UNIVERSITY
OF TURKU

Positioning and Sensing System Based on Impulse Radio Ultra-Wideband Technology

Smart Systems
Master's Degree Programme in Information and Communication Technology
Department of Computing, Faculty of Technology
Master of Science in Technology Thesis

Author:
Shule Wang

Supervisors:
MSc. Paola Torrico Moron
Assoc. Prof. Tomi Westerlund

July 2022

Master of Science in Technology Thesis
Department of Computing, Faculty of Technology
University of Turku

Subject: Smart Systems
Programme: Master's Degree Programme in Information and Communication Technology
Author: Shule Wang
Title: Research on Positioning and Sensing System Based on Impulse Radio Ultra-Wideband Technology
Number of pages: 101 pages
Date: July 2022

Impulse Radio Ultra-Wideband (IR-UWB) is a wireless carrier communication technology using nanosecond non-sinusoidal narrow pulses to transmit data. Therefore, the IR-UWB signal has a high resolution in the time domain and is suitable for high-precision positioning or sensing systems in IIoT scenarios. This thesis designs and implements a high-precision positioning system and a contactless sensing system based on the high temporal resolution characteristics of IR-UWB technology. The feasibility of the two applications in the IIoT is evaluated, which provides a reference for human-machine-thing positioning and human-machine interaction sensing technology in large smart factories. By analyzing the commonly used positioning algorithms in IR-UWB systems, this thesis designs an IR-UWB relative positioning system based on the time of flight algorithm. The system uses the IR-UWB transceiver modules to obtain the distance data and calculates the relative position between the two individuals through the proposed relative positioning algorithm. An improved algorithm is proposed to simplify the system hardware, reducing the three serial port modules used in the positioning system to one. Based on the time of flight algorithm, this thesis also implements a contactless gesture sensing system with IR-UWB. The IR-UWB signal is sparsified by downsampling, and then the feature information of the signal is obtained by level-crossing sampling. Finally, a spiking neural network is used as the recognition algorithm to classify hand gestures.

Keywords: Impulse-Radio Ultra-Wideband, TOF Algorithm, Indoor Positioning, Hand Gesture Recognition, Spiking Neural Network.

Contents

1 Introduction 1

1.1 Background 1

1.2 IR-UWB Technology 3

1.2.1 IR-UWB Technology in Positioning Systems 5

1.2.2 IR-UWB Technology in HGR Systems 6

1.2.3 IR-UWB Technology in Other Applications 7

1.3 Thesis Work and Contributions 9

1.4 Thesis Organization 11

2 Positioning System Design 13

2.1 Overview 13

2.2 IR-UWB Ranging Methods 14

2.2.1 Single-Sided Two-way Ranging 14

2.2.2 Double-Sided Two-way Ranging 15

2.3 IR-UWB Positioning Algorithms 16

2.3.1 Time of Arrival/Time of Flight 16

2.3.2 Time Difference of Arrival 19

2.3.3 Angle of Arrival 21

2.3.4 Evaluation of Positioning Algorithms 23

2.4 IR-UWB Relative Positioning System Design 24

2.4.1 System Architecture Design 24

2.4.2 Positioning Model 25

2.4.3 Hardware Design 26

2.4.4 Positioning Algorithm Design 27

2.5 Summary 29

3 Positioning System Implementation and Experiments 30

3.1 Overview 30

3.2 Implementation of IR-UWB Ranging Module 30

3.2.1 Ranging Algorithm Implementation 30

3.2.2 Ranging Module Experiment 32

3.3 Implementation of IR-UWB Relative Positioning System 34

3.3.1 Hardware Implementation 35

3.3.2 Positioning Algorithm Implementation 36

3.4 Experiment of IR-UWB Relative Positioning System 38

3.4.1 Experimental Setup 38

3.4.2 GUI Display of Positioning Results 39

3.4.3 Accuracy of the Relative Positioning Algorithm 41

3.5 Improvement of IR-UWB Relative Positioning System 44

3.6 Evaluation of Positioning Systems 46

3.7 Summary 48

4 Sensing System Design 49

4.1 Overview 49

4.2 System Design Principle 50

4.2.1 Sensor Design Principle 50

4.2.2 Pre-processing Method Design Principle 52

4.2.3 Classification Method Design Principle 54

4.3 System Architecture Design 57

4.3.1 IR-UWB Contactless Sensor and Dataset 57

4.3.2 Pre-processing Method Design 59

4.3.3 Classification Method Design 60

4.4 Summary 61

5 Sensing System Implementation and Evaluation 62

5.1 Overview 62

5.2 IR-UWB Contactless Sensor Dataset 63

5.3 Pre-processing Method 63

5.4 Classification Method 67

5.5 Experiment of Hand Gesture Recognition System 69

5.5.1 Training Setup 69

5.5.2 Accuracy of the Algorithm 69

5.6 Evaluation of Hand Gesture Recognition System 70

5.6.1 Evaluation of Pre-Processing Methods 70

5.6.2 Evaluation of Classification Algorithm 71

5.6.3 Power Consumption Complexity Calculation 73

5.6.4 Comparison with Other HGR Systems 75

5.7 Summary 75

6 Conclusion and Future Work 77

6.1 Conclusion 77

6.2 Future Work 78

References 80

Academic achievements during the master's degree 91

Acknowledgement 92

List of Figures

1.1 Smart factory positioning and sensing application scenario, adapted from [2] . . 2

1.2 Narrowband signal 4

1.3 IR-UWB signal 4

1.4 IR-UWB positioning system by N. Macoir *et al.* 6

1.5 Overview of the hand gestures processing algorithm, taken from [13] 7

1.6 Healthcare applications based on IR-UWB 8

1.7 The overview scheme of the thesis work 10

2.1 Diagram of the single-side two-way ranging 15

2.2 Diagram of the double-side two-way ranging 16

2.3 TOF positioning algorithm 18

2.4 TDOA positioning algorithm 20

2.5 AOA positioning algorithm 22

2.6 Positioning system architecture design 25

2.7 Relative positioning system model 26

2.8 BPHero-UWB module 27

2.9 Relative positioning algorithm design 28

3.1 Message transmission between IR-UWB Modules with DS-TWR 31

3.2 Experimental setup of ranging accuracy test 32

3.3 Measurement data results 34

3.4 Original hardware system design of relative positioning system 35

3.5 Improved hardware system design of relative positioning system 36

3.6 Actual hardware architecture of relative positioning system 37

3.7 Experimental setup of IR-UWB relative positioning system 39

3.8 Eight positional situations for system A and system B 40

3.9 Real-time GUI display of positioning results 41

3.10 Measurement coordinate results 42

4.1 The IR-UWB contactless sensors 52

4.2 Hand gestures recognition system architecture 57

4.3 The overview of the 6 dynamic hand gestures dataset, adapted from [69] 58

4.4 Sampling approach comparison 60

5.1 IR-UWB signal for an example recording (Finger Slide) 64

5.2 Data matrix of IR-UWB contactless sensor 64

5.3 LC sampling of one data point 65

5.4 IR-UWB signals after LC-sampling 66

5.5 Spiking MLP *‘512-fc128-fc32-fc5’*(fc means full connection) 68

5.6 Accuracy under different firing rates and sparseness 73

List of Tables

2.1 Comparison of positioning algorithms 23

3.1 Accuracy of ranging modules 33

3.2 Accuracy of the relative positioning system 43

3.3 Comparison with other positioning methods 47

4.1 Parameters used to record the 5 and 6 gestures dataset 59

5.1 Parameters setting in experiments, adapted from [77] 70

5.2 Accuracy of the algorithm on the respective train/test set 70

5.3 Algorithm performance under different pre-processing methods 72

5.4 Comparison with other hand gesture recognition methods 76

1 Introduction

1.1 Background

At the HANNOVER MESSE in April 2013, the German Industry 4.0 Working Group published a report entitled *Securing the future of German manufacturing industry: Recommendations for implementing strategic initiative INDUSTRIE 4.0*[1]. Since then, the German government, industry, and academia have listed Industry 4.0 as a strategic goal in the future, which also marks the beginning of the fourth industrial revolution. The second and third industrial revolutions have completed the automation of production equipment and liberated human beings from tedious and heavy manual labor. Then the goal of the fourth industrial revolution is to truly realize the intelligent construction of smart factories and even the entire industry.

In addition, with the development and application of high-tech such as the Internet of Things (IoT) and artificial intelligence (AI), related research on the integration of human-machine-thing has attracted widespread attention. IIoT and smart factories are typical staged applications in the historical process of human society moving towards an intelligent integration of humans, machines, and things society. Precise positioning technology and contactless human-computer interaction sensing technology are two key technologies to realize smart factories and the IIoT.

Figure 1.1 shows a common work scenario in a smart factory, modified from the positioning system diagram of the IWL IoT platform[2]. The robotic arm processes and packs the products on the assembly line, and then the staff transports the packed products to the next location The positioning system of the smart factory can accurately locate the position of tags on employees, vehicles, and assets in real-time, display the position information of people, vehicles, and objects in the factory control center with zero delays, and conduct safety area management and control, people on-the-job monitoring, vehicles Real-time trajectory monitoring. The gesture recognition system of the smart factory can judge the gesture information of the operator in real time so that the robot arm can execute the corresponding instructions for cargo processing and assembly line production.

Figure 1.1: Smart factory positioning and sensing application scenario, adapted from [2]

However, the precise positioning and sensing systems in a smart factory environment require sensors with high temporal resolution. Among them, the impulse radio ultra-wideband (IR-UWB) sensor has lately received wide attention as a promising solution in smart factories due to its high time-domain resolution[3], [4]. IR-UWB uses extremely narrow pulses as the signal propagation carrier, so the duty cycle of the IR-UWB signal is low. IR-UWB can effectively avoid multipath signals by working in a short time, leading to a high temporal resolution, which is very suitable for high-precision positioning or sensing systems. Moreover, the IR-UWB signal occupies a large bandwidth in the frequency domain and a low power spectral density. Therefore, the power consumption of the IR-UWB system can also be extreme-low, which is very suitable for power-sensitive IoTs applications in the smart factory environment. In recent years, IR-UWB has been applied in some commercial scenarios because of these characteristics. Since the iPhone 11 series of mobile phones, Apple has successively applied its IR-UWB solution in products such as Apple Watch S6, HomePod mini smart speaker, iPhone12/13 series mobile phones, and Airtag wireless trackers. Samsung mainly introduced the IR-UWB solution on mobile phones. The specific models include Galaxy Note 20 Ultra, Galaxy Z Fold 2, S21+, and S21 Ultra. At the same time, Samsung also launched the SmartTag+ smart tracker using IR-UWB technology. Xiaomi has launched an IR-UWB connection technology called *One Finger Link* to give full play to the advantages of its smart home ecology and will be able to link with various smart home products in the future. In addition, OPPO has also demonstrated the precise orientation and precise control capabilities between mobile phones and IoT devices at the technology conference.

Based on the above background, this thesis did research on IR-UWB technology. A positioning system and a sensing system are designed and implemented based on the time-of-flight (TOF) algorithm, which provides a new thought for human-machine-thing positioning and human-machine interaction sensing in the smart factory environment. We built two experimental platforms to test systems, and the results show high accuracy in both positioning and hand gesture recognition (HGR) systems due to the high temporal resolution of IR-UWB.

1.2 IR-UWB Technology

IR-UWB is a wireless carrier communication technology that uses non-sinusoidal narrow pulses to transmit data. The working frequency band is from 3.25 GHz to 6.75 GHz, and the typical bandwidth is 500 MHz or 1 GHz. Although IR-UWB uses wireless communication, its data transfer rate can reach more than a few hundred megabits per second, enabling sub-nanometer precise timing. Unlike traditional wireless radio frequency signals (Figure 1.2) with fixed frequency carriers (such as Bluetooth is 2.4 GHz), IR-UWB (Figure 1.3) only sends pulse signals with a narrow width (such as 1 ns) when sending data, which is natally suitable for event-driven mode.

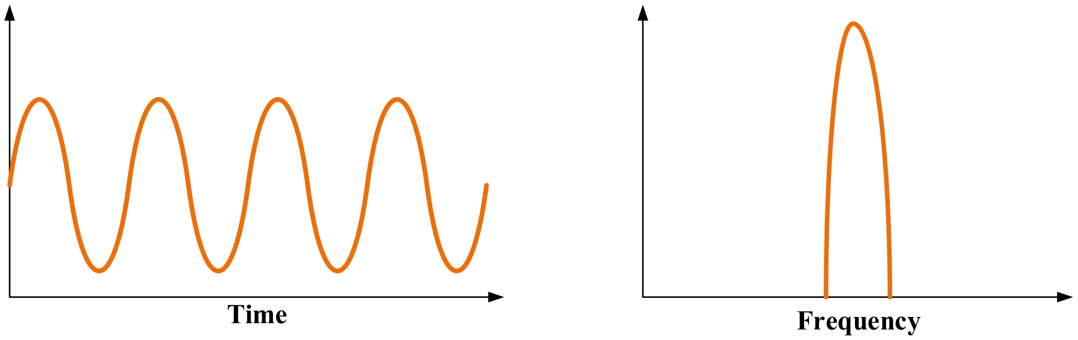


Figure 1.2: Narrowband signal

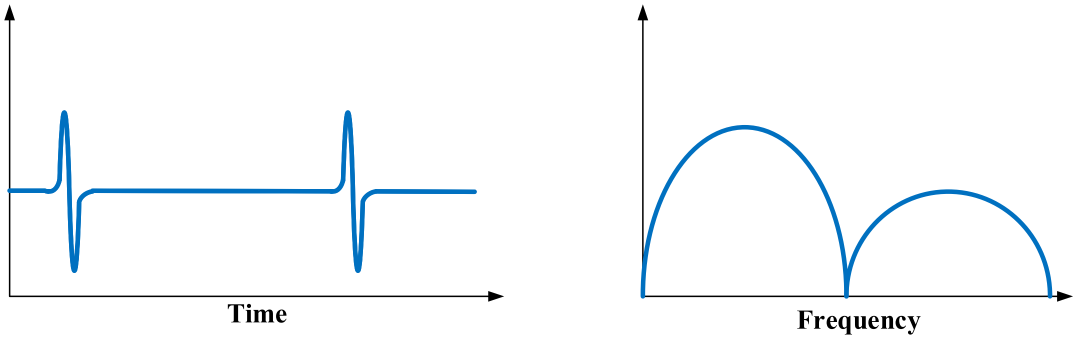


Figure 1.3: IR-UWB signal

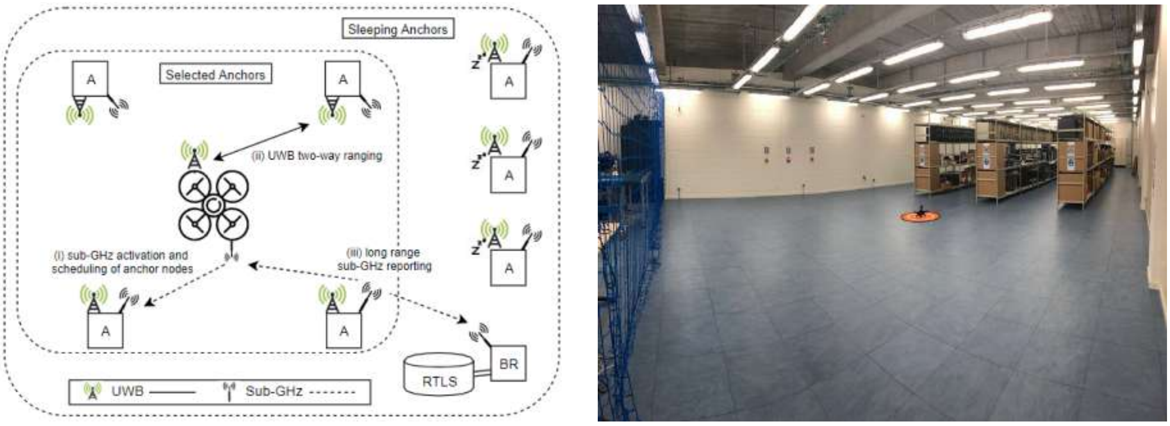
The IR-UWB signal occupies a large frequency domain, so it is called ultra-wideband. Because the IR-UWB time-domain signal pulse is relatively narrow, it has a good resolution in time and space, and it is easier to resist the influence of the common multipath effect of indoor signal transmission. Therefore, the accuracy of ranging and positioning is relatively high. Furthermore, signals in the form of spikes are very similar to neural signals, which are well suited for neuromorphic processing systems.

IR-UWB has a high temporal resolution because it uses short pulses for communication in the time domain. Its extremely-low power spectral density derives from the wide bandwidth in the frequency domain[5]. Therefore, IR-UWB has been widely used in some scenarios that are sensitive to accuracy and power consumption, such as indoor positioning[6]–[10], human sensing[11]–[13], health monitoring[14], [15], and people counting[16].

1.2.1 IR-UWB Technology in Positioning Systems

N. Macoir *et al.* created a multi-technology IR-UWB localization MAC protocol for IIoT applications that may be used for unmanned aerial vehicle- (UAV-) based inventory management [7]. Figure 1.4a depicts the high-level system diagram of the IR-UWB localization system they proposed. Multiple IR-UWB anchors were placed throughout the warehouse by them, who also created a procedure to save energy usage by turning the IR-UWB modules off when no drones were in the area. Instead, the IR-UWB anchor node uses its low-power sub-GHz radio to listen continuously. When the drone is nearby, the drone will use the low-energy sub-GHz activation beacon to activate the anchor and range with the tag module on the drone during short timeslots. The positioning mechanism finally attained a 5 cm accuracy. In a later work, N. Macoir *et al.* additionally evaluated the planned IR-UWB system in two separate scenarios: (i) automated drone inventory management navigation and (ii) placing runners on an undeveloped indoor track [8]. Figure 1.4b depicts the IDLABimec-UGent small-scale warehouse test bed facility. Experimental results illustrate the validity and usefulness of IR-UWB-based positioning systems for real-world industrial applications.

In the field of rescue, T. H. Mogale *et al.* designed a portable wireless sensor network for personnel tracking based on IR-UWB technology for emergency scenarios[9]. All IR-UWB modules are derived from the secondary development of Decawave's DWM1000 module and


Sleeping Anchors
Selected Anchors
A
(ii) UWB two-way ranging
(i) sub-GHz activation and scheduling of anchor nodes
(iii) long range sub-GHz reporting
BR
RTLS
UWB
Sub-GHz

(a) IR-UWB positioning system, taken from [7]   (b) Small-scale warehouse test bed, taken from [8]

Figure 1.4: IR-UWB positioning system by N. Macoir *et al.*

define modules as anchors or tags through programming. The wireless sensor network consists of four anchor points and one label, and the sink node connects to the PC. Firefighters only need to wear the tag, and through the TOA positioning algorithm, the system can understand its location in real-time, helping to coordinate rescue activities in the building. Sometimes, even in a global navigation satellite system (GNSS), IR-UWB can be used for localization error reduction and the accuracy increasing. For instance, Y.-C. Chen *et al.* designed a more precise UAV traffic control system by integrating GPS, LoRaWAN, and IR-UWB range measurement [10]. As IR-UWB transceivers, they employed Decawave DW1000 chips and dispersed many anchor nodes around the test area's structures. Real-world tests revealed that the technology could cut the inaccuracy of outside relative location between the devices under test from 6 meters to just 10 centimeters. IR-UWB can help locate and keep an eye on UAVs in urban environments.

1.2.2 IR-UWB Technology in HGR Systems

Li *et al.* propose an IR-UWB radar-based method for hand gestures recognition using ShuffleNet V2[11]. They collected the radar data of 7 gestures and performed 2D fast Fourier transform (FFT) to get RGB images, called Doppler processing. Then after clutter suppression by subtracting background and noise filtering, they get a 224*224 size continuous Range-Doppler

Map. ShuffleNet V2, a masterpiece of lightweight convolutional neural network (CNN), was used to train the processed signals and achieved 98.52% accuracy on seven gestures. Ahmed *et al.* propose a gesture recognition system using IR-UWB based on another CNN algorithm, GoogLeNet [12]. They first convert the 1D IR-UWB sensor signals into 3D image patterns and then use a variant of GoogLeNet based on the Inception module to analyze the patterns in the images to recognize different gestures. The proposed model can classify eight gestures with 95% accuracy.

Scherer *et al.* illustrated a model mixing temporal convolutional network (TCN) and CNN model in Figure 1.5 bringing about high accuracy and low memory footprint by using IR-UWB radar for data processing[13]. What's more, they came up with new hand gesture recognition with a low-power processor and low-power sensor, which added weight to the embedded devices' gesture recognition. They put various kinds of hand gestures in their dataset, which achieve an accuracy of 89.52% (11 gestures) and 93.83% (5 gestures) with a surprisingly low power consumption of 21mW. The conclusion meant that the model has high effectiveness and potential.

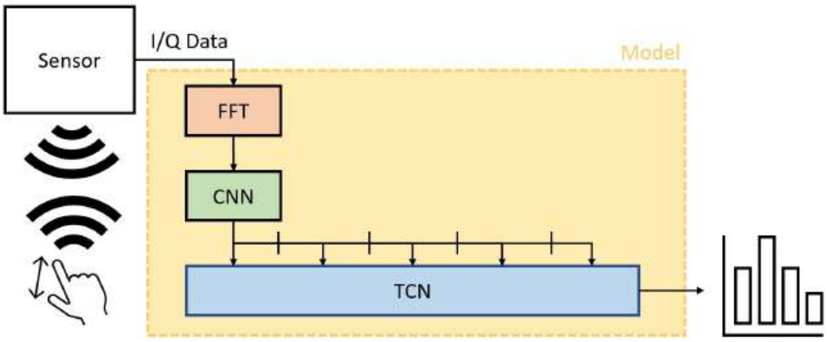
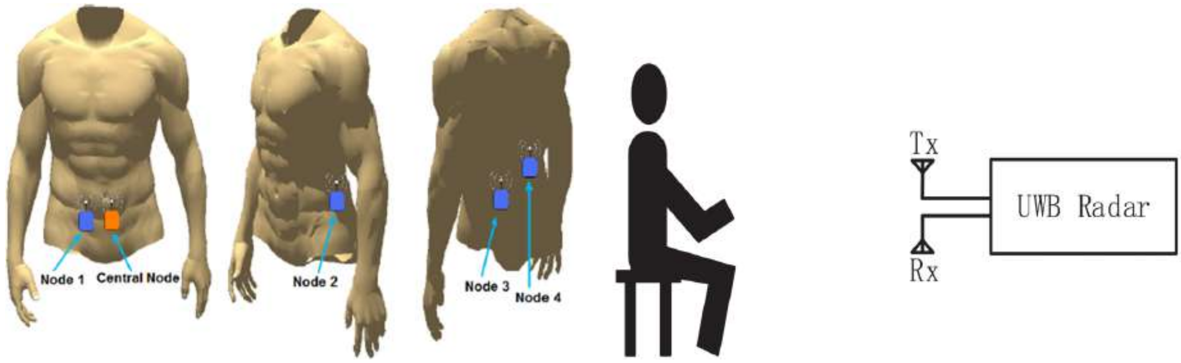


Figure 1.5: Overview of the hand gestures processing algorithm, taken from [13]

1.2.3 IR-UWB Technology in Other Applications

Chehri *et al.* use IR-UWB modules for a closed wireless network named wireless body area networks [14]. These static biomedical nodes are installed on the patient's body to collect vital data and send it to a central node wirelessly for subsequent analysis by healthcare professionals. Through experiments, they evaluated the performance of the IR-UWB monitoring system and, for example, the influence of node location, the number of transmitted symbols, multi-user interference, and inter-symbol interference. It is proved that IR-UWB technology has many unique characteristics in safe transmission, low noise, and low energy consumption. Shen *et al.* proposed a new method based on autocorrelation to measure the respiration rate and heartbeat rate by using IR-UWB radar[15]. They collected the breathing data of the human body through IR-UWB radar echoes and then obtained the breathing frequency by applying the FFT. Then an autocorrelation method is used for object localization, followed by variational modal decomposition algorithms and separation of respiration and heartbeat information. Finally, they conducted experiments using PulsOn410 IR-UWB radar, and the results also show that the proposed algorithm can achieve higher accuracy with lower complexity.



(a) The system schema, taken from [14]

(b) The system test scenario, taken from [15]

Figure 1.6: Healthcare applications based on IR-UWB

Choi *et al.* used IR-UWB radar as a counting sensor for inbound and outbound people [16]. They proposed a method using two IR-UWB radar sensors to solve the problem of counting people in dense environments such as subway stations or airports. Due to the high resolution of IR-UWB signals, radar sensors can theoretically detect most people entering and leaving. Through the test in the laboratory environment and the actual subway station environment, the average error can be limited to 7%.

Kim *et al.* used IR-UWB radar to collect environmental data and adopted proposed algorithms to determine the presence of humans in indoor environments[17]. They adopted singular value decomposition to estimate clutter signals by comparing various filtering methods. Then the smallest cell-averaging constant false alarm detector and reference method are used to detect people standing or lying down in the indoor environment, respectively. Finally, they used the single-chip radar sensor of the XeThru module X4M03 developed by Noveda to conduct experiments in different environments to verify the robustness.

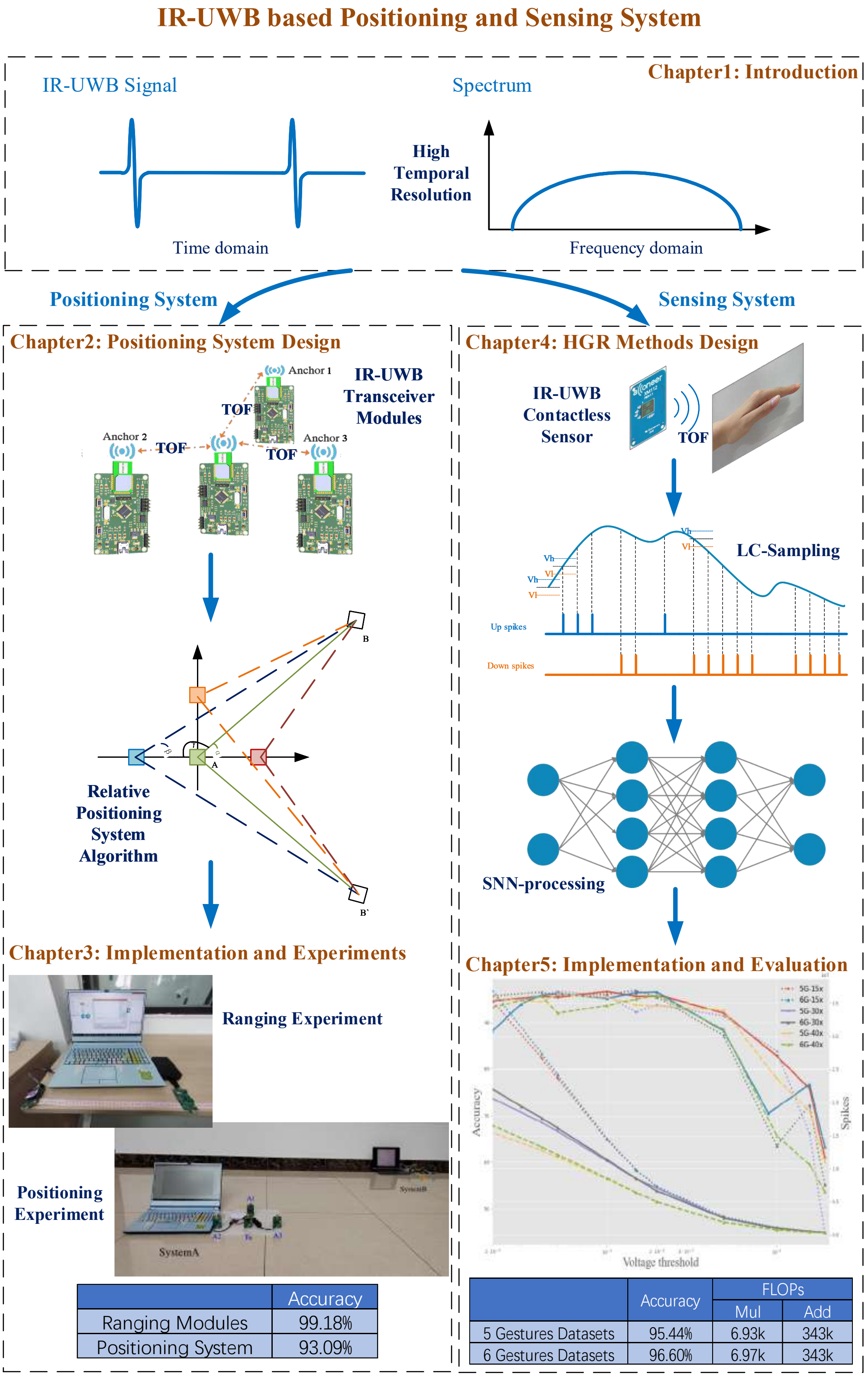
1.3 Thesis Work and Contributions

As a wireless carrier communication technology, IR-UWB signals have a high time-domain resolution due to using nanosecond non-sinusoidal narrow pulses to transmit data. This thesis designs and implements a high-precision positioning and a sensing system based on the high temporal resolution characteristics of IR-UWB technology. The feasibility of the two applications in IIoT is evaluated, which provides a reference for human-machine-thing positioning and human-machine interaction sensing technology in large smart factories.

Figure 1.7 demonstrates an overview of this work, consisting of three parts: introduction, positioning system, and sensing system. In the introduction stage, we obtain the high time resolution characteristic of IR-UWB by analyzing its time-domain and frequency-domain characteristics. Then the IR-UWB technology is applied to positioning and sensing systems. The positioning system contains the positioning algorithms and ranging and positioning experiments and results. The sensing system shows gesture recognition algorithms and their performance, including recognition accuracy on the two datasets and power consumption with floating-point operations per second (FLOPs).

The contributions of this thesis are presented as follows:

1. By analyzing the commonly used positioning algorithms in IR-UWB systems, this thesis designs an IR-UWB relative positioning system based on the TOF algorithm. The system uses



	Accuracy
Ranging Modules	99.18%
Positioning System	93.09%

	Accuracy	FLOPs	
		Mul	Add
5 Gestures Datasets	95.44%	6.93k	343k
6 Gestures Datasets	96.60%	6.97k	343k

Figure 1.7: The overview scheme of the thesis work

the IR-UWB transceiver modules to obtain the distance data and calculates the relative position between the two individuals through the proposed relative positioning algorithm.

2. The platform for ranging and positioning experiments is built using IR-UWB modules from Bphero. The system is tested in a laboratory environment. The results suggest that the system achieved a relatively high accuracy on the positioning. An improved algorithm is proposed to simplify the system hardware, reducing the three serial port modules used in the positioning system to one.

3. Based on the TOF algorithm, this thesis designs and implements a contactless hand gesture recognition system with IR-UWB. The IR-UWB signal is sparsified by downsampling, and the feature is extracted by level-crossing sampling. Finally, a spiking neural network (SNN) is used as the recognition algorithm to classify hand gestures.

4. The system is verified on Acconeer's IR-UWB sensor gesture dataset. Results suggest that our algorithm can achive relatively high accuracy with extremely low power consumption. This thesis also compares the proposed recognition algorithm with the existing gesture recognition methods. The results illustrate that the proposed algorithm has the lowest computational complexity when achieving similar classification accuracy as other works.

1.4 Thesis Organization

This thesis consists of the following six chapters.

- In Chapter 1, we introduce the background of IIoT. IR-UWB technologies have the potential to solve precision problems in IIoT and smart factories applications due to the high temporal resolution. Then we investigate the commercial application of IR-UWB technology. After that, we analyze the application of IR-UWB technology in positioning, HGR, and other systems. It is revealed that IR-UWB technology is a good solution for positioning and sensing systems in smart factories scenarios. In the end, we make a brief introduction to the overview of the work and organization of this thesis.

- In Chapter 2, we introduce the IR-UWB ranging and positioning algorithm first. Then we explain the relative positioning system design in detail, including the system architecture design, positioning model design, hardware design, and positioning algorithm design.

- In Chapter 3, we first introduce the implementation and experiments of IR-UWB ranging modules. Then we show the implementation of relative positioning systems, including hardware architecture and software algorithms. After that, experimental platforms are built to obtain the system's positioning accuracy. The results suggest that the system achieved an accuracy of 99.18% on the ranging and 93.09% on the positioning. Finally, we evaluated other existing positioning systems with WiFi, Bluetooth, Lidar, and Camera sensors and compared them with our system.

- In Chapter 4, we show another application of the IR-UWB: sensing system. Firstly, the design principle of the sensing system is analyzed, such as sensor design principle, data processing design principle, and classification algorithm design principle. Then we introduce the design of the IR-UWB sensor dataset, preprocessing method, and the classification algorithm of the sensing system in detail.

- In Chapter 5, we describe the sensing system implementation from dataset to classification method. Then we analyze the accuracy of the algorithm through multiple experiments. To demonstrate the superiority of the algorithm, we also compare the accuracy and power consumption of our algorithm with other methods. The data indicates that the algorithm has achieved an accuracy of 95.44% on five gesture datasets and 96.60% on six gesture datasets.

- In Chapter 6, we discuss the conclusions from this work and outline future work.

2 Positioning System Design

2.1 Overview

In GNSS-Denied situations, accurate localization is a crucial component of autonomous robot navigation [18]. IR-UWB localization systems, based on wireless radio modules, have recently developed as high-accuracy solutions, on tens of centimeters or even centimeters [19]. In application cases where a localization precision of the order of tens of centimeters is adequate, IR-UWB based localization is likely to take the place of more complicated and expensive motion-capture arenas based on visual markers [20].

In this chapter, we first characterize various techniques used to measure the distance between two IR-UWB transceivers, primarily using one-way and two-way ranging in the TOF algorithm. Then we introduce three kinds of positioning algorithms commonly used in IR-UWB positioning systems: TOF/time of arrival (TOA), time difference of arrival (TDOA), and angle of arrival (AOA), and summarize their advantages and disadvantages. Finally, based on the TOF positioning algorithm, we designed an IR-UWB relative positioning system. The system design includes system architecture design positioning model design, hardware design, and positioning algorithm design.

2.2 IR-UWB Ranging Methods

We can calculate the distance between an IR-UWB transceiver that emits a signal and the reception node by using the flight of time and the known speed of electromagnetic wave propagation in the air. TOF or TOA obtains the distance between devices by multiplying the time of flight by the speed of light. The propagation time of the wireless signal from the transmitter node to the receiver node is determined by recording the sending and receiving timestamps of the range message [21], [22]. TOF can be divided into one-way and two-way ranges depending on how the devices are synced. In one-way ranging, the ranging message can only travel in one direction. The clocks on both must remain precisely synchronized. Clock synchronization is not required for two-way ranging. Due to its decreased complexity, it is a more popular strategy.

The two-way ranging approach is mostly used in IR-UWB ranging. The two-way ranging method requires two-way communication between devices, which may be separated into two approaches based on available information regarding antenna delays and packet processing latency: single-sided two-way ranging (SS-TWR) and double-sided two-way ranging (DS-TWR) [23].

2.2.1 Single-Sided Two-way Ranging

The procedure is shown in Fig. 2.1. Device A eliminates a ranging request message in the SS-TWR method, and device B gets the ranging and answers to a T_{reply} message processing delay. Device A calculates the round-trip delay T_{round} and utilizes equation (2.1) to measure the distance between nodes A and B when it receives the response message from B.

$$T_{prop} = 0.5 * (T_{round} - T_{reply}) \tag{2.1}$$

However, the condition for the establishment of the formula (2.1) is to assume that the propagation times of the two signals are the same, that is, the distance between the sender sending

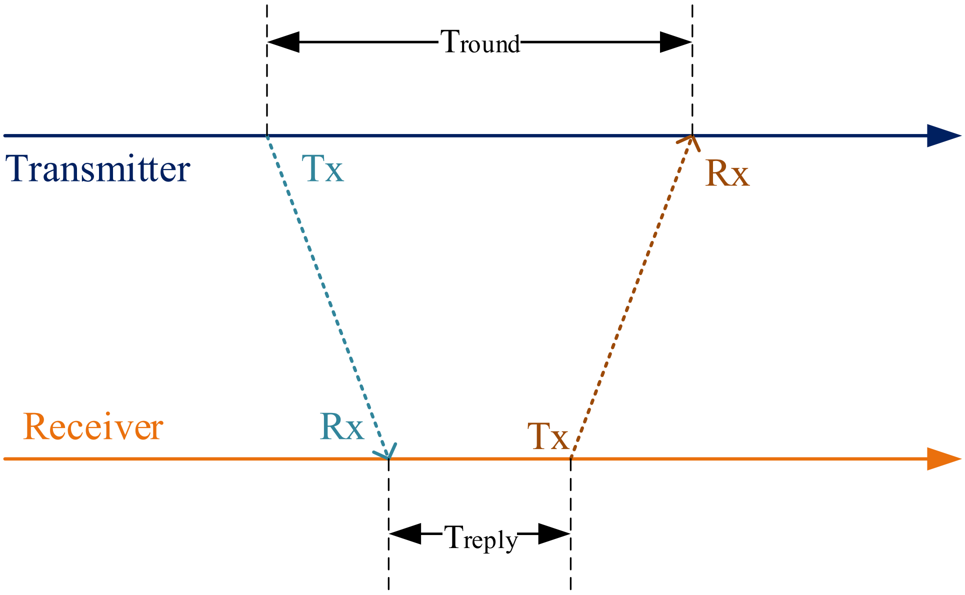


Figure 2.1: Diagram of the single-side two-way ranging

information to the receiver and the receiver sending information to the sender is constant within the interval, which requires T_{reply} to be very short. What's more, due to individual differences in modules, the internal clock frequencies of the two devices cannot be guaranteed to be the same, that is, the speed of time change is not necessarily equal. Therefore, there will be some fixed errors in the SS-TWR method, which cannot be eliminated but reduced. The reduction method is to send the information as quickly as possible after receiving the information to prevent the accumulation of errors. Moreover, since the information is only sent twice between the two devices (once each), there may be interference from obstacles or noise, and the range changes greatly.

2.2.2 Double-Sided Two-way Ranging

In order to greatly reduce the time of T_{reply}, the three-message ranging algorithm: DS-TWR was created. Because in SS-TWR, only two messages are sent between the two devices (once each), there may be obstacles or noise interference, and the ranging changes are large, so it is better to take the average multiple times to improve the measurement. Distance accuracy, here is the ranging principle of DS-TWR. In the DS-TWR algorithm, both Transmitter A and Receiver B send a ranging request, equivalent to the two times SS-TWR by devices A and B. According to equation (2.2), we can calculate the flight time between nodes A and B. Figure 2.2 illustrates the description of variables in the equation. The accuracy is often higher than the SS-TWR method.

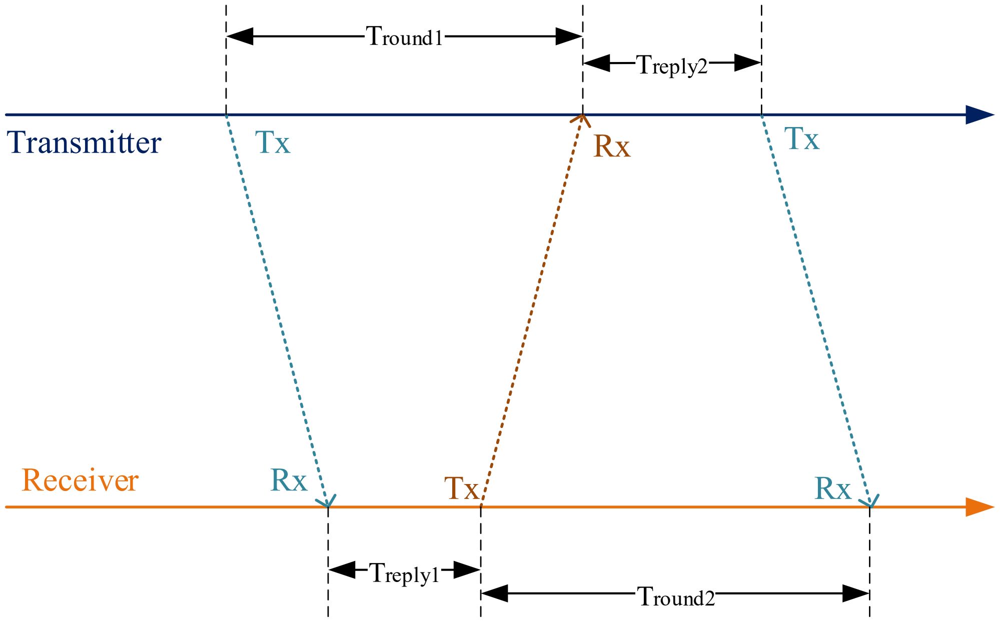


Figure 2.2: Diagram of the double-side two-way ranging

$$T_{prop} = \frac{T_{round1} * T_{round2} - T_{reply1} * T_{reply2}}{T_{round1} + T_{round2} + T_{reply1} + T_{reply2}} \tag{2.2}$$

2.3 IR-UWB Positioning Algorithms

2.3.1 Time of Arrival/Time of Flight

In terms of range, the TOF-based positioning approach is equivalent to the TOA-based positioning method. There is no error due to the clock synchronization variance since the TOF ranging method does not rely on the base station and tag in time synchronization. The precision of the clock determines how quickly the TOF ranging mechanism operates, and a clock offset will result in mistakes. The measurement method in both positive and negative directions is typically used to reduce the ranging error brought on by the clock offset. To do this, the remote base station sends the ranging information, the tag receives it and responds, after which the tag initiates the ranging message, and the remote base station sends it. By figuring out the average value of the flight time, the end base station responds and decreases the time offset between the two, increasing the ranging accuracy. As explained in [24], the coordinates of the IR-UWB positioning base station are known. The location of the IR-UWB positioning tag is determined by drawing three circles using the three-point positioning method after calculating the distance between the tag and the base station.

Figure 2.3 illustrates the working principle of three-circle positioning. A_1, A_2, and A_3 are three IR-UWB base stations, the positions are fixed during deployment. Assuming that the coordinates of three anchors are (x_1, y_1), (x_2, y_2), (x_3, y_3), the coordinates of the required positioning label are $T_0(x_0, y_0)$. With the positioning label T_0 determined by the signal propagation time and d_1, d_2, d_3 as the relative distances between the three positioning base stations, each base station will draw a circle with the relative distance as the radius. The three circular equations may be used to compute the distinct intersection point, and the computation formula is as follows:

$$\begin{cases} \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2} = d_1 = vt_1 \\ \sqrt{(x_0 - x_2)^2 + (y_0 - y_2)^2} = d_2 = vt_2 \\ \sqrt{(x_0 - x_3)^2 + (y_0 - y_3)^2} = d_3 = vt_3 \end{cases} \tag{2.3}$$

TOF is widely used in IIoT scenarios due to its high precision and ease of use. S. G. Pease *et al.* designed a real-time semantic tracking system with TOF and received signal strength indication (RSSI) in IIoT scenarios [25]. They proposed a novel, communication-economical method for ranging based on RSSI/TOF. The performance evaluation shows that the system ranging accuracy is 6 meters with TOF (the ranging range is 40 meters in indoor industrial locations),

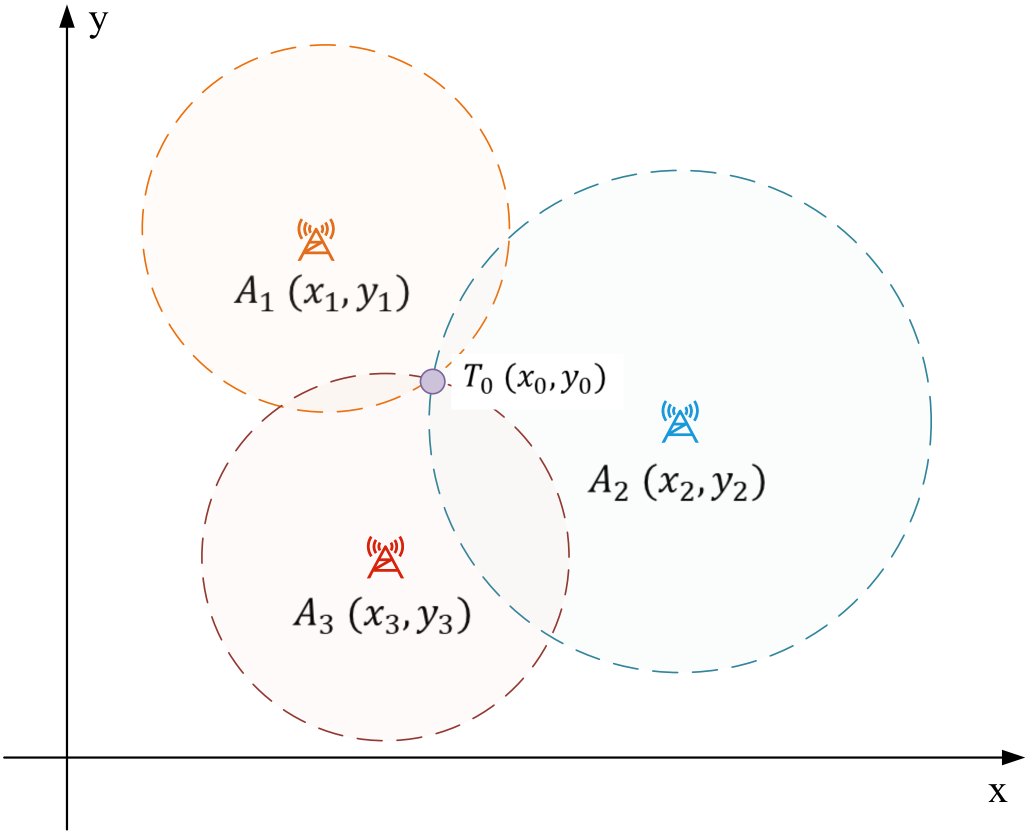


Figure 2.3: TOF positioning algorithm

and the average positioning accuracy is 12.6-13.8 meters. F. Bonafini *et al.* adopted a LoRaWAN network to complement indoor and outdoor real-time location systems (RTLS) [26]. Interestingly, their indoor positioning system uses the DWM1001-Dev board, which uses the DS-TWR ranging method in TOF. By sharing time common sense in LoRaWAN, microsecond-level errors are obtained in UWB ranging devices, resulting in higher positioning accuracy. The experimental results show that the GPS-processed time reference pulse has a maximum jitter of 180 ns with a standard deviation of 40 ns, while if the UWB-processed time reference pulse is considered, the maximum jitter is 3.3 μs with a standard deviation of 0.7 μs.

2.3.2 Time Difference of Arrival

To determine the relative distance between a tag and each of the anchors, TDOA, also known as hyperbolic positioning, evaluates the difference in propagation time between an IR-UWB signal from a transmitting tag and an IR-UWB signal from two or more receiving anchors [27], [28]. Over TOA, it may generally increase localization accuracy. To determine the position of a moving IR-UWB tag, the TDOA method calculates the time difference between signals received by various IR-UWB anchors rather than directly using the difference between the signal emission and arrival times [29]. For clock synchronization between tags and anchors, it is not necessary to provide a particular time signal, but anchors must be linked or synced [23]. Since measuring the precise absolute time is difficult, as stated in [24], TDOA analyzes the delay in the signal's arrival at each IR-UWB positioning base station to calculate the signal's distance from each base station. At the point where three sets of hyperbolas cross, the placement label is situated.

An illustration of the TDOA algorithm is shown in Figure 2.4. The base stations for IR-UWB positioning are located at the following coordinates: $A_1(x_1, y_1)$, $A_2(x_2, y_2)$, $A_3(x_3, y_3)$, and $A_4(x_4, y_4)$. During installation and deployment, base stations A_1, A_2, A_3, and A_4 are in fixed places. The required placement label's coordinates is $R_0(x_0, y_0)$. The pulse signal propagates at a constant rate of $v = 3 * 10^8 km/s$. Three sets of hyperbolas can be obtained, and the intersection of the hyperbolas is the coordinate of the positioning tag O. Assuming that the time it takes for the pulse signal to reach the base stations R1, R2, R3, and R4 from the tag O is t1, t2, t3, and t4, respectively, and that the distance between the signal sent by the positioning tag Ro and the two base stations is a constant. The following formula gives the set of equations needed to solve the coordinates (x_o, y_o):

$$\begin{cases} \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2} - \sqrt{(x_0 - x_4)^2 + (y_0 - y_4)^2} = v(t_1 - t_4) \\ \sqrt{(x_0 - x_2)^2 + (y_0 - y_2)^2} - \sqrt{(x_0 - x_4)^2 + (y_0 - y_4)^2} = v(t_2 - t_4) \\ \sqrt{(x_0 - x_3)^2 + (y_0 - y_3)^2} - \sqrt{(x_0 - x_4)^2 + (y_0 - y_4)^2} = v(t_3 - t_4) \end{cases} \tag{2.4}$$

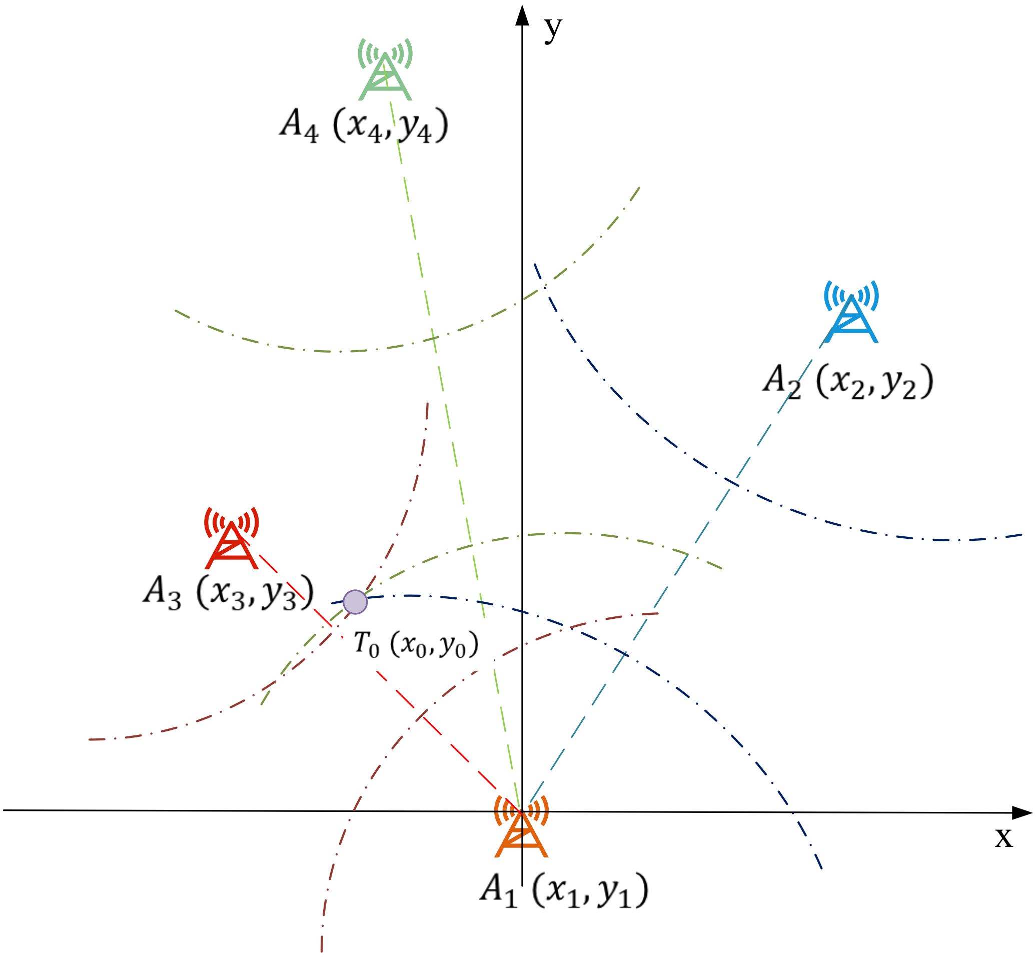


Figure 2.4: TDOA positioning algorithm

M. Martaló *et al.* solved the problem of object localization in a special scenario where the object is outside the area where the anchor is located [27]. Based on the TDOA algorithm, they compared and analyzed the linear hyperbolic positioning system (geometric) or particle swarm optimization (soft computing-oriented) algorithms. The results show that a UWB TDOA-based hotspot can accurately estimate the position around the target itself, and is also suitable for medium/large target-hotspot distances. In particular, the designed algorithm can estimate the target angle of arrival with an error of only about a few degrees. Because the TDOA algorithm is highly dependent on the synchronization of the module clock, the positioning accuracy is relatively low. P. Zhao *et al.* proposed a high precision UWB-TDOA positioning system for unmanned vehicle systems [28]. Referring to the working principle of GNSS-RTK (real-time kinematic), a differential UWB positioning system is developed to reduce synchronization errors and improve system stability. The experiment also proved that the system can achieve a positioning accuracy of 10 cm.

2.3.3 Angle of Arrival

The AOA positioning is a positioning algorithm based on the angle of incidence of the signal. Generally speaking, a simple AOA positioning system only needs two base stations, as shown in Figure 2.5. Base stations A_1 and A_2 will radiate electrical signals outward 360°. When the echo signal in a certain direction is obtained, the angle between the connection between the devices and the reference direction will be recorded as α and β with reference The direction is north. According to the size of the included angle, we can draw two rays l_1, l_2. At this time, the intersection of l_1 and l_2 is the specific position of the device. The detailed calculation formula is as follows.

The positions of the two base stations and the device form a triangle, so to get the position of the device $T_0(x_0, y_0)$, you need to use some knowledge of trigonometric functions. Suppose we know that the coordinates of the IR-UWB positioning base stations are $A_1(x_1, y_1)$ and $A_2(x_2, y_2)$, and the angles between the device and the reference direction are α and β. Therefore, the angle between the device and base station A and base station B is $\angle 1 + \angle 2 = \alpha + \beta$. Assume that the distance between the base station A_1 and the device is d_1.

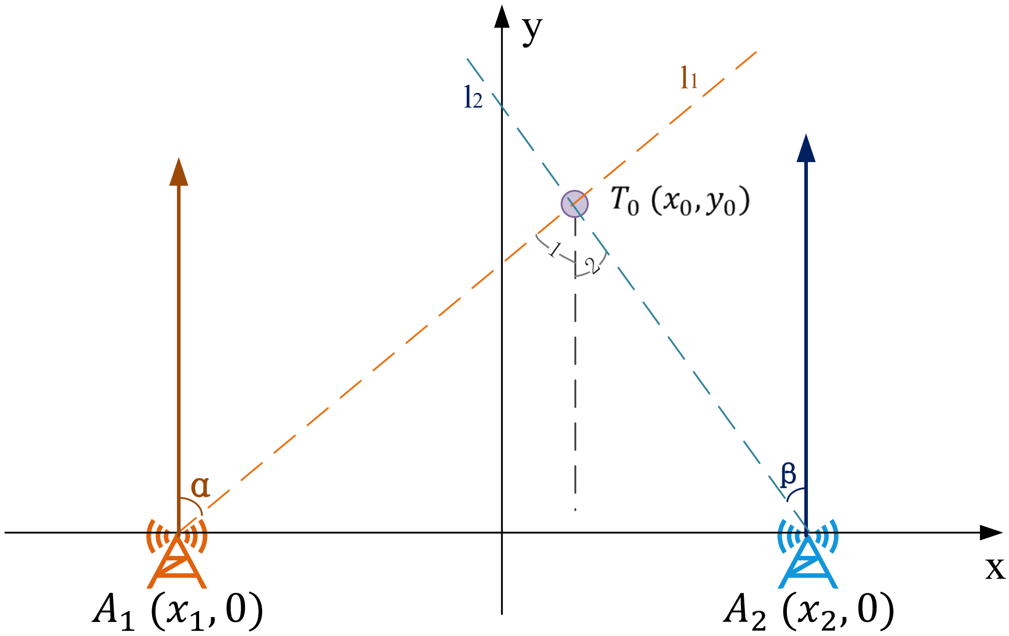


Figure 2.5: AOA positioning algorithm

$$\frac{x_1 + x_2}{sin(\alpha + \beta)} = \frac{d_1}{sin(\pi - \beta)} \tag{2.5}$$

$$\begin{cases} x_0 - x_1 = d_1 cos(\pi - \alpha) \\ y_0 - \ 0 \ = d_1 sin(\pi - \alpha) \end{cases} \tag{2.6}$$

The AOA algorithm is generally not used alone in applications, because the angular resolution of the UWB antenna is very demanding. It is often used in combination with TOF or TDOA in localization algorithms. L. Taponecco *et al.* designed a UWB localization system jointing TOA and AOA estimation [30]. The biggest advantage of this method is that the number of base stations can be drastically reduced to one, which will greatly reduce costs in large factories. It is experimentally demonstrated that a ranging error of about 10 cm and an angular error of about 1° is achieved using a pulse with a 3 GHz sampling rate and two antennas with a distance of 50 cm.

2.3.4 Evaluation of Positioning Algorithms

L. Barbieri *et al.* evaluated the performance of three commercial off-the-shelf UWB technologies: Decawave, Sewio, and Ubisense [31]. According to their industrial experiments, the positioning error of Decawave devices with TOF estimation is lower than 18 cm in line-of-sight conditions, while 33 cm and 28 cm are achieved in the non-line-of-sight and shadowing cases. However, Sewio attains an error value of 1.17 m with the TDOA algorithm. Ubisense achieved less than 40cm error combining TDOA and AOA measurements. Table 2.1 summarizes some characteristic comparisons of the three positioning algorithms: TOA/TOF, TDOA, and AOA.

Table 2.1: Comparison of positioning algorithms

	TOA/TOF	**TDOA**	**AOA**
Precision	high	medium	low
Min Anchors	3	3	2
Scalability	low	high	high
Robustness	high	low	high
Battery Life	short	long	long
Availability	low	high	high
Cost per module	low	low	high

All in all, the algorithm implementation of TOF is relatively simple, and the accuracy is the highest. However, since the base station and the tag need to communicate multiple times to measure the distance, the module consumes a lot of power. Moreover, the positioning needs to deploy base stations in advance, so the scalability of the system is poor. To expand the positioning range, it is necessary to deploy new base stations. The TDOA algorithm requires each base station to perform high-precision time synchronization with each other first, so the development of the algorithm is difficult and the algorithm is relatively complex. In addition, the positioning result strongly depends on the time synchronization of the module, so the accuracy is not as good as the TOF algorithm. However, since only one communication between the base station and the tag is required for positioning, the relative power consumption of the system will be very low, and it is very easy to expand to a larger positioning range. As for the AOA algorithm, the biggest advantage is that the number of base stations required is very small, and only two base stations are needed to obtain the label position. However, to obtain the angle information, a high-cost IR-UWB module is required, and the antenna of the module is required to transmit in a ring shape and to obtain accurate angle information. To make matters worse, this positioning method is sensitive to signal occlusion and is generally not used in positioning systems alone.

2.4 IR-UWB Relative Positioning System Design

In the traditional positioning system, we can get the specific position of the tag in the indoor environment with fixed anchors. However, in some specific scenarios, we need to know the relative positions of Tags rather than absolute position information, such as multi-robot collaborative exploration and mapping, search and rescue, and formation control. Therefore, we design a relative positioning system based on IR-UWB for the above cases. In this section, we will introduce the design of the relative positioning system in detail, mainly including system architecture, system hardware, and software design.

2.4.1 System Architecture Design

Figure 2.6 shows the architecture of the relative positioning system based on IR-UWB, including the hardware architecture, positioning algorithm, and GUI display.

In the hardware architecture, the three base station modules first calculate the distance between the tag based on the DS-TWR algorithm and then transmit data to the host computer through the USB to TTL serial port. In the implementation part of the positioning algorithm, after the host computer receives the three distance information, it first distinguishes the data

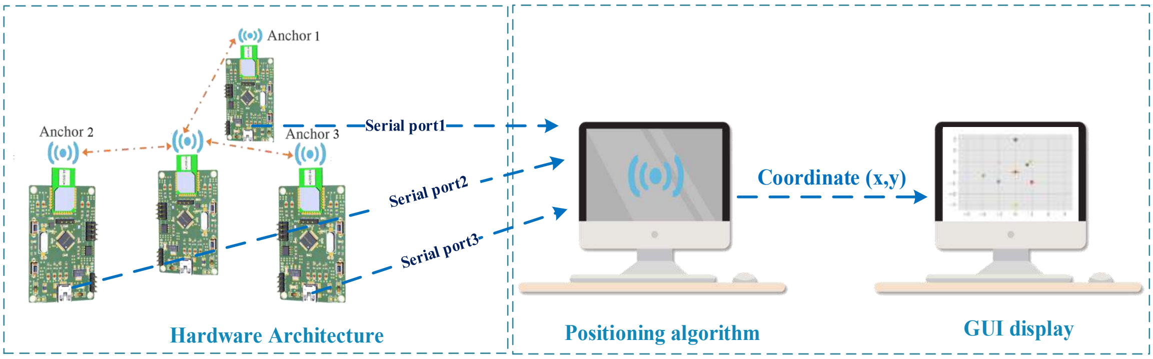


Figure 2.6: Positioning system architecture design

source according to the serial port ID and then calculates the measured label position through the positioning algorithm we designed. Finally, in the result GUI display module, the position information is displayed in the GUI interface in real-time through the *matplotlib.pyplot* library in python.

2.4.2 Positioning Model

Figure 2.7 shows the relative positioning system model we proposed.

We use robots A, B, and C to represent the multi-robot task scenario, and the robots communicate with each other through the IR-UWB transceiver module mounted on it. Three receiving modules and one transmitting module are deployed on each robot, and the receiving module can receive the IR-UWB signals transmitted by all the transmitting modules within the range for distance measurement. Because the IR-UWB module will send a series of messages during the ranging process, we also write the ID of the module into the message to distinguish the source of the message for the receiving module. For example, robot A can communicate with the transmitting module T_b of robot B through the three receiving modules A_1, A_2 and R_3, obtain the distance of the three sides: d_1, d_2, d_3 through the ranging algorithm. Then the relative positioning algorithm we designed is used to calculate the relative position of robot B in the coordinate system of the T_a.

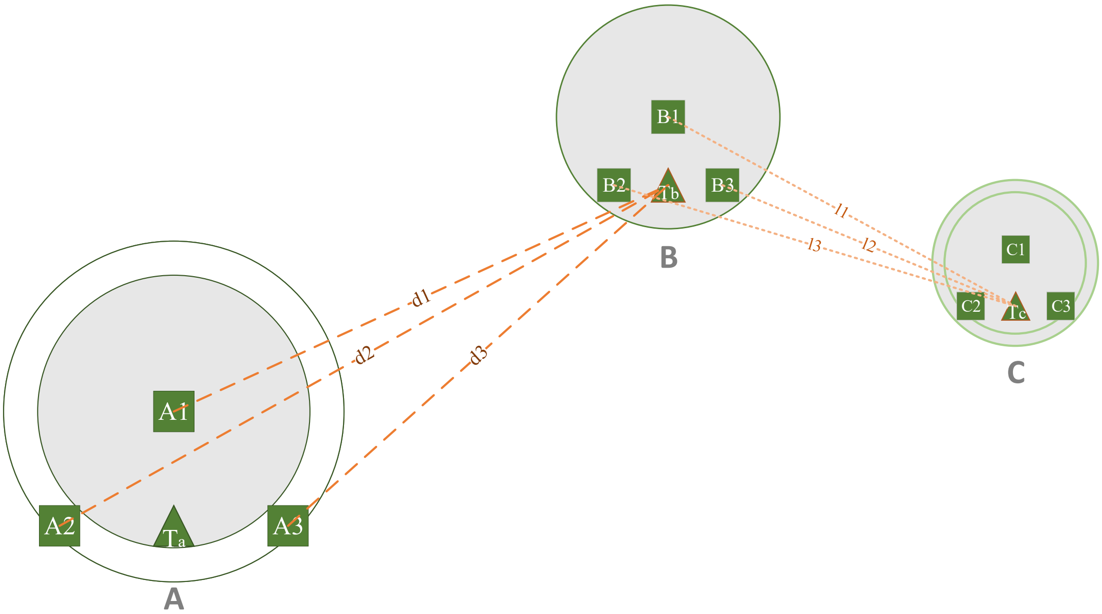


Figure 2.7: Relative positioning system model

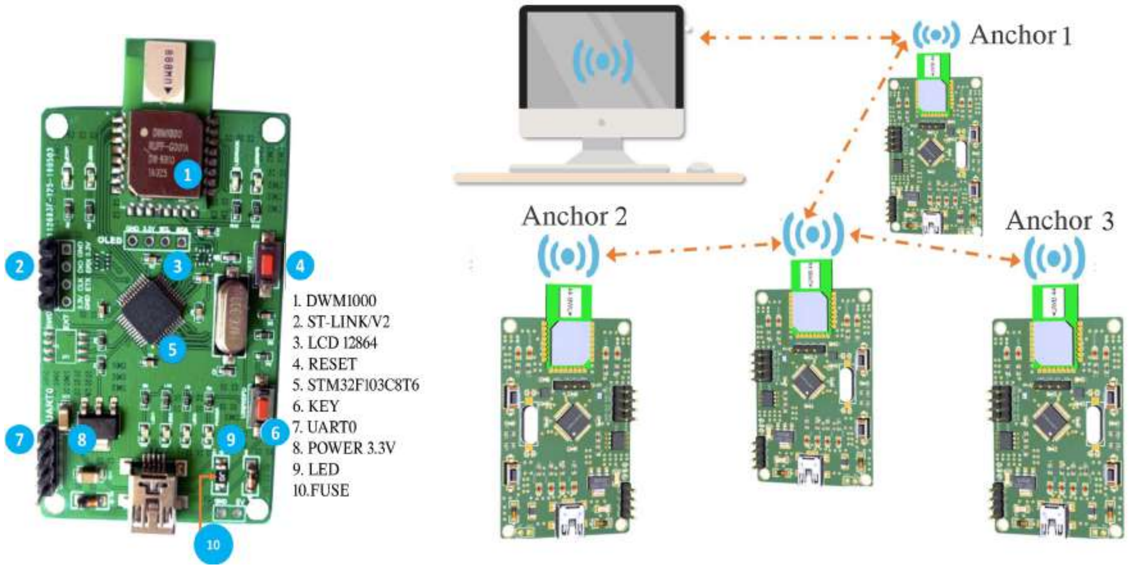
2.4.3 Hardware Design

The hardware we use is the BP-TWR-50 IR-UWB positioning module developed by the Bphero team based on the DWM1000[1] module. The DWM1000 is a wireless IR-UWB transceiver module developed by Decawave's DW1000 IC[2] that conforms to the $IEEE802.15.4 - 2011$ IR-UWB standard. The module enables object localization in RTLS to 10 cm indoor accuracy, high data rate communication up to 6.8 Mbps through coherent receiver technology, and an excellent communication range of up to 300 meters. On this basis, the Bphero team uses the officially provided TWR algorithm to calculate the distance between two IR-UWB modules, and then increases the ranging accuracy through functions such as mean filtering, and increasing the LCD screen to display the distance information between modules in real-time. The most important thing is that the module provides ST-LINK/V2 and USB to TTL interfaces, which are convenient for users to debug and load programs and connect to serial ports. A single BPHero-UWB module is shown in Figure 2.8a. It is equipped with a DWM1000 UWB ranging module

[1] https://www.decawave.com/product/dwm1000-module/

[2] https://www.decawave.com/product/dw1000-radio-ic/

and uses STM32F103C8T6 as MCU. In the relative positioning system, each robot is equipped with a set of BP-TWR-50 positioning modules (Fig. 2.8b), and the positioning algorithm is calculated by placing each module at a certain distance.



(a) BPHero-UWB module

(b) BPHero-UWB position module

Figure 2.8: BPHero-UWB module

2.4.4 Positioning Algorithm Design

In order to calculate the relative coordinates between robots, we design a relative positioning algorithm based on the TOF algorithm, shown in Figure 2.9. The coordinates are calculated by measuring the distance between the transceiver modules. Since the module is deployed in advance, we can know the specific location of the IR-UWB transceiver module on each robot. According to formula (2.3), the coordinates (x, y) of robot B can be solved. However, to simplify the calculation process, we take the transmitting module of robot A as the origin, and the three receiving modules C, D, and E are placed at $(0, a)$, $(-a, 0)$, $(a, 0)$, respectively. First, we can obtain the flight time from the transmitter module of robot B to the three-receiver modules of robot A through the DS-TWR algorithm, to calulate the three distances of d_1, d_2, and d_3. Then, the following formula is listed based on the cosine law, so that the distance between robot A and robot B and the angle γ between the connection line and the negative x-axis can be calculated.

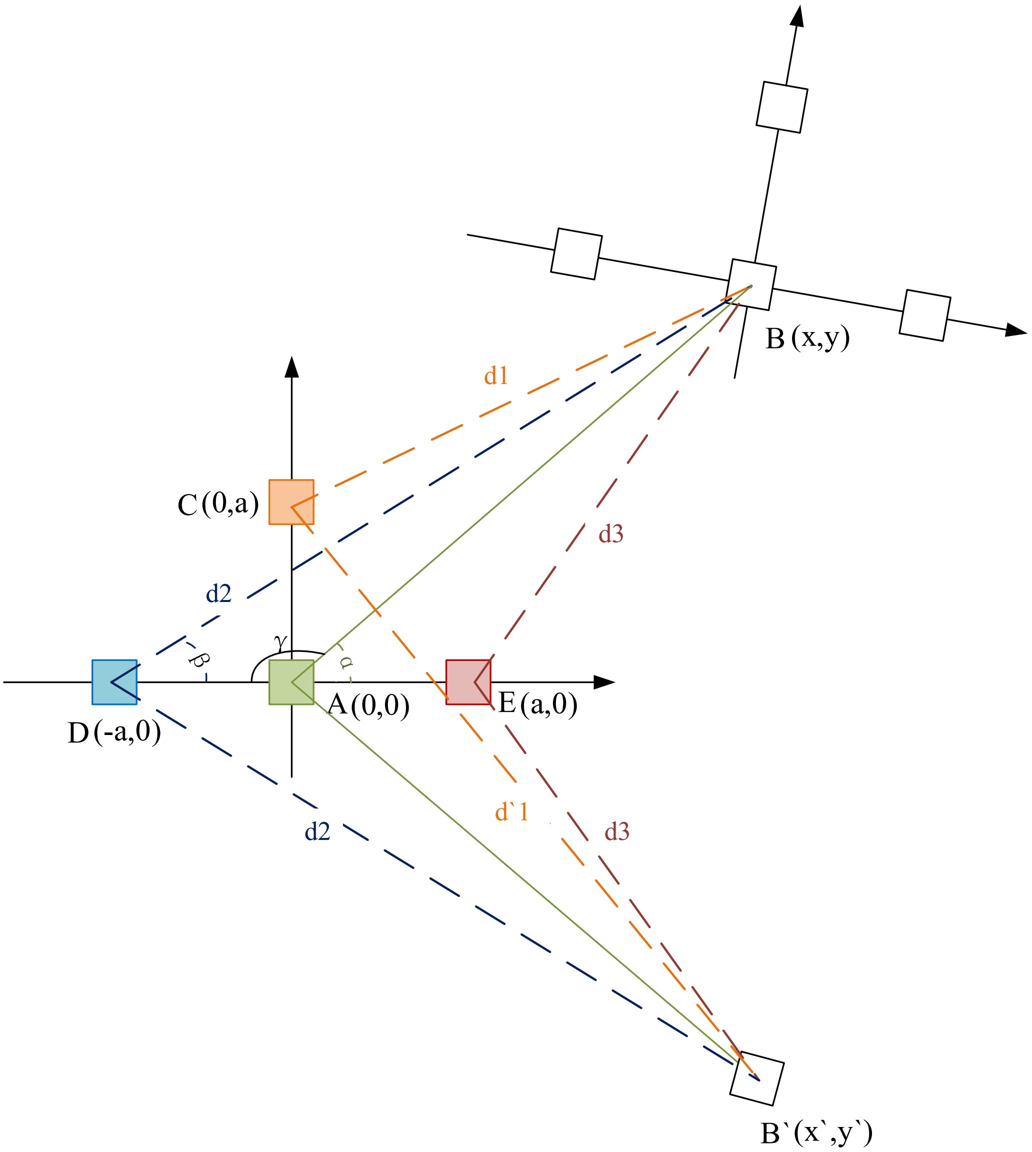


Figure 2.9: Relative positioning algorithm design

$$\begin{cases} {d_3}^2 = {d_2}^2 + (2a)^2 - 2d_2(2a)cos\beta \\ |AB|^2 = {d_2}^2 + (a)^2 - 2d_2acos\beta \\ {d_2}^2 = |AB|^2 + (a)^2 - 2a\,|AB|\,cos\gamma \end{cases} \tag{2.7}$$

Then, calculate the angle α between AB and the positive semi-axis of x through $\alpha = \pi - \gamma$. Now, we have obtained the polar coordinates of robot B $(\pm\alpha, |AB|)$. Due to the symmetry of the coordinate system, the coordinates of B have two possibilities, up and down. Two cases (x, y) and $(x, -y)$ can be obtained through the conversion of polar coordinates and rectangular coordinates (Equ. 2.8). Finally, we need to use d_1 to determine the specific location of point B.

$$\begin{cases} x = |AB| * cos\alpha \\ y = |AB| * sin\alpha \end{cases} \tag{2.8}$$

If $(x-0)^2 + (y-a)^2 = {d_1}^2$ holds, then the coordinates of robot B are (x, y). Otherwise, the coordinates of robot B are $(x, -y)$.

2.5 Summary

This chapter mainly introduces related positioning methods and positioning system design. Firstly, the principle of IR-UWB ranging and positioning is elucidated. Then we investigate and evaluate three popular positioning methods TOF, TDOA, and AOA with IR-UWB sensors. The design of the relatively calculated system is expounded, including positioning model design, hardware design, and positioning algorithm design. We adopt the BP-TWR-50 IR-UWB positioning module as the hardware of the relative positioning system and design a positioning algorithm based on TOF.

3 Positioning System Implementation and Experiments

3.1 Overview

This chapter first introduces the one-to-one ranging experiment to illuminate the characteristics of IR-UWB modules. Then the realization of the relative positioning system is shown in detail. To examine the accuracy of the IR-UWB relative positioning system, we tested eight experiments of position situations and collected the positioning results. Finally, we compare the proposed localization method with existing localization systems (WiFi-, Bluetooth-, Lidar-based) and analyze the advantages and disadvantages of these methods.

3.2 Implementation of IR-UWB Ranging Module

3.2.1 Ranging Algorithm Implementation

The ranging algorithm of IR-UWB modules adopts the DS-TWR algorithm, that is, the three-message ranging method. The specific message transmission between devices is shown in Figure 3.1.

The Tag module will first send an IR-UWB $Poll$ signal and record the sending timestamp

T_1. If the Anchor module receives it, the Anchor will first record the received timestamp T_2, and then return a $Resp$ signal to the Tag after T_{reply1}. At the same time, the sending message timestamp T_3 is recorded. Then, when the Tag receives the response signal, it will record the timestamp T_4, and return a $Final$ signal to the Anchor at the time T_5 after T_{reply2}. The Anchor will record the current timestamp T_6 when it receives it. It is worth noting that each time the Tag module sends information to the Anchor module, it will append the recorded timestamp information to the sent information. In this way, when the last $Final$ message is received, Anchor can directly read the values of T_1 to T_6 from the buffer. Finally, the accurate distance between the Tag and the Anchor can be calculated by the formula (3.1).

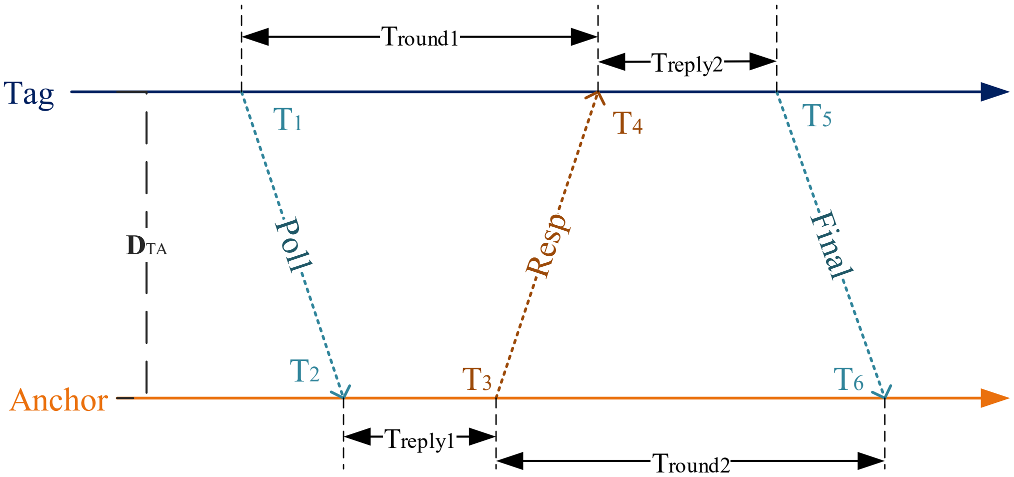


Figure 3.1: Message transmission between IR-UWB Modules with DS-TWR

$$\begin{aligned} T_{prop} &= \frac{(T_4 - T_1) * (T_6 - T_3) - (T_3 - T_2) * (T_5 - T_4)}{(T_4 - T_1) + (T_6 - T_3) + -(T_3 - T_2) + (T_5 - T_4)} \\ &= \frac{(T_4 - T_1) * (T_6 - T_3) - (T_3 - T_2) * (T_5 - T_4)}{T_6 + T_5 - T_1 - T_2} \\ D_{TA} &= T_{prop} * c \quad (c = 3 \times 10^8 \ m/s) \end{aligned} \tag{3.1}$$

where T_{prop} is the propagation time of information from the Tag module to the Anchor module. Therefore, the distance between the two modules (D_{TA}) can be calculated by multiplying the propagation time (T_{prop}) by the propagation speed (c) of the electromagnetic wave in the air.

3.2.2 Ranging Module Experiment

Experimental Setup

Fig. 3.2 shows the experimental setup for the ranging accuracy test. Since the final distance is calculated on the MCU of the Anchor module, we connect the IR-UWB Anchor module to the computer serial port through the USB to the TTL module and the IR-UWB Tag module to the power bank for easy movement. We place the Anchor module at the origin, use the ruler as the standard distance as a reference, and change the distance between the two modules by moving the Tag module.

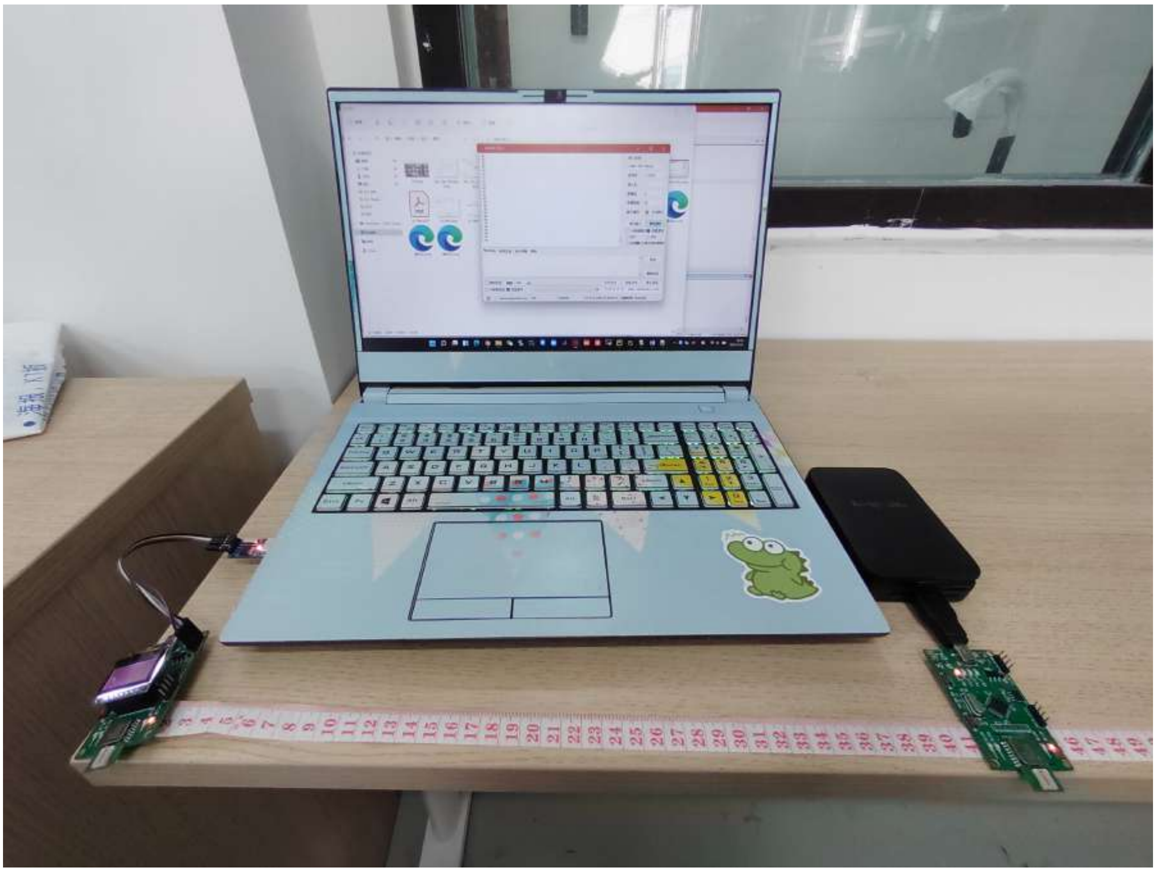


Figure 3.2: Experimental setup of ranging accuracy test

Accuracy of Ranging Modules

We collected 13 sets of test data from 0.5m to 18m, respectively. To increase the confidence of the data, we recorded 540 data points at each distance point and then took the average as our measurement data. The absolute error, relative error, and measurement accuracy are also recorded. The experimental result is shown in Table 3.1.

Table 3.1: Accuracy of ranging modules

Real Distance(m)	**Measurement Points**	**Average Measured distances(m)**	**Absolute Error(m)**	**Relative Error**	**Precision**
0.5	540	0.498	0.002	0.42%	99.58%
1.0	540	0.999	0.001	0.13%	99.87%
1.5	540	1.515	0.015	1.02%	98.98%
2.0	540	1.963	0.037	1.84%	98.16%
2.5	540	2.529	0.029	1.15%	98.85%
3.0	540	2.938	0.062	2.08%	97.92%
3.5	540	3.470	0.030	0.85%	99.15%
4.0	540	3.988	0.012	0.30%	99.70%
4.5	540	4.491	0.009	0.19%	99.81%
5.0	540	4.909	0.091	1.82%	98.18%
6.0	540	6.019	0.019	0.31%	99.69%
10	540	9.992	0.008	0.08%	99.92%
15	540	14.848	0.152	1.02%	98.98%
18	540	17.958	0.042	0.24%	99.76%

The results show that the ranging accuracy of the IR-UWB module can reach about 99.18%, indicating that the ranging results of IR-UWB are very reliable in the case of a large sample base. In order to better show the volatility of the data and the deviation from the standard quantity, we use python's matplot library to plot all the measurement results. As shown in Figure 3.3.

As can be seen from the picture, when the actual distance is less than 5m, the data fluctuates greatly, but when the actual distance reaches more than 10m, the overall trend of the data tends to be flat. According to the table, the test accuracy rate above 5m exceeds 99.5%.

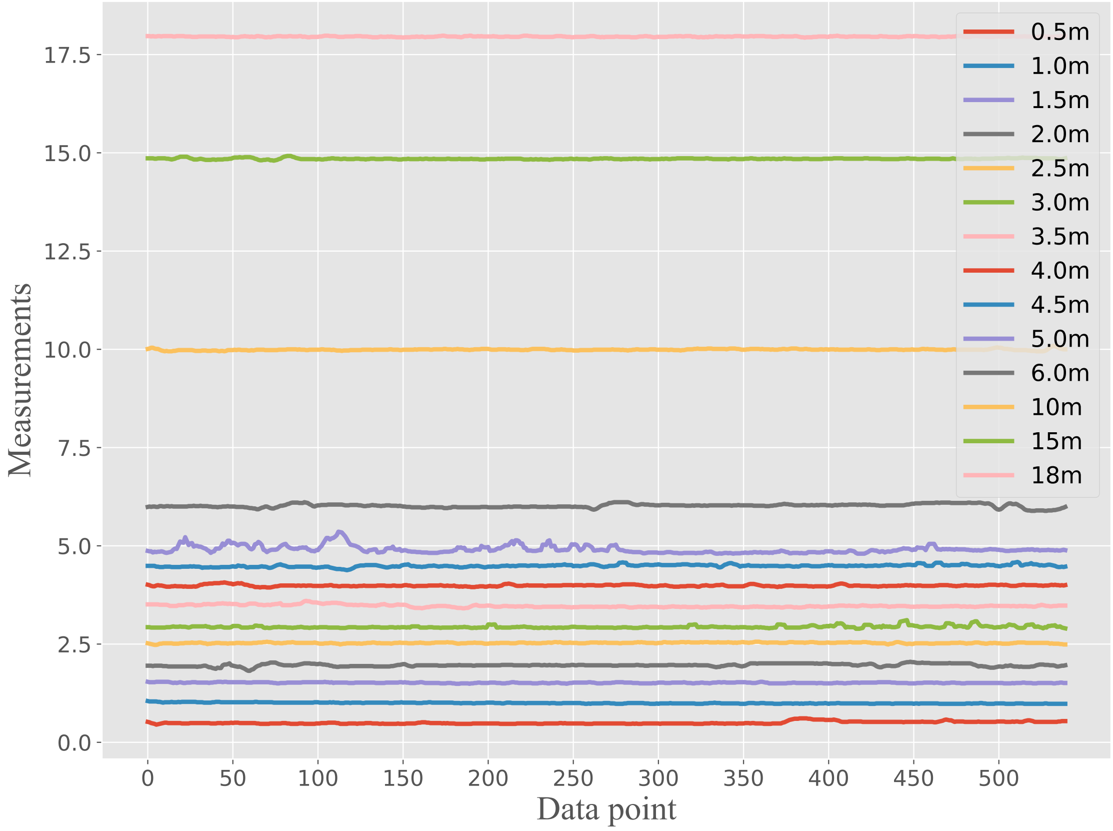


Figure 3.3: Measurement data results

3.3 Implementation of IR-UWB Relative Positioning System

In the relative positioning system, we carried four IR-UWB modules (one transmitting module and three receiving modules) for each tested individual. Each transmitter module will send IR-UWB signals to other individual receiver modules, and also through the DS-TWR ranging technology, the transmitter module will obtain the distance to the three-receiver modules respectively. Then through serial communication, the three receiving distances are transmitted to the upper computer (pc), and then the relative positions of other robots are calculated by the positioning algorithm we proposed, and finally displayed through python visualization.

3.3.1 Hardware Implementation

The actual placement is shown in Figure 3.4. To simplify the calculation of the positioning algorithm, the receiving module is placed at the origin, and the three transmitting modules are placed in the left, up, and down directions at a distance of $15cm$ from the origin, respectively.

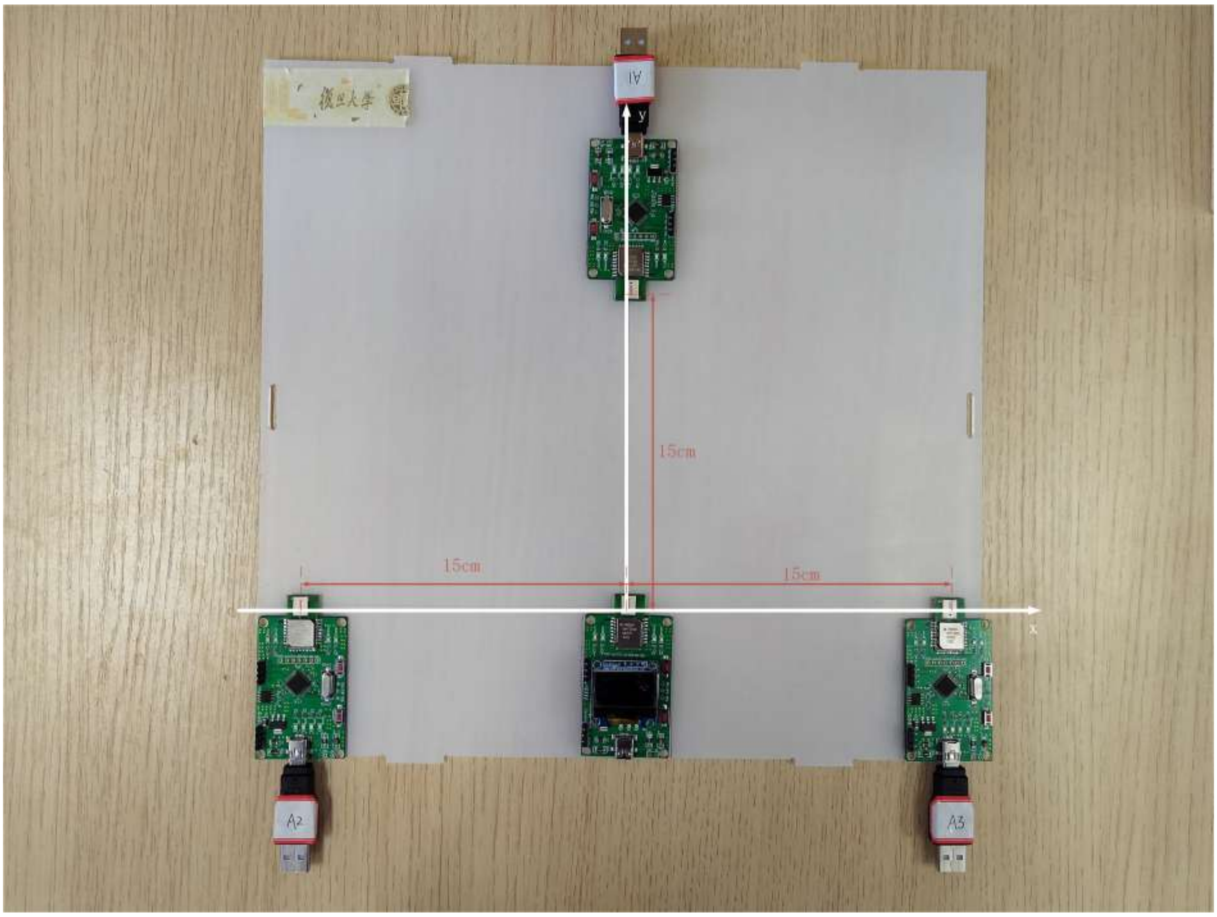


Figure 3.4: Original hardware system design of relative positioning system

However, the experiment found that the crosstalk of the power supply will affect the ranging results, the power supply module and the UWB module cannot be placed on the same level. Therefore, we thought of placing the UWB ranging module vertically and horizontally, as shown in Figure 3.5.

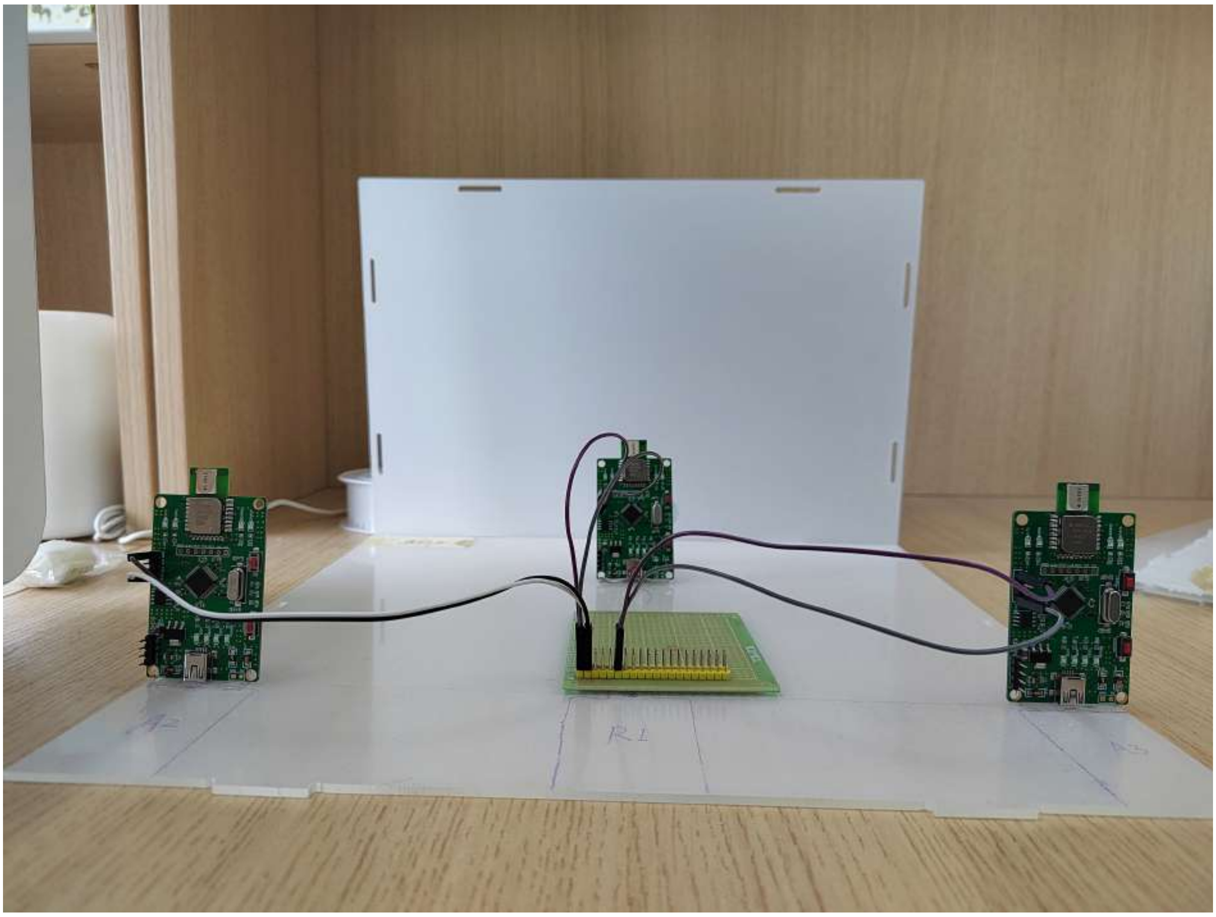


Figure 3.5: Improved hardware system design of relative positioning system

Then, connect the three Anchor modules to the serial port of the host computer, and a label module power supply is powered by one of the Anchor power supplies. Finally, the actual hardware architecture of the relative positioning system is shown in Figure 3.6.

3.3.2 Positioning Algorithm Implementation

As shown in Figure 2.9 in Chapter 2, we design a localization algorithm to calculate the relative positions of other individuals. Algorithm 1 is programmed in a Python language environment with Pycharm software. Read the information of the three serial ports of the computer in turn by installing the serial port library of python.

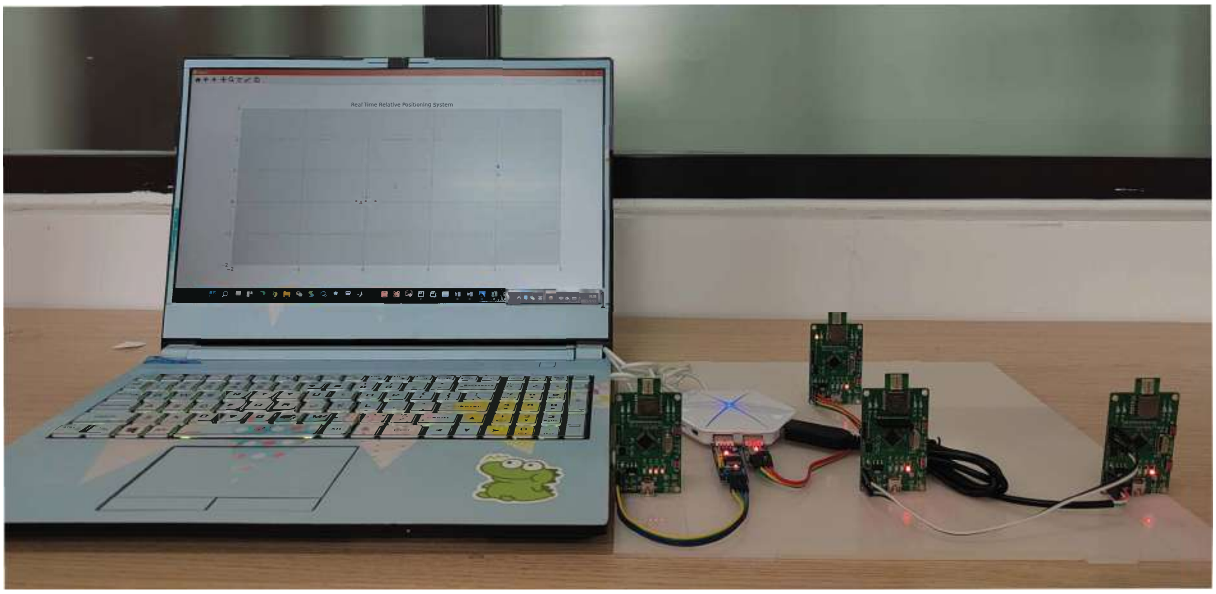


Figure 3.6: Actual hardware architecture of relative positioning system

Algoritmi 1 The Relative Position Algorithm

```
Require: None;
Ensure: Position coordinates;
open three serial ports: $serial.Serial("COM1", 115200, timeout = 5)$;
                         $serial.Serial("COM2", 115200, timeout = 5)$;
                         $serial.Serial("COM3", 115200, timeout = 5)$;
while True do
    get serial buffer data: $ser.inWaiting()$;
    if $count$ != 0 then
        read three serial data: $ser.read(ser.in\_waiting).decode("gbk")$
        get three distance values: $d_1$, $d_2$, $d_3$
    end if
    compute $cos\beta = ({d_2}^2 + 2*a^2 - {d_3}^2)/(2*d_2*2*a)$
    compute $dis_{ab} = \sqrt{{d_2}^2 + a^2 - 2*d_2*a*cos\beta}$
    compute $cos\gamma = ({dis_{ab}}^2) + 2*a^2 - {d_2}^2)/(2*dis_{ab}*a)$
    compute $\alpha = \pi - acos(cos\gamma)$
    compute $x = dis_{ab}*cos(\alpha),\ y = dis_{ab}*sin(\alpha)$
    if $x^2 + (y-a)^2 == {d_1}^2$ then
        return $(x, y)$
    else
        return $(x, -y)$
    end if
end while
```

To improve the accuracy of positioning, we measure the distance multiple times to obtain the average value by setting the serial port reading time. At the same time, to ensure the real-time performance of the algorithm, we only take the average of 10 data as the ranging result. The obtained three distance information are respectively defined as d_1, d_2 and d_3. Then calculate the distance between system A and system B dis_{ab} and the angle α between the two and the x-axis through the knowledge of trigonometric functions. We first calculated two possible cases (x, y) and $(x, -y)$ of point B by the two distances d_2 and d_3. Assuming that one of the cases (x, y) holds, calculate the distance between the target and Anchor 0, and then compare the result with the size of d_1 to determine whether it holds. If the assumption is true, the coordinates are (x, y), otherwise, it is $(x, -y)$.

3.4 Experiment of IR-UWB Relative Positioning System

3.4.1 Experimental Setup

We employed a set of BP-TWR-50 UWB modules to simulate individuals in a relative positioning system. The movement of the individual in the actual situation is simulated by moving the whole set of modules, and then the relative positions of other individuals are obtained through the visual interface of the host computer. The experimental setup is shown in Figure 3.7. Three Anchor modules are connected to the host computer through a serial port, and one Tag is powered through the power port of the Anchor module. The three anchor modules on system A will continuously send IR-UWB signals, and both tags T_a and T_b will receive the signals, but before replying, the Tag module will first make an ID judgment. In this way, the information received by the Anchor module on system A can only come from T_b, and what is obtained is the distance information d_1, d_2, and d_3 between the three modules A_1, A_2, and A_3. Then, through the positioning algorithm we proposed, the position of system B relative to system A is calculated and displayed in the coordinate system with A as the origin.

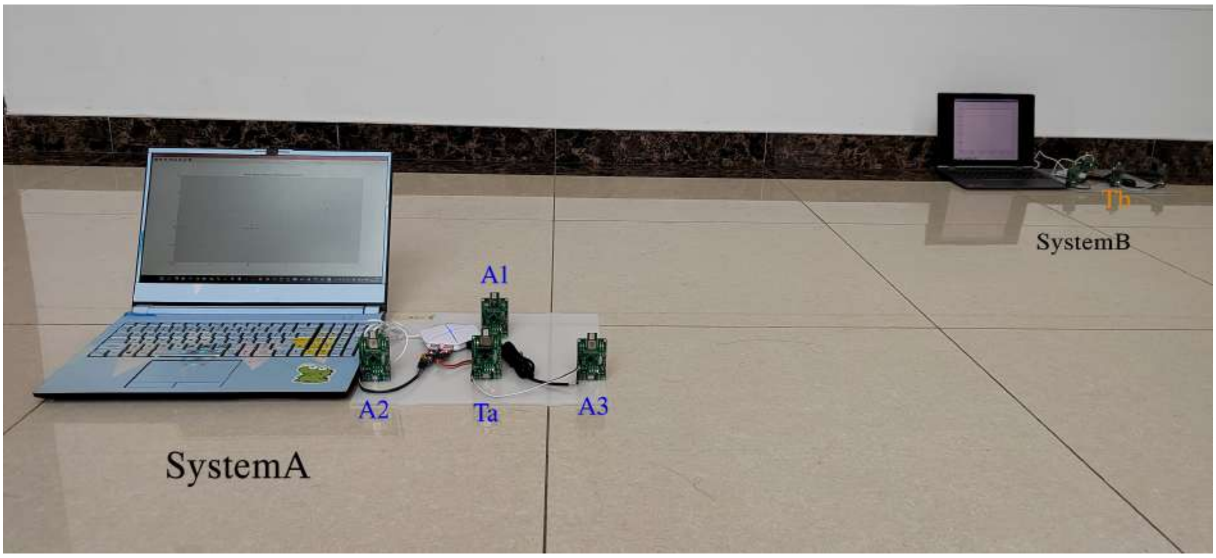
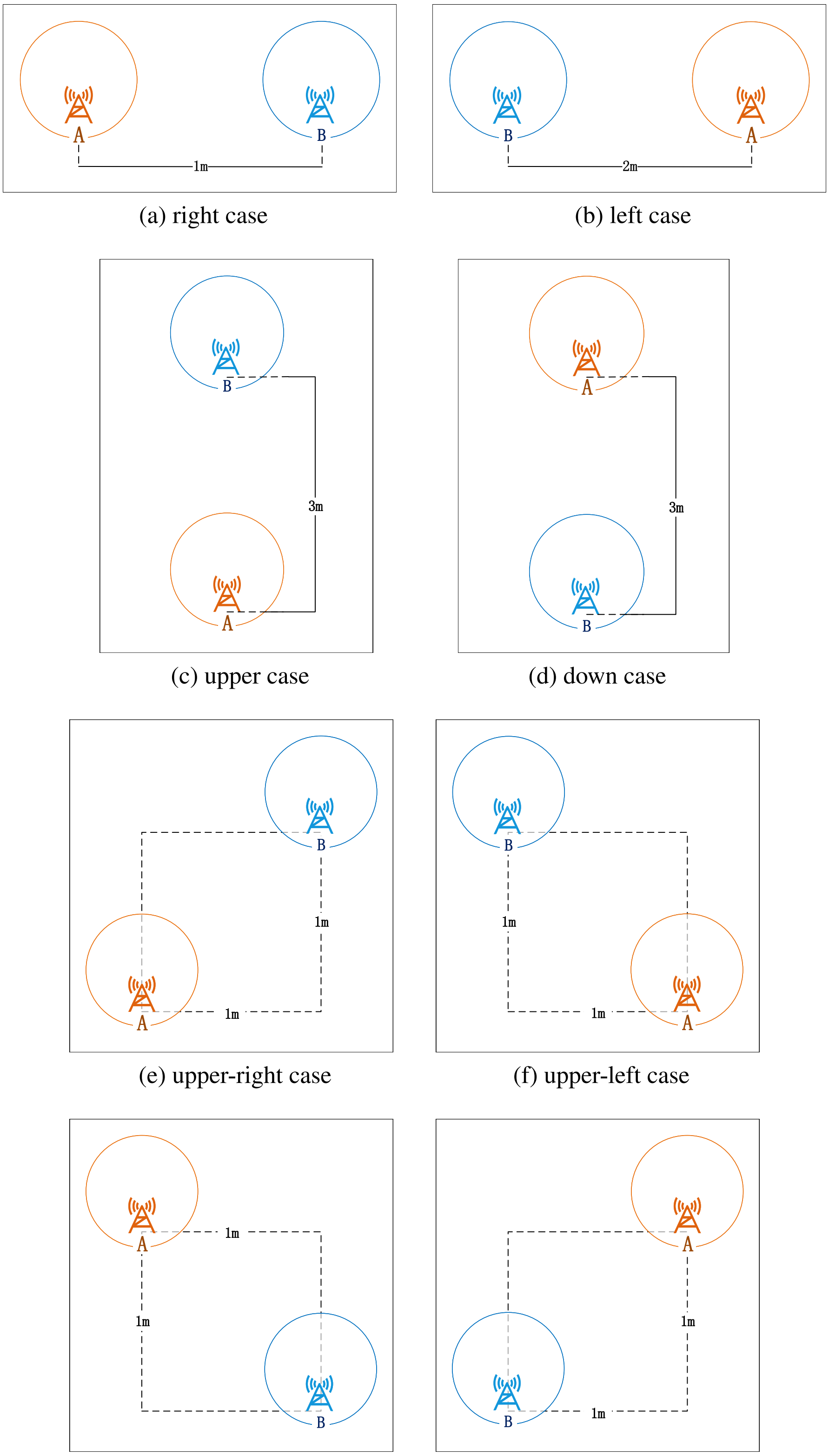


Figure 3.7: Experimental setup of IR-UWB relative positioning system

To verify the reliability of the algorithm, we set up 8 location situations for testing. As shown in Figure 3.8, system B is directly left, right, directly above, just below, upper-right, upper-left, down-right, and down-left of system A, respectively.

3.4.2 GUI Display of Positioning Results

We created a Cartesian coordinate system image with Python's matplotlib library and displayed the generated coordinate points in real time. Since our experiment uses System A as the main system, the positioning results will be displayed in the main system of System A. As shown in Figure 3.9, we set the Tag module of system A as the origin of the coordinates. According to the specific placement of the module, the anchor module is identified at three positions $(0, 0.15)$ $(-0.15, 0)$ and $(0.15, 0)$. We obtained three distance information from the three serial ports, and then calculated the position (x, y) of the label T_b in system B through the positioning algorithm, and displayed it on the canvas. Since the serial port has been transmitting data, the position of system B is also constantly updated. To achieve the effect of real-time display, every time we calculate position information, we will refresh the canvas once, and then display the new position information again.



(a) right case

(b) left case

(c) upper case

(d) down case

(e) upper-right case

(f) upper-left case

(g) down-right case

(h) down-left case

Figure 3.8: Eight positional situations for system A and system B

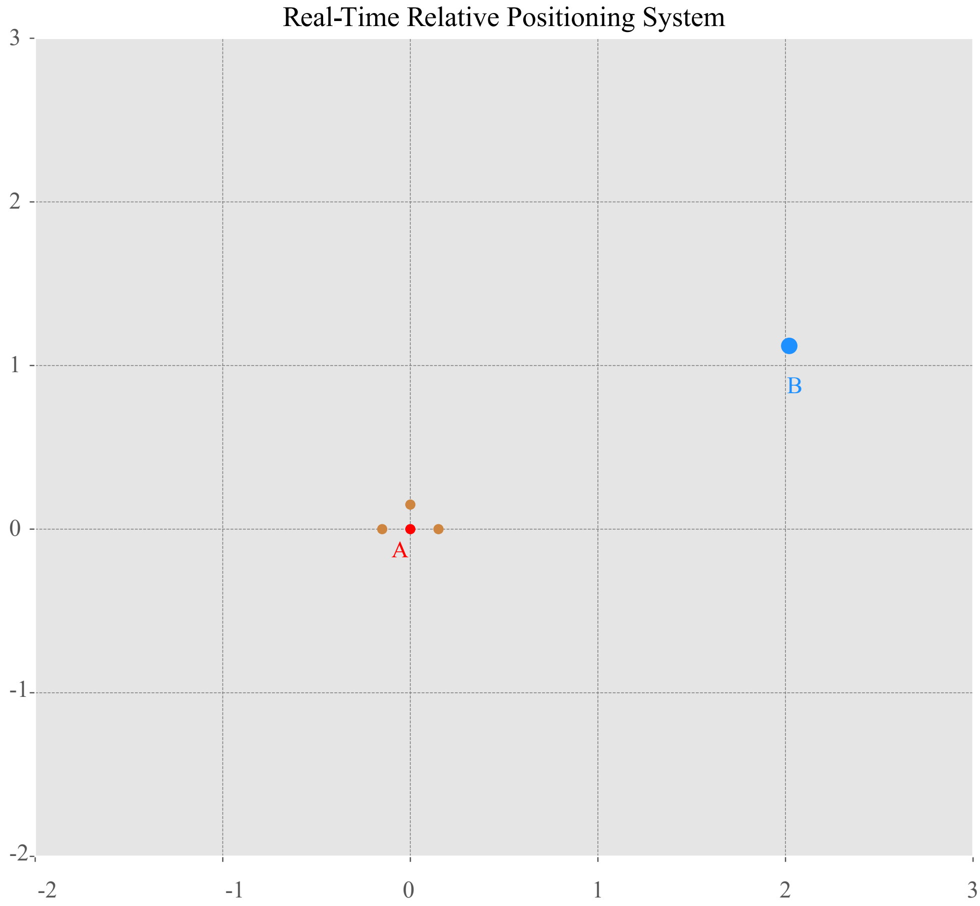


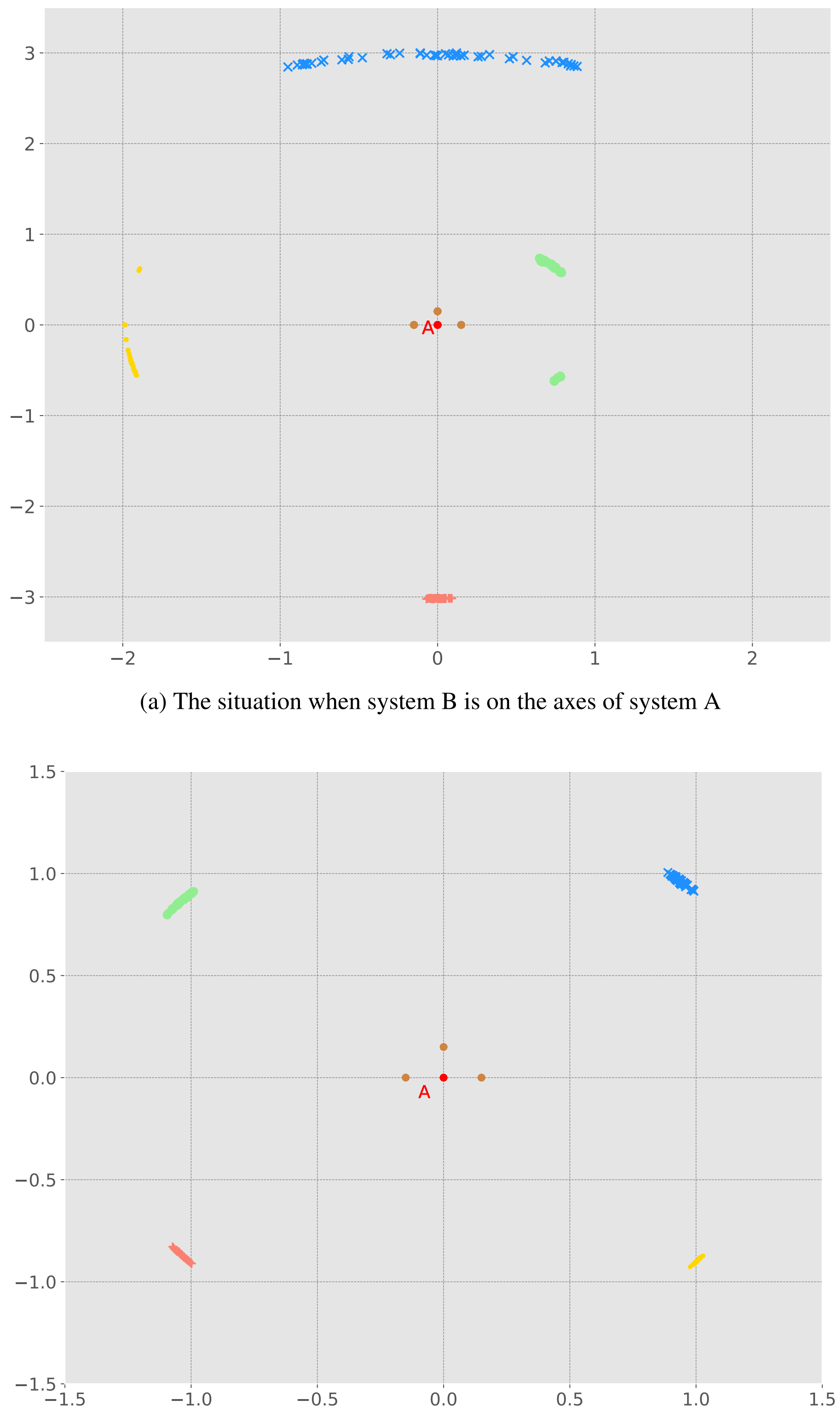
Figure 3.9: Real-time GUI display of positioning results

3.4.3 Accuracy of the Relative Positioning Algorithm

We collected 500 data points respectively according to the eight positions of the experimental setup. The results in the form of scatter plots are shown in Figure 3.10.

The positioning error is defined as:

$$Pos_{err} = \sqrt{(x - x_0)^2 + (y - y_0)^2} \tag{3.2}$$



(a) The situation when system B is on the axes of system A

(b) The situation when system B is in the four quadrants of system A

Figure 3.10: Measurement coordinate results

where (x_0, y_0) is the actual position of the measured target, and (x, y) is the result from the calculation.

So, the accuracy of the positioning system can be calculated as:

$$Acc = 1 - \delta = 1 - \frac{Pos_{err}}{Pos_{real}} = 1 - \sqrt{\frac{(x - x_0)^2 + (y - y_0)^2}{x^2 + y^2}} \times 100\% \tag{3.3}$$

where δ represents the relative error of the coordinate results, Pos_{real} means the actual coordinate position.

The formula for calculating Root Mean Square Error (RMSE) is:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}[(x - x_0)^2 + (y - y_0)^2]} \tag{3.4}$$

Through the calculated data of 50 coordinate points, we obtain an average coordinate result and calculate the positioning error through the formula (3.2). Finally, the positioning accuracy and RMSE of the algorithm are obtained by formula (3.3) and (3.4). The detailed results are shown in Table 3.2.

Table 3.2: Accuracy of the relative positioning system

Real Position (x_0, y_0)	**Measurement Points**	**Average Measured coordinate**	**Position Error**	**Precision**	**RMSE**
(1,0)	50	(0.714,0.662)	0.721	27.94%	0.721
(-2,0)	50	(-1.961,0.324)	0.327	83.67%	0.356
(0,3)	50	(0.009,2.989)	0.014	99.52%	0.583
(0,-3)	50	(0.012,-3.018)	0.021	99.29%	0.043
(1,1)	50	(0.940,0.961)	0.072	94.90%	0.079
(-1,1)	50	(-1.029,0.874)	0.129	90.86%	0.136
(1,-1)	50	(1.002,-0.900)	0.100	92.91%	0.101
(-1,-1)	50	(-1.036,-0.870)	0.135	90.46%	0.138

The relative error decreases greatly with the increase of the actual distance. When the relative distance reaches about 3m, the accuracy can reach about 99%. When the relative distance is 1m, the accuracy is less than 28%, so in the applicable scenario: large-scale smart factories, the distance between robots is more than 3m; when the absolute distance is similar, the accuracy on the coordinate axis is not as high as in the four quadrants. There may be an overlap between the IR-UWB modules, and crosstalk occurs between the IR-UWB modules, which affects the ranging results. Moreover, the accuracy of the four quadrants does not require high relative distances. Even when the relative distance is about 1.5m, the accuracy can reach more than 90%.

3.5 Improvement of IR-UWB Relative Positioning System

During the implementation of relative positioning, we found that since each Anchor module needs to communicate with the host computer, it needs to consume a lot of USB to TTL modules, and the USB interface of the host computer sometimes cannot meet the requirements. Therefore, we consider passing the distance information to the Tag module, so that only one USB serial port is needed to obtain three distance information at the same time. But in the implementation process, we must first solve the problem of module identification and distinction. Since each individual carries only one IR-UWB transmitter module and three IR-UWB receiver modules, how to make the transmitter module distinguish the three-receiver modules is the challenge we face.

Therefore, we thought of the method of adding an ID to the receiving module to distinguish, and then in the transmitting module, the ID is used to determine which receiving module the signal is returned from. In the code, we use Anchor to represent the receiving module, Tag represents the transmitting module, and $ANCHOR\_IND$ is used to distinguish the three receiving module numbers, which are 0, 1, and 2 in sequence.

The specific implementation steps are shown in Algorithm 2. We first define the IDs of Tag and Anchor and the number of Anchors outside the main function. Then we created two new arrays, $Anthordistance[]$ to store the distance information of the three Anchors and labels, and $Anthordistance\_count[]$ to count the number of Anchors. Each bit of the array represents an Anchor. In the main function, we need to initialize the peripherals first, including GPIO initialization, SPI peripheral initialization, OLED initialization, and USART initialization. Then it is the one-to-many ranging between Tag and Anchor. The ranging algorithm still uses the DS-TWR method. The difference is that the Tag will determine the ID of the base station once, and then transmit the UWB message in turn. When the communication between Tag and this Anchor is successful, then the value on $Anthordistance\_coun[Anchor\_IND]$ will be increased by one, and finally, the calculated result will be stored in $Anthordistance[Anchor\_IND]$.

Algorithm 2 is compiled in the C language environment, and the code is compiled using the keil5 software. After the compilation is successful, it is burned into the MCU of the IR-UWB module through ST-Link. By changing the $ANCHOR\_IND$ parameter from 0 to 2, load the program into Anchor0, Anchor1, and Anchor2 respectively. Finally, the distance calculation and ID determination algorithm will be implemented in the Tag module, and the obtained three distances d_1, d_2, and d_3 will be sent to the serial port.

Finally, we can obtain the distance information between the Tag module of system A and the three anchor modules of system B. Through our proposed positioning algorithm, the relative position of system A and system B can be obtained. Then, at this time our coordinate origin is the Tag module of system B. Therefore, if we want to obtain the relative position of system B, we also need to do one step of coordinate transformation. For example, assuming that the positive directions of system A and system B are the same, the position of system A relative to B is (a, b), then the position of system B relative to A becomes $(-a, -b)$.

Algoritmi 2 Receiving Module ID Discrimination Algorithm

```
**Require:** None;
**Ensure:** Distance between transmitter module and three receiver modules;
define Tag ID: $\#define\ TAG\_ID\ 0x0F$;
define the number of Anchors: $\#define\ ANCHOR\_MAX\_NUM\ 3$;
define Anchor ID: $\#define\ ANCHOR\_IND\ 0$; // 0 1 2
define distance array: $int\ Anthordistance[]$;
define distance count array: $int\ Anthordistance\_count[]$;
while True do
    Peripheral initialization: $peripherals\_init()$;
    calculate distance: $DS - TWR\ algorithm$
    if $Anthordistance\_count[0]\ >\ 0$ then
        get the distance between Tag and Anchor0:
        $sprintf(dist\_str, ”d1 : \%3.2f”, (float)Anthordistance[0]/1000)$;
        $printf(”\%s”, dist\_str)$;
        $printf(”/r/n”)$;
    end if
    if $Anthordistance\_count[1]\ >\ 0$ then
        get the distance between Tag and Anchor1:
        $sprintf(dist\_str, ”d2 : \%3.2f”, (float)Anthordistance[1]/1000)$;
        $printf(”\%s”, dist\_str)$;
        $printf(”/r/n”)$;
    end if
    if $Anthordistance\_count[2]\ >\ 0$ then
        get the distance between Tag and Anchor2:
        $sprintf(dist\_str, ”d3 : \%3.2f”, (float)Anthordistance[2]/1000)$;
        $printf(”\%s”, dist\_str)$;
        $printf(”/r/n”)$;
    end if
end while
```

3.6 Evaluation of Positioning Systems

We compare the proposed relative positioning system with other similar works and the results are presented in Table 3.3. According to the adopted positioning module, we analyze the positioning systems based on six technologies: WiFi, BLE, Lidar, Camera, and UWB.

Interestingly, we found that WiFi and BLE have the widest coverage but the lowest accuracy. Although the work of Lin *et al.* [32] and Subhan *et al.*[33] has reduced the positioning errors of WiFi and BLE to 1.4m and 1.325m, the positioning errors of both may reach more than 2m

Table 3.3: Comparison with other positioning methods

Study and Year	**Hardware**	**Anchors**	**Ranging**	**Algorithm**	**Position error or Accuracy**
Lin *et al.*[32](2016)	WiFi	6	$50 \times 20m$	ISWF, AoA	1.4 m
Subhan *et al.*[33](2019)	BLE	4	$10 \times 10m$	RSSI	1.325m
Ramezani *et al.* [34] (2020)	Lidar	1	100m,250m	Verified Lidar-SLAM	0.09m
				Non-Verified Lidar-SLAM	0.64m
J. Lin *et al.*[35](2020)	Camera	2	$0.2 - 0.5m$	Binocular stereo vision	95%
Che *et al.*[36](2020)	UWB	4	$32m^2$	the NB ML algorithm	87%
Relative Positioning system	UWB	4	>18m	TOF	93.09%

under the circumstances. Y. Yun *et al.*[37] collected both Wi-Fi and BLE RSS data on the same smartphone to perform a detailed comparison of the two positioning methods. According to the experimental results at 16m×7m indoors, the system based on WiFi and BLE technology obtained indoor positioning errors of 2.39m and 2.52m, and outdoor positioning errors of 4.18m and 4.07m, respectively. Lidar-based positioning technology has been a hotspot in recent years. It scans the environment to obtain point cloud data, establishes an environment model, and then locates it through a matching algorithm. Ramezani *et al.* [34] designed a 3D factor-graph LiDAR-SLAM system and achieved 0.09m relative pose error with verification, 0.64m without verification. So as the camera-based method, the work of Lin *et al.*[35] obtained the proportional error within 5%. However, both Lidar-based and vision-based methods have one of the biggest drawbacks: lighting. What's worse, the power consumption of image processing is also a large overhead. Fortunately, the emergence of UWB technology perfectly solves the privacy and lighting impact problems caused by cameras. The work of Q. V. Brande *et al.*[6], N. Macoir *et al.*[7], and Mogale *et al.*[9] demonstrate the superiority of IR-UWB technology in indoor positioning. The use of IR-UWB technology enables the positioning system not only to reduce multipath interference but also to achieve high accuracy. Che *et al.* developed a naive Bayes (NB) machine learning (ML) algorithm to improve the localization accuracy of IR-UWB technology [36]. Good classification results also illustrate the potential of IR-UWB technology in ML, providing a theoretical basis for future AI-related applications.

3.7 Summary

This chapter introduces the specific implementation method of the relative positioning system, including ranging function, hardware construction, and software algorithm implementation. Through experiments, we calculated that the ranging accuracy of IR-UWB is about 99.18%. However, the positioning algorithm is limited by direction and distance. Except for one extreme case (when the distance with a target is less than 2m), the average positioning accuracy of rest cases has reached 93.09%. Moreover, we have also improved the original relative positioning system, unifying three serial ports into one serial port, thus simplifying the hardware device. Finally, the comparison with other positioning methods shows the superiority of our proposed system, such as high accuracy, scalability, and flexibility. It also provides a thought for future IIoT positioning applications for robot arrays or flexible scheduling of staff groups.

4 Sensing System Design

4.1 Overview

Exploring novel sensing technologies to promote new interaction methods is still an active research topic in human-computer interaction[38]. As a vital part of the non-verbal form of communication, hand gesture recognition provides a user-friendly interface between humans and computers[39]. In the IIoT application scenario, especially in smart factories, HGR has emerged as a popular solution for human-machine interfaces (HMI)[40]–[43], enabling people to communicate with machines and interact naturally. Hand gesture recognition has emerged in recent years as a robust sensing method in non-contact HMI[44].

In this chapter, we first introduce the principle of sensing system design. We compare the IR-UWB sensor with other sensing methods, illustrating its feasibility in IIoT applications. Then we find a suitable approach for IR-UWB signals via the analysis of commonly used signal pre-processing and classification methods. Last but not least, we show the overall architecture of the sensing system, including the IR-UWB contactless sensor, hand gesture dataset, the pre-processing method design, and the classification method design.

4.2 System Design Principle

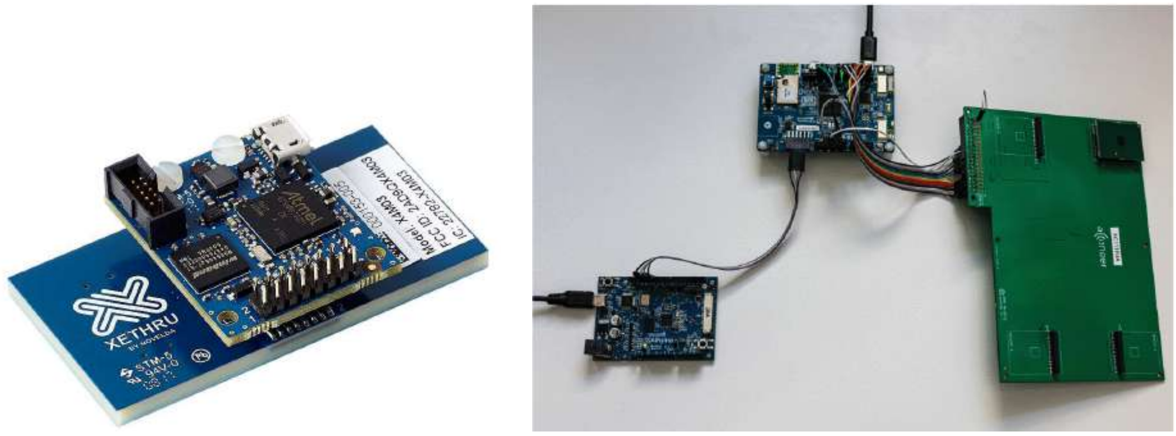
4.2.1 Sensor Design Principle

In the past decade years, various HGR methods have been proposed, which can divide into two categories according to the data collection method: a) contact sensors[45]–[47], b) contactless sensors[48]–[50]. Among the various contact sensor-based methods, Bahador *et al.* collected hand gesture information through an Electromyography signal and reached an accuracy of 99.99% with 96 electrodes [45]. But the experiment needs to place electrodes on the arm and keep the electrode plate in the same direction as the muscle. So are wearable devices in Yuan *et al.* [46] and Bedico *et al.* [47], hard to adapt to differences in user physique and behavior and troublesome to get signals we need. As for non-contact sensors, the camera-based method is the most widely used in HGR. Lin *et al.* proposed an efficient and compact image-based pose representation method[48]. It's called the *Poseimage Pyramid* because of encodes the spatial and temporal information of human posture or hand posture into an image pyramid. They get the hand joints from a video, then calculate the pairwise Euclidean distance and geodesic distance between the joints to build a Pose image clip. Event-based sensors-DVS Camera also receives more attention from researchers because of the extremely low power consumption. Massa *et al.* design a DNN-to-SNN model for the DvsGesture dataset and achieves 89.64% classification accuracy on 11 hand gestures[49]. The DvsGesture dataset they used for gesture recognition is fully event-based from IBM. Each gesture is recorded with a DVS128 camera, providing 1342 samples divided into 122 trials.

However, camera-based approaches are sensitive to the environment, such as contrast, exposure, and brightness [51]. People also usually feel uncomfortable when viewed by the camera. Furthermore, the image sensors have relatively high power consumption. Even the power consumption of Kinect[50] sensors has reached several hundred watts in actual applications[52]. In addition to the camera-based recognition system, the radar sensor is a promising sensing technology for HGR, as not affected by lighting conditions or any related privacy issues[53]. IR-UWB non-contact sensor can generate a typical spikes signal with non-sinusoidal narrow pulses ranging from nanoseconds to microseconds. IR-UWB sensor has lately received great attention for gesture recognition applications due to its high temporal resolution, low risk of exposure to the human body, lower power consumption, and robustness to multipath[3], [4].

Ahmed *et al.* [12] collected eight hand gestures data with two Xethru-X4 Radars and stored them as a two-dimensional data matrix. They used a simple loop-back filter for signal preprocessing thanks to its simple structure and low computation expense[4], [54]. However, they converted the radar data matrix into a 3D image to train a 3D CNN, increasing algorithm calculation overhead. Choi *et al.* used IR-UWB radar as a counting sensor for inbound and outbound people [16]. They collected data using two IR-UWB radar sensors equipped with narrow-beam antennas, thereby improving the performance of the algorithm through mutual information between two adjacent radars. The radar chip they use is made by NOVELDA in Norway. Antennas are arrayed to narrow the azimuth angle when transmitting pulsed signals. Finally, through the test in the laboratory and real subway station, the average error rate can be controlled at about 7%, which also shows the potential of IR-UWB radar in people counting. Scherer *et al.* used A1 Radar sensors of Acconeer to collect gesture datasets [13], due to its low power consumption and high resolution. They collected data from five basic gestures with an A1 radar and acquired 11 gesture datasets by increasing the number of radar sensors to two. A model mixing TCN and CNN was proposed to process radar dataset, which achieve an accuracy of 89.52% (11 gestures) and 93.83% (5 gestures) with a surprisingly low power consumption of 21mW.

Therefore, in the sensor design part of the sensing system, we decided to use the IR-UWB radar module to collect gesture information. Therefore, in the sensor design part of the sensing system, we decided to use the IR-UWB radar module to collect gesture information. This approach does not require the user to wear any sensors, and also avoids the privacy concerns of cameras.



(a) Xethru-X4 radar chip, taken from [12] (b) A111 RADAR sensors setup, taken from [13]

Figure 4.1: The IR-UWB contactless sensors

4.2.2 Pre-processing Method Design Principle

With the increasing attention to IR-UWB technology in recent years, more and more methods for IR-UWB signal pre-processing have been proposed. Generally speaking, common processing methods are a) filtering method, b) FFT, and c) sampling.

The filtering method is generally used in the positioning system based on IR-UWB because only one filtering or impurity removal operation is required on the signal to obtain the distance signal we need. Generally, no subsequent identification algorithms need to process the filtered signal. V.-H. Nguyen *et al.* [55] designed a location detection and tracking of moving targets system based on a 2D IR-UWB Radthe ar. Generally speaking, the received radar signal divides into three parts: target signal, clutter, and noise. The authors introduce a Kalman filter (KF) based clutter estimation method to estimate the clutter and subtract it from the received signal. The KF equation for clutter reduction can be greatly simplified by taking the coefficients of the KF formula to 0 or special values such as the identity matrix I.

FFT is a commonly used method in IR-UWB based sensing systems. Due to the maturity and superiority of CNN, most scholars convert the one-dimensional IR-UWB signal into a two-dimensional matrix through Fourier transform related methods and input the simulated picture information into the CNN for identification. Li *et al.* propose an IR-UWB radar-based method for hand gestures recognition using ShuffleNet V2[11]. They collected 7 gesture information through an IR-UWB radar. The radar transmitting antenna will emit electromagnetic waves $S_T(t)$, and the electromagnetic waves will be reflected in the receiving antenna when encountering obstacles (human hands), the received signal is recorded as $S_R(t)$. To convert a 1D waveform signal into an input signal suitable for ShuffleNet, Li et al. transformed $S_R(t)$ into a 2D matrix according to the scanning frequency. Finally, 2D FFT is performed on the 2D matrix signal to obtain more frequency domain information, wider range, and Doppler accuracy, which is also called Doppler processing. Scherer *et al.* [13] adopted the range frequency Doppler map to extract feature maps, which are based on the Fourier transform of the time axis. They first transformed the I/Q signal from the time domain to the frequency domain by Fourier transform to detect changes in the frequency domain. Then, the results obtained from multiple scans of the radar are spliced together, and the Fourier transform is performed on the data at the same distance point. Finally, since each scan signal is discrete in time and distance, a discrete Fourier transform is used to obtain the final feature maps. A 2D CNN will be used to process these feature maps, and then combined with TCN to reduce system power consumption, and finally realize the recognition of 11 gesture data.

As for the sampling method, it is commonly used in power-sensitive systems. In general, downsampling reduces signal complexity and the processing power of the system. Kim *et al.* used the sampling method to process 1D IR-UWB signals[56]. The IR-UWB signal received by the receiving antenna first goes through an amplifier and a low-pass filter for impurity removal and then is digitized by a high-speed oscilloscope with a sampling speed of 20 GHz. Although the whole pre-processing process is not complicated, and the recognition algorithm only uses 1D CNN, the classification result is not very satisfactory, which only reaches 90%. It can be seen that simple down-sampling or over-sampling is not suitable as a pre-processing method for IR-UWB signals.

Recently, level-crossing (LC) sampling has become a novel and popular method in analog-to-digital converter (ADC), called level-crossing ADCs (LC-ADCs). LC-ADCs are wildly used in the medical field, processing Biomedical Signals, such as electrocardiogram (ECG) [57], [58], electroencephalogram[59], [60], and electrocorticography[61] signals. The common point of these signals is that the sparseness is relatively large. If the traditional ADC method is used, more sampling information will be required. If the sampling rate is reduced, information will be missing due to insufficient sampling accuracy. M. Saeed *et al.* implemented an event-driven ECG classification method based on an open-source, LC-ADC non-uniformly sampled dataset[58]. In their work, the compression performance of three LC-ADC designs is presented and the three models are evaluated using the MIT-BIH Arrhythmia dataset[62]. The results show that the LC-ADC can help event-driven ECG signals reduce sampling complexity while achieving comparable classification accuracy compared to uniformly sampled data. S. M. Qaisar and S. F. Hussain obtained electroencephalogram signals by using LC-ADCs, which reduced the number of samples obtained by a factor of 1 and 3.7, respectively, compared to traditional datasets [59]. This also reduces the computational complexity of subsequent signal processing methods by a factor of more than 14, with significant reductions in transmitter power, system bandwidth usage, and classifier computational load. Moreover, the proposed system achieves 100% classification accuracy in most of the study cases, illustrating the superiority of LC-ADCs in bio-signaling.

Since the IR-UWB signal is also an event-driven pulse signal, and the data has a certain sparseness, we also tried to use the LC-ADC method to preprocess the IR-UWB radar echo signal. What's more, the classification method we adopted is SNN, which only accepts neurons in the form of spikes as input. So we sampled the analog signal quantity of radar echo into a digital signal containing only 0/1 sequence.

4.2.3 Classification Method Design Principle

In recent years, hand gesture recognition has become a hot topic. Various papers are published about it. However, the traditional methods for processing hand gesture recognition are high power consumption because these methods need to sample data constantly. Therefore, many models and deep learning algorithms are coming out to solve recognition problems more efficiently with less energy.

Elboushaki *et al.* present an effective multi-dimensional feature learning approach for human gesture recognition based on MultiD-CNN with RGB-D videos[63]. They use convolutional residual networks for training extremely deep models and convolutional long short-term memory networks (ConvLSTMs) for dealing with time-series connections. Long short-term memory networks are a special type of recurrent neural network, which defines a cyclic hidden state, each activation of which depends on the previous time[64]. Elboushaki *et al.* adopt one ConvLSTM layer with four ConvLSTM units. At different time steps, a 2D feature map will be given as their input to capture the temporal dependencies between the RGB-D dataset. It turns out that the design shows efficiency in video gesture classification tasks and achieves state-of-the-art results on several datasets.

CNN is a crucial algorithm in the field of deep learning invented before SNN. Its potential has been excavated by researchers since the 1980s, as TDNN and LeNet5 came out. And one of the best-known utilizations is hand gesture recognition. Ikram and Liu have proposed a dynamic hand gestures model by leap motion controller, based on CNN and long-short term memories [65]. The controller can process the location and velocity in that 3-D space. The model acquired an accuracy of 98% compared with state-of-art HGR methods. The system architecture consists of gesture detection and gesture classification, and these two modules are connected through a post-processing module. Usually, the detection will find a series of gestures, and the processing module will separate similar frames and generate a large number of groups within the frame. Although it has high accuracy, power is not mentioned, which means that power cannot be the advantage of the model. Ahmed *et al.* [12] improved one of the classic models of CNN, GoogLeNet, as an algorithm for IR-UWB radar gesture recognition. They reduced the 9 Inception modules in the original GoogLeNet model to 7 to process 3D intensity images converted from IR-UWB radar signals. The architecture of the system consists of seven basic inception modules, each of which is equipped with three different convolutional layers and a max-pooling layer. Finally, experiments show that this new CNN architecture is more robust and accurate than previous architectures, achieving 95% accuracy on eight gesture datasets. In addition, Li *et al.* [11] and Scherer *et al.* [13] also used CNN-related models as their gesture recognition algorithms, and achieved high accuracy.

However, power consumption remains a significant challenge in IIoT applications. Although the CNN model can achieve very high accuracy, the entire classifier consumes a lot of power due to the complexity of the convolution calculation. Therefore, in recent years, researchers have also been exploring more efficient recognition networks. Singh *et al.* proposed a low-energy algorithm for gesture recognition based on SNN called Gesture-SNN[66]. They used Gradient-Descent to train deep artificial neural networks (ANNs) and converted them to SNNs. What's more, they designed three novel techniques: a) Combining multiple frames; b) Aggregating the first layer membrane potential; c) Partially gating the first layer membrane potential to improve the accuracy of the conversion from ANNs to SNNs. The evaluation showed that the model is 38% more accurate with 35% lower energy and 55% lower EDP compared to its traditional SNN counterpart.

Tsang *et al.* proposed an SNN approach for radar-based hand gesture recognition while using frequency modulated continuous wave millimeter-wave radar[67]. They convert radar Doppler maps into spike trains with a signal-to-spike conversion scheme. The spikes were used as input for different classifiers, including logistic regression, random forest, and support vector machine. Finally, they achieved over 98% accuracy on 10-fold cross-validation for two publicly available reference datasets.

SNNs simulate the coding method of nerve synapses transmitting information via electrical impulses to process pulse sequence information[68]. Therefore, the SNN neuron model is closer to the brain nerve cell than other ANNs, naturally suitable for processing IR-UWB signals.

4.3 System Architecture Design

Fig.4.2 presents the neuromorphic processing system for low-power hand gestures classification based on IR-UWB Radar[44]. The system mainly consists of three parts: gesture data set, pre-processing, and SNN processing. The gesture datasets are periodic samples with range scan vectors recorded by the IR-UWB radar sensor. So, we can regard these samples as a time-series signal with a sweep period and do some time-domain signal processing. The details of each part of the system design are given.

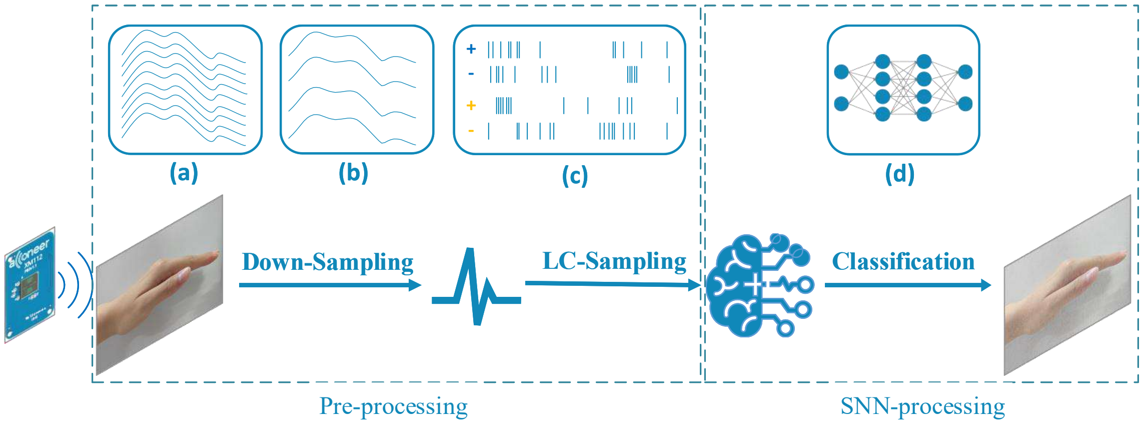


Figure 4.2: Hand gestures recognition system architecture

4.3.1 IR-UWB Contactless Sensor and Dataset

The datasets were collected by Integrated Systems Lab at ETH Zurich using the Acconeer XR111 sensor [13]. These gestures were selected by collectors through a long process, including literature review, design seminars, focus groups, and user evaluations of the Wizard of Oz[69]. The datasets are divided into one dataset with 5 gesture classes and one dataset with 11 gesture classes. Since the data set of 11 gestures is processed data, we cannot obtain the amplitude and phase information of the original signal, so we chose 5 gesture classes as our data set. We renumbered the five-gesture data of *PullUp*, *PushDown*, *SwipeRL*, *FingerSlider*, and *PalmTilt* as a-e, and added the *NoHand* data (f) as our sixth gesture, see Fig.4.3.

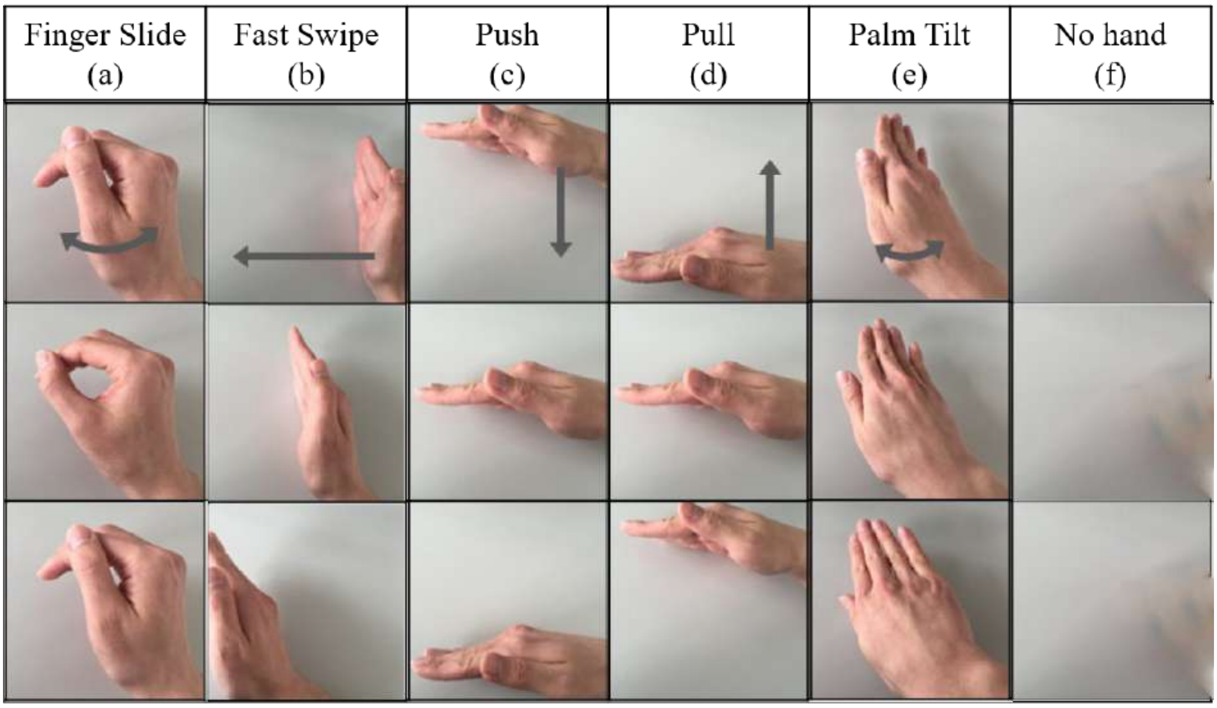


Figure 4.3: The overview of the 6 dynamic hand gestures dataset, adapted from [69]

1. Finger Slide: Slide the thumb left and right on the index finger to imitate the user's control on the electronic screen.

2. Fast Swipe: Move the palm quickly from the right to the left.

3. Push: Palm from top to bottom, imitating pushing action.

4. Pull: The palm moves from the bottom up, imitating the action of pulling up.

5. Palm Tilt: The process of tilting the palm to laying flat.

6. No hand: Surrounding environment information.

IR-UWB non-contact sensor transmits short pulses at the sweep frequency and hits the human hands. A single sensor records the phase and amplitude of the echo signal with a sweep rate of 256Hz in the 5-gesture dataset, including 10 sessions with 50 data each of a single user. The parameters of the dataset are depicted in Table4.1.

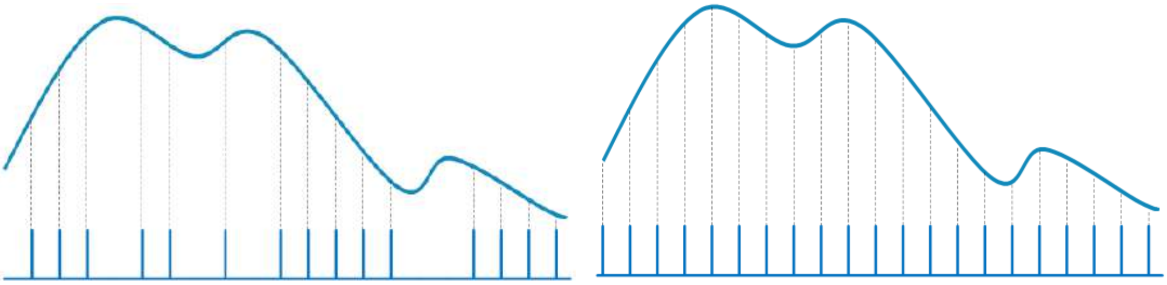
Table 4.1: Parameters used to record the 5 and 6 gestures dataset

Parameters	**5 gestures**	**6 gestures**
Sweep frequency	256Hz	256Hz
Recording length	3s	3s
People	1	1
Instances per gestures	500	500
Instances per person	2500	3000
Total instances	2500	3000
Sweep ranges	10-30cm	10-30cm
Sensor module used	Acconeer XR111	Acconeer XR111

4.3.2 Pre-processing Method Design

LC-ADCs sample the data according to the slope of the signal[58]. The basic principle is that when the input signal changes significantly, the sampling event is considered to occur, and the position of the sampling point is recorded[70]. Since the LC-ADCs contain temporal information naturally and the output is a spikes signal, it is more suitable for neuromorphic computing than Nyquist sampling ADCs. Fig. 4.4 elucidates two sampling methods for neuromorphic algorithm pre-processing: a) Nyquist sampling and b) LC sampling. At the same sampling rate, the number of sampling pulses of Nyquist is much greater than that of the LC sampling method. A recent study compared the power efficiency of using LC-ADC and traditional Successive-Approximation-Register ADC to process various biological signals[71], clarifying that the power efficiency of LC-ADC is higher than that of standard ADC in low- and medium-resolution applications.

In order to convert the data information into an input signal suitable for SNN, we first perform down-sampling firstly to increase the sparsity of the raw data, then LC-sampling is used to convert the original floating-point amplitude information into spikes signals as the input of SNN. We also improve the LC-ADCs method from one threshold to two, i.e., LC-ADCs will



(a) Nyquist sampling

(b) Traditional LC sampling

Figure 4.4: Sampling approach comparison

sample up spikes when the amplitude is above a pre-defined high level or down spikes when it is below a low-level threshold [44].

4.3.3 Classification Method Design

Like in the brain, the information of the SNN is usually in the form of an event stream encoding by spike timing, spike latencies, and spike rates[72]. The synaptic weight modulates the pre-spike to generate a resultant current into the post-neuron, modeled as integrate-and-fire or leaky-integrate-and-fire (LIF)[73]. Bio-inspired spike-timing-dependent plasticity (STDP) has emerged as an unsupervised learning rule for synaptic weight modification. Therefore, SNN is biologically interpretable and has potential in neuromorphic real-time computing and control. Stuijt *et al.* [74] and Ceolini *et al.* [75] demonstrate the superiority of SNN in neuromorphic computing and lead to a close-loop (from development to deployment) in the application of AI and IoT (collectively, AIoT).

In the SNN processing stage, we use a spiking multi-layer perceptron (MLP) for the hand gestures classification. In the perceptron, each neuron is represented by a LIF model. As a biologically inspired unsupervised learning rule, STDP trains the synaptic weights and biases of the hidden layer. The final output neuron will converge to five different gesture data, illustrating the classification results of the IR-UWB radar gesture dataset.

4.4 Summary

This chapter first explains the principle of sensing system design and then proposes the system architecture of our gesture recognition system. The whole system consists of three parts, gesture data set, pre-processing module, and SNN processing module. We found the most suitable HGR sensors (IR-UWB contactless sensor), pre-processing method (LC-sampling), and gesture recognition algorithm (SNN) for the sensing system by consulting the relevant literature. Finally, we detail the design work of each stage.

5 Sensing System Implementation and Evaluation

Contol

Portions of text and figures reproduced from author's work [44].

5.1 Overview

This chapter first introduces the raw gesture data format of the IR-UWB non-contact sensor. Then it explains the implementation process of the sensing system, including data pre-processing and classification algorithms. We obtain the recognition accuracy of the algorithm through experiments. To illustrate the excellence of the algorithm, we also evaluate the algorithm and compare the effects of different pre-processing methods and different algorithm parameters on the accuracy. Finally, we calculate the power complexity of the algorithm and compare it with other methods, indicating that our proposed HGR method can achieve high accuracy in low-power scenarios

5.2 IR-UWB Contactless Sensor Dataset

To verify the performance of our gesture recognition algorithm, we adopt the dynamic hand gestures data in Scherer *et al.*[13] as our sample [44]. The raw data is a series of analog I/Q signal parameters, including the distance of the sensor, amplitude, and phase information of the signal. Figure 5.1 shows an example IR-UWB hand gesture signal, the width and height dimensions correspond to the timing ranging and the sampling range, respectively. While the whole recording time is 3s and the sweep frequency is 256Hz, we can get 768 sweep numbers called slow time. At each sweep, the non-contact sensor emits a pulse to get information with another frequency, regarded as fast time. After recording, we can get a 768*414 data matrix [44].

Since the Acconeer sensor uses one antenna for transmitting and receiving, the transmitter and receiver are at the same position in the sensor. The trajectory of the signal is twice the distance between the object and the non-contact sensor. According to the ToF principle $2 * d = c * t$ (where d is the distance between the object and the IR-UWB sensor, c is the speed of light, and t is the time of flight), we can regard time as a variable related to distance. Therefore, we can convert Range-Amplitude data into Time–Amplitude data [44].

5.3 Pre-processing Method

To convert the raw data matrix into the input of the SNN (spikes sequence), we rearranged the data and processed downsampling and level-crossing sampling. Here are the steps in detail [44].

1. Rearrange the amplitude information in IR-UWB data with data points (fast time) as the x-axis and sweep numbers (slow time) as the y-axis. A numpy array am_data was used to store these data. Figure 5.2 shows the rearranged data matrix.

2. To increase the data sparsity, we have performed 10× and 3× down-sampling on the x-axis and y-axis of the data, respectively. The size of the sparse matrix is 256*42.

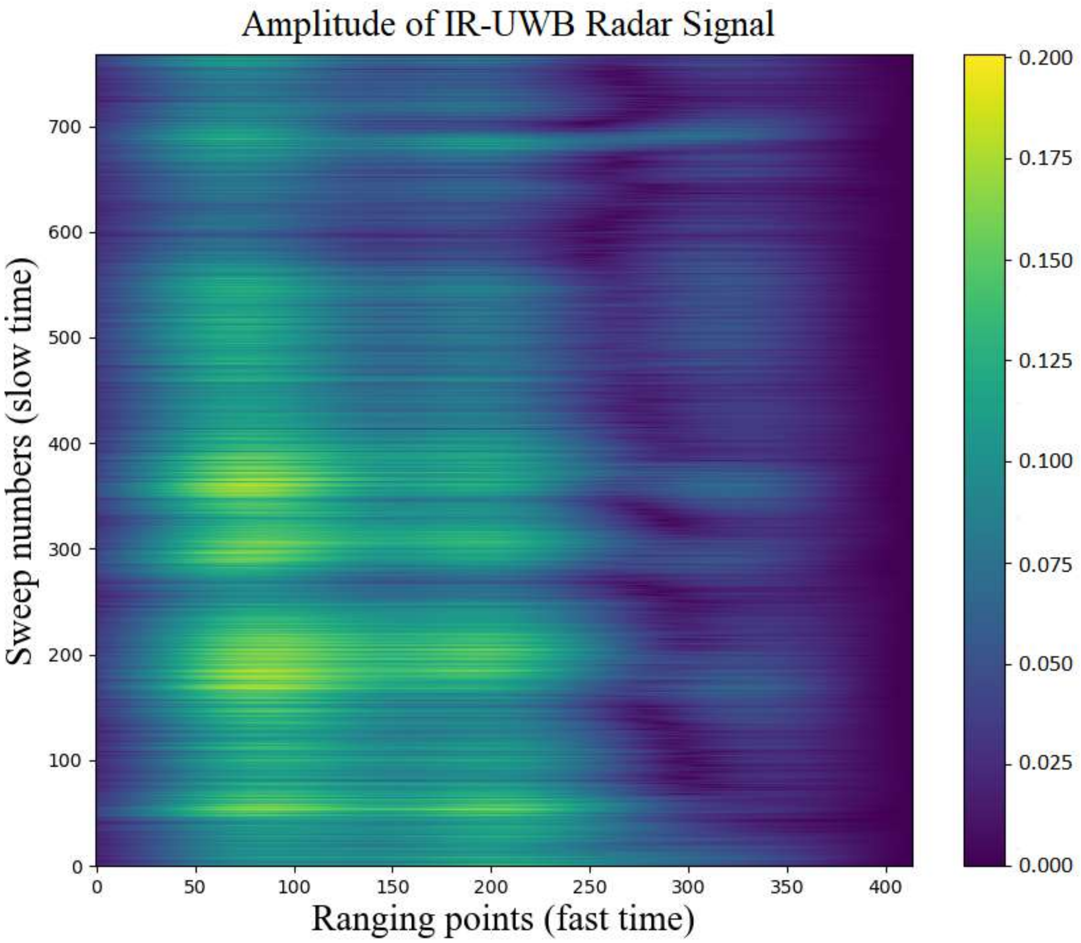


Figure 5.1: IR-UWB signal for an example recording (Finger Slide)

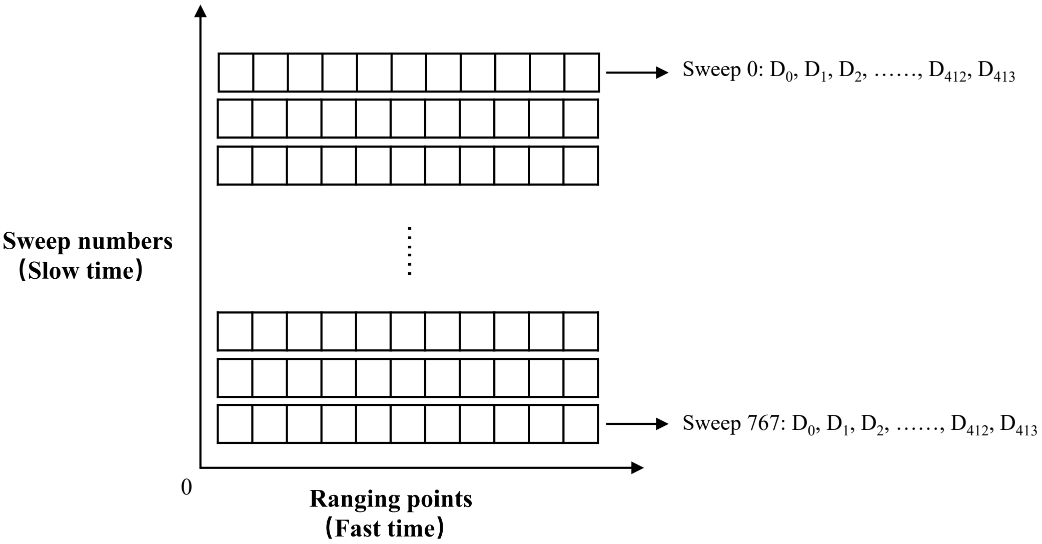

Sweep 0: D_0, D_1, D_2, ..., D_{412}, D_{413}
Sweep numbers (Slow time)
Sweep 767: D_0, D_1, D_2, ..., D_{412}, D_{413}
0
Ranging points (Fast time)

Figure 5.2: Data matrix of IR-UWB contactless sensor

3. We perform an improved level-crossing sampling method on each row of data. We first define two all-zero numpy arrays: up_spikes and $down\_spikes$. Set the first point's amplitude as the reference value (V_{am}), then the high-level value (V_h) is the reference value add threshold (V_{th}) and the low-level value (V_l) is the reference value minus threshold. Judge whether the value of the next point is between V_h and V_l. If the amplitude is beyond this range, we will sample an up_spike and change the reference value to the new value (like point 4 in Figure 5.3). Otherwise, if the value is lower than V_l, we will sample a $down\_spike$ and refresh the reference value. Finally, we can get two arrays called up_spikes and $down\_spikes$ after sampling (Figure 5.4).

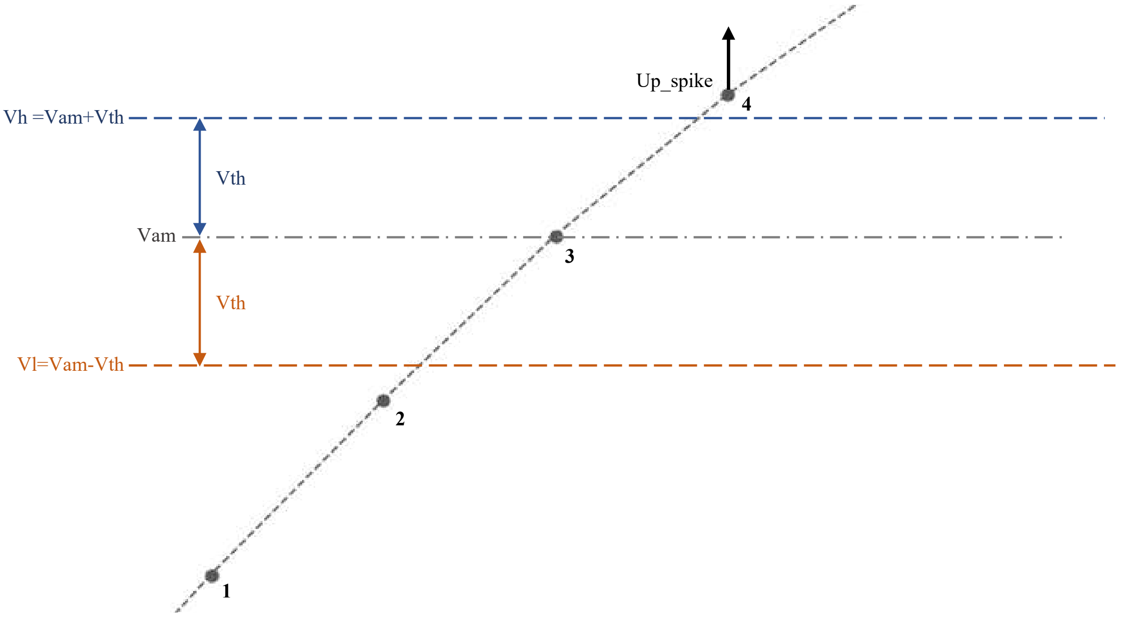
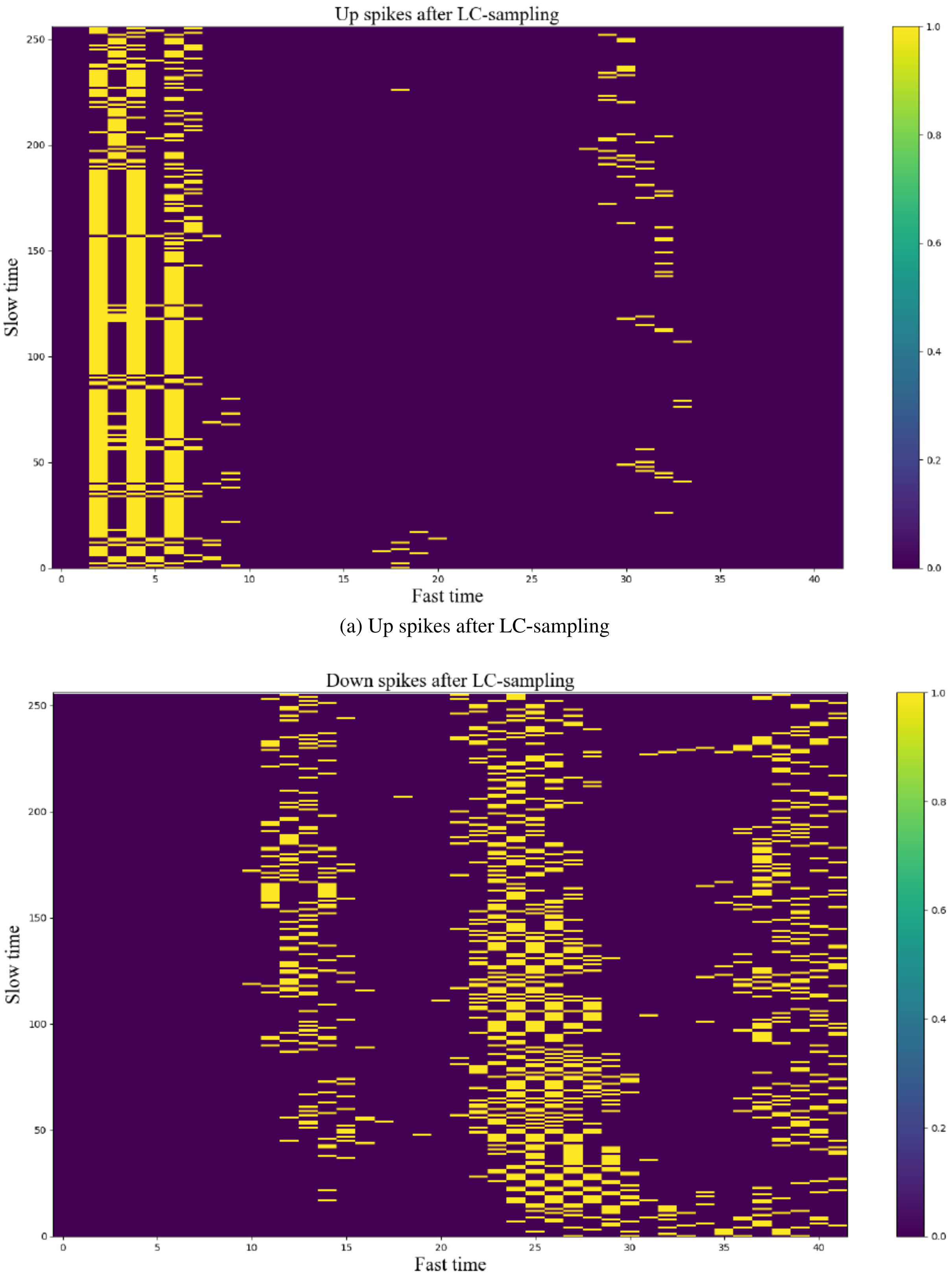


Figure 5.3: LC sampling of one data point

4. The size of up_spikes and $down\_spikes$ array is 256*42. To transform them as the input of SNN, we combine the two arrays and get a 512*42 one.

After the down-sampling and LC-sampling, a 768*414 sensor data matrix has become a 512*43 spikes array. The data type has also changed from floating-point (Figure 5.1) to integer (Figure 5.4) for computational complexity reduction. The large dark area in Figure 5.4 also illustrates that the value of 0 occupies a large proportion of the sampled data.



(a) Up spikes after LC-sampling

(b) Down spikes after LC-sampling

Figure 5.4: IR-UWB signals after LC-sampling

5.4 Classification Method

Recently, the three popular SNN training algorithms are a) conversion-based method [76], b) spike-timing-dependent plasticity [72], and c) error backpropagation[73]. The conversion-based technique, on the other hand, uses the firing rate to represent information when converting the SNN parameters from the trained ANN, resulting in high-density spikes. As for a biologically-inspired training method, STDP changes the weight of a synapse according to the time correlation between the peaks of presynaptic and postsynaptic neurons. Combined with the error backpropagation method, a graph-based spatio-temporal backpropagation was proposed to train SNN[77].

In this study, we employed a graph-based spatio-temporal backpropagation training approach to train SNN on the IR-UWB non-contact sensor gesture dataset. The backpropagation route is established using a differentiable LIF neural dynamic model. To reduce the spike firing rate and increase effectiveness, sparse regularization is proposed.

A common LIF model can be expressed as:

$$\tau \frac{du(t)}{dt} = -u(t) + I(t) \tag{5.1}$$

where τ is the time constant. $I(t)$ represents presynaptic input, determined by pre-neuronal activity or external injection and synaptic weight. $u(t)$ is the membrane voltage at time t. When the membrane voltage reaches the threshold, a pulse is released then its electrical position resets to U_{rest}[78]. We also designed a differentiable LIF model to solve the non-differentiable issues with backpropagation in SNN. It defines the i-th LIF neuron membrane potential u_i^t with Equ. (5.2) and the state s_i^t with Equ. (5.3) at discrete time t.

$$u_i^t = u_i^{t-1} \cdot \tau \cdot g_1(-s_i^{t-1}) + \sum_j w_{ij} \cdot s_j^{t-1} + b_i \tag{5.2}$$

$$s_i^t = g_2(u_i^t - U_{th}), s_i^t \in \{0, 1\} \tag{5.3}$$

where τ denotes the potential leakage ratio, w_{ij} denotes the synaptic weight from neuron j-th to neuron i-th, and b_i is the bias of neuron i-th. Backpropagation's route is gated by $g_1^{'}$. Wu *et al.*[78] suggested the approximation $g_2^{'}$, whose form is determined by the coefficient a.

$$g_1^{'}(x) \approx \begin{cases} 0 & x = 0 \\ 1 & x \neq 0 \end{cases}, \; g_2^{'}(x) = \frac{1}{\sqrt{2\pi a}} e^{-\frac{x^2}{2a}}, \; x \in R \tag{5.4}$$

This improved model has been proved in [77] with 97.3% accuracy in MNIST.

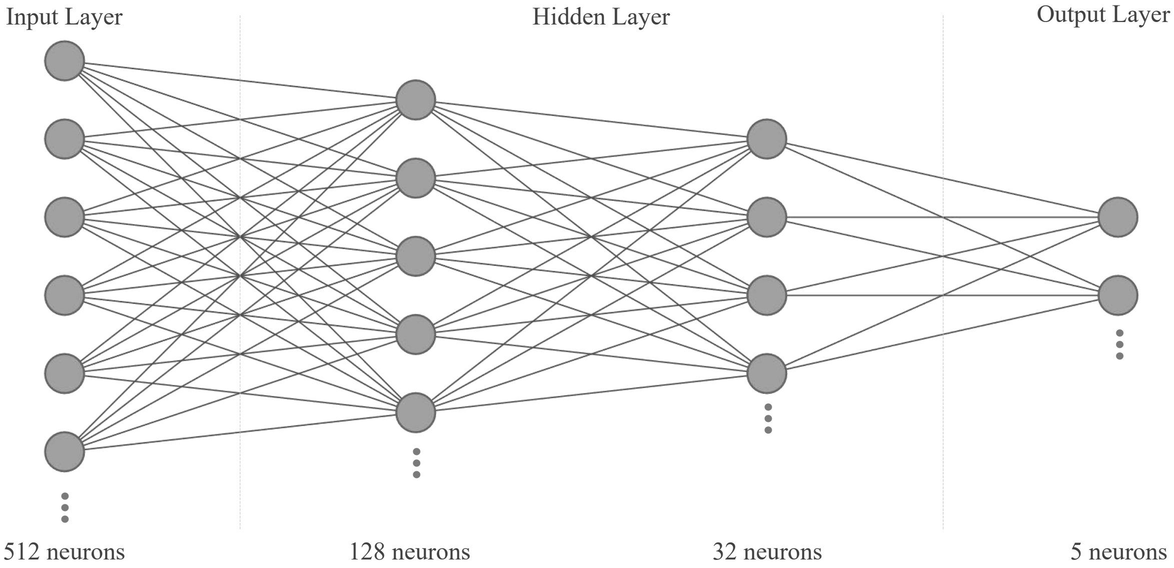


Figure 5.5: Spiking MLP *'512-fc128-fc32-fc5'*(fc means full connection)

Figure 5.5 is the spiking MLP denoted by *'512-fc128-fc32-fc5'* in the following discussions. It contains 512 neurons in its input layer, 128 and 32 neurons in each of its two hidden layers, and 5 neurons in its output layer, which correspond to five different hand movements. We regard the fast time of the sensor as the running time of the SNN with the 512 neurons we input per unit time carrying different temporal information in the same space. According to the differentiable LIF model we proposed, the network topology can be represented as a directed weighted graph.

The adjacency matrix of weights is written as $W(i, j) = w_{ij}$, meaning the corresponding weight between synaptic i and synaptic j. When the value is zero, there is no connection between the two synaptic. Then we put the 512*42 size spikes array into the network, where 42 represents the runtime of SNN. Therefore, in each unit time, we will input 512 neurons into SNN, lasting for a total of 42 unit times. The input 512 data points will take a dot product with the weight graph $W(i, j)$ between the input layer and the hidden layer and then add a neuron bias $B(i)$, conforming to the Gaussian distribution. The error is backpropagation through time. The edge of the graph, the weight matrix W, and the bias matrix B are all updated by an optimization algorithm. Finally, the MLP uses the sigmoid activation function to map the weighted inputs to the next layer of each neuron again and again until the next layer is the output layer [44].

5.5 Experiment of Hand Gesture Recognition System

5.5.1 Training Setup

Using the Pytorch framework, the SNN was developed and is powered by AMD Ryzen-3970X (CPU) and NVIDIA RTX-3080 (GPU). We divided the whole dataset into a training dataset and a testing dataset with a ratio of 4 to 1 randomly. The LIF neuron and the training procedure employ the settings in the table, which were successfully tested in [77]. To calculate the accuracy of the algorithm, we performed validation and 5-fold cross-validation (CV5) training on the data set. The results of CV5 are in the following.

5.5.2 Accuracy of the Algorithm

The inference accuracy of the algorithm is discussed in terms of per-sequence accuracy. To make the data more convincing, we did ordinary verification and CV5, respectively. The firing rate is 0.1 and $30\times$ sparseness. Table5.2 displays the results, indicating that the proposed

Table 5.1: Parameters setting in experiments, adapted from [77]

Parameter	Description	Value
U_{th}	Potential threshold	1.5
τ	Ratio of leakage	0.1%
a	$g_2^{'}(\cdot)$ coefficient	1.0
p_r	Recurrent probability	0.5
T	Runtime of SNN	42
λ	Sparse regularization	5×10^{-5}
E	Training epochs	40
l_r	Learning rate	0.01
S_{ch}	Decay learning rate	decay:0.2
	Patience epochs improvement	patience: 5

method can reach an accuracy of 95.44% of 5 hand gesture recognition and 96.60% of 6 gestures.

Table 5.2: Accuracy of the algorithm on the respective train/test set

	5G-Vali	**5G-CV5**	**6G-Vali**	**6G-CV5**
Train	98.40%	98.97%	99.09%	99.18%
Test	95.44%	95.36%	96.60%	96.00%

5.6 Evaluation of Hand Gesture Recognition System

5.6.1 Evaluation of Pre-Processing Methods

To reduce power consumption to a lower level, we evaluated different pre-processing methods for neural networks. Fourier transform of the time axis is proved as an effective method in feature extraction[79]–[81]. We tried the FFT on signal and compared it with the downsampling.

Since the raw sensor signal data contains the amplitude and phase information, I converted them into I/Q signals expressed in complex numbers to perform FFT by formula (5.5), where A represents Amplitude and P for Phase. Do the FFT on a complex array and get the absolute value with the $scipy.fft()$ and $abs()$, respectively. Then we can get the frequency response of the signal.

$$S = A * cos(P) + A * sin(P)j \tag{5.5}$$

In addition to LC sampling, we also designed an absolute threshold sampling method to convert floating-point values into a 0/1 sequence. We tested the accuracy of the four pre-processing methods combining FFT, downsampling, absolute threshold sampling, and LC sampling. The number of spikes in the data set represents the power consumption of the entire network. Table 5.3 elucidates the experimental results, adopting the same spareness (30×) and firing rate (10%).

Table 5.3 shows that the accuracy of combining FFT and SNN is much lower than downsampling and LC sampling with SNN. The reason is that FFT converts the signal data into a spectrogram like the range frequency Doppler map that is more suitable for CNN rather than SNN. FFT makes the signal lose the time domain information, leaving only the spectrum information. Although Scherer *et al.*[13] have proved the efficiency of FFT as the feature extraction of CNN and TCN, for SNN, the accuracy rate in our experiment is only about 54.60%. In conclusion, combining down-sampling and level-crossing sampling is the highest accuracy preprocessing method for SNN so far.

5.6.2 Evaluation of Classification Algorithm

In SNN training, we did some experiments to balance the accuracy and spike numbers. We use firing rate as a metric, determined by the number of spikes in the data set divided by the amount

Table 5.3: Algorithm performance under different pre-processing methods

Pre-pro methods	**5G**		**6G**	
	Acc	Spikes	Acc	Spikes
FFT-AS	54.60%	2551438	59.00%	2774386
FFT-LCS	68.00%	4968153	63.00%	6451200
DS-AS	76.00%	3006395	62.50%	3225600
DS-LCS	**95.44%**	5523145	**96.60%**	6627774

of data in the data set. Finding a proper value for the firing rate is necessary. If the firing rate is too large, operands and the power consumption of the network will be huge. If the firing rate is too low, plenty of information will be lost, leading to the low accuracy of the algorithm. We can limit the firing rate by changing the size of the threshold during level-crossing sampling. The smaller the threshold value, the greater the firing rate, and the more operations the subsequent network needs to perform. We explored the influence of voltage threshold on the accuracy under different sparsity and showed the results with figure 5.6.

The ordinate on the left presents the accuracy achieved by the algorithm, while the ordinate on the right presents the number of spikes in each data set, representing the power consumption of SNN. Figure 5.6 depicted that the performance and power consumption can be balanced by an appropriate voltage threshold (0.02) to achieve high accuracy and low power SNN. When the threshold is 0.02, the firing rate is about 10%. According to the comparison of the parameters between 30x sparsity and 15x sparsity, sparsity has little effect on accuracy, while the spikes number of 30x is much smaller than 15x. As for 40x sparsity, though the spikes number is the smallest, its accuracy is lower than 95%. So we adopt a firing rate with 10% and 30x sparsity as the parameter of our algorithm, the accuracies with two data sets are over 95% with 6.6×10^6 spikes.

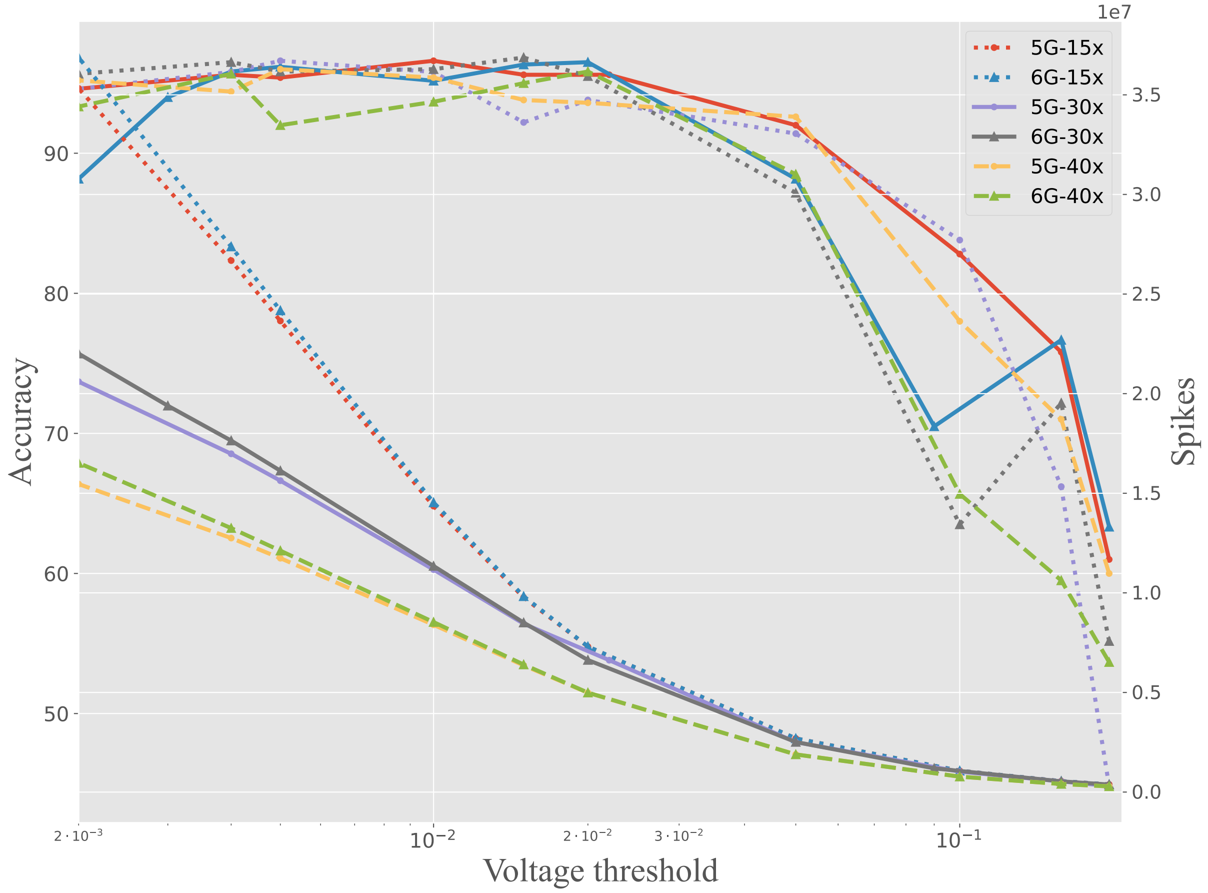


Figure 5.6: Accuracy under different firing rates and sparseness

5.6.3 Power Consumption Complexity Calculation

Since the proposed model is implemented on the CPU at this stage, we use the operands required for each instance to compare the power consumption [44]. The value of each neuron is only 0 and 1, so we can ignore all operations with 0 and regard the multiplication and addition calculation with 1 as an addition calculation between the weights. Furthermore, the LIF we used needs to be multiplied by a step function when each neuron outputs. So the total number of multiplication operations is only the number of neurons in the hidden layer and the output layer. As for the addition operands, the model is a fully connected structure, equivalent to an addition operation for each synapse at each time point. The firing rate also plays an important role in power consumption complexity calculation. However, the firing rate here is different from that discussed in the pre-processing method. The firing rate we discussed now is a term in rate coding referring to various averaging procedures, such as an average over time (rate as a single-neuron spike count) or an average over several repetitions (rate of peristimulus time histogram) of the experiment. The firing rate means the number of synapses fired between every two layers in the MLP accounted for the total number of synapses, represented by f_r in the equation.

Multiplication operand calculation:

$$Mul = (N_{h1} + N_{h2} + N_o) * \triangle t \tag{5.6}$$

Addition operand calculation:

$$Add = (N_i * N_{h1} * fr_i + N_{h1} * N_{h2} * fr_1 + N_{h2} * N_o * fr_2) * \triangle t \tag{5.7}$$

In this work, the spiking MLP for five gesture recognition is denoted by *'512-fc128-fc32-fc5'*. So, the multiplication operand and the addition operand are:

$$Mul_{5g} = (128 + 32 + 5) * 42 = 6930$$

$$\begin{gathered} Add_{5g} = (512 * 128 * 0.1042 + 128 * 32 * 0.3109 \\ + 32 * 5 * 0.4374) * 42 = 343236 \end{gathered}$$

To make the data more convincing, we add the addition and multiplication operand to represent the power consumed by each instance in the network. Therefore, the performance of the proposed network is 350 kFLOPs per inference, while TinyRadarNN[13] is 1582 MFLOPs with the same dataset.

5.6.4 Comparison with Other HGR Systems

It's interesting to compare our work with Scherer *et al.*[13] because we use the same dataset of IR-UWB non-contact sensor. The total operands required by TinyRadarNN is 31.8MFLOPs. As for our algorithm, the calculation complexity is 0.35MFLOPs, which is $90\times$ smaller than the TinyRadarNN according to the operation numbers. To be more convinced, we also researched some other neural networks with IR-UWB contactless sensors. Table 5.4 shows the parameters of each network, including the network name, the accuracy of the algorithm, the number of gestures, and the operands (presenting the power consumption). The definition of the data format comes from Ahmed *et al.* [53]. Among them, Kim *et al.* adopt the same data format of IR-UWB signal and pre-processing method with our work[56]. However, the accuracy of their gesture recognition algorithm is around 90%, elucidating that SNN is more suitable than CNN to process pulsed data with sampling pre-processing. Therefore we also do not count these network operands as the accuracy has become uncompetitive. Ahmed *et al.*[12] and Li *et al.*[11] adopt GoogLeNet and ShuffleNet V2 as the classification method. Although the accuracy rate is higher than our proposed SNN algorithm with more gestures, the power consumption is much higher than ours. According to the table, the model we proposed can achieve a relatively high accuracy rate with low computational complexity. Moreover, the pre-processing method we adopted is LC sampling, which is power-efficient. Compared to FFT, LC sampling has almost no computational complexity and is more suitable for SNN.

5.7 Summary

This chapter presents a novel hand-gesture recognition model combining the IR-UWB signal and SNN model with high accuracy and ultra-low-power consumption [44]. We converted IR-UWB hand gesture signals into a 0/1 sequence that conforms to the SNN input by reorganizing, sampling, and encoding. An improved LIF appeals to solve the non-differentiable problem

Table 5.4: Comparison with other hand gesture recognition methods

Study and year	**Algorithm**	**No. of gestures**	**Data Format**	**Pre-Processing**	**FLOPs**		**Acc**
					Mul	**Add**	
Kim *et al.*[56] (2017)	(1-D) CNN	6	Time–Amplitude(1D)	Filtering	-	-	90%
Ahmed *et al.*[12] (2020)	GoogLeNet	8	Time–Range(3D-RGB)	Clutter Removal	1582M	1582M	95%
Li *et al.*[11] (2021)	ShuffleNet V2	7	Time–Range(2D)	2D-FFT	2.76M	2.76M	98.52%
Scherer *et al.*[13] (2021)	TCN+CNN	5	Time–Range(2D)	2D-FFT	15.90M	15.90M	95%
This paper (2021)	SNN	5	Time–Amplitude (1D)	LC-sampling	6.93k	343k	95.44%
		6			6.97k	313k	96.60%

for backpropagation in SNN. By comparing experiments with other pre-processing methods, we proved the superiority of using LC-sampling and SNN algorithms in IR-UWB non-contact sensor gesture recognition. Finally, the accuracy can reach 95.44% with five hand gestures and 96.60% with six gestures. The operation numbers are 350 kFLOPs per data sequence on the five hand gestures dataset, which is 90× less than the previous network.

6 Conclusion and Future Work

6.1 Conclusion

In recent years, IR-UWB has received much attention from IIoT and smart factory scenarios due to its high temporal resolution, low power consumption, and high robustness. In this thesis, we made an investigation and analysis of IR-UWB technology. A positioning system and a sensing system are designed and implemented based on the TOF algorithm, exploring a new thought for equipment positioning and HMI sensing in the smart factory environment. For example, both systems can be applied to multi-robot or drone arrays in smart factories, helping them understand the location of other devices in real-time and recognize human gesture information to complete HMI's instructions.

In the positioning system, we abandon the need to deploy IR-UWB base stations in traditional absolute positioning and realize a flexible and convenient relative positioning system by carrying multiple IR-UWB modules on a single robot or drone. We used the UWB transceiver module of Bluepoint Technology to test the accuracy of the relative positioning system and tested the positioning results of 8 possible positions of the robot pair/group. The results show that when the distance is greater than 1m between the target robot and the main robot, the positioning system can achieve 93.09% accuracy. In the sensing system, we use Acconeer's IR-UWB radar gesture dataset, perform signal preprocessing through LC-ADC sampling, and then use SNN as the gesture recognition algorithm, thus realizing a high-precision, low-power gesture sensing system. Through model training on PyTorch, the algorithm finally achieved 95.44% accuracy on five gesture datasets and 96.60% accuracy on six gesture datasets. In addition, the computational complexity of this model is also about 90× lower than other similar works.

6.2 Future Work

IR-UWB based positioning and sensing systems provide a good starting point for discussion and further research on applications in IIoT and smart factories scenarios. However, due to the time and experimental limitations, these two systems still have some parts that need to be improved. Several recommendations for future research are given as follows.

First of all, we believe that apart from looking for how to improve the performance of the two systems, future research should focus on how to integrate the two systems. Because both the positioning and sensing systems use IR-UWB as the sensor, it would be an interesting and feasible topic to combine these two systems into an integrated system for future work. This integrated system can collect environmental information through a non-contact IR-UWB sensor, achieve distance measurement with the target through the TOF algorithm, and then combine the required positioning or sensing algorithm to complete the positioning or gesture recognition function.

Secondly, further research on more human postures and motion recognition is warranted. Looking forward, human motion detection might play a vital role for HMI in IIoT and smart factories scenarios. It will be important for future research to build a data collection platform for more gesture, posture, and motion information. In addition, we can also integrate the IR-UWB signal pre-processing method and the HGR recognition algorithm into one module. Then provide an application programming interface for the dataset. Through the API, we can connect different IR-UWB gesture datasets, test the performance of the sensing system in various datasets, and get the generalization ability of the system.

Last but not least, future research could continue to explore the methods for improving the ranging accuracy of the IR-UWB modules, which could prove directly beneficial to the performance of positioning and sensing systems.

References

[1] H. Kagermann, W. Wahlster, and J. Helbig, “Securing the future of german manufacturing industry: Recommendations for implementing the strategic initiative industrie 4.0”, Tech. Rep., 2013.

[2] Shandong USR Internet of Things Co., Ltd., *Worker positioning system project*, `http://www.iwl.iiot.com/case/800001032.html`, Accessed June 4, 2020, 2020.

[3] P. Wang, F. Qi, M. Liu, *et al.*, “Noncontact heart rate measurement based on an improved convolutional sparse coding method using ir-uwb radar”, *IEEE Access*, vol. 7, pp. 158 492–158 502, 2019.

[4] F. Khan, S. K. Leem, and S. H. Cho, “Hand-based gesture recognition for vehicular applications using ir-uwb radar”, *Sensors*, vol. 17, no. 4, p. 833, 2017.

[5] J. R. Fernandes and D. Wentzloff, “Recent advances in ir-uwb transceivers: An overview”, in *Proceedings of 2010 IEEE international symposium on circuits and systems*, IEEE, 2010, pp. 3284–3287.

[6] Q. V. d. Brande, S. Lemey, and H. Rogier, “Planar sectoral antenna for ir-uwb localization with minimal range estimation biasing”, *IEEE Antennas and Wireless Propagation Letters*, vol. 20, no. 2, pp. 135–139, 2021.

[7] N. Macoir, J. Bauwens, B. Jooris, *et al.*, “Uwb localization with battery-powered wireless backbone for drone-based inventory management”, *Sensors*, vol. 19, no. 3, p. 467, 2019.

[8] M. Ridolfi, N. Macoir, J. Vanhie-Van Gerwen, J. Rossey, J. Hoebeke, and E. De Poorter, "Testbed for warehouse automation experiments using mobile agvs and drones", in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, IEEE, 2019, pp. 919–920.

[9] T. H. Mogale, B. J. Silva, and G. P. Hancke, "A portable ir-uwb based wsn for personnel tracking in emergency scenarios", in *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, 2016, pp. 961–965.

[10] Y.-C. Chen, I. Alexsander, C. Lai, and R.-B. Wu, "Uwb-assisted high-precision positioning in a utm prototype", in *2020 IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNeT)*, IEEE, 2020, pp. 42–45.

[11] Y. Li, X. Wang, B. Shi, and M. Zhu, "Hand gesture recognition using ir-uwb radar with shufflenet v2", in *Proceedings of the 5th International Conference on Control Engineering and Artificial Intelligence*, 2021, pp. 126–131.

[12] S. Ahmed and S. H. Cho, "Hand gesture recognition using an ir-uwb radar with an inception module-based classifier", *Sensors*, vol. 20, no. 2, p. 564, 2020.

[13] M. Scherer, M. Magno, J. Erb, P. Mayer, M. Eggimann, and L. Benini, "Tinyradarnn: Combining spatial and temporal convolutional neural networks for embedded gesture recognition with short range radars", *IEEE Internet of Things Journal*, 2021.

[14] A. Chehri and H. T. Mouftah, "Internet of things-integrated ir-uwb technology for healthcare applications", *Concurrency and Computation: Practice and Experience*, vol. 32, no. 2, e5454, 2020.

[15] H. Shen, C. Xu, Y. Yang, *et al.*, "Respiration and heartbeat rates measurement based on autocorrelation using ir-uwb radar", *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 10, pp. 1470–1474, 2018.

[16] J. W. Choi, S. H. Cho, Y. S. Kim, N. J. Kim, S. S. Kwon, and J. S. Shim, “A counting sensor for inbound and outbound people using ir-uwb radar sensors”, in *2016 IEEE Sensors Applications Symposium (SAS)*, 2016, pp. 1–5.

[17] J.-E. Kim, J.-H. Choi, and K.-T. Kim, “Robust detection of presence of individuals in an indoor environment using ir-uwb radar”, *IEEE Access*, vol. 8, pp. 108 133–108 147, 2020.

[18] W. Shule, C. M. Almansa, J. P. Queralta, Z. Zou, and T. Westerlund, “Uwb-based localization for multi-uav systems and collaborative heterogeneous multi-robot systems”, *Procedia Computer Science*, vol. 175, pp. 357–364, 2020.

[19] J. P. Queralta, C. M. Almansa, F. Schiano, D. Floreano, and T. Westerlund, “Uwb-based system for uav localization in gnss-denied environments: Characterization and dataset”, in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2020, pp. 4521–4528.

[20] J. S. Furtado, H. H. Liu, G. Lai, H. Lacheray, and J. Desouza-Coelho, “Comparative analysis of optitrack motion capture systems”, in *Advances in Motion Sensing and Control for Robotic Applications*, Springer, 2019, pp. 15–31.

[21] D. Ni, O. A. Postolache, C. Mi, M. Zhong, and Y. Wang, “Uwb indoor positioning application based on kalman filter and 3-d toa localization algorithm”, in *2019 11th International Symposium on Advanced Topics in Electrical Engineering (ATEE)*, IEEE, 2019, pp. 1–6.

[22] G. Yang, L. Zhao, Y. Dai, and Y. Xu, “A kfl-toa uwb indoor positioning method for complex environment”, in *2017 Chinese Automation Congress (CAC)*, IEEE, 2017, pp. 3010–3014.

[23] W. Wang, J. Huang, S. Cai, and J. Yang, “Design and implementation of synchronization free tdoa localization system based on uwb”, *Radioengineering*, vol. 27, no. 1, pp. 320–330, 2019.

[24] L. Wei, J. Yu, X. Zhao, and W. Pan, “Design of intelligent supervision system for metrology center based on UWB technology”, *Journal of Physics: Conference Series*, vol. 1983, no. 1, p. 012 070, Jul. 2021.

[25] S. G. Pease, P. P. Conway, and A. A. West, “Hybrid tof and rssi real-time semantic tracking with an adaptive industrial internet of things architecture”, *Journal of Network and Computer Applications*, vol. 99, pp. 98–109, 2017.

[26] F. Bonafini, A. Depari, P. Ferrari, *et al.*, “Exploiting localization systems for lorawan transmission scheduling in industrial applications”, in *2019 15th IEEE International Workshop on Factory Communication Systems (WFCS)*, 2019, pp. 1–8.

[27] M. Martalò, G. Ferrari, S. Perri, G. Verdano, F. De Mola, and F. Monica, “Uwb tdoa-based positioning using a single hotspot with multiple anchors”, in *2019 4th International Conference on Computing, Communications and Security (ICCCS)*, IEEE, 2019, pp. 1–7.

[28] P. Zhao, X. Zhu, L. He, Z. Yang, S. Zuo, and Z. Zhao, “Uwb-rtk positioning system based on tdoa”, in *2019 UK/China Emerging Technologies (UCET)*, IEEE, 2019, pp. 1–4.

[29] Y. Cheng and T. Zhou, “Uwb indoor positioning algorithm based on tdoa technology”, in *2019 10th international conference on information technology in medicine and education (ITME)*, IEEE, 2019, pp. 777–782.

[30] L. Taponecco, A. D’Amico, and U. Mengali, “Joint toa and aoa estimation for uwb localization applications”, *IEEE Transactions on Wireless Communications*, vol. 10, no. 7, pp. 2207–2217, 2011.

[31] L. Barbieri, M. Brambilla, A. Trabattoni, S. Mervic, and M. Nicoli, “Uwb localization in a smart factory: Augmentation methods and experimental assessment”, *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–18, 2021.

[32] K. Lin, W. Wang, Y. Bi, M. Qiu, and M. M. Hassan, “Human localization based on inertial sensors and fingerprints in the industrial internet of things”, *Computer Networks*,

vol. 101, pp. 113–126, 2016, Industrial Technologies and Applications for the Internet of Things, ISSN: 1389-1286.

[33] F. Subhan, A. Khan, S. Saleem, *et al.*, "Experimental analysis of received signals strength in bluetooth low energy (ble) and its effect on distance and position estimation", *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 2, e3793, 2022, e3793 ETT-19-0391.R1.

[34] M. Ramezani, G. Tinchev, E. Iuganov, and M. Fallon, "Online lidar-slam for legged robots with robust registration and deep-learned loop closure", in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 4158–4164.

[35] J. Lin, W. Li, and J. Xia, "Circle detection and location based on binocular stereo vision", in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1678, 2020, p. 012 104.

[36] F. Che, A. Ahmed, Q. Z. Ahmed, S. A. R. Zaidi, and M. Z. Shakir, "Machine learning based approach for indoor localization using ultra-wide bandwidth (uwb) system for industrial internet of things (iiot)", in *2020 International Conference on UK-China Emerging Technologies (UCET)*, 2020, pp. 1–4.

[37] Y. Yun, J. Lee, D. An, S. Kim, and Y. Kim, "Performance comparison of indoor positioning schemes exploiting wi-fi aps and ble beacons", in *2018 5th NAFOSTED Conference on Information and Computer Science (NICS)*, 2018, pp. 124–127.

[38] H.-S. Yeo and A. Quigley, "Radar sensing in human-computer interaction", *Interactions*, vol. 25, no. 1, pp. 70–73, Dec. 2017, ISSN: 1072-5520.

[39] M. Patel, S. Pandya, and S. Patel, "Hand gesture based home control device using iot.", *International Journal of Advanced Research in Computer Science*, vol. 8, no. 5, 2017.

[40] R. Nair, D. K. Singh, S. Yadav, S. Bakshi, *et al.*, "Hand gesture recognition system for physically challenged people using iot", in *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, IEEE, 2020, pp. 671–675.

[41] V. Sideridis, A. Zacharakis, G. Tzagkarakis, and M. Papadopouli, “Gesturekeeper: Gesture recognition for controlling devices in iot environments”, in *2019 27th European Signal Processing Conference (EUSIPCO)*, 2019, pp. 1–5.

[42] A. Sultana, S. Fatima, H. Mubeen, R. Begum, K. Sohelrana, and A. Jameel, “A review on smart iot based gesture controlled grass cutting vehicle”, in *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*, 2020, pp. 440–444.

[43] T. M. Niamat Ullah Akhund, W. B. Jyoty, M. A. B. Siddik, N. T. Newaz, S. A. Al Wahid, and M. M. Sarker, “Iot based low-cost robotic agent design for disabled and covid-19 virus affected people”, in *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, 2020, pp. 23–26.

[44] S. Wang, Y. Yan, H. Chu, *et al.*, “Hand gesture recognition using ir-uwb radar with spiking neural networks”, in *2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2022, pp. 423–426.

[45] A. Bahador, M. Yousefi, M. Marashi, and O. Bahador, “High accurate lightweight deep learning method for gesture recognition based on surface electromyography”, *Computer Methods and Programs in Biomedicine*, vol. 195, p. 105 643, 2020, ISSN: 0169-2607.

[46] G. Yuan, X. Liu, Q. Yan, S. Qiao, Z. Wang, and L. Yuan, “Hand gesture recognition using deep feature fusion network based on wearable sensors”, *IEEE Sensors Journal*, vol. 21, no. 1, pp. 539–547, 2020.

[47] S. R. Bedico, E. M. L. Lope, E. J. L. Lope, E. B. Lunjas, A. P. D. Lustre, and R. E. Tolentino, “Gesture recognition of basketball referee violation signal by applying dynamic time warping algorithm using a wearable device”, in *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, IEEE, 2020, pp. 249–254.

[48] Z. Lin, W. Zhang, X. Deng, C. Ma, and H. Wang, "Image-based pose representation for action recognition and hand gesture recognition", in *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, 2020, pp. 532–539.

[49] R. Massa, A. Marchisio, M. Martina, and M. Shafique, "An efficient spiking neural network for recognizing gestures with a DVS camera on the loihi neuromorphic processor", *CoRR*, vol. abs/2006.09985, 2020. arXiv: `2006.09985`. [Online]. Available: `https://arxiv.org/abs/2006.09985`.

[50] K. D. Mankoff and T. A. Russo, "The kinect: A low-cost, high-resolution, short-range 3d camera", *Earth Surface Processes and Landforms*, vol. 38, no. 9, pp. 926–936, 2013.

[51] A. Sayles, A. Hooda, M. Gupta, R. Chatterjee, and E. Fernandes, "Invisible perturbations: Physical adversarial examples exploiting the rolling shutter effect", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 666–14 675.

[52] M. S. Alam, K.-C. Kwon, and N. Kim, "Implementation of a character recognition system based on finger-joint tracking using a depth camera", *IEEE Transactions on Human-Machine Systems*, vol. 51, no. 3, pp. 229–241, 2021.

[53] S. Ahmed, K. D. Kallu, S. Ahmed, and S. H. Cho, "Hand gestures recognition using radar sensors for human-computer-interaction: A review", *Remote Sensing*, vol. 13, no. 3, p. 527, 2021.

[54] J. W. Choi, X. Quan, and S. H. Cho, "Bi-directional passing people counting system based on ir-uwb radar sensors", *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 512–522, 2017.

[55] V.-H. Nguyen and J.-Y. Pyun, "Location detection and tracking of moving targets by a 2d ir-uwb radar system", *Sensors*, vol. 15, no. 3, pp. 6740–6762, 2015, ISSN: 1424-8220. [Online]. Available: `https://www.mdpi.com/1424-8220/15/3/6740`.

[56] S. Y. Kim, H. G. Han, J. W. Kim, S. Lee, and T. W. Kim, "A hand gesture recognition sensor using reflected impulses", *IEEE Sensors Journal*, vol. 17, no. 10, pp. 2975–2976, 2017.

[57] M. Tlili, M. Ben-Romdhane, A. Maalej, F. Rivet, D. Dallet, and C. Rebai, "Level-crossing adc design and evaluation methodology for normal and pathological electrocardiogram signals measurement", *Measurement*, vol. 124, pp. 413–425, 2018.

[58] M. Saeed, Q. Wang, O. Märtens, *et al.*, "Event-driven ecg classification using an open-source, lc-adc based non-uniformly sampled dataset", in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, IEEE, 2021, pp. 1–5.

[59] S. M. Qaisar and S. F. Hussain, "Effective epileptic seizure detection by using level-crossing eeg sampling sub-bands statistical features selection and machine learning for mobile healthcare", *Computer Methods and Programs in Biomedicine*, vol. 203, p. 106 034, 2021.

[60] S. F. Hussain and S. M. Qaiser, "Epileptic seizure classification using level-crossing eeg sampling and ensemble of sub-problems classifier", *Expert Systems with Applications*, p. 116 356, 2021.

[61] N. S. Artan, "Signal-adaptive analog-to-digital converters for ulp wearable and implantable medical devices: A survey", in *Design and Modeling of Low Power VLSI Systems*, IGI Global, 2016, pp. 199–228.

[62] A. Goldberger, L. Amaral, L. Glass, *et al.*, "Physiobank, physiotoolkit, and physionet : Components of a new research resource for complex physiologic signals", *Circulation*, vol. 101, E215–20, Jul. 2000.

[63] A. Elboushaki, R. Hannane, K. Afdel, and L. Koutti, "Multid-cnn: A multi dimensional feature learning approach based on deep convolutional networks for gesture recognition in rgb-d image sequences", *Expert Systems with Applications*, vol. 139, p. 112 829, 2020.

[64] S. Hochreiter and J. Schmidhuber, “Long short-term memory”, *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[65] A. Ikram and Y. Liu, “Skeleton based dynamic hand gesture recognition using lstm and cnn”, in *2020 2nd International Conference on Image Processing and Machine Vision*, ser. IPMV 2020, Bangkok, Thailand: Association for Computing Machinery, 2020, pp. 63–68, ISBN: 9781450388412.

[66] S. Singh, A. Sarma, S. Lu, A. Sengupta, V. Narayanan, and C. R. Das, “Gesture-snn: Co-optimizing accuracy, latency and energy of snns for neuromorphic vision sensors”, in *2021 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, 2021, pp. 1–6.

[67] I. J. Tsang, F. Corradi, M. Sifalakis, W. Van Leekwijck, and S. Latré, “Radar-based hand gesture recognition using spiking neural networks”, *Electronics*, vol. 10, no. 12, p. 1405, 2021.

[68] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. Maida, “Deep learning in spiking neural networks”, *Neural Networks*, vol. 111, pp. 47–63, 2019.

[69] S. Wang, J. Song, J. Lien, I. Poupyrev, and O. Hilliges, “Interacting with soli: Exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum”, in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, 2016, pp. 851–860.

[70] N. Ravanshad and H. Rezaee-Dehsorkh, “Chapter 12 - level-crossing sampling: Principles, circuits, and processing for healthcare applications”, in *Compressive Sensing in Healthcare*, ser. Advances in ubiquitous sensing applications for healthcare, M. Khosravy, N. Dey, and C. A. Duque, Eds., Academic Press, 2020, pp. 223–246.

[71] J. Van Assche and G. Gielen, “Power efficiency comparison of event-driven and fixed-rate signal conversion and compression for biomedical applications”, *IEEE Transactions on Biomedical Circuits and Systems*, vol. 14, no. 4, pp. 746–756, 2020.

[72] Q. Zhou, Y. Shi, Z. Xu, R. Qu, and G. Xu, “Classifying melanoma skin lesions using convolutional spiking neural networks with unsupervised stdp learning rule”, *IEEE Access*, vol. 8, pp. 101 309–101 319, 2020.

[73] G. Srinivasan, C. Lee, A. Sengupta, P. Panda, S. S. Sarwar, and K. Roy, “Training deep spiking neural networks for energy-efficient neuromorphic computing”, in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8549–8553.

[74] J. Stuijt, M. Sifalakis, A. Yousefzadeh, and F. Corradi, “μBrain: An event-driven and fully synthesizable architecture for spiking neural networks”, *Frontiers in neuroscience*, vol. 15, p. 538, 2021.

[75] E. Ceolini, C. Frenkel, S. B. Shrestha, *et al.*, “Hand-gesture recognition based on emg and event-based camera sensor fusion: A benchmark in neuromorphic computing”, *Frontiers in Neuroscience*, vol. 14, p. 637, 2020.

[76] S. Zhou, Y. Chen, X. Li, and A. Sanyal, “Deep scnn-based real-time object detection for self-driving vehicles using lidar temporal data”, *IEEE Access*, vol. 8, pp. 76 903–76 912, 2020.

[77] Y. Yan, H. Chu, X. Chen, *et al.*, “Graph-based spatio-temporal backpropagation for training spiking neural networks”, in *2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2021, pp. 1–4.

[78] Y. Wu, L. Deng, G. Li, J. Zhu, and L. Shi, “Spatio-temporal backpropagation for training high-performance spiking neural networks”, *Frontiers in neuroscience*, vol. 12, p. 331, 2018.

[79] Z. Yang and X. Zheng, “Hand gesture recognition based on trajectories features and computation-efficient reused lstm network”, *IEEE Sensors Journal*, vol. 21, no. 15, pp. 16 945–16 960, 2021.

[80] Y. Kim and B. Toomajian, “Hand gesture recognition using micro-doppler signatures with convolutional neural network”, *IEEE Access*, vol. 4, pp. 7125–7130, 2016.

[81] Y. Sun, T. Fei, S. Gao, and N. Pohl, “Automatic radar-based gesture detection and classification via a region-based deep convolutional neural network”, in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 4300–4304.

Academic achievements during the master's degree

Three accepted international conference papers:

1. **Wang Shule**; Carmen Martínez Almansa; Jorge Peña Queralta; Zhuo Zou; Tomi Westerlund. UWB-Based Localization for Multi-UAV Systems and Collaborative Heterogeneous Multi-Robot Systems[J/OL]. Procedia Computer Science, 2020, 175: 357-364.

2. **Shule Wang**; Yulong Yan; Haoming Chu; Guangxi Hu; Zhi Zhang; Zhuo Zou; Lirong Zheng. Hand gesture recognition using ir-uwb radar with spiking neural networks", in 2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS), 2022, pp. 423–426.

3. Carmen Martínez Almansa; **Wang Shule**; Jorge Peña Queralta; Tomi Westerlund. Autocalibration of a Mobile UWB Localization System for Ad-Hoc Multi-Robot Deployments in GNSS-Denied Environments[J/OL]. International Conference on Localization and GNSS (ICL-GNSS 2020).

Acknowledgement

Foremost, I would like to express my sincere gratitude to my advisors Professor Tomi Westerlund and my thesis examiner Paola Torrico Morón, whose sincerity and encouragement I will never forget. They were always there when I needed help and always managed to inspire me with their immense knowledge and motivation.

Many thanks go to the seniors Jorge Peña Queralta, Chu Haoming, and Yan Yulong, who directly and indirectly provided me with inspiration and valuable suggestions during the work of this thesis. Without their help, it would have been very difficult to design the algorithms and experiment setups I needed.

I thank my fellow students for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. I still remember the days when we were in Finland. The epidemic was raging abroad, and their solidarity and encouragement made me no longer afraid. Then, I owe my deepest gratitude to my boyfriend. I am forever thankful for the unconditional love and support throughout the entire thesis process and every day.

Last but not least, I would like to thank my parents for their strong support economically as well as regular encouragement in every step to make me in the present stage. They have provided me with the support I needed when things were falling apart. Similarly, other relatives are also subjects to special thanks for their inspiration and cooperation in my study. They were always there for me right to the end.