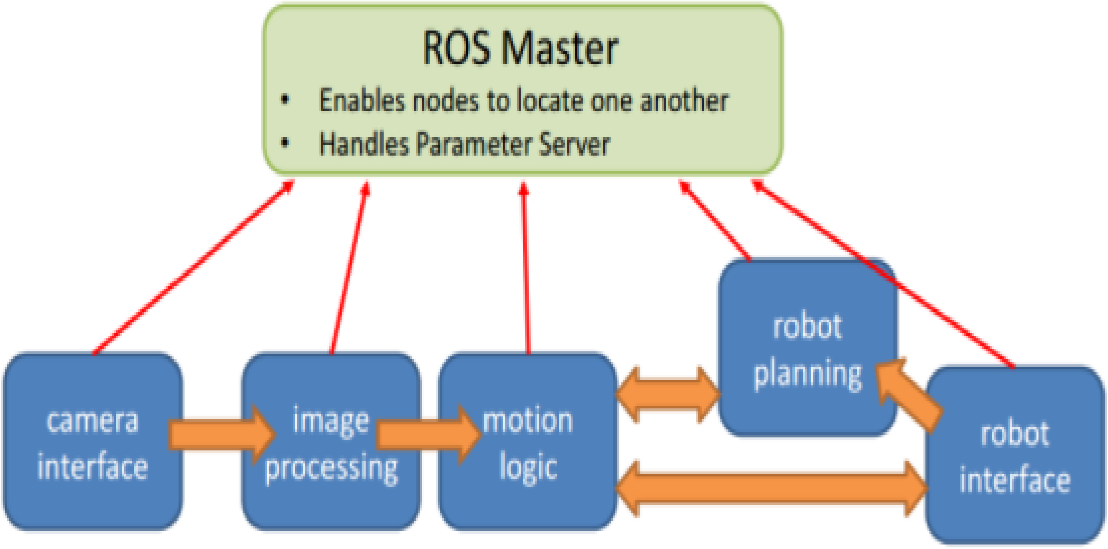


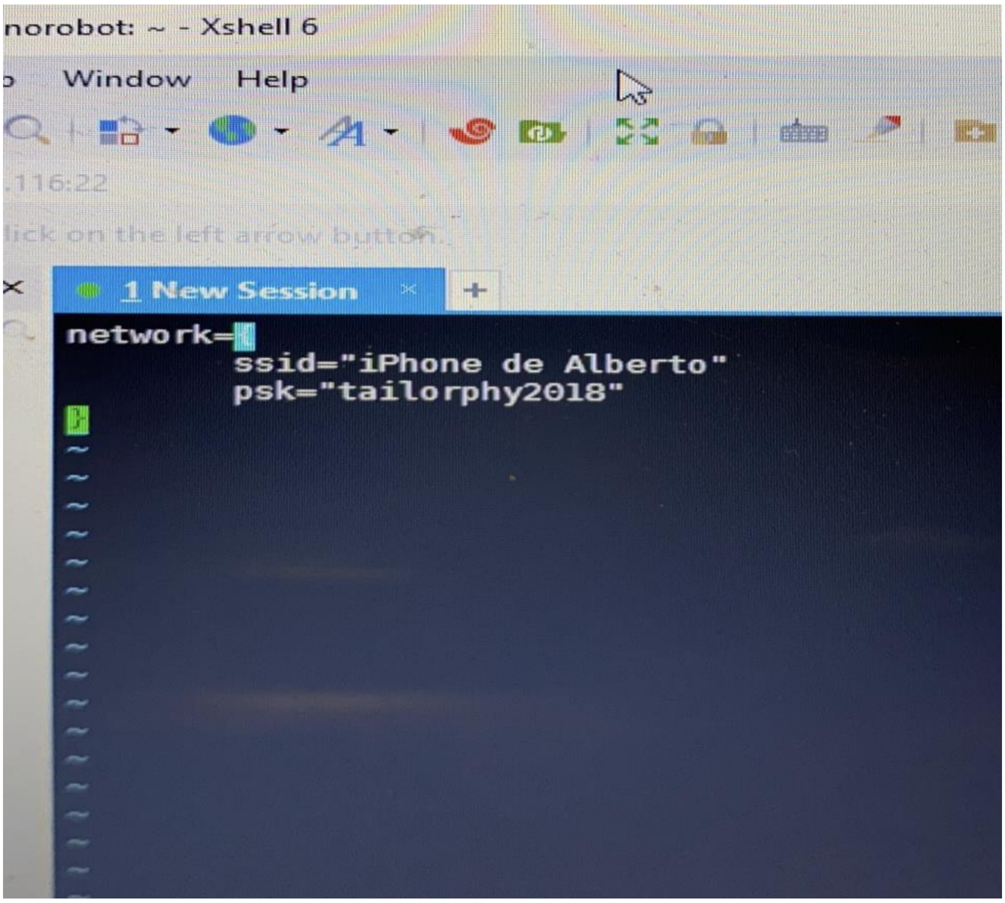
*Figure 27 – ROS features*

### 4.4.4 Xshell6

The key point that distinguishes the robot from other projects is that the controller moves with the robot carrier, and we chose the Raspberry Pi as the controller, which means that we can’t connect the Raspberry Pi to the display like we write other programs. . The solution is to install the terminal simulation software and write a debugging window on the Raspberry Pi, so that we can easily write and debug our robot program on the computer or mobile phone. In this project, we chose Xshell6 as the terminal simulation software.

Xshell [32] is a powerful secure terminal emulation software, which supports SSH1, SSH2, and the TELNET protocol of the Microsoft Windows platform. Xshell's secure connection to the remote host via the Internet and its innovative design and features help users enjoy their work in a complex network environment. Xshell can be used to access servers under different remote systems under the Windows interface, so as to better achieve the purpose of remote control of the terminal.

In order to directly control the movement of the robot on Xshell6, the robot and PC must be connected to the same WIFI network (Figure 28). At this time, we need to configure the Raspberry Pi network of the robot in Xshell6.



```
network={
        ssid="iPhone de Alberto"
        psk="tailorphy2018"
}
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

*Figure 28 – Configure the WIFI network in Xshell6*

After configuring the network of the Raspberry Pi, we can directly use the existing written movement control program in Xshell6, and we can directly control the movement of the robot through the keyboard.

### 4.4.5 CoIDE

CoIDE [33] is a free ARMCortexMCU embedded integrated development environment (IDE) professionally created by CooCox. CoIDE integrates CoBuilder and CoDebugger, which is suitable for compiling, programming and debugging embedded system applications. Suitable for developers of different levels, including professional application development engineers and beginners in embedded software development. It provides a complete integrated development environment for developers of the ARM Cortex-M series, including project management, editing, compilation tools, and debuggers.

In this project, We need to view the original positioning program embedded in the UWB mini3 microcontroller chip and calibrate it. Modify the original positioning code through the corrected linear equation, and recompile it through the CoIDE software. The calibration process will be described in detail in chapter 5.1.

# Chapter 5 – Preparation for Simulation

## 5.1.UWB indoor localization system calibration

Since the place and environments we are using are different, and are interfered by factors such as latitude and longitude.  In order to adapt to the local environment, ensure the accuracy of UWB kit positioning, the UWB module must be calibrated before the UWB system is positioned.

Under normal circumstances, we only need to calibrate once on site, and get the correction coefficient through the ranging of 1 Anchor and 1 Tag. It is not necessary to calibrate each Anchor and Tag.

Through multiple sets of measured actual distances and UWB positioning distances, use Excel software to perform data fitting and generate a fitting formula. There are many fitting formulas, the simplest is the linear equation, because in the Cartesian coordinate system, the representation of any linear equation is a straight line.

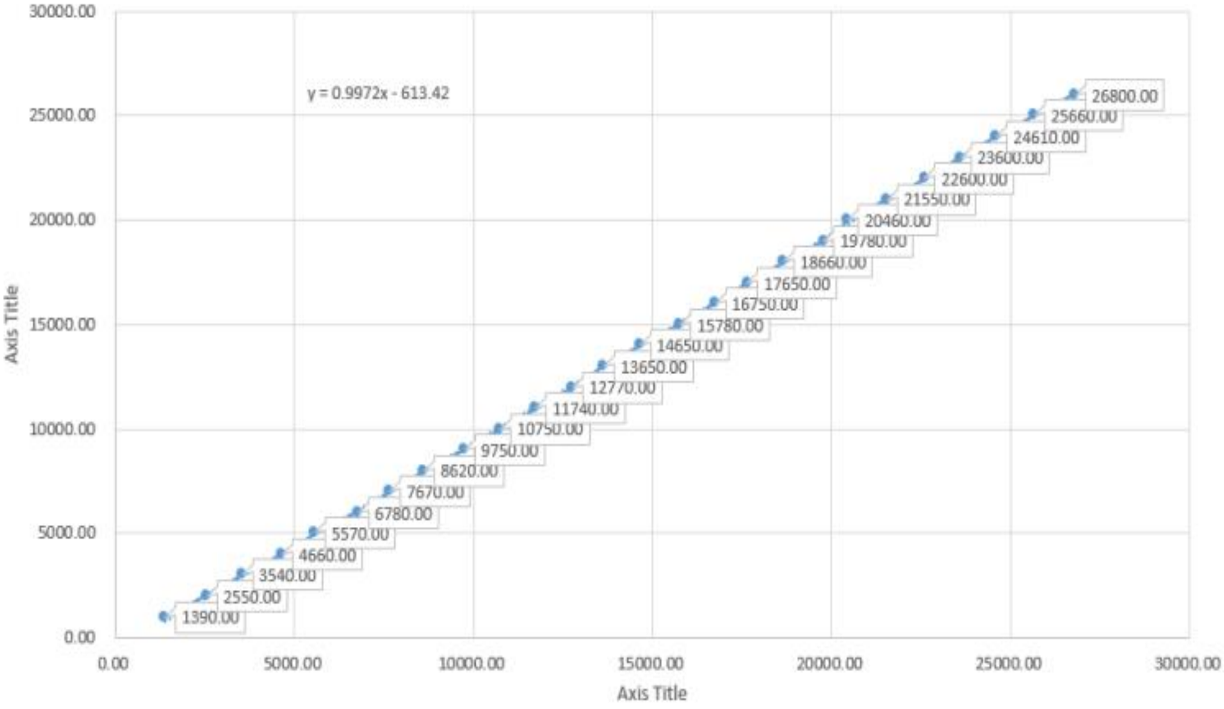
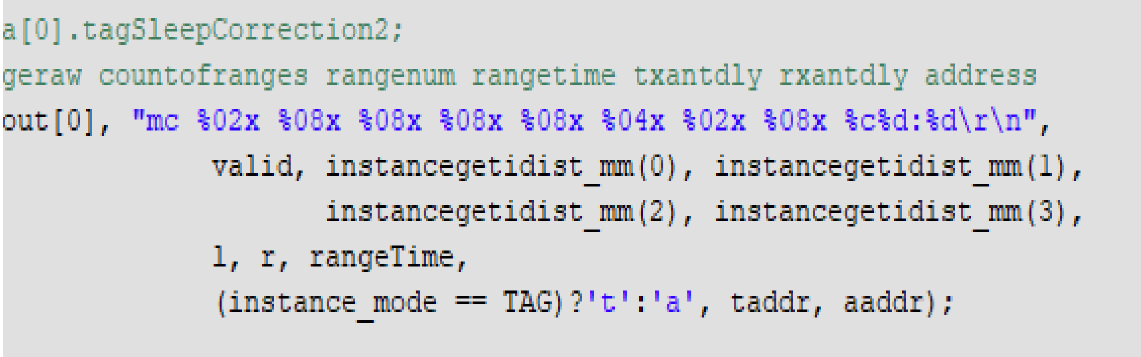


*Figure 29 – Calibration by linear function [mm]*

Just select any one anchor and any one tag for calibration, the X axis represents the distance value between one tag and one anchor obtained by UWB ranging software decarangeRTLS, and the Y axis represents the actual distance of it in millimeters.

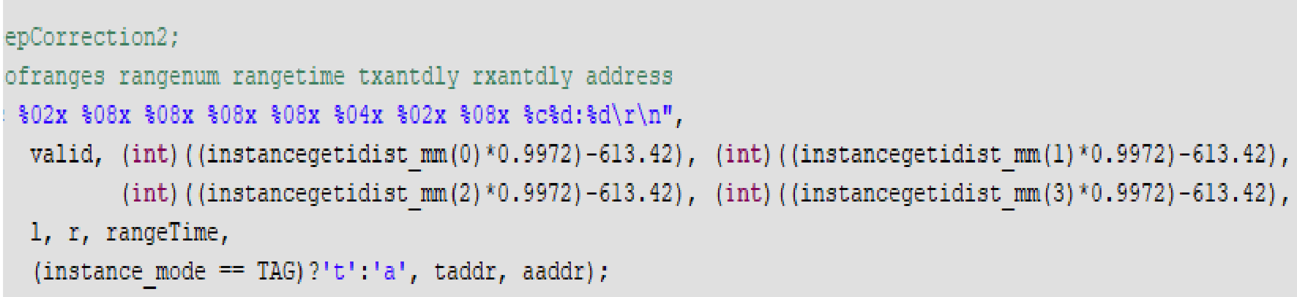
According to the linear function graph, the functional relationship between UWB ranging value and actual distance value is : y=0.9972x-613.42 (Figure 29). According to this linear function, modify the calibration in the original positioning code. The original positioning program embedded in UWB mini3 is :



```
a[0].tagSleepCorrection2;
geraw countofranges rangenum rangetime txantdly rxantdly address
out[0], "mc %02x %08x %08x %08x %08x %04x %02x %08x %c%d:%d\r\n",
                valid, instancegetidist_mm(0), instancegetidist_mm(1),
                        instancegetidist_mm(2), instancegetidist_mm(3),
                l, r, rangeTime,
                (instance_mode == TAG)?'t':'a', taddr, aaddr);
```

*Figure 30 – The original program  embedded in UWB mini3*

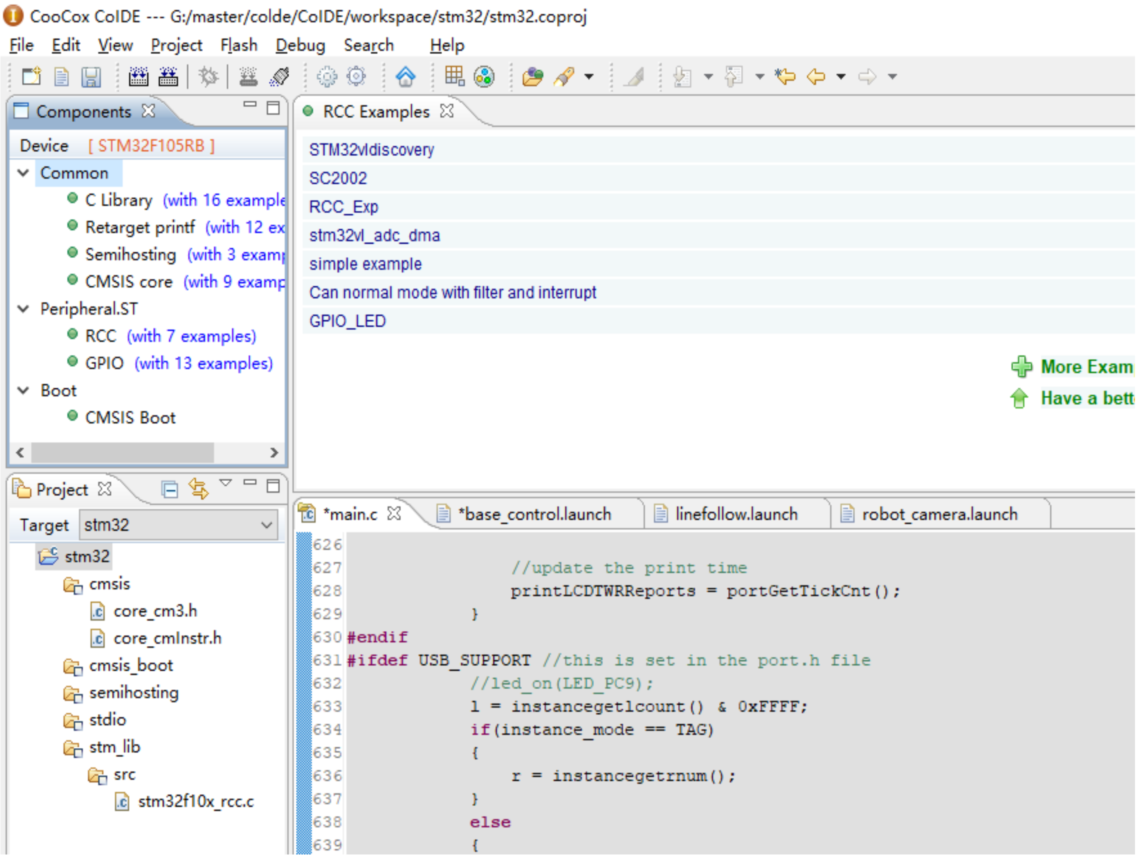
Add the linear functional relationship between UWB ranging value and actual distance value : y=0.9972x-613.42 after the original definition (instancegetidist_mm) of the distance between the anchor and tag obtained, can be calibrated as:



```
epCorrection2;
ofranges rangenum rangetime txantdly rxantdly address
 %02x %08x %08x %08x %08x %04x %02x %08x %c%d:%d\r\n",
  valid, (int)((instancegetidist_mm(0)*0.9972)-613.42), (int)((instancegetidist_mm(1)*0.9972)-613.42),
         (int)((instancegetidist_mm(2)*0.9972)-613.42), (int)((instancegetidist_mm(3)*0.9972)-613.42),
  l, r, rangeTime,
  (instance_mode == TAG)?'t':'a', taddr, aaddr);
```

*Figure 31 – Procedure after calibration*

By recompiling the program through software CoIDE, we only need to download the program to the UWB module connected to the computer, without downloading every module. Through data correction, the distance value measured by the UWB module is more accurate.



```

            //update the print time
            printLCDTWRReports = portGetTickCnt();
        }
#endif
#ifdef USB_SUPPORT //this is set in the port.h file
        //led_on(LED_PC9);
        l = instancegetlcount() & 0xFFFF;
        if(instance_mode == TAG)
        {
            r = instancegetrnum();
        }
        else
        {
```

*Figure 32 – Recompiled by COIDE*

## 5.2. Theory hypothesis

Our goal is to eliminate the interference of NLOS on UWB ranging, and we must first measure the error percentage of the impact of various interference media on ranging. Several simulated scenarios were considered in the laboratory when metal, board, and workers are blocking the signal transmission between the tag and the anchors. The distance values provided by UWB software decarangeRTLS in time are recorded.

When deliberately placed a metal object block the signal transmission line of anchor_A (left one) and tag (right one) as well as adjusted the distance of anchor_A and tag (Figure 33), the distance value displayed in the decarangeRTLS is quite different from the actual distance value.

Recorded distance value between anchor_A and tag displayed in decarangeRTLS and the actual distance value between anchor_A and tag by manual measurement are used as part of linear function that is used to get the interference coefficient.

$$\text{interference coefficient} = 1 - \frac{actual\ distance}{\text{displayed distance value}} \qquad (5\text{-}1)$$

Through linear processing of the actual distance value and the recorded data value, we can get the interference coefficient of metal is 9%. Through the same experiment, we can respectively get the interference coefficient of board and worker is 5% and 14%. These interference coefficients can be used as compensation basis for correcting errors.

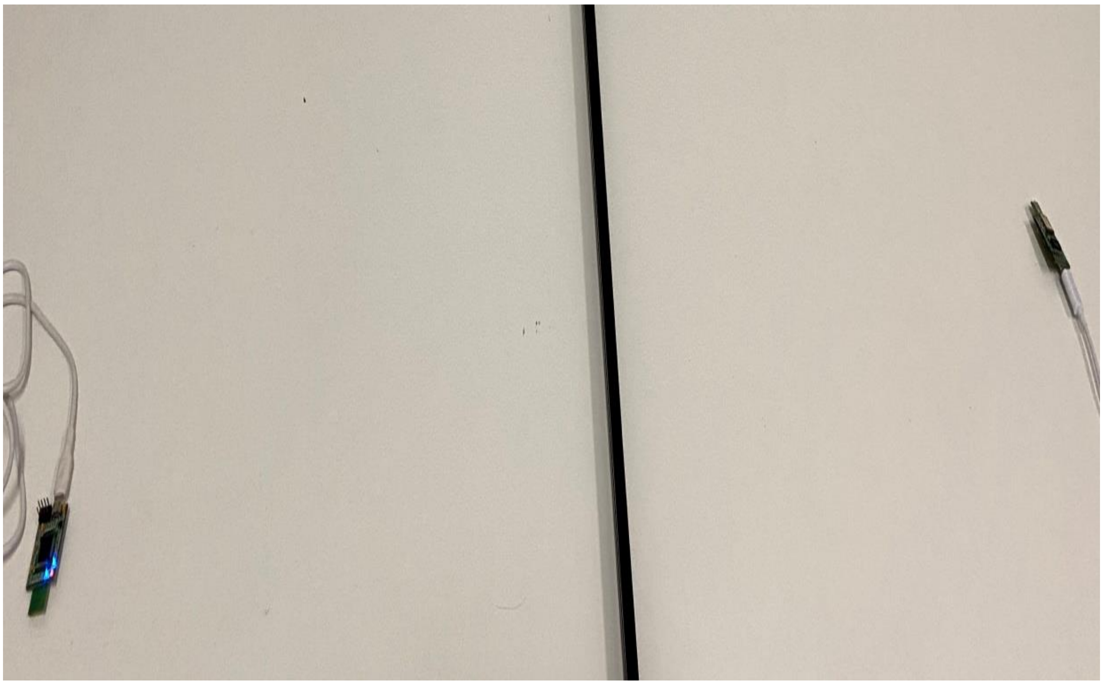


*Figure 33 – Place metal to measure the interference coefficient*

### 5.3. Method of evaluation

The experimental part was carried out in laboratory conditions. The running route of logistics packages were simulated based on driving route of a robot. Since the size of the laboratory space is not very large, a tape measure for distance measurement tools is sufficient. The aim of the mounted anchors is to capture the location of the tag. Thus it is better to put up three anchors in different corners of the considering room,  where keep a certain distance as much as possible and let tag move in the area covered of three anchors. In the laboratory experiment, the placement of obstacles is arranged in the way mentioned in the "theory hypothesis" chapter, the anchor_A was considered as the origin of coordinates and the east direction was considered as the positive direction of the x-axis and the north direction as the positive direction of the y-axis.

Considering the above principles， were set another two anchors in two different places and the horizontal and vertical distances between anchor_A, anchor_B and anchor_C were measured to get the coordinate points of anchor_B (-0.3,4), and anchor_C (4.5,3.5). Simulation experiment concept is shown in Figure 34.

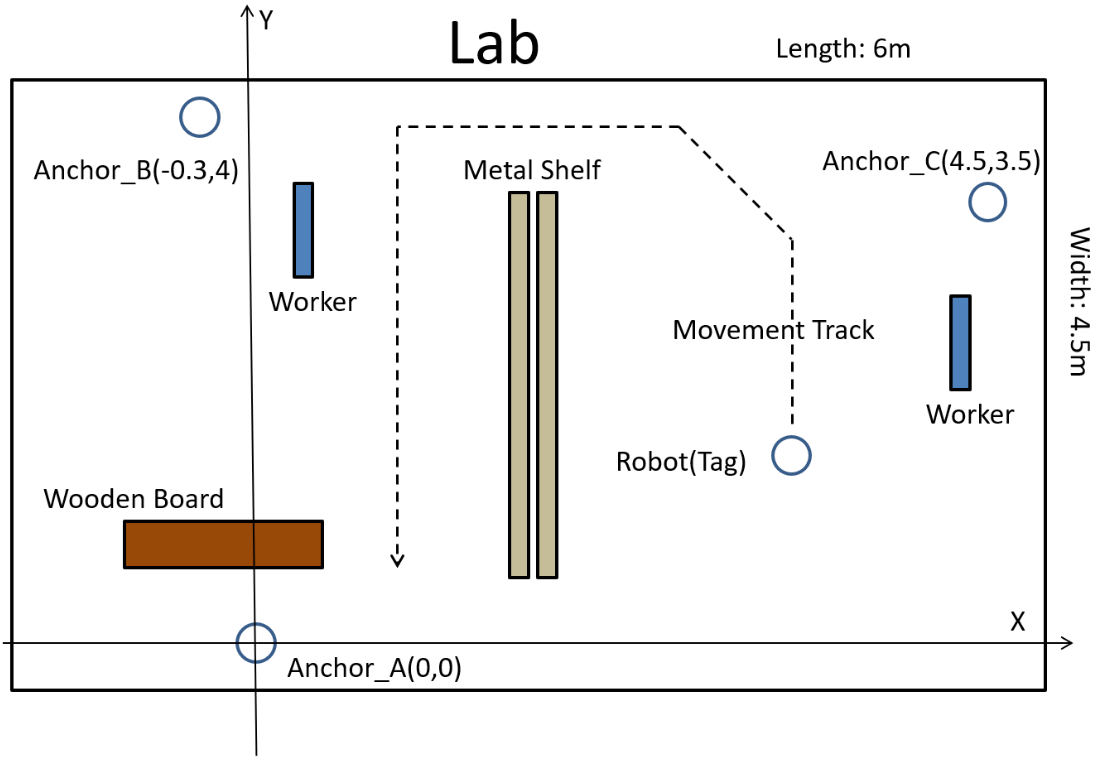


*Figure 34 – Equipment setup*

It is assumed that the movement track of a logistics package from entering the warehouse to reaching the shelf is fixed. Simulate robot in the laboratory, randomly arrange the positions of the metal, board and worker in the laboratory and fix them. And let the robot move a certain track within 40 seconds as well as directly observe whether there are obvious obstacles affecting line of sight between the tag mounted on robot and anchor during the movement of the robot, and what the obstacle is. Take the current state where the UWB system is not blocked by any obstacles as an example, every 2 seconds record status that what obstacle block signal transmission, record 20 times, observe and count the following results: In 20 records, There are 4 records that the line of sight between tag and anchor is blocked by metal, 2 records are due to wooden boards, 3 records are due to workers, and 11 records that the line of sight is not blocked by any obstacles.

So there are respectively 4/20 = 20%, 2/20 = 10%, 3/20 = 15% probability that the line of sight is blocked  by metal, board and worker; there is a 11/20 = 55% probability that it will not be affected by any medium, and UWB runs in a perfect environment. In a total of 9 cases of being affected, anchor_A, anchor_B, and anchor_C are each affected 3 times, so the probability of anchor_A, anchor_B, anchor_C being affected are respectively 1/3. The probabilities under other conditions can be measured through the same experiment. According to the theoretical definition of the probability transition matrix of Table 4, all probability data is made into a Markov probability matrix for use in the simulation program, as shown in the figure 34.

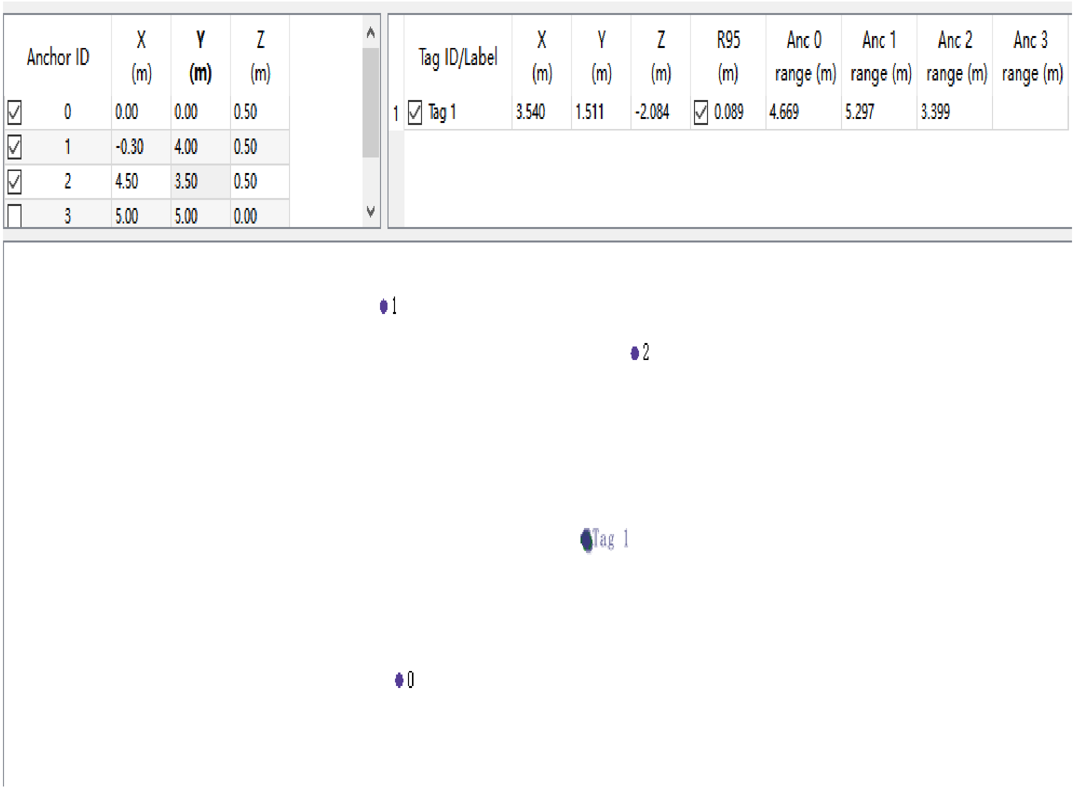
$$P = \begin{bmatrix} 0.55 & 0.20 & 0.10 & 0.15 \\ 0.20 & 0.50 & 0.15 & 0.15 \\ 0.15 & 0.20 & 0.45 & 0.20 \\ 0.20 & 0.15 & 0.15 & 0.50 \end{bmatrix}$$

*Figure 35 – Probability transition matrix of Markov*

# Chapter 6 – Simulation Results and Discussion

Using real-time positioning software decarangeRTLS to get distance value of anchor_A and tag, anchor_B and tag, anchor_C and tag. Storing data and running the Kalman filter algorithm and Markov algorithm to process them by Matlab.

## 6.1.Simulation results



| | Anchor ID | X (m) | Y (m) | Z (m) |
|---|---|---|---|---|
| ☑ | 0 | 0.00 | 0.00 | 0.50 |
| ☑ | 1 | -0.30 | 4.00 | 0.50 |
| ☑ | 2 | 4.50 | 3.50 | 0.50 |
| □ | 3 | 5.00 | 5.00 | 0.00 |

| | Tag ID/Label | X (m) | Y (m) | Z (m) | R95 (m) | Anc 0 range (m) | Anc 1 range (m) | Anc 2 range (m) | Anc 3 range (m) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | ☑ Tag 1 | 3.540 | 1.511 | -2.084 | ☑ 0.089 | 4.669 | 5.297 | 3.399 | |

*Figure 36 – Collect data by decarangeRTLS under TOA algorithm embedded in chip*

Under the original positioning algorithm—TOA algorithm, we collected distance values between anchor and tag on real time positioning software decarangeRTLS， as shown in Figure 36. Taking 120 for each of the three values which are the distance between anchor_A and tag, anchor_B and tag, anchor_C and tag as well as storing totally 360 sets of distance values in MATLAB. Due to the huge amount of data, the Table 5 only shows 120 distance values between anchor_A and tag , the unit is meters.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 4.650 | 5.029 | 4.445 | 5.035 | 4.824 | 4.423 | 4.559 | 4.760 | 5.068 | 5.074 | 4.468 | 5.078 |
| 4.732 | 4.713 | 4.352 | 4.666 | 4.682 | 4.967 | 4.534 | 4.561 | 4.877 | 4.760 | 4.399 | 4.982 |
| 4.468 | 4.682 | 4.629 | 4.388 | 4.526 | 4.635 | 4.783 | 4.735 | 4.996 | 5.027 | 4.902 | 4.410 |
| 4.555 | 5.034 | 4.882 | 4.596 | 4.410 | 4.003 | 5.086 | 4.399 | 4.375 | 4.879 | 4.358 | 4.648 |
| 4.717 | 4.920 | 5.030 | 5.019 | 4.786 | 4.656 | 4.489 | 4.901 | 5.003 | 4.950 | 4.477 | 4.419 |
| 4.826 | 4.632 | 4.378 | 4.520 | 4.884 | 4.909 | 4.628 | 4.687 | 4.962 | 5.017 | 4.800 | 4.562 |
| 4.617 | 4.773 | 4.951 | 5.083 | 4.685 | 5.091 | 4.485 | 4.593 | 4.842 | 4.615 | 4.892 | 5.039 |
| 4.752 | 4.951 | 4.421 | 4.644 | 4.621 | 4.985 | 4.395 | 4.475 | 4.685 | 4.960 | 5.069 | 4.684 |
| 4.965 | 4.632 | 4.776 | 4.920 | 4.841 | 4.358 | 4.821 | 4.665 | 4.388 | 4.698 | 4.541 | 5.021 |
| 5.007 | 4.396 | 4.521 | 4.741 | 4.665 | 4.958 | 4.548 | 4.886 | 5.036 | 4.551 | 4.858 | 4.627 |

*Table 5 – Record 120 distance values [m] between anchor_A and tag by decarangeRTLS and Store in MATLAB*

Due to the influence of NLOS, the values present fluctuation in a larger range. Taking the distance value of anchor_A and tag as an example, using Kalman filter algorithm to process data, we can get the optimized distance value of anchor_A and tag as shown in Figure 37. The red line represents the actual distance value between anchor_A and tag measured manually, observations represent the distance value between anchor_A and tag which stored in Matlab through manual record that displayed in software decarangeRTLS. The Kalman filter algorithm filters the noise based on the 120 observations stored in Matlab and performs optimization processing to obtain the Kalman filter value, which is the green line. The sampling time can be understood as a total of 120 sets of sample data, which have been sampled 120 times.

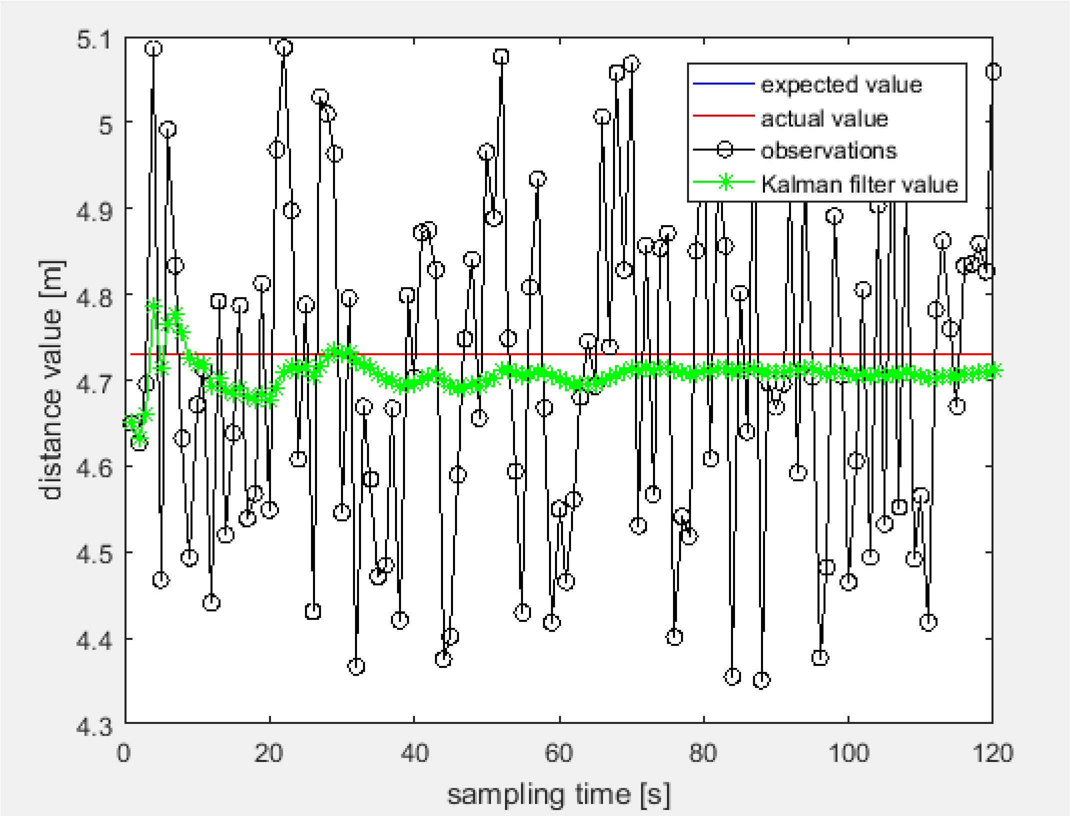


*Figure 37* – Optimized value of anchor_A and tag

In order to express the effect of Kalman filtering more intuitively, we define measurement error = |observations – actual value|, Kalman filter error = | Kalman filter value – actual value|, as shown in Figure 38. It can be seen that the Kalman filter error is significantly lower than the measurement error.

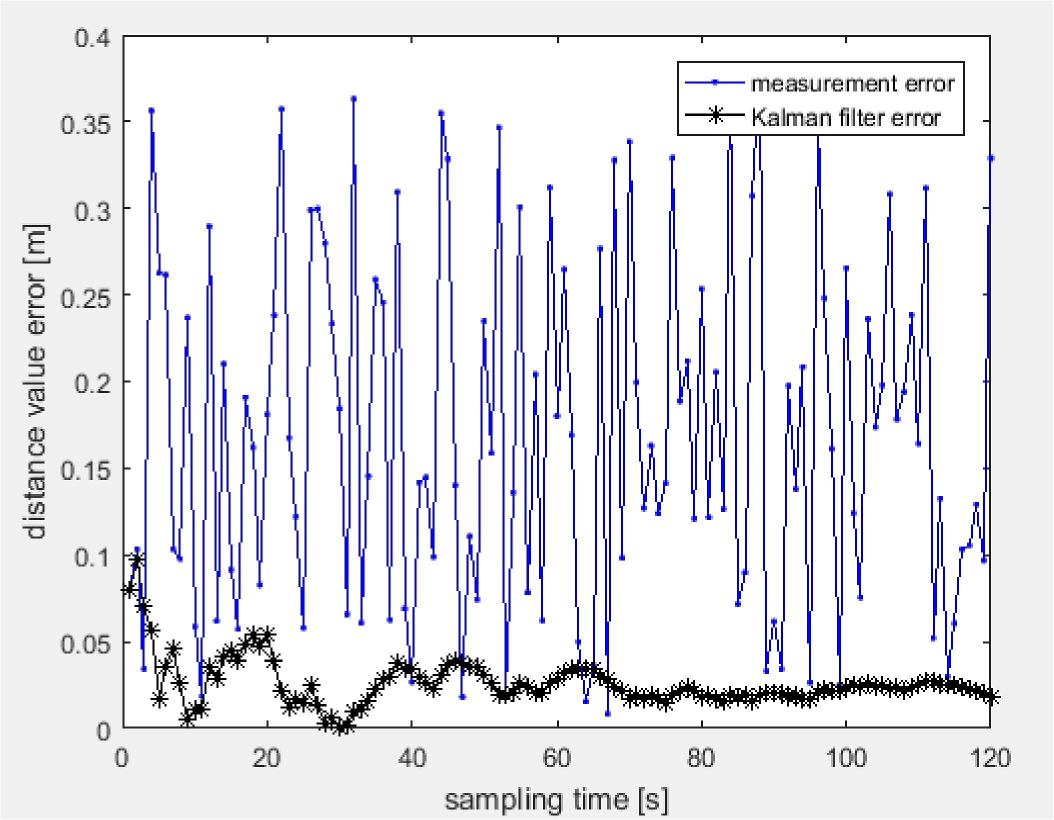


*Figure 38 – Measurement error and Kalman filter error of the distance between anchor_A and tag*

The distance value of anchor_B and tag (Figure 39 and Figure 40) , anchor_C and tag (Figure 41 and Figure 42) are shown the similar results in MATLAB by the same method. In almost every simulation, the Kalman filter value obtained is very close to the actual value, and the Kalman filter error is much smaller than the measurement error. And even if adjust the location of robot, Kalman filter algorithm processing results are also very good by MATLAB.

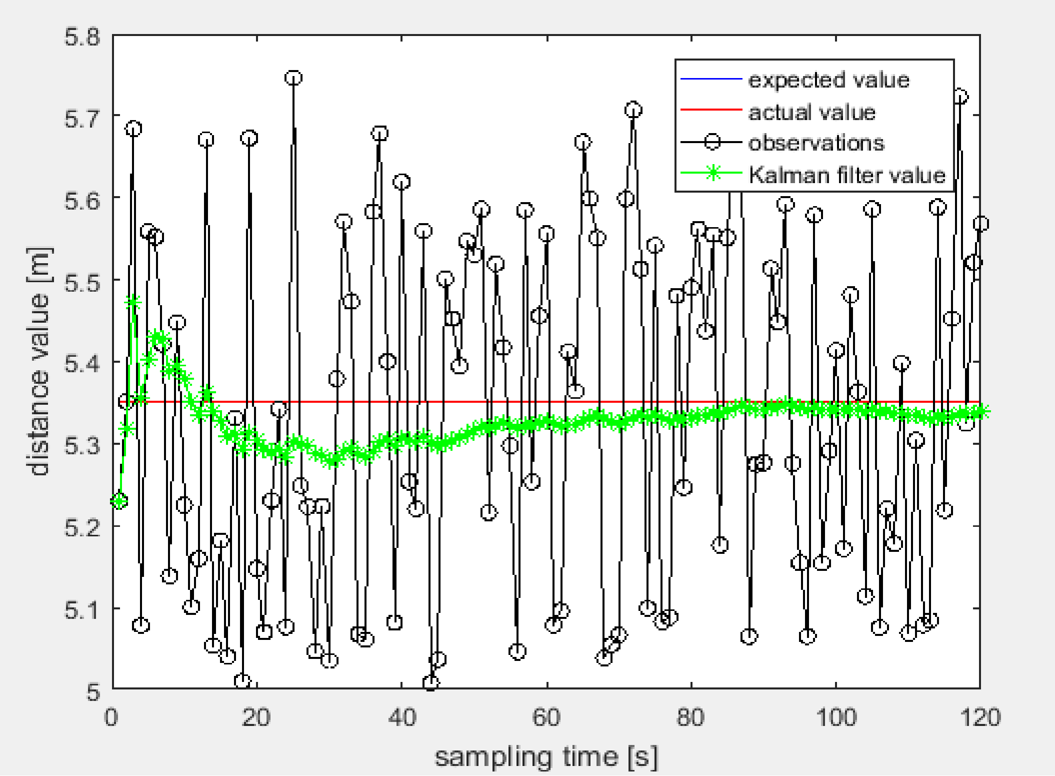


*Figure 39* – Optimized value of anchor_B and tag

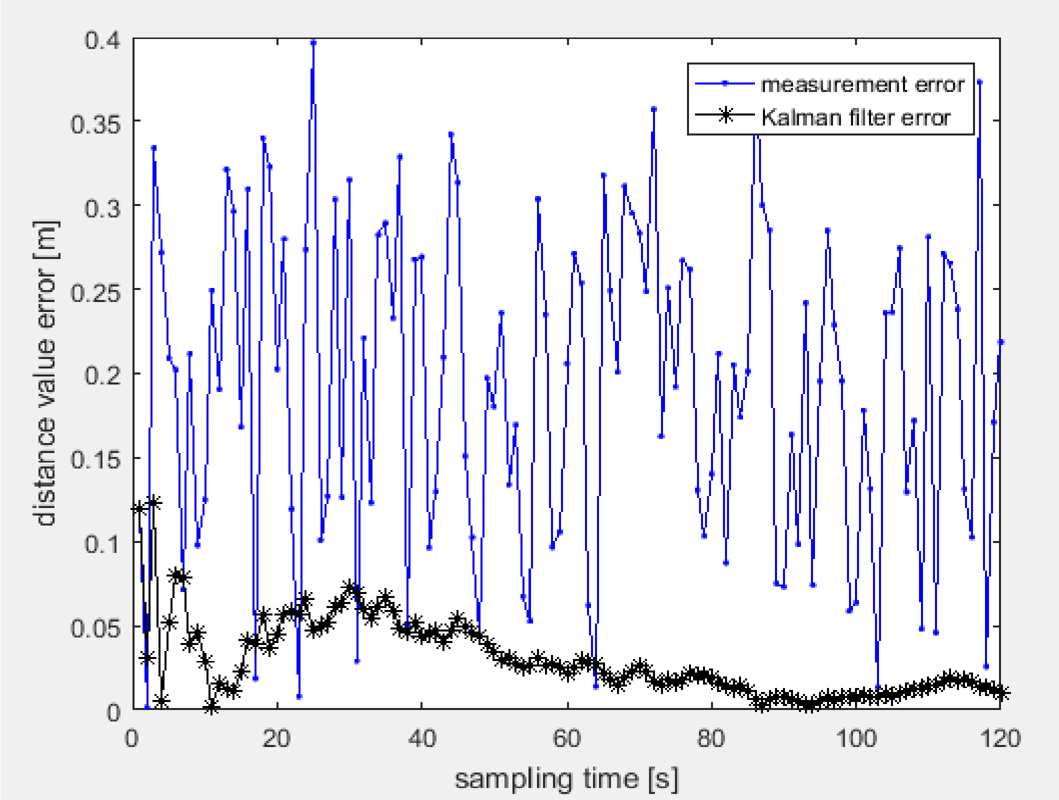


*Figure 40 – Measurement error and Kalman filter error of the distance between anchor_B and tag*

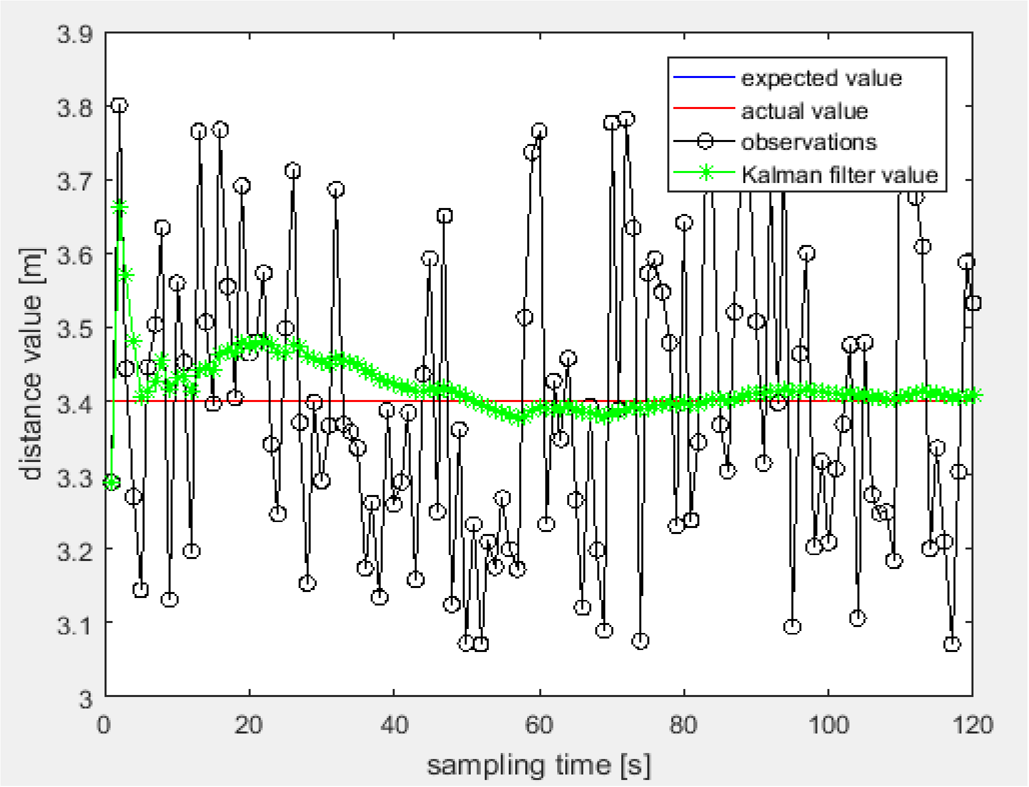


*Figure 41 – Optimized value of anchor_C and tag*

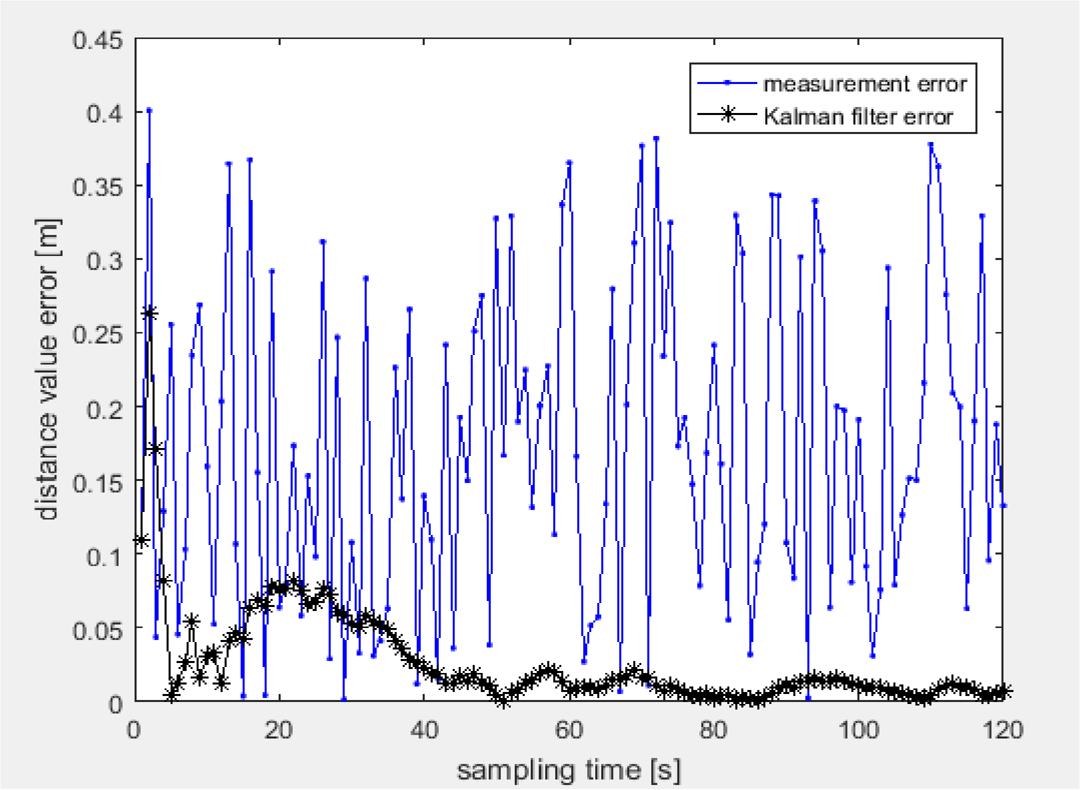


*Figure 42 – Measurement error and Kalman filter error of  the distance between anchor_C and tag*

Table 6 and Table 7 present the positioning error in comparative way. Taking into account the distance of Anchor_A and tag as example, we collect 120 measured value which fluctuate from 4.352 to 5.091 (Table 5) from small to large when we use original algorithm—TOA algorithm; But using Kalman filter algorithm, the fluctuation range of measured distance value of anchor_A and tag is from 4.660-4.802 (Figure 37). Compared with real value, the Kalman filter algorithm is easier to narrow the range of data fluctuations and make the measurement error smaller.

| | Real Value | Measured Value |
|---|---|---|
| Anchor_A-tag | 4.73m | 4.352m~5.091m |
| Anchor_B-tag | 5.35m | 5.034m~5.758m |
| Anchor_C-tag | 3.40m | 3.079m~3.806m |

*Table 6 – The original algorithm (meter)*

| | Real Value | Measured Value |
|---|---|---|
| Anchor_A-tag | 4.73m | 4.660m~4.802m |
| Anchor_B-tag | 5.35m | 5.281m~5.433m |
| Anchor_C-tag | 3.40m | 3.371m~3.428m |

*Table 7 – After Kalman filter algorithm (meter)*

### 6.2. Experimental results

The protocol involve the randomly selection of static coordinate point during the movement of robot, and draw the intersection point of the three circles according to the three distance values of anchor_A and tag, anchor_B and tag, and anchor_C and tag which are optimized by Kalman filter algorithm.

As shown, the red circle, green circle and blue circle respectively indicate that the circle is drawn with the distance values of anchor_A and tag, anchor_B and tag, anchor_C and tag (Figure 43). At this time, the tag tested coordinate (3.5,1.5) obtained by the real-time positioning software decarangeRTLS are in the intersection area of the red circle, the green circle and the blue circle, the range is large, and the position is not accurate enough.

According to the probability transition matrix of Markov (Figure 35) and interference coefficient, The program of Markov algorithm implemented in MATLB calculates the proportion of probability (Figure 24) and judges (Figure 25) that the distance value of anchor_B and tag may be affected by obstacle-metal under the current position of the robot, and recalculates a new distance value of anchor_B and tag based on interference coefficient as well as draws a new circle based on the new distance value of anchor_B and tag to obtain a black circle (Figure 43).

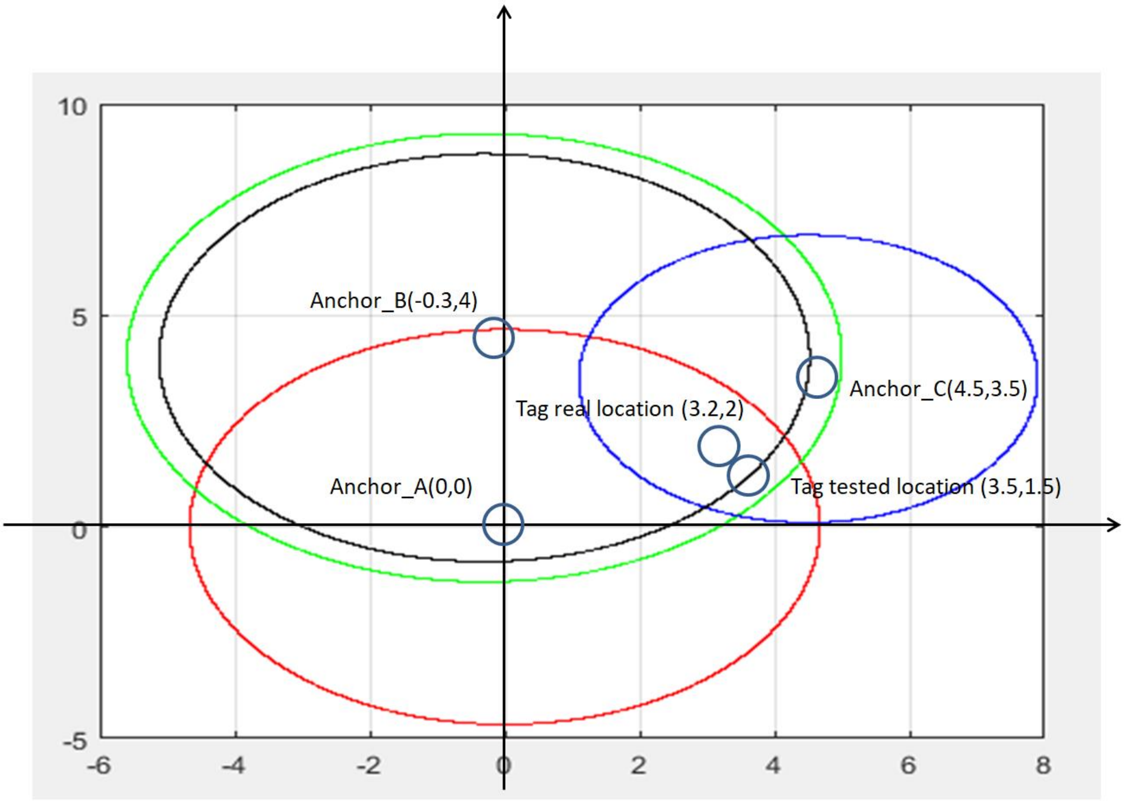


*Figure 43 – Markov algorithm positioning result [m]*

According to the experimental scene in Figure 34 , the obtained coordinates of anchor_A (0,0) , anchor_B (-0.3,4) , anchor_C (4.5,3.5) are presented in the simulation in Figure 43. The tag tested coordinate (3.5,1.5) obtained by the real-time positioning software decarangeRTLS. According to the simulation diagram, it can be clearly seen that the intersecting area of the black circle, red circle and blue circle is smaller than the former area which determined by red circle, the green circle and the blue circle. The real coordinate of the tag measured by a tape measure is (3.2,2) in practice, it is found that the tag is located at the intersection of the red circle, blue circle and black circle, and it locates the real tag coordinates more accurately than the original algorithm of the area obtained by the intersection of the green circle, the red circle and the blue circle. the conclusion of the simulation is confirmed and the validity of the Markov algorithm is also proved.

# Chapter 7 – Conclusions and Future Work

## 7.1. Conclusions

In the present work, by simulating a warehouse environment with three interference media in the laboratory, the UWB system is used to locate the robot mounted the tag. The Kalman filter and Markov algorithms were successfully applied to improve the tag positioning by comparison with the original UWB classic positioning algorithm. Based on the obtained results the proposed solution can be considered for application in logistics warehouses. The performed simulations show that Kalman filter algorithm and Markov algorithm can effectively eliminate NLOS errors and improve the positioning accuracy.

However, the Markov algorithm relies on random probability, which requires a large amount of data associated with the transportation trajectory of the real logistics warehouse scenario. Thus, in the simulation part，360 sets of distance values, 16 sets of probability values and 4 sets interference coefficients were tested. If the experiment is placed in a real logistics warehouse, the amount of data required will be greatly increased, the types of obstacles will be more complicated. But the research process and method of this simulation experiment can be greatly extended to the experiment of logistics warehouse, providing reference for future research.

## 7.2. Future Work

As mentioned in the conclusion section, this project still has many shortcomings and needs to be improved. Such as the following:

①：This project is to use real-time software to obtain data and save it offline in Matlab. Try to use software such as Labview to save data online and complete dynamic positioning simulation?

②：What some algorithm else be used to process the distance value of the anchor and the tag better than the Kalman filter?

③：If BP neural network and Kalman filter are used to jointly process the obtained distance value, will better optimization results be obtained? In the future, I can study some artificial neural network processing data sets.

④：In this project, by default, each interference can only be caused by one interference medium. If the interference factors are more complicated in a real logistics warehouse, it may be affected by two or more interference media at the same time. Therefore, in future work, we need to try to modify the algorithm that matches the simultaneous influence of multiple interferences.

# Bibliografia

[1] Xin Li , Fucheng Cao, Location Based TOA Algorithm for UWB Wireless Body Area Networks, 2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing.

[2] Maria-Cabriella Di benedetto, Thomas Kaiser, Andreas F.Molisch, UWB Communication Systems: A Comprehensive Overview, Hindawi Publishing Corporation.

[3] Jelena Kaitovic , Miljko Erić, TDOA localization in ir UWB systems, 2012 19th International Conference on Systems, Signals and Image Processing (IWSSIP).

[4] Yongping He, Ran Liu, Wenpeng Fu, UWB-based indoor dynamic target positioning in non-line-of-sight environments, Transducer and Microsystem Technologies, 10.13873 /J.1000—9787( 2020) 08—0046—04.

[5] Lei Ding, Xinhua Zhu , Tong Zhou ,Research on UWB-Based Indoor Ranging Positioning Technolog and a Method to Improve Accuracy, 2018 IEEE Region Ten Symposium (Tensymp).

[6] Harris Perakis, Vassilis Gikas , Evaluation of range error calibration models for indoor UWB positioning applications, 2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 24-27 September 2018, Nantes, France.

[7] Hao Chen , Jie He, Kai Ma, UWB precision positioning algorithm based on BP neural network compensation Kalman filter, Electronic Design Engineering, 1674-6236（2019）24-0103-05.

[8] Jun Fu, Da Xu, Yang Fu, Application of an adaptive UKF in UWB indoor positioning, Surveying and Mapping, 2019 (S1): 12-17. DOI: 10.13474 /j. 11-2246. 2019.0503.

[9] Longliang Chen, En Yuan, Combined indoor positioning algorithm based on UWB/PDR, Information Technology and Network Security, 2019, 38 (5): 53-57.

[10] Oh-Soon Shin, S.S. Ghazzemzadeh, L.J. Greenstein, V. Tarokh, Performance evaluation of MB-OFDM and DS-UWB systems for wireless personal area networks, 2005 IEEE International Conference on Ultra-Wideband

[11] Abolfazl Mehbodniya, Sonia Aissa, Effects of MB-OFDM System Interference on the Performance of DS-UWB, IEEE Transactions on Vehicular Technology ( Volume: 58, Issue: 8, Oct. 2009).

[12] "What are the applications based on UWB? " [Online]. Available: https://jingyan.baidu.com/article/455a9950d9f213e167277833.html. [Accessed: 03-Jul- 2020].,

[13] "Mainstream Connectivity line, Arm Cortex-M3 MCU with 256 Kbytes of Flash memory, 72 MHz CPU, CAN, USB 2.0 OTG" [Online]. Available: https://www.st.com/zh/microcontrollers-microprocessors/stm32f105vc.html. [Accessed: 17- Jul- 2020].

[14] "DWM1000 Module" [Online]. Available: https://www.decawave.com/product/dwm1000-module/. [Accessed: 19- Jul- 2020].

[15] "ST-LINK/V2 in-circuit debugger/programmer for STM8 and STM32" [Online]. Available: https://www.st.com/zh/development-tools/st-link-v2.html. [Accessed: 23-Jul- 2020].

[16] Luiz Henrique Nunes, Luis Hideo Vasconcelos Nakamura, Heitor De Freitas Vieira, A Study Case of Restful Frameworks in Raspberry Pi: A Performance and Energy Overview, 2014 IEEE International Conference on Web Services.

[17] Xin Li , Fucheng Cao, Location Based TOA Algorithm for UWB Wireless Body Area Networks, 2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing.

[18] Do-Jin An , Joon-Ho Lee, Derivation of an Approximate Location Estimate in Angle-of-Arrival Based Localization in the Presence of Angle-of-Arrival Estimate Error and Sensor Location Error, 2018 IEEE World Symposium on Communication Engineering (WSCE).

[19] Jelena Kaitovic , Miljko Erić, TDOA localization in ir UWB systems, 2012 19th International Conference on Systems, Signals and Image Processing (IWSSIP).

[20] Fang Dong, Chong Shen, Jie Zhang, A TOF and Kalman filtering joint algorithm for IEEE802.15.4a UWB Locating, 2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference.

[21] Örner Çetin, Hakki Nazh, Ridvan Gürcan, An experimental study of high precision TOA based UWB positioning systems, 2012 IEEE International Conference on Ultra-Wideband.

[22] Lorenz Schmid, David Salido-Monzú, Andreas Wieser, Accuracy Assessment and Learned Error Mitigation of UWB ToF Ranging, 2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN).

[23] "Formula Derivation of TOF Ranging Method in UWB" [Online]. Available: https://blog.csdn.net/tyst08/article/details/106315892/. [Accessed: 5- Aug- 2020].

[24] Waldemar Gerok, Juergen Peissig, Thomas Kaiser, Study of the TDOA Assisted RSSD Localization with UWB in Real Indoor Environment, European Wireless 2013; 19th European Wireless Conference.

[25] Stefan Galler, Waldemar Gerok, Combined AOA/TOA UWB localization, 2007 International Symposium on Communications and Information Technologies.

[26] Simon Haykin, Kalman Filtering and Neural Networks, A Wiley-Interscience Publication, 2001.

[27] "Detailed derivation of Kalman filter algorithm" [Online]. Available: https://blog.csdn.net/victor_zy/article/details/82862904. [Accessed: 12-OCT- 2020].

[28] Zirui Wang, Shaoxian Li, Zhengyuan Zhang, Fan Lv, Research on UWB Positioning Accuracy in Warehouse Environment, Procedia Computer Science 131 (2018) 946–951.

[29] Christophe Andrieu, Arnaud Doucet, Roman Holensteim, Particle markov chain monte carlo methods, Journal of the Royal Statistical Society: Series B(Statistical Methodology) 72(3), 269-342,2010.

[30] N. Leena, B. Durai Raj, R. Gunabalan, Computer-based laboratory teaching tools: An overview of LabVIEW and MATLAB, 2012 IEEE International Conference on Engineering Education: Innovative Practices and Future Trends (AICERA).

[31] Kyon-Mo Yang, Jong-Boo Han, Kap-Ho Seo, A Multi-robot Control System based on ROS for Exploring Disaster Environment, 2019 7th International Conference on Control, Mechatronics and Automation (ICCMA).

[32] "Xshell 6" [Online]. Available: https://www.netsarang.com/en/xshell/. [Accessed: 1-Aug- 2020].

[33] "CoIDE"[Online].Available:https://baike.baidu.com/item/CoIDE/1477708?fr=aladdin. [Accessed: 8-Feb- 2020].

# Annex

# a. UWB System and Algorithms for Indoor Positioning

Yuankang Gao
*ISCTE-Instituto Universitário de Lisboa*
*IT – Instituto de Telecomunicações*
SMU, Shanghai, China
Lisbon, Portugal
ygogu@iscte-iul.pt

Octavian Postolache
*ISCTE-Instituto Universitário de Lisboa*
*IT – Instituto de Telecomunicações*
Lisbon, Portugal
opostolache@lx.it.pt

Yongsheng Yang
*Shanghai Maritime University*
Shanghai, China
ftranda@yahoo.com

Bin Yang
*Shanghai Maritime University*
Shanghai, China
binyang@shmtu.edu.cn

***Abstract*—This paper presents of study of ultra-wide band (UWB) indoor positioning considering different type of obstacles that can affect the localization accuracy. In the actual warehouse, a variety of obstacles including metal, board, worker and other obstacles will have NLOS (non-line-of-sight) impact on the positioning of the logistics package, which influence the measurement of the distance between the logistics package and the anchor , thereby affecting positioning accuracy. A new developed method attempts to improve the accuracy of UWB indoor positioning, through and improved positioning algorithm and filtering algorithm. In this project, simulate the warehouse environment in the laboratory, several simulation proves that the used Kalman filter algorithm and Markov algorithm can effectively reduce the error of NLOS. Experimental validation is carried out considering a mobile tag mounted on a robot platform.**

***Keywords—UWB, indoor positioning, Kalman filter algorithm, Markov algorithm.***

## • Introduction

Ultra Wide Band (UWB ） is a wireless carrier communication technology. Its main features are high transmission rate, large range capacity and low power consumption. UWB positioning technology plays an important role in indoor/outdoor positioning.

In practical applications such as warehouse, the problem of NLOS (non-line-of-sight) will affect the accuracy of UWB positioning. Heavy metallic machinery often blocks the line of sight path, metal walls and metal surfaces increase reflections and multi-path effects. In this context, the research regarding NLOS problem has become the focus of improving positioning accuracy.

In the past, the accuracy of UWB positioning research mainly focused on the improvement of hardware and UWB localization algorithms. Many classic algorithms or new algorithms were used in UWB indoor localization field, which were carried out slightly to improve positioning accuracy. There are three classic positioning methods which include: Time of Arrival (TOA) [1], Angle of Arrival (AOA) [2], Time Difference of Arrival (TDOA) [3]. And some new positioning algorithms were proposed recently，thus in [4] proposes a line to approximate hyperbola positioning algorithm based on UWB aiming at the problems of low-accuracy positioning, non-line-of-sight phenomenon and multipath propagation in the precise positioning of mobile tags, using some lines to approximate hyperbola. It can convert quadratic hyperbolic equation into linear equation. Then it can achieve the high-accuracy position combining the time difference of arrival and triangle centroid position algorithm; [5] have been reported the usage of TOF principle combined with the spatial 4-point positioning method to determine the position of the target point and adopted STM32 to control the DW1000 to realize the UWB signal transceiving on the hardware; In study [6], a general methodology for performing calibration for each indoor area was proposed. Because UWB ranges suffer systematic errors in the form of bias for each ranging conversation. The elimination of this bias is attempted through collecting UWB ranges at known intermodal distances between surveyed calibration points. UWB ranging paired with the appropriate calibration methods is able to provide indoor positioning of high accuracy in the centimeter level.

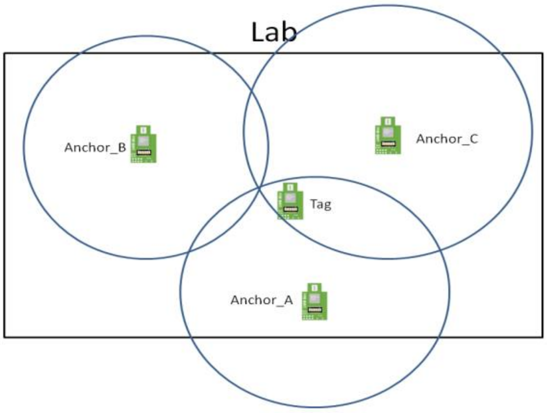
In this work, the experimental part was carried out in laboratory conditions. The running route of logistics packages were simulated based on driving route of a robot. The laboratory equipment can simulate the NLOS of warehouse as well. We use three anchors to locate a mobile tag mounted on a robot and a host computer software for data acquisition and processing. Several algorithms such Kalman filter algorithms, Markov algorithm were implemented and tested for accurate indoor localization.

## • System Model and Algorithm Principle

A complete indoor positioning study is supported by hardware equipment and algorithms implemented in PC computation platform. The simulated experiments' equipment consists of 4 UWB mini3 and a robot. Algorithm is divided into two parts： Kalman filter algorithm and Markov algorithm.

*Hardware Description*

The indoor positioning is carried out with a UWB indoor localization kit that consists of 4 UWB Mini3 which integrates UWB positioning system (Fig.1). The UWB Mini 3 modules are based on STM32F105 microcontroller. The STM32F105/107 devices use the Cortex-M3 core, with a maximum CPU speed of 72 MHz They are intended for applications where connectivity and real-time performances are required such the indoor localization application. Peripheral circuits are expressed by DWM1000 module, power module, LED indicator module, DIP switch and reset circuit. This module can be used as an anchor or a tag associated with the indoor localization system.



Anchors and tag setup.

In the present work three anchors which were mounted on tripod (Fig. 2.a) were considered. The tag was mounted on the level of a robot to assure the static and dynamic experiments (Fig.2.b). The specifications regarding efficient range of indoor positioning is 30 meters and the efficient range of outdoor positioning is 50 meters underline that our experience in laboratory scenario can be considered appropriate.

a)

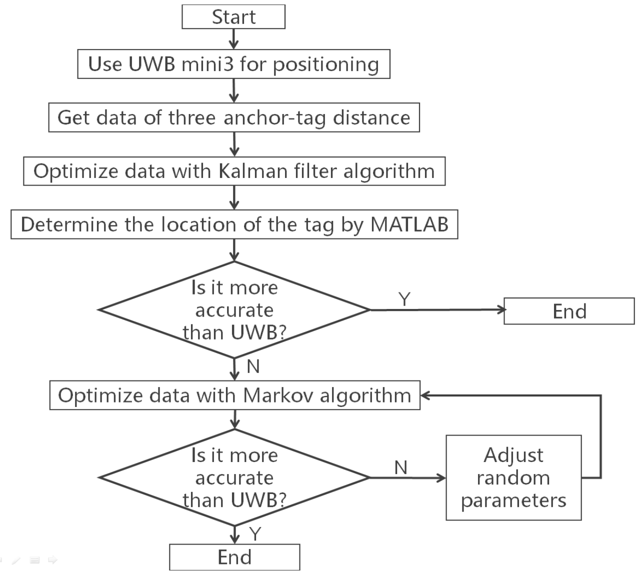
b)

Experimental setup of indoor UWB positioning system: a) UWB mini3 anchor supported by tripod; b) UWBmini3 suported by robot.

The robot is controlled by a PC through Wi-Fi PC, thus the Wi.Fi network includes the PC and Raspberry Pi computation platform mounted on the mechanical part of the robot. The implemented setup can be used to determine the trajectory of the moving tag and study the positioning error of UWB. The work condition of UWB system is shown in chapter III.

*Algorithm Description*

The software components for indoor localization system include the firmware deployed on the level of UWB mini3, the robot control software module and data analysis software module. The interaction between different software modules is presented in the Software flowchart (Fig.3). Thus, the real-time positioning software-decarangeRTLS under TOA algorithm is responsible for providing the distance value between each anchor and tag and the coordinates of the tag under the TOA algorithm. The MATLAB is used to store the data and to implement the Kalman filter algorithm and Markov algorithm for data processing.


Start
Use UWB mini3 for positioning
Get data of three anchor-tag distance
Optimize data with Kalman filter algorithm
Determine the location of the tag by MATLAB
Is it more accurate than UWB?
Y
End
N
Optimize data with Markov algorithm
Is it more accurate than UWB?
N
Adjust random parameters
Y
End

Software modules interaction diagram.

Among classical algorithms, TOA algorithm is the most classic and the most widely used algorithm. In personal experiments, TOA algorithm is used to calculate the distance between the tag and the anchor, so that the position coordinates of the tag can be obtained through the three-sided distance.

It is assumed that under perfect conditions, tag and anchor have synchronized time standards. At a certain moment, anchor_C sends a signal to tag, records the time when tag receives the signal, and calculates the time difference based on this. Multiply the time difference by the speed of light to get the distance from anchor_C to tag, and use the same method to get the distances from three anchors to tag respectively. Use the three distance values as the radius to draw a circle, and you can get the intersection of the three circles, that is the coordinate position of the tag.

①：distance=speed of light ×time/ 2

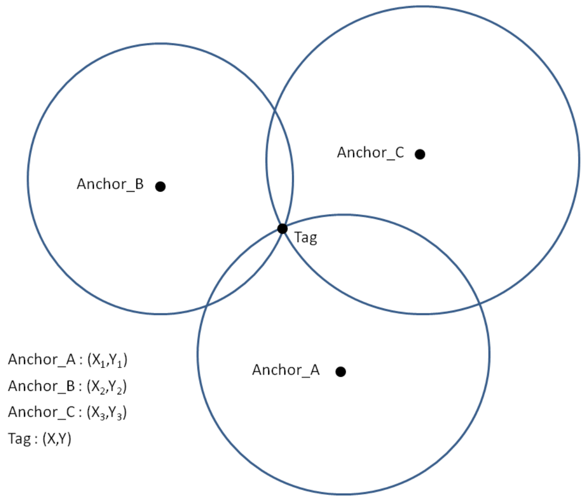
②：On the XY level, 3 circles can determine a point

③：In XYZ space, 4 circles can determine a space point

$$\mathrm{r_A} = \sqrt{(\mathrm{x}-\mathrm{x_1})^2+(\mathrm{y}-\mathrm{y_1})^2} \qquad (1)$$

$$\mathrm{r_B} = \sqrt{(\mathrm{x}-\mathrm{x_2})^2+(\mathrm{y}-\mathrm{y_2})^2} \qquad (2)$$

$$r_C = \sqrt{(x-x_3)^2+(y-y_3)^2} \qquad (3)$$



TOA algorithm theory.

In positioning and ranging, UWB can achieve decimeter-level positioning accuracy and strong pulse signal penetration, Suitable for indoor positioning, UWB is currently the most accurate positioning technology in the indoor positioning. However， the problems of NLOS existing in warehouse environment will greatly affect UWB indoor positioning accuracy. The main factors that affect the positioning accuracy are the logistics machines, cement walls, metal shelves, or workers. These obstacles may block the transmission of pulse signals and interfere with positioning accuracy. To increase the accuracy different algorithms were considered and following discussed.

*Kalman  Filter Algorithm*

Affected by NLOS, the signal transmission between the single-sided tag and the anchor is interfered, and the obtained distance value has an error. The original WUB system will get fluctuating and unstable distance values. If the existing TOA algorithm is used to draw a circle, it will make the three circles intersect into an area instead of a point.

Kalman filtering[7] is suitable for estimating the optimal state of a dynamic system. Even if the observed system state parameters contain noise and the observed values are not accurate, Kalman filtering can complete the optimal estimation of the true state values.

Observation value is Z, and the estimated value is X, then the new estimated value should be

$$Xnew = X + K\,(Z\text{-}X) \qquad (4)$$

This will ensure that the new estimate must be more accurate than the current one. Once it calculates again, it will become more and more accurate. The difficulty is how to determine the value of Kalman gain K, which needs to be continuously tested.

*Markov Algorithm*

Markov chain is a random process in a state space. This process requires a “no memory” property: the probability distribution of the next state can only be determined by the current state. There have been many studies based on Markov chains in recent years [8].

It is used to indicate the random probability of ranging from state i to state j after N times, and is used to indicate the probability that the ranging situation of the Nth cycle is exactly at state i.

$$\pi^{N} = \pi^{N\text{-}1} \times P = \pi^{0} \times P^{N} \qquad (5)$$

If we know the initial state vector and probability matrix of the predicted state, we can get the vector of the next state. It is assumed that a Markovian chain as Fig.5.

$$\mathrm{P} = \begin{bmatrix} \mathrm{P}_{11} & \mathrm{P}_{12} & \mathrm{P}_{13} & \mathrm{P}_{14} \\ \mathrm{P}_{21} & \mathrm{P}_{22} & \mathrm{P}_{23} & \mathrm{P}_{24} \\ \mathrm{P}_{31} & \mathrm{P}_{32} & \mathrm{P}_{33} & \mathrm{P}_{34} \\ \mathrm{P}_{41} & \mathrm{P}_{42} & \mathrm{P}_{43} & \mathrm{P}_{44} \end{bmatrix}$$

Random probability.

According to the nature of the Markov chain state transition matrix, $\mathrm{P_{ij}}$(i,j=1,2,3,4) meaning can be explained intuitively. And there are known proofs to get the following conclusions:

$$Pij > 0, \sum_{j=1}^{4} Pij = 1 \qquad (6)$$

Taking into account the common interferences in the actual warehouse (e.g. metal obstacles) were considered the above obstacles to emulate the warehouse environment in the laboratory.

In order to effectively eliminate NLOS interference, use the reliable blocking probability to "predict" possible interference methods in advance and perform data compensation. This kind of fuzzy control can be applied to experiments.

As shown Fig.5 , i represents the current state of robot, j represents the new state of robot, when i=1,2,3,4, the current state respectively indicate that the robot affected by nothing, metal, board and worker, when j=1,2,3,4, the new state respectively indicate that the robot affected by nothing, metal, board and worker (Table 1). so for instance: P11 represents the possibility that the robot is not affected by any medium from current state to new state; P12 represents the possibility that the robot is affected by metal from current state to new state; P23 represents the probability that the robot will change from the current state affected by metal to the new state affected by board.

STATE TRANSITION

| new<br>current | Blocked by nothing | Blocked by metal | Blocked by board | Blocked by worker |
|---|---|---|---|---|
| Blocked by nothing | $P_{11}$ | $P_{12}$ | $P_{13}$ | $P_{14}$ |
| Blocked by metal | $P_{21}$ | $P_{22}$ | $P_{23}$ | $P_{24}$ |
| Blocked by board | $P_{31}$ | $P_{32}$ | $P_{33}$ | $P_{34}$ |
| Blocked by worker | $P_{41}$ | $P_{42}$ | $P_{43}$ | $P_{44}$ |

- SIMULATION

*Theory Hypothesis*

Our goal is to eliminate the interference of NLOS on UWB ranging, and we must first measure the error percentage of the impact of various interference media on ranging. Several simulated scenarios were considered in the laboratory when metal, board, and workers are blocking the signal transmission between the tag and the anchors. The distance values provided by UWB software decarangeRTLS in time are recorded. In lab experiment, no matter how far anchor A and tag are, compared with the actual distance value, the distance value of the anchor_A between tag shown in decarangeRTLS is almost accurate; When deliberately placed a metal object block the signal transmission line of anchor_A and tag as well as adjusted the distance of anchor_A and tag (Fig.6), the distance value displayed in the decarangeRTLS is quite different from the actual distance value. Recorded distance value and the actual distance value are used as part of linear function that is used to get the interference coefficient. Through linear processing of the actual distance value and the recorded data value, we can get the interference coefficient of metal is 9%. Through the same experiment, we can respectively get the interference coefficient of board and worker is 5% and 14%. These interference coefficients can be used as compensation basis for correcting errors.

Place metal to measure the interference coefficient.

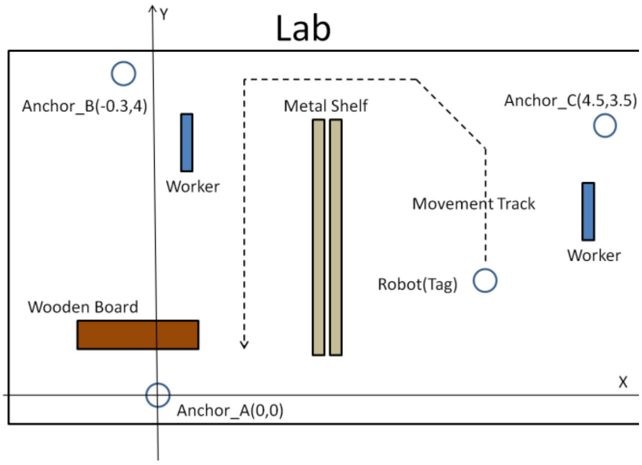
It is assumed that the movement track of a logistics package from entering the warehouse to reaching the shelf is fixed. Simulate robot in the laboratory, randomly arrange the positions of the metal, board and worker in the laboratory and fix them. And let the robot move a certain track within 40 seconds as well as observe whether there are obvious obstacles affecting line of sight between the tag mounted on robot and anchor during the movement of the robot, and what the obstacle is. Record status every 2 seconds, record 20 times, take the current state where the UWB system is not blocked by any obstacles as an example, observe and count the following results: In 20 records, there are 4 records that the line of sight between tag and anchor is blocked by metal, 2 records are due to wooden boards, 3 records are due to workers, and 11 records that the line of sight is not blocked by any obstacles. So there are respectively 20%, 10%, 15% probability that the line of sight is blocked by metal, board and worker; there is a 55% probability that it will not be affected by any medium, and UWB runs in a perfect environment. The probabilities under other conditions can be measured through the same experiment. All probability data is made into a Markov probability matrix for use in the simulation program, as shown in the Fig.7.

$$P = \begin{bmatrix} 0.55 & 0.20 & 0.10 & 0.15 \\ 0.20 & 0.50 & 0.15 & 0.15 \\ 0.15 & 0.20 & 0.45 & 0.20 \\ 0.20 & 0.15 & 0.15 & 0.50 \end{bmatrix}$$

Blocking probability matrix of Markov.

*Method Evaluation*

The aim of the mounted anchors is to capture the location of tag. Thus it is better to put up three anchors in different corners of the considering room, where keep a certain distance as much as possible and let tag move in the area covered of three anchors. In the laboratory experiment, the placement of obstacles is arranged in the way mentioned in the "theory hypothesis" chapter, the anchor_A was considered as the origin of coordinates and the east direction was considered as the positive direction of the x-axis and the north direction as the positive direction of the y-axis. Considering the above principles， were set another two anchors in two different places and the horizontal and vertical distances between anchor_A, anchor_B and anchor_C were measured to get the coordinate points of anchor_B(-0.3,4), and anchor_C(4.5,3.5). Simulation experiment concept is shown in Fig.8.

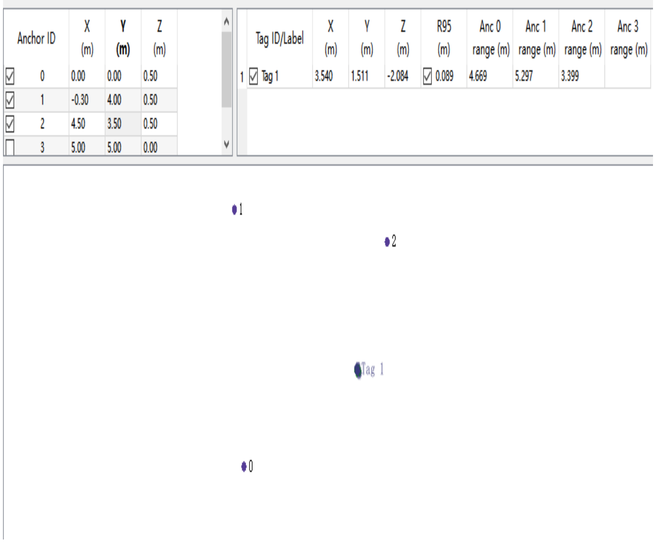


Equipment distribution.

- Results and Discussion

Using real-time positioning software decarangeRTLS to get distance value of anchor_A and tag, anchor_B and tag, anchor_C and tag. Storing data and running the Kalman filter algorithm and Markov algorithm to process them by MATLAB.

### *Simulation Results*



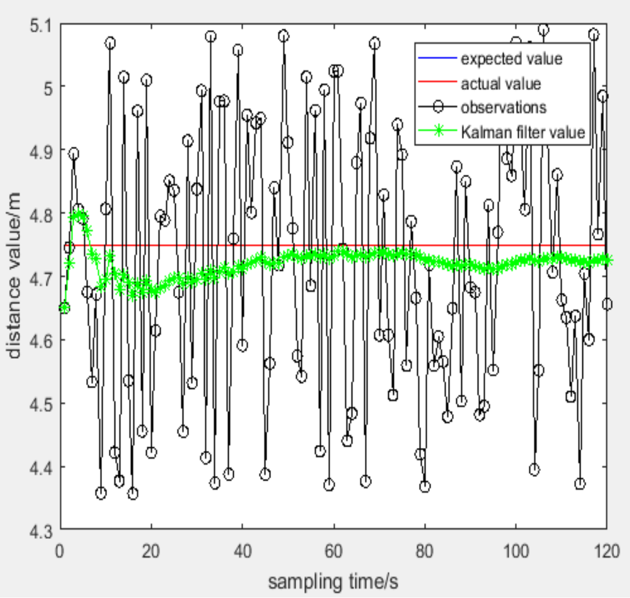
| | Anchor ID | X (m) | Y (m) | Z (m) |
|---|---|---|---|---|
| ☑ | 0 | 0.00 | 0.00 | 0.50 |
| ☑ | 1 | -0.30 | 4.00 | 0.50 |
| ☑ | 2 | 4.50 | 3.50 | 0.50 |
| □ | 3 | 5.00 | 5.00 | 0.00 |

| | Tag ID/Label | X (m) | Y (m) | Z (m) | R95 (m) | Anc 0 range (m) | Anc 1 range (m) | Anc 2 range (m) | Anc 3 range (m) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | ☑ Tag 1 | 3.540 | 1.511 | -2.084 | ☑ 0.089 | 4.669 | 5.297 | 3.399 | |

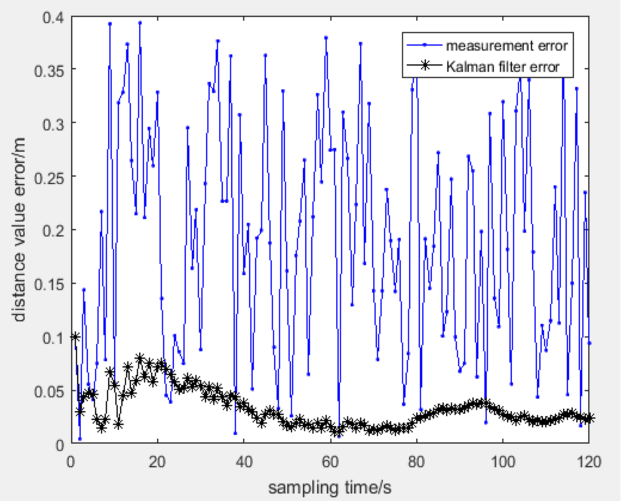
Collect data by decarangeRTLS under TOA algorithm.

Under the original positioning algorithm—TOA algorithm, we collected distance values between anchor and tag on real time positioning software decarangeRTLS， as shown in Fig.9. Taking 120 for each of the three values which are the distance between anchor_A and tag, anchor_B and tag, anchor_C and tag as well as storing 360 sets of distance values in MATLAB.

Due to the influence of NLOS, the values present fluctuation in a larger range. For convenience, taking the distance value of anchor_A and tag as an example, using Kalman filter algorithm to process data, we can get the optimized distance value of anchor_A and tag as shown in Fig.10 and Fig.11.



Optimized value of anchor_A and tag.



Measurement error and Kalman filter error of the distance between anchor_A and tag

Table2 and Table3 present the positioning error in comparative way. Taking into account the distance of Anchor_A and tag as example, we can collect 120 measured value which fluctuate from 4.35 to 5.09 from small to large when we use original algorithm—TOA algorithm; But using Kalman filter algorithm, the fluctuation range of measured distance value of anchor_A and tag is from 4.66-4.80. Compared with real value, the Kalman filter algorithm is easier to narrow the range of data fluctuations and make the measurement error smaller. The distance value of anchor_B and tag, anchor_C and tag are shown the similar results in MATLAB by the same method. And even if adjust the location of robot, Kalman filter algorithm processing results are also very good by MATLAB.

The Original Algorithm (Meter)

| | Real Value | Measured Value |
|---|---|---|
| Anchor_A-tag | 4.73 | 4.35~5.09 |
| Anchor_B-tag | 5.35 | 5.13~5.68 |
| Anchor_C-tag | 3.40 | 3.16~3.66 |

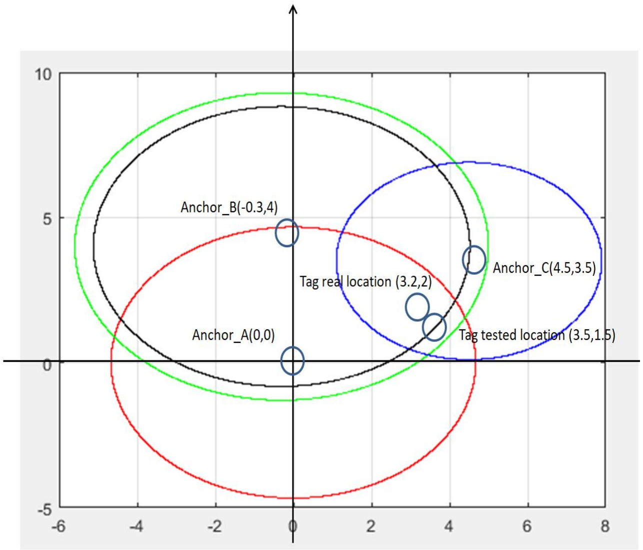
AFTER KALMAN FILTER ALGORITHM (METER)

| | Real Value | Measured Value |
|---|---|---|
| Anchor_A-tag | 4.73 | 4.66~4.80 |
| Anchor_B-tag | 5.35 | 5.28~5.43 |
| Anchor_C-tag | 3.40 | 3.34~3.47 |

### *Experimental Results*

The protocol involve the randomly selection of static coordinate point during the movement of robot, and draw the intersection point of the three circles according to the three distance values of anchor_A and tag, anchor_B and tag, and anchor_C and tag which are optimized by Kalman filter algorithm. As shown, the red circle, green circle and blue circle respectively indicate that the circle is drawn with the distance values of anchor_A and tag, anchor_B and tag, anchor_C and tag (Fig.12). At this time, the tag tested coordinate (3.5,1.5) obtained by the real-time positioning software decarangeRTLS are in the intersection area of the red circle, the green circle and the blue circle, the range is large, and the position is not accurate enough.

According to the blocking probability matrix of Markov (Fig.7) and interference coefficient, The program of Markov algorithm by MATLB calculates the proportion of probability and judges that the distance value of anchor_B and tag may be affected by obstacle-metal under the current position of the robot, and recalculates the distance between anchor_B and tag based on interference coefficient as well as it draws a new circle based on the new distance value between the anchor_B and tag to obtain a black circle (Fig.12). According to the simulation diagram, it can be clearly seen that the intersecting area of the black circle, red circle and blue circle is smaller than the former area which determined by red circle, the green circle and the blue circle. By measuring the real tag coordinate(3.2,2) in practice, it is found that the tag is located at the intersection of the red circle, blue circle and black circle, the conclusion of the simulation is confirmed and the validity of the Markov algorithm is also proved.



Markov algorithm positioning result

## • CONCLUSIONS

In the present work the Kalman filter and Markov algorithms were successfully applied to improve the tag positioning by comparison with the original UWB classic positioning algorithm. Based on the obtained results the proposed solution can be considered for application in logistics warehouses. The performed simulations show that Kalman filter algorithm and Markov algorithm can effectively eliminate NLOS errors and improve the positioning accuracy. However, the Markov algorithm relies on random probability, which requires a large amount of data associated with the transportation trajectory of the real logistics warehouse scenario. Thus, in the simulation part， 360 sets of distance values, 16 sets of probability values and 4 sets of interference coefficients were tested. If the experiment is placed in a real logistics warehouse, the amount of data required will be greatly increased, the types of obstacles will be more complicated. But the research process and method of this simulation experiment can be greatly extended to the experiment of logistics warehouse, providing reference for future research.

## • ACKNOWLEDGMENT

This work was supported by Fundação para a Ciência e Tecnologia Project UIDB/50008/2020, Iscte-Instituto Universitario de Lisboa and Instituto de Telecomunicaçẽs.

## • REFERENCES

Xin Li and Fucheng Cao, "Location Based TOA Algorithm for UWB Wireless Body Area Networks," 2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing.

Do-Jin An and Joon-Ho Lee, "Derivation of an Approximate Location Estimate in Angle-of-Arrival Based Localization in the Presence of Angle-of-Arrival Estimate Error and Sensor Location Error," 2018 IEEE World Symposium on Communication Engineering (WSCE).

Jelena Kaitovic and Miljko Erić, "TDOA localization in ir UWB systems," 2012 19th International Conference on Systems, Signals and Image Processing (IWSSIP).

Xizhong Lou and Yaqiu Zhao, "High-Accuracy Positioning Algorithm Based on UWB," 2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM).

Lei Ding, Xinhua Zhu and Tong Zhou , "Research on UWB-Based Indoor Ranging Positioning Technolog and a Method to Improve Accuracy," 2018 IEEE Region Ten Symposium (Tensymp).

Harris Perakis and Vassilis Gikas , "Evaluation of range error calibration models for indoor UWB positioning applications," 2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 24-27 September 2018, Nantes, France.

Simon Haykin, "Kalman Filtering and Neural Networks," A Wiley-Interscience Publication, 2001.

Zirui Wang, Shaoxian Li, Zhengyuan Zhang and Fan Lv, "Research on UWB Positioning Accuracy in Warehouse Environment," Procedia Computer Science 131 (2018) 946–951.

## Anexo A

Por vezes surge confusão entre Anexo e Apêndice, eis a diferença:

"apêndices e anexos em tudo se assemelham, residindo a única diferença na sua autoria: no primeiro caso, o autor é o próprio autor do texto; no segundo, a autoria é de terceiros." (Oliveira 2011, 7).

**Anexo B**

## Apêndice A

*Designações em português e inglês a colocar na capa da Dissertação*

| Português | Inglês |
|---|---|
| Departamento de Ciências e Tecnologias da Informação | Department of Information Science and Technology |
| Dissertação submetida como requisito parcial para obtenção do grau de | Dissertation submitted as partial fulfillment of the requirements for the degree of |
| Mestrado em Engenharia de Telecomunicações e Informática | Master in Telecommunications and Computer Engineering |
| Mestrado em Engenharia Informática | Master in Computer Science Engineering |
| Mestrado em Gestão de Sistemas de Informação | Master in Information Systems Management |
| Mestrado em Informática e Gestão | Master in Computer Science and Business Management |
| Mestrado em Sistemas Integrados de Apoio à Decisão | Master in Business Intelligence |
| Mestrado em Software de Código Aberto | Master in Open Source Software |

# Apêndice B