**Coventry University**

**Coventry University**

**DOCTOR OF PHILOSOPHY**

**Localisation and Tracking of Acoustic Sources in Augmented Space and 3D Objects for Human-Computer Interaction**

You, Hongyu

*Award date:*
2023

*Awarding institution:*
Coventry University

[Link to publication](#)

Coventry University

Faculty of Engineering, Environment and Computing

# Localisation and Tracking of Acoustic Sources in Augmented Space and 3D Objects for Human-Computer Interaction

Coventry University

By

Hongyu You

PhD thesis

June 2022

# Localisation and Tracking of Acoustic Sources in Augmented Space and 3D Objects for Human-Computer Interaction

*A thesis submitted in partial fulfilment of the University's requirements for the Degree of Doctor of Philosophy*

June 2023

**Coventry University**

# Certificate of Ethical Approval

Applicant:

Hongyu You

Project Title:

Localisation and Tracking of Acoustic Sources in Augmented Space and 3d Objects for Human-Computer Interaction

This is to certify that the above named applicant has completed the Coventry University Ethical Approval process and their project has been confirmed and approved as Low Risk

Date of approval:

20 June 2018

Project Reference Number:

P62582

# Abstract

Acoustic localisation technology promoted by modern engineering is of significant meaning commercially, militarily and medically. Commercially, Human-Computer Interfaces (HCI) powered by the acoustic localisation technology are attractive to consumers. Militarily, modern Sound Navigation and Ranging (SONAR) systems are essential equipment for the Navy and the acoustic localisation technology is one of the core technologies in SONAR systems. Medically, acoustic localisation technology is used to locate and smash stones inside human bodies. Acoustic localisation technology has a series of applications in various fields. However, the indoor localisation is an exception.

The localisation demands of individuals in indoor environments are rising. For example, the Virtual Reality (VR) technology allows users to explore virtual worlds, but the VR technology also brings challenges to human-computer interaction. Traditional Time Difference of Arrival-based electromagnetic localisation technologies and touchtone HCIs cannot provide users with immersive user experiences. However, HCIs powered by the acoustic localisation technology can locate users passively and such HCIs may improve the user experience by removing mobile restrictions of touchtone HCIs and wearable devices.

At present, two approaches are applicable to achieve acoustic localisation. The first approach is the time difference localisation. The time difference acoustic localisation technology locates an acoustic source according to the time differences measured by sensors located at multiple locations. The Time Difference of Arrival (TDOA) is the most representative time difference localisation methodology. However, TDOA-based localisation technologies are inaccurate in short-range scenarios due to the interferences caused by the multipath effect. The second approach is the Template Pattern Matching (TPM) localisation. It is a similarity-based matching methodology developed for string examination in computer science originally. Regarding localisation, TPM-based acoustic localisation technologies locate acoustic sources by matching the acoustic signals of the acoustic sources with the template acoustic signals. The template acoustic signals are previously collected from pre-defined locations. In template

signal matching, the template acoustic signal which has the highest similarity to the input acoustic signal is identified as the matched signal thus the acoustic source is located at the pre-defined location of the matched template signal. The TPM-based acoustic localisation system has a reasonable system cost since it does not rely on the deployment of multiple sensors and system synchronisation in comparison to the TDOA-based localisation system. But TPM-based localisation technologies need adequate data and exact matching algorithms to ensure accurate matching results. A representative application of TPM is the passive SONAR system on submarines. This project aims to achieve three-dimensional short-range acoustic source localisation with the TPM approach.

The short-range acoustic localisation technology has been applied to two-dimensional human-computer interaction but has not been applied to three-dimensional human-computer interaction. Since the short-range localisation technology is a key enabling technology, current research on three-dimensional human-computer interaction tends to realise three-dimensional human-computer interaction with the optimised TDOA-based electromagnetic localisation technology. However, specific problems such as the deployment of sensors and the high system cost are exposed in the implementations. These disadvantages restrict the applicability of the short-range localisation technology; thus, an acoustic localisation technology is developed in this project to overcome the disadvantages of the TDOA-based electromagnetic localisation technology and enhance the applicability of the short-range localisation technology.

Acoustic waves can be generated actively by speaking, knocking and tapping therefore acoustic sources are ideal signal sources for passive localisation. In this thesis, a passive pattern matching-based acoustic localisation technology - Location Template-based Positioning Model (LTPM) is successfully designed, implemented and tested in terms of three-dimensional human-computer interaction. The proposed technology determines the coordinates of acoustic sources by matching input acoustic signals with pre-collected template signals according to 43 acoustic features and 17 signal images. The test results indicate that LTPM has achieved a three-dimensional accuracy of 173 mm in 95% of the location estimates in an indoor environment.

LTPM successfully utilises the acoustic multipath effect to achieve three-dimensional acoustic localisation and LTPM does not rely on the deployment of sensor arrays, measurement of time difference of arrival, path optimisations and signal filtering in comparison to the TDOA-based positioning system. Meanwhile, the LTPM-based positioning system has robust environmental adaptability. Correspondingly, the positioning performance of LTPM depends on the data volume, the integrated features, and the matching accuracy of the algorithm.

In this thesis, LTPM is applied to two-dimensional surfaces and an indoor three-dimensional space for localisation tests. The two-dimensional LTPM-based localisation system has achieved a two-dimensional accuracy of 30 mm in 80% of the location estimates. While the three-dimensional LTPM-based localisation system has achieved a three-dimensional accuracy of 173 mm in 98% of the location estimates. These results imply that three-dimensional acoustic localisation in indoor environments is feasible, and LTPM has the potential to be the enabling technology for three-dimensional human-computer interfaces.

# Acknowledgements

# Table of Contents

# List of Figures

# List of Tables

# 1    Introduction

Localisation technology has various commercial and military applications. For example, geological localisation of the earthquake sources; the localisation of air bubbles and cracks within metal materials in Non-Destructive Tests (NDT); electromagnetic and acoustic localisation of military targets such as aircraft, missiles, ships and submarines. Meanwhile, localisation technology is also a critical enabling technology for human-computer interaction.

In two-dimensional applications, touch screens such as capacitive and resistive screens utilise TDOA-based localisation technologies to locate the user's fingers. In three-dimensional applications such as Virtual Reality (VR) and Augmented Reality (AR), the user's limbs, head and body are located with the short-range localisation technology. Traditional TDOA-based electromagnetic localisation and inertial localisation technologies have many limitations and shortcomings, such as mobile restriction, system installation restrictions and constant requirements for displacement calibrations and path optimisations. As a result, the user experience is poor. New three-dimensional localisation technologies are therefore needed to enhance the user experience.

Since conventional electromagnetic localisation technologies can hardly be applied to short-range indoor environments due to the expensive system cost and the low localisation accuracy. Novel short-range localisation technologies represented by the Ultra-WideBand (UWB) localisation technology and the acoustic localisation technology have been developed since the 2000s. The UWB is an extension of the remote electromagnetic localisation technology but optimised explicitly for indoor localisation [1]. Acoustic localisation technology, on the other hand, has many technical advantages in comparison to electromagnetic localisation technology. For example, the acoustic Natural User Interface (NUI) can locate the user by collecting the user's speech, tapping and knocking. Since the signal carrier is the acoustic wave, users no longer need to wear electromagnetic transceiver devices so the cost of the system is reduced and the user experience is improved.

The TDOA-based acoustic localisation technology and the TPM-based acoustic localisation technology are the two mainstream acoustic localisation technologies. The TDOA-based acoustic localisation technology has been applied to two-dimensional surfaces and achieved millimetre-level localisation accuracy [2]. But the three-dimensional positioning performance of the acoustic localisation technology regarding human-computer interaction has yet to be discovered. In this thesis, a Template Pattern Matching (TPM)-based acoustic localisation technology is proposed, tested, and discussed, especially from the human-computer interaction perspective.

## 1.1 Background introduction

Acoustic waves have many applications, as shown in Figure 1.1-1. Localisation is one of the most representative applications. The general principles of acoustic localisation originated from biology and bionics. In the early 18th century, Lazzaro Spallanzani discovered that bats have the superpower to fly at night without hitting obstacles. Later research revealed that bats utilise echoes to locate barriers and take evading actions accordingly [3]. The discovery of echo localisation attracted scholars' attention, and systematic studies of acoustic localisation began in the early 1900s.
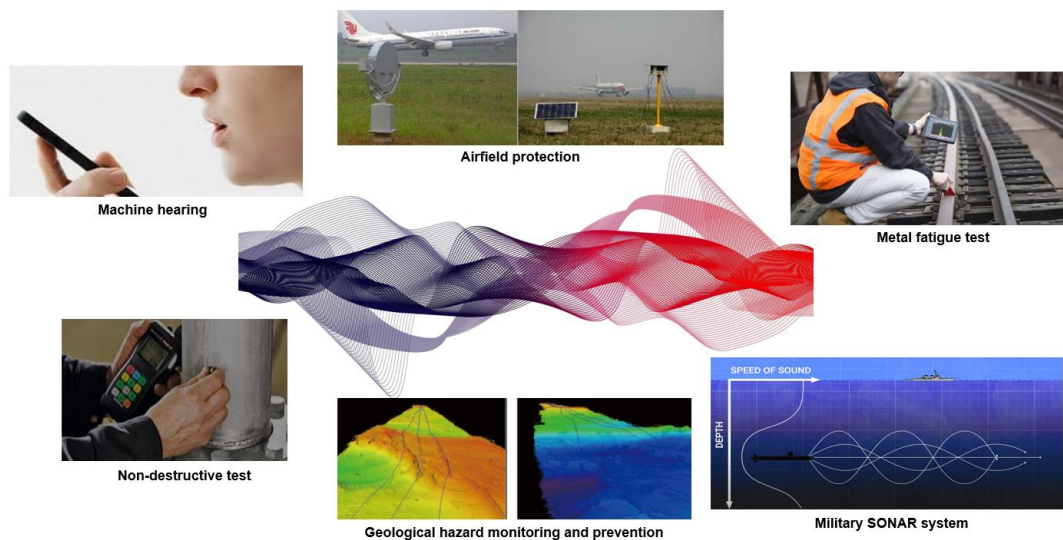


Figure 1.1-1 Various applications of the acoustic wave. Acoustic waves have been applied to multiple areas, and one of the most critical applications is the underwater localisation system - Sound Navigation and Ranging (SONAR)

The Royal Navy realised the importance of underwater detection after the famous RMS Titanic sank [4]. Back then, the electromagnetic wave had been utilised to locate airborne

2

targets. However, the strength of the electromagnetic wave attenuates rapidly in underwater environments; a new signal carrier is therefore needed for underwater localisation.

Unlike the electromagnetic wave, mechanical waves, such as the acoustic wave, has a low attenuation rate in liquids. Hence, the acoustic wave was applied to underwater localisation with the support from the British Royal Navy. Soon afterwards, the first applicable Sound Navigation Ranging (SONAR) system was built and tested successfully [5].

SONAR systems are one of the most representative engineering products. The SONAR technology combines bionics and electronics with acoustics to achieve tasks such as underwater detection, underwater localisation and underwater communication. There are two types of SONAR systems: the active SONAR system and the passive SONAR system [5]. An active SONAR system is similar to a Radio Detection and Ranging (RADAR) system, which locates the target by measuring time differences between emitted signals and the signals reflected by the target. On the contrary, a passive SONAR system compares acoustic signals generated by the target with pre-collected template signals to identify and locate the target. For a passive SONAR system, the environmental parameters such as water temperature, water depth and velocity profile of acoustic waves have less impact on the positioning performance since the performance depends on the pre-collected data and the classification performance of matching algorithms [5]. Another advantage of the passive SONAR system is that the system structure of passive SONAR systems is relatively simple. The localisation technology proposed in this thesis is also developed according to the passive SONAR system.

Institutions and universities have launched research on indoor localisation since the 2000s. Current localisation technologies consist of the Time Difference of Arrival (TDOA)-based localisation technologies and the Template Pattern Matching (TPM)-based localisation technologies [6]. The TDOA-based acoustic localisation technology is similar to the active SONAR system, while the TPM-based acoustic localisation technology is similar to the passive SONAR system.

The EU project, Tangible Acoustic Interfaces for Computer and Human Interaction (TAI-CHI), has achieved the localisation of a moving acoustic source and the localisation of multiple acoustic sources on two-dimensional surfaces with the TDOA-based localisation algorithm [7]. On the other hand, Microsoft has developed a TPM-based electromagnetic localisation system which utilises the signal strength and the Euclidean distance to locate mobile devices and achieved metre-level (range error from 1 m to 14 m) resolution [8]. Researchers from the University of Bristol have created an ultrasonic feedback system, allowing users to physically touch the three-dimensional shape formed by ultrasonic waves [9]. These novel acoustic wave-based technologies revealed the application potential of acoustic localisation in human-computer interaction.

There is a research gap in the study of three-dimensional acoustic localisation since existing research on acoustic localisation is two-dimensional-based, while the research on three-dimensional localisation is electromagnetic wave-based. Therefore, three-dimensional acoustic localisation is determined as the primary research aim of this project.

Conventional localisation technologies have low localisation accuracy and resolution in short-range scenarios due to the complex multipath effect and the fast electromagnetic propagation speed. Thus, high-performance sampling systems, multi-sensor arrays and path optimisations are essential to maintain a high localisation accuracy and a high indoor resolution. Still, these solutions increase the system cost and the complexity of the localisation system [10]. Hence, a new acoustic wave-based localisation technology is developed in this project to address these issues.

The requirements for the new localisation technology are summarised as follows:

- The localisation system which utilises the proposed acoustic localisation technology should reach a 500 mm-level accuracy in 80%-90% of the location estimates.
- The localisation system equipped with the proposed localisation technology should have excellent adaptability to complex indoor environments while reducing system cost and system complexity.

Acoustic localisation brings a turning point to short-range localisation. However, from the literature review, current acoustic wave-based research led by universities and associations lies in speech recognition, acoustic field effects and realistic acoustic transmission model. Research on short-range acoustic localisation is extremely finite. Therefore, this research aims to propose a TPM-based short-range acoustic localisation technology. Technically, the proposed acoustic localisation technology comprises acoustic signal processing, feature engineering, TPM and machine learning to reach the following targets:

- Validate the feasibility of acoustic source localisation in a closed three-dimensional space.

- Overcome the drawbacks of traditional TDOA-based localisation technology while increasing system adaptability to complex indoor environments.

- Evaluate the performance of machine learning algorithms in acoustic signal classification and summarise the advantages and disadvantages of different machine learning algorithms in short-range acoustic localisation.

To ensure the listed targets are fully covered, localisation technologies, machine learning algorithms, noise elimination technologies and signal analysis are studied. In terms of system construction, skills and techniques such as circuit design, circuit welding, signal processing, application programming interface, and algorithm compilation are also mastered and studied for system implementations.

## 1.2   Aim and objectives

This research aims to develop a three-dimensional short-range acoustic localisation technology to support natural three-dimensional human-computer interaction. More specifically, to achieve three-dimensional short-range acoustic source localisation with signal processing, feature engineering and machine learning algorithms.

The key objectives are as follows:

(1) Develop a three-dimensional short-range acoustic source localisation technology.

(2) Design and build corresponding acoustic localisation systems.

5

(3) Validate the proposed technology and evaluate its performance.

(4) Summarise the advantages and disadvantages of the proposed localisation technology and analyse potential applications.

(5) Advance the short-range localisation research progress by providing an overview of the acoustic localisation technology.

## 1.3   Research challenges, research novelty and contributions

*1.3.1 Research challenges*

The EU project, Tangible Acoustic Interfaces for Computer-Human Interaction (TAI-CHI), has achieved two-dimensional acoustic localisation on solid surfaces [7]. Snapdragon has also developed sound ID scanning technology for smartphones [11]. These technologies hint that the acoustic wave is an ideal signal source for short-range localisation. However, current acoustic localisation technologies are TDOA-based. These technologies are only applicable in solid media and are difficult to be applied to indoor environments fulfilled with air; thus, the primary challenge is:

- Overcome the drawbacks of TDOA-based electromagnetic/acoustic localisation technology with the TPM-based acoustic localisation technology.

Therefore, a TPM-based acoustic localisation technology is proposed and implemented. Detailed research challenges encountered during the design of the localisation technology and implementation are introduced below.

The management of the multipath effect is the first research question. An acoustic wave interacts with objects inside the medium and multiple acoustic waves are generated during the propagation of the acoustic wave. Hence, a received acoustic signal usually contains direct signal components and multiple indirect signal components. The phenomenon is called the acoustic multipath effect [12]. The multipath effect complicates the solution to short-range localisation, as shown in Figure 1.3-1.

Specific technical problems caused by the multipath effect, such as source signal identification, signal filtering and signal separation, are exposed during the development of short-range localisation technologies. Noise elimination techniques (such as adaptive,

digital, and predictive filters), acoustic field simulation and high-order acoustic wave equations are developed against these problems, and these techniques have only finitely improved the accuracy of localisation systems.

The multipath effect becomes more complex indoors [12]. Therefore, a key research challenge in acoustic localisation is:

Eliminate or utilise the multipath effect.

In this project, the multipath effect is utilised to achieve localisation instead of being eliminated because:

- The elimination of the multipath effect is impracticable and complicated.
- The multipath effect endows acoustic signals with unique signal patterns which can be utilised for matching.

The multipath effect is essential to the proposed technology - Location Template-based Positioning Model (LTPM). Theoretically, the more complicated the multipath effect is, the better the localisation results will be. In comparison to TDOA-based electromagnetic localisation technologies, the system cost of LTPM is lower because LTPM does not rely on the sensor array deployment, path optimisations and high-frequency sampling systems and LTPM has robust environmental adaptability.

Two factors determine the performance of LTPM; the first factor is the signal feature (match target). The second factor is the classification performance of the matching algorithm. In the previous research on short-range localisation, the most common signal features used for matching are signal strength and frequency shift. It is clear that the signal is not fully utilised; thus, the first challenge in applying TPM to acoustic localisation is:

- Improve the acoustic signal utilisation. Explore, define and add more acoustic signal features to improve the performance of LTPM.

The second challenge in applying TPM to acoustic localisation is:

- Compile a pattern matching algorithm that accurately matches the input signal with the template signals according to the defined acoustic signal features.

Moreover, acoustic signals received by a sensor are always a mixture of the source signal components, the multipath signal components, and the background noise. Therefore, noise processing is also necessary and another technical criterion of the proposed localisation technology is eliminating the noise while preserving the applicable signal components, especially the multipath components in a received signal.
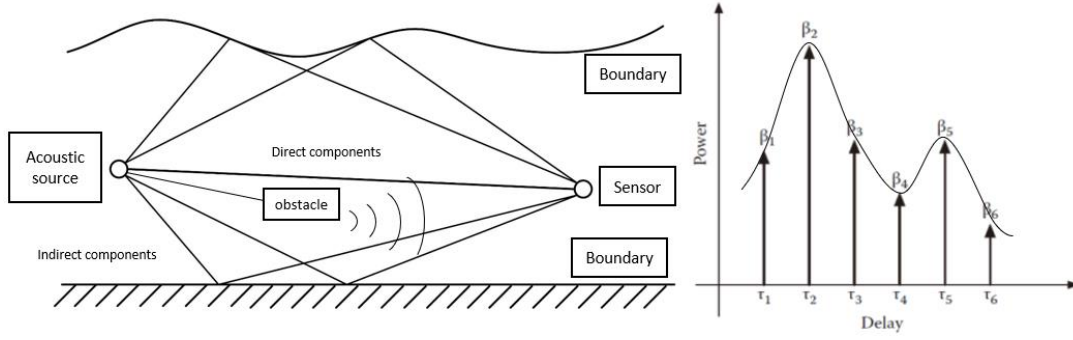


Figure 1.3-1 Illustration of the multipath effect (left) and the power distribution of a received acoustic signal (right). 6 signal components (corresponding to sets {$\beta1$, $\beta2$, $\beta3$, $\beta4$, $\beta5$, $\beta6$} and {$\tau1$, $\tau2$, $\tau3$, $\tau4$, $\tau5$, $\tau6$} respectively) are sampled, including both direct signal components and indirect signal components

### 1.3.2 Novelty and contribution

In the secondary research, 62% of the literature is about electromagnetic localisation, 12% is about TPM localisation, and 15% is about two-dimensional acoustic localisation. Most localisation technologies and related research are electromagnetic and TDOA based. There is only limited literature on three-dimensional acoustic localisation.

In this project, a localisation technology - Location Template-based Positioning Model (LTPM) is designed and tested, with the aim of providing human-computer interaction with a reliable and accurate three-dimensional acoustic localisation technology. Signal processing, feature engineering and template pattern matching are utilised to achieve acoustic source localisation in the indoor environment. Machine learning has also been applied to recognise the complex acoustic patterns hidden in template signals and classify acoustic signals accordingly.

The proposed Location Template-based Positioning Model (LTPM) utilises the acoustic multipath effect and it has a three-dimensional accuracy of 173 mm. LTPM overcomes

the drawbacks of TDOA localisation technologies as it does not rely on time difference measurement and path optimisations. Meanwhile, technical solutions to challenges listed in 1.3.1 are presented.

The research on short-range acoustic localisation is conducted from the human-computer interaction perspective as this research intends to find a better localisation solution for three-dimensional human-computer interaction. The proposed acoustic localisation technology can be integrated into applications such as Augmented Reality (AR), Virtual Reality (VR), natural user interface, human-computer interaction in autonomous driving, speech localisation, smart stage, smart home, and battlefield sniper localisation as an enabling acoustic localisation technology

## 1.4    Scope and thesis structure

The research background, research challenges and research objectives are described in Chapter 1. A state-of-the-art literature review considering existing localisation technologies, human-computer interaction technologies, machine learning and signal processing techniques is presented in Chapter 2. While in Chapter 3, the spiral research methodology is discussed.The localisation principles of LTPM, acoustic signal features for matching and machine learning algorithms are introduced in Chapter 4.

In Chapter 5, two-dimensional localisation tests are designed to preliminarily test the positioning performance of LTPM. Detailed test results are presented and discussed.

In Chapter 6, the LTPM is applied to achieve three-dimensional localisation. Details on devices, implementations and signal processing are illustrated and elaborated. The test results are analysed and discussed.

Chapter 7 presents the research conclusions. A review of accomplished work, the technical defects of LTPM, suggestions and possible applications are summarised to establish a solid foundation for subsequent short-range localisation research.

# 2    Literature review

In this chapter, technical details of the three-dimensional acoustic source localisation technology are determined by reviewing acoustic properties, mainstream human-computer interaction technologies, existing localisation technologies and machine learning technologies.

## 2.1    A brief review of acoustics and acoustic localisation

Modern acoustics consists of classic acoustics, modern acoustics, architectural acoustics, hydroacoustic, electroacoustics, biomedical acoustics, and musical acoustics. The review begins with the introduction of acoustic decomposition and superposition in classic acoustics.

### 2.1.1 Classic acoustics and decomposition of mechanical waves

The exploration of acoustic localisation starts with the vibration theory. The string motion theory proposed by Galileo Galilei is widely acknowledged. Galileo Galilei was one of the founders of acoustics. He started formal research on the relationship between vibration and acoustic waves in the 17th century [13]. His writing "Two New Sciences" first clarified the word vibrating strings or frequency.

Frequency represents the number of occurrences of a periodic event per second. It is a physical quantity often used to describe periodic movements. In signal processing, the frequency domain provides researchers with a new approach to analysing acoustic signals. Some problems that cannot be resolved in the time domain can be easily addressed in the frequency domain.

The frequency domain analysis has become an essential part of signal processing. In this project, features of acoustic signals in the frequency domain are utilised to realise accurate signal matching. The frequency $f$ is academically defined as:

$$f = \frac{1}{T} \hspace{4cm} 2-1$$

where $T$ is the time to perform a periodic change.

From then on, scholars and musicians connected frequency with acoustic terms such as tone, pitch and timbre to exploit the correlations between frequency and auditory sensory. The famous British mathematician Brook Taylor, inventor of the **Infinite Series**, successfully deduced the strict preliminary solution of the one-dimensional wave equation (Eq. 2-2) for single-string vibration with Newton's motion equation [14].

$$\frac{\partial^2 u}{\partial t^2} = a^2 \frac{\partial^2 u}{\partial x^2}, a = \sqrt{\frac{T_E}{\rho}} \qquad\qquad 2-2$$

where $a$ reflects the propagation speed of the acoustic wave, which is determined by the string, $u$ is the lateral displacement of the mass point at $x$ on the string relative to the equilibrium position at time $t$, $T_E$ is the tension on the string, and $\rho$ is the density of the material.

The wave equation is an important model for the vibration analysis of elastomers. It provides an intuitive research basis for developing the Fourier Transform since the one-dimensional wave equation represents the simplest string vibration. In 1822, the Fourier Series was proposed by Fourier [15], and it was applied to partial differential equations immediately because the Fourier series transfers the acoustic wave in the time domain into plural sinusoidal or cosine waves with different frequencies.

Fourier series of the periodic function in sinusoidal form:

$$f_T(x) = c_0 + \sum_{n=1}^{\infty} c_n \sin\left(n\frac{2\pi}{T}x + \theta_n\right) \qquad\qquad 2-3$$

where $c_0$ is the direct current component, $c_n$ stands for the amplitude (the weights of different sinusoidal components), $\frac{2\pi}{T}$ is the fundamental frequency while $\theta_n$ represents the initial phase of different sinusoidal components.

However, the famous electrician Ohm pointed out that most acoustic signals are aperiodic. And he solved the aperiodic decomposition problem with differentiation [16]. An aperiodic function is considered periodic when $T \to +\infty$. The fundamental frequency $\frac{2\pi}{T}$ is turned into a differentiation: $d\omega$. Meanwhile, the sum operation in (Eq.

11

2-3) becomes integral. The Fourier transformation formula is therefore proposed as follows:

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F(\omega) \cdot e^{i2\pi\omega x} d\omega \qquad\qquad 2-4$$

where $f(x)$ is the aperiodic function. $\omega$ is the reciprocal of $T$.

Nowadays, the power spectrum is calculated by projecting a signal into the frequency domain through the Fourier Transform. The Fourier Transform converts realistic, complex acoustic waves into periodic waves with single frequencies, presenting an intuitive relationship between power and frequency. Thus, the Fourier Transform is a powerful mathematical tool, and it has been used to find scientific solutions to complex multi-frequency vibration questions.

A realistic acoustic wave consists of one fundamental wave and many harmonic waves, these waves work together to determine the physical properties of the acoustic wave [17]. (Eq. 2-4) is widely used to extract an acoustic signal's fundamental frequency and harmonic frequencies. Besides, Ohm also illustrated that the human brain could decode mixed acoustic waves to extract information about acoustic sources [16]. These research achievements promoted the development of modern acoustic signal processing profoundly.

In addition, the inverse Fourier Transform is always used to restore a signal in the frequency domain to a signal in the time domain. The general formula for the inverse Fourier Transform is as follows:

$$F(\omega) = \int_{-\infty}^{+\infty} e^{-i2\pi\omega x} \cdot f(x) dx \qquad\qquad 2-5$$

(Eq. 2-4) and (Eq. 2-5) are the Fourier Transform pair. The appearance of the Fourier Transform pair allows signals to be converted freely between the time domain and the frequency domain. The Fourier Transform pair is used for signal filtering and feature extraction in this research.

According to the Fourier Transform, an acoustic wave is generated by linear

superpositioning a series of periodic acoustic waves with different frequencies. Therefore, if acoustic source localisation with a single-frequency acoustic wave is feasible, realistic acoustic signals with multiple frequency bands can also be decomposed and utilised to achieve acoustic source localisation.

Based on the above conclusion, this project focuses on processing audible acoustic signals with single frequencies. This strategy simplifies the design of localisation tests and enables deep mining of acoustic signal features.

From then on, Helmholtz validated Ohm's research achievements with the Fourier Series [18]. Helmholtz also proposed the resonance theory in his famous tone-listening experiment [18]. Nowadays, the Helmholtz resonator is still an essential tool for instrument design and modern acoustic engineering. Figure 2.1-1 shows the Helmholtz resonators.

Resonance occurs when a periodic acoustic wave enters a closed space. Resulting in a series of resonant frequencies called formant frequencies in signal processing [19]. This project utilises the resonance feature-formant band to achieve acoustic source localisation.



Figure 2.1-1 Helmholtz resonators. The resonators capture and amplify specific mechanical waves. The Helmholtz resonance theory was applied widely from internal-combustion engines to aircraft engines [20]

By the end of the 18th century, the research priority of acoustics inclined to the three-dimensional and analytical solution of the wave equation. French mathematician Poisson

presented solutions in 1820 [21]. Poisson researched the reflection of the mechanical wave; and the acoustic penetration between two different liquid media. In his research 'regarding acoustic wave propagation in open pipes', he claimed that the sound pressure could change significantly at the cross-section of pipes [22]. Considering that such a phenomenon could happen during the propagation of acoustic waves, the sound pressure of the acoustic signal is extracted as a signal feature for pattern matching in this research.

In 1866, German physicist Kundt gave birth to the standing wave tube (or Kundt tube, as shown in Figure 2.1-2) for sound absorption coefficient measurement [23]. The standing wave tube has become a standard method for measuring the material's acoustic impedance. The acoustic absorption coefficients of the composite table and the glass plate are also measured with the standing wave tube in this project.
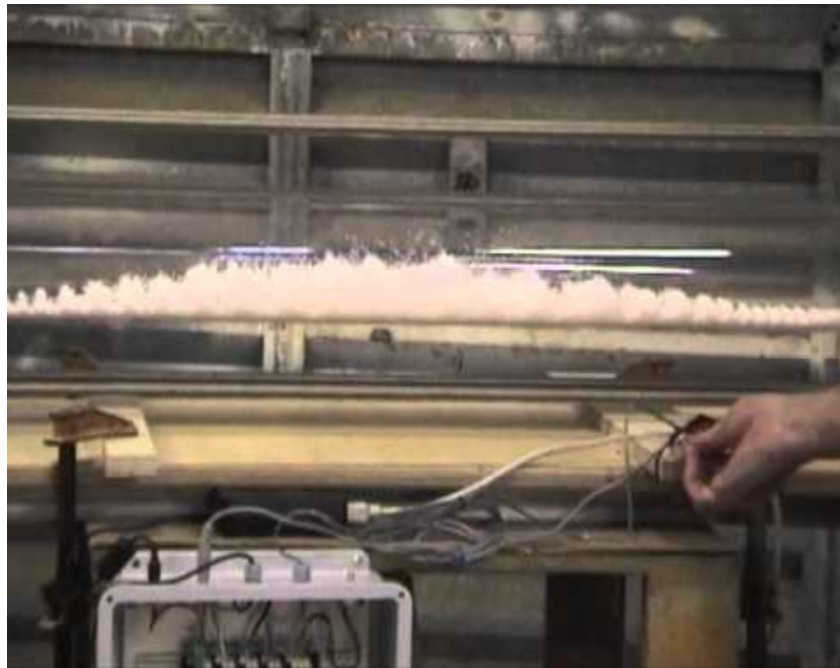


Figure 2.1-2 The Kundt tube is widely used to display mechanical waves visually. On the other hand, the tube is often used to measure the acoustic propagation velocity in solid objects [23]

In order to achieve two-dimensional acoustic localisation and three-dimensional acoustic localisation, it is important to understand the characteristics of acoustic propagation in solids, liquids, and gases. Ernst Chladni invented the Chladni plate to demonstrate the propagation patterns of acoustic waves inside solid objects: shear and longitudinal waves [24]. The shear wave originates from shear stress, while the shear stress in liquids and

14

gases is approximately zero; thus, acoustic waves propagate only in the form of longitudinal waves in liquids and gases. This character of acoustic waves simplifies two-dimensional acoustic localisation because signal components are complex and well-preserved inside solid objects. Hence, the two-dimensional localisation test will be conducted first, followed by the three-dimensional localisation test.

The propagation speed of the acoustic wave is also an important parameter in existing acoustic localisation technologies, especially in TDOA-based localisation technologies. Technically, the acoustic propagation speed affects the localisation accuracy and the sampling rate of acoustic wave-based localisation systems. But since the localisation technology proposed in this project relies on pattern matching rather than time differences, acoustic velocity barely affects the performance of LTPM-based localisation systems.

In the review of classic acoustics, basic requirements for acoustic localisation have been determined. Acoustic sources with a single vibration frequency will be used in localisation tests to simplify signal processing and enable deep mining of acoustic features. In signal pattern matching, multiple signal features (match objects) will be defined first then the input signal is located by searching for a sample signal which has the most similar features to the input signal's features. The biggest difference from traditional pattern matching-based localisation technologies lies in the utilisation of multiple acoustic features.

From the perspective of the localisation test, the test follows the 'from simple to complex' principle. Localisation tests will be conducted on two-dimensional surfaces first and then proceed to three-dimensional spaces.

Although there is limited literature on acoustic localisation in classic acoustics, classical acoustics still laid the foundation for pattern matching-based localisation technologies. Next, modern acoustics are reviewed.

### 2.1.2 Architectural acoustics and the multipath effect

Architectural acoustics focuses on improving indoor acoustic quality and indoor noise

elimination [25]. Thus, acoustic field simulation, sound quality control and noise suppression techniques are essential for architectural acoustics and these techniques are also referenced to achieve acoustic localisation in this project.

The multipath effect-based reverberation theory was proposed by W. C. Sabine at the beginning of the 20th century [26]. As shown in Figure 2.1-3, the multipath effect causes interferences for signal processing in TDOA-based localisation technologies. The management of the multipath effect has become an inevitable research question since the multipath effect significantly reduces the localisation accuracy. The multipath effect is always considered a negative effect in architectural acoustics because it causes specific differences between the source signal and the received signal. The most effective and practical solution to eliminate the multipath effect is the soundproof room; as shown in Figure 2.1-4, the room maintains -20 dB of background noise level [27].
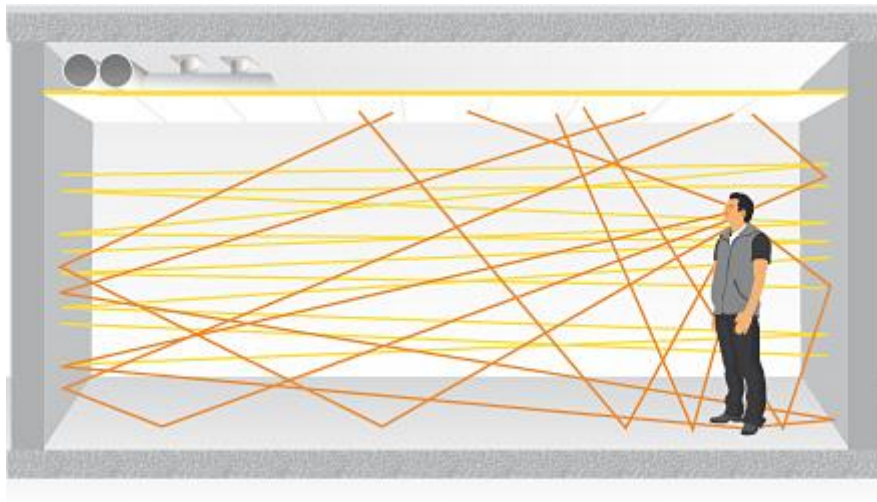


Figure 2.1-3 Illustration of the indoor acoustic multipath effect. The multipath effect becomes less obvious in the open space. While in a closed space, the multipath effect is amplified and it causes inconsistent signal perceptions [26]

The acoustic multipath effect is considered a positive effect in this project as it endows the received acoustic signals with distinguishable features for signal matching. Previous pattern matching-based electromagnetic localisation test results achieved by Ahonen and Eskelinen showed that the signal strength could be used for mid-range (200 m localisation resolution) localisation [28]. Inspired by their achievements, this project focuses on realising short-range localisation (500 mm - 200 mm localisation resolution)

with the multipath effect of audible acoustic waves.

Acoustic waves are elastic waves, and acoustic waves have different characteristics in different elastomers. For example, acoustic waves propagate simultaneously in longitudinal and transverse waves inside solid objects, whereas only longitudinal waves exist in liquids and gases. The transverse, longitudinal, and leaky surface acoustic waves in solid objects have different propagation velocities and properties. Predictably, the coexistence of different acoustic waves endows the received acoustic signal with unique features; thus, it is considered a positive effect for signal feature-based pattern matching algorithms.



Figure 2.1-4 Sound-proof room built by Massachusetts Institute of Technology [27]. The room has a noise level of 20 dB and the wall absorbs most acoustic waves; thus, the multipath effect barely exists. The room is used for sound recording, sound quality research etc.

Different from the attitude of architectural acoustics towards the multipath effect, the multipath effect is utilised to achieve localisation rather than being suppressed in this project. At the same time, sound propagation models from architectural acoustics [29] are referenced to support the simulation and modelling of indoor acoustic fields.

*2.1.3 Hydroacoustics and SONAR systems*

Acoustic waves are essential in underwater detection. Underwater acoustic localisation technology played a great role during World War II, and this technology received rapid development in military and civilian fields after the war [30]. Nowadays, the main application fields of hydroacoustics are marine exploration and the marine military.

17

D. Colldon first measured the underwater acoustic velocity in 1827 [31]. The high propagation speed of 1435m/s and low attenuation rate of the acoustic wave have aroused researchers' interest in underwater applications. In 1912, the sinking of the most famous British passenger liner RMS Titanic shocked the world. Then the British Navy developed echolocation systems for battleships and cruise ships to prevent such accidents from happening again [32]. As a result, hydroacoustic has received gorgeous development since then.

Nowadays, SONAR is no longer a mysterious word to the public, and it plays a vital role in multiple fields such as fishing, marine, and military. In 1916, P. Langevin conducted a test on underwater echolocation. In the test, he successfully received reflected acoustic waves from the seabed and a huge armour plate placed at 200 meters of depth [33]. While in 1918, he invented two key devices for SONAR; the first applicable piezoelectric transducer, which produces ultrasonic waves; and the vacuum tube amplifier for power amplification [33]. Since then, the military constantly developed SONAR systems for anti-submarine purposes during the World War. Figure 2.1-5 shows the SONAR array on a submarine and a realistic SONAR scanning map.

Multiple acoustic localisation technologies are integrated into the SONAR technology to ensure the accuracy and versatility of SONAR systems. Localisation technologies integrated into the SONAR technology include echolocation, time difference localisation, angular localisation, and beamforming localisation technologies [34].

Technically, the SONAR technology can be divided into active SONAR technology (which has acoustic wave generation devices) and passive SONAR technology (which only receives acoustic waves emitted by the target) [35]. Both of the SONAR technologies are reviewed since it is an acoustic wave-based localisation technology. In the proposal of the pattern matching-based three-dimensional acoustic localisation technology, the SONAR technology, especially the passive SONAR technology is referenced to complete the technical details of LTPM.

In the military and marine field, a passive SONAR system provides information such as

the distances, the thrust modes and the types of the target ships. Compared with the active SONAR system, the passive SONAR system has excellent concealment as it only receives acoustic waves. A passive SONAR system identifies targets by matching the acoustic signals received by hydrophones with the template signals in the matching database [35]. Therefore, the nature of the passive SONAR is consistent with the acoustic localisation technology proposed in this project. Namely, a passive acoustic localisation technology that overcomes the shortcomings of the TDOA-based localisation technologies is proposed according to the localisation principles of the passive SONAR system. Technologies integrated into the passive SONAR technology are also referenced and utilised in the proposed acoustic localisation technology. The sensor array, feature extraction and pattern matching technologies are the core technologies for a passive SONAR system [35]. These technologies are also studied and selected to achieve onshore three-dimensional acoustic source localisation.
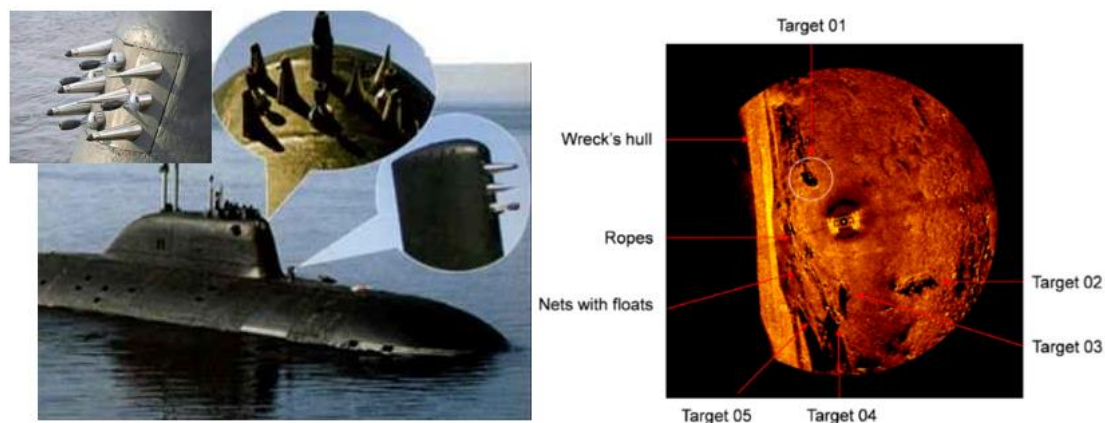


Figure 2.1-5 A SONAR array system on a submarine (left) and a scanning SONAR image of the seabed [36]

Another advantage of the passive SONAR system is that the systematic structure is simpler in comparison to the structure of the active SONAR system. The localisation system is built according to the systematic structure of the passive SONAR system to control the hardware cost and the system complexity.

*2.1.4 Electroacoustics and the state-of-the-art acoustic equipment*

Electroacoustics is an interdisciplinary subject which combines electronics with acoustics. It is an essential science for the application of acoustics. Gramophones

invented by T.A. Edison and carbon microphones invented by A.G. Bell signed the birth of electroacoustics [37]. Electroacoustics received rapid development during the two World Wars, too. During the two World Wars, various speakers, walkie-talkies, and eavesdroppers were manufactured for communication and eavesdropping purposes [38]. In this project, sensors and sampling modules used in the two-dimensional and three-dimensional localisation tests are selected according to the test requirements and the electrical characteristics.

Another rapid development period of electroacoustics began in the 1970s [39]. Powerful microprocessors and SoC systems were applied to acoustic devices, and acoustic systems became smaller, lighter and more powerful. Typical applications of electroacoustics are shown in Figure 2.1-6.

In the implementation of the proposed localisation technology, the MAX9814 acoustic amplifier is selected because of its excellent dynamic response range and self-adaptability. The sensor used in the two-dimensional localisation test is the Murata PKS1-4A10 shock sensor, a cost-effective sensor with stable performance.

The sampling system is built based on an ADLINK DAQ-2010 data acquisition card, which can be configured through the API functions in MATLAB. The highest sampling rate of the DAQ-2010 is 2 MHz. The high sampling rate meets the sampling requirements of all the localisation tests.

The GRAS 146AE measurement microphone and the AA12 amplification module are selected for the three-dimensional localisation test. The 146AE measurement microphone has an extremely low input impedance and a wide dynamic response range and weighs only 35 g. It can easily be fixed with a lightweight microphone stand or attached to solid surfaces for acoustic signal detection.

In this project, the contribution of electroacoustic lies in the system setups. Devices such as sensors, sampling cards, cables and modules such as signal perception, acoustic signal processing, blind acoustic source isolation, noise cancellation module, etc., are selected carefully to ensure that devices used in the tests cover the test requirements. For more

technical details, please refer to Chapters 5 & 6.



Figure 2.1-6 Pioneer DJ-400 DJ controller and GRAS measurement microphone. The advanced electroacoustic engineering products allow users to take control of acoustic waves and record the most realistic acoustic waves for exhaust analysis

At the 2020 advanced engineering exhibition, new localisation systems and acoustic equipment were exhibited, including the SINUS integrated localisation system. It is an interesting short-range localisation system that detects acoustic source movements in front of the microphone array [40], as shown in Figure 2.1-7. SINUS combines computer vision with TDOA-based acoustic localisation technology to achieve short-range visual localisation [40]. The digital data acquired from cameras and microphones is converted into a real-time stereoscopic image. However, the system still needs some improvements regarding the system cost and the detective range.

Figure 2.1-7 The SINUS localisation system developed by AcSoft. The whole system consists of 72 transducers and one optical camera

*2.1.5 Inspirations from Bionics*

The study of audible acoustic waves began with explorations of the human auditory system. Two crucial parameters decide whether humans can hear acoustic waves. The first parameter is the time domain's energy threshold (or amplitude). In Anatomy, scholars indicate that the minimum average human hearing threshold is $10^{-12}\text{W}/m^2$ [41]. Assume that the surface of the eardrum is 0.66 cm$^2$, then the minimum watts that can be heard with human ears is $6.6*10^{-17}\text{W/s}$ approximately. If the minimum hearing interval is set to 0.1s, the minimum energy required to activate the nerve system is $6.6*10^{-18}$J. These conclusions are also referenced to select appropriate acoustic sources for localisation tests.

Another parameter for hearing is the frequency [42]. The hearing threshold of human beings was first determined from 8hz to 24000hz by a French scientist Savoie in 1830 [43]. Afterwards, Biot, Knig, Helmholtz, and their peers kept verifying and adjusting the threshold values [44]. The results acquired by those scientists proved that human beings' high-frequency threshold decreases with age. Hence, the acknowledged range of the auditory frequencies is confirmed from 20 Hz to 20,000 Hz. This audible frequency range is also referenced to set the sampling frequency.

*2.1.6 Speech recognition and machine learning technologies*

Nowadays, machine learning algorithms are mostly used in complex classification tasks due to the powerful classification functionality of machine learning. Speech recognition is a typical example.

Machine learning has been applied to speech recognition since the 2000s [45]. In some cases, words have different meanings under different contexts and accents. As a pioneer in Mandarin speech recognition, iFlytek developed multiple deep learning algorithms to classify tones, accents, and contexts [46].

The speech recognition system developed by iFlytek has a powerful autonomous learning ability, and it reached a word-matching accuracy of 97%. As a result, the speech recognition system developed by iFlytek became the most accurate intelligent speech

recognition software for Mandarin [46].

Similarly, inspired by the fact that machine learning recognises complicated signal patterns in speech signal processing, machine learning algorithms are used to classify acoustic signals from different physical locations in this project.

The decomposition and superposition of acoustic waves, the multipath effect, and SONAR systems are briefly introduced in this chapter. At the same time, the electrical characteristics of acoustic devices are also investigated to build a functional experimental platform. Lastly, speech recognition supported by machine learning is discussed. Machine learning will be utilised to process complex acoustic signal patterns in the proposed acoustic localisation technology.

## 2.2   Mainstream human-computer interaction technologies

Localisation technologies are essential enabling technologies for human-computer interaction [47]. Thus, the working principles of existing human-computer interfaces and the localisation technologies integrated into these human-computer interfaces are reviewed to provide a comprehensive understanding of the localisation technology that could be applied to support three-dimensional human-computer interaction.

According to the interactive models, Human-Computer Interfaces (HCI) are classified into tangible and intangible interfaces [48]. The most popular tangible interfaces are physical touch-tone interfaces such as keyboards, mice, and touchpads. In comparison, intangible user interfaces are next-generation interaction technologies that do not rely on specific physical entities. The acoustic localisation technology proposed in this project can be integrated into the above two interfaces. The working principle, advantages and disadvantages of existing touchtone interfaces are reviewed and summarised to ensure that the proposed technology stays within the scope of practical applications.

The touch screen is the most representative tangible interface. It can be found on most modern devices. In general, touch screens can be divided into the following four categories:

(1) Capacitive Touch Screen

(2) Resistive Touch Screen

(3) Surface Acoustic Wave and Acoustic Pulse Recognition Touch Screen

(4) Infrared Grid and Infrared Acrylic Projection (Figure 2.2-1)



Figure 2.2-1 An infrared virtual keyboard deployed on a table

Capacitive and resistive touch screens are the most commonly used tangible interfaces. The interaction medium between the user and the screen comprises plexiglass, conductor layers, isolation layers, and multiple electrodes [49]. The direct interaction on the screen ensures that the user receives real-time visual feedback. These two touch screens have been applied to different electronic appliances to provide users with an intuitive human-computer interaction experience and reduce the operation difficulty.

While Surface Acoustic Wave (SAW) is a novel technology which uses ultrasonic waves for precise localisation [50], the maximum localisation resolution of SAW reaches the micrometre level so that ridges of a fingerprint are identifiable. Infrared grid or infrared projection technology is popular among young generations because of its high-tech sensations. However, its user interaction experience is bad. It allows users to deploy a virtual keyboard on flat surfaces. Still, its performance is unstable when the environmental light is bright, or the solid surface is rough, and it barely gives users intuitive feedback [51].

## 2.2.1 Capacitive touch screen

A surface-capacitive touch screen continuously releases high-frequency currents on the screen's surface. And when conducts (fingers) slide across or touch the capacitive touch screen, four compensation currents are generated from four electrodes fixed at four corners of the screen. Each electrode provides part of the compensation current, and the current supplied by the electrode is proportional to the length between the touchpoint and the electrode; hence the touchpoint is calculated according to the compensation currents [52]. The localisation procedure is similar to TDOA, except that the signal is current instead of electromagnetic waves.

A projective capacitive screen uses a similar localisation theory to a surface-capacitive touch screen, but it measures capacitance instead of current. There are two types of projective touch screens: absolute capacitive screens and mutual capacitive screens. An absolute capacitive screen uses the sensed object (e.g., fingers) as the second electrode of the capacitor [53]. When the object touches the screen, a charge is induced between the internal electrode and the sensed electrode; then, the coordinate of the touchpoint is calculated by comparing the capacitance change.

The mutual capacitive screen has multiple internal capacitors. It locates the touchpoint according to the change of coupling capacitance caused by finger contacts [54]. In other words, the coordinate of the touchpoint is calculated with the change of coupling capacitance at the intersection of the X and Y directions. In addition, this localisation technology distinguishes the distance between the finger and the capacitor layer. For instance, fingers can cause the original capacitance on the X and Y axes to change without touching the screen. Thus, the projective capacitive screen work normally even if the user wears gloves or the internal capacitors are isolated by thick glass.

The projective touch screen enables multi-touch, but the cost is high. Therefore, only premium devices have a projective capacitive screen [55]. Common applications of the projective capacitive screen include mobile phones, mp4 players, digital cameras, etc.

The localisation steps of a capacitive touch screen are shown in Figure 2.2-2, and the

coordinate template distribution of a projective screen is shown in Figure 2.2-3.
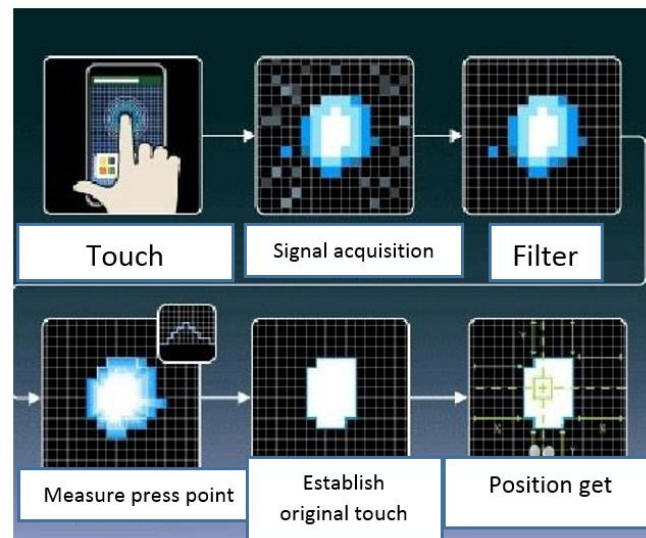


Figure 2.2-2 Illustration of the localisation steps of a capacitive screen [52]

In summary, the capacitive touch screen technology is a planar HCI which uses two-dimensional localisation technology as the enabling technology. On the one hand, capacitive screens require precise structures to ensure accurate localisation results. Thus, the cost is high and the system is fragile. On the other hand, capacitive touch screens are still touchtone interfaces; therefore, the user's mobility is limited.
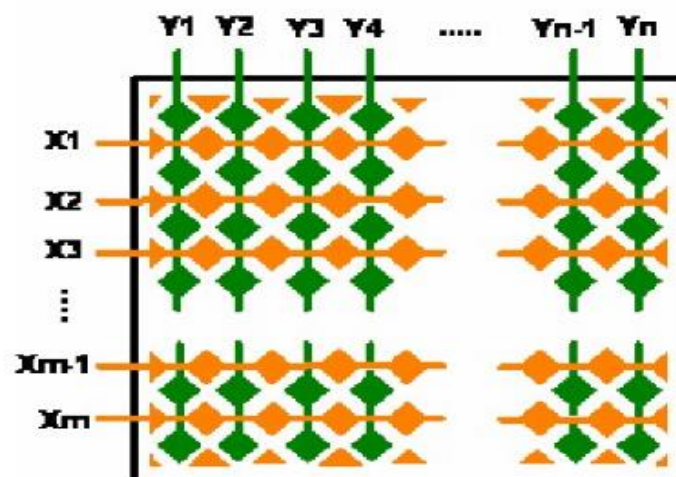


Figure 2.2-3 The coordinate template distribution of a projective screen [55]. The screen usually consists of 5 layers. By examining the coupling capacitance on the X-axis and the Y-axis, the location of fingers is calculated precisely

*2.2.2 Resistive touch screen*

The resistive touch screen has a longer developing history than the capacitive touch screen. The resistive touch screen's hardware cost is lower than the capacitive touch screen because the resistive touch screen utilises elastic deformation of solid materials for localisation instead of measuring capacitance [56]. Curvatures on a resistive touch screen surface cause imbalanced conductivity and the increased or decreased conductivity is measured by electrodes for localisation.
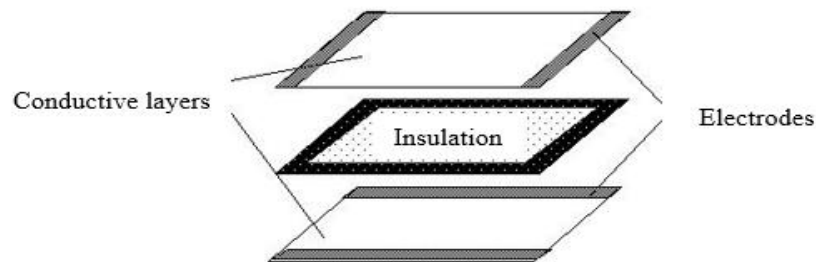


Figure 2.2-4 The structure of a 4-wire resistive touch screen. The two conductive layers represent the X-axis direction and the Y-axis direction, respectively. When a force is applied to the surface, the touchpoint is located by measuring the voltage change [57]

There are two representative resistive screens: the 4-wire resistive touch screen with three layers, as shown in Figure 2.2-4, and the 5-wire resistive touch screen with four layers [57]. For a 4-wire resistive touch screen, the three layers are two resistive conductor layers and one insulation layer. The 5-wire resistive touch screen was developed based on the 4-wire resistive touch screen. The layers of a 5-wire resistive touch screen include one conductive layer, two resistive conductor layers, and one insulation layer.

The first resistive conductor layer is for X-direction coordinate calculation, and the third layer is for Y-direction coordinate calculation. When an external force is applied to the surface of the touch screen, the upper layer is forced to contact the lower layer; the x conductor layer is activated firstly for the voltage measurement in the parallel electric field. The X-coordinate of the touchpoint is calculated with the voltage change. Next, the y conductor layer is activated for coordinate measurement. And the Y-coordinate of the touchpoint is calculated in the same way [57]. It can be concluded that the localisation

27

procedure equals the measurements of the terminal voltages from different directions. The x conductor and y conductor layers are alternatively used as the drive and detective layers to provide standard terminal voltage. Unlike capacitive touch screens, a resistive touch screen is available for multi-touch while maintaining excellent localisation accuracy. Table 2.2-1 compares the capacitive and the resistive touch screens.

Table 2.2-1 Comparison between the resistive touch screen and capacitive touch screen

|  | Resistive | Capacitive |
|---|---|---|
| Stability | Excellent | Medium |
| Accuracy | High | High but later to resistive screens |
| Interaction Objects | Anything causes external forces | Conductors only |
| Electromagnetic interference | Excellent | Medium |
| Contaminant resistance | Strong | Poor |
| Point drift | None | Real-time calibrations |
| Response time | Fast | Very fast |

2.2.3 Acoustic tangible technology

Elo Touch developed Acoustic Pulse Recognition (APR) [58]. Similar to the capacitive touch screen technology, four piezoelectric sensors are deployed at the corners of a glass screen to detect mechanical wave propagation within the screen. A mechanical vibration wave is generated when an external force is applied to the surface of the glass screen, and the mechanical wave will transmit to the four piezoelectric sensors [58]. The location of the contact point is determined by the proportional analysis of the signals received by the four sensors. Filtering, signal processing, and pattern recognition are the core

technologies for APR localisation technology.

Surface Acoustic Wave (SAW) was initially proposed for flow measurement, but it can also be used for acoustic localisation on solid surfaces. Transducers are required in SAW localisation technology. The transducers continuously emit acoustic waves, and reflectors reflect the emitted acoustic waves. If there is no touch action on the surface of the screen, the reflected acoustic waves will always travel back to transducers within a certain time delay with certain patterns. Once there is a touch action on the surface, the finger will cause the acoustic waves passing through the touchpoint to attenuate partially. Then the location of the touchpoint is determined with the time delay and energy attenuation [59]. In addition, the localisation accuracy of SAW localisation technology depends on the wavelength of the emitted acoustic wave. For example, the acoustic wave used in Snapdragon Sense ID is a special mechanical wave with 15μm wavelengths so Sense ID can locate ridges of fingers for the fingerprint lock screen [11].

Although the two acoustic localisation technologies have been applied to Human-Computer Interfaces, the interfaces are still restricted to two-dimensional surfaces. This project aims to develop an acoustic localisation technology which can be applied to three-dimensional spaces.

*2.2.4 Research on acoustic localisation*

The two-dimensional acoustic localisation technology TAI-CHI was applied to various human-computer interactive applications and gained favour from the public [60]. Tangible Media, proposed by MIT, locates the user's arms and fingers with cameras and motion projection. Then the real-time information is uploaded to motors to perform the user's hand movements [61]. This technology can be used for remote control, precisely mimicking the user's actions. Ultraoptics forms specific patterns in the three-dimensional space by gathering acoustic waves [9]. This pattern can be observed visually. The user's fingers can be located by touching the pattern, and appropriate force feedback is provided during the interaction. Existing acoustic localisation-related research achievements are listed in Table 2.2-2.

Table 2.2-2 A summary of acoustic localisation technologies

| | Tai-Chi | Tangible Media (MIT) | Ultrahaptics (UOB) |
|---|---|---|---|
| **Commercial status** | Prototypes | Prototypes | Prototypes |
| **Localisation pattern** | Passive and Active | Active and Passive | Active |
| **Sensors/Transducers** | Piezoceramic and Piezoelectric polymers | Microsoft Kinect /Motion sensor | Ultrasound transducers |
| **Applicable scale** | Table size | Table size | N/A |
| **Dimensions of tangible surfaces** | 2-D | 3-D | 3-D |
| **Single-point detection** | Yes | Yes | Yes |
| **Multiple-point detection** | Yes | Yes | Yes |
| **Acoustic patterns visualisation** | Yes | Yes | Yes |
| **Interaction media** | Solid | Solid/Air/Liquid | Air and liquid |

In summary, localisation technologies are prerequisites for human-computer interaction technology. Different localisation technologies are closely related to the mobility limitations and interaction efficiency of human-computer interaction technologies. From the perspective of localisation principles, the localisation technologies used by human-computer interaction technologies have similar localisation principles. Below, different localisation principles are introduced.

## 2.3  Localisation technologies

There are five mainstream localisation technologies. Namely, differential localisation, Ultra-Wide Band (UWB) localisation, Time Reversal Mirror (TRM), computer vision and TPM localisation technologies. Differential localisation technology is the most

commonly used localisation technology. While UWB is an extension of differential localisation technology aiming to improve the indoor localisation accuracy of the TDOA-based electromagnetic localisation technology. Time Reversal Mirror (TRM) is often used in passive underwater localisation; thus, it is an essential technology for the SONAR system. Below, the most widely used differential localisation technology is introduced.

### 2.3.1 Time Difference of Arrival (TDOA) and Ultra-Wide Band (UWB)

The Time Difference of Arrival (TDOA) technology is a classic localisation technology, and it is used widely in telecommunication and GPS localisation [62]. TDOA utilises time delay detected by multiple sensors to locate the location of the signal source. Theoretically, the location of the signal source can be accurately calculated with precise time difference measurements and the real-time signal propagation velocity profile of the medium and the localisation accuracy of TDOA can be improved by adding more sensors to the localisation network [63]. However, a shortcoming of TDOA is that all sensors must be synchronised strictly to maintain the system's stability and accuracy [64]. The application of TDOA localisation technology in indoor environments is therefore restricted because the deployment and layout of sensors vary significantly in different indoor environments.

The main application of TDOA is remote localisation. With the development of communicative satellites, metre-scale localisation accuracy is achieved with civilian equipment such as mobile phones and GPS [65]. Nowadays, TDOA has become the most common localisation technology, and TDOA-based localisation systems have been applied to navigation, area surveillance and intelligent driving.

Danicki argued that the localisation accuracy of TDOA will be reduced with the increasing electromagnetic interferences [66]. The electromagnetic interference generated when electromagnetic waves pass through another electromagnetic field is a problem that cannot be ignored as more powerful electromagnetic fields are being deployed, especially for RADAR detection and telecommunication [67]. Similarly, TDOA localisation technology encounters similar technical problems in indoor localisation, especially with the popularity of wireless Wi-Fi and Bluetooth networks.

TDOA can be combined with the acoustic wave to locate the acoustic source [68]. Figure 2.3-1 illustrates a TDOA-based two-dimensional acoustic localisation system. Similar to the capacitive screen, four acoustic sensors are placed at the corners of a glass plate. The acoustic source location is located by measuring the time differences of the acoustic wave arriving at four sensors. The main difference between the electromagnetic TDOA localisation technology and the acoustic TDOA localisation technology is that the propagation speed of acoustic waves is much slower than the propagation speed of electromagnetic waves. Therefore, the acoustic TDOA localisation technology has lower sampling requirements and system costs. Correspondingly, the effective range of the acoustic TDOA localisation technology decreases significantly (a few millimetres to a few metres, depending on the energy density of the acoustic signal and the media).
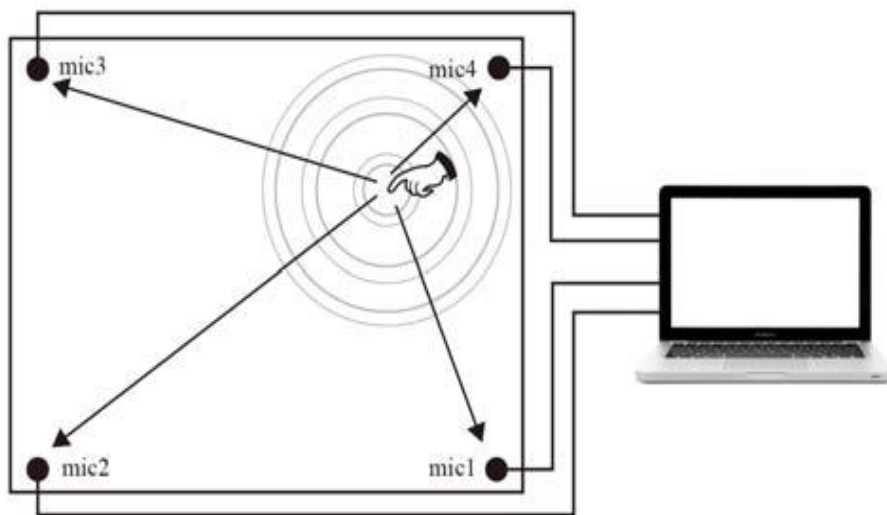


Figure 2.3-1 Illustration of a short-range acoustic TDOA localisation system

In the case that conventional electromagnetic TDOA localisation technology is impractical in complex indoor environments, Ultra-WideBand (UWB) was proposed against indoor localisation tasks. UWB is an extension of the TDOA localisation technology, but it uses dedicated high-frequency electromagnetic waves and mini-sensor arrays to achieve short-range localisation tasks [69]. However, specific problems such as electromagnetic interferences, signal source power and sensor matrix deployment still exist in the implementation of UWB.

*2.3.2 Time-reversal and related applications*

TRM is an array signal processing technology. TRM has the functionality to gather physical vibration energy at a specific point for tasks such as metal inspection and crushing target objects. It can also locate signal sources and weaken the multipath effect in terms of the energy density of acoustic signals.

However, TRM requires the participation of transducers and consumes tremendous energy [70]. The TRM technology is an ideal technology for tasks that require energy focus. For example, the assembled energy generated by ultrasonic TRM technology is used to shatter stones hidden in human organs medically.

The working principle of the time-reversal theory is as follows; firstly, a signal source emits an electromagnetic or acoustic wave to a transducer array, and each transducer partially receives the wave. Secondly, the host processor makes records of these signals received by each transducer and then controls the transducer array to emit phase-reversed waves. Thirdly, the waves generated by the transducer array retrace to the acoustic source, and the energy-gathering effect emerges due to phase stackings of the consecutive waves.

The stackings of the waves lead to a rapid energy increase at the signal source. The accumulated energy can even destroy the signal source. Kuperman achieved another large-scale undersea Time Reversal Mirror (TRM) test in 1997 [71]. In this test, Kuperman successfully verified the energy-gathering effect of TRM in the underwater environment and located acoustic signal sources with hydrophones.

Ray Liu, the president of IEEE, promoted the development of time reversal-based localisation technology by combining time-reversal technology with electromagnetic beamforming technology for indoor localisation [72]. The time reversal-based localisation technology is being applied to indoor localisation as indoor wireless networks such as Wi-Fi and Bluetooth are widely deployed.

Acoustic TRM is inappropriate for three-dimensional localisation, especially for air-filled spaces. The thin chaotic molecular motion fades the energy-gathering effect.

Besides, the acoustic wave has a faster propagation speed in solids and liquids (e.g., 3000 - 5000 m/s in metals and 1500 m/s in seawater) [73]; thus, the energy can be accumulated quickly. In comparison, the acoustic wave propagates at approximately 340m/s in the air. As a result, the time required for energy accumulation in the air and system reaction is longer. Hence, the main applications of acoustic are underwater detection and non-destructive test.

*2.3.3 Pattern matching*

Pattern matching originally refers to the basic string operation, which is used to find all substrings that share a similar pattern to the target string [74]. This concept has been extended and transferred to multiple fields. Pattern matching is used in signal processing to match input signals with sample signals.

The most frequently used cross-correlation processing of signal sequences is a typical pattern matching algorithm. The pattern matching algorithm calculates the similarity between two signal sequences by point-to-point comparing the amplitudes of the two signal sequences [75]. In addition, since the matching database must be established before matching, the sample collection work needs to be accomplished in advance in pattern matching-based applications.

However, collecting samples in advance for comparison is not just a characteristic of pattern matching. In fact, this matching processing has also been widely applied to localisation technology. In most cases, locating a signal source solely with time differences is impractical and inaccurate. Therefore, most localisation technologies require matching processing to improve the localisation accuracy. For example, modern SONAR systems are equipped with various acoustic fingerprint databases to acquire detailed information such as the speed, course and type of the target ship for tracking and attacking tasks [76]. Similarly, acoustic signals collected from different locations contain location information of the acoustic source. In the proposed localisation technology, acoustic sources are located by comparing the signal pattern or features of the input signal with the signal pattern or features of samples stored in the matching database.

The concept of signal pattern emerged during the development of signal analysis. The signal pattern represents the whole signal sequence with a vector with single values; thus, the matching processing is simplified by reducing the number of peer-to-peer computations.

Signal pattern improves the operability and matching accuracy of pattern matching. Pattern matching is currently applied to applications requiring signal classification [77]. Typical applications include voice recognition, image recognition, and cyber security.

The EU project TAI-CHI also utilised pattern matching, TDOA and predictive algorithms to realise acoustic human-computer interaction. The technology is able to transform solid materials into a touch-screen-like human-computer interface, allowing continuous tracking, multi-touch and acoustic source localisation on the surface of three-dimensional objects [7]. The novel tangible technology benchmarked next-generation natural user interfaces.

From the mathematical perspective, similarity criterion, maximum likelihood estimation, Euclidean distance and correlation analysis are all applicable in matching algorithms. TAI-CHI used cross-correlation as the matching methodology and signal peaks as the matching target [78], as shown in Figure 2.3-2. Its pattern matching-based localisation procedure consists of two steps, learning and recognition. In the first step, the acoustic wave signal generated by tapping is labelled with the location coordinate. In the second step, the matching algorithm finds a signal in the template with a peak value similar to the input signal, and the coordinate of the matched signal is determined as the system output.

Pattern Matching-based systems usually have simple system structures and strong environmental adaptability compared to TDOA-based systems in terms of short-range localisation. This project utilises pattern matching to realise acoustic source localisation in the three-dimensional space.
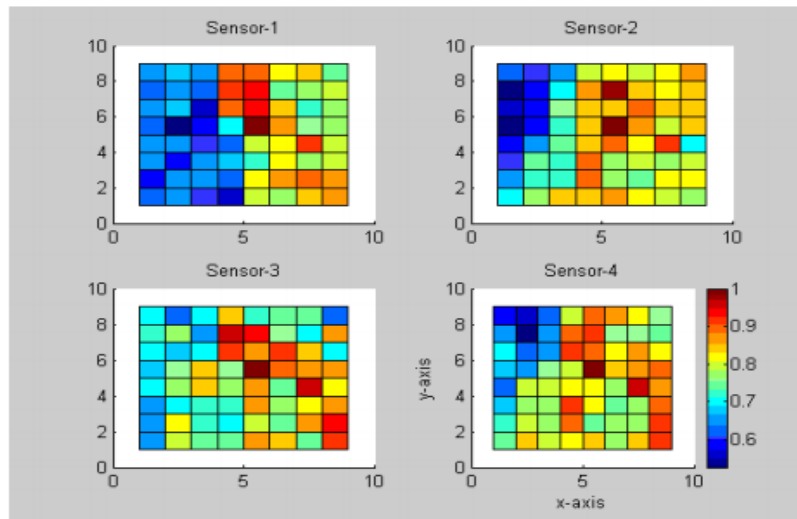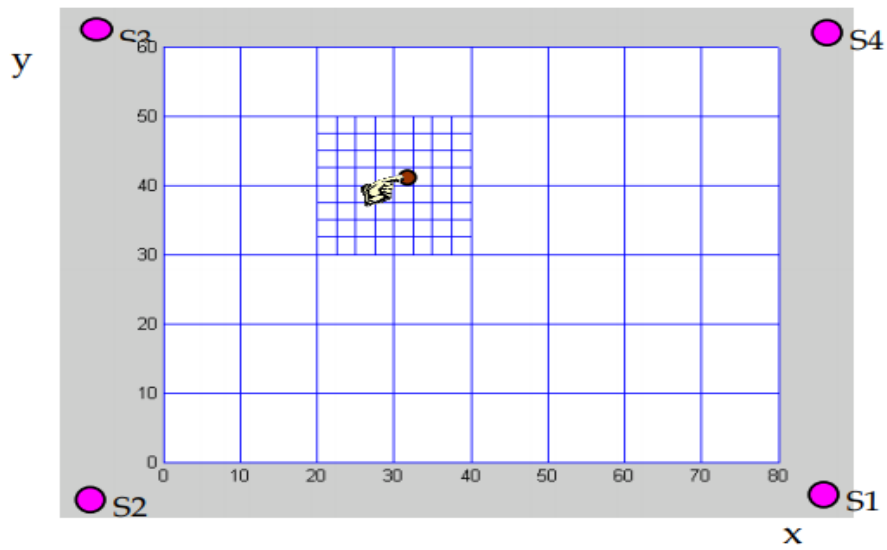
Figure 2.3-2 The peak value-based pattern matching template developed by the TAI-CHI project [78]

### 2.3.4 Difficulties in practical acoustic localisation cases

Danicki summarised positioning problems encountered in battlefield acoustic localisation. He pointed out that the analysis of the location of a practical acoustic source is dynamic, and each acoustic source in the environment should be comprehensively referenced to estimate the location of the concerned acoustic source [79]. For example, the shooter should be located with the muzzle blast, the acoustic wave generated by bullet movements and the acoustic waves generated by the bullet hitting objects [79]. Figure 2.3-3 illustrates the cone blast wave of the muzzle. The cone blast wave is constantly

36

spreading; thus, the cone wave analysis is dynamic.

The dynamic analysis of supersonic bullets begins with the muzzle blast analysis. The muzzle of a gun can be regarded as a tip of an acoustic cone, and the spreading of the cone is related to the acoustic wave caused by the flying bullet. Danicki proposed an acoustic wave diffusion model based on these conclusions [79]. The model was combined with TDOA-based localisation technologies to provide detailed gunshot information.
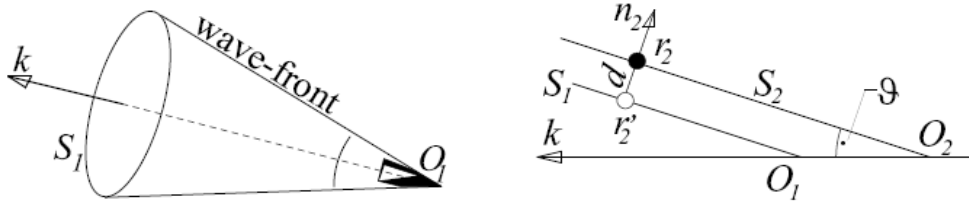


Figure 2.3-3 Cone wave analysis of gunfire. The model developed by Danicki pointed out that the realistic analysis of acoustic waves should always be dynamic [79]

Danicki also tested different combinations of sensor arrays to capture the detailed acoustic signals for analysis. Directional microphones, omnidirectional microphones, and three combination schemes have been tested, namely, 2+4 (2 directional microphones and 4 omnidirectional microphones), 3+3 (3 directional microphones and 3 omnidirectional microphones), and 4+2 (4 directional microphones and 2 omnidirectional microphones). The test results showed that the 2+4 scheme is the most appropriate solution for mid-range gunfire localisation. Danicki's work provided scholars with comprehensive instruction for practical acoustic localisation tasks, especially in the analysis of multiple acoustic sources simultaneously and the deployment of acoustic sensors.

*2.3.5 A summary of localisation technologies*

Two-dimensional acoustic localisation technologies can turn solid objects into interactive interfaces. Unlike traditional localisation systems, system synchronisations, path optimisations, and sensor networks are unnecessary in Template Pattern Matching-based acoustic localisation systems. The cost of the TPM-based system is therefore reduced.

Correspondingly, a disadvantage of TPM is that sample signals must be collected from multiple physical locations in advance. To solve this problem, a tool named 'Location Template' is proposed in this thesis to accomplish the sample collection work. Details of the location template are explained in Chapter 4.

Other localisation technologies are also worth exploring, but most of the localisation technologies are based on two-dimensional surfaces. For instance, Cambridge University conducted a two-dimensional side-channel attack project in which the microphones on mobile phones are fused with angular localisation algorithms to create a side-channel attack system [81]. Their test results showed that cracking personal PINs with internal microphones on the phone is possible. Xiang proposed a localisation technology that utilises Doppler Frequency shift for indoor localisation, and they achieved a ranging error between 30 cm and 40 cm [80].In the pioneering project TAI-CHI, 14 mm localisation accuracy was achieved on the surface of a glass plate with multiple sensor deployments and predictive algorithms [7]. The details of different localisation technologies are shown in Table 2.3-1.

Table 2.3-1. Comparisons between different localisation technologies. TAI-CHI has the best localisation accuracy on solid surfaces. While UWB is the most comprehensive and competitive localisation technology which can be applied to 3D spaces though it needs certain environmental optimisations and infrastructure installations

| Method | Pattern | Mode | Appliacable medium | Accuracy | Sensor | Sensor deployment | Calculation cost | Anti interference | Application scale |
|---|---|---|---|---|---|---|---|---|---|
| Location Template Matching [82] | Pattern matching | Active | Air and solid | 540 mm | Wireless switch | 2 or more | Low | Strong | 3D room scale |
| TAI-CHI [7] | TDOA | Passive | Solid | 14mm | Piezoceramic sensor | 4 | Low | Medium | 2D 1000mm square solid plate |
| UWB [69] | TDOA | Active | Air | 225 mm | Transceiver | 5 or more | High | Strong | 3D room scale |
| Tangible Media [61] | Computer vision | Active | Air | - | Optical camera | 1 | High | Medium | 3D interface |
| 3D haptic shape [9] | Ultrasonic holograms | Active | Air and liquid | - | Transducer | Array | Medium | Low | 3D interface |

Nowadays, various localisation technologies are being developed iteratively. State-of-the-art localisation technologies tend to integrate environmental variables to achieve precise indoor localisation. The concept has been adopted to the proposed localisation

technology. In the proposed localisation technology, machine learning algorithms are used to classify acoustic signals according to the defined signal features (the multipath effect influences acoustic signals and related acoustic features. Therefore, environmental variables are integrated into the localisation system indirectly).

## 2.4 Acoustic localisation

### 2.4.1 Electromagnetic and acoustic waves in localisation

Acoustic localisation is developed mainly for underwater detection and environments with severe electromagnetic interferences [30]. However, acoustic localisation showed great commercial value in natural human-computer interfaces.

The development of human-computer interfaces is inferior to the development of graphic technology. Traditional touchtone HCIs can hardly meet the interactive requirements of novel graphic technologies. For example, in VR games, users have to wear a VR headset and take two controllers to interact with virtual objects. This interactive mode has drawbacks of movement restrictions. However, HCIs powered by two-dimensional and three-dimensional acoustic localisation technologies and possibly, the acoustic force feedback technology may free the user from holding controllers, removing the movement restrictions and bringing users a realistic immersive user experience.

Current mainstream localisation technologies use the electromagnetic wave as the signal carrier. However, applying these electromagnetic localisation technologies to indoor localisation is troublesome. Since the velocity of electromagnetic waves is close to the speed of light, electromagnetic localisation systems require ultra-high-speed sampling systems to ensure accurate time difference measurements.

On the other hand, the multipath effect of the electromagnetic signal also causes multiple interferences, which will seriously affect the localisation accuracy of TDOA-based electromagnetic localisation systems [67]. Techniques such as signal filtering and differential sensor array are developed to eliminate the interferences but these techniques further reduce the robustness of the system and increase the system cost. Therefore, it is impractical to apply TDOA-based electromagnetic localisation technologies to indoor

environments.

Electromagnetic waves are destined for long-range localisation. Unlike the electromagnetic wave, the slow acoustic propagation speed becomes an advantage in short-range localisation. For example, the acoustic propagation velocity within the Gorilla Glass 3 (the screen material of LG Nexus 5) is 4154.44 m/s [81]. Theoretically, systems equipped with a standard 44 kHz A/D converter (a regular sampling frequency for audio cards) and the Time Difference of Arrival (TDOA) positioning algorithm can reach a localisation accuracy of 100 mm. In the air, the velocity of acoustic waves decreases to 340 m/s, and the theoretical localisation accuracy is further improved to 8 mm. Therefore, acoustic localisation is more practical to achieve short-range localisation tasks.

Problems caused by the multipath effect also exist in acoustic localisation. The acoustic wave emitted by an acoustic source is subject to reflection (flat surfaces, e.g., walls, floor and table), diffraction (sharp edges, e.g., small apertures), and scattering; such a superposition of acoustic waves affects the perception of the source signal.

However, the multipath-related signal components in a received acoustic signal also create distinctive signal patterns [83]. Therefore, the multipath signal components in received acoustic signals are utilised to achieve acoustic source localisation. Besides, pattern matching-based localisation systems are robust to environmental noises and interferences. Therefore, a TPM-based acoustic localisation technology is designed in this thesis.

*2.4.2 Machine learning in acoustic localisation*

To achieve pattern matching-based acoustic localisation and utilise the acoustic multipath effect, the relationship between acoustic features and different three-dimensional locations must be determined. However, it is difficult to manually observe, summarise and utilise the hidden patterns in acoustic signals since multiple features (43 features) are defined and more than 120,000 acoustic signals are collected.

Machine learning provides a solution to this problem because machine learning

algorithms specialise in finding hidden patterns in the training data and classifying input data with the summarised patterns. A practical example is that the Random Forest algorithm is used to predict stock prices. A Random Forest-based prediction model can be constructed by learning historical stock data, including stock price, trading volume, market index, etc. An accurately trained model is able to summarise the changing trend of the stock price from historical data and predict future stock prices.

Another example is the Convolutional Neural Network (CNN)-based computer vision. A CNN consists of a series of layers. These layers accomplish tasks from extracting local and global features from the training images, dimensionality reduction, and feature aggregation. Therefore, CNN can automatically extract features from the training data and construct corresponding classifiers. A well-trained classification model can accurately classify input images without feature definitions.

These characteristics of machine learning have been fully considered in this research. A scheme that utilises different machine learning algorithms to classify signals from different locations according to defined signal features and signal images is designed. Therefore, this project utilises signal processing, feature engineering and machine learning to achieve short-range acoustic source localisation in the indoor environment.

## 2.5   Machine learning

### 2.5.1 An introduction to machine learning

A powerful classifier is necessary to utilise the signal components affected by the acoustic multipath effect. Machine learning is particularly suitable for such complex classification tasks. Currently, machine learning has been applied to many aspects, especially in analysing big data and computer vision [84]. According to National Centre for Missing and Exploited Children (NCMEC), 8.2 million missing children reports were received in 2016 [86]. The processing of each report takes up to 30 days since time and labour are required to process pictures and videos in these reports. With the help of AI software (with deep learning frame) developed by Intel, the average single processing period is reduced to 1-2 days, potentially saving thousands of children. Thus, significant

contributions to multiple social fields can be made with the reasonable utilisation of machine learning.

Machine learning consists of supervised learning, unsupervised learning, and semi-supervised learning [85]. The training data must be labelled in advance for model training in supervised learning. SVM, Random Forest, and decision tree are all supervised learning methodologies. Unsupervised learning is widely applied to data mining; it automatically captures specific data patterns from unordered data according to the defined input and output. Unlike supervised learning, a typical unsupervised learning algorithm generates a classification model without labelling data [85].

Semi-supervised learning lies in between the two learning methods listed above. The data modelling starts with unlabelled data then the model predicts results with labelled data [85]. The data distribution across the training phase is stochastic.

*2.5.2 Introduction of deep learning*

Deep learning is the most representative machine learning algorithm because of the superb classification performance and the low requirement for feature processing. The deep learning classification model process complex and repetitive work efficiently; thus, it is an appropriate technology for image and video classification.

A deep learning neural network consists of layers with different functionalities, as shown in Figure 2.5-1. This hierarchical structure allows feature extraction and feature transmission through layers so that the deep learning network automatically matches the input with the output [87]. In the situation that sufficient samples are provided, the classification performance of a deep learning network exceeds that of the traditional deterministic algorithms easily.

The deep learning algorithm's performance scales with the amount of data and may never reach a limit [88]. In other words, the potential of deep learning is infinite. However, deep learning still has some drawbacks. For example, disastrous memory loss occurs when a trained deep learning network is assigned with new tasks [89]. Humans can accumulate experience and master many skills for different tasks, and most importantly,

apply previous experience and skills to new challenges. In contrast, deep learning networks do not have similar mechanisms and structures for experience inheritance.
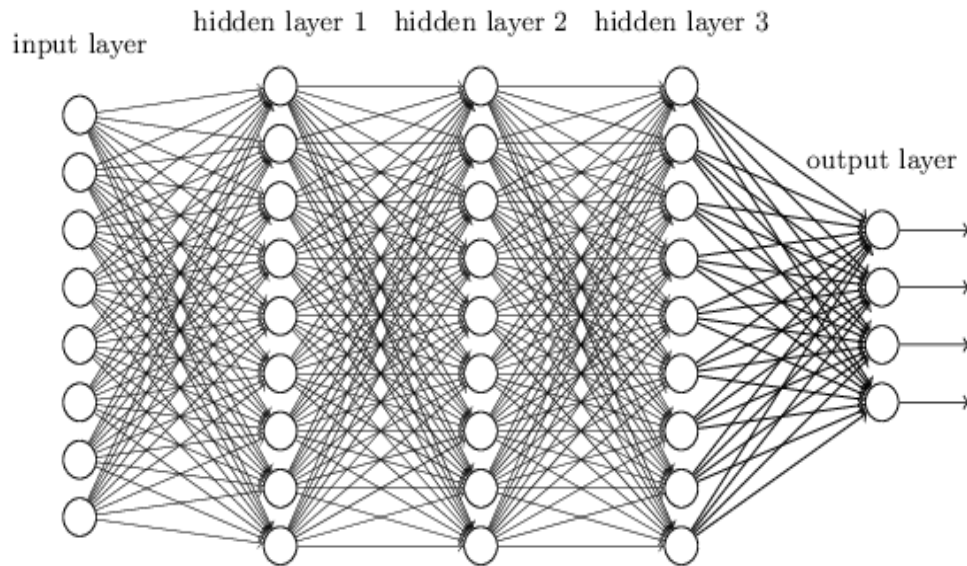


Figure 2.5-1 A typical Deep Neural Network (DNN) structure. A DNN consists of many layers with different functionalities. The training usually requires a large amount of data and huge computational resources

Another defect of deep learning is that it requires a large amount of data and sufficient computational resources for model training. Table 2.5-1 below shows the estimated training time of an English speech recognition model [90]. Thousand hours of training are required to train the speech recognition model. Therefore, applying machine learning to acoustic localisation also comes at a cost. As an attempt, this project established a benchmark for three-dimensional acoustic source localisation and systematised basic concepts for future research on acoustic localisation.

Table 2.5-1 Training hours consumed by an English word error rate model [90]. The model is a 9-layer CNN model with 68M parameters

| Fraction of Data | Hours | Regular Dev | Noisy Dev |
| --- | --- | --- | --- |
| 1% | 120 | 29.23 | 50.97 |
| 10% | 1200 | 13.80 | 22.99 |
| 20% | 2400 | 11.65 | 20.41 |
| 50% | 6000 | 9.51 | 15.90 |
| 100% | 12000 | 8.46 | 13.59 |

*2.5.3 The development of machine learning in commercial applications*

In 2016, Intel announced the Lake Crest platform (Nervana platform) and Xeon Phi and stated that these products are the most powerful processors [91]. As a response, Nvidia illustrated their Pascal GPU and new image AI software immediately.

Nvidia's AI software manipulates weather and natural light in videos and images. Nvidia used an innovative strategy to enhance the performance of its AI software. Namely, two models were created for imaging processing; the first one is Variational Auto-Encoder (VAE), and the second one is Generative Adversarial Nets (GAN) [92]. The two models compete against each other simultaneously. VAE is designed to cheat GAN with similar images, while GAN is designed to find the differences in images and restore the images. As a result, VAE and GAN evolved simultaneously to a perfect dynamic equilibrium state. The images created by VAE and GAN are so vivid that some researchers even started to worry about the spam of fake videos and images on the Internet.

Machine learning has also been applied to speech recognition. In 2016, the initial vocal software VoCo developed by Adobe was tested with data collected from individual voice lines [93]. The voice line model can transfer texts into natural and fluent voice lines, as shown in Figure 2.5-2.

This item has been removed due to third party copyright. The unabridged version of the thesis can be viewed at the Lanchester library, Coventry University

Figure 2.5-2 The announcement of the Adobe VoCo. The live event showed that modern AI software could mimic anyone's voice line

On the other hand, Justus Thies from FAU published their real-time face transformation application, Face 2 Face, in 2016 [94]. This application allows users to modify and

change the face and looks of individuals in an image or a video. The facial expressions can be replicated perfectly so that people can't distinguish the authenticity of videos. Soon after the launch of Face 2 Face, Face 2 Face and VoCo were used together to create vivid videos of celebrities; thus, these technologies were banned soon as they may cause negative influences and social panic [95].

From these examples, it can be summarised that machine learning algorithms are powerful tools that can be integrated into multiple applications. In this project, machine learning is also applied to improve the localisation performance of the proposed acoustic localisation technology.

## 2.6    Acoustic signal processing technologies for localisation

### 2.6.1 Beamforming and microphone array

Beamforming technology is a modern signal array processing technology for directional signal transmission and reception. The technology has two major applications: localisation and signal transmission [96]. The main beamforming applications include electronic countermeasures, phased array RADAR, and SONAR.

Take the beamforming-based SONAR system as an example; the system obtains multiple acoustic signals with the hydrophone array. The signals are then processed with power spectrum estimation to determine the incoming direction of the signals [97]. Assuming that the hydrophone array has $m$ hydrophone, then $m$ directional vectors can be calculated. The signal source's location is calculated by converging these directional vectors. However, since the semiconductor technology was immature in the early years, the accuracy of beamforming localisation was poor.

Compared to the application of beamforming in localisation, beamforming technology is mainly used to improve the signal-to-noise ratio currently. Beamforming technology relies on the deployment of a multi-antenna system. Not only multiple receiving antennas are required, but multiple transmitting antennas are required. So that a wireless signal corresponding to a spatial stream from the transmitting end to the receiving end is transmitted through multiple paths, the signal-to-noise ratio at the receiving end can be

significantly improved by integrating the beamforming algorithm at the receiving end for multiple antenna signal processing. Even if the receiving end is miles away, the signal quality is still lossless [98].

The beamforming technology can be divided into delay-sum, filter-sum, and adaptive beamforming technologies according to the summary of synchronised array data. A beamforming sensor array is illustrated in Figure 2.6-1.
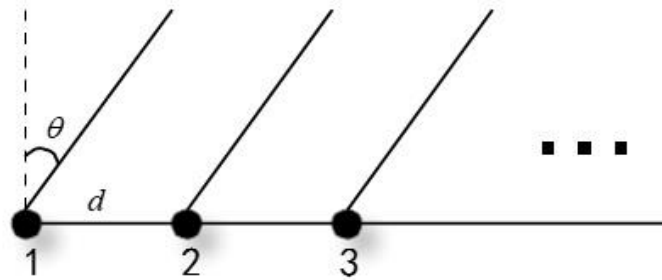


Figure 2.6-1 Illustration of a beamforming sensor array. Every sensor (black dots 1,2,3) receives an independent signal. Multiple Input and Multiple Output (MIMO) is one of the most representative characteristics of a beamforming system

Beamforming has become part of the 802.11 ac WLAN standard [99]. Interestingly, WLAN-based wireless network localisation technology has been proposed and developed. Beamforming technology has once again been applied to localisation. However, the serious indoor multipath effect restricts the accuracy of WLAN-based localisation technology. Currently, the WLAN-based wireless local network localisation technology is still in the laboratory stage.

*2.6.2 Microphone array*

The differential microphone array technology is similar to the beamforming technology but biased toward acoustic signal perception. In the early 1970s, the first microphone array system was built and applied to speech signal processing for noise suppression [100]. Since then, the microphone array technology has been gradually applied to various research fields, such as sound field measurement, sound pressure measurement and acoustic localisation. A microphone array consists of multiple microphones. The microphones are deployed in a specific pattern to sample part of the acoustic wave emitted by an acoustic source. Special sampling modules and signal processing

algorithms are required to support the microphone array.

A microphone array is shown in Figure 2.6-2. The first-order difference in sound pressure is obtained by subtracting the outputs of the two closely placed omnidirectional microphones. Similarly, N microphones deliver up to N-1 order of differential pressure. Similarly, signals with a high signal-to-noise ratio are acquired by superpositioning acoustic wave signals collected by each microphone [101].

The microphone array technology also has shortcomings. A large amount of data is collected due to the deployment of many microphones; processing the data consumes more computational resources and time. Therefore, the placement of microphones is always optimised to ensure the best acoustic measurements with the least number of microphones.

The interval between microphones should be smaller than the acoustic wavelength to prevent spatial phase ambiguity, as shown in (Eq. 2-6) [102].

$$\frac{\omega \cdot d}{c} = \omega \tau_0 \ll 2\pi \qquad\qquad 2-6$$

where $d$ is the interval between microphones. $c$ is the propagation speed of the acoustic wave. $\omega$ is the frequency of the acoustic signal to be measured. $\tau_0$ represents the time for the acoustic wave to travel between microphones.
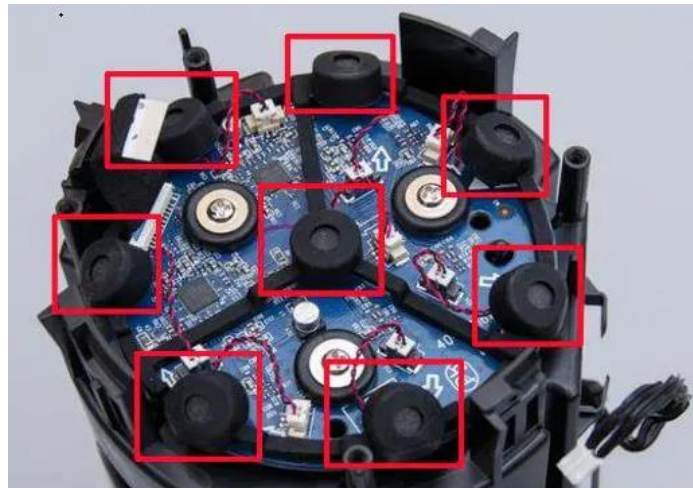


Figure 2.6-2 Illustration of a microphone array [102]. There are 8 microphones placed in an FPGA-driven voice input board. The microphone array provides multiple input acoustic signals for speech recognition

In conclusion, beamforming has many technical advantages; it improves the stability of wireless systems, and it can be used to achieve localisation. The microphone array technology shares similarities with the beamforming technology; it ensures the reception of multi-order acoustic waves and potentially reduces noises. These technologies provide certain references for the collection of acoustic signals in this project.

## 2.7 Summary

The literature review aims to find the problems of existing localisation technologies in short-range localisation applications and make technical reservations for the proposed acoustic localisation. The new localisation technology is designed against these problems.

In this thesis, signal processing is combined with signal feature-based pattern matching to achieve two-dimensional acoustic localisation. On the other hand, signal processing is combined with multiple acoustic features-based machine learning classification algorithms to achieve three-dimensional acoustic localisation.

62% of the literature is about electromagnetic TDOA-based localisation. Only 12% of the papers are about pattern matching. And most papers are signal strength and frequency drift-based. This project aims to realise the localisation of audible acoustic sources in the three-dimensional space with multiple acoustic features and pattern matching localisation algorithm.

Overall, Research Challenges in acoustic localisation are listed as follows:

(1) The limited literature on acoustic signal pattern matching. The design and implementation of the acoustic localisation technology begin with limited studies on two-dimensional acoustic localisation and electromagnetic localisation.

(2) The definition of acoustic features and the utilisation of multiple acoustic features.

(3) The application of machine learning algorithms in terms of acoustic localisation. And the collection and processing of the training and test data.

(4) Technical problems such as the labelling of three-dimensional locations and system implementations.

For the first research challenge, the characteristics, merits and drawbacks of the three-dimensional electromagnetic localisation technologies are carefully analysed and studied to determine the most appropriate localisation method for short-range acoustic localisation. As a result, TPM-based localisation systems are robust, adaptive and cost-efficient; thus, it is selected as the core localisation method. Next, previous studies and research on two-dimensional acoustic localisation are reviewed. Characteristics of acoustic waves, acoustic signal processing techniques and pattern matching algorithms are integrated into the proposed localisation technology. Finally, localisation systems and localisation tests are built and designed to verify the feasibility of the proposed acoustic localisation technology.

A conclusion is drawn by summarising the pattern matching-based localisation test results achieved by other institutions. That the single feature-based TPM is not accurate enough to achieve indoor localisation. Therefore, a series of features are defined in the proposed localisation technology to improve the localisation resolution and classification accuracy.

For the second research challenge, signal strength and frequency shift are the most commonly used matching objects according to previous research on pattern matching-based localisation. As an attempt, the proposed localisation technology uses multiple signal features (from the time domain to the frequency domain) to achieve acoustic source localisation. Another challenge is to ensure that the classification algorithm effectively utilises these features for accurate signal matching. In this project, the two-dimensional acoustic localisation system utilises three acoustic features and a weight-allocated cross-correlation matching algorithm for signal matching. The three-dimensional acoustic localisation system utilises 43 signal features, 17 signal images, and two machine learning algorithms for signal matching.

For the third research challenge, similar to the machine learning algorithms used in other classification tasks, different classification models are trained with numeric data and picture data. More specifically, acoustic feature extraction functions are compiled to generate input data for feature value-based machine learning algorithms. For image

recognition-based machine learning algorithms, various acoustic transform functions are compiled. These functions generate various signal images including waveform in the time domain, spectrum, power spectrum, and sound pressure for the training of the classification model.

For the fourth research challenge, a special tool-location template is designed to label the three-dimensional space. Signal acquisition, signal processing, and signal separation modules are constructed in the implementation of the proposed acoustic localisation technology.

A detailed comparison between electromagnetic and acoustic waves is presented in 2.4. In summary, acoustic waves have unique advantages in short-range localisation; thus, the acoustic wave is selected as the signal carrier for the proposed short-range localisation technology. Applicable acoustic properties, human-computer interaction technologies, localisation technologies and machine learning are reviewed to propose the short-range localisation technology.

Acoustic waves propagate in two patterns in solids [24]. Thus, the multipath effect in solid media is more complex. As a result, acoustic signals collected at different locations on a solid surface have significant differences. Therefore, localisation tests are first conducted on solid surfaces. Three-dimensional localisation tests are executed after the proposed localisation technology is preliminarily verified with the test results of the two-dimensional localisation test.

According to Fourier transform theory, acoustic sources with single frequencies are selected to ensure deep mining of acoustic features. In the formal three-dimensional localisation test, buzzers with single frequencies are used as acoustic sources to provide template signals and input signals for the localisation system.

# 3 Research methodology

The project aims to propose a novel acoustic localisation technology which can be applied to short-range scenarios; thus, scientific tests are necessary to validate the proposed technology.

Based on this fact, primary research is the core research. The primary research is further developed into four steps. In the first step, the limitations of existing localisation technologies and novel requirements for short-range localisation technology are summarised. Meanwhile, research objectives are established according to the summary. The second step is to design the localisation technology according to the proposed objectives and determine the technical details. The third step is the implementation of the proposed technology. In the last step, the performance of the localisation system is evaluated with localisation tests and the proposed localisation technology is optimised according to the exposed problems.

The secondary research provides an overview of the development of short-range localisation technologies and the latest technologies that can be utilised to achieve acoustic localisation.

## 3.1 Project development model

The spiral development model developed from the software life cycle model is selected as the project development model [103]. The spiral development model is a widely used engineering model. The processing flow of the spiral model is as follows:

(1) Determine the aim and objectives and identify the requirements for the product.

(2) Develop a prototype according to the objectives and requirements.

(3) Evaluate the prototype with an appropriate benchmark and analyse the result.

(4) Make modifications according to the evaluation and the result analysis.

(5) Return to the first stage. Review and correct the objectives and requirements, then proceed again.

Figure 3.1-1 The spiral development model. The model consists of four steps: identifying the research gap, proposing the solution and evaluation, developing, and reviewing. The four steps may be repeated a few times to regulate the research outcomes [103]

The spiral development model is illustrated in Figure 3.1-1. One of the advantages of the spiral model is that the scope is constantly adjusted during development. A benefit of the spiral development model is that research progress is supervised strictly until the research achievements meet expectations. Based on the spiral model shown in Figure 3.1-1, four specific stages are outlined for this project, as shown in Figure 3.1-2.

Figure 3.1-2 The spiral model explicitly designed for this project. Instead of producing commercial products, this project focuses on creating a novel acoustic localisation technology for both indoor and outdoor applications

Stage 1: Determination of the research gap and objectives. This project aims to find an approach to locating an acoustic source in a three-dimensional space. Secondary research is accomplished first to provide an overview of human-computer interaction technologies, existing localisation technologies and related algorithms, and acoustic properties that can be exploited for localisation.

Stage 2: The design of the acoustic localisation technology. The localisation theory and technical details are proposed according to the research objectives summarised in Stage 1.

Stage 3: the implementation of the proposed acoustic localisation technology.

Stage 4: Test and evaluation. The research direction and research progress are calibrated by evaluating the test results. If the initial aim and objectives are missed, changes will be made to ensure the missed objectives are covered in the next loop.

At the same time, the system and algorithms are optimised constantly to improve the

53

system efficiency and minimise inherent and random system errors.

Unlike the linear research model, the spiral development model has multiple loops through the entire research procedure; each loop has independent tasks and bullet points. As a result, the acoustic localisation technology is continuously optimised to meet the primary research objectives.

## 3.2 Preliminary research outlines

The acoustic localisation technology proposed in this research is developed to locate acoustic sources in the three-dimensional space. However, it is easier to locate acoustic sources on a two-dimensional plane compared to locating acoustic sources in a three-dimensional space. Therefore, the proposed acoustic localisation technology is first applied to two-dimensional localisation.

The two-dimensional localisation test is conducted on the surface of a glass plate. The test simulates the interaction between a user and a touch screen. The location of the user's finger is located with acoustic waves generated by tapping on the glass plate, as shown in Figure 3.2-1. From the perspective of human-computer interaction, such an interaction process is consistent with the interaction process of the touch screen.

The three-dimensional localisation test is conducted in an indoor laboratory. The localisation system locates acoustic sources with the sound waves emitted by the acoustic source. In the scenario, three-dimensional human-computer interaction technologies powered by the proposed acoustic localisation technology can locate users according to the acoustic signals emitted by the user, as shown in Figure 3.2-2.

Two-dimensional localisation tests:

The two-dimensional localisation system is relatively easy to build since the solid medium has a high signal-to-noise ratio. As a result, acoustic signals are less susceptible to external interference. Moreover, the multipath effect is more complex inside solid objects due to the coexistence of longitudinal and transverse waves. Figure 3.2-1 illustrates the two-dimensional acoustic localisation.

Only one shock sensor is used in the two-dimensional localisation test because pattern matching does not rely on time difference measurements. Instead, the location of the acoustic source is calculated by comparing the input acoustic signal with template acoustic signals. Thus, the sample collection work must be accomplished in advance.

In the sample collection work, multiple acoustic sample signals are sampled from the labelled locations on the surface of solid objects. Next, the acoustic features of these acoustic sample signals are extracted. Finally, a matching database is established by merging extracted acoustic features with coordinate labels.

The matching algorithm used in the two-dimensional localisation test is the cross-correlation algorithm. The cross-correlation algorithm calculates matching coefficients between an input signal and sample signals in the matching database. The coefficient represents the similarity between the input signal and a sample signal. The sample signal with the highest correlation coefficient is confirmed as the matched signal, and its coordinate label is identified as the system output.
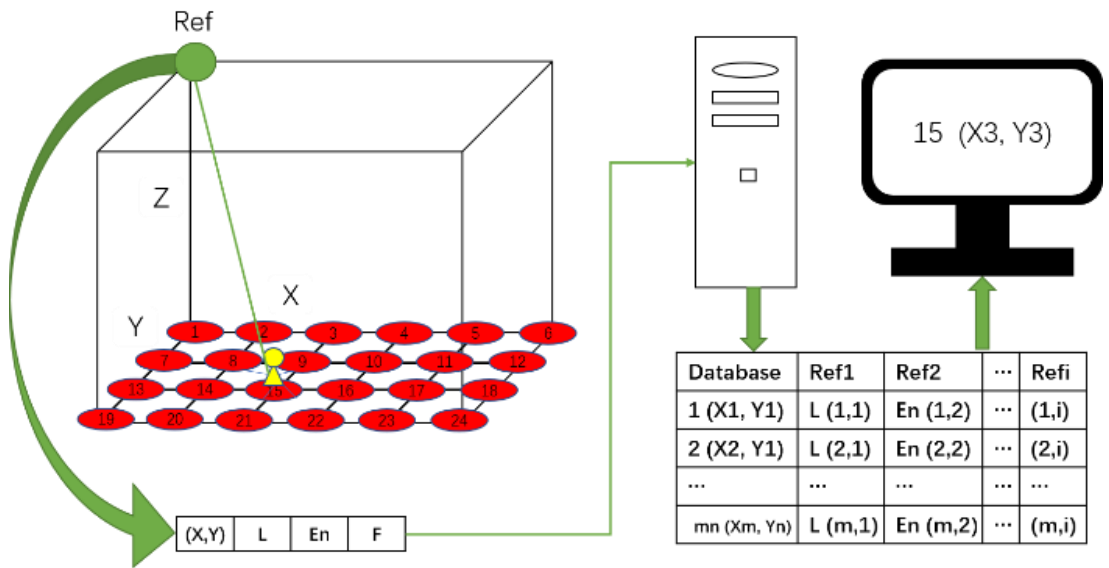


Figure 3.2-1 Illustration of the pattern matching-based two-dimensional acoustic localisation

Two steps are needed for the implementation of the pattern matching-based two-dimensional acoustic localisation. In the first step, template acoustic signals are collected from specific areas on the glass plate and stored in a matching database. In the second

step, the stored samples are called for matching processing when an input signal is provided.

The three-dimensional localisation test:

The medium in a three-dimensional space is air. Therefore, the signal-noise ratio, the acoustic velocity, and the wave scattering pattern change dramatically. In addition, acoustic attenuation and interferences such as environmental noises and spatial disturbances further complicate the solution to pattern matching-based three-dimensional acoustic localisation. As a result, the positioning performance of the single feature-based pattern matching algorithm will significantly decrease.

Multiple signal features are therefore extracted to address the problem; multiple dimensional and dimensionless signal features are extracted to ensure the diversity of template signals. One drawback of using multiple signal features is that the three-dimensional localisation system requires more computational resources for sampling and signal processing.

In addition, the point-to-point cross-correlation algorithm also consumes substantial computational resources when dealing with long signal sequences; thus, the matching algorithm for three-dimensional localisation needs to be upgraded to handle the processing of vast amounts of data.

Machine learning-based matching algorithms are integrated into the system to solve the above problem. Unlike the cross-correlation algorithm used in the two-dimensional localisation test, classification models (function models trained with different learning strategies) are high-dimensional functions [104]. A trained classification model directly calculates a system output when an input is provided to the system without comparing the input signal with each sample signal. However, massive data is required to train an accurate classification model. In the three-dimensional localisation test, a robot system is built to provide matching learning algorithms with sufficient training data, and a High-Performance Computer (HPC) is utilised to provide computational resources for the training of the classification model.

In the three-dimensional acoustic signal collection work, location templates are used to label the location of the acoustic source in a three-dimensional space. The concept of location templates is introduced in 6.2.1. Multiple cubic location templates are designed to label pre-defined locations in the three-dimensional localisation test. Meanwhile, location templates are merged with the robot coordinate system to simplify the data collection work. The zero point of the robot is also regarded as the origin point of location templates. In this case, each spatial location defined by location templates can be seen visually in the teach pendant of the robot system.

Figure 3.2-2 illustrates the proposed three-dimensional localisation methodology. Three steps are needed to achieve three-dimensional acoustic source localisation. The first step is data collection; sample signals are collected from pre-defined locations and then merged with coordinates to create a training database. Next, a localisation model is trained with the training database in the second stage. In the third stage, the trained model calculates the coordinates of the acoustic source when an input acoustic signal is provided to the system. The location of the acoustic source is estimated at one of the pre-defined locations on the location templates.



| Database | Ref1 | Ref2 | ... | Ref(i) |
|---|---|---|---|---|
| $1(X_1,Y_1,Z_1)$ | L(1,1,1) | En(1,1,1) | ... | (1) |
| $2(X_2,Y_1,Z_1)$ | L(2,1,1) | En(2,1,1) | ... | (2) |
| ... | ... | ... | ... | ... |
| $N(X_i,Y_j,Z_m)$ | L(i,j,m) | En(i,j,m) | ... | (n) |

| (X,Y,Z) | Length | Energy & Entropy | Frequency | Zeroing | Etc. |
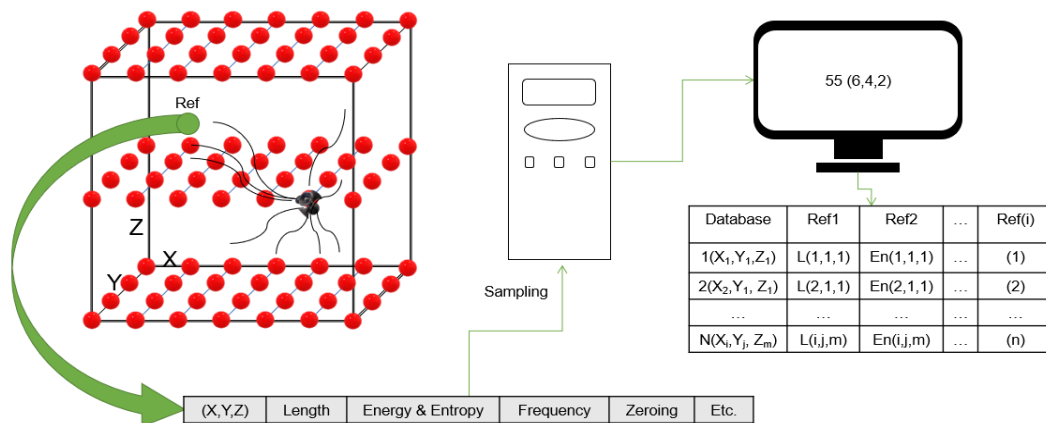
Figure 3.2-2 Illustration of the pattern matching-based three-dimensional acoustic localisation

The main difference between two-dimensional localisation and three-dimensional localisation is that the cross-correlation matching algorithm is replaced with the machine learning algorithm.

## 3.3 Risk assessment

This project relies on primary research. A robot system is built to execute repetitive data collection tasks in the three-dimensional localisation test. Thus, potential humanity issues are avoided. Meanwhile, safety regulations are followed strictly to prevent accidents from happening. The robot operator must wear protective equipment since the laboratory is set inside an industrial factory. Furthermore, operators must follow the safety rules and the electricity safety guidance to prevent potential dangers.

## 3.4 Ethics

Students at Coventry University are required to submit an ethics proposal to ensure that there are no ethical arguments. Any research plan and equipment changes must be documented and integrated into the ethics proposal for review. In this project, the most critical ethical point is the ingenuity of the three-dimensional acoustic source localisation technology. Thus, related papers, projects and studies are reviewed to ensure the novelty and contributions of this project. The proposed localisation technology, implementations and test strategies are examined continuously according to the spiral development model to avoid ethical issues. The whole project is completed strictly according to the IEEE Code of Ethics.

## 3.5 Summary

The project development methodology - the spiral development model, together with the two-dimensional and the three-dimensional localisation principles of the proposed acoustic localisation technology are briefly introduced in this chapter. Rigorous research programs are designed according to the two-dimensional and three-dimensional localisation requirements. Details of the proposed localisation technology are explained in the next chapter.

# 4 Location Template-based Positioning Model

The Location Template-Based Positioning Model (LTPM) is a pattern matching-based short-range acoustic localisation technology. In 2007, Ze J et al. [105] proposed the Location Template Matching (LTM) localisation technology which utilises pattern matching, TDOA and recursive algorithm to achieve 40 mm level acoustic localisation on two-dimensional surfaces. Similarly, the LTPM aims to achieve three-dimensional acoustic localisation with the acoustic multipath effect, signal processing, pattern matching and machine learning algorithms. Unlike the LTM, the LTPM abandons the multi-sensor deployment and TDOA algorithms and uses multiple acoustic features to achieve three-dimensional acoustic source localisation.

## 4.1 Pattern matching-based localisation

Unlike the TDOA-based localisation technology, signal processing and pattern matching are combined to match the input signal with the template signals in pattern matching-based localisation technology. Signal features and matching algorithms are the two factors which decide the positioning performance.

In pattern matching-based localisation, the matching algorithm locates the signal source by comparing the input signal with pre-collected sample signals. The pre-collected samples are labelled with location coordinates; thus, once a matched sample is found, the coordinate of the matched sample is confirmed as the system output.

Signal propagation is confined to various physical effects; thus, the received signals vary significantly in the time domain when the acoustic source is deployed at different locations. LTPM utilises this character to achieve signal source localisation.

The matching between signals is achieved through feature matching. One advantage of using signal features for matching is that comparisons between complete signal sequences are avoided. In this project, features from different domains are used for pattern matching instead of features from a single domain.

*4.1.1 Introduction of the classic Location Fingerprint*

Generally, any signal feature that correlates with the location of the signal source is described as a location fingerprint. Therefore, localisation technologies using location fingerprints and pattern matching algorithms refer to the Location Fingerprint localisation technology.

The classic Location Fingerprint achieves localisation with the attenuation and deformation of the received signal caused by the environment. In 2003, the first Location Fingerprint localisation system, which utilised electromagnetic Received Signal Strength (RSS) as a location fingerprint, was successfully built and tested by Ahonen and Eskelinen [28]. In their hypothesis, the electromagnetic wave keeps attenuating during the propagation due to the energy attenuation and the multipath effect. Thus, the RSS decreases linearly with the propagation distance.

The RSS is defined in (Eq. 4-1). The illustration of their test setups is shown in Figure 4.1-1. According to their test results, the first Location Fingerprint localisation system achieved a three-dimensional accuracy of 188 m in 95% of the location estimates and a three-dimensional accuracy of 10 m in 50% of the location estimates [28].

$$RSS = p_t - \text{K} - 10\alpha \log_{10} d \qquad\qquad 4-1$$

where $p_t$ represents the emission power of the signal source. $\alpha$ is the path loss exponent. K is a constant value related to the environment and transmission frequency. $d$ is the distance between the emission point and the access point.

Though Ahonen and Eskelinen verified that Location Fingerprint localisation is feasible, the test result showed that RSS is not an appropriate location fingerprint in indoor environments as the signal feature has overlapping boundary problems, as shown in Figure 4.1-2. The problem is severe when the sample size increases. As a result, the Thiessen polygons generated by samples from different locations can hardly be approximately treated with traditional deterministic algorithms, resulting in a decreased positioning accuracy [28]. In conclusion, the performance of the classic Location Fingerprint technology is limited with RSS; thus, the localisation technology needs to be

updated and extended for practical applications.
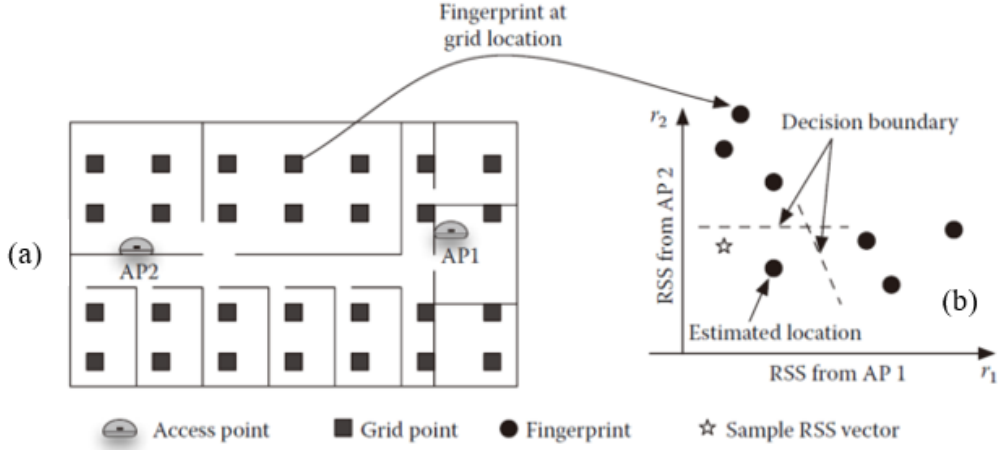


Figure 4.1-1 The illustration of RSS-based Location Fingerprint (a). (b) shows the distribution of RSS. Two Access Points (AP) are set in this test. For a signal source, AP1 and AP2 receive different RSS values. Samples received by these APs are stored in the matching database [28]

Ahonen found that the localisation performance of the Location Fingerprint-based system could be improved by increasing the number of Access Points (AP) and analysing the spatial distribution of RSS [28]. Since then, the Location Fingerprint technology's performance has been gradually improved, and indoor localisation research has adopted these techniques and optimisations. A typical application of Location Fingerprint technology is the short-range wireless network-based indoor localisation. For example, Peizheng realised a positioning accuracy of 500 mm with the Wi-Fi network, machine learning and sensor deployment optimisation techniques in 2020 [82]. Such indoor localisation technologies are always required in smart homes, wireless charging and security monitoring system.

Overall, the Location Fingerprint localisation technology utilises RSS and the deterministic matching algorithm to achieve mid-range signal source localisation (positioning accuracy between 0.5m and 200m). A summary is that the location fingerprint and the matching algorithm determine the performance of the Location Fingerprint-based localisation technology.
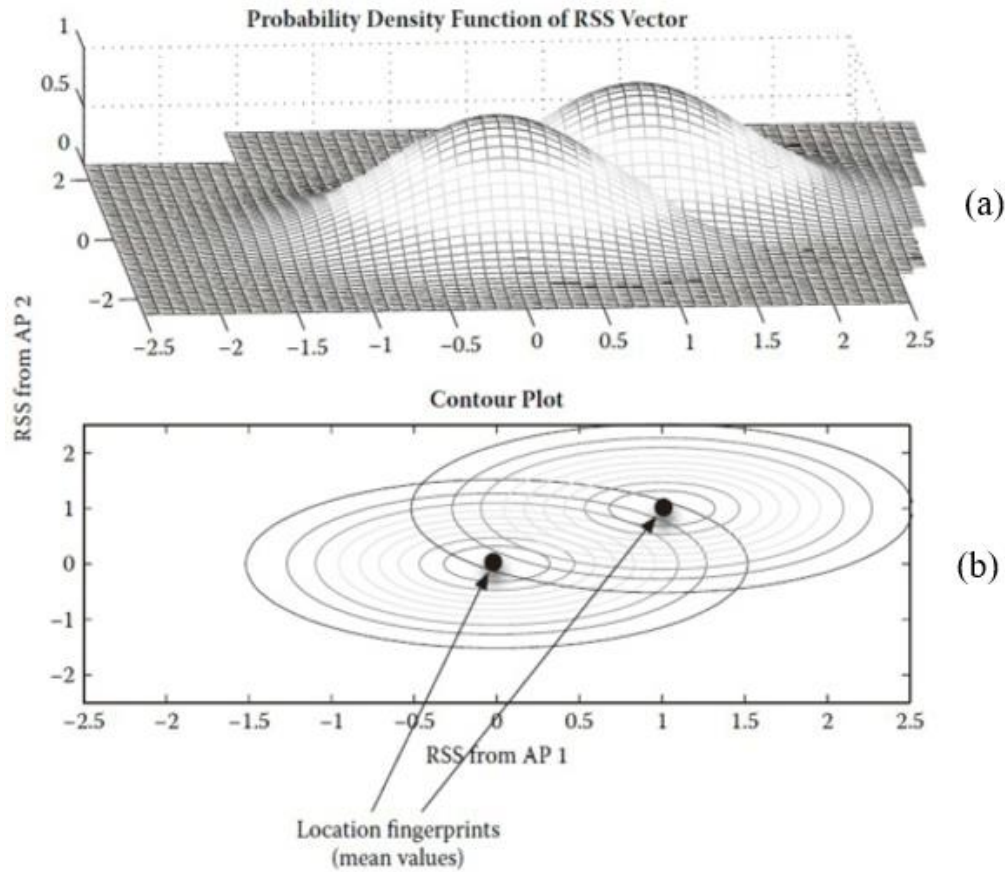
Figure 4.1-2 Two signal samples collected from the exact location (a). The two signals are always covered and stacked partly. The overlap of multiple sample signals causes Thiessen Polygon (b) [28]

Secondly, specific differences exist in different pattern matching-based localisation technologies. However, a common point is that different pattern matching-based localisation technologies rely on location-related signal features. Factors that affect signal features include the multipath effect, path loss, environmental variables such as temperature, humidity, and homogeneity of the medium etc. Therefore, an extensive database is necessary to achieve accurate acoustic signal matching.

*4.1.2 Challenges in acoustic Location Fingerprint*

In classic Location Fingerprint, pre-defined locations are associated with signal features distorted at these physical locations. Ideally, an injective mapping relation between a physical location and a set of feature values is established. For example, the Received Signal Strength (RSS) reduces with propagation distance, and each RSS value corresponds to one location, as shown in Figure 4.1-3 (a). But in practical cases, the

mapping relations are consistently bijective or surjective due to the blurred edge of congener samples in the database. As a result, the localisation system performs poorly because one RSS value corresponds to multiple locations.
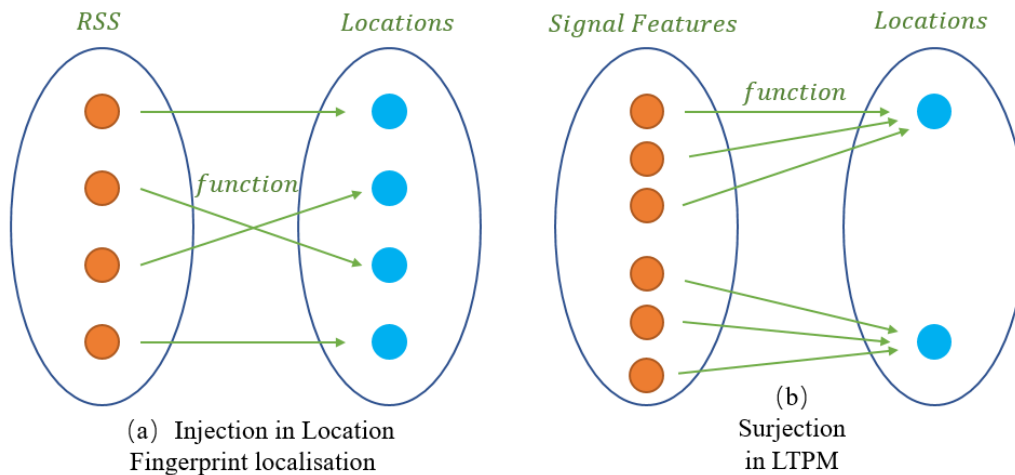


Figure 4.1-3 Mapping relations of Location Fingerprint (a) and LTPM (b). The injections in (a) often turn into surjections in practical applications, resulting in wrong matching results. In LTPM, signals from different locations are more distinguishable since multiple signal features are defined

As shown in Figure 4.1-2 (b), the classic Location Fingerprint uses RSS in pattern matching, and RSS leads to sample overlapping. As a result, the boundaries of different sample sets are blurred.

Hence, the first challenge is establishing surjective relations between signals and source locations with the acoustic multipath effect and feature extraction techniques. From the perspective of matching efficiency, any quantity that contributes to identifying different source locations is a valid signal feature. Direct physical quantities in the time domain, such as signal strength and duration, or indirect signal features from other domains, such as power, energy density, frequency distribution, and signal entropy, are imported to LTPM to form a feature set for accurate matching. The distribution of samples from different locations becomes more centralised after multiple features are defined.

Therefore, a feature set with multiple signal features is defined for the LTPM-based three-dimensional localisation test; the feature set contains 43 signal features and location labels. According to the test results presented in Chapter 6, the feature set is more

effective than the RSS used in the classic Location Fingerprint, as the LTPM-based localisation system achieved a three-dimensional accuracy of 173 mm.

The matching between the input signal and the template signals is another challenge. Traditional point-to-point correlation matching algorithms compare the entire signal sequences of the input signal and sample signals. Consequently, it consumes enormous computational sources and is inefficient if the database contains millions of signals.

Machine learning algorithms are appropriate algorithms for processing extensive data. Instead of calculating coefficients, the classification model is trained with samples stored in the database. A trained model directly calculates an output according to an input signal, as shown in Figure 4.1-4.
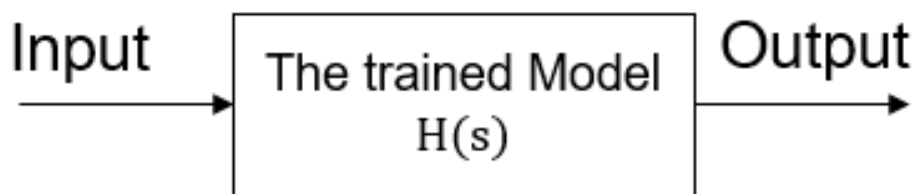


Figure 4.1-4 The trained model is a transfer function with the input signal being an excitation. The classification model is optimised with the training data, and the trained model calculates the system output according to the training data

The Thiessen polygon boundary problem is solved with machine learning. Linearly inseparable samples in a low-dimensional space are linearly separable in high-dimensional spaces [106]. The kernel function of machine learning algorithms maps features to high-dimensional spaces for accurate separations, as shown in Figure 4.1-5.

Therefore, traditional correlation analysis algorithms in the classic Location Fingerprint technology are replaced with machine learning algorithms. The Random Forest and the Convolutional Neural Network are selected regarding training efficiency and classification performance. Details of selected machine learning algorithms are introduced in 4.2.
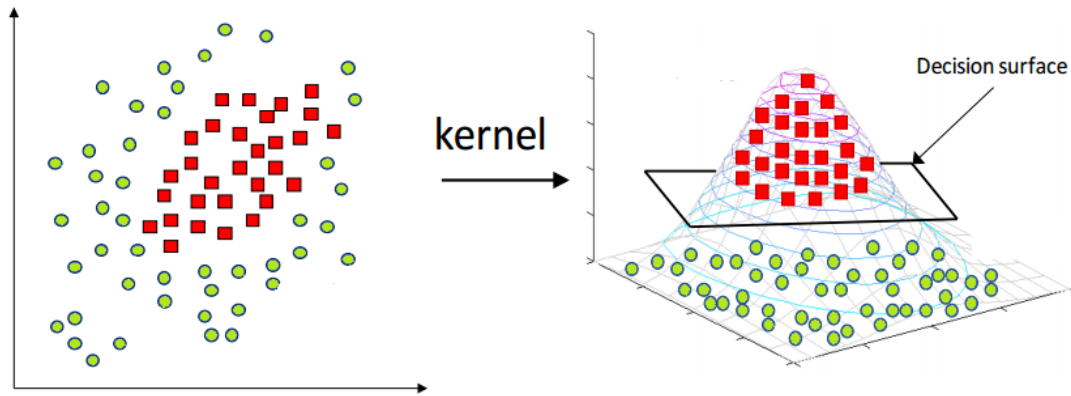
Figure 4.1-5 The kernel function performs hyperplane separation on two-dimensional distributed samples and accurately classifies samples in a high-dimensional space

### 4.1.3 Advantages of the acoustic signal

Electromagnetic localisation dominates the development of short-range localisation technologies. The current research direction of electromagnetic localisation inclines to use existing wireless networks for indoor localisation. For example, the daily used Wi-Fi routers with 802.11n specification is an ideal wireless platform for electromagnetic short-range localisation. Nevertheless, localisation systems based on Wi-Fi wireless networks usually perform poorly because of inherent defects in signal processing and the communication specificity of the wireless network.

Unlike the electromagnetic wave, the acoustic wave is an elastic wave generated by mechanical vibrations; it has some unique physical characteristics compared to the electromagnetic wave. For example, acoustic waves require certain mediums for propagation; acoustic waves' vibration frequency and propagation speed are much lower than electromagnetic waves. These acoustic characteristics facilitate signal sampling. Besides, acoustic waves can be artificially generated; thus, acoustic signals are ideal signals for natural human-computer interaction.

Acoustic signals received by acoustic sensors located at a fixed location vary significantly when the acoustic source is placed at different locations because the direct, reflection, diffraction, and refraction signal components in the received signal change with the location of the acoustic source. And LTPM uses this acoustic characteristic to

locate acoustic sources.

In LTPM, an acoustic source is located passively by receiving the acoustic waves emitted by the acoustic source. Users do not need to hold or wear signal transmission devices in this localisation mode. Technically, the localisation mode is classified as network localisation [107]. An advantage of network localisation is that network localisation-based localisation technologies have no mobile restrictions. All calculations are handled by the localisation network. Therefore, HCIs powered by such localisation technologies could bring users a natural and immersive interaction experience.

## 4.2    Machine learning algorithms for LTPM

Since classic location fingerprint technology and TDOA localisation technologies have limitations, the project aims to improve the utility and the positioning accuracy of the short-range localisation technology with the acoustic multipath effect, signal processing technologies and matching algorithms (deterministic correlation matching algorithms and machine learning algorithms are implemented to classify acoustic signals from different locations).

Unlike standard machine learning applications, acoustic features are extracted as training data. The classification model is trained with extracted features instead of acoustic signals. An advantage is that signal classification accuracy is improved significantly with multiple signal features from different domains. The disadvantage is that a series of signal transformations and feature extraction functions must be executed before training.

### 4.2.1 Random Forest

Random Forest (RF) belongs to supervised learning. It is an efficient machine learning algorithm which consists of multiple decision trees. Each decision tree is trained with defined features for an independent classification output [108].
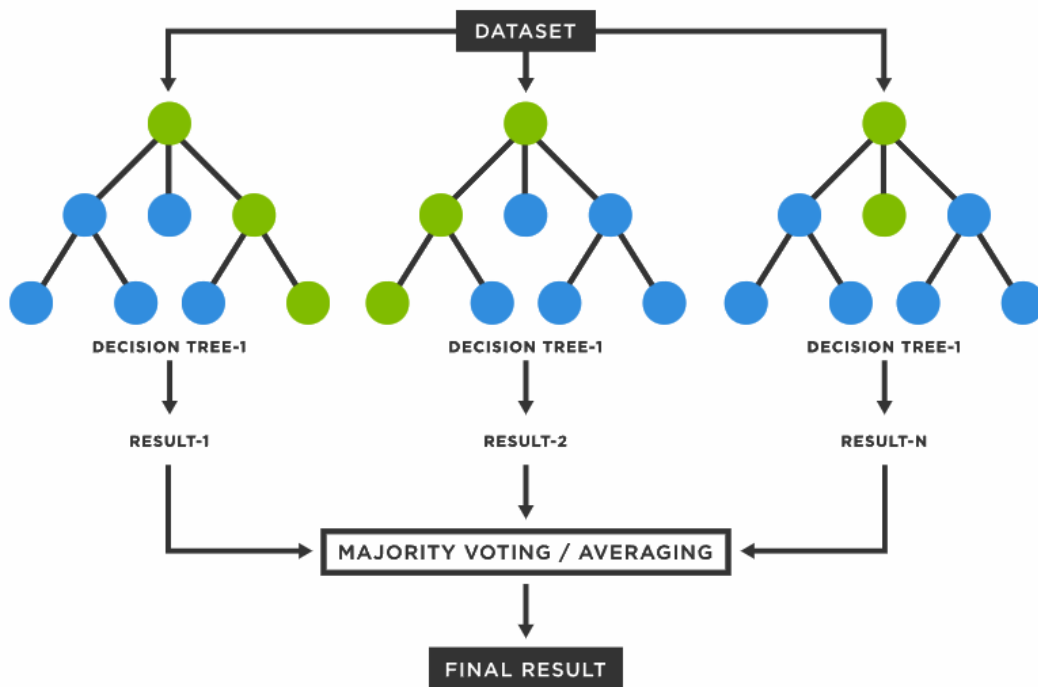
Figure 4.2-1 Structure of Random Forest. The algorithm is based on Bayesian Criterion. There are three decision trees in this illustration, and each tree is trained with the bagging approach until it reaches the limitation of dividing attributes

Figure 4.2-1 illustrates the structure of the Random Forest. The decision tree starts splitting from a root node with no incoming branches. The branches of the root node then provide inputs for the internal nodes (also known as decision nodes). Homogeneous subsets are formed by internal nodes performing the functionalities-based evaluation. These subsets are leaf nodes (also known as terminal nodes). Leaf nodes represent all possible outcomes within the training dataset. As a result, the diversity of RF not only comes from samples but also from nodes. When an input is provided, the input goes through every decision tree, and multiple results will be acquired. The most frequent result will be confirmed as the output [108].

Ideally, there is no correlation between decision trees. Therefore, Random Forest has a robust fault-tolerant ability. Random Forest even maintains classification accuracy in the situation that part of the data is missing [109]. At the same time, RF requires few computational resources as each decision tree is trained simultaneously.

Generally, four steps are needed to apply RF to the LTPM-based localisation system.

67

Regulation of input signals is the first step. Sample signals are regulated and stored in a database. The second step is feature extraction; signal features are extracted and written into an Excel file. In the third step, the RF algorithm reads the Excel file, and the splitting of each decision tree begins. Each node selects its subsets according to the information gain until the information gain reaches its limit. At this point, a decision tree is built. In the last step, step 3 is repeated to create multiple decision trees. Finally, a Random Forest is constructed.

Below is a detailed implementation flow chart of the Random Forest-based LTPM; the four steps are further divided into 12 exact steps, as shown in Figure 4.2-2.
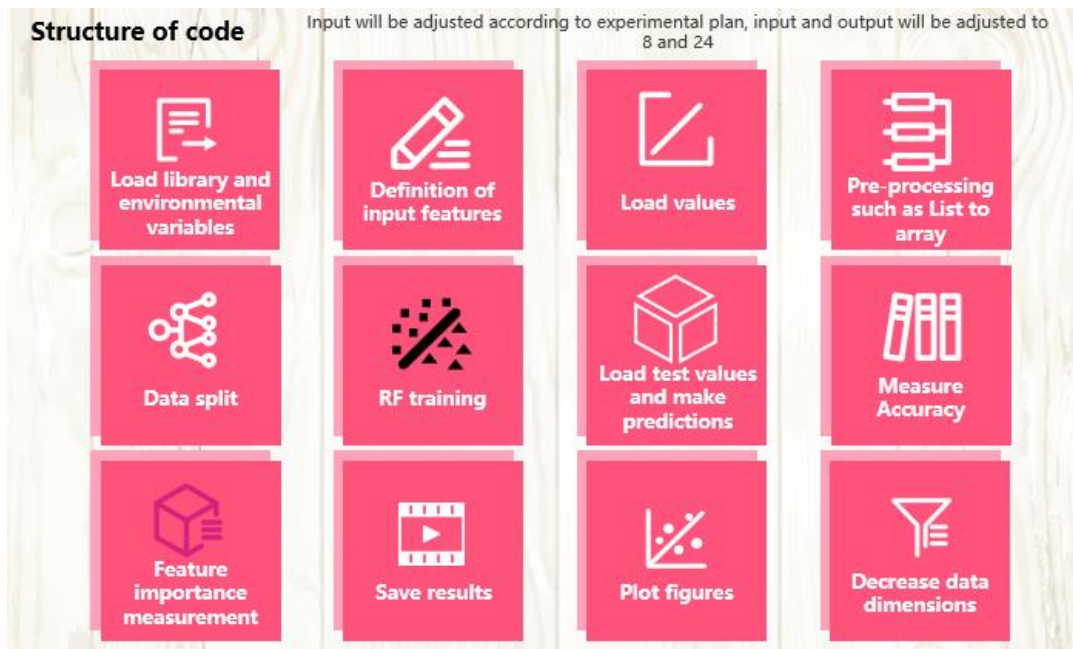


Figure 4.2-2 A brief illustration of RF-based LTPM. Overall, 12 steps are designed. One significant difference between the proposed four steps and the applicable 12 steps is that matching tests are executed for evaluation after the training completes

### 4.2.2 Convolutional Neural Network (CNN)

The convolutional neural network represents deep learning. A neural network is a mathematical or computational model that mimics the structure and functions of biological neural networks [110] and comprises many artificial neurons. Different networks are constructed according to different layer connection methods. Neural networks have decision-making and judgment abilities. Nowadays, neural networks have been applied to image and speech recognition widely.

CNN performs translation-invariant classification due to its hierarchical structure [111]. A CNN consists of various layers, each with specific functionality for automatic classification. CNN has fewer parameters in comparison to RF, but the classification performance of CNN is superior. Another advantage of CNN is that feature extraction is unnecessary because CNN learns autonomously from the training data [111]. Therefore, CNN is a reasonable classification tool for LTPM since it summarises signal patterns hidden in signal images and classifies acoustic signals from different locations according to the summarised patterns.

RNN has the same status as CNN. These two networks are extensions of traditional neural networks with multi-layer neural network connections. CNN is selected over RNN because of the following reasons:

(1) CNN is a spatial expansion convolutional model; the convolution happens between neural functions and features (signal patterns). While RNN is a time-expansion memory model, neural functions are correlated to time-variant outputs [112]. Therefore, CNN is more appropriate for acoustic source localisation tasks because the precise spatial location is marked by features affected by the multipath effect, regardless of the time span.

(2) CNN has a variety of structures and more robust adaptability than RNN.

A typical CNN consists of three parts: the convolutional layer, the pooling layer and the fully connected layer [113]. The convolutional layer is responsible for feature extraction, while the pooling layer reduces the extracted features' dimensionality and prevents overfitting. The fully connected layer has a similar function to a classifier; it maps the learned "distributed feature" to the sample tag space.

On top of that, CNN establishes feature maps automatically by matching the output data with the input data. Predictably, CNNs with different structures perform differently, which may require specific application optimisations [114].

Figure 4.2-3 shows the structure of a CNN model. The model is an animal recognition model, whereas Figure 4.2-4 illustrates the structure of the CNN for LTPM.
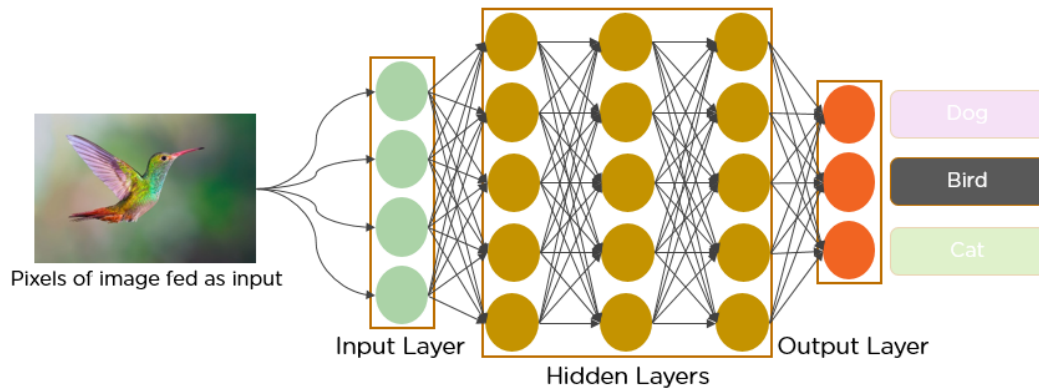
Figure 4.2-3 The structure of a CNN. With the fast development of deep learning, the latest CNN algorithms are powerful enough to generate vivid videos and images [92]. CNN automatically extracts features from pictures according to input and output and avoids overfitting. It has been widely applied to multiple applications

The first layer in Figure 4.2-4 is the input layer. The second layer is the convolutional layer for feature extraction. The third layer is the pooling layer for downsampling. The fourth and fifth layers are the second convolutional and second pooling layers. The sixth layer is the transit layer, while the seventh is a fully connected layer. The eighth layer to the twelfth layer are hidden layers for linear partition.

Compared with the training of Random Forest, the training of CNN is concise. Three steps are required for the implementation of CNN. In the first step, MATLAB functions are compiled to convert the time domain signal into 17 signal images. Secondly, the images are loaded by the deep learning algorithm for training. In the third step, the performance of the trained model is tested with test datasets. Figure 4.2-5 shows the detailed implementation steps of the CNN-based LTPM.
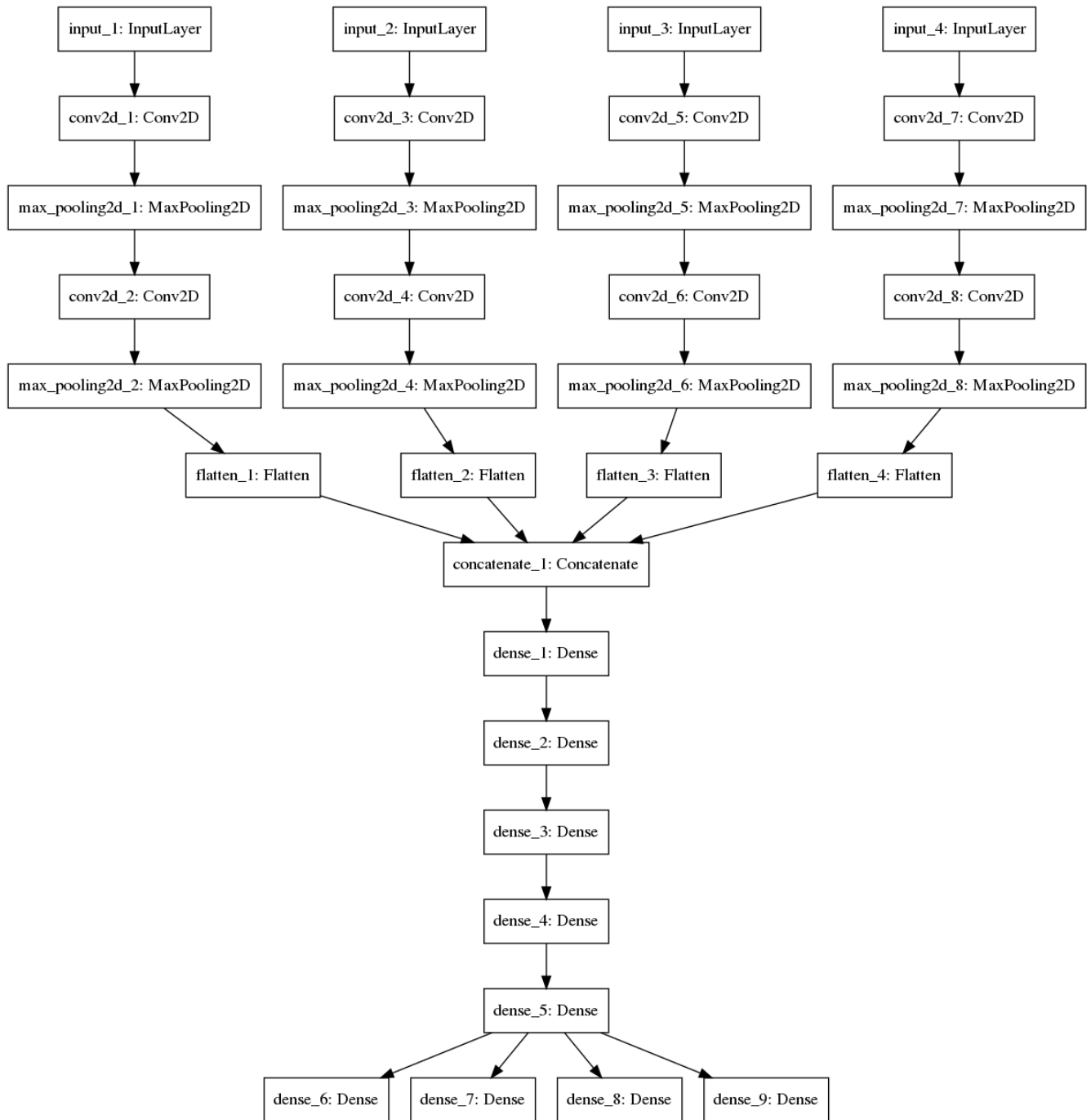
Figure 4.2-4 The weight-shared convolutional neural network used in the three-dimensional localisation test. 4 inputs and 4 outputs are defined in this network
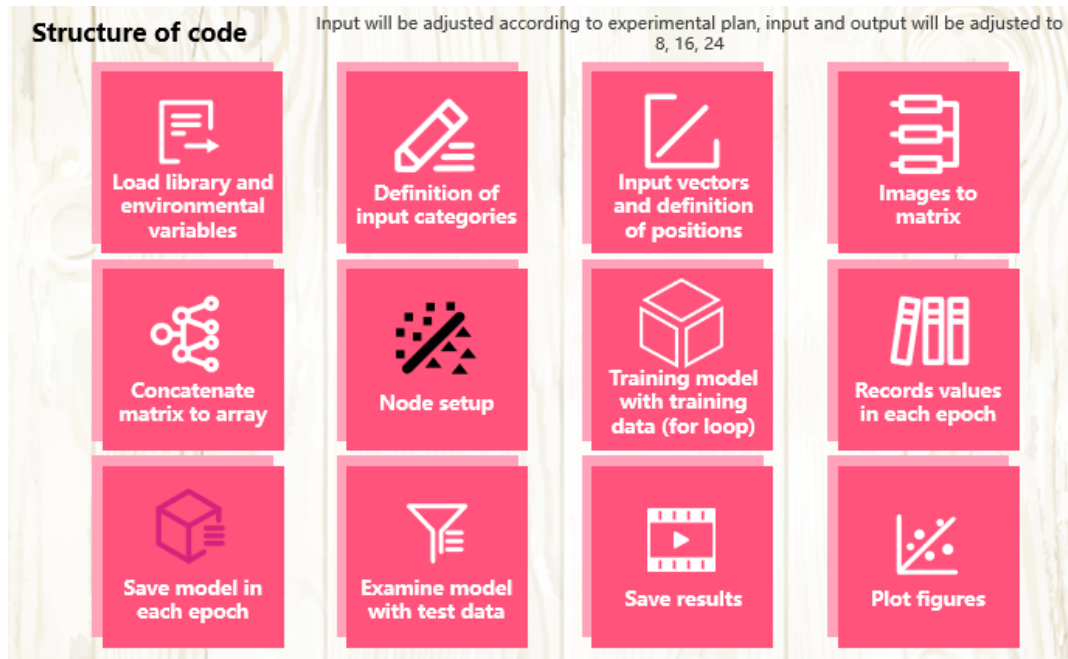
Figure 4.2-5 A brief illustration of CNN-based LTPM. Similar to the Random Forest training procedure, input data are divided into training and test sets, and test datasets are loaded for test after the training is finished

### 4.2.3 Feature introduction

Two feature sets which contain 43 signal features and 17 signal images are assigned to Random Forest-based LTPM and CNN-based LTPM, respectively.

For Random Forest-based LTPM, time domain features, frequency domain features, power spectrum features, and entropy features of acoustic signals are calculated as the training data. Complete features are listed in Table 6.2-2.

Some features, such as the deviation of a signal sequence and peak and valley value, are mathematical features calculated directly with the signal's time distribution. Frequency features such as centroid frequency, skewness of frequency, and STD of frequency components are computed from the spectrum of the signal. Entropies of the acoustic signals, such as max or min power, occupied bandwidth, power entropy, signal entropy and information entropy, are also calculated.

Below is the introduction of two features: the length of a signal sequence and the margin factor.

The power of an acoustic signal is definite; thus, the strength of the signal attenuates with

propagation distance. Therefore, signal length is selected to reflect effective sampling points.

Assume that a signal sequence received by a sensor is:

$$X(i) = s(i - t_d) + \eta(i) \qquad\qquad 4-2$$

whereas $s(i)$ is the source signal. $t_d$ is the time delay. $\eta(i)$ is random Gaussian white noise. And the max value of $i$ represents the last component sampled by the sensor; therefore, $i$ equals the length of the signal sequence.

The square root value of $X(i)$ is the square of the mean value of the arithmetic square root and can be described as $X_r$:

$$X_r = \left( \frac{1}{N} \sum_{i=1}^{N} \sqrt{|X_i|} \right)^2 \qquad\qquad 4-3$$

And margin factor can be described as follows:

$$L = \frac{X_{peak_{valley}}}{X_r} \qquad\qquad 4-4$$

The margin factor is illustrated because it indicates the power distribution of multipath components [115]. In total, 43 signal features are projected to the Random Forest algorithm. For more details, please refer to 6.2.5.

Similarly, 17 signal images are defined for the convolution neural network algorithm, as shown in Table 6.2-3.

Two-dimensional images display the signal variations over time. CNN classification models are trained with signal sequences that change synchronously with time in different domains. Four signal images are shown in Figure 4.2-6. For more details, please refer to 6.2.5.

The spectrum of an acoustic signal shows the correlation between power and frequency. The correlation can hardly be described with a single value. From the signal processing perspective, the pattern residing in signal images contains more location-related

information about the signal source; CNN extracts the information and classifies acoustic signals according to the extracted information.
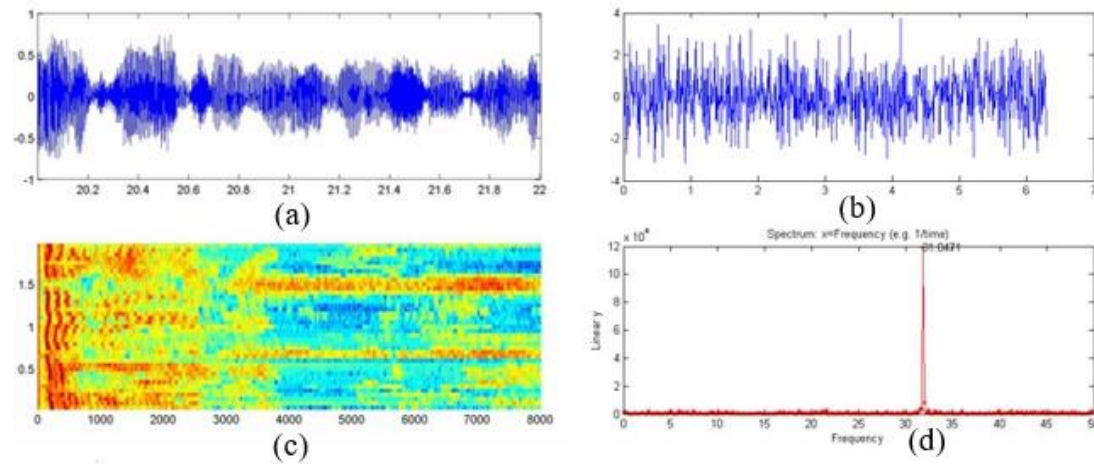


Figure 4.2-6 Illustration of signal images. (a) is the time distribution of an acoustic signal. (b) is the change of centroid frequency of the acoustic signal. The left colourful image (c) is the spectrum of the acoustic signal, while the right image (d) is the frequency image converted with the Fast Fourier Transform (FFT). Different images contain different information regarding physical locations

## 4.3    Analysis of applications

LTPM is different from the classic Location Fingerprint and TDOA-based localisation technologies. In LTPM, sample signals are collected from predetermined locations in advance. Then the LTPM-based localisation system outputs the coordinate of an input signal by comparing specific acoustic features of the input acoustic signal with that of sample signals.

Compared to traditional TDOA technologies, the acoustic source is located with matching calculations rather than real-time differential calculations. Therefore, sensor array technology and noise reduction technology are unnecessary in LTPM. In addition, the algorithm of LTPM is flexible and straightforward, as filtering and system synchronisation are unnecessary. Besides, LTPM has strong adaptability to complex indoor environments. Although presampling work must be completed in advance, LTPM successfully located different acoustic sources in a complex indoor environment in the three-dimensional localisation test.

From the perspective of human-computer interaction, LTPM is an enabling technology

which provides human-computer interaction technologies with an essential ability to locate acoustic sources. Natural User Interfaces (NUIs) equipped with LTPM technology potentially free users from wearing heavy devices and enhance the interaction experience gained from VR technology. Furthermore, LTPM can be potentially applied to battlefield acoustic source localisation, SONAR and smart home systems.

A commercial application and a military application are presented below.

Automatic acoustic tracking technology:

In indoor conferences, speakers need a microphone to make a speech. Inconvenience caused by holding wired or wireless microphones can be solved by automatic acoustic tracking technology.

In this scenario, microphones with different directivities and automated acoustic tracking technology can be combined to automatically track the speaker's location, as shown in Figure 4.3-1. Automatic acoustic tracking technology locates the speaker. Then, it adjusts the microphone system to collect acoustic waves from the specified area.



Figure 4.3-1 Illustration of the automatic acoustic tracking technology. The technology frees the speaker from holding a microphone and allows the speaker to walk freely in the space

One advantage of an automatic acoustic tracking system is that speakers no longer need to hold and wear a microphone during the speech. In addition, microphone systems that

are heavier but more stable and powerful than portable microphones are available in this case.

Battlefield acoustic localisation technology:

Another critical application of acoustic waves lies in the military. For example, acoustic waves are utilised to achieve target localisation, armour welding, acoustic attack, near-field communication, etc.

In complex and noisy battlefield environments, acoustic localisation is essential. For example, in skirmishes, soldiers may lose the ability to determine enemies' locations due to surprise attacks and sudden casualties for a short period. Technically, the impact and explosion generated by gunshots, bullets and shells are ideal acoustic sources for battlefield acoustic localisation technology.

Firearms produce unique acoustic waves while firing, and gun bullets also have unique shock waves. These acoustic signals can be used as training samples for LTPM. The LTPM-based acoustic localisation system locates shooters by matching acoustic signals generated by bullets and firearms with sample signals, as shown in Figure 4.3-2. After the shooter is located, a counterattack can be organised quickly to reduce casualties.
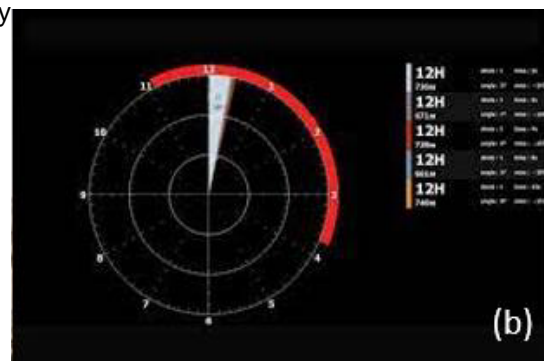
Figure 4.3-2 Illustration of battlefield acoustic localisation technology. Figure (a) shows a military jeep equipped with an acoustic localisation system. Figure (b) is the UI of the acoustic localisation system. This system quickly helps users to determine the direction of artillery fire [116]

## 4.4 Summary

The Location Fingerprint technology is referenced and exploited in this chapter. Unlike conventional TDOA-based localisation technologies, the Location Fingerprint technology is an electromagnetic pattern matching-based localisation technology. The acoustic wave-based LTPM is developed according to the localisation principles of Location Fingerprint.

The limitations of the classic Location Fingerprint localisation technology and the challenges of applying Location Fingerprint to acoustic localisation have been analysed and summarised. Traditional Location Fingerprint localisation technology relies on one specific signal feature such as RSS to locate mobile devices. However, in practical applications, the signal feature exposed a severe problem of sample overlapping, which leads to unclear mapping relations between the signal feature and locations. Therefore, multiple features are defined in the proposed pattern matching-based localisation technology.

As mentioned in Chapter 2, the multipath effect is always considered a negative effect. However, from the perspective of signal processing, the multipath effect also provides acoustic signals from different locations with unique features. LTPM utilises the acoustic features distorted by the multipath effect to achieve localisation and theoretically, the stronger the multipath effect is, the better the accuracy will be.

To solve the Thiessen polygon boundary problem caused by sample overlapping. Two representative machine learning algorithms are applied; Random Forest and Convolutional Neural Network. These machine learning algorithms perform high-dimensional classification and ensure accurate matching results. This is also the leading application of machine learning in three-dimensional acoustic localisation.

Notably, LTPM uses processed features as input for the training of positioning models. LTPM innovatively uses multi-dimensional signal features and signal images from different domains as training data. 43 signal features and 17 signal images are defined for Random Forest-based LTPM and CNN-based LTPM, respectively.

Overall, machine learning-based LTPM implementation consists of three steps, as shown in Figure 4.4-1: data collection, model training and model testing. In the data collection stage, acoustic samples are collected from pre-defined locations. The raw data is then processed with signal processing techniques to generate training datasets for machine learning algorithms. In the model training stage, the processed training datasets are loaded by machine learning algorithms for classification model training. In the model test stage, the system calculates the locations of input signals with the trained model.
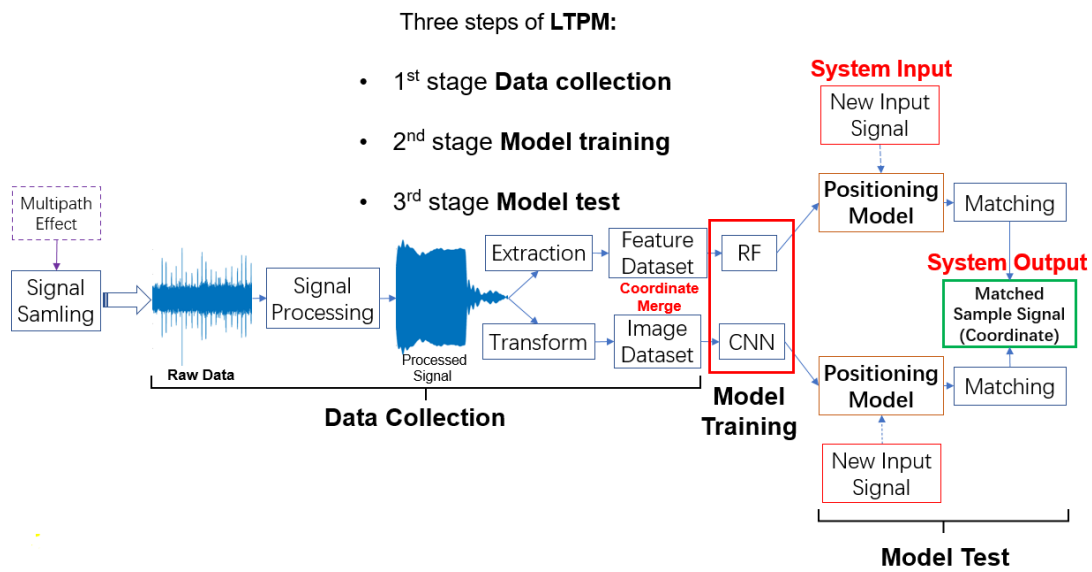


Figure 4.4-1 The implementation of LTPM. The input of the system is an acoustic signal collected by the sampling system, and the output of the system is the location of the acoustic signal source

Machine learning significantly extends the applicability of the pattern matching-based localisation technology. As a result, LTPM has robust environmental adaptability and it is capable of locating acoustic sources in complex indoor environments with one microphone. However, one disadvantage of LTPM is that samples from different locations must be collected in advance to ensure the positioning performance of LTPM. This disadvantage may affect the actual application of LTPM. Related solutions are discussed in Chapter 7.

# 5    Localisation on two-dimensional surfaces

This project aims to develop a pattern matching-based three-dimensional acoustic localisation technology which adapts to complex indoor environments while maintaining a three-dimensional accuracy of 200 mm - 500 mm. Since the preliminary research is used to validate the proposed localisation technology, system implementations and localisation tests are essential to this project.

The initial system aims to realise acoustic source localisation on two-dimensional surfaces. Acoustic waves caused by tapping or knocking propagate inside the solid object and are received by shock sensors deployed on the solid surface. Solid objects provide ideal propagation conditions for acoustic waves. Thus, details of the acoustic wave are well preserved, including signal components generated by the multipath effect.

LTPM is first applied to the two-dimensional localisation test. This chapter introduces the setups of the two-dimensional localisation system, the design of two-dimensional localisation tests and test results. The feasibility, accuracy, and constraints of LTPM are discussed in the result analysis and summary.

## 5.1    Test objectives

It is easier to locate an acoustic source on the surface of solid objects than to locate an acoustic source in the air, not only because of the lower dimensions but also because of the coexistence of transverse and longitudinal waves. The high signal-to-noise ratio and different acoustic propagation patterns (transverse and longitudinal waves) intensify the acoustic multipath effect.

The enhanced multipath effect amplifies the multipath signal components in the received acoustic signals, resulting in distinguishable differences between signals from different locations. These differences are utilised by the pattern matching-based localisation system to classify acoustic signals from different locations.

A deterministic cross-correlation matching algorithm is compiled to validate LTPM preliminarily since the number of samples is small (640 samples). All sample signals are

collected manually in the two-dimensional localisation test.

The deterministic cross-correlation matching algorithm calculates the correlation coefficient between an input signal and sample signals. The sample signal with the highest correlation coefficient will be determined as the output of the localisation system. Its coordinate is extracted as the final system output; thus, the location of the acoustic source is revealed.

The objectives of the two-dimensional localisation test are listed below:

(1) Build a complete localisation system. The system should have functionalities such as sample collection, signal processing, acoustic feature extraction and pattern matching.

(2) Compile a cross-correlation matching algorithm that matches input signals with template signals in the matching database in terms of three acoustic features.

(3) Design appropriate accuracy tests and illustrate the effectiveness of selected signal features with test results.

(4) Summarise the feasibility, the accuracy and the classification accuracy of the LTPM-based two-dimensional acoustic localisation system.

Therefore, a practical localisation system and appropriate tests for performance evaluation are required in the two-dimensional localisation test.

In the two-dimensional localisation test, acoustic signals are generated manually; a handheld stylus pen is used to tap on the surface of a glass plate. Besides, acoustic sources are always treated as mass points without practical volume in the test.

## 5.2 Introduction of system modules and electronic devices

### 5.2.1 The location template and coordinate setups

The location template is the tool for signal collection. In the two-dimensional localisation test, pre-defined locations for acoustic sources are labelled with the location template. Thus, the location template is crucial in signal acquisition, and it affects the accuracy of the localisation system.

In the two-dimensional localisation test, the location template is built by dividing the

surface of the target object. An initial two-dimensional localisation test is first conducted on a composite table. The table has a size of $1500 \times 870 \times 23$ mm. The acoustic impedance of the table is between 23.90 and 25.00 kg/(m$^2$s)$^3$. Two points (A and B) are labelled on the table, as shown in Figure 5.2-1. The distance between point A and point B is 500 mm.
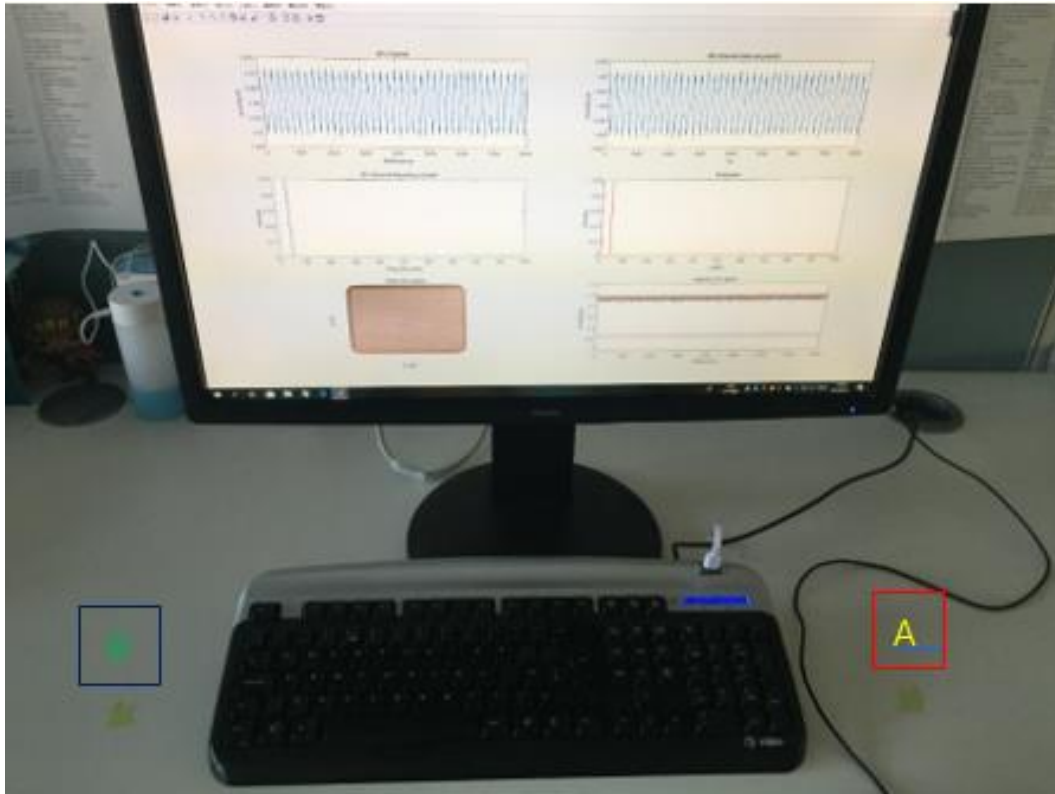


Figure 5.2-1 Illustration of the location template for the initial two-point localisation test. It was used in the early stage of the research to collect acoustic signals

The second localisation test is conducted on a rectangular glass plate. The dimensions of the glass plate are $400 \times 300 \times 4$ mm. Figure 5.2-2 illustrates the hardware deployment and the actual setups are shown in Figure 5.2-3.

64 grids are labelled on the glass plate, as shown in Figure 5.2-3. The horizontal and vertical labels are {1, 2, 3, 4, 5, 6, 7, 8} and {A, B, C, D, E, F, G, H}, respectively. These square grids are the areas for acoustic sources. Acoustic signals are generated by tapping or knocking on the labelled area. A stylus pen is used to interact with the glass plate to mitigate the differences caused by different tapping angles.

Acoustic waves generated by physical impact (tapping or knocking) propagate within the glass plate. The acoustic waves are then collected by a piezoelectric ceramic sensor glued to the corner of the glass plate. A buffer area between the sensor and the localisation region is set to balance different signal components. The closest grids are 100 mm from the sensor, ensuring evenly distributed multipath components.

Signals received by the piezoelectric sensor are electronic signals that must be amplified before sampling. Hence, a MAX 9814-based amplification module is designed for signal amplification. The output of the amplification module is maintained at 5.0 V with a single power supply. Details of each module are presented below.
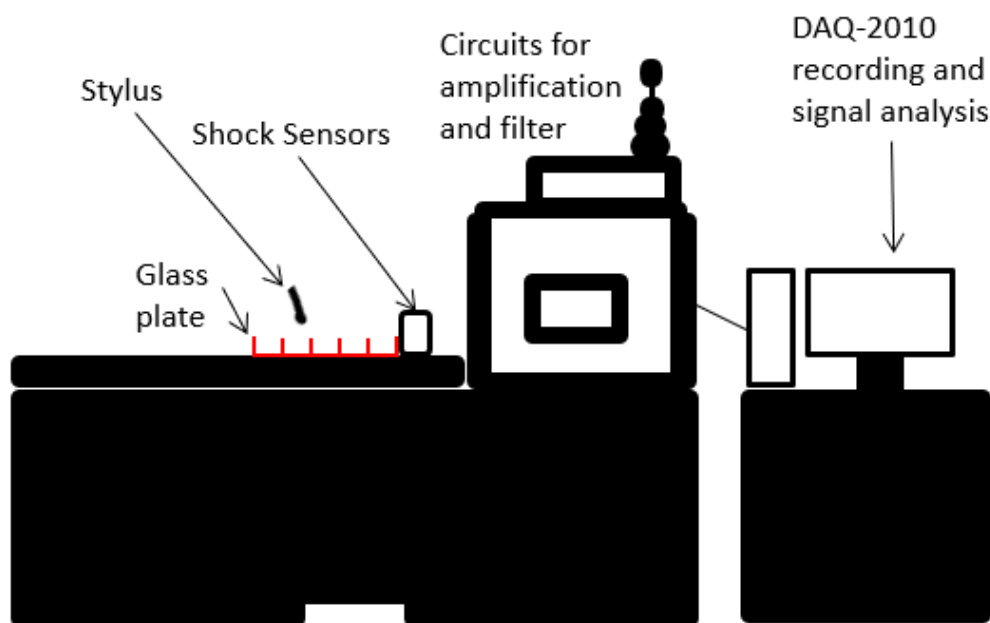


Figure 5.2-2 Illustration of the two-dimensional experimental platform
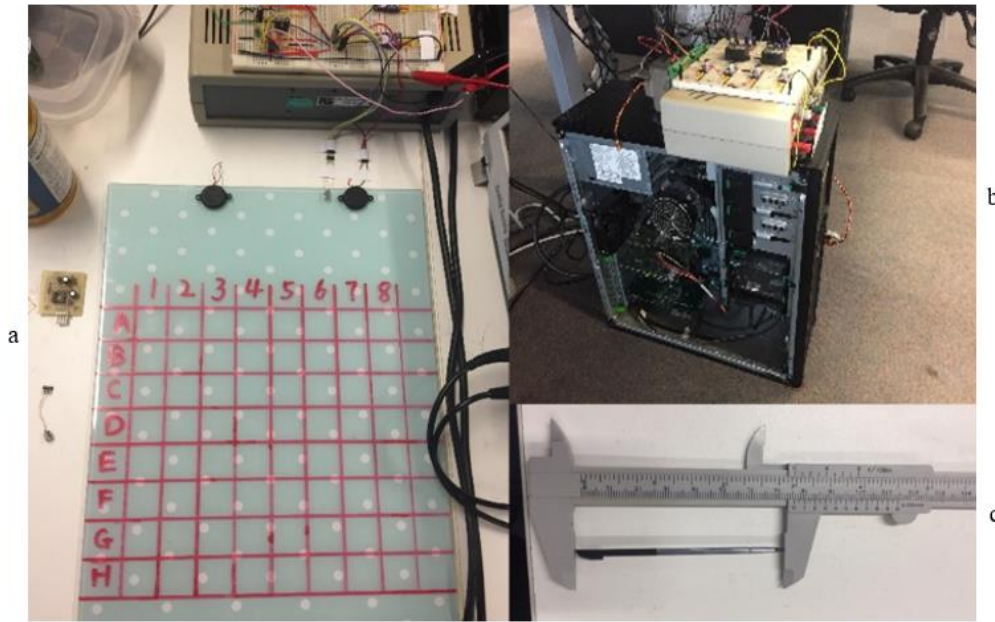
Figure 5.2-3 Location template for the two-dimensional localisation system (a). The glass plate has been divided into 8 × 8 grids, similar to a chess board. Each grid is a square with a side length of 30 mm. (b) shows the amplification module and sampling module. (c) is the stylus pen for tapping and knocking

### 5.2.2 The piezo electronic sensor and the amplification module

The piezoelectric vibration sensor is a sensitive sensor that transforms physical vibration energy into an electron signal. When a force is applied to a piezoelectric material, the surface of the piezoelectric material will generate an electrical voltage proportional to the magnitude of the force [117]. The characteristic of piezoelectric materials allows acoustic-electric transition; thus, any vibrations can be converted into electric signals.

The Murata PKS1-4A10 is a shock sensor with a single axis. It has a thickness of 4.5 mm and a diameter of 34.4 mm. In the two-dimensional localisation test, the sensor is glued to the surface of the glass plate for signal collection. A Murata PKS1-4A10 shock sensor is shown in Figure 5.2-4. The frequency response and the impact response of the shock sensor are shown in Figure 5.2-5.

Figure 5.2-4 Murata PKS1-4A10 piezo shock sensor. It has excellent electric performance and strong versatility. The sensor is widely used for vibration signal detection. Applications include security, strain instrumentation and wearable device

Signal processing begins with signal amplification. As a transmission channel, signal amplification is not the only objective for the signal amplification module. A qualified amplification module should also prevent signals from noise interference since noise can also be synchronously amplified [118]. Therefore, hardware RC filters are fitted into the amplification module to eliminate noises lower than 20 Hz. Meanwhile, it is important to note that the PKS1-4A10 shock sensor has an inherent 50 Hz vibration frequency. To facilitate subsequent signal processing, the 50 Hz frequency is filtered out with a digital filter. Besides, technical details such as coupling, grounding, slew rate, power supply, gain, bandwidth, and wiring are carefully considered to ensure the best amplification performance.

| (a) | (b) |
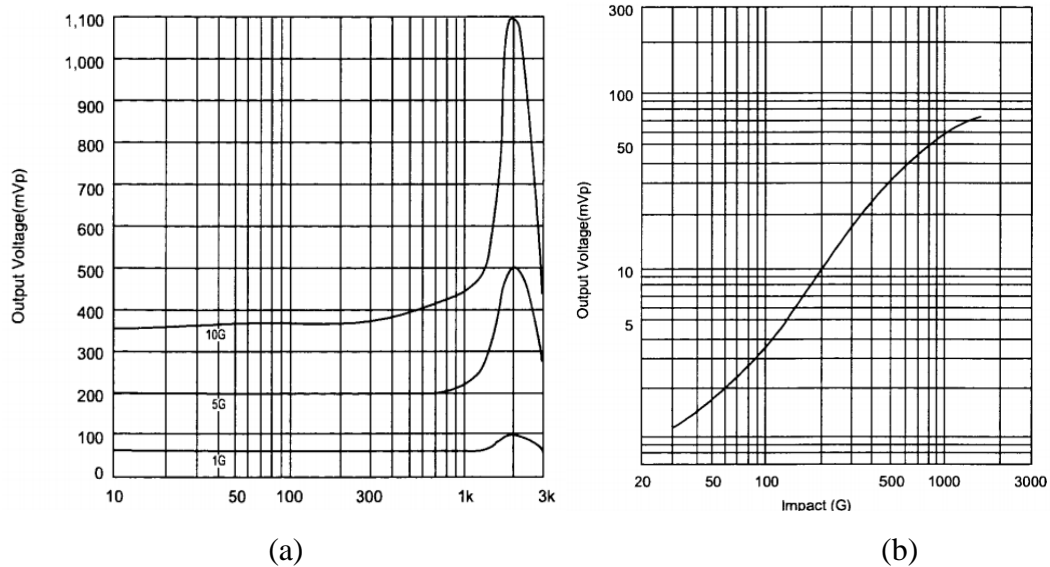
Figure 5.2-5 Frequency response and impact response of the PKS1-4A10 shock sensor. Murata PKS1-4A10 has a smooth impulse response curve and a balanced low-frequency response. But its frequency response varies greatly from 1K Hz to 3K Hz. However, since most of the acoustic frequencies generated by tapping and knocking are less than 1000Hz, PKS1-4A10 meets the test requirements; thus, it is selected as the vibration sensor for acoustic signal collection

As no commercial amplification modules are found to connect the PKS1-4A10 shock sensor with the DAQ-2010 data acquisition card, an amplification module is built in this project to amplify the signal received by the shock sensor and modulate the signal for sampling. The main amplifier is MAX9814. Characteristic curves of MAX9814 are shown in Figure 5.2-6, whereas the main features of MAX9814 are listed as follows:

- Functions: Amplification and de-noise

- With a pre-amp Auto Gain Control (AGC) circuitry

- Gain adjustable from 40 dB to 60 dB as well as the attack/release ratio

- input noise voltage: 30 nV/√Hz at 1k Hz

- Input impedance: 100k Ω

- Cut-off frequencies: 20 Hz – 20k Hz

- Power Supply Rejection Ratio (PSRR): 55 dB

- Signal to Noise Ratio (SNR): 61 dB

GAIN vs. FREQUENCY

MICBIAS NOISE vs. FREQUENCY
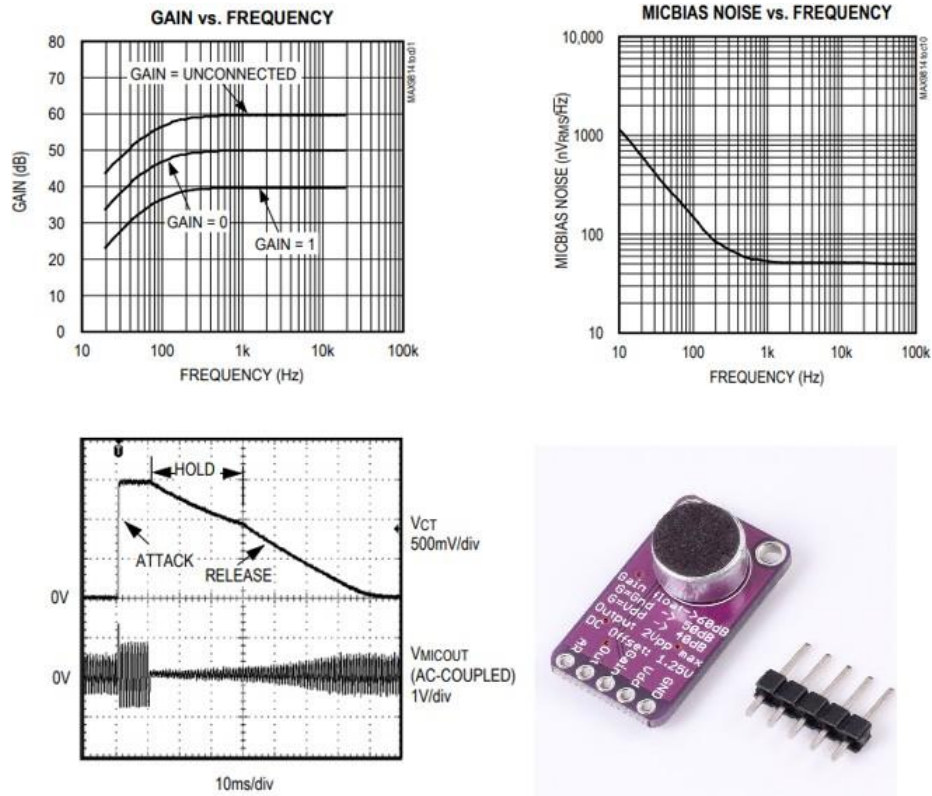


Figure 5.2-6 Features of MAX9814. The MAX9814 is a highly integrated amplifier. It amplifies the output of the Murata PKS1-4A10 for sampling, and it has a free gain control mode that could simplify the circuit design



(a)

(b)

(c)
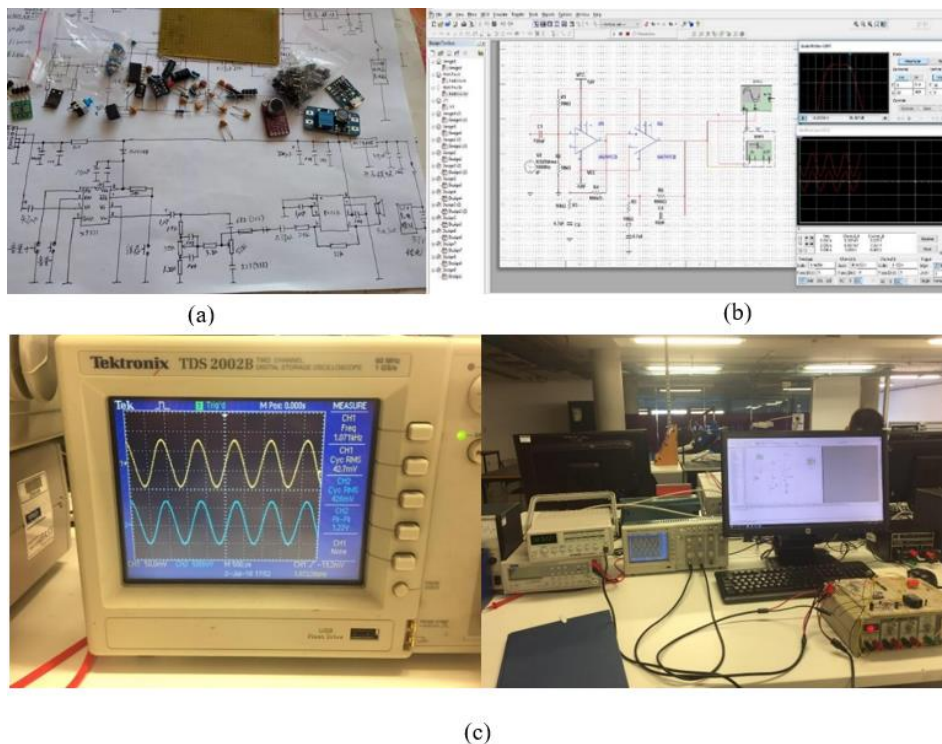
Figure 5.2-7 (a) The amplification circuit design. (b) Simulation of the designed circuit in NI Multism. (c) An amplification performance test

Figure 5.2-7 (a) shows the design of the amplification circuit followed by a simulation in NI Multism (b) and an actual performance test (c). The amplified signal meets the input requirements of DAQ-2010.

*5.2.3 The sampling module*

The sampling module is designed to sample analogue signals; an analogue signal acquired from the shock sensor is converted into a digital signal after sampling. The sampled digital signals are then sent to RAM directly through Direct Memory Access (DMA) for digital signal processing.



Figure 5.2-8 The DAQ-2010 data acquisition card. Parameters and functionalities such as voltage trigger, sampling rate, and pointer are adjustable through API, making it a perfect tool for particular scenarios that require acoustic sampling

Instead of using an audio card to sample acoustic signals, a data acquisition card DAQ-2010 is used for sampling because most commercial audio cards have integrated algorithms explicitly tailored for audible acoustic waves.

The sampling rates of on-sale audio cards are uncontrollable, and the internal algorithm integrated into the audio card may filter specific frequency bands out and change the sampled signals. To avoid this risk, a controllable sampling module is built. Unlike commercial on-sale audio cards, the data acquisition card DAQ-2010 has no internal algorithms that could potentially optimise sampled signals.

The sampling module is built based on a DAQ-2010 data acquisition card. The sampling module for the two-dimensional localisation test has a consistent frequency response across the 20 Hz - 20000 Hz frequency band. The parameters of the sampling module are adjustable for different localisation scenarios. Figure 5.2-8 shows the DAQ-2010 acquisition card used in the test.

The sampling rate is set to 10k Hz regarding the centroid frequency of tapping acoustic signals. The centroid frequencies of tapping acoustic signals are confirmed to vary from 300 Hz to 1000 Hz. Hence, according to the Nyquist sample theorem, the minimum sampling rate for the system should be twice the max frequency. The sampling rate should be higher than 2000 Hz at least. In the test, the sampling rate is set to 10k Hz to facilitate the calculation of sampling parameters. Moreover, the Murata PKS1-4A10 shock sensor has an inherent resonate frequency of 50 Hz. A multi-order Butterworth lowpass filter is applied to the sampling model to filter out the 50 Hz signal components.

Sampling and data transfer functions are compiled with ADLINK Application Programming Interface (API) in MATLAB. The drive code for DAQ-2010 consists of four parts: module initialisation, channel configuration, pointer assignment and data transfer. The drive codes are attached in Appendix B.

Real-time data monitoring is also integrated into the driver code for sampling surveillance. The real-time signal display and frequency response of the sampled signal are presented in a $3 \times 2$ grid panel, as shown in Figure 5.2-9.

The refresh rate of the panel equals the transfer rate of a single buffer under the double buffering mode. The sampled signals are stored in a pre-allocated self-extend matrix. The matrix transforms into a MAT file automatically after the sampling sequence completes. So far, the raw data collection has been completed, and the collected data is ready for further signal processing.
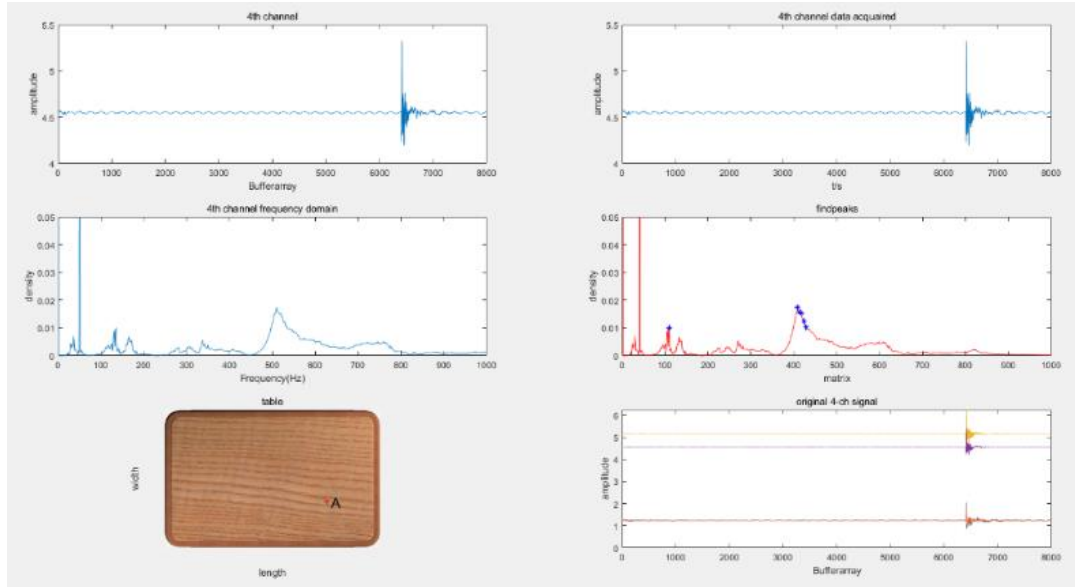
Figure 5.2-9 Real-time data display panel. Waveforms of the acoustic signal in the time and frequency domain can be observed in real-time with the panel

### 5.2.4 Signal processing: signal separation and coordinate merge

The signal collected by the sampling module is a huge matrix with quantised values. Thus, the matrix must be cut into independent signal arrays for coordinate merge and feature extraction operations.

Another reason for signal separation is that the acoustic signal received by the shock sensor is a variation of the signal source and environmental noises; thus, the received acoustic signal is always merged with background noises. Thus, acoustic signals need to be separated to reduce noise energy.

A digital energy density-based signal separation algorithm is developed to separate acoustic signals accurately. Figure 5.2-10 illustrates the steps of signal separation, and Figure 5.2-11 compares an unseparated signal sequence and a separated signal.

89
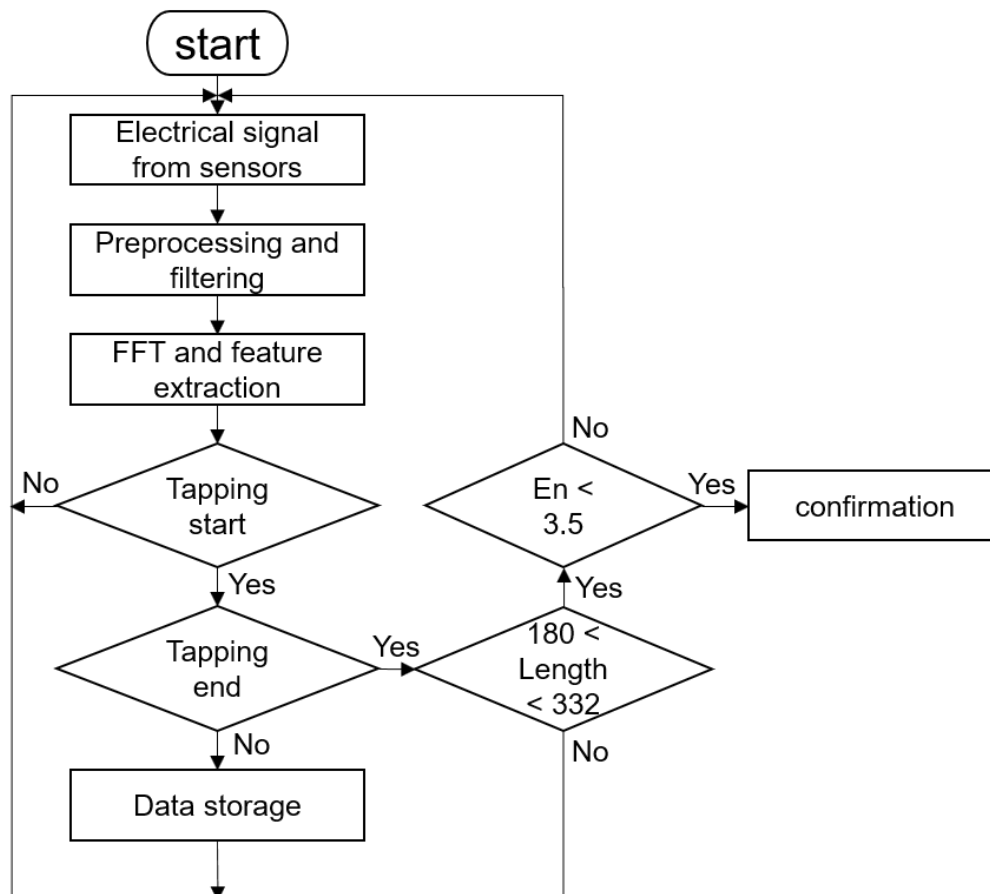
Figure 5.2-10 Signal separation illustration. En is the energy density acquired from the logarithmic (10log) power spectrum. The algorithm amplifies low amplitude components, revealing signal components which are masked in low amplitude noise for accurate signal separation

At the beginning of signal separation, the Fast Fourier Transform (FFT) is applied to extract the timings of peaks. These time points are recorded, and multiple small arrays are assigned to separate the signal sequence according to the recorded time points. These small arrays are intentionally extended to ensure enough space to cover each complete signal.

After initial separations, each array is examined with signal length and energy thresholds. An array is confirmed as a separated signal when the length of the array is between 180 and 332 points and the energy is between 1.2 and 3.5 dB. Arrays that pass the examination are identified as an independent signal sequence. At this point, acoustic signals are separated and stored in multiple arrays. Figure 5.2-11 (b) shows an illustration of a separated signal.
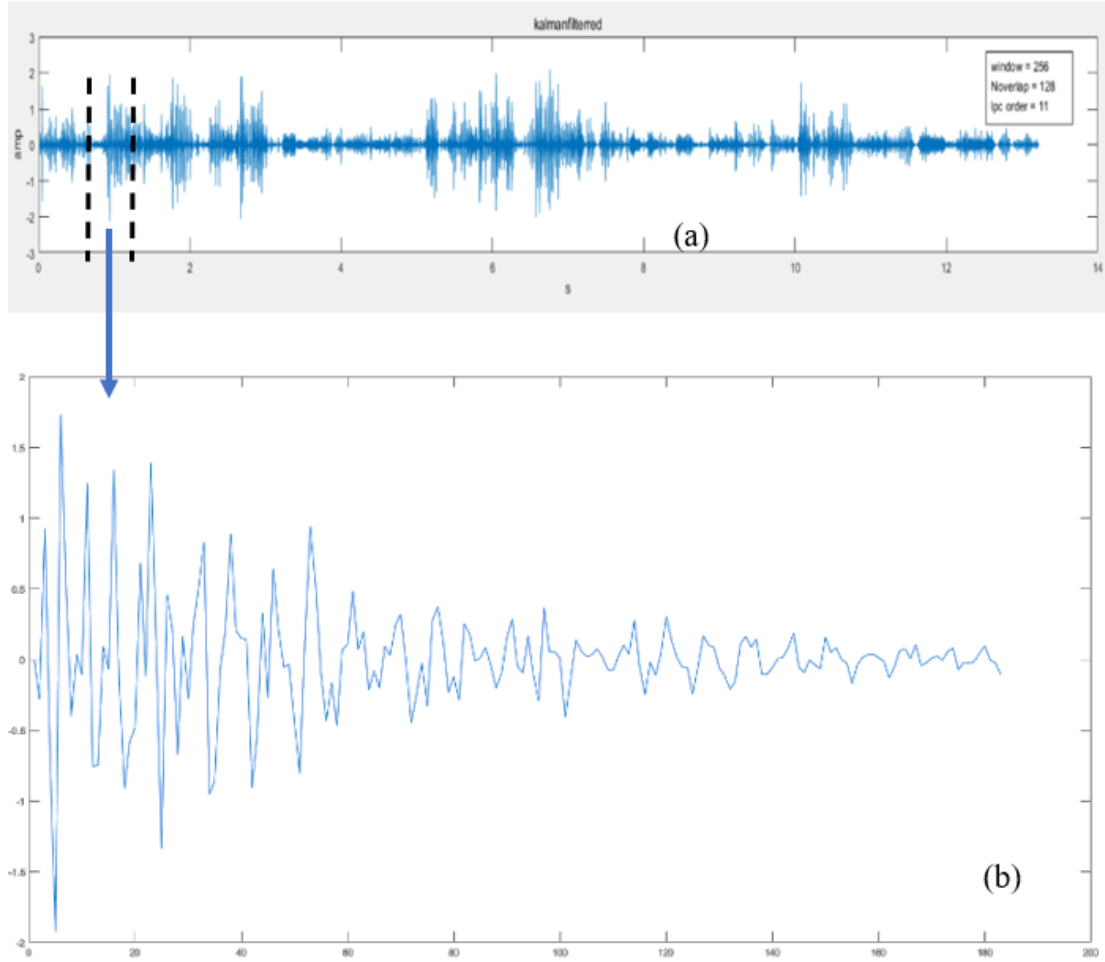
Figure 5.2-11 Illustration of a separated signal (b). (a) is a complete signal sequence obtained from the sampling module. The X-axis represents the number of sampling points, and the Y-axis represents the signal amplitude. The average signal length is around 250. The changing trend of the signal can be observed clearly after separation: each signal segment is excited quickly in the beginning. After reaching a maximum amplitude value, the signal gradually fluctuates until the vibration disappears

The coordinate is not added to the signal arrays directly. The separated acoustic signals are saved in folders labelled with the location tags {1, 2, 3, 4, 5, 6, 7, 8} and {A, B, C, D, E, F, G, H}. After a matched sample is acquired, the tag name of the folder to which the matched sample belongs will be identified as a system output. An advantage of using folder tags is that no extra coordinate arrays need to be added to the signal array, thus facilitating subsequent operations such as extraction and matching.

*5.2.5 Features for cross-correlation calculation*

In the two-dimensional localisation test, a deterministic cross-correlation matching

algorithm matches the input signal with the template signals. The cross-correlation matching algorithm is selected because the data volume is small (640 sample signals in the database). Whenever an input signal is provided to the algorithm, 640 correlation coefficients between the input signal and the template signals are calculated and the template signal with the highest correlation coefficient is determined as the matched signal.

In the signal processing, cross-correlation functions are usually used to examine the similarity between signals by comparing an unknown signal with a known signal [119]. The cross-correlation function describes the statistical correlation between two signals, while the correlation coefficient represents the similarity between the two signal series, as shown in Figure 5.2-12.

In the two-dimensional localisation test, the length of each signal is short (approximately 250 sampling points) due to the low sampling frequency. And the number of sample signals in the database is small (640 samples). Even if the cross-correlation calculation is performed between an input signal and each sample signal, the amount of computations per iteration is small. Thus, a deterministic cross-correlation matching algorithm is compiled to classify acoustic signals from different locations.
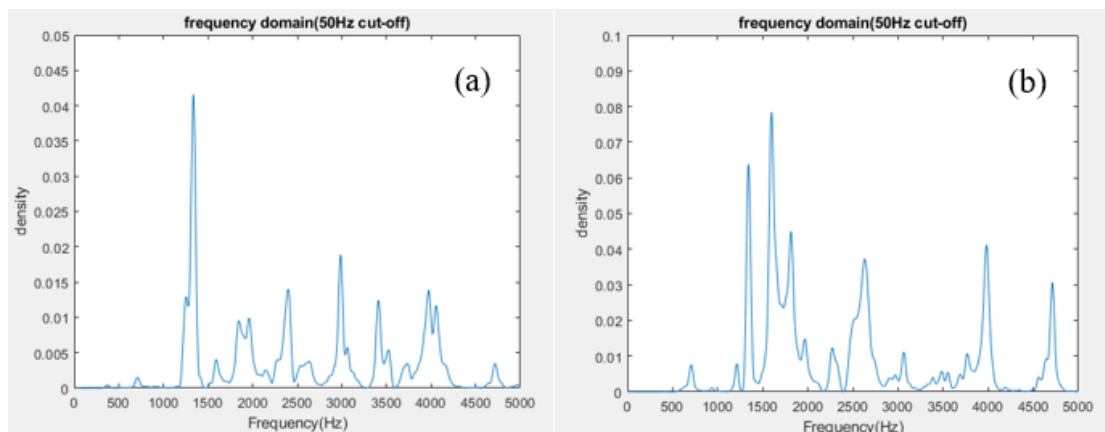


Figure 5.2-12 Comparison between two sample signals. (a) is an acoustic signal collected from A1, and (b) is an acoustic signal collected from H8. The correlation coefficient of these two signals is 0.694. These two signals have certain similarities, but many differences exist as well. The cross-correlation function is used to perform point-to-point comparisons and find accumulated differences in the form of the correlation coefficient

92

In addition to the signal correlation coefficient, the signal length from the time domain and the centroid frequency from the frequency domain are also utilised to improve the matching accuracy of the cross-correlation matching algorithm.

The length of a signal equals the sum of the signal's Direct Components (DC) sampling points and the Multipath Components (MC) sampling points. The direct components of an acoustic signal vary slightly at different locations, but the multipath components caused by the multipath effect vary greatly at different locations. An intuitive example is that the signal length decreases when an acoustic source is far from the sensor location. Therefore, signal length is selected to reflect the number of sampled signal components.

The centroid frequency describes the frequency of signal components in the spectrum [120]. It reflects the distribution of the signal power spectrum. A characteristic of the centroid frequency is that for a given frequency range, the energy contained in signal components lower than the centroid frequency is always half of the total energy of the signal. In summary, centroid frequency represents the energy distribution of a received signal. Therefore, the feature is selected to reflect the energy difference between different acoustic signals.

Signal length and centroid frequency are scalar features. The reason for choosing scalar features is to utilise changes in signal components caused by the multipath effect in the solid medium for localisation.

The matched sample's coordinate will be identified as the location coordinate of the signal source when the above three features (coefficient, signal length and centroid frequency) of an input signal are consistent with the three features of a sample signal in the database. The definition of signal length has been introduced in 4.2.1. Signal length represents the effective sampling points of the sampling module while centroid frequency is calculated with (Eq. 5-1):

$$\text{Centroid Frequency} = \frac{\int_0^\infty f S(f) df}{\int_0^\infty S(f) df} \qquad 5-1$$

where $S(f)$ is the amplitude corresponding to bin $f$ in the Fast Fourier Transform

(FFT) power spectrum.

Solid medium preserves the details of acoustic signals. The three features are sufficient to achieve a two-dimensional accuracy of 30 mm on solid surfaces.

*5.2.6 The matching algorithm*

The cross-correlation matching algorithm calculates the similarity between an input signal and a sample signal. The sample with the highest cross-correlation coefficient is determined as the matched template signal, and the matched sample's coordinate is the system output.

In the two-dimensional localisation test, features integrated into the cross-correlation matching algorithm are the distribution of frequency components, signal length and centroid frequency. The cross-correlation coefficient is calculated according to (Eq. 5-2).

$$R_{ij}(corrcoef) = \frac{C_{ij}}{\sqrt{C_{ii}C_{jj}}} \qquad\qquad 5-2$$

where $C_{ij}$ is the covariance of variables $C_i$ and $C_j$. $\sqrt{C_{ii}}$ is the standard deviation of $C_i$. $\sqrt{C_{jj}}$ is the standard deviation of $C_j$.

In practical cases, the variables $C_i$ and $C_j$ in the above formula are replaced with vectors because the point-to-point correlation calculation between two signal arrays is required. The equation listed above is therefore extended to (Eq. 5-3):

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{(\sum_m \sum_n (A_{mn} - \bar{A})^2)(\sum_m \sum_n (B_{mn} - \bar{B})^2)}} \qquad\qquad 5-3$$

where $\bar{A}$ and $\bar{B}$ are average values of vector $A$ and vector $B$. The operators $m$ and $n$ represent the dimensions of vector $A$ and vector $B$. The operator $r$ represents the correlation between the selected sample signal and the input signal. The higher the $r$ value is, the more relevant the two signals are.

The criteria of the cross-correlation coefficient are not academically unified yet. Table 5.2-1 provides a general reference for the effectiveness of the coefficient:

94

Table 5.2-1 Relations between coefficient level and extent of relativity

| Coefficient level | Extent of relativity |
|---|---|
| 0.00 - $\pm 0.30$ | Minor correlation |
| $\pm 0.30$ - $\pm 0.50$ | Positive correlation |
| $\pm 0.50$ - $\pm 0.80$ | Significant correlation |
| $\pm 0.80$ - $\pm 1.00$ | High correlation |

The outline of the acoustic signal-based cross-correlation matching algorithm is described in Table 5.2-2.

Table 5.2-2 Cross-correlation matching algorithm

System Input: Real-time electronic signal, $\overrightarrow{s_t}$

System Output: Grid tag of the location, $G_I$

1. System initialisation.
2. Define labels. For example, the 4 corner grids are labelled as follows;
   $G_a$ $(A1), G_b$ $(A8), G_c$ $(H1)$ $and$ $G_d$ $(H8)$.
3. Grid calibration and data transition. Each grid is correlated with 10 sample signals,
   $\overrightarrow{s_{1_1}}, \dots, \overrightarrow{s_{1_{10}}} \in G_a; \overrightarrow{s_{2_1}}, \dots, \overrightarrow{s_{2_{10}}} \in G_b; \dots etc.$
4. Calculate the 11th 'average' signal based on the loaded 10 sample signals,
   $\overline{s_a} \in G_a, \overline{s_b} \in G_b, \overline{s_c} \in G_c$ $and$ $\overline{s_d} \in G_d$.
5. Establish a matching database by applying Fast Fourier Transform (FFT) and feature extraction functions (signal length extraction and centroid frequency extraction) to each sample signal,
   $f_{a_1,1}, f_{a_1,2}, f_{a_1,3} \in F_{a_1} \dots f_{a_{10},1}, f_{a_{10},2}, f_{a_{10},3} \in F_{a_{10}}, \overline{f_{a,1}}, \overline{f_{a,2}}, \overline{f_{a,3}} \in$
   $\overline{F_a}$ $\dots$ $f_{d_1,1}, f_{d_1,2}, f_{d_1,3} \in F_{d_1} \dots f_{d_{10},1}, f_{d_{10},2}, f_{d_{10},3} \in F_{d_{10}}, \overline{f_{d,1}}, \overline{f_{d,2}}, \overline{f_{d,3}} \in \overline{F_d}. f_{x,1}$
   represents frequency components.
6. Real-time acoustic signal transmission, $\overrightarrow{s_t}$.
7. Apply FFT to $\overrightarrow{s_t}$.
8. Features extraction of $\overrightarrow{s_t} \to f_{t,1}, f_{t,2}, f_{t,3} \in F_t$.
9. For $f_{t,1} \in F_t$, where $f_{t,1}$ is frequency components:
   a. Calculate cross-correlation coefficients $(c_a, c_b, c_c, c_d)$ between $f_{t,1}$ against $\overline{f_{a,1}}, \overline{f_{b,1}}, \overline{f_{c,1}}$, and $\overline{f_{d,1}}$ respectively.
   b. Find $max(c_a, c_b, c_c, c_d)$.
   c. Store $G_j | f_1$ where $j$ is the grid label of the max coefficient.
10. For $f_{t,2} \in F_t$, where $f_2$ is the signal length:
    a. Calculate signal length ranges of $f_{a_i,2}$, $f_{b_i,2}$, $f_{c_i,2}$, and $f_{d_i,2}$,
       $[min(f_{a_i,2}), max(f_{a_i,2})], [min(f_{b_i,2}), max(f_{b_i,2})], [min(f_{c_i,2}), max(f_{c_i,2})] \dots$
       where $i = 1, \dots, 10$.
    b. Compare $f_{t,2}$ against
       $[min(f_{a_i,2}), max(f_{a_i,2})], [min(f_{b_i,2}), max(f_{b_i,2})], [min(f_{c_i,2}), max(f_{c_i,2})] \dots$
       where $i = 1, \dots, 10$.

c. If $f_{t,2} \in [min(f_{j_i,2}), max(f_{j_i,2})]$ & $f_{t,2} \notin [min(f_{k_i,2}), max(f_{k_i,2})]$ for $j = a, b, c \text{ or } d$ and $\forall k \neq j$, store $G_j|f_2$ where $j$ is the grid label of the matched signal length range.

d. Else

   i. Calculate difference between $f_{t,2}$ and $ave(f_{a_i,2}), ave(f_{b_i,2}), ave(f_{c_i,2})$ and $ave(f_{d_i,2})$.

   ii. Store $G_j|f_2$, where $j$ is the grid label of the $f_{x,2}$ with the minimum difference.

11. For $f_{t,3} \in F_t$, where $f_3$ is the centroid frequency:

a. Calculate the centroid frequency ranges of $f_{a_i,3}$, $f_{b_i,3}$, $f_{c_i,3}$, and $f_{d_i,3}$, $[min(f_{a_i,3}), max(f_{a_i,3})], [min(f_{b_i,3}), max(f_{b_i,3})], [min(f_{c_i,3}), max(f_{c_i,3})] \ldots$ where $i = 1, \ldots, 10$.

b. Compare $f_{t,3}$ against $[min(f_{a_i,3}), max(f_{a_i,3})], [min(f_{b_i,3}), max(f_{b_i,3})], [min(f_{c_i,3}), max(f_{c_i,3})] \ldots$ where $i = 1, \ldots, 10$.

c. If $f_{t,3} \in [min(f_{j_i,3}), max(f_{j_i,3})]$ & $f_{t,3} \notin [min(f_{k_i,3}), max(f_{k_i,3})]$ for $j = a, b, c \text{ or } d$ and $\forall k \neq j$, store $G_j|f_3$ where $j$ is the grid label of the matched signal length range.

d. Else

   i. Calculate difference between $f_{t,3}$ and $ave(f_{a_i,3}), ave(f_{b_i,3}), ave(f_{c_i,3})$ and $ave(f_{d_i,3})$.

   ii. Store $G_j|f_3$, where $j$ is the grid label of the $f_{x,3}$ with the minimum difference.

12. If $G_j|f_1 \neq G_j|f_2 \neq G_j|f_3$

a. Calculate cross-correlation coefficients between $f_{t,1}$ and all sample signals.

b. Find the maximum coefficient and its grid label.

c. Output $G_j|f_1$, where $j$ is the grid label.

13. Else

   Output $mode(G_j|f_1, G_j|f_2, G_j|f_3)$

14. End

Firstly, the system's output is defined by determining the positioning area (grid labels) and loading corresponding sample signals. Then a matching database is established to extract features from the loaded signals. $f_{x,1}$ is the frequency component of a signal, $f_{x,2}$ is the length of the signal, while $f_{x,3}$ is the centroid frequency of the signal. To this extent, the matching database is ready for matching. The corresponding steps are 1-5.

The real-time signal $\vec{s_t}$ is a system input. FFT is applied to extract frequency components $f_{t,1}$, signal length $f_{t,2}$ and centroid frequency $f_{t,3}$ of $\vec{s_t}$. The cross-correlation coefficient between $f_{t,1}$ and $f_{x,1}$ is calculated first. The grid label of the sample corresponding to the maximum cross-correlation coefficient is stored in $G_j|f_1$.

96

The corresponding steps are 6-9.

Next is the matching of signal lengths. Firstly, at each labelled grid, the range of signal length is confirmed by searching for the minimum and maximum signal length values of the 10 sample signals. If the input signal length $f_{t,2}$ of $\vec{s_t}$ only belongs to a known signal length range, the corresponding grid label of the matched range is stored in $G_j|f_2$. If the signal length $f_{t,2}$ of $\vec{s_t}$ belongs to multiple known signal length ranges, or no matched range is found, $f_{t,2}$ will then be compared with the average signal length of the 10 sample signals of each grid. The grid of the average signal length with the minimum difference is stored in $G_j|f_2$. The corresponding step is 10.

Next is the matching of centroid frequencies. Similar to matching signal lengths, the first step is to determine the range of centroid frequency by searching for the minimum and maximum centroid frequency values of the 10 sample signals. When the centroid frequency $f_{t,3}$ of $\vec{s_t}$ only belongs to a known centroid frequency range, the corresponding grid label of the range is stored in $G_j|f_3$. And if the centroid frequency $f_{t,3}$ of $\vec{s_t}$ belongs to multiple known centroid frequency ranges, or no matched range is found, $f_{t,3}$ will be compared with the average centroid frequency of the 10 sample signals of each grid. The grid of the average signal length with the minimum difference is stored in $G_j|f_3$. The corresponding step is 11.

In the case that the 3 grid labels $G_j|f_1, G_j|f_2, G_j|f_3$ have different values, recalculate the cross-correlation coefficients between $f_{t,1}$ and all sample signals. The grid label of the sample signal with the highest cross-correlation coefficient is selected as the final output of the system. While in the case that the 3 grid labels $G_j|f_1, G_j|f_2, G_j|f_3$ have 2 or 3 identical values, the grid label with the most occurrences are selected as the final output of the system. The corresponding steps are 12-14.

The two-dimensional localisation algorithm is a deterministic pattern matching algorithm which determines the location of the acoustic source signal according to the three defined features. Although only 640 sample signals were collected, the localisation system still achieved high accuracy on the surface of a glass plate. The hardware platform and

applicable algorithms required for the test have been built and compiled to this extent. Next, the test design and test results are presented.

## 5.3  Two-dimensional localisation test design

The two-point localisation test is proposed against the aim and objectives of the research to lay a foundation for the multiple points localisation test. After the performance of the proposed localisation technology and the effectiveness of the pattern matching approach are examined preliminarily, a more complex and comprehensive multiple points localisation test is designed.

In addition, a stylus pen is used to tap at the pre-defined area on the composite table and the glass plate. Vibrations received by the shock sensor are input signals for the localisation system, while the system output is the coordinate of the acoustic source.

### 5.3.1 Two-point test

Electromagnetic localisation technologies have been widely applied and stable progress has also been made in the development of acoustic wave-based localisation technology. In the EU project TAI-CHI [7], pattern matching is combined with TDOA, time-reversal and predictive algorithm to achieve short-range acoustic localisation on solid surfaces. Test results of the TAI-CHI project validated the feasibility of acoustic localisation on two-dimensional surfaces.

Inspired by the TAI-CHI project and the two-dimensional localisation test achieved by Ze J [105], a simple two-dimensional localisation test is designed to determine the localisation performance of the cross-correlation matching algorithm. The two-point (A and B) localisation test is conducted on a composite table, and the distance between point A and point B is set to 500 mm, as shown in Figure 5.3-1.
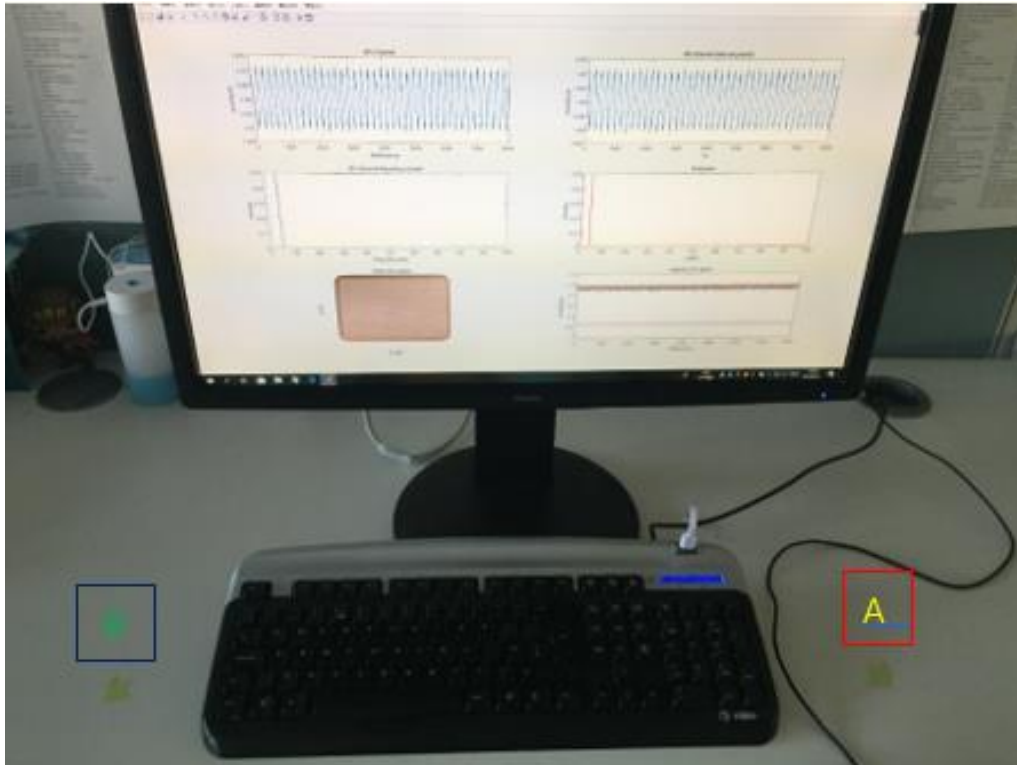
Figure 5.3-1 The two points on the table. The pattern matching-based localisation algorithm is used to classify acoustic signals from the two physical locations. In the test, only the frequency distribution of the signal is used as the matching object, and the matching database contains 20 sample signals

Differences in frequency distributions between points A and B can be observed in Figure 5.3-2. The matching algorithm uses the differences to locate acoustic sources. By tapping points A and B, the classification accuracy of the system is verified. The localisation system has a two-dimensional accuracy of 500 mm in 96% of the location estimates.

The localisation system miscalculated the location 4 times out of 100 tapping tests. In conclusion, it is proven that the acoustic source at different locations has different waveforms and frequency distributions in terms of received acoustic signals. The test result verified the system functionalities and the two-point test laid a solid foundation for multiple points localisation test.
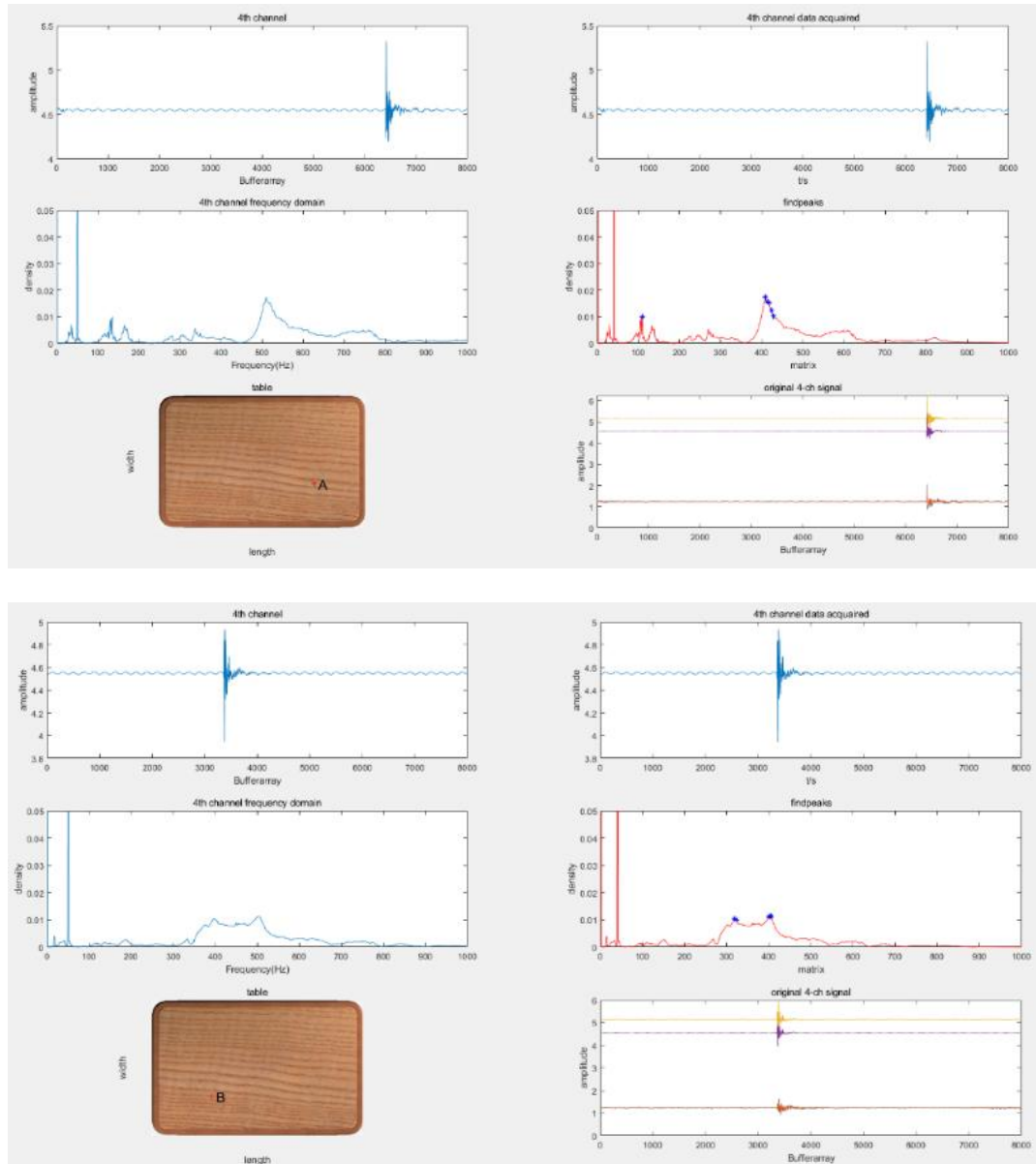
Figure 5.3-2 The frequency distribution comparison between point A and point B. The waveform of the acoustic signal at location A is similar to the waveform of the acoustic signal at location B. But the frequency distributions of acoustic signals from different locations are different due to the multipath effect; LTPM utilises the differences to distinguish signals from A and B in the two-dimensional localisation test

## 5.3.2 Multiple points test

The design principle of the two-dimensional localisation test is to deliver the research objectives with a simple and clear test method. Since the surface of the glass plate is divided into 64 grids, localisation tests on every single grid would lead to a huge workload, making the localisation results difficult to explain and understand. Therefore,

a special test method is selected to reduce the workload burden while providing scientific test results.

The multiple points localisation test is developed on the basis of the two-point localisation test. The medium is replaced with a glass plate. The dimensions of the glass plate are $400 \times 300 \times 4$ mm. 64 points (grids) are labelled on the surface of the glass plate. A special test method is therefore designed due to the increased points. The test method is developed from the Taguchi methods.

The Taguchi method is a statistical methodology proposed by the Japanese statistician Fangfu Taguchi in 1950 [121]. The initial purpose of the Taguchi method is to improve product quality, product reliability and functionality and reduce production costs by managing and controlling variables during production. The statistical methodology improves the quality of products effectively in many practical cases. In engineering, the Taguchi method has been gradually developed into a test method which uses a minimum number of tests to illustrate all the factors affecting the performance of a system [122]. A Taguchi method-based diagonal progressive test strategy is designed to examine the relation between the positioning accuracy and the classification accuracy, as shown in Figure 5.3-3.

Figure 5.3-3 illustrates the Taguchi method-based test design, and Figure 5.3-5 shows the physical setups of the two-dimensional multiple points test.
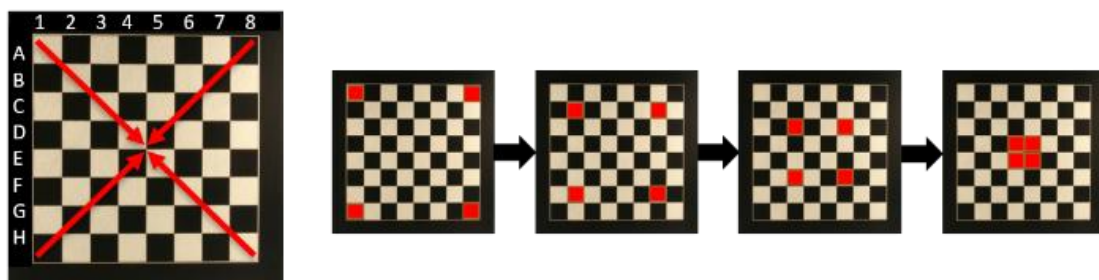


Figure 5.3-3 Taguchi chessboard for the two-dimensional localisation test. The progressive interval is set to 30 mm. The first-round test begins with four corner grids. Then the test indents along the diagonal line until the four test grids meet in the central area

A series of converging localisation tests are designed to measure the classification accuracy of the localisation system. The test begins with the four corner grids on the

edges of the glass plate and then progresses towards the central grids on the glass plate along the diagonal lines until the four grids meet each other in the centre of the glass plate. The classification accuracy is calculated with (Eq. 5-4).

$$Accuracy = \frac{TP}{TP + FP} * 100\% \qquad 5-4$$

where TP stands for True Positive while FP stands for False Positive, the accuracy rate equals the number of correctly classified samples divided by the total number of test samples.

At the beginning of the test, the test is conducted at four corner test points. The four points have the farthest distance from each other. Theoretically, the localisation system should have a high classification accuracy because acoustic signals from four locations have unique signal features. As the localisation test progresses, the grid used for the accuracy test will progress diagonally and eventually reach the four central grids (D4, D5, E4, E5, as shown in Figure 5.3-3).

Predictably, acoustic signals generated at the four central grids are similar, resulting in decreased classification accuracy. Thereby, localisation tests on all grids are unnecessary so that confusion and difficulties in all grids tests are avoided. Instead, the classification accuracy of the system is determined in several rounds of diagonal progressive tests.

<u>Sample collection</u>:

Each grid is randomly tapped 10 times, and the acoustic signals generated by tapping are sampled and stored in the matching database. Each marked grid corresponds to a signal cluster consisting of 10 tapping signals. A tapping signal collected from A1 and a tapping signal collected from H8 are shown in Figure 5.3-4 as an illustration.

To prevent data overflow and system asynchronism from happening, time control functions are used to regulate computational resources consumed by each function and call for timeout when an error occurs.

As introduced above, the test starts with the four corner grids, A1, A8, H1, and H8, and then indents along the diagonal line. B2, B7, G2 and G7 are tapped in the 2nd round test,

while C3, C6, F3 and F6 are tapped in the 3rd round test.

After reaching the central area, the localisation system still has a classification accuracy of 83%. Thus the four grids in the central area are further divided into grids with 15 mm side lengths for further localisation tests, as shown in Figure 5.3-5 (blue arrows). In total, six rounds of progressive tests corresponding to distance {210 mm, 150 mm, 90 mm, 45 mm, 30 mm, 15 mm} are conducted.
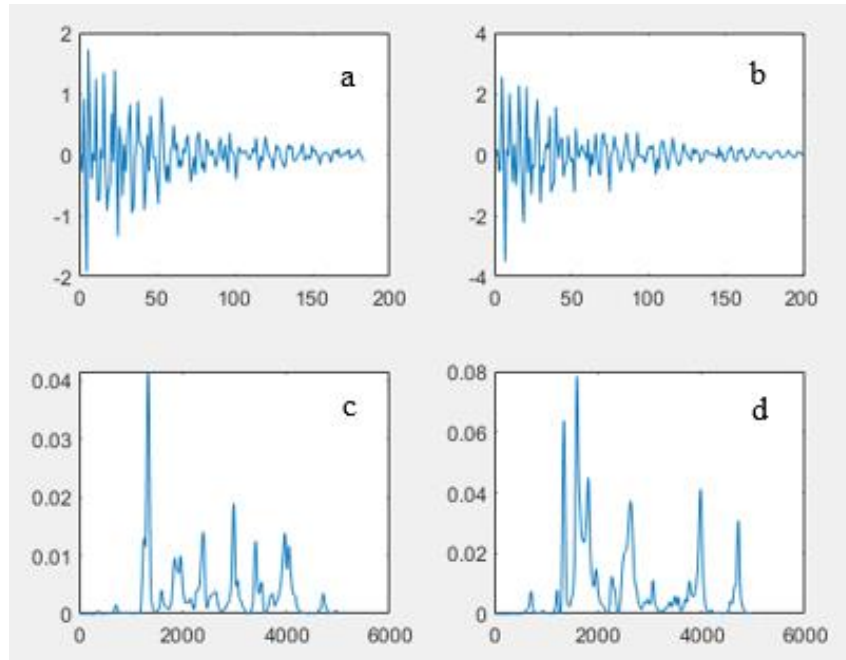


Figure 5.3-4 Signal illustration. (a) is the signal collected from location A1 in the time domain. The X-axis represents the sampling points, and the Y-axis is the amplitude of the signal. While (c) is the same signal but in the frequency domain. The X-axis in (c) represents frequency, and the Y-axis in (c) represents energy density. (b) is a signal collected from H8 in the time domain, and (d) is the same signal in the frequency domain. Acoustic frequency components in the frequency domain are different though their waveforms in the time domain are similar
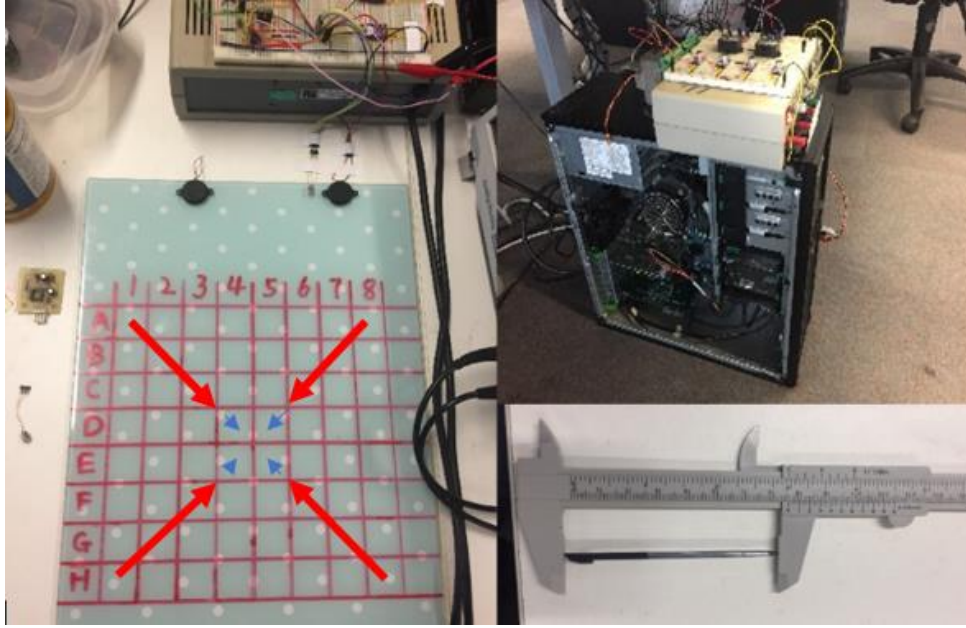
Figure 5.3-5 Physical setups of the two-dimensional multiple points accuracy test. The glass plate has been divided into 8 × 8 grids, similar to a chess board. Each grid is square with dimensions of 30 × 30 mm. The sensor on the right is activated for signal reception, and a stylus pen from HTC Touch is used for tapping

## 5.4    Result analysis

The localisation performance of the two-dimensional localisation system is evaluated in this section. The test results of the six rounds of progressive tests corresponding to distance {210 mm, 150 mm, 90 mm, 45 mm, 30 mm, 15 mm} are shown in Figure 5.4-1. The overall test result indicates that the system has a two-dimensional accuracy of 30 mm in 85% of the location estimates.

As expected, the classification accuracy varies with the localisation distance. The two-dimensional localisation system has a two-dimensional accuracy of 210 mm in 91% of the location estimates. The classification accuracy decreases to approximately 85% when the two-dimensional accuracy is set to 150 mm. The classification accuracy decreases to 78% when the two-dimensional accuracy is set to 30 mm. A conclusion can be drawn that the connections between the three acoustic features and locations are disrupted when the two-dimensional accuracy is within 30 mm.

Two main factors affect the decrease in classification accuracy. The first factor is the distance between the four grids. As shown in Figure 5.3.5, the four grids in the centre

have adjacent edges. In this situation, the acoustic waves emitted by tapping within these four grids will have similar frequency distributions and feature values, resulting in a decreased classification accuracy. The second factor is the decrease in the matching accuracy of the localisation algorithm. From the algorithm's perspective, internal factors, such as the parameters of the algorithm and feature extraction functions, also affect the positioning performance of the localisation system. For instance, in the compilation of the algorithm, the coefficient of $c_a$ and $c_b$ are always similar to $c_c$ and $c_d$; thus, four empirical values $\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4$ are set to offer preferences when the four coefficients of four grids have approximate values. In the two-dimensional localisation test, such empirical algorithm parameters cannot be adaptively adjusted; thus, the classification accuracy will inevitably decrease with the decrease in the distance between grids.

Overall, the system has reached a two-dimensional accuracy of 30 mm in 80% of the location estimates on the surface of a glass plate.
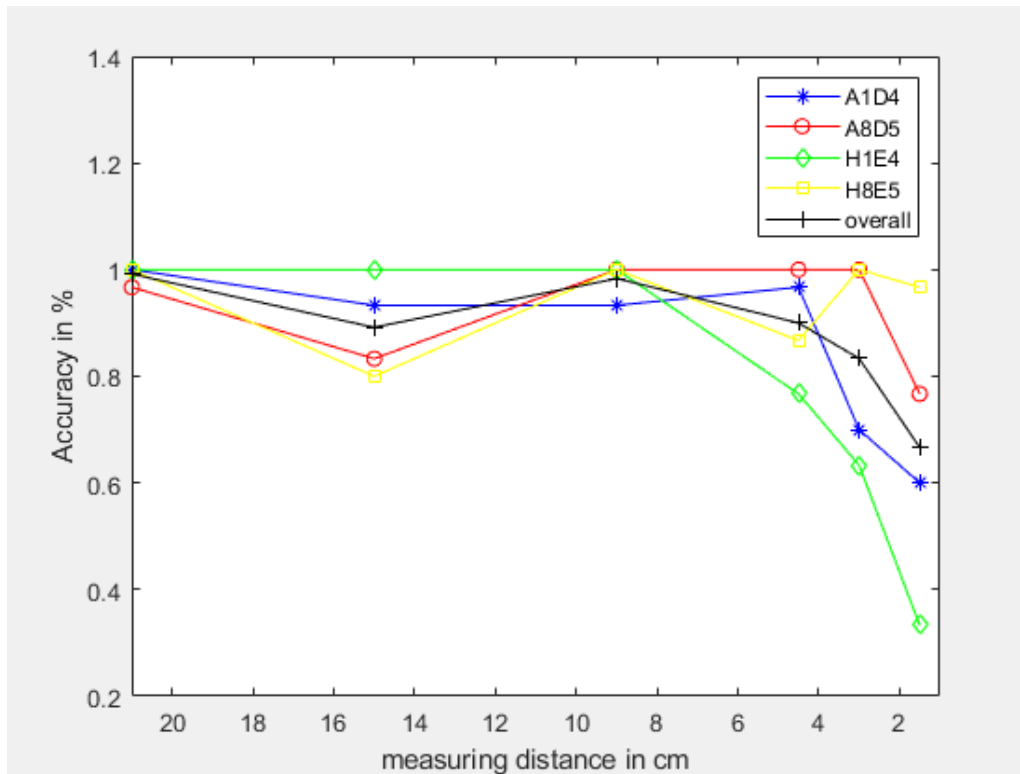


Figure 5.4-1 Two-dimensional test results. The X-axis represents the positioning accuracy; the Y-axis represents the classification accuracy. The classification accuracy of the system decreases as the positioning accuracy decreases

Compared to the two-point A/B localisation test, two features (signal length and centroid frequency) are integrated into the localisation system. The test result verifies the feasibility of pattern matching-based acoustic localisation on the surface of solid objects.

What's more, it appears that the phase spectrum of the acoustic signal contains more information than the power spectrum. The proof is that the time domain signal restored with the phase spectrum has a higher correlation coefficient than the signal restored with the power spectrum. To make use of this phenomenon, phase-related features are added to the three-dimensional localisation system to improve the signal discrimination ability.

## 5.5  Summary

The preliminary exploration of short-range acoustic pattern matching localisation has been accomplished in this chapter. In the test, a glass plate is equidistantly divided into 64 grids. Each grid is assigned with a corresponding tag number. Template signals and input signals are generated by tapping on these grids. The localisation system captures acoustic signals and matches the captured signals with template signals. When a matched template signal is determined, the coordinate number of the matched sample signal will be identified as the system output.

The sensor used in the two-dimensional localisation test is a Murata PKS1-4A10 shock sensor. The output of Murata PKS1-4A10 is amplified and filtered by a MAX9814 amplification module. A controllable sampling module then samples amplified signals. The sampling module is built based on a DAQ-2010 data acquisition card, and the sampling rate is set to 10,000 Hz. The sampled signals are sent to the computer memory directly for matching processing.

The format of the collected signal is a multi-dimensional matrix; thus, signals in the matrix must be separated for signal processing. An energy density and signal length-based separation algorithm is developed for the signal separation. The separated signals are then stored in a location-named folder and waiting to be retrieved.

The deterministic cross-correlation matching algorithm is selected as the classification algorithm due to the small sample size (640 samples). The frequency distribution of the

acoustic signal, signal length, and centroid frequency are used to classify acoustic signals. Thus, the cross-correlation matching algorithm is developed to match input signals with template signals based on the three features.

The localisation system is first applied to the two-point test. The two-point test aims to examine the functionality of each module and collect acoustic signals, and preliminarily verify the feasibility of LTPM. The distance between point A and point B is 500 mm in the two-point test and the localisation system has a classification accuracy of 96%.

A Taguchi method-based test method is developed for the 64-point localisation test. A location template similar to a chess board is designed, and localisation tests are conducted following the diagonal indentation principle from the far end to the near end. The multiple points test consists of 6 rounds of tests.

The following objectives are accomplished in the two-dimensional localisation test;

(1) A complete signal processing system is built. The system meets the test requirements from sampling to signal matching. Technical problems in acoustic localisation are preliminarily solved.

(2) A cross-correlation matching algorithm is compiled and applied to the localisation system. It classifies acoustic signals from different physical locations according to three defined features.

(3) A special test methodology is developed according to the Taguchi method. The test methodology examines the classification accuracy of the localisation system and avoids the hassle of repetitive tests.

(4) The validity of the selected features is verified. Test results indicate that the pattern matching-based localisation system achieved an accuracy of 30 mm in 80% of the location estimates.

# 6    Localisation in the three-dimensional space

Speech, limb collisions, finger clicks, clapping, tapping and knocking are all available acoustic signals for short-range acoustic localisation. Similar to the TDOA-based electromagnetic localisation and inertia orientation localisation technologies, short-range acoustic localisation technology can be used to support the interaction between users and computers. In this chapter, an LTPM-based three-dimensional acoustic localisation system is designed and tested from the aspect of acoustic human-computer interaction.

In the two-dimensional test, three signal features and a cross-correlation pattern matching algorithm are used to achieve the short-range acoustic localisation on solid surfaces. Excellent test results are acquired due to the inconsistency of acoustic features and the high signal-to-noise ratio. However, the matching algorithm used in the two-dimensional localisation test is a deterministic correlation algorithm, which is inappropriate for the three-dimensional acoustic localisation since the number of samples in the matching database has increased to 120,000, and the number of acoustic features has increased to 43. In this case, a high-performance classification algorithm is required to classify acoustic signals from different three-dimensional locations.

Two machine learning algorithms are utilised to classify acoustic signals from different locations. In traditional signal matching, sample signals are compared with template signals then the differences are summarised artificially and compiled into the matching algorithm to ensure accurate matching results. Unlike the traditional deterministic signal matching algorithms, machine learning algorithms dig into the training data deeply and automatically summarise signal patterns in the form of classification models.

Machine learning has been applied to different fields, especially in computer vision and image processing. It analyses factors that affect visual perception and utilised the factors for image processing and image classification [84]. Machine learning has been widely applied. In image processing, images modified by machine learning algorithms are so vivid that even a human being is hard to tell if the processed image is true [92].

Although powerful machine learning algorithms have been applied to various fields,

machine learning has not been applied to acoustic localisation. In this project, machine learning algorithms are integrated into the LTPM-based acoustic localisation system to classify acoustic sources at different spatial locations. The Random Forest (RF)-based localisation system and the Convolutional Neural Network (CNN)-based localisation system have achieved excellent short-range acoustic localisation results. Below, the objectives of the three-dimensional test are presented, followed by the introduction of different system modules, test design, test results and a comprehensive summary.

## 6.1 Test objectives

In machine learning, massive data are required to train a classification model. Therefore, a robot-based sampling system is built to collect massive acoustic signals.

A robot is programmed to move an acoustic source to designated three-dimensional locations. A microphone-based sampling system then detects acoustic waves generated by the acoustic source. The sampled acoustic signals (training data) are then processed and stored in a matching database.

Next, the signal processing functions are executed to extract signal features. The extracted signal features and the coordinates of sampled signals are then processed as the training data for machine learning algorithms.

Two machine learning algorithms are utilised to achieve three-dimensional localisation. Similar to the objectives listed for the two-dimensional localisation test, the objectives of the three-dimensional localisation test are summarised as follows:

(1) Design and build an acoustic signal acquisition and a signal processing system for three-dimensional acoustic signal collection.

(2) Collect adequate samples (100,000 to 150,000 signals) for the training of classification models.

(3) Extract acoustic features with signal processing techniques to form multi-dimensional input data and compile machine learning algorithms.

(4) Train Location Templated-based Positioning Models (LTPM) with different training datasets and evaluate the positioning performance of the trained model.

109

(5) Summarise the characteristics and properties of LTPM.

In this chapter, rigorous tests are designed to evaluate the performance of LTPM. In addition, acoustic sources are always approximately treated as mass points without practical volumes in the three-dimensional localisation test.

## 6.2 Introduction of system modules and electronic equipment

The acoustic signal acquisition in the three-dimensional space is divided into three steps:
(1) Design three-dimensional location templates
(2) Robot programming
(3) Compile sampling programs

The acoustic source is an intermittent sound buzzer with a resonant frequency of 3100Hz. A UR-10 robot is programmed to carry the buzzer to pre-defined locations then the buzzer is activated to generate acoustic waves, which are detected by a microphone at a fixed location. Finally, the amplification module sends amplified analogue signals to the sampling module for sampling.

The sampling module comprises a DAQ-2010 data acquisition card and a host computer. The DAQ-2010 data acquisition card is pre-configured with API functions in MATLAB. Channel 1 on DAQ-2010 is activated since only one microphone is deployed. The sampling rate is set to 50,000 Hz. After the sampling system is initialised, a pre-allocated array in MATLAB is charged with the sampled signal components alternately from two internal buffers of DAQ-2010. Next, the sampled signal sequences are separated and saved in designated folders for the training of LTPM.

*6.2.1 The design of three-dimensional location templates and coordinate setups*

According to the test design, a robot moves the acoustic source to prescriptive locations. Thus, location templates are designed first to provide three-dimensional coordinates for the robot. Technically, the location template is the tool for acoustic signal collection and it correlates with the accuracy of the localisation system.

Three-dimensional location templates are designed according to the space volume of the

test area. The origin point of the robot is the zero point of all location templates. The operational space is a sphere centred at the origin point and the sphere has a radius equal to the robot's arm span, as shown in Figure 6.2-1 (b).
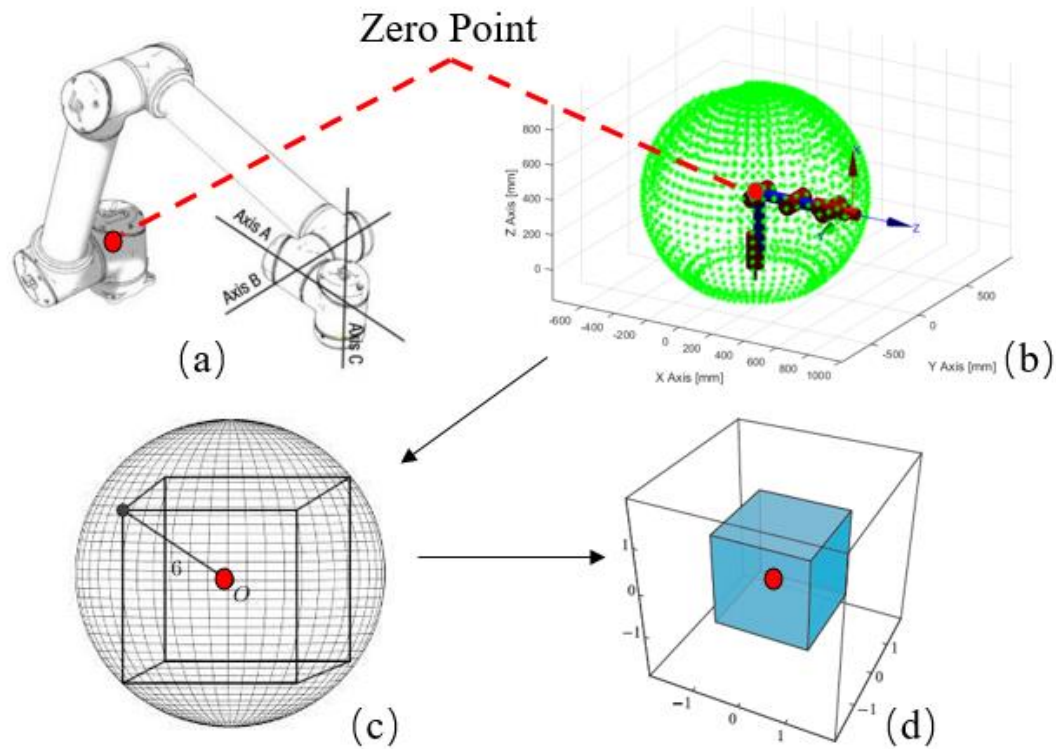


Figure 6.2-1 (a) is the UR robot, and the red point is the origin point. (b) is the sphere operational space centred at the origin point. (c) shows a cube inside the sphere, and (d) illustrates two cubic location templates with different side lengths

The surface of the sphere is the boundary for three-dimensional location templates. In other words, the robot is able to send the buzzer to any location that is inside the sphere. To simplify the coordinate calculations in practical robot programming, the sphere is replaced by a cube, as shown in Figure 6.2-1 (c). The length of the diagonal of the cube equals the diameter of the sphere. By changing the side length of the cube, location templates with different scales are set; in this way, three-dimensional locations are labelled.

A buzzer is attached to the endpoint of the robot manipulator and the robot transports the buzzer to a labelled spatial location by inputting coordinate parameters. Limited by the length of the robot arm, the maximum and minimum side lengths of the template are set to 1200 mm and 460 mm, respectively.

The shape of the three-dimensional location template is cubic because the square cross-section facilitates calculations of three-dimensional coordinates potentially. Two cubic location templates with 1200 mm and 460 mm side lengths are established first. Next, cubic location templates are progressively established along the diagonal axis from the maximum side length to the minimum side length with a step interval of 200 mm. Practical three-dimensional location templates are illustrated in Figure 6.2-2.

Multiple location templates are established by changing the side length of the initial cubic location template. Figure 6.2-2 (a) shows three location templates with 1200 mm, 1000 mm and 800 mm side lengths. Each cubic location template has eight vertices; these vertex locations are the pre-defined locations for the acoustic source.

All three-dimensional vertex locations on the location templates are compiled into the UR-10 robot operational program so that the robot can transport buzzers to any designated vertex location for acoustic signal collection.

A microphone is deployed next to the cubic location template to sample acoustic signals. Next, acoustic signals are collected from the pre-defined locations on cubic location templates. Then, filtering, signal separation, feature extraction, and coordinate merging functions are executed to generate the training and test data for machine learning algorithms.

In practical localisation applications, the localisation system locates acoustic sources to pre-defined locations. The positioning procedure is passive as acoustic sources are located by receiving acoustic waves emitted by these acoustic sources. Compared to electromagnetic localisation systems, signal transmission devices are unnecessary for acoustic wave-based localisation systems thus the system complexity of the acoustic localisation system is low. Indoor localisation is a typical application of the proposed acoustic localisation technology. The localisation system can locate users with speech, knocking, and collision signals emitted by users and provide location information to the interactive system to perform passive human-computer interaction.
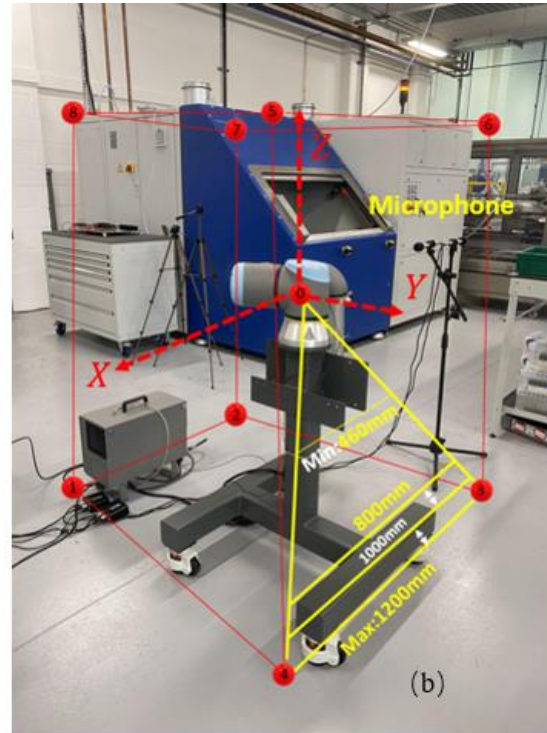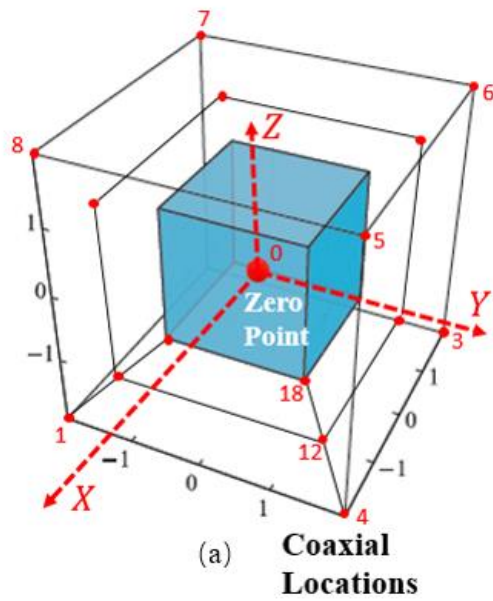
Figure 6.2-2 Illustration of three-dimensional location templates (a) and practical setups (b). The robot arm span is between 1200 mm and 460 mm; thus, the max and min lengths of the cubic location templates are set to 1200 mm and 460 mm correspondingly. Each cubic location template has 8 vertex locations, and locations 4, 12, and 18 are linear coaxial locations

### 6.2.2 Robot for acoustic source movement

Unlike the manual signal collection strategy used in the two-dimensional localisation test, the three-dimensional localisation test requires hundreds of thousands of acoustic sample signals from different locations for model training. It is inefficient to collect such a volume of acoustic signals manually. Therefore, an industrial robot is integrated into the localisation system to collect acoustic signals automatically.

The UR-10 robot, as shown in Figure 6.2-3, is used for the repetitive sample collection work. The robot moves acoustic sources to pre-defined locations and activates the buzzer for acoustic signal sampling. In this way, acoustic signals are generated and collected continuously and accurately. The sample generation rate is 8 samples per minute.

Since the acoustic multipath effect is utilised to locate acoustic sources, the robot and appliances deployed in the space enhance the multipath effect resulting in an improved positioning performance. In other words, the acoustic waves reflected and blocked by the

robot and related appliances could add more multipath components to the received acoustic signal, making acoustic signals collected from different locations more distinguishable.



Figure 6.2-3 The UR-10 robot used in the three-dimensional localisation test. The UR robots are widely applied to industrial manufacturing. The robot system consists of a robot arm, a power module and a teach pendant. Operational programs are compiled in the teach pendant with the URScript Programming Language

### 6.2.3 Sampling module

Similar to the sampling module built for the two-dimensional localisation test, a DAQ-2010 data acquisition card-based sampling module is applied to the three-dimensional localisation system. Corresponding adjustments are made according to the test environment, including the sampling time extension, trigger adjustment, filter implementation, microphone connection etc. At the beginning of the three-dimensional localisation research, a differential microphone array was applied to the localisation system to provide differential acoustic signals. However, subsequent research showed that the microphone array technology requires additional hardware and specific differential algorithms. Besides, according to the test results, signal features extracted

from acoustic signals sampled with a single microphone are sufficient to achieve the expected three-dimensional localisation results. Therefore, the differential microphone array is abandoned in this research.

Microphone:

G.R.A.S Ltd. provides excellent acoustic equipment and techniques for acoustic measurement and analysis. The state-of-the-art sensors from G.R.A.S ensure accurate acoustic signal measurement. The measurement microphone used in the three-dimensional test is the 146AE measurement microphone. Parameters of the G.R.A.S 146AE microphone, such as dynamic range, sensitivity, equivalent resistance, and frequency responding range, are shown in Table 6.2-1.

The frequency response and dynamic range of measurement microphones are better than that of studio microphones. Besides, measurement microphones are shorter and lighter; thus, they can be easily deployed in the three-dimensional space with stands, tapes and glues. 146AE is selected because of its stable performance under different environments; the output of 146AE maintains at a stable level under different temperatures and pressure.

Another merit of 146AE is that this microphone has no resonant frequency compared to the Murata PKS1 shock sensor used in the two-dimensional localisation test. Therefore, hardware filtering is unnecessary for the 146AE measurement microphone.

The 146AE microphone is powered by an AA12 power module, as shown in Figure 6.2-4. The AA12 provides three amplification options. Namely, -20 dB, 20 dB and 40 dB. This amplification characteristic of AA12 makes subsequent sampling work more convenient and efficient. Different amplification options are selected according to the changes in the acoustic source.

Table 6.2-1 Comparison between CO2 microphones (ordinary microphones) and G.R.A.S microphones (measurement microphones)

| Usage of CO2/46AE/146AE | CO2 (Studio Microphone) Studio | 46AE (Measurement Microphone) Measurement | 146AE (Measurement Microphone) Measurement |
|---|---|---|---|
| Type | Condenser | Condenser | Condenser |
| Diaphragm Size Polar Pattern | Small Diaphragm Cardioid | Free Field | Free Field |
| Width | 20 mm | 13.2 mm | 12 mm |
| Height | 150 mm | 84 mm | 86.5 mm |
| Weight | 170 g | 33 g | 35 g |
| Sensitivity | -40 dBV/Pascal | -26 dBV/Pascal | -26 dBV/Pascal |
| Impedance | 200 Ohm | < 50 Ohm | < 50 Ohm |
| Frequency Response | 40 Hz - 20000 Hz | 3.15 Hz – 20000 Hz | 3.15 Hz - 20000 Hz |
| Connector Type | Microphone (XLR) | BNC | BNC |
| Dynamic Range | 112 dB | 138 dB | 133 dB |

Figure 6.2-4 The AA12 power module and the input adapter. The power module is a 2-channel power module with gain control and CCP preamplifier

Amplification:

The AA12 module has an output voltage of 2 mv with no biases. In contrast, the CO2-based sampling module has an output voltage of 25 mv. The characteristic of AA12 is beneficial for improving the signal-to-noise ratio and maintaining stable signal output. The amplified signals are sent to the DAQ-2010 data acquisition card for sampling.

Data interface and sampling configuration:

According to pre-defined location templates, the UR robot continuously transports buzzers to designated locations. Analogue acoustic signals are then collected by the microphone and sent to the DAQ-2010 for analogue-to-digital conversion. The DAQ-2010 sampling module is optimised in the three-dimensional localisation test to perform four hours of continuous sampling. The monitor interface and control panel used in the two-dimensional sampling module are removed to reduce the computational burdens and improve the stability of the sampling system.

The optimised sampling module work continuously for four hours at a sampling rate of 50,000 Hz. The 4-hour acoustic data has a size of 1.5 GB (MATLAB data format, MAT).

A sampled acoustic signal is shown in Figure 6.2-5.
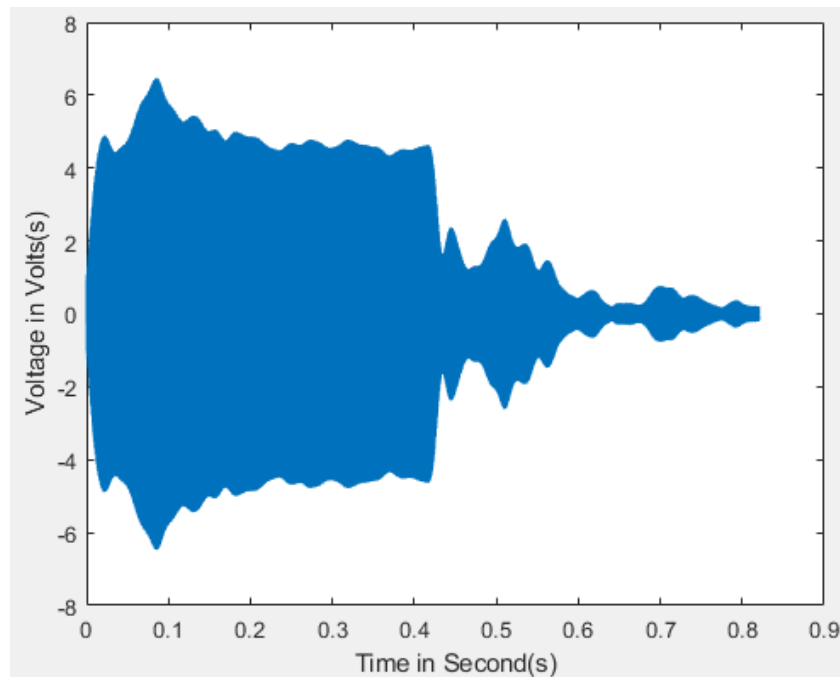


Figure 6.2-5 An acoustic signal collected from the laboratory. The X-axis is the
time, while the Y-axis stands for the amplitude of the signal

*6.2.4 Pre-processing for feature extraction: signal separation and coordinate merge*

The three-dimensional acoustic signal matrices are converted into a one-dimensional
column array for faster processing speed. The one-dimensional array is the raw data
which contains acoustic signals sampled from different locations. Thus, independent
acoustic signals must be separated for subsequent signal processing.

Signal filtering:

The first step is to filter the array with a bandpass filter to eliminate the unwanted
frequency components. As shown in Figure 6.2-6, signal components between 2,000 Hz
- 15,000 Hz are preserved.

Signal separation:

A waveform-based separation algorithm is developed to separate acoustic signals. The
block diagram of the separation algorithm is shown in Figure 6.2-7. The algorithm has a
similar functionality to the hardware trigger, but it is more efficient as it automatically

118

filters out unqualified signals and separates acoustic signals.
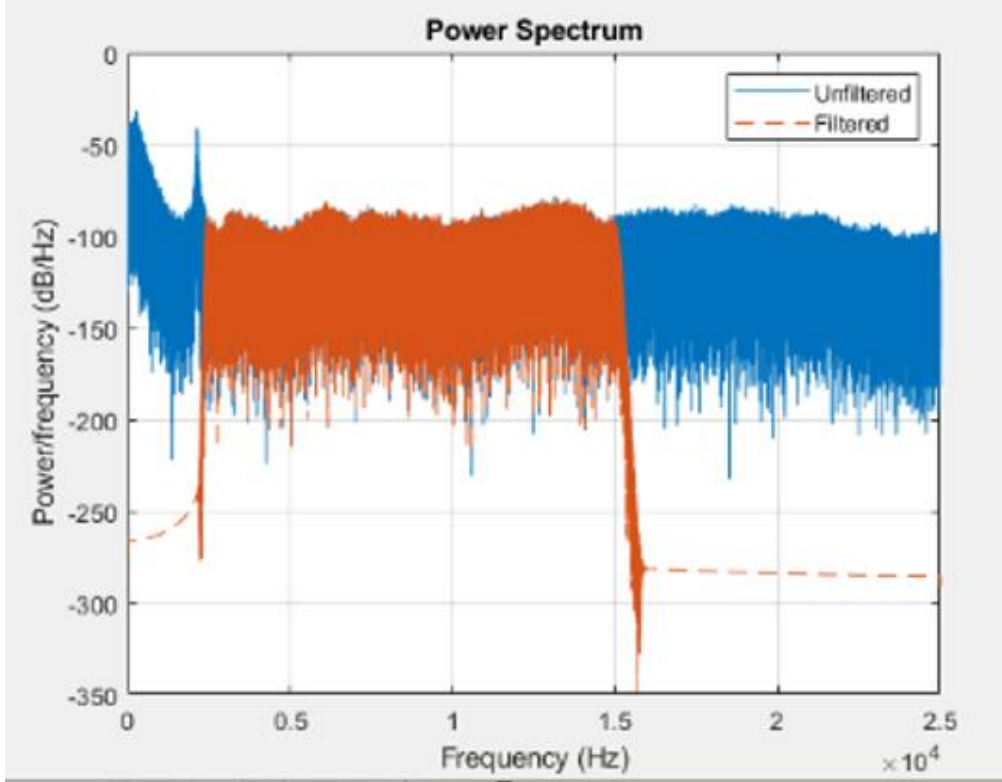


Figure 6.2-6 Signal power spectrums after passing a bandpass filtering (orange). The blue power spectrum stands for the raw acoustic signal. The bandpass filter removes low-frequency interferences

In the initial separation of acoustic signals, a maximum peak value of an acoustic signal $P_{max}$ is extracted first. Next, the signal sequence is initially separated into two signal sequences according to $P_{max}$. The two separated signal sequences are defined as the forward separation sequence and the backward separation sequence.

The location of the microphone is fixed; thus, one of the eight locations on a location template always has the shortest distance to the microphone. Since the energy of an acoustic source is finite, acoustic signals at this location always have the highest peak value $V_p$, while peak values $V_1, V_2, ... V_7$ of acoustic signals at the other 7 locations are smaller than $V_p$.

The $V_p$ appears periodically in the sampled signal sequence. Therefore, periodic secondary separations on the forward separation sequence and backward separation sequence are performed by extracting multiple $V_p$ values.

119

Taking the forward separation sequence as an example, the forward separation sequence is divided into multiple periodical signal sequences according to $V_p$ values. Each separated signal sequence contains eight signals, corresponding to eight vertex locations of a cubic location template. The same separation is applied to the backward separation sequence.
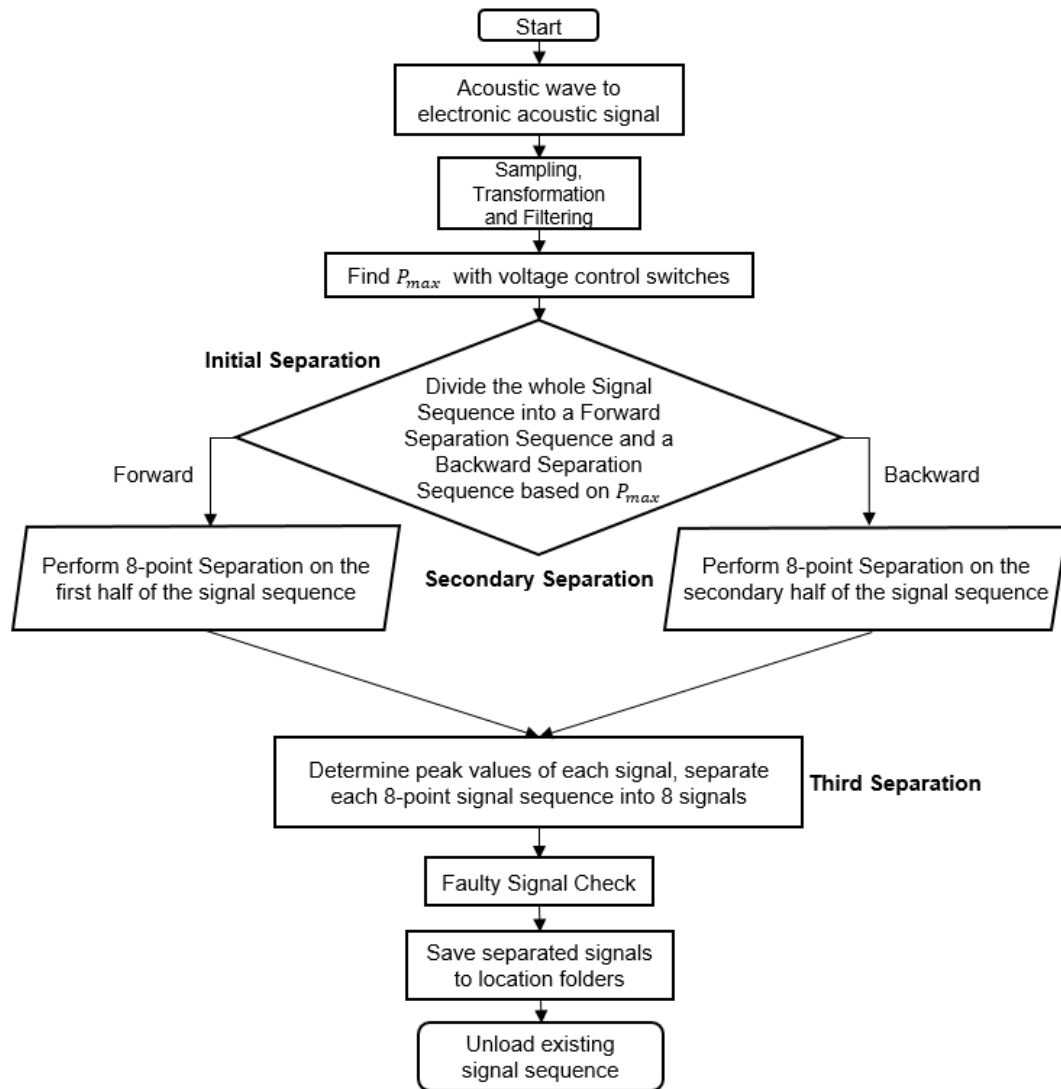


Figure 6.2-7 Block diagram of the signal separation algorithm in the three-dimensional test

The third separation separates independent acoustic signals from each periodical signal sequence acquired from the secondary separation. Since the robot needs a certain time to take the buzzer to a specified location, the third separation is performed based on the operational time difference of the robot system.

120

To this extent, acoustic signals at different three-dimensional locations are separated. Next, separated signals are examined again to eliminate faulty signals. Separated signals are stored in location folders for feature extraction and signal transformation.
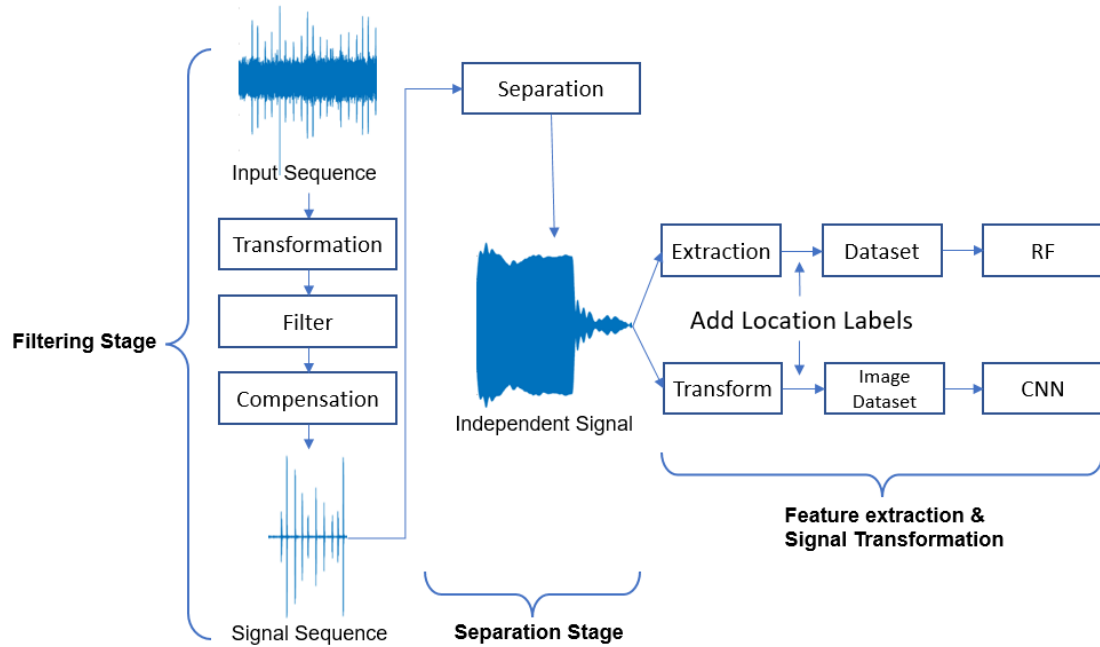


Figure 6.2-8 Signal processing flow chart for three-dimensional localisation. Separated signals are then processed with feature extraction functions and transformation functions to generate two types of training datasets for Random Forest and Convolutional Neural Networks, respectively

The complete signal processing procedure is shown in Figure 6.2-8. Signal processing consists of three parts: signal filtering, signal separation, and feature extraction. But before feature extraction, all separated acoustic signals are normalised with a MATLAB library function: mapminmax. Different features have different dimensions and the value of different features varies significantly. Therefore, the collected signals are normalised to eliminate the potential impact caused by differences in the dimensions of features and value ranges between different features.

In data normalisation, a signal sequence is centralised first according to the minimum value in the signal sequence. Then the sequence is scaled according to the range decided by the maximum and minimum values in the sequence. Values of a normalised signal are converged to between [0,1].

Values of separated signals are then regulated between [-10,10] to facilitate feature

extraction. The separated signals are ready for feature extraction and transformation after normalisation. Further normalisation processings on signal feature values extracted from these separated signals and signal images converted from these separated signals are unnecessary.

Two types of data are generated according to two machine learning algorithms integrated into LTPM. Namely, acoustic features and images of acoustic signals. The acoustic features are in the format of a bijective array, while signal images are the transformations of acoustic signals in different domains.

*6.2.5 Signal processing: feature extraction and signal image transformation*

Acoustic waves propagate omnidirectionally in three-dimensional space; the microphone deployed in the space only receives part of the acoustic wave emitted by an acoustic source. Thus, more features are required to accurately classify acoustic signals from different locations.

Features for Random Forest:

Features for Random Forest consist of statistical features and information theoretic features. Statistical features include dimensional features (such as the mean value, the variance, the power, the centroid frequency, and the peak or valley value) and dimensionless features (such as the Kurtosis, crest factor and L factor) extracted from the time domain, the frequency domain and the spectrum of an acoustic signal.

The selection of statistical features is related to the physical properties of acoustic waves. For example, the nature of acoustic propagation is the transmission of mechanical energy. Each sampled signal component is regarded as an energy integration in the time domain. The sum of the energy of the direct components and the energy of the multipath components forms the total energy of an acoustic signal; thus, the signal energy is selected. Similarly, the power distribution of different frequency components varies with propagation distance; thus, power features are added to the feature set.

Information theoretic features quantify the information contained in a signal from a mathematical perspective [123]. Entropy is an important concept in information theory;

it describes data uncertainty. Different entropy values represent the amount of information, the probability distributions of signals and the complexity of a time series. For example, permutation entropy indicates the complexity of a time series (acoustic signals are time series). The permutation entropy value of an acoustic signal will be higher if the time series of the acoustic signal is more complex than those of other acoustic signals [124]. Other information theoretic features include fuzzy entropy, information entropy, approximate entropy, etc.

In this test, 43 acoustic features are used in acoustic signal matching. Acoustic features used in the test are listed in Table 6.2-2. In a practical case, 43 acoustic features are extracted from each acoustics signal. These feature values are input data for a supervised machine learning algorithm (Random Forest).

Table 6.2-2 Illustration of statistical features and information theoretic features

| Signal domains | Signal features |
| --- | --- |
| Time domain | Signal Length, Energy, Mean value, Absolute Mean value, Variance, Standard Deviation, Absolute sum, Peak value, Valley value, Valley-Peak, Median value, Kurtosis, Skew, Root Mean Square, Sin factor, Crest factor, Impulse factor, L factor, Zero Crossing Rate, Singular Value Decomposition |
| Frequency domain | Max Amplitude, Min Amplitude, Median Amplitude, Mean Amplitude, Valley-Peak, Centroid Frequency, Mean Square Frequency, Mean Square Root Frequency, Frequency Variance, Frequency Standard Deviation, Centroid Frequency of Kurtosis Diagram |
| Power Spectrum | Max Power, Min Power, Median Power, Mean Power, Centroid Frequency, Signal to Noise Ratio, Occupied Bandwidth, Pitch |
| Entropy | Signal Entropy, Spectrum Entropy, Sample Entropy, Permutation Entropy |

Part of the features are introduced below:

$$\text{Mean Square Frequency} = \frac{\int_0^\infty f^2 S(f) df}{\int_0^\infty S(f) df} \qquad 6-1$$

**Mean Square Frequency**: $f$ is the frequency component of the signal. $S(f)$ is the amplitude of the corresponding frequency $f$ of the acoustic signal. The feature describes the change of the main frequency band on the power spectrum.

$$\text{RMS Frequency} = \sqrt{\frac{\int_0^\infty f^2 S(f) df}{\int_0^\infty S(f) df}} \qquad 6-2$$

**Root Mean Square Frequency**: This feature is the arithmetic square root of Mean

Square Frequency. The root Mean Square frequency of a spectrum is a value which represents the overall level of energy across the frequency range.

$$\text{Variance of Frequency} = \frac{\int_0^N (f - CF)^2 S(f) df}{\int_0^\infty S(f) df} \qquad 6 - 3$$

**The variance of Frequency**: $CF$ is the centroid frequency of the acoustic signal, representing the frequency of the largest signal component in the spectrum. The feature describes the fluctuation degree of signal frequency or the distribution of power spectrum energy. The frequency variance is small if the spectral amplitude near the centroid frequency is high.

$$\text{STD Frequency} = \sqrt{\frac{\int_0^N (f - CF)^2 S(f) df}{\int_0^\infty S(f) df}} \qquad 6 - 4$$

**Standard Deviation of Frequency**: The standard deviation of frequency is the square root of the frequency variance. It is often used to describe the dispersity of the power spectrum energy distribution. The standard deviation of frequency has the same dimension as the original signal, which represents the degree of dispersion of frequency components.

$$\text{Crest} = \frac{X_{peak}}{X_{RMS}} \qquad 6 - 5$$

**Crest Factor**: The crest factor is a time-invariant signal feature. $X_{peak}$ is the peak value of the acoustic signal while $X_{RMS}$ is the signal's root mean square value. It represents the relation between the signal peak and the signal waveform.

$$K = \frac{1}{N} \sum_{i=1}^N \frac{(x(i) - \mu)^4}{\sigma^4} \qquad 6 - 6$$

**Kurtosis**: $\sigma$ is the standard deviation of a signal sequence, $\mu$ is the mean value of the signal, $x(i)$ is the amplitude, and $N$ is the signal length. Kurtosis is the ratio of the 4th-order central moment to the 4th power of the standard deviation. The feature is often used to detect the impact of signals. It reflects the sharpness of peaks in the signal distribution.

$$I = \frac{X_{peak}}{\frac{1}{N}\sum_{i=1}^{N}|x_i|} \qquad\qquad 6-7$$

**Impulse factor**: The impulse factor is the ratio of the signal peak to the absolute mean value of the signal. The difference between the impulse indicator and the crest factor lies in the denominator. The absolute mean value of a signal is less than the RMS value of the signal; thus, the impulse indicator is greater than the crest factor.

$$CL = \frac{X_{peak}}{\left(\frac{1}{N}\sum_{i=1}^{N}|x_i|\right)^2} \qquad\qquad 6-8$$

**Clearance factor (L factor)**: The feature is the ratio of the peak value of a signal to the square mean value of the square roots of the absolute amplitude values of the signal.

$$S = \frac{X_{RMS}}{|\bar{x}|} \qquad\qquad 6-9$$

**Waveform factor (sin factor)**: The waveform factor is the ratio of a signal's RMS value to the signal's absolute mean value. The physical concept of the waveform factor is the current ratio. Namely, it is the current ratio of a signal's Direct components (DC) to the alternative components (AC).

$$Skewness = \frac{1}{N}\sum_{i=1}^{N}\frac{(x(i)-\mu)^3}{\sigma^3} \qquad\qquad 6-10$$

**Skewness (skew)**: Skewness is the ratio of the 3rd-order central moment to the 3rd power of the standard deviation. It represents the asymmetry in the probability distribution of a signal.

$$H(X) = -\sum_{i=1}^{n}p(x_i)\log p(x_i) \qquad\qquad 6-11$$

**Information Entropy (signal entropy)**: X represents random variables. $p(x_i)$ represents the probability function, which is essentially a mathematical expectation of the amount of information. The greater the information entropy, the more chaotic the signal is.

$$H_{pe}(m) = -\sum_{j=1}^{m} p_j \log p_j \qquad\qquad 6-12$$

**Permutation Entropy**: m is the embedded dimension. The original signal sequence is partitioned into vectors to determine the order relations between signal values. The permutation entropy is calculated by calculating the probability of each permutation of vectors. It is an indicator of the complexity of the time series.

Signal images for CNN:

Unlike acoustic features, two-dimensional images show synchronised changes in a time-variant signal. For example, the spectrum of an acoustic signal shows the relation between signal power and frequency, which cannot be sufficiently described with a single type of parameter. From the perspective of signal processing, signal patterns residing in signal images are ideal resources for signal classification but it is not practical to summarise and utilise the hidden signal patterns for signal classification artificially.

However, computer vision provides a solution to this problem. Since computer vision identifies hidden patterns in images and utilises the patterns to achieve object identification, similar algorithms are applied to classify acoustic signals from different locations. Namely, deep learning algorithms are utilised to identify patterns hidden in signal images and classify acoustic signals from different locations according to the summarised signal patterns. In this case, signal transformation techniques are applied to generate multiple signal images for the training of CNN. Part of the images are shown in Figure 6.2-9. And Image types are summarised in Table 6.2-3.

Part of the image transformation formulas are introduced below:

$$\mathrm{amdf}(\tau) = \sum_{t=\tau}^{n-1} |s(t) - s(t-\tau)| \qquad\qquad 6-13$$

**Average Magnitude Difference Function (AMDF)**: $s(t)$ is a frame of an acoustic signal. $\tau$ is the shift sampling number of the frame. In signal analysis, acoustic signals are considered periodic signals. The periodic characteristic of acoustic signals under stationary noise conditions can be analysed and observed with AMDF.
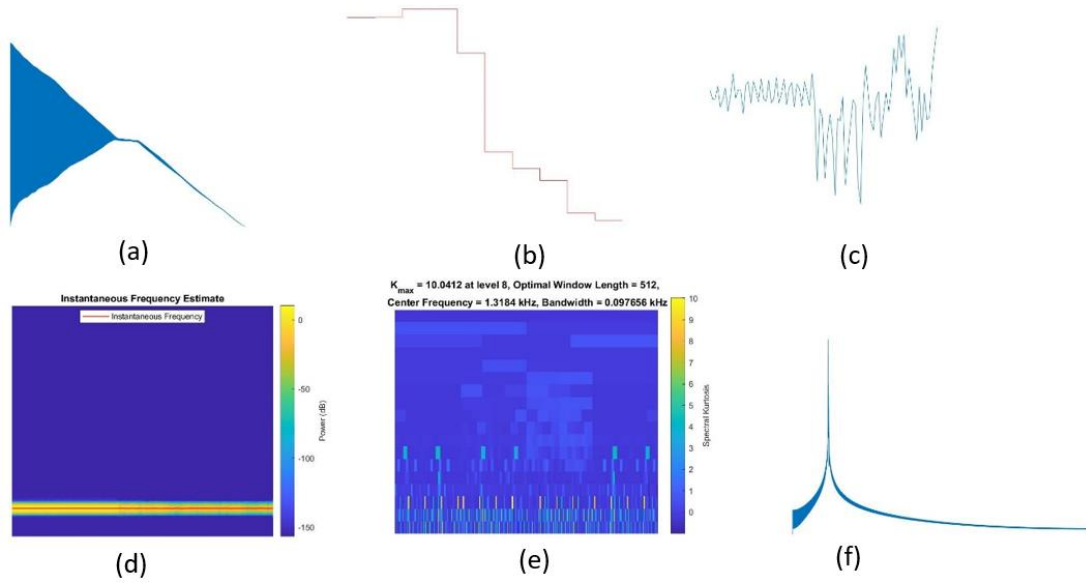
Figure 6.2-9 Display of signal images. (a) Average Magnitude Difference (b) Sound Pressure (c) The change of Centroid Frequency (d) Instant Frequency (e) Kurtogram (f) Power Density

Table 6.2-3 Illustration of signal images

| Signal domains | Signal images |
|---|---|
| Amplitude-Time | Average Magnitude Difference, Waveform |
| Amplitude-Frequency | Peaks, Frequencies of Peak Values, Formant Band, Spectrum, Cepstrum |
| Frequency-Time | Spectrogram, Change of Centroid Frequency, Hilbert-Huang Transform, Instant Frequency |
| Kurtosis-Frequency | Kurtosis gram, Spectral Kurtosis |
| Power Spectral Density-Frequency | Power Spectral Density (Periodogram) |
| Power-Frequency | Power Spectrum (Direct Fourier Transform), Power Spectrum (Fourier Transform of Autocorrelation Function) |
| Sound Pressure Level-Frequency | Change of Sound Pressure |

$$S(t) = \sum_{i=1}^{K} IMF_i(t) + r_K \qquad\qquad 6-14$$

$$H[IMF_i(t)] = \frac{1}{\pi}\int_{-\infty}^{+\infty}\frac{IMF_i(\tau)}{t-\tau}d\tau \qquad\qquad 6-15$$

**Hilbert-Huang Transform (HHT)**: HHT consists of 2 parts. The 1st part is Empirical Mode Decomposition (EMD). EMD adaptively performs local time-frequency analysis on a signal and extracts mode functions from the signal. (Eq. 6-14) is the formula of EMD [125]. $S(t)$ is the acoustic signal, $IMF_i(t)$ are Intrinsic Mode Functions (IMF). $r_K$ is the signal redundancy. The functionality of EMD is to decompose $S(t)$ into a

series of IMFs.

The 2nd part is Hilbert Transform. Hilbert Transform is usually applied to spectral analysis. (Eq. 6-15) is the Hilbert Transform. H[·] stands for Hilbert Transform while $\tau$ is the transition time of the controlled variable t. It can be summarised that $H[IMF_i(t)]$ is the convolution of $IMF_i(t)$ and $\frac{1}{\pi t}$ intrinsically.

HHT reflects the time-frequency relation of a signal. Compared to the signal frequency components acquired with the Fourier transform, HHT delivers detailed variations of frequency components over time because of EMD [128].

$$S_x(f) = \int_{-\infty}^{+\infty} R_x(\tau)e^{-2\pi if\tau} \, d\tau \qquad\qquad 6-16$$

$$R_x(\tau) = \int_{-\infty}^{+\infty} S_x(f)e^{2\pi if\tau} \, d\tau \qquad\qquad 6-17$$

**Power Spectral Density (PSD)**: The power density spectrum represents the distribution of signal power in the frequency domain in terms of density (in W/Hz). According to the Wiener-Khinchin theorem, the autocorrelation function of a signal and its power density spectrum are a pair of Fourier transform pairs [126]. The power density spectrum of a signal can be acquired by performing the Fourier transform on the signal's autocorrelation function. (Eq. 6-16) and (Eq. 6-17) are mathematical expressions of the Wiener-Khinchin theorem, where X(t) is the signal, $R_x(\tau)$ is the autocorrelation function of X(t). $S_x(f)$ is the Fourier transform of $R_x(\tau)$.

$$Power\ Specturm = psd * \frac{f_{sampling}}{N} \qquad\qquad 6-18$$

**Power Spectrum**: The power spectrum is calculated by multiplying the power density function with the sampling rate. $psd$ is the power spectral density function, $f_{sampling}$ is the sampling rate, and $N$ is the number of sampling points. The power spectrum shows the relation between signal power and frequency. In the three-dimensional localisation test, although the frequency of the acoustic source is remained at $f_{source}$, the power of the received signal varies with different distances (locations). Therefore, the

power spectrum is an important input data for CNN-based LTPM.

$$Pressure = \sqrt{\frac{1}{N}\sum_{n=1}^{N} x^2(n)} \qquad\qquad 6-19$$

**Sound Pressure**: The sound pressure is calculated by calculating the RMS of instantaneous sound pressure over time (in Pa). $N$ is the number of sampling points, and $x(n)$ is an acoustic signal. In signal processing, the acoustic signal is divided into frames first, and then the sound pressure of each frame is calculated. Next, a sound pressure variation curve is acquired by connecting these frames.

*6.2.6 The training of Random Forest*

Training method for Random Forest-based LTPM:

Bagging, also known as bootstrap aggregation, is an ensemble training technique in machine learning [127]. It was proposed by Leo Breiman in 1996 [136]. Bagging is combined with classification and regression algorithms to improve classification accuracy while preventing overfitting.

Assuming a training dataset contains $m$ samples. A sample is randomly taken from the training dataset and added to a sampling dataset; then, the sample is returned to the training dataset. In other words, the samples added to the sampling dataset may be selected multiple times during the sampling. After $m$ sampling operations, a sampling dataset containing $m$ samples is obtained. A characteristic of samples in the sampling dataset is that some samples in the training dataset appear multiple times in the sampling dataset, while some samples have never appeared [127].

Each dataset contains $m$ training samples, and $T$ sampling datasets are obtained by repeating the sampling processing. Next, a base learner is trained with each sampling dataset. These trained base learners are then combined to create a classifier. Generally, the final output of the classifier is calculated by voting since each base learner calculates an output. In some tasks with biased requirements, each learner's confidence level can also be further examined to determine the final output.

Overall, Bagging is a technique which reduces generalisation error by combining several classification models. The idea of bagging is to train several different models separately; then, all models vote to generate a final output. Bagging is a classic learning technique in machine learning. Machine learning algorithms that utilise this technique are categorised as ensemble learning.

Since the training dataset is randomly selected with replacements; thus, learners are not correlated. This characteristic is consistent with the training pattern of decision trees. Therefore, bagging is selected to train the Random Forest-based LTPM.

Evaluation of Random Forest-based LTPM:

The data is divided into three categories: training dataset, validation dataset and test dataset. The training dataset is used for model training. The classification model is continuously optimised with the data in the training dataset during the training. The test dataset is not involved in the model training. It is only used to test the classification performance of the trained model. Similarly, the validation dataset is also not involved in the model training, it is used to select the model with the most appropriate hyperparameter.

In the training of RF-based LTPM, the data is only divided into multiple training datasets and test datasets due to bagging. Unbiased estimates of generalisation errors can be calculated in the training of the classification model with the Out-of-Bag (OOB) data [136]; thus, the validation dataset is unnecessary in Random Forest. In other words, the classification accuracy of the model can be directly evaluated with OOB datasets. At the same time, since no validation data is used for hyperparameter adjustment, all training data is preserved. This is a unique characteristic of Random Forest and the reason for choosing Random Forest since the data volume is small (120,000 signals).

Illustration of three-dimensional acoustic localisation:

43 features are extracted from each acoustic signal. After the completion of the feature extraction, the features, along with their coordinates, are written into an Excel sheet file

waiting for the RF machine learning algorithm to read. The algorithm divides the data into a training dataset and a test dataset. Below, the matching between an input acoustic signal and acoustic sample signals is illustrated.

The format of a processed input signal in a training dataset is a 43-dimensional array:

$$S_i := [ s_1, s_2, s_3 \dots s_{43}, \text{coordinate}]$$

So the format of the training data is as follows:

$$S_{Ni} := \begin{bmatrix} s_{11}, s_{12}, s_{13} \dots s_{143}, \text{coordinate1} \\ s_{21}, s_{22}, s_{23} \dots s_{243}, \text{coordinate2} \\ \dots \\ s_{N1}, s_{N2}, s_{N3} \dots s_{N43}, \text{coordinateN} \end{bmatrix}$$

Part of the database is shown in Table 6.2-4. 4 features are selected as an illustration: the minimum valley value, the maximum valley to peak value, the centroid frequency, and the occupied bandwidth of a signal.

Table 6.2-4 Illustration of the training dataset. The example training dataset is a matrix, but the complete training dataset is a 43-dimensional matrix which contains both signal feature values and coordinates

**Example Fingerprint Database/Training Dataset for RF-based LTPM**

| Coordinates of Spatial Location | Valley value in V | Valley to Peak value in V | Centroid Frequency in Hz | Occupied Bandwidth in dB/Hz |
|---|---|---|---|---|
| 1 (600,-600,-600) | -5.04 | 10.07 | 21484.37 | 40.87 |
| 4 (600,600,-600) | -5.84 | 11.69 | 19921.87 | 26.41 |
| 8 (600,-600,600) | -8.43 | 16.88 | 20247.4 | 22.43 |

When a test signal is sent to the trained model, the signal is decomposed into a feature array $T_0 := [t_1, t_2, t_3 \dots t_{43}]$ with the same signal processing and feature extraction functions which have been applied to generate the training data. Next, the trained model matches values in $T_0$ with values in row vectors of $S_{Ni}$ and finds a row vector $S_T$ with the highest similarity to $T_0$. The corresponding coordinate of $S_T$ is confirmed as the location of the acoustic source. For example, if

$$T_0 = [-5.00, 10.00, 21525.00, 41.50]$$

Location 1 (the 1st row) in Table 6.2-4 will be identified as the matched output $S_{T0}$, thus the acoustic source is located at location 1 (600, -600, -600).

*6.2.7 The training of CNN*

Signals are converted into tensors (images) for the training of CNN. Images must be pre-processed first to increase the convergence speed and improve learning efficiency. Therefore, image processing techniques such as scaling, rotating, flipping, and normalisation are applied. The data pre-processing includes the following steps:

Data conversion: Signals are converted into three-dimensional tensors (RGB colour images) before training. Image types are introduced in Section 6.2.5 (b).

Data augmentation: Data augmentation refers to additional processing of the training data. The purpose of data augmentation is to increase the number and diversity of training data, thereby improving the generalisation ability of the model. Common data augmentation methods include image scaling, rotation, flip, translation, cropping, brightness adjustment and contrast adjustment.

Data loading: Reading data from a training dataset and storing the data in memory. Generally, datasets are divided into training and test datasets, which are used for model training and performance evaluation.

Data partitioning: Data are divided into three datasets; the training, validation and test datasets.

Pre-processing on images improves CNN models' training efficiency, generalisation, and classification accuracy.

Stochastic Gradient Descent (SGD) for CNN:

Gradient Descent is applied to find out model parameters in machine learning. From a mathematical perspective, a model is a multivariate function. The gradient of the multivariate function is calculated by calculating the partial derivative $\partial$ of each variable [128].

For example, for a function $f(x, y)$, the partial derivative of $x$ and $y$ are calculated respectively. Thus, the gradient $grad\ f(x, y)$ of $f(x, y)$ is $(\partial f / \partial x, \partial f / \partial y)^T$, or

$\nabla f(x, y)$. Similarly, for a function which contains three variables, the gradient of the function is $(\partial f / \partial x, \partial f / \partial y, \partial f / \partial z)^T$.

The gradient represents the curvature of a function in terms of geometrics. For example, for a function $f(x, y)$ at point $(x_0, y_0)$, the gradient vector is $(\partial f / \partial x_0, \partial f / \partial y_0)^T$. The function curve along the gradient vector is the direction where $f(x, y)$ increases the fastest. Alternatively, along the direction of the gradient vector, the maximum value of the function $f(x, y)$ will be found [128]. On the other hand, along the opposite direction of the gradient vector, which is $-(\partial f / \partial x_0, \partial f / \partial y_0)^T$, the gradient decreases, and the minimum value of the function $f(x, y)$ will be found.

Stochastic gradient descent (SGD) updates the model based on each stochastic sample. Technically, the classification model is updated in every execution period [129]. Below, a function example is used to illustrate the implementation of SGD.

Assume that the objective function is:

$$J(x) = \frac{1}{n} \sum_{i=1}^{n} J(x_i) \qquad\qquad 6-20$$

where $x_i$ is the training sample, $J(x)$ is the objective function. The gradient of the objective function is $\nabla J(x)$:

$$\nabla J(x) = \frac{1}{n} \nabla \sum_{i=1}^{n} J(x_i) \qquad\qquad 6-21$$

If the model is trained with the Batch Gradient Descent method (BGD), gradient calculations on $n$ samples must be performed during each iteration, resulting in a very high calculation cost $O(n)$.

The idea of Stochastic Gradient Descent is to update model parameters according to a random sample $J(x_i)$. Thus the calculation cost decreases from $O(n)$ to $O(1)$. As a result, SGD converges faster than BGD while maintaining the same level of classification accuracy.

SGD has specific problems as well. SGD does not optimise the loss functions [130]. Instead, the loss function of a stochastic training sample is optimised in each iteration.

As a result, not all loss functions are biased toward a global optimal solution, but most loss functions are biased toward the global optimal solution.

As mentioned above, the neural network trained with SGD hardly achieves global optimisation; thus, the iterative trajectory of an independent variable trained with SGD is more tortuous than that trained with BGD, as shown in Figure 6.2-10.
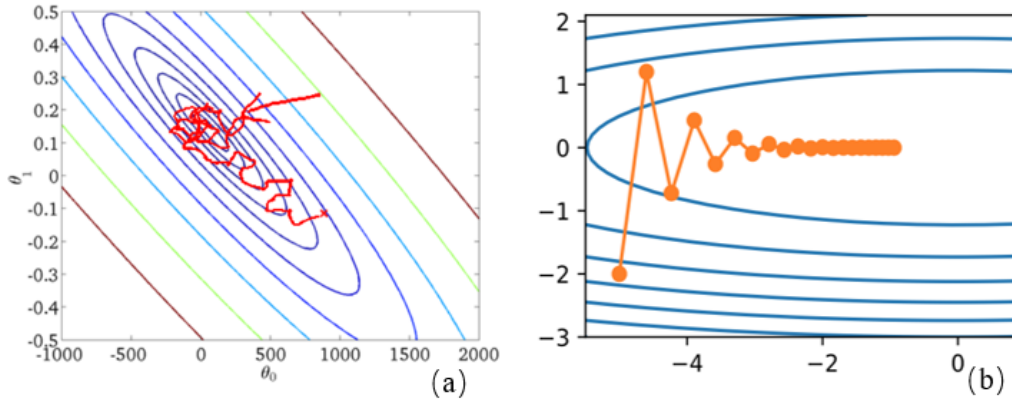


Figure 6.2-10 Illustration of Stochastic Gradient Descent (SGD) training (a). Compared with batch gradient descent (BGD, (b)), the convergence curve of SGD is disordered. However, SGD has a lower computational cost

Evaluation of CNN-based LTPM:

The dataset is divided into three categories: training dataset, validation dataset, and test dataset. The training dataset is used for model training. The classification model is continuously optimised with the data in the training dataset during the training.

The validation dataset is not involved in the model training. The validation dataset is used to verify the convergence of the classification model during the training. It is typically used to adjust hyperparameters. The validation dataset can also be used to examine if the model is overfitting. For example, in the situation that the classification accuracy of a trained model keeps increasing while the classification accuracy of the model using a validation dataset keeps decreasing, the model is considered overfitting.

The test dataset is also not involved in the model training. It is only used to test the classification performance of the trained model.

In the training of CNN-based LTPM, the data is divided into training, validation, and test datasets with a ratio of 3:1:1 [135]. The classification accuracies of a CNN-based LTPM

using the validation data and the training data are shown in Figure 6.2-11.
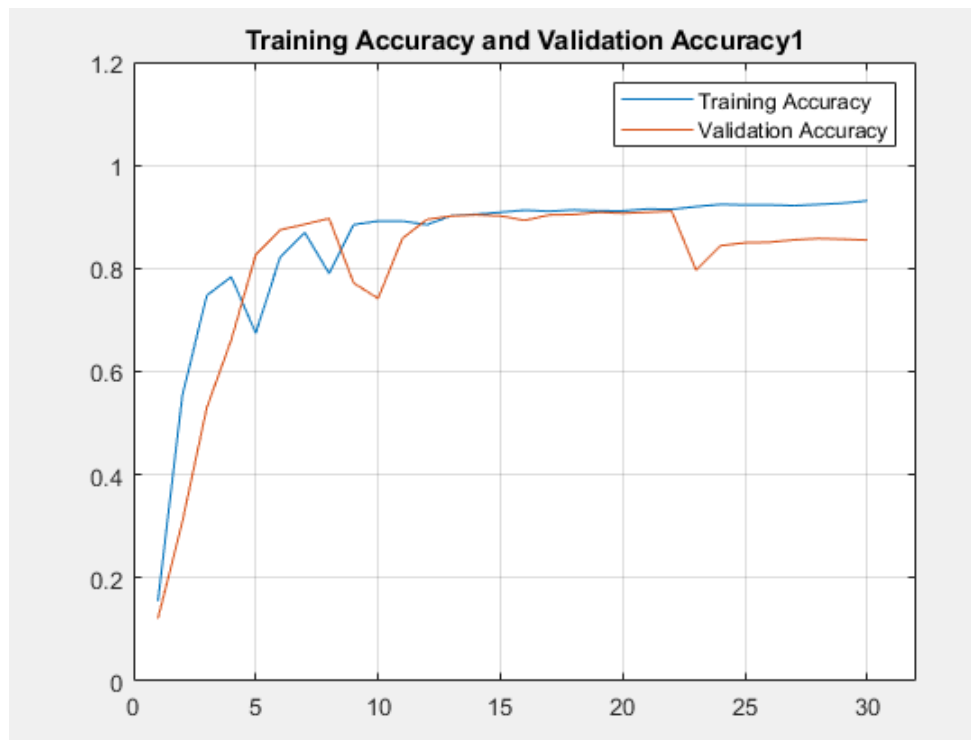


Figure 6.2-11 The accuracy change of a CNN-based LTPM during training. The X-axis represents the epoch. The Y-axis is the percentage accuracy

In the iterations, the accuracy of the LTPM using the validation dataset stabilises at 20 epochs, indicating that the validation accuracy has converged. The training should be stopped to prevent overfitting. Therefore, the epoch number is determined to be 20.

In the three-dimensional localisation test, the training dataset contains 17 types of images. The classification accuracy of CNN in the localisation test stabilises within 20 epochs, as shown in Figure 6.2-12.
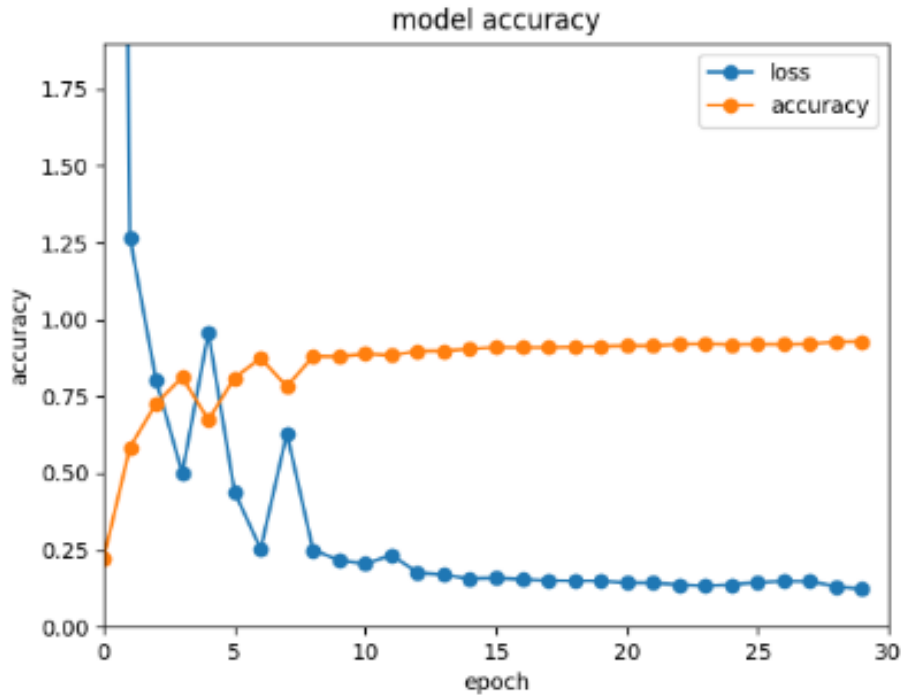
Figure 6.2-12 Iterations of a CNN-based LTPM (using the training dataset). The classification accuracy of the LTPM tends to be stabilised after 20 epochs

## 6.3    Three-dimensional localisation test design

In the three-dimensional localisation test, samples in the database are divided into two types of data: training data and test data. The training data are used to train classification models, and the test data are used to evaluate the classification performance of the trained model. The test design starts with the introduction of tests and input data.

*6.3.1 Three-dimensional localisation test types and input Data*

The three-dimensional localisation tests are divided into the first level tests (training/testing data from the same location template) and the second level tests (training/testing data from different location templates).

Similarly, each level of tests consists of independent tests (each vertex location of location templates serves as an independent system output) and coaxial tests (coaxial vertex locations of multiple location templates serve as a coaxial system output) based on the axial correlation.

The input data consists of primary datasets and secondary datasets. The primary datasets are used for training and testing, while the secondary datasets are mainly used for testing.

136

Details of input datasets have been introduced in Section 6.2.5. 43 acoustic features and coordinates are written into an Excel file for the compiled Random Forest algorithm to read. Similarly, 17 signal images are transformed and stored in designated folders for the compiled CNN algorithm to read.

Two levels of test:

Generally, a dataset refers to signals collected from the same location template. The dataset is split into a training dataset to train a model and a test dataset to test the trained model's classification performance.

In some tests, datasets from other location templates are also used to test the performance of a trained model. Such tests aim to examine the signal discernment of the trained model, especially in situations that the acoustic source shifts from pre-defined three-dimensional locations.

In the 1st level test, signals collected from a location template are divided into two subsets: a sub-dataset for model training and a sub-dataset to test the trained model. For example, 48,000 samples are collected from the location template with a 1200 mm side length. These samples are divided into a training dataset (40,000 samples) and a test dataset (8000 samples). An LTPM is trained with the training dataset and then tested with the test dataset. The 1st level test aims to determine the classification accuracy of the LTPM.

In the 2nd level test, the LTPMs are trained with samples collected from a specific location template and then tested with samples collected from other location templates. For example, samples collected from the 1200 mm location template are used to train a model then the trained model is tested with samples collected from the 1000 mm location template. The 2nd level test aims to verify the uniqueness of the acoustic features embedded in the sampled signals.

Independent test and Coaxial test:

Acoustic signals collected at the coaxial direction share similar acoustic features. The

independent and coaxial tests are therefore designed to examine the coaxial localisation performance of LTPM.

The independent test is a standard point-to-point comparison test. The input and output of the LTPM are identical. For example, in a 24 points independent test, the input dataset contains 24 locations, and the same 24 locations are defined in the output of the LTPM.

In the coaxial test, the vertex locations of different cubic location templates in the diagonal direction are defined as coaxial locations. As shown in Figure 6.2-2 (a), Locations 4,12,18 are coaxial locations. Since these locations are linear, these locations are also referred to linear locations.

The coaxial test aims to verify:

(1) If an acoustic source can be located at the closest pre-defined location and the coaxial location when LTPM cannot find a matched sample signal.

(2) Whether the multipath effect changes linearly on the propagation path.

(3) The relationship between the classification accuracy and the deviation distance.

Since the coaxial locations are regarded as a coaxial location, the output of LTPM only contains 8 locations (corresponding to 8 cube diagonals) regardless of the number of location templates. For example, in an 8 points coaxial test, 24 locations are defined in the input as the datasets are collected from 3 cubic location templates. However, only 8 locations are defined in the output; every 3 locations at the coaxial direction are regarded as 1 linear location.

Thus, the output of the LTPM has multiple patterns. For example, 8 points coaxial test (2-4 cubic location templates), 8 points independent test (single cubic location template), 16 points independent test (2 cubic location templates), and 24 points independent test (3 cubic location templates).

The primary datasets:

3 primary datasets are collected from the 1200 mm, 1000 mm and 800 mm location templates, as shown in Table 6.3-1.

Table 6.3-1 The number of samples of the 3 primary datasets

|  | 1200 mm | 1000 mm | 800 mm |
|---|---|---|---|
| Number of samples | 48,000 | 40,000 | 32,000 |

<u>The secondary dataset:</u>

5 secondary datasets are collected from the 460 mm, 1100 mm, 1050 mm, 900 mm, and 750 mm location templates, as shown in Table 6.3-2.

Table 6.3-2 The number of samples of all datasets

|  | 460 mm | 750 mm | 900 mm | 1050 mm | 1100 mm |
|---|---|---|---|---|---|
| Number of samples | 3,200 | 100 | 100 | 100 | 100 |

<u>Datasets for Random Forest:</u>

3 primary datasets (from 1200 mm, 1000 mm and 800 mm location templates), 4 secondary datasets (from 1100 mm, 1050 mm, 900 mm, and 750 mm location templates) and 1 dataset with a medium number of samples (from 460 mm location template) are established for the training of RF-based LTPM.

<u>Datasets for CNN:</u>

The training dataset contains 2.1 million images (generated with 17 transformation functions). 1/20 of the images are used to train CNN-based LTPM. The train-validation-test ratio is set to 3:1:1. The ratio can be adjusted to ensure efficient data utilisation. The 3:1:1 ratio is the most reasonable ratio since the total amount of collected samples is small (120,000 signals) and the validation dataset is essential to prevent overfitting. The 3:1:1 train-validation-test ratio ensures the efficient training of LTPM. On the other hand, this project attempts to apply machine learning algorithms to acoustic localisation instead of optimising machine learning algorithms; thus, the most commonly used 3:1:1 train-validation-test ratio is selected.

The image compression rate of the image is 0%, and the image resolution is set to 875×656 while all pixel values are within 255. The batch size is 40, and the epoch is set

to 20.

Classification accuracy measurement:

For a given test dataset, the number of samples that are correctly classified by the LTPM (True Positive, TP) to the total number of samples is the accuracy rate (Eq. 6-22).

$$Accuracy = \frac{TP}{TP + FP} * 100\% \qquad 6 - 22$$

where TP stands for True Positive while FP stands for False Positive; thus, the classification accuracy is the proportion of all "correctly matched results (TP)" to all "matched results (TP+FP)".

Localisation accuracy:

Unlike the classification accuracy measurement, the localisation accuracy is determined by location templates. It is calculated according to the gap distances between location templates. Location templates have different side lengths, and the gap distance between two vertex locations of two different cubic location templates represents the localisation accuracy. For example, the minimum distance between the 1200 mm cubic location template and the 1000 mm cubic location template is 173 mm and the maximum distance is 2078 mm. In this case, LTPM trained with the 1000 mm dataset is tested with the 1200 mm dataset and the localisation accuracy is 173 mm.

Figure 6.3-1 illustrates the calculation of the localisation accuracy between different location templates. The hypotenuse of the triangle is 173 mm, which is the minimum gap between location templates. It is also the highest localisation accuracy that LTPM achieved in the three-dimensional localisation test.
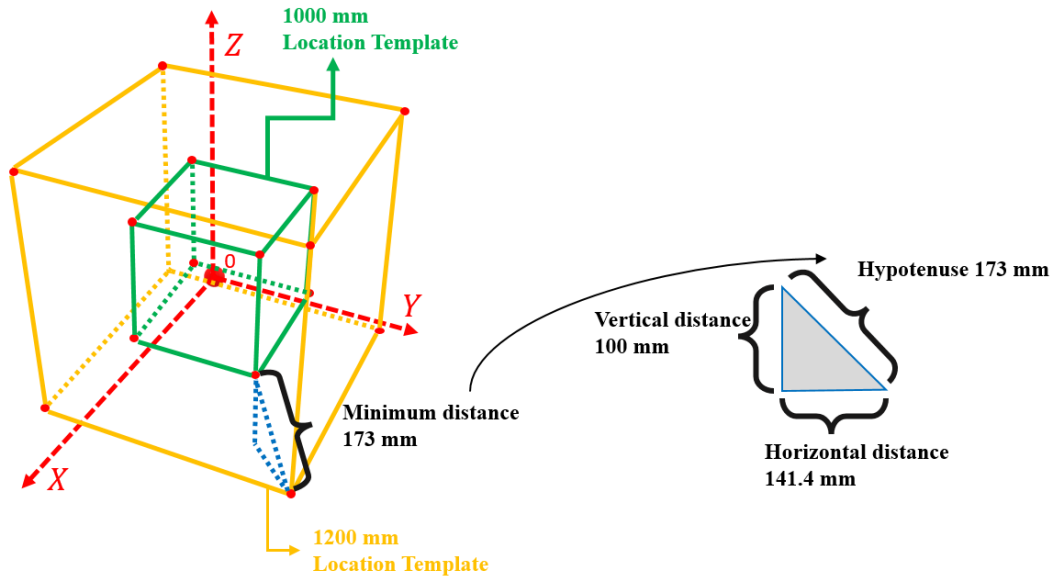
Figure 6.3-1 Illustration of the minimum localisation accuracy

The minimum localisation accuracy is the distance between the coaxial vertices of two cubic location templates, while the maximum localisation accuracy is the diagonal distance (2078 mm) of the 1200 mm cubic location template.

Deviation distance:

Deviation distance is the distance between the acoustic source and the closest pre-defined location. In the 2nd level test, LTPMs are trained with a dataset from a specific location template and then tested with datasets from other location templates. The deviation distance represents the gap between the training location template and the test location template, as shown in Figure 6.3-2. The deviation distance in the three-dimensional localisation test varies between 50 mm to 450 mm.

Assume that the training data is collected with the green location template, and the test data is collected with the yellow location template. The distance between location templates is defined as the deviation distance. The deviation distance is used to describe the shift between the acoustic source and the closest pre-defined location.
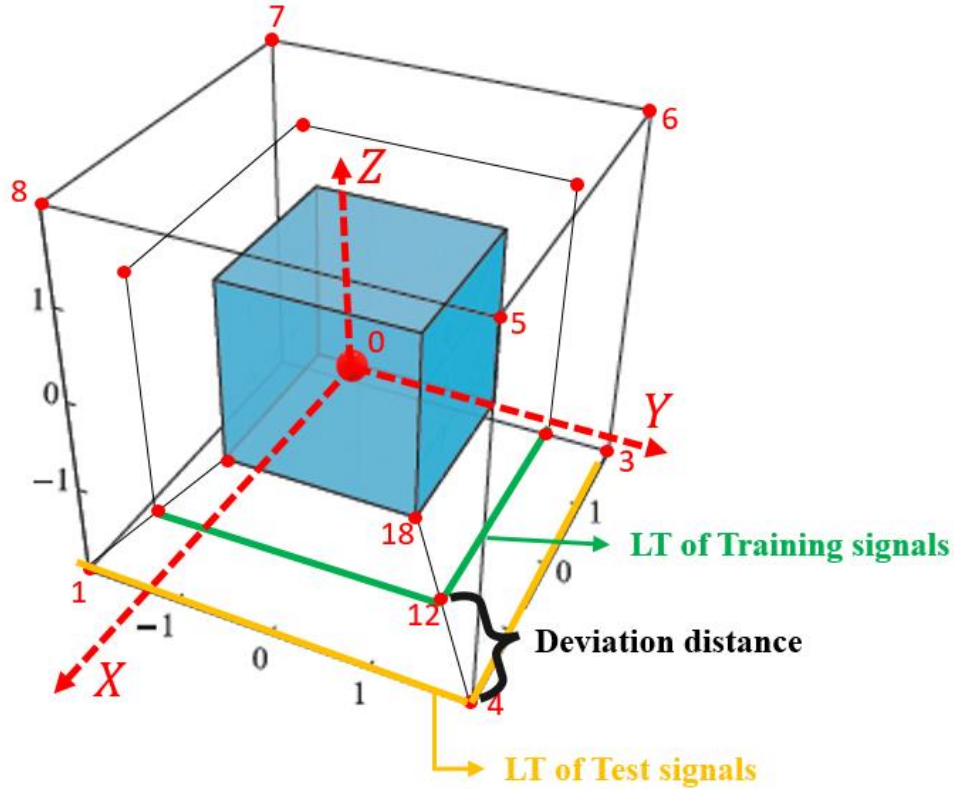
Figure 6.3-2 Illustration of deviation distance

*6.3.2 The first level of cross test*

The first level of cross test between 3 primary datasets:

The test consists of an independent test (24 points) and a coaxial test (8 points). Since 3 primary datasets are used (corresponding to 3 location templates), the output includes 24 locations (3 location templates have 24 cubic vertex locations). While in the 8 points coaxial test, locations on the cubic diagonals are connected to form linear locations. An example is shown in Figure 6.2-2 (a); locations 4, 12 and 18 are coaxial locations.

In this test, 120,000 samples from the 3 primary datasets are divided into a training dataset which contains 100,000 samples and a test dataset which contains 20,000 samples. The 20,000 samples in the test dataset are randomly selected from the 3 primary datasets. This test aims to evaluate the accuracy of LTPM preliminarily. Test results are shown in the first 2 columns of Table 6.3-3.

Next, models trained with different datasets are tested separately. In these tests, input and output are based on a single location template with 8 locations. For example, in the 1200

142

mm dataset (LTPM-LT-8(1200)), 48,000 samples are divided into a training dataset (40,000 samples) and a test dataset (8,000 samples). The same test approach is applied to the 1000 mm dataset and 800 mm dataset. The test results are displayed in the 3rd, 4th and 5th columns of Table 6.3-3.

Lastly, the 460 mm dataset, which contains 3,200 samples, is used for a localisation test (2,700 samples for training and 500 samples for testing). The test is designed to evaluate the performance of the LTPM-based localisation system when the number of training data is reduced. The test results are shown in the last column of Table 6.3-3.

In addition, to provide an intuitive comparison between RF-based LTPM and CNN-based LTPM, test results of RF-based LTPM and test results of CNN-based LTPM are presented in the same table.

Table 6.3-3 Test results of the first level test between 3 primary datasets. LTPM-8(all) denotes the coaxial test and LTPM-24(all) represents the 24 points (3×8) independent test. The classification accuracy is presented in the corresponding columns in percentage

| Method | LTPM-8(all) | LTPM-24(all) | LTPM-8(1200) | LTPM-8(1000) | LTPM-8(800) | LTPM-8(460) |
|---|---|---|---|---|---|---|
| RF(%) | **95.97** | **98.90** | 98.21 | 96.47 | 98.90 | 88.66 |
| CNN(%) | **89.25** | **88.72** | 87.30 | 88.48 | 89.85 | 82.02 |

Note: the number inside the parentheses indicates the side length of the location template used for signal collection. For example, LTPM-8(1200) means the model is trained with signals collected from the 1200 mm location template.

The classification accuracy of LTPM is the primary concern. The RF-based LTPM achieved 98.9% classification accuracy in the 24 points independent test. Multi-dimensional acoustic features and sufficient training data ensure the classification accuracy. In contrast, the classification accuracy of the RF-based LTPM trained with the 460 mm dataset decreases to 88.66% since the data volume of the training data is reduced. The test result indicates that the classification accuracy of LTPM varies with data volume.

The RF-based LTPM achieved a classification accuracy of 95.97% in the 8 points coaxial test, indicating that the RF-based LTPM distinguishes coaxial locations, and the classification accuracy decreases slightly if adjacent locations on the coaxial lines of the cubic location templates are defined as linear locations forcibly.

The classification accuracy of the CNN-based LTPM is inferior to the classification accuracy of the RF-based LTPM but the CNN-based LTPM still has an average classification accuracy of 87%. In conclusion, the CNN-based LTPM achieved excellent localisation results with a limited amount of training data. The test result in the 6th column (82.02%) indicates that the classification accuracy of the CNN-based LTPM also varies with the data volume of the training data.

The test results preliminarily verified that LTPM is a feasible three-dimensional acoustic localisation technology which can be applied to indoor environments, and different machine learning algorithms have different positioning performances in the background of indoor acoustic source localisation.

*6.3.3 The second level of cross test*

This test is more complex than the first level of cross test. It consists of a series of tests, and it is designed to determine the LTPM-based localisation system's specifications.

The test includes spatial adjacent location test, coaxial test and data volume test. A characteristic of the second level of cross test is that the LTPM is trained with data collected from a specific location template, while the test data are collected from other location templates. Below, the second level of cross tests are introduced.

460 mm dataset (test dataset) to 3 primary datasets (training dataset, 8 points and 24 points):

Considering that the effectiveness of acoustic features corresponds to the distance between the acoustic source and the microphone. 3 primary datasets (800 mm, 1000 mm, 1200 mm) and the 460 mm dataset (collected from a location template with the shortest side length) are selected for the test. In this test, LTPMs are trained with 3 primary datasets and then tested with the 460 mm dataset.

The test aims to determine whether the acoustic source can be accurately or axially located to the closest location in the situation that acoustic sources deviate from locations defined by the location template.

The test consists of 5 parts; in the 1st part, the 460 mm dataset is used to test the LTPM trained with all 3 primary datasets. Coaxial locations are defined as linear locations; thus, the output of the LTPM includes 8 locations only. The 2nd part is similar to the first part, but all 24 locations from the 3 primary datasets are defined as output locations; thus, the output of the LTPM includes 24 locations. The 3rd part is about testing the LTPM (trained with the 800 mm dataset) with the 460 mm dataset. The 4th part is about testing the LTPM (trained with the 1000 mm dataset) with the 460 mm dataset. And the 5th part is about testing the LTPM (trained with the 1200 mm dataset) with the 460 mm dataset.

In this test, the locations of the training data are gradually moving away from the locations of the test data. Thus, the relation between acoustic features and distances is revealed. Test results are shown in Table 6.3-4.

Table 6.3-4 Cross test results of 460 mm against 800 mm,1000 mm and 1200 mm

| Method | LTPM-CT-8(460-(800+1000+1200)) | LTPM-CT-24(460-(800+1000+1200)) | LTPM-CT-8(460-(800)) | LTPM-CT-8(460-(1000)) | LTPM-CT-8(460-(1200)) |
|---|---|---|---|---|---|
| RF(%) | 18.15 | 10.56 | 10.5 | 15.65 | 19.56 |
| CNN(%) | 87.20 | - | 81.21 | 79.93 | 86.62 |

The RF-based LTPM has a low classification accuracy in the situation that acoustic sources are not placed at exact pre-defined locations. For acoustic sources that are deviated from pre-defined locations, the average classification accuracy of the RF-based LTPM is 15%.

However, the CNN-based LTPM shows a different property. The CNN-based LTPM coaxially locate acoustic sources that are not at defined locations. The CNN-based LTPM has a classification accuracy of 81.21% in the situation that acoustic sources are 340 mm away from the pre-defined locations. The CNN-based LTPM still has a classification accuracy of 86.62% when the deviation distance further increases to 740 mm. These test results indicate that acoustic signals of acoustic sources at coaxial locations share similar signal patterns. Therefore, the CNN-based LTPM is able to locate acoustic sources that deviate from pre-defined locations to the closest coaxial location. This property is named

inheritance in the three-dimensional localisation test. The inheritance of the CNN-based LTPM is crucial to practical applications. In practical cases that the actual location of an acoustic source deviates from a pre-defined location, the inheritance of the CNN-based LTPM ensures that the acoustic source can be located at the closest pre-defined location.

Correspondingly, the RF-based LTPM shows a property of non-inheritance. The classification accuracy of the RF-based LTPM in the 8 points coaxial localisation test is 18% (1st column). In contrast, the classification accuracy of the RF-based LTPM decreases to 10.56% in the 24 points localisation test (2nd column), indicating that acoustic features of acoustic signals from coaxial locations share similar changes; thus, the classification accuracy is improved. The non-inheritance endows the RF-based LTPM with an automatic filtering ability and ensures high classification accuracy in the situation that acoustic sources are placed at exact pre-defined locations.

750 mm dataset to 3 primary datasets (8 points and 24 points):

The purpose of this test is to measure the localisation accuracy of the system in a situation that an acoustic source is placed next to a pre-defined location. The 3 primary datasets (800 mm, 1000 mm, 1200 mm) and the 750 mm dataset are selected for the test. In this test, LTPMs are trained with the 3 primary datasets and then tested with the 750 mm dataset. Next, LTPMs are independently trained with 800 mm, 1000 mm, 1200 mm, and 460 mm datasets and tested with the 750 mm dataset.

This test determines the relation between localisation accuracy and classification accuracy. For a successful localisation, any signals from the 750 mm location template are located at the closest location on the training location template or a coaxial linear location.

The test consists of 6 parts; in the 1st part, the model trained with 3 primary datasets is tested with the 750 mm dataset. Locations at the coaxial line are defined as linear locations; thus, the output of the LTPM includes 8 locations. The 2nd part is similar to the 1st part, but all locations are independent.

In the 3rd part, the 750 mm dataset tests the LTPM trained with the 800 mm dataset.

Since the 800 mm location template is the closest location template to the 750 mm location template, the classification accuracy is expected to increase. In the 4th part, the 750 mm dataset tests the LTPM trained with the 1000 mm dataset. In the 5th part, the 750 mm dataset tests the LTPM trained with the 1200 mm dataset. In the 6th part, the 750 mm dataset tests the LTPM trained with the 460 mm dataset. The 6th part is for the overfitting examination. Test results are shown in Table 6.3-5.

Table 6.3-5 Cross test results of 750 mm against 800 mm,1000 mm, 1200 mm and 460 mm

| Method | LTPM-CT-8(750-(800+1000+1200)) | LTPM-CT-24(750-(800+1000+1200)) | LTPM-CT-8(750-(800)) | LTPM-CT-8(750-(1000)) | LTPM-CT-8(750-(1200)) | LTPM-CT-8(750-(460)) |
|---|---|---|---|---|---|---|
| RF(%) | 35 | 27.5 | 40 | 5 | 32.5 | 2.5 |
| CNN(%) | 82.25 | - | 80.45 | 80.60 | 85.79 | 83.89 |

RF still shows the property of non-inheritance. Acoustic features extracted from the received acoustic signal can hardly be recognised and matched by the RF-based LTPM when acoustic sources are not at pre-defined spatial locations, resulting in a decreased classification accuracy.

In the 8 points coaxial test, the RF-based LTPM has a classification accuracy of 35%. In the 24 points independent test, the RF-based LTPM has a classification accuracy of 27.5%. These test results are similar to the test results from the previous test (the 1st column and the 2nd column), which further verified that the RF-based LTPM is unable to locate acoustic sources that are not at pre-defined locations. On the other hand, the CNN-based LTPM maintains a coaxial classification accuracy of over 80% throughout the test.

900 mm dataset to 3 primary datasets (8 points and 24 points):

An extension of the previous test, the test dataset is replaced with the 900 mm dataset. In the previous test, the test location template is wrapped by training location templates visually. But in this test, the test location template is sandwiched between training location templates, as shown in Figure 6.3-3. This test aims to examine the localisation performance of LTPM when an acoustic source is nearby multiple pre-defined locations

on different location templates.
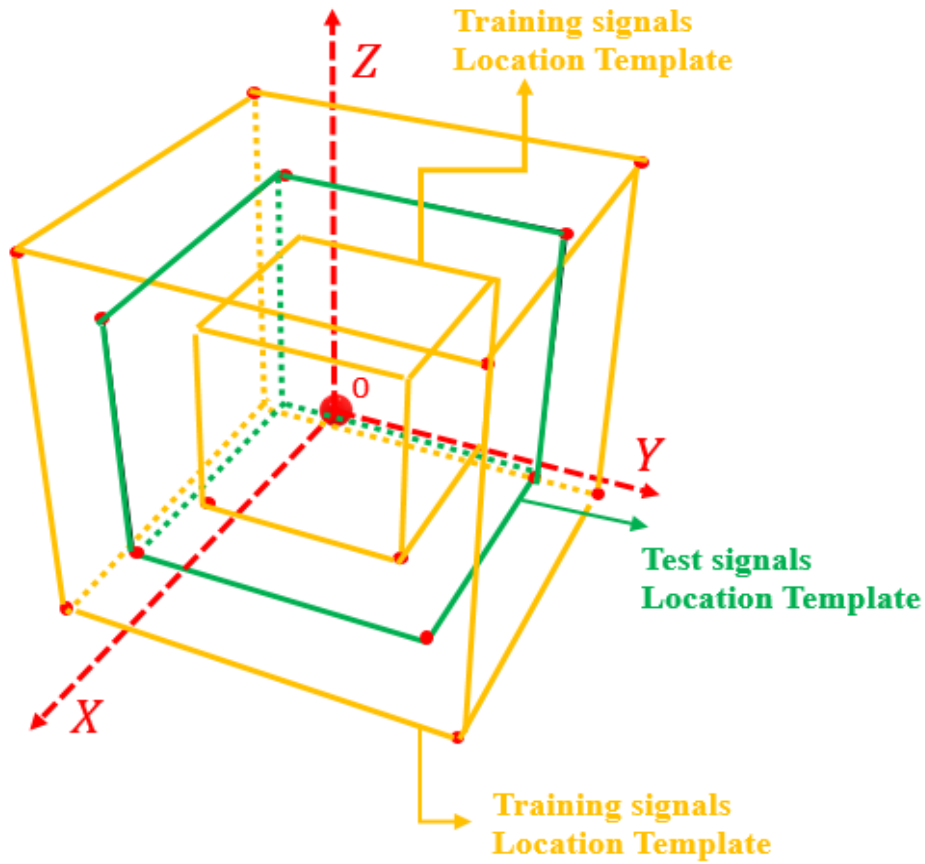


Figure 6.3-3 Illustrations of the location templates for the test signals and training signals. The side lengths of the two yellow location templates are 800 mm and 1000 mm, respectively. The green template is used for testing, with a side length of 900 mm

The test consists of four parts; in the 1st part, the model trained with 3 primary datasets is tested with the 900 mm dataset. Locations at the coaxial lines are defined as linear locations; thus, the output of the LTPM includes 8 locations. For a successful localisation, signals from the 900 mm dataset are supposed to be located at adjacent coaxial locations. The 2nd part is similar to the 1st part, but all locations are independent. A localisation is considered successful when signals from the 900 mm location template are located to any nearest adjacent locations.

In the 3rd part, the 900 mm dataset is used to test the LTPM trained with the 800 mm dataset. In the 4th part, the 900 mm dataset tests the LTPM trained with the 1000 mm dataset. These tests examine the classification accuracy of LTPM in the situation that an

148

acoustic source is placed in the middle of two location templates (100 mm deviation distance to both locations templates). The results are shown in Table 6.3-6.

Table 6.3-6 Cross test results of 900 mm against 800 mm,1000 mm and 1200 mm

| Method | LTPM-CT-8(900-(800+1000+1200)) | LTPM-CT-24(900-(800+1000+1200)) | LTPM-CT-8(900-(800)) | LTPM-CT-8(900-(1000)) |
|---|---|---|---|---|
| RF(%) | 5 | 5 | 2.5 | 12.5 |
| CNN(%) | 81.15 | - | 82.04 | 81.03 |

The test results are similar to the test results acquired from the 750 mm dataset to the 3 primary datasets test. However, the overall localisation accuracy is lower. The reason is that the deviation distance between the test acoustic signals and the pre-defined locations has increased from 50 mm to 100 mm. Still, CNN maintains accurate coaxial classification accuracy, with an average accuracy of 81% throughout the test.

1050 mm dataset to 3 primary datasets (8 points and 24 points):

Similar to the previous test, the classification accuracy of LTPM is measured with 1050 mm location templates in this test. Acoustic signals collected from the 1050 mm location template are used to test the localisation performance of the model trained with 800 mm, 1000 mm and 1200 mm datasets.

The test consists of 4 parts; in the 1st part, the model trained with 3 primary datasets is tested with the 1050 mm dataset. Locations at the coaxial lines are defined as linear locations; thus, the output of the LTPM includes 8 locations. For a successful localisation, signals from the 1050 mm location template are located at corresponding coaxial linear locations. In the 2nd part, all locations are independent; thus, 24 locations are defined in the output.

In the 3rd part, the 1050 mm dataset tests the LTPM trained with the 1000 mm dataset (50 mm deviation distance). In the 4th part, the 1050 mm dataset tests the LTPM trained with the 1200 mm dataset (150 mm deviation distance).

The results are shown in Table 6.3-7.

Table 6.3-7 Cross test results of 1050 mm against 800 mm, 1000 mm and 1200 mm

| Method | LTPM-CT-8(1050-(800+1000+1200)) | LTPM-CT-24(1050-(800+1000+1200)) | LTPM-CT-8(1050-(1000)) | LTPM-CT-8(1050-(1200)) |
|--------|--------|--------|--------|--------|
| RF(%) | 15 | 17.5 | 12.5 | 40 |
| CNN(%) | 88.97 | - | 79.60 | 86.25 |

The classification accuracy of the RF-based LTPM is between 12.5% and 40%. The classification accuracy of the CNN-based LTPM is between 80% and 89%. A comprehensive summary is presented in the cross test between 3 primary datasets.

1100 mm dataset to primary datasets (8 points and 24 points):

In this test, the dataset collected from the 1100 mm location template is used to measure the classification accuracy of models trained with 800 mm, 1000 mm and 1200 mm datasets.

The test consists of four parts; in the 1st part, the model trained with the 3 primary datasets is tested with the 1100 mm dataset, while the 2nd part is an independent localisation test (24 locations). In the 3rd part, the 1100 mm dataset tests the LTPM trained with the 1000 mm dataset. In the 4th part, the 1100 mm dataset tests the LTPM trained with the 1200 mm dataset. The results are shown in Table 6.3-8.

Table 6.3-8 Cross test results of 1050 mm against 1000 mm and 1200 mm

| Method | LTM-CT-8(1100-(800+1000+1200)) | LTM-CT-24(1100-(800+1000+1200)) | LTM-CT-8(1100-(1000)) | LTM-CT-8(1100-(1200)) |
|--------|--------|--------|--------|--------|
| RF(%) | 27.5 | 17.5 | 0.00 | 30 |
| CNN(%) | 79.03 | - | 82.72 | 86.58 |

The classification accuracy of the RF-based LTPM is between 0% and 30%. The localisation accuracy of the CNN-based LTPM is between 80% and 87%. The test results are similar to the test results of the last test. A comprehensive summary is presented in the cross test between 3 primary datasets.

The secondary dataset to the primary datasets (8 points and 24 points):

All secondary datasets are used to test the LTPM trained with the 3 primary datasets in

this test. The test consists of an 8 points test and a 24 points test. The results are shown in Table 6.3-9.

Table 6.3-9 Cross test results of 750 mm, 900 mm, 1050 mm and 1100 mm against 800 mm,1000 mm and1200 mm

| Method | LTPM-CT-8((750+900+1050+1100)-(800+1000+1200)) | LTPM-CT-24((750+900+1050+1100)-(800+1000+1200)) |
|---|---|---|
| RF(%) | 20.62 | 16.88 |
| CNN(%) | - | - |

In this test, the input acoustic signals are collected from 4 cubic location templates thus the number of locations in the input data is 32. The acoustic signals for the training are collected from 3 cubic location templates thus the number of locations in the training data is 24. The number of location labels of the test data and the number of location labels of the training data is different. Hence, the CNN-based LTPM is not involved in this test since it requires consistent input and output labels.

In the 8 points test, the classification accuracy of the RF-based LTPM is 20.62% because independent locations on the coaxial line are linearly connected. The classification accuracy of the RF-based LTPM further decreases to 16.88% in the 24 points test. In summary, the classification accuracy of the RF-based LTPM decreases to 26% approximately (previous test results referenced) when acoustic sources are 50 mm away from pre-defined locations. The classification accuracy further decreases to 15% when the deviation distance increases to 100 mm. In conclusion, the classification accuracy of the RF-based LTPM is reduced with the increase of the deviation distance.

To this extent, localisation tests on 50 mm and 100 mm deviation distances have been accomplished. The distance between the acoustic source and the pre-defined locations is further increased to 200 mm in the next test.

Cross test between 3 primary datasets (8 points and 16 points):

3 primary datasets are used mutually for classification accuracy tests. Similar to previous cross tests, this test aims to measure the classification accuracy when the distance between the acoustic source and pre-defined locations increases to 200 mm. Sufficient

training data and test data are provided in this test. The amount of training samples is over 80,000.

The 3 primary datasets are combined in pairs to train LTPMs. Therefore, the output contains 16 locations (2 location templates have 16 cube vertices). The test consists of six parts; in the 1st part, the model is trained with the 1000 mm dataset and the 1200 mm dataset is tested with the 800 mm dataset. Locations at the coaxial line are defined as linear locations; thus, the output of the LTPM includes 8 locations. The 2nd part is similar to the 1st part, but all locations are independent; thus, the output contains 16 locations.

The 3rd part is a coaxial test; the 1000 mm dataset is used to test the LTPM trained with the 800 mm dataset and 1200 mm dataset. The test location template (1000 mm) is sandwiched between the two training location templates (800 mm and 1200 mm). The 4th part is similar to the 3rd part, but it is an independent test with 16 output locations.

The 5th part is a coaxial test; the 1200 mm dataset is used to test the LTPM trained with the 800 mm dataset and 1000 mm dataset. The two training location templates (800 mm and 1000 mm) are wrapped by the test location template (1200 mm). The 6th part is similar to the 5th part, but it is an independent test with 16 output locations.

The test determines the localisation accuracy when an acoustic source is placed outside or within the location template. In the second level of cross test, the location template of the test dataset is gradually moving away from the location template of the training dataset so that the relation between acoustic signals and distance can be observed. The results are shown in Table 6.3-10.

Table 6.3-10 Cross test results of 800 mm,1000 mm and 1200 mm

| Method | LTPM-CT-8(800-(1000+1200)) | LTPM-CT-16(800-(1000+1200)) | LTPM-CT-8(1000-(800+1200)) | LTPM-CT-16(1000-(800+1200)) | LTPM-CT-8(1200-(800+1000)) | LTPM-CT-16(1200-(800+1000)) |
|---|---|---|---|---|---|---|
| RF(%) | 5.01 | 4.74 | 9.72 | 8.67 | 8.55 | 0.00 |
| CNN(%) | 89.14 | - | 79.55 | - | 83.10 | - |

The overall classification accuracy of the RF-based LTPM is less than 10% in this test. The RF-based LTPM can barely locate any acoustic sources to the closest pre-defined

locations when acoustic sources are 200 mm away from pre-defined locations. While the CNN-based LTPM maintains a classification accuracy between 79.55% and 89.14% as the CNN-based LTPM still accurately locates acoustic sources to adjacent locations.

The 800mm, 1000mm, 1200mm, 460mm, 1100 mm, 1050 mm, 900 mm and 750 mm datasets are used as the test data in the above seven tests to examine the classification accuracy of LTPM in the situation that acoustic sources deviate from the pre-defined locations. The overall classification accuracy of the RF-based LTPM reduces to 26% at a deviation distance of 50 mm. Furthermore, the classification accuracy is reduced to 15% at a deviation distance of 100 mm and 8% at a deviation distance of 200 mm. The relationship between the classification accuracy and the deviation distance is shown in Figure 6.3-4.
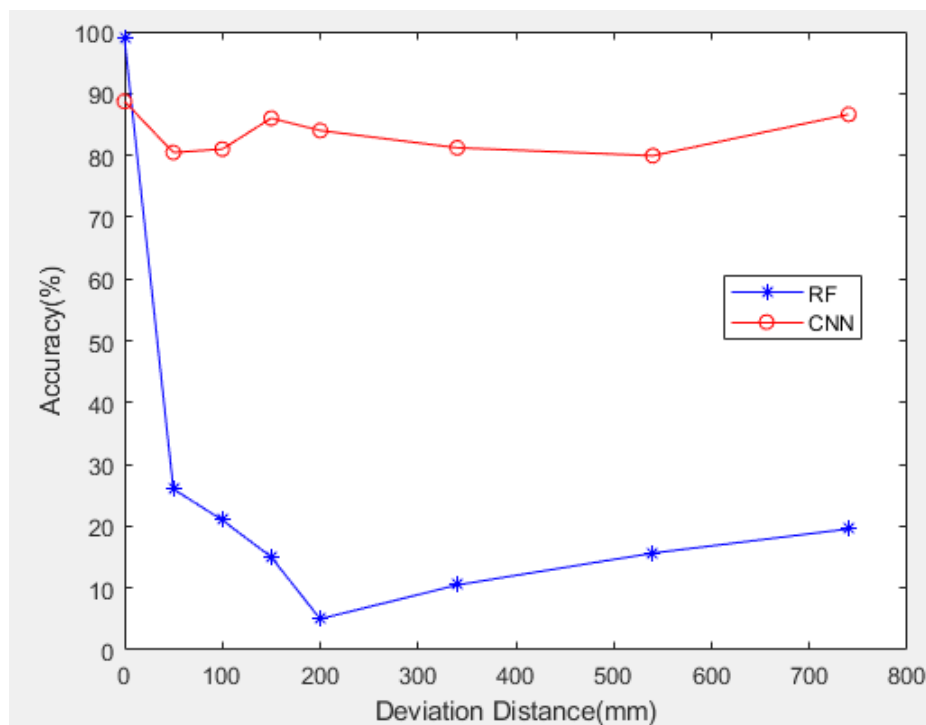


Figure 6.3-4 The relation between the classification accuracy and the deviation distance. The classification accuracy of RF-based LTPM decreases with the distance, while the classification accuracy of CNN-based LTPM maintains the same level

The CNN-based LTPM, on the other hand, shows the property of inheritance. It accurately locates acoustic sources to adjacent pre-defined locations in all tests.

Reduced scale tests:

In this test. The volume of the training data is reduced to examine whether the trained model maintains the same level of classification accuracy achieved in previous tests. The training datasets used in this test consist of the 3 primary datasets. The test datasets are the 460 mm, 1050 mm and 1200 mm datasets.

Another test objective is to conduct a comparison test between RF-based LTPM and CNN-based LTPM. The same amount of training data is used to train RF-based LTPM and CNN-based LTPM. Since the data volume of CNN is smaller than the data volume of RF, the threshold control of the data volume in this test is based on the data volume of CNN. Namely, 40800 samples in each dataset.

The test consists of three parts; in the 1st part, the model trained with 3 primary datasets is tested with the 460 mm dataset. The training data volume of RF-based LTPM is reduced to 32640 samples, and the test data volume is 8160. Training and test data volume of CNN-based LTPM maintains the same since they are the benchmark in this test. In the 2nd part, the model trained with 3 primary datasets (reduced scale) is tested with the 1050 mm dataset. The 3rd part belongs to the first level of cross test. The data from a location template is divided into two parts for training and testing; thus, a dataset which contains 40800 samples is extracted from the 1200 mm dataset. The results are shown in Table 6.3-11.

Table 6.3-11 Reduced scale test results. RS represents for Reduced-Scale

| Method | LTPM-CT-RS-8(460-(800+1000+1200)) | LTPM-CT-RS-8(1050-(800+1000+1200)) | LTPM-LT-RS-8(1200) |
|---|---|---|---|
| RF(%) | 17.75 | 27.50 | 94.00 |
| - | LTPM-CT-8(460-(800+1000+1200)) | LTPM-CT-8(1050-(800+1000+1200)) | LTPM-LT-8(1200) |
| CNN(%) | 87.20 | 88.97 | 87.30 |

The classification accuracy of the RF-based LTPM increases slightly (17.75% and 27.5%) compared to previous test results because the data volume of the training data is

reduced. The test result in the 3rd column indicates that the LTPM still maintains an excellent classification accuracy with less training data.

The RF-based LTPM achieved a classification accuracy of 94% in the first level of cross test, but the result does not imply that the RF-based LTPM is superior to the CNN-based LTPM. The two machine learning algorithms have different properties. The CNN-based LTPM always outperforms the RF-based LTPM in coaxial localisation tests. And the RF-based LTPM has a high classification accuracy in independent tests. In summary, a single machine learning-based LTPM always has bottlenecks; reasonable utilisation of complementary machine learning algorithms may improve the versatility of LTPM significantly. Related discussions are presented in future works.

## 6.4 Result analysis

### 6.4.1 Systematic and environmental factors

In an ideal situation, acoustic features extracted from an input acoustic signal should perfectly match a set of features in the training dataset so that the location of the input acoustic signal is calculated precisely by outputting the matched sample's coordinate. However, potential systematic and environmental variables cause certain localisation errors in practical situations.

Firstly, the distortion of the signal source is an important factor. A sampling sequence is shown in Figure 6.4-1. The signal amplitude changes with time though the environmental conditions are strictly maintained. The change of the acoustic source causes the quantised values of sampled signals to float unexpectedly, resulting in matching errors.

To solve the problem, more signals are collected and added to the training dataset. The idea is to cover signal variations by extending the data volume. Time-varying acoustic signals are collected by sampling acoustic signals at different periods. LTPM classifies acoustic signals from different locations stably (with an average classification accuracy of 94%) after the number of signals in the training dataset reaches 30,000. In other words, LTPM adapts to signal variations (e.g., sudden changes in amplitude) by adding more training data to the training dataset, eliminating errors caused by signal source distortions.
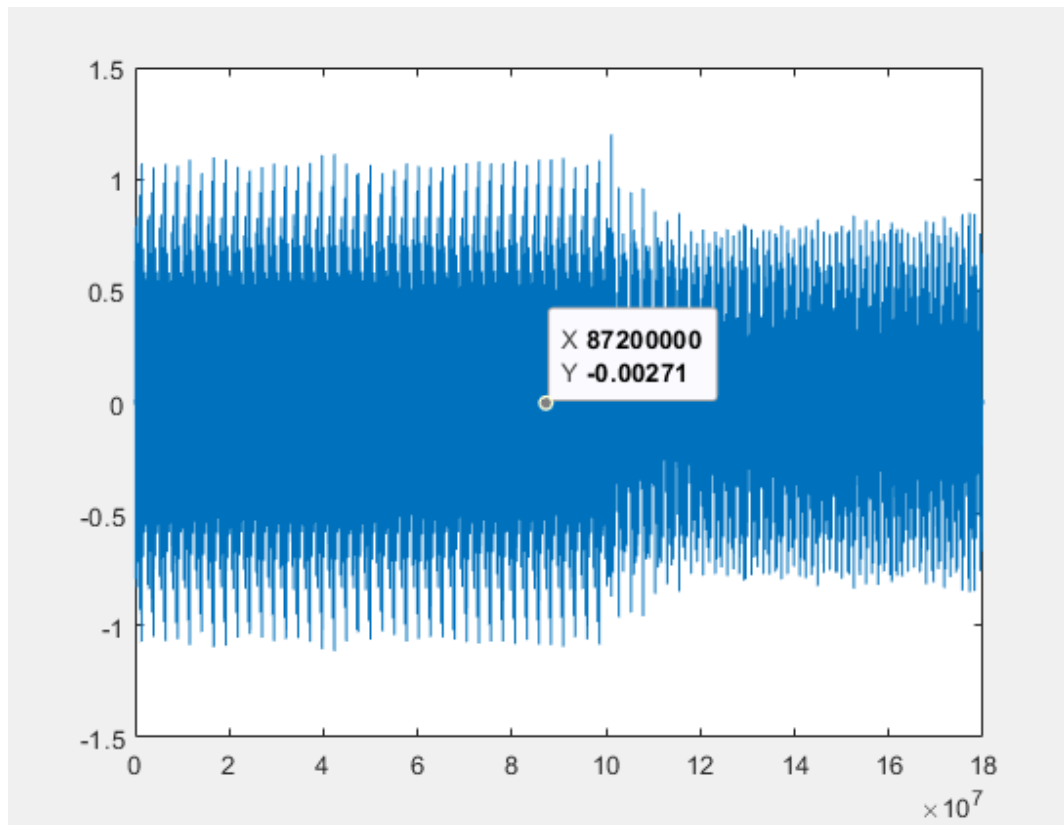
Figure 6.4-1 Sudden change of a signal sequence. The X-axis is the sampling point (time), and the Y-axis is the signal's amplitude (voltage). Although the power supply is stable and the environment is strictly maintained, signal variations still exist. The absolute mean amplitude of signals reduces to 0.75 at the X-axis point $10 * 10^7$. In the test, LTPM adapts to the variations due to the utilisation of multiple features and extensive data collection

Another factor is feature similarity. An acoustic source may generate similar signal features at different locations due to the multipath effect, resulting in feature overlaps. In the three-dimensional localisation test, the overlapping rate of acoustic features is less than 2%. Thus feature similarity is not a serious problem. Related solutions include adjusting the weights of acoustic features, non-symmetry sensor deployment and pre-correlation of the database.

In addition, if the sensor is too far from the localisation zone. The path loss will replace the multipath effect and play a dominant role in matching processing. In this case, LTPM is similar to the classic Location Fingerprint technology. Predictably, the localisation accuracy and classification accuracy of LTPM will decrease rapidly due to the inefficient utilisation of acoustic features. Solutions include using adaptive filters, adjusting the

156

locations of sensors, and using a microphone array instead of a single microphone.

*6.4.2 Probability distribution of features*

The probability distribution is the foundation of machine learning, while the foundation of probability distribution is the normal distribution [19]. Understanding and estimating the probability distributions of object variables is crucial in analysing the localisation performance of LTPM from the perspective of the high-dimensional joint probability distribution. Meanwhile, the training of the positioning model can be simplified if the probability distributions of object variables (acoustic features) follow the normal distribution.

Assuming that the distribution of each feature follows the normal distribution, the probability of feature distribution can be calculated according to the probability density function (Eq. 6-23).

$$f(x) = \frac{1}{\sigma\sqrt{2*\pi}}e^{-\frac{1}{2}(\frac{x-u}{\sigma})^2}$$

6-23

where $\sigma$ is the standard deviation of the signal feature, and $u$ is the mean value. And f(x) is the probability of the feature's distribution. Suppose the probability distribution of each acoustic feature follows the normal distribution. The distribution curve of a signal feature from different locations should be similar to that of the normal distribution. Meanwhile, the variances of the normal distributions of acoustic features from different locations should be at the same level.

However, as shown in Figure 6.4-2, the two distributions of signal length from two different locations are inconsistent. The left distribution is asymmetric, while the right distribution can be roughly regarded as a left-biased normal distribution curve. Besides, the variances of acoustic features at different locations increase with the distance between the microphone and the acoustic source. Therefore, it is preliminarily judged that distributions of acoustic features do not follow the normal distribution.
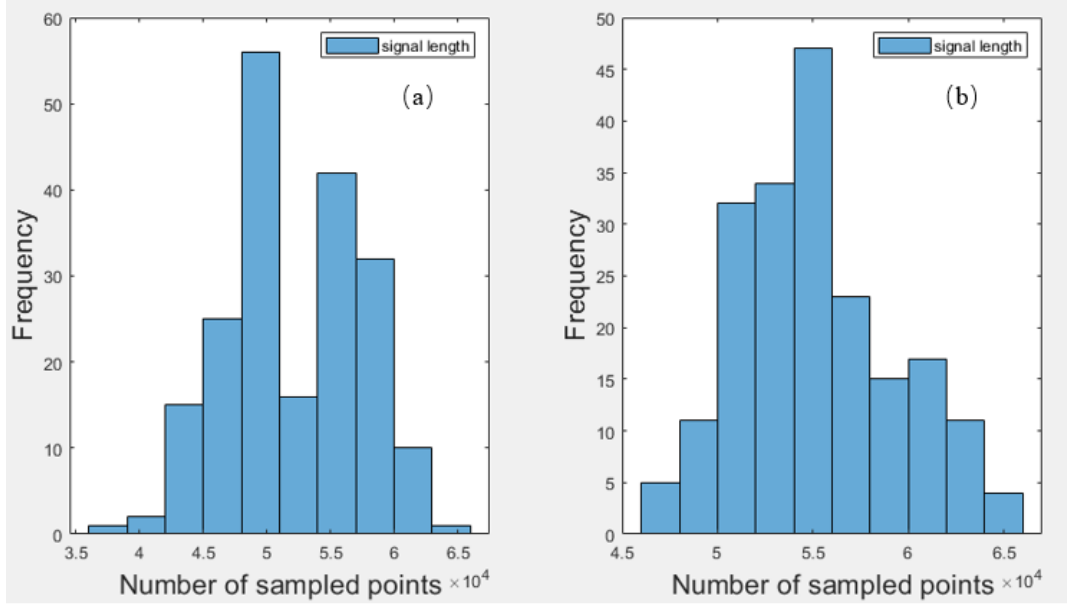
Figure 6.4-2 Signal length distributions of two signals sampled at location 2 (a, close to the microphone), and location 8 (b, far away from the microphone). The variances of these two distributions are 1.8043 and 2.8794, respectively

Moreover, the variance of the probability distribution of each acoustic feature should affect the overall classification accuracy of LTPM [134]. A criterion is that the classification accuracy should increase as the variance of each signal feature decreases. However, variances of probability distributions of features vary significantly, and the classification accuracy barely changes in the test. This phenomenon may be caused by utilising multiple features since each feature has a limited impact on the final output.

In summary, the probability distributions of acoustic features do not follow the normal distribution, as no similar distribution patterns are observed. The variances of signal features also vary greatly. Thus, the probability expression of the classification accuracy cannot be deduced according to the normal distribution. But acoustic signals at different locations are still distinguishable with machine learning. The test results indicate that machine learning algorithms are appropriate for pattern matching-based acoustic localisation tasks. Acoustic localisation is also the new application field of machine learning.

## 6.5 Summary

The proposed localisation technology, Location Template-based Positioning Model, is

tested strictly in the three-dimensional localisation test. Test results show that the LTPM-based localisation system has achieved a three-dimensional localisation accuracy of 173 mm.

In traditional time difference-based localisation technologies, the multipath effect is usually considered negative as it causes interferences to signal reception; thus, various techniques are used to eliminate the interferences caused by the multipath effect. In indoor environments, the multipath effect is further amplified, and it becomes difficult to eliminate signal components generated by the multipath effect.

LTPM successfully utilises the multipath effect to achieve three-dimensional acoustic localisation rather than eliminating the multipath effect. LTPM uses multiple acoustic features, pattern matching and machine learning to achieve acoustic source localisation in a three-dimensional space. The proposed acoustic localisation technology does not rely on sensor array deployment, signal filter and time differential localisation algorithms. Besides, LTPM overcomes the drawbacks of TDOA-based electromagnetic localisation technologies as it has robust environmental adaptability. The positioning performance of LTPM varies with the data volume, the defined acoustic features, and the classification performance of the matching algorithm.
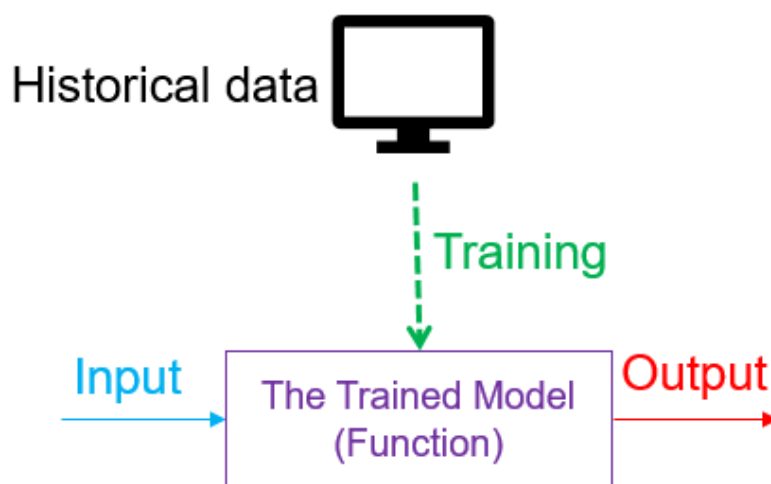


Figure 6.5-1 System block diagram of LTPM. The positioning model is continuously optimised in the training stage to match training acoustic signals with allocated coordinates. The trained model directly calculates the coordinate of the acoustic source when an input signal is provided

159

In this project, signal processing has been successfully combined with machine learning to achieve three-dimensional acoustic source localisation. Machine learning algorithms are used to find and utilise hidden signal patterns in the training data. The signal features integrated into LTPM are no longer limited to time-domain features. 43 features from multiple domains are extracted with feature extraction and signal transformation techniques. The signal usage rate has been improved significantly in comparison to the signal usage rate of the classic Location Fingerprint. The test results demonstrate that LTPM is a successful short-range acoustic localisation technology. At the same time, this research opened a new research field for machine learning, namely, three-dimensional acoustic source localisation in terms of human-computer interaction.

The Random Forest and the Convolutional Neural Network are utilised to classify acoustic signals from different locations in LTPM. The format of input data for RF is a 43-dimensional feature matrix. In contrast, 17 signal images are defined for CNN. The classification model is updated iteratively during the training and models that calculate the coordinates of acoustic sources according to the input acoustic signal are obtained after the training completes. The system block diagram is illustrated in Figure 6.5-1.

In the implementation of LTPM, an automatic robot is used to collect acoustic signals. The collected acoustic signals are processed as input data according to the requirements of different machine learning algorithms. Other modules, such as sampling modules, signal processing modules and feature extraction modules, are also optimised to meet the requirements of the three-dimensional acoustic localisation test.

A series of tests are designed to determine the positioning performance of LTPM. The relationship between the localisation accuracy and the classification accuracy of the LTPM-based three-dimensional acoustic localisation system is shown in Figure 6.5-2.
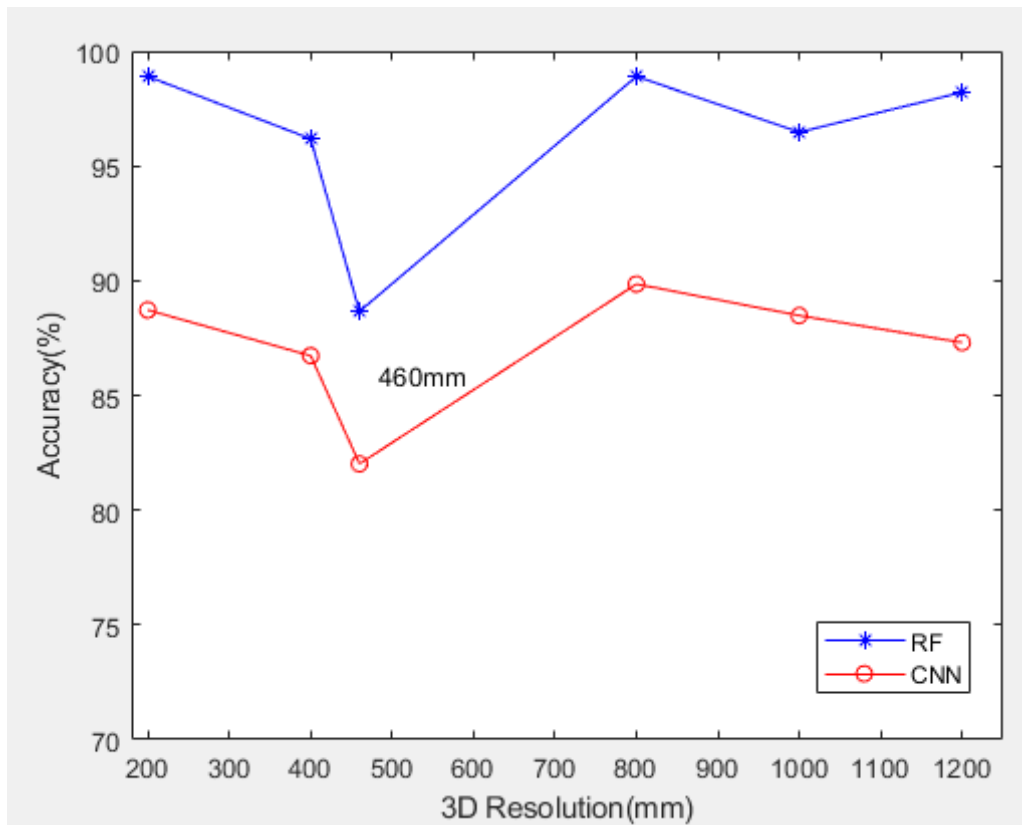
Figure 6.5-2 The Localisation accuracy and classification accuracy line chart

As shown in Figure 6.5-2, the RF-based LTPM maintains a classification accuracy of 95%, and the CNN-based LTPM maintains a classification accuracy of 85% throughout the test. The classification accuracy is decreased at the 460 mm accuracy since the 460 mm dataset only contains 3,200 template acoustic signals. Both machine learning algorithms used in the three-dimensional localisation test reached an accuracy of 173 mm, but the classification accuracy of the CNN-based LTPM is inferior to the classification accuracy of the RF-based LTPM.

The RF-based LTPM is highly sensitive to acoustic sources that deviate from pre-defined locations. The classification accuracy of the RF-based LTPM decreases when the acoustic source deviates from pre-defined locations, and the attenuation of the accuracy is related to the deviation distance. The average classification accuracy of the RF-based LTPM reduces to 26% when the deviation distance between the acoustic source and the pre-defined locations increases to 50 mm. Furthermore, the classification accuracy is reduced to 15% at a deviation distance of 100 mm and 8% at a deviation distance of 200 mm. The relationship between the classification accuracy and the deviation distance is

161

summarised in Figure 6.3-4.

Unlike the RF-based LTPM, the CNN-based LTPM has a property of inheritance as it always maintains a precise classification accuracy. On the other hand, it can be summarised that acoustic signals collected from coaxial locations share some connections in consecutive signal transformations; the neural network summarises the hidden signal patterns and utilises the signal patterns for accurate signal classification; thus, the coaxial classification accuracy of the CNN-based LTPM is always maintained at 80% approximately.

Overall, the RF-based LTPM has achieved a superior three-dimensional accuracy of 173 mm in 95% of the location estimates in the indoor environment. In contrast, the CNN-based LTPM has achieved a three-dimensional accuracy of 173 mm in 85% of the location estimates in the indoor environment. However, the CNN-based LTPM is still worth exploiting as it has a higher coaxial classification accuracy. The three-dimensional localisation test results verify that the multipath effect endows acoustic signals with unique features, and these features are utilised by LTPM successfully to achieve short-range acoustic localisation.

The following objectives are accomplished in the three-dimensional localisation test;

(1) A complete acoustic localisation system is built.

(2) A large number (150,000) of acoustic signals are collected.

(3) 43 acoustic features are defined for the Random Forest-based LTPM, and 17 signal images are converted for the Convolutional Neural Network-based LTPM.

(4) The positioning performance of LTPM is rigorously evaluated with a series of three-dimensional localisation tests.

(5) Factors that cause matching errors and characteristics of the LTPM-based localisation system are discussed and summarised.

In the three-dimensional localisation test, one microphone is deployed in the indoor environment for signal collection and acoustic source localisation; thus, LTPM has a superior advantage in system cost as it only requires one sensor. LTPM is a low-cost

enabling localisation technology for human-computer interfaces which have indoor positioning demands. But LTPM needs sufficient data to ensure accurate localisation results. As a localisation technology which has fused multiple technologies, the advantages and disadvantages of LTPM and a comprehensive analysis of LTPM is presented in Chapter 7.

# 7 Final discussions and conclusions

## 7.1 Conclusions

### 7.1.1 The accomplishment of the research aim and research objectives

The research aims to provide three-dimensional human-computer interaction with a passive enabling localisation technology. The ubiquitous acoustic waves are ideal signal sources for the TPM-based acoustic localisation technology. In this research, acoustic signal processing is combined with pattern matching to create a novel acoustic localisation technology - Location Template-based Positioning Model (LTPM). LTPM accurately locates acoustic sources in the three-dimensional indoor space and it overcomes the drawbacks of TDOA-based electromagnetic localisation technologies.

The Location Fingerprint is the early pattern matching-based localisation technology. It achieved different scales of accuracy (from 0.5m to 188m) [28]. Soon afterwards, the EU project - TAI-CHI developed a TDOA-based acoustic localisation system, which achieved a two-dimensional accuracy of 14 mm [7]. Inspired by these achievements, the Location Template-based Positioning Model is proposed to realise three-dimensional acoustic localisation in indoor environments.

The positioning performance of LTPM is determined with a series of two-dimensional and three-dimensional localisation tests. Both two-dimensional and three-dimensional localisation systems use pattern matching-based localisation principles but have different technical details. From the perspective of project development, the two-dimensional acoustic localisation system is the foundation of the three-dimensional acoustic localisation system. More acoustic samples and more complex machine learning algorithms are utilised to achieve acoustic source localisation in the three-dimensional localisation test.

The entire research procedure follows the spiral project processing model proposed in Chapter 3. The advantages and disadvantages of LTPM have been documented during the system implementations and localisation tests. Details of technical problems are discussed in future works.

The accomplished research objectives are listed as follows:

(1) An acoustic localisation technology - LTPM, which is capable of locating acoustic sources in complex three-dimensional indoor environments, is proposed, implemented and tested successfully.

(2) Two fully functional localisation systems have been built successfully for two-dimensional and three-dimensional localisation tests. These systems cover all the functions required by the proposed LTPM (such as signal sampling, signal filtering, feature extraction and signal matching).

(3) In the two-dimensional localisation test, three acoustic features are defined and a deterministic cross-correlation matching algorithm is compiled to match the input acoustic signal with the template acoustic signals. Overall, the localisation system achieved a two-dimensional accuracy of 30mm in 80% of the location estimates. In the three-dimensional localisation test, 43 acoustic features and 17 signal images are extracted and transformed for the RF-based LTPM and the CNN-based LTPM. The RF-based LTPM has achieved a superior three-dimensional accuracy of 173 mm in 95% of the location estimates in the indoor environment. In contrast, the CNN-based LTPM has also achieved a three-dimensional accuracy of 173 mm in 85% of the location estimates in the indoor environment.

(4) Signal processing is successfully combined with machine learning to realise acoustic signal matching. Two machine learning algorithm-based LTPMs are built to find and utilise the hidden signal patterns in the training data. On the other hand, the training data provided to the two machine learning algorithms is also extended to multiple signal domains with feature extraction and signal transformation techniques. The signal utilisation rate has been improved significantly in comparison to the signal utilisation rate in the RSS-based Location Fingerprint.

(5) The test results indicate that three-dimensional short-range acoustic localisation is feasible. LTPM locates acoustic sources with only one microphone, and it has robust adaptability to complex indoor environments.

(6) To design LTPM and build the LTPM-based localisation systems, interactive technologies, electromagnetic and acoustic localisation technologies, signal processing technologies and machine learning are studied and reviewed. The latest achievements in short-range localisation and the research direction of indoor localisation technology are listed and analysed. Overall, this research provides a comprehensive reference for short-range acoustic localisation technology.

In this project, a novel acoustic localisation technology is proposed and tested. At the beginning of the research, properties, characteristics, physical effects, and applications of acoustic waves were reviewed and referenced carefully to establish the physical foundations and the localisation theory for the proposed three-dimensional acoustic localisation technology. Since the acoustic multipath effect plays a vital role in acoustic propagation and it is challenging to eliminate the acoustic signal variations affected by the acoustic multipath effect, a localisation theory that utilises the multipath effect for acoustic source localisation is established.

Meanwhile, various localisation technologies were evaluated and the shortcomings and technical limitations of TDOA-based localisation technologies were summarised. The proposed localisation technology utilises the pattern matching-based matching algorithm as the localisation algorithm in response to these shortcomings.

To collect massive template acoustic signals for LTPM, the concept of the location template is proposed. The location template is an essential element for the Location Template-based Positioning Model. The features and images of acoustic signals are extracted and transformed by processing the collected acoustic signals. Next, the acoustic features and images are loaded by the machine learning algorithm for the training of the positioning model.

From the perspective of signal processing, feature selection is crucial to the positioning performance of LTPM. As mentioned in Chapter 4, the positioning performance of LTPM depends on the effectiveness of selected acoustic features and the classification performance of the localisation algorithm. Therefore, multiple acoustic features are

defined and integrated into LTPM to provide balanced matching references for acoustic signal matching.

Machine learning algorithms are used to solve the boundary problem caused by sample overlapping. The two machine learning algorithms are capable of classifying acoustic signals from different physical locations according to the test results. Moreover, supervised learning and deep learning-based LTPMs have different properties. Test results are summarised in the next section.

*7.1.2 Comparison with other localisation technologies*

LTPM is first applied to two-dimensional localisation. The LTPM-based localisation system achieved a two-dimensional accuracy of 30 mm in 80% of the location estimates.

The feasibility of LTPM is preliminarily verified with the two-dimensional localisation test result. Next, a three-dimensional localisation system is developed based on the two-dimensional localisation system. A series of localisation tests have been conducted to evaluate the positioning performance of the LTPM-based three-dimensional localisation system.

In the situation that 120,000 sample signals are collected and processed as the training data, the RF-based LTPM localisation system reached a three-dimensional accuracy of 173 mm in 98% of the location estimates. On the other hand, the CNN-based LTPM localisation system reached the same three-dimensional accuracy in 80% of the location estimates.

However, the CNN-based LTPM can locate acoustic sources that are deviated from pre-defined spatial locations to adjacent pre-defined locations. On the contrary, the RF-based LTPM only locates an acoustic source accurately if acoustic sources are precisely placed at pre-defined locations.

Technically, difficulties in pattern matching-based acoustic localisation lie in the signal collection, signal processing and classification of acoustic signals. These difficulties are systematically addressed one by one in this project. In contrast, problems in TDOA-based electromagnetic localisation technologies lie in the sensor deployment and

environmental optimisations. As a result, the system cost of TDOA-based electromagnetic localisation technologies is high. In this case, the pattern matching-based acoustic localisation technology is competitive because of its strong environmental adaptability and low system cost. As indoor localisation receives more and more attention, LTPM has the potential to become the cornerstone for indoor localisation applications.

Table 7.1-1 presents a comprehensive comparison between existing short-range localisation technologies. LTPM has the best three-dimensional accuracy and it only needs one microphone. TAI-CHI has the best two-dimensional accuracy, while UWB is the most promising TDOA-based electromagnetic indoor localisation technology.

Table 7.1-1 Comparison between different localisation technologies

| Method | Pattern | Mode | Appliacable medium | Accuracy | Sensor | Sensor deployment | Calculation cost | Anti interference | Application scale |
|--------|---------|------|--------------------|----------|--------|-------------------|------------------|-------------------|-------------------|
| Location Template-based Positioning Model | Improved pattern matching | **Passive** | Air and solid | **200mm** | Microphone | **1** | **High in training** Low in positioning | **Medium** | 3D 1200mm cubic space |
| Location Template Matching [82] | Pattern matching | Active | Air and solid | 540 mm | Wireless switch | **2 or more** | Low | Strong | 3D room scale |
| TAI-CHI[7] | TDOA | **Passive** | Solid | **14mm** | Piezoceramic sensor | 4 | Low | Medium | 2D 1000mm square solid plate |
| UWB[69] | TDOA | Active | Air | **225 mm** | Transceiver | **5 or more** | **High** | **Strong** | 3D room scale |
| Tangible Media[61] | Computer vision | Active | Air | - | Optical camera | 1 | High | Medium | 3D interface |
| 3D haptic shape[9] | Ultrasonic holograms | Active | Air and liquid | - | Transducer | Array | Medium | Low | 3D interface |

## 7.2 Criteria of finished work

Some technical problems are exposed during the implementation of LTPM. Criteria that affect the performance of LTPM and issues in practical application are listed as follows:

(1) The Kalman filter did not achieve the expected signal processing results. According to the research plan, the Kalman filter is implemented to improve the signal-to-noise ratio, as shown in Figure 7.2-1. However, the comparison between the original signal and the filtered signal shows that the Kalman filter erased most of the multipath signal components in the received acoustic signal. This property of

the Kalman filter reduces the accuracy of signal matching. Thus, the Kalman filter has not been adopted in formal localisation tests.
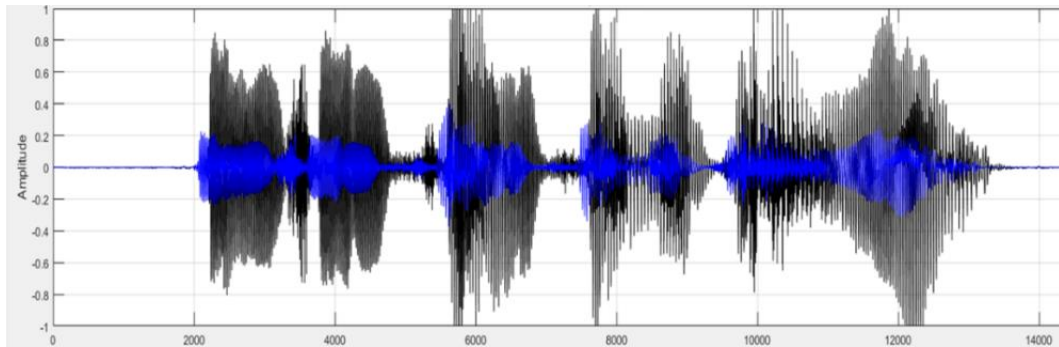


Figure 7.2-1 Comparison between an original acoustic signal (black) and the acoustic signal after passing the Kalman filter (blue). The X-axis is the sampling point, while the Y-axis is the normalised amplitude. It can be observed that the components of the filtered signals are reduced by half, and the tail of each independent signal is compressed

However, the Kalman filter unexpectedly facilitated blind source separation of acoustic signals. After a signal sequence passes the Kalman filter, the starting point of each independent acoustic signal can be observed clearly. Therefore, in future work, the performance of the signal separation algorithm can be enhanced by integrating the Kalman filter.

(2) Acoustic sources with single frequencies are used in localisation tests. Though different buzzers are deployed as acoustic sources in the three-dimensional localisation test, the frequency bands of these buzzers are fixed. Such acoustic sources facilitated the signal processing, but acoustic waves generated by these buzzers differ from natural acoustic sources in terms of frequency, signal energy and randomness.

According to the decomposition principle of acoustic waves (introduced in Chapter 2), if acoustic waves with single frequencies can be utilised to achieve three-dimensional acoustic localisation, a realistic acoustic source with complex frequency components can also be located with the same localisation method and the Fourier decomposition. However, acoustic waves generated by biological sources may change drastically and stochastically in actual localisation tasks.

169

Therefore, research on applicable natural acoustic sources is essential before applying LTPM to practical human-computer interactive applications.

(3) The utilisation of the microphone array is unsuccessful. Although LTPM only requires one sensor to achieve acoustic localisation, the deployment of multiple sensors still benefits LTPM since a microphone array receives more signal details, thus making signal features more distinctive. As a result, clear boundaries between signal clusters can be formed with the microphone array technology. However, a microphone array requires complicated differential or summary algorithms to ensure precise acoustic signal receptions and no commercial microphone array systems are found; thus, the microphone array is abandoned.

(4) The design of the location template is conservative. The location templates used in the three-dimensional localisation test should be denser, and more sample signals should be collected. Although LTPM achieved a three-dimensional accuracy of 173 mm, its localisation potential has yet to be fully exploited.

However, if more location templates are designed and compiled, the signal processing and feature extraction modules would require more computational resources and time to process the collected acoustic signals. Therefore, the balance between the design of the location template and the data processing must be considered according to practical test requirements.

(5) Machine learning algorithms used in the three-dimensional localisation test are not optimised. It is widely known that CNN has various structures. CNN requires deep optimisations for a specific application to maximise its classification performance. Theoretically, model optimisations and parameter adjustments against indoor localisation will significantly improve the universality and positioning performance of LTPM.

## 7.3 Limitations and future works

Although the two-dimensional and three-dimensional tests are accomplished, some technical problems and empirical defects still exist. In this section, latent defects and feasible solutions are clarified and discussed in terms of the static test. Namely, using

logical analysis and reasoning skills to assess the system deficiencies.

*7.3.1 Factors causing acoustic scattering and signal variations*

Although expected test results are acquired, the scattering of acoustic waves in the three-dimensional space still needs to be valued.

In the two-dimensional localisation test, acoustic scattering is confined within the solid medium; thus, the multipath components of acoustic signals are well-preserved. The well-preserved multipath components endow the received acoustic signal with unique features and ensure superb signal matching results.
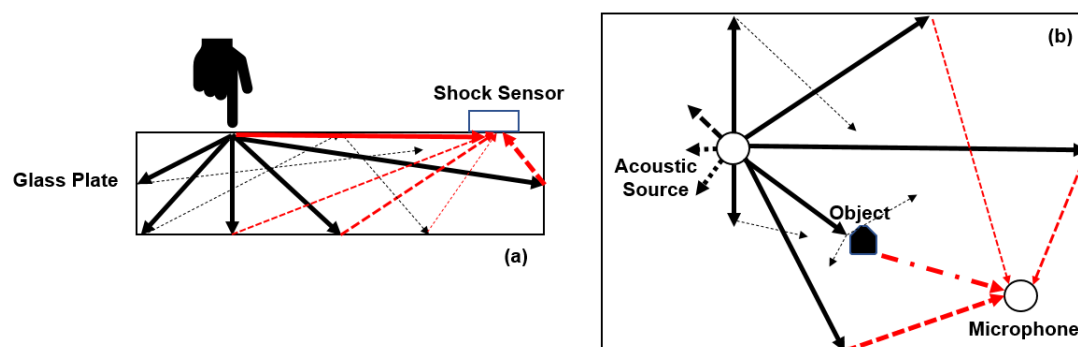


Figure 7.3-1 Illustration of two-dimensional acoustic scattering (a) and three-dimensional scattering (b). From the perspective of acoustic signal reception, acoustic propagation in a three-dimensional space is less complex due to the long propagation distance, significant energy attenuation, acoustic diffraction, spatial interference etc.

However, acoustic waves spread omnidirectionally and attenuate fast in the three-dimensional space. Signal components in the received acoustic signals, especially the multipath signal components, are weakened significantly. The received signal is dominated by direct acoustic components, as shown in Figure 7.3-1.

Possible solutions to this problem include enhancing the reception of acoustic signals with multiple sensors and using active acoustic sources. Enhancing the reception of acoustic signals is the most practical solution since the multi-sensor array receives acoustic waves from multiple locations thus ensuring the reception of multipath signal components. As a cost, the system's cost and complexity will inevitably increase.

Using active acoustic sources will further improve the adaptability of the LTPM-based

localisation system and simplify signal filtering and signal processing. Ultrasonic sources are in top priority as ultrasonic waves are out of the auditory sense of human beings, and the acoustic energy of ultrasonic waves is higher than most audible acoustic waves. A problem caused by integrating active acoustic sources into LTPM is that users may need to hold or wear active acoustic sources. As a cost, the user's mobility will be restricted inevitably.
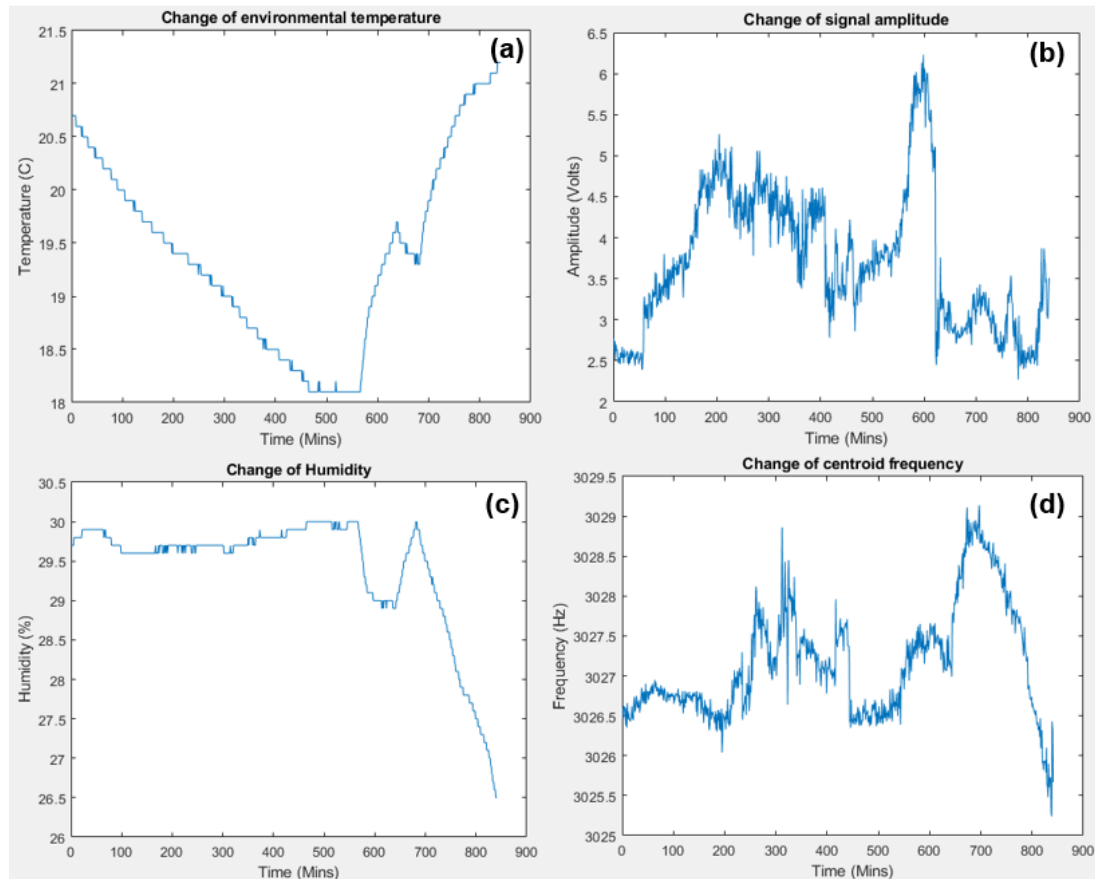


Figure 7.3-2 Change of temperature (a), change of Humidity (c), change of signal amplitude (b) and change of centroid frequency (d). The X-axis represents the sampling time, while the Y-axis varies with analysis objects. The cross-correlation coefficient of temperature to amplitude is 0.5053. The cross-correlation coefficient of temperature to centroid frequency is 0.1897. The cross-correlation coefficient of humidity to amplitude is 0.2619. The cross-correlation coefficient of humidity to centroid frequency is 0.2747. The relations between coefficient and relativity are shown in Table 5.2-1

Similarly, environmental variables cause acoustic signal variations, thereby affecting the positioning performance of the LTPM-based localisation system. Sudden changes in acoustic signals are observed and recorded during the three-dimensional localisation test,

as shown in Figure 7.3-2. The factors leading to signal variations are initially speculated as the environmental temperature, humidity and dewpoint. However, according to the cross-correlation coefficients, environmental temperature and humidity are not correlated with the change of signal amplitudes and frequencies. In addition, the correlation coefficients between temperature, humidity, signal amplitude, and signal frequency are maintained within 0.05-0.50. Therefore, environmental temperature and humidity are not the primary factors causing the acoustic signal variations.

To minimise the impact of external variables, a large number of sample signals are collected. Based on the test results of the RF-based LTPM and the CNN-based LTPM, LTPMs trained with samples from different periods effectively eliminated positioning errors caused by external variables. In future research, it is necessary to identify variables that cause signal variations so that the number of template signals can be reduced.

*7.3.2 The fusion of technologies*

LTPM is a pattern matching-based localisation technology. Other localisation technologies and auxiliary techniques can be integrated into LTPM to achieve better positioning accuracy and enhance the versatility of LTPM.

A single localisation technology always has some drawbacks and limitations. Therefore, modern localisation technologies, represented by the TDOA-based localisation technology, have integrated the pattern matching algorithm and predictive algorithms into the localisation system to enhance the system performance [131].

Similar technology fusion strategies can be applied to LTPM. The technology fusion is divided into the localisation technology fusion and the auxiliary technology fusion. LTPM can be fused with electromagnetic localisation technologies, e.g., the Ultra-Wideband (UWB), to achieve complementary indoor localisation in terms of acoustic sources and electromagnetic sources.

In the fusion of auxiliary technologies, LTPM can be combined with acoustic force feedback technology such as the UltraSound, developed by the University of Bristol [9], or the SoundFORM, developed by MIT [61], to locate users and bring users realistic

tactile sensations.

In this project, different machine learning algorithms are independently integrated into LTPM. But the fusion of different machine learning algorithms is also one of the future research directions. Machine learning has dramatically improved the application potential of the pattern matching-based acoustic localisation technology. Therefore, research on integrating different machine learning algorithms into LTPM is advocated.

*7.3.3 Approaches to reduce the workload of sample collection*

One primary concern about LTPM is the signal collection work. This repetitive work challenges equipment and researchers since massive acoustic samples need to be collected. It is impossible to ensure that template signals are collected from exact pre-defined locations manually. In the three-dimensional localisation test, the acoustic source is transported to designated locations by a robot for signal collection. Therefore, the signal collection includes the automatic signal collection (with robots) and the organic construction of a public database (manual collection).

In the implementation of LTPM, the total number of collected signals is 150,000. Collecting and processing such an amount of signals consume considerable time and massive computational resources. However, results of the reduced scale test indicated that the RF-based LTPM localisation system maintains a classification accuracy of 94% when the number of training signals is reduced to 32,000. Therefore, the number of training signals can be adjusted according to the requirements of the application and empirical schema. On top of that, two solutions that could potentially reduce the sampling workload are discussed:

The sub-region localisation:

The workload of the signal collection work can be reduced by optimising the number and size of the location template. For example, instead of establishing massive location templates in the entire indoor space, a small number of location templates should be established within certain interactive areas. Since the size of each location template in the sub-area is small and the number of in-area location templates is reduced, the required

number of signals for the training of LTPM is reduced accordingly.

Meanwhile, a multistage location template can be formed by connecting location templates from different sub-areas. A benefit of the multistage location template is that machine learning algorithms with different properties can be deployed to maximise their strengths. For example, the CNN-based LTPM determines the coaxial location of the acoustic source first then the RF-based LTPM is applied to determine the precise location of the acoustic source.

The organic construction of acoustic location template database:

Alternatively, the organic construction of a public database is a potential strategy that reduces the workload of signal collection. The organic construction of public databases has already been applied to computer vision [132]. GitHub, Google, Nvidia and universities established open-source databases, and these databases are expanding constantly. Similarly, an acoustic signal database can be established collectively to reduce the workload of a single localisation task.

*7.3.4 Applications of LTPM*

LTPM is an acoustic source localisation technology which can be applied to two-dimensional human-computer interaction and three-dimensional human-computer interaction. It is an appropriate enabling localisation technology for human-computer interaction technologies which have potential localisation demands of individuals in complex indoor environments. LTPM utilises acoustic signals generated by the user to locate the user thus it has the advantages of strong environmental adaptability and low system costs.

A typical application of LTPM is the human-computer interaction inside vehicles since the interior space inside the vehicle is small and enclosed. With the rapid development of autonomous driving technology, speech recognition and speech control are gradually integrated into the autonomous driving system [133]. The speech-based human-computer interaction also has localisation demands. As an enabling localisation technology which also utilises acoustic waves for acoustic source localisation, LTPM can be combined with

the speech-based interaction technology to bring passengers a better user experience by providing the autonomous drive system with implicit information such as the number of passengers, locations of passengers and body motions of passengers.

Another typical application of LTPM is the Virtual Reality (VR) technology. Modern VR technologies provide users with a realistic visual experience but the user experience is poor because users are required to wear and hold transceiver devices to provide computers with location information. The most commonly used localisation technologies in VR are the inertial localisation technology and the TDOA-based electromagnetic localisation technology. However, the proposed LTPM provides the VR technology with a third option. The interaction between users and virtual objects can be realised with speech, knocking, clapping and tapping, thereby bringing users a better immersive interaction experience.

# Reference

[1] Alarifi, A., Al-Salman, A., Alsaleh, M., Alnafessah, A., Al-Hadhrami, S., Al-Ammar, M.A. and Al-Khalifa, H.S. Ultra wideband indoor positioning technologies: Analysis and recent advances. Sensors 2016;16(5):707.

[2] Hostettler, L., Özgür, A., Lemaignan, S., Dillenbourg, P. and Mondada, F. Real-time high-accuracy 2D localization with structured patterns. IEEE International Conference on Robotics and Automation (ICRA) 2016:4536-4543.

[3] Griffin, D.R. Return to the magic well: echolocation behavior of bats and responses of insect prey. Bioscience 2001;51(7):555-556.

[4] Goudie, B. The Importance of Sonar in Modern Warfare. KDC RESOURCE. https://www.kdcresource.com/insights/the-importance-of-sonar-in-modern-warfare/.

[5] D'amico, A. and Pittenger, R. A brief history of active sonar. Space And Naval Warfare Systems Center San Diego CA. 2009.

[6] Akbarov, D., Choi, H., Park, Y., Han, G. and Moon, J. Hybrid pattern matching/TDOA positioning method for CDMA networks. IEEE 4th Workshop on Positioning, Navigation and Communication 2007:199-203.

[7] Pham, D.T., Ji, Z., Yang, M., Wang, Z. and Al-Kutubi, M. A novel human-computer interface based on passive acoustic localisation. In Human-Computer Interaction. Interaction Platforms and Techniques: 12th International Conference, HCI International 2007;12:901-909.

[8] Bahl, P. and Padmanabhan, V.N. RADAR: An in-building RF-based user location and tracking system. IEEE 19th INFOCOM 2000;2:775-784.

[9] Long, B., Seah, S.A., Carter, T. and Subramanian, S. Rendering volumetric haptic shapes in mid-air using ultrasound. ACM Transactions on Graphics (TOG) 2014;201433(6):1-10.

[10] Basiri, A., Lohan, E.S., Moore, T., Winstanley, A., Peltola, P., Hill, C., Amirian, P. and e Silva, P.F. Indoor location based services challenges, requirements and usability of current solutions. Computer Science Review 2017;24:1-12.

[11] Nandini, S., Kiran, V.R., Amith, N. and Manish, B. Under Display Finger Print Scanner for Bezel-Less Displays. 2018.

[12] Kos, T., Markezic, I. and Pokrajic, J. Effects of multipath reception on GPS positioning performance. IEEE International Symposium on Electronics in Marine (ELMAR) 2010:399-402.

[13] Berg, R. E. Acoustics. Encyclopedia Britannica. https://www.brita--nnica.com/science/-acoustics.

[14] Robinson, E.A. History of the wave equation and transforms in engineering. Review of Progress in Quantitative Nondestructive Evaluation 1990:29-35.

[15] Gray, R.M. and Goodman, J.W. Fourier transforms: an introduction for engineers. vol. 322. 2012.

[16] Kromhout, M.J. The Unmusical Ear: Georg Simon Ohm and the Mathematical Analysis of Sound. International Review Devoted to the History of Science and Its Cultural Influences 2020;111(3):471-492.

[17] Fahy, F.J. Foundations of engineering acoustics. 2000.

[18] Helmholtz, H.L. On the Sensations of Tone as a Physiological Basis for the Theory of Music. 2009.

[19] Wolfe, J., Garnier, M. and Smith, J. Vocal tract resonances in speech, singing, and playing musical instruments. Frontiers in Life Science 2009;3(1):6-23.

[20] Rees, T. and Lipton, J. Helmholtz Resonators: Tools for the Analysis of Sound 2009. https://www.whipplemuseum.cam.ac.uk/explore-whipple-collections/acoustics/-herman-von-helmholtz/helmholtz-resonators-tools-analysis-sound.

[21] Foong, S.K. and Van Kolck, U. Poisson random walk for solving wave equations. Progress of theoretical physics 1992;87(2):285-292.

[22] Rayleigh, J.W.S.B. The theory of sound. vol. 2. 1896.

[23] Sarvazyan, A.P., Rudenko, O.V. and Nyborg, W.L. Biomedical applications of radiation force of ultrasound: historical roots and physical basis. Ultrasound in medicine & biology 2010;36(9):1379-1394.

[24] Comer, J.R., Shepard, M.J., Henriksen, P.N. and Ramsier, R.D. Chladni plates revisited. American journal of physics; 2004;72(10):1345-1346.

[25] Jaramillo, A.M. and Steel, C. Architectural acoustics. 2014.

[26] Eyring, C.F. Reverberation time in "dead" rooms. The Journal of the Acoustical Society of America 1930;1(2A):217-241.

[27] Nevard, S.P. and Fourcin, A.J. The phonetics and linguistics anechoic room (UCL). Speech, Hearing & Language 1995;8:1-10.

[28] Ahonen, S. and Eskelinen, P. Performance estimations of mobile terminal location with database correlation in UMTS networks. 2003.

[29] Long, M. Architectural acoustics. 2005.

[30] Stojanovic, M. Wiley Encyclopedia of Telecommunications. 2003.

[31] Gerdes, F., Hofmann, H.G., Jans, W., Künzel, S., Nissen, I. and Dol, H. Measurements and simulations of acoustic propagation loss in the Baltic Sea. 2009.

[32] Llewellyn-Jones, M. The Royal Navy and anti-submarine warfare, 1917-49. 2005.

[33] Buckingham, M.J. The Naval Science of Albert Beaumont Wood, OBE, D. Sc. In Proceedings of Meetings on Acoustics 169 Acoustical Society of America. vol. 23. 2015.

[34] Tardós, J.D., Neira, J., Newman, P.M. and Leonard, J.J. Robust mapping and localization in indoor environments using sonar data. The International Journal of Robotics Research 2002;21(4):311-330.

[35] Kolev, N. ed. Sonar systems. 2011.

[36] Grządziel, A. Results from developments in the use of a scanning sonar to support diving operations from a rescue ship. Remote Sensing 2020;12(4):693.

[37] Shephard, D.A. The contributions of Alexander Graham Bell and Thomas Alva Edison to medicine. Bulletin of the History of Medicine 1977;51(4):610-616.

[38] Kleiner, M. Electroacoustics. 2013.

[39] Manning, P. The Influence of Recording Technologies on the Early Development of Electroacoustic Music. Leonardo Music Journal 2003;13(1):5-10.

[40] Shelton, J. SINUS Messtechnik - Manufacturer of Sound & Vibration Equipment. https://acsoft.co.uk/manufacturer/sinus/.

[41] National Research Council. Hearing loss: Determining eligibility for social security

benefits. 2004.

[42] D'Andrea, P., Veronesi, V., Bicego, M., Melchionda, S., Zelante, L., Di Iorio, E., Bruzzone, R. and Gasparini, P. Hearing loss: frequency and functional studies of the most common connexin26 alleles. Biochemical and biophysical research communications 2002;296(3):685-691.

[43] Masterton, B., Heffner, H. and Ravizza, R. The evolution of human hearing. The Journal of the Acoustical Society of America 1969;45(4):966-985.

[44] Mion, M. and Martini, A. Hermann von Helmholtz and the aesthetics of the acoustics. Hearing, Balance and Communication 2021;19(1):65-69.

[45] Ganapathiraju, A., Hamaker, J.E. and Picone, J. Applications of support vector machines to speech recognition. IEEE transactions on signal processing 2004;52(8):2348-2355.

[46] Xu, Y., Song, Y., Long, Y.H., Zhong, H.B. and Dai, L.R. The description of iFlyTek speech lab system for NIST2009 language recognition evaluation. IEEE 7th International Symposium on Chinese Spoken Language Processing 2010:157-161.

[47] Streitz, N.A., Tandler, P., Müller-Tomfelde, C. and Konomi, S.I. Roomware: Towards the next generation of human-computer interaction based on an integrated design of real and virtual worlds. Human-computer interaction in the new millennium 2001;553:578.

[48] Ishii, H. and Ullmer, B. Tangible bits: towards seamless interfaces between people, bits and atoms. ACM SIGCHI Conference on Human factors in computing systems 1997:234-241.

[49] Hoye, T. and Kozak, J. Touch screens: A pressing technology. 10th Annual Freshman Engineering Sustainability in the New Millennium Conference. vol. 10. 2010.

[50] Lindner, G. Sensors and actuators based on surface acoustic waves propagating along solid-liquid interfaces. Applied Physics 2008;41(12):123002.

[51] Kabulov, B., Tashpulatova, N. and Akbarova, M. Virtual Keyboard and Fingers. IEEE International Conference on Information Science and Communications Technologies (ICISCT) 2019:1-3.

[52] Du, L. An overview of mobile capacitive touch technologies trends. arXiv preprint arXiv:1612.08227. 2016.

[53] Baharav, Z. and Kakarala, R. Capacitive touch sensing: signal and image processing algorithms. In Computational Imaging IX. vol. 7873. 2011.

[54] Lee, C.J., Park, J.K., Piao, C., Seo, H.E., Choi, J. and Chun, J.H. Mutual capacitive sensing touch screen controller for ultrathin display with extended signal passband using negative capacitance. Sensors 2018;18(11):3637.

[55] Barrett, G. and Omote, R. Projected-capacitive touch technology. Information Display 2010;26(3):16-21.

[56] Krithikaa, M. Touch screen technology–a review. International Journal of Trend in Research and Development (IJTRD) 2016;3(1):2394-9333.

[57] Downs, R. Using resistive touch screens for human/machine interface. Analog Applications Journal 2015;3:5-9.

[58] North, K. and D Souza, H. Acoustic pulse recognition enters touch-screen market. Information Display 2006;22(12):22.

[59] Bhalla, M.R. and Bhalla, A.V. Comparative study of various touchscreen technologies. International Journal of Computer Applications 2010;6(8):12-18.

[60] Pham, D.T., Ji, Z., Yang, M., Wang, Z. and Al-Kutubi, M. A novel human-computer interface based on passive acoustic localisation. Interaction Platforms and Techniques: 12th International Conference 2007;12:901-909.

[61] Ishii, H., Lakatos, D., Bonanni, L. and Labrune, J.B. Radical atoms: beyond tangible bits, toward transformable materials. Tangible Interactions 2012;19(1):38-51.

[62] Chou, T.R. and Lo, J.C. Research on the tangible acoustic interface and its applications. In Conference of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE) 2013:913-916.

[63] Kaune, R., Hörst, J. and Koch, W. Accuracy analysis for TDOA localization in sensor networks. IEEE 14th international conference on information fusion 2011:1-8.

[64] Wang, D., Yin, J., Tang, T., Chen, X. and Wu, Z. Quadratic constrained weighted least-squares method for TDOA source localization in the presence of clock synchronization bias: Analysis and solution. Digital Signal Processing 2018;82:237-257.

[65] Drawil, N.M., Amar, H.M. and Basir, O.A. GPS localization accuracy classification: A context-based approach. IEEE Transactions on Intelligent Transportation Systems 2012;14(1):262-273.

[66] Mahmoud, A.A., Ahmad, Z.U., Haas, O.C. and Rajbhandari, S. Precision indoor three-dimensional visible light positioning using receiver diversity and multi-layer perceptron neural network. IET Optoelectronics 2020;14(6):440-446.

[67] Dong, Z., Ye, S., Gao, Y., Fang, G., Zhang, X., Xue, Z. and Zhang, T. Rapid detection methods for asphalt pavement thicknesses and defects by a vehicle-mounted ground penetrating radar (GPR) system. Sensors 2016;16(12):2067.

[68] Gong, J., Gupta, A. and Benko, H. Acustico: surface tap detection and localization using wrist-based acoustic TDOA sensing. The 33rd Annual ACM Symposium on User Interface Software and Technology 2020:406-419.

[69] Deak, G., Curran, K. and Condell, J. A survey of active and passive indoor localisation systems. Computer Communications 2012;35(16):1939-1954.

[70] Fink, M. Time-reversal mirrors. Applied Physics 1993;26(9):1333.

[71] Kuperman, W.A., Hodgkiss, W.S., Song, H.C., Akal, T., Ferla, C. and Jackson, D.R. Phase conjugation in the ocean: Experimental demonstration of an acoustic time-reversal mirror. The Journal of the Acoustical Society of America 1998;103(1):25-40.

[72] Liu, K.R. and Wang, B. Wireless AI: Wireless Sensing, Positioning, IoT, and Communications. Cambridge University Press; 2019.

[73] Fink, M. Time-reversed acoustics. Scientific American 1999;281(5):91-97.

[74] Jones, S.P. Haskell 98 language and libraries: the revised report. Cambridge University Press; 2003.

[75] Apostolico, A. General pattern matching. Springer US; 1997.

[76] Yuan, J., Li, G.H., Zhu, W. and Lai, X. Sonar Target Classification with Sonar Fingerprint. GESTS International Transactions on Computer Science and Engineering 2005;20(1):41-52.

[77] Zhang, X.D., Shi, Y. and Bao, Z. A new feature vector using selected bispectra for signal classification with application in radar target recognition. IEEE Transactions on Signal Processing 2001;49(9):1875-1885.

[78] Al-Kutubi, M. Sensor fusion for tangible acoustic interfaces for human-computer

interaction. PhD thesis. Cardiff University (United Kingdom). 2007.

[79]    Danicki, E. The shock wave-based acoustic sniper localization. Nonlinear analysis: theory, methods & applications 2006;65(5):956-962.

[80]    Li, X., Zhang, D., Lv, Q., Xiong, J., Li, S., Zhang, Y. and Mei, H. IndoTrack: Device-free indoor human tracking with commodity Wi-Fi. ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 2017;1(3):1-22.

[81]    Shumailov, I., Simon, L., Yan, J. and Anderson, R. Hearing your touch: A new acoustic side channel on smartphones. arXiv preprint arXiv:1903.11137. 2019.

[82]    Li, P., Cui, H., Khan, A., Raza, U., Piechocki, R., Doufexi, A. and Farnham, T. Deep transfer learning for WiFi localization. IEEE Radar Conference (RadarConf) 2021:1-5.

[83]    Badiey, M., Katsnelson, B.G., Lin, Y.T. and Lynch, J.F. Acoustic multipath arrivals in the horizontal plane due to approaching nonlinear internal waves. The journal of the acoustical society of America 2011;129(4):141-147.

[84]    O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G.V., Krpalkova, L., Riordan, D. and Walsh, J. Deep learning vs. traditional computer vision. In Advances in Computer Vision: Computer Vision Conference (CVC) 2019;1:128-144.

[85]    Reddy, Y.C.A.P., Viswanath, P. and Reddy, B.E. Semi-supervised learning: A brief review. International Journal of Engineering & Technology 2018;7(1.8):81.

[86]    Bursztein, E., Clarke, E., DeLaune, M., Elifff, D.M., Hsu, N., Olson, L., Shehan, J., Thakur, M., Thomas, K. and Bright, T. Rethinking the detection of child sexual abuse imagery on the internet. In The world wide web conference 2019:2601-2607.

[87]    Chen, Z., Fan, K., Wang, S., Duan, L., Lin, W. and Kot, A.C. Toward intelligent sensing: Intermediate deep feature compression. IEEE Transactions on Image Processing 2019;29:2230-2243.

[88]    Zappone, A., Di Renzo, M. and Debbah, M. Wireless networks design in the era of deep learning: Model-based, AI-based, or both?. IEEE Transactions on Communications 2019;67(10):7331-7376.

[89]    Serra, J., Suris, D., Miron, M. and Karatzoglou, A. Overcoming catastrophic forgetting with hard attention to the task. In International Conference on Machine Learning 2018:4548-4557.

[90]    Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G. and Chen, J. Deep speech 2: End-to-end speech recognition in English and Mandarin. In International conference on machine learning 2016:173-182.

[91]    Saletore, V., Karkada, D., Sripathi, V., Sankaranarayanan, A. and Datta, K. Boosting deep learning training inference performance on intel xeon and intel xeon phi processors. 2018.

[92]    Walker, J., Doersch, C., Gupta, A. and Hebert, M. An uncertain future: Forecasting from static images using variational autoencoders. In Computer Vision–ECCV: 14th European Conference 2016;14:835-851.

[93]    Mirra, N. Putting words in your mouth: The evidentiary impact of emerging voice editing software. 2018.

[94]    Thies, J., Zollhofer, M., Stamminger, M., Theobalt, C. and Nießner, M. Face2face: Real-time face capture and reenactment of RGB videos. IEEE conference on computer vision

and pattern recognition 2016:2387-2395.

[95]  Lighthouse, R. Government Criminals & Adobe Voco. 2017.

[96]  Steyskal, H. Digital beamforming. IEEE 18th European Microwave Conference 1988: 49-57.

[97]  Li, J., White, P.R., Bull, J.M., Leighton, T.G., Roche, B. and Davis, J.W. Passive acoustic localisation of undersea gas seeps using beamforming. International Journal of Greenhouse Gas Control 2021;108:103316.

[98]  Gannot, S., Burshtein, D. and Weinstein, E. Signal enhancement using beamforming and nonstationarity with applications to speech. IEEE Transactions on Signal Processing 2001;49(8):1614-1626.

[99]  Kim, J. and Lee, I. 802.11 WLAN: history and new enabling MIMO techniques for next generation standards. IEEE Communications Magazine 2015;53(3):134-140.

[100]  Benesty, J., Chen, J., Huang, Y. and Dmochowski, J. On microphone-array beamforming from a MIMO acoustic signal processing perspective. IEEE Transactions on Audio, Speech, and Language Processing 2007;15(3):1053-1065.

[101]  Widrow, B. and Luo, F.L. Microphone arrays for hearing aids: An overview. Speech Communication 2003;39(1-2):139-146.

[102]  McCowan, I. Microphone arrays: A tutorial. Queensland University; 2001.

[103]  Boehm, B.W. A spiral model of software development and enhancement. Computer 1988;21(5):61-72.

[104]  Theodoridis, S. and Koutroumbas, K. Pattern recognition. 2006.

[105]  Ji, Z., 2007. Development of tangible acoustic interfaces for human-computer interaction. PhD thesis. Cardiff University (United Kingdom). 2007.

[106]  Wei, H., Hu, C., Chen, S., Xue, Y. and Zhang, Q. Establishing a software defect prediction model via effective dimension reduction. Information Sciences 2019; 477:399-409.

[107]  Karimi, H.A. Advanced location-based technologies and services. Taylor & Francis. 2013.

[108]  Ali, J., Khan, R., Ahmad, N. and Maqsood, I. Random forests and decision trees. International Journal of Computer Science Issues (IJCSI) 2012;9(5):272.

[109]  Strobl, C., Malley, J. and Tutz, G. An introduction to recursive partitioning: rationale, application, and characteristics of classification and regression trees, bagging, and random forests. Psychological methods 2009;14(4):323.

[110]  Maind, S.B. and Wankar, P. Research paper on basic artificial neural network. International Journal on Recent and Innovation Trends in Computing and Communication 2014;2(1):96-100.

[111]  Yang, J., Yu, K. and Huang, T. Supervised translation-invariant sparse coding. IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2010: 3517-3524.

[112]  Hoffmann, J., Navarro, O., Kastner, F., Janßen, B. and Hubner, M. A survey on CNN and RNN implementations. The 7th International Conference on Performance, Safety and Robustness in Complex Systems and Applications. vol. 3. 2017.

[113]  Albawi, S., Mohammed, T.A. and Al-Zawi, S. Understanding of a convolutional neural network. IEEE International conference on engineering and technology 2017:1-6.

[114] Bergstra, J. and Bengio, Y. Random search for hyper-parameter optimization. Journal of machine learning research 2012;13:2.

[115] Caesarendra, W. and Tjahjowidodo, T. A review of feature extraction methods in vibration-based condition monitoring and its application for degradation trend estimation of low-speed slew bearing. Machines 2017;5(4):21.

[116] AVISA, M. Vehicle Based Gunshot Localisation System. 2017.

[117] Katzir, S. The discovery of the piezoelectric effect. The Beginnings of Piezoelectricity: A Study in Mundane Physics 2006:15-64.

[118] Hartwell, L.H., Hopfield, J.J., Leibler, S. and Murray, A.W. From molecular to modular cell biology. Nature 1999;402:47-52.

[119] Yoo, J.C. and Han, T.H. Fast normalized cross-correlation. Circuits, systems and signal processing 2009;28:819-843.

[120] Grey, J.M. and Gordon, J.W. Perceptual effects of spectral modifications on musical timbres. The Journal of the Acoustical Society of America 1978;63(5):1493-1500.

[121] Karna, S.K. and Sahai, R. An overview on Taguchi method. International journal of engineering and mathematical sciences 2012;1(1):1-7.

[122] Krishnaiah, K. and Shahabudeen, P. Applied design of experiments and Taguchi methods. PHI Learning Pvt. Ltd; 2012.

[123] Ramírez-Gallego, S., Mouriño-Talín, H., Martinez-Rego, D., Bolón-Canedo, V., Benítez, J.M., Alonso-Betanzos, A. and Herrera, F. An information theory-based feature selection framework for big data under apache spark. IEEE Transactions on Systems, Man, and Cybernetics: Systems 2017;48(9):1441-1453.

[124] Riedl, M., Müller, A. and Wessel, N. Practical considerations of permutation entropy: A tutorial review. The European Physical Journal Special Topics 2013;222(2):249-262.

[125] Hamdi, S.E., Le Duff, A., Simon, L., Plantier, G., Sourice, A. and Feuilloy, M. Acoustic emission pattern recognition approach based on Hilbert–Huang transform for structural health monitoring in polymer-composite materials. Applied Acoustics 2013;74(5):746-757.

[126] Zbilut, J.P. and Marwan, N. The Wiener–Khinchin theorem and recurrence quantification. Physics Letters 2008;372(44):6622-6626.

[127] Lee, T.H., Ullah, A. and Wang, R. Bootstrap aggregating and random forest. Macroeconomic forecasting in the era of big data: Theory and practice 2020:389-429.

[128] Donges, N., Powers, J., and Pandey, P. Gradient Descent in Machine Learning: A Basic Introduction. https://builtin.com/data-science/gradient-descent.

[129] Bottou, L. Large-scale machine learning with stochastic gradient descent. The 19th International Conference on Computational Statistics 2010:177-186.

[130] Chaudhari, P. and Soatto, S. Stochastic gradient descent performs variational inference and converges to limit cycles for deep networks. IEEE Information Theory and Applications Workshop (ITA) 2018:1-10.

[131] Akbarov, D., Choi, H., Park, Y., Han, G. and Moon, J. Hybrid pattern matching/TDOA positioning method for CDMA networks. IEEE 4th Workshop on Positioning, Navigation and Communication 2007:199-203.

[132] Man, K. and Chahl, J. A Review of Synthetic Image Data and Its Use in Computer Vision. Journal of Imaging 2022;8(11):310.

[133] Jain, J., Lund, A. and Wixon, D. The future of natural user interfaces. Extended Abstracts on Human Factors in Computing Systems 2011:211-214.

[134] Kaemarungsi, K. and Krishnamurthy, P. Modeling of indoor positioning systems based on location fingerprinting. IEEE Infocom 2004;2:1012-1022.

[135] Joseph, V.R. Optimal ratio for data splitting. Statistical Analysis and Data Mining: The ASA Data Science Journal 2022;15(4):531-538.

[136] Breiman, L. Bagging predictors. Machine learning 1996;24:123-140.

# Appendix A Feature definitions and signal transformations

## A.1 Acoustic features for Random Forest-based LTPM

*A.1.1 Acoustic features*

43 signal features are defined and extracted for the training of LTPM. 14 features have been introduced in Chapter 4, Chapter 5 and Chapter 6. The rest of the features are listed as follows:

Assume that a signal sequence received by a sensor is:

$$X(i) = s(i - t_d) + \eta(i) \qquad\qquad A-1$$

whereas $s(i)$ is the source signal. $t_d$ is the time delay. $\eta(i)$ is random Gaussian white noise. And the max value of $i$ represents the last component sampled by the sensor; therefore, $i$ equals the length of the signal sequence.

Features from the time domain:

$$\text{Signal Energy} = \int |X(i)|^2 di \qquad\qquad A-2$$

**Signal Energy**: $X(i)$ is the signal sequence. The energy of a signal equals the integral of the square of its amplitude on the time axis.

$$\text{Mean Value } (\mu) = \frac{1}{N} \sum_{i=0}^{N-1} X(i) \qquad\qquad A-3$$

$$\text{Absolute Mean Value} = \frac{1}{N} \sum_{i=0}^{N-1} |X(i)| \qquad\qquad A-4$$

**The Mean value** and **the Absolute Mean value** represent DC components in a signal.

$$\text{Variance} = \frac{1}{N-1} \sum_{i=0}^{N-1} (X(i) - \mu)^2 \qquad\qquad A-5$$

$$\text{Standard Deviation} = \sqrt{\frac{1}{N-1} \sum_{i=0}^{N-1} (X(i) - \mu)^2} \qquad\qquad A-6$$

where $\mu$ is the mean value of a signal, **variance** represents the power of a signal

deviating from its mean value. **STD** represents the magnitude of the signal deviation.

$$\text{Absolute Summary} = \sum_{i=0}^{N-1} |X(i)| \qquad\qquad A-7$$

$$\text{Peak Value} = MAX(X(i)) \qquad\qquad A-8$$

$$\text{Valley Value} = MIN(X(i)) \qquad\qquad A-9$$

$$\text{Valley to Peak Value} = |MIN(X(i)) - MAX(X(i))| \qquad\qquad A-10$$

$$\text{Median Value} = MEDIAN(X(i)) \qquad\qquad A-11$$

$$\text{RMS} = \sqrt{\frac{1}{N} \sum_{i=0}^{N-1} X(i)^2} \qquad\qquad A-12$$

$$\text{Zero Crossing Rate} = \frac{1}{2} \sum_{i=0}^{N-1} |sgn[X(i)] - sgn[X(i-1)]| \qquad\qquad A-13$$

**Zero Crossing Rate**: $sgn[x]$ is a sign function. $sgn[x] = 1$ when $X(i) > 0$. $sgn[x] = -1$ when $X(i) < 0$. Zero crossing rate refers to the rate of sign change in a signal. This feature is widely used in Electromagnetism, and it is the main feature for current classification.

$$\text{Singular Value Decomposition} = U\Sigma V^T \qquad\qquad A-14$$

**Singular Value Decomposition**: $U$ is a matrix composed of eigenvectors of $X(i)X(i)^T$. $V$ is a matrix composed of eigenvectors of $X(i)^T X(i)$. $\Sigma$ is a singular value matrix with the same dimensions as $X(i)$, its values are the square root of the eigenvalues of $X(i)^T X(i)$.

Features from the frequency domain:

$$\text{CF of Kurtosis Diagram} = MAX\left(\frac{\left(\int_{-\infty}^{\infty} X(i)\omega(i-\tau)e^{-2\pi i f}di\right)^4}{\int_{-\infty}^{\infty} X(i)\omega(i-\tau)e^{-2\pi i f}di^4} - 2\right) \qquad A-15$$

**Center Frequency of Kurtosis Diagram (CFKD)**: The Kurtosis in the frequency domain is calculated with the Short Time Fourier Transform (STFT). $\omega(i)$ is the window function used in STFT. CFKD corresponds to the maximum spectral Kurtosis value in the Kurtosis diagram.

The spectrum of $X(i)$ is calculated with the Fast Fourier Transform (FFT). The following features are all based on the transformed signal: $X_{FFT}(f)$.

$$\text{Max Amplitude} = MAX(X_{FFT}(f)) \qquad\qquad A-16$$

$$\text{Min Amplitude} = MIN(X_{FFT}(f)) \qquad\qquad A-17$$

$$\text{Median Amplitude} = MEDIAN(X_{FFT}(f)) \qquad\qquad A-18$$

$$\text{Mean Amplitude} = \frac{1}{T}\sum_{f=0}^{T-1} X_{FFT}(f) \qquad\qquad A-19$$

$$Valley\ to\ Peak\ Value = |MIN(X_{FFT}(f)) - MAX(X_{FFT}(f))| \qquad\qquad A-20$$

Although the above features appear similar to features in the time domain, the processing objects are the frequency components of the signal.

Features from the power spectrum:

The power spectrum is the Power Spectral Density function (PSD), which is defined as the signal power in the unit frequency band.

The derivation formula of the power spectrum is complex. Still, according to the Wiener–Khinchin theorem, the power spectrum of a signal is equal to the Fourier Transform of the autocorrelation of the same signal. Thus, the power spectrum of $X(i)$ is calculated by performing autocorrelation Fourier Transform on $X(i)$. The following features are all based on the transformed signal: $X_{psd}(f)$.

$$\text{Max Power} = MAX(X_{psd}(f)) \qquad\qquad A-21$$

$$\text{Min Power} = MIN(X_{psd}(f)) \qquad\qquad A-22$$

$$\text{Median Power} = MEDIAN(X_{psd}(f)) \qquad\qquad A-23$$

$$\text{Mean Power} = \frac{1}{T}\sum_{f=0}^{T-1} X_{psd}(f) \qquad\qquad A-24$$

$$\text{Centroid Frequency of Spectrum} = \frac{\int_{0}^{\infty} f^2 X_{psd}(f)df}{\int_{0}^{\infty} X_{psd}(f)df} \qquad\qquad A-25$$

$$\text{Signal to Noise Ratio} = 10\log_{10}\frac{MAX(X_{psd}(f))}{\sum_{i=0}^{N-1}|X_{psd}(f)| - MAX(X_{psd}(f))} \qquad\qquad A-26$$

**Signal to Noise Ratio (spectral)**: It is the ratio of the maximum value of the signal power

to the sum of the signal power (minus the maximum value).

$$\text{Occupied Bandwidth} = f_{max} - f_{min} \qquad\qquad A - 27$$

**Occupied Bandwidth**: It represents the frequency bandwidth occupied by 99% of signal components.

$$\text{Pitch} = \frac{1}{MEAN(diff(f_{peak})/f_s)} \qquad\qquad A - 28$$

$$f_{peak} = findpeaks(ifft(fft(X_{psd}(f)) \times fft(X_{psd}(f)))) \qquad A - 29$$

**Pitch (Fundamental Frequency)**: $fft$ is the Fast Fourier Transform, while $ifft$ is the inverse Fourier Transform. $ifft(fft(X_{psd}(f)) \times fft(X_{psd}(f)))$ is a matrix which contains the autocorrelation and cross-correlation sequences for all combinations of the column vectors in $X_{psd}(f)$.

Entropy features:

$$\text{Spectrum Entropy} = -\sum_{f=1}^{m} X_{psd}(f) \log X_{psd}(f) \qquad\qquad A - 30$$

**Spectrum Entropy**: $X_{psd}(f)$ is the PSD function of a signal. The entropy calculation formula is identical to the signal entropy calculation formula in Chapter 6. The combination of energy distribution and information entropy represents the uncertainty of the signal energy in terms of the power spectrum.

$$\text{Sample Entropy (m, r, N)} = \lim_{N\to}\left\{-\ln\frac{A^m(r)}{A^{m+1}(r)}\right\} \qquad\qquad A - 31$$

**Sample Entropy**: $m$ is the dimension parameter, $r$ is the similarity tolerance for sequence matching, and $N$ is the sequence length. The signal sequence is divided into multiple subsequences according to $m$. $A$ is the ratio of the approximate combination quantity to the total combination quantity. $m$ and $r$ have the same impact on sample entropy. Similar to permutation entropy, sample entropy is also used to measure the complexity of a time series.

*A.1.2 Feature extraction codes (MATLAB)*

```
clear all;clc;
tic;
```

```matlab
signalsper=2000;
points=8;
signals=signalsper*points;
signalset=zeros(signals,100000);
Lenset=zeros(signals,1);
maindir = 'D:\data for localisation 1200mm';
subdir  = dir( maindir );
i1=1;
load('D:\data for localisation 1200mm\errorlog.mat','berrorall');
%
for i = 1 : length( subdir )
    if( isequal( subdir( i ).name, '.' )||...
        isequal( subdir( i ).name, '..')||...
        ~subdir( i ).isdir)
        continue;
    end
    subdirpath = fullfile( maindir, subdir( i ).name, '*.mat' );
    dat = dir( subdirpath );
    % ??
    nameCell = cell(length(dat)-2,1);
    for i2 = 1:length(dat)
        %disp(dat(i2).name);
        nameCell{i2} = dat(i2).name;
    end
    dat1 = sort_nat(nameCell);
    % ?error??
    [m,n]=size(berrorall);
    nameCell1 = cell(m,1);
    for i3 = 1 : m
        nameCell1{i3}=cell2mat(berrorall(i3,1));
    end
    for i4 = 1 : 2300%length(dat)
        i5 = 1:1:m;
        jg=strcmp(dat1(i4),nameCell1(i5));
        if sum(jg)>0
            dat1(i4)=[];
        end
    end

    for j = 1 : 0 + signalsper%length( dat )
        datpath = fullfile( maindir, subdir( i ).name, dat1( j ));
        datpath = cell2mat(datpath);
        a=cell2mat(struct2cell(load( datpath )));
        Len=length(a);
```

```matlab
            Lenset(i1)=Len;
            signalset(i1,1:Len)=a;
            i1=i1+1;

    end
end
%%
clearvars -except signalset Lenset signalsper
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
D=[];
DA=[];
fs=50000;
%%
for i=1:1:m
    d=data(i,1:Lenset(i));
    s0=length(d);
    FFT_Data1 = fft(d);
    power2 = abs(FFT_Data1).^2/fs;   % power spectrum
    power2 = power2(1:floor(s0/2));
    ff2=(1:floor(s0/2))*fs/s0;
    FC=sum(ff2.*power2)/sum(power2);
    clearvars -except signalset Lenset FC d D DA data fs power2 m
signalsper
    peak_value=max(d);%max
    valley_value=min(d);%min
    median_value = median(d);
    mean_value=mean(d);
    abs_sum=sum(abs(d));%sum
    vp=peak_value-valley_value;
    abs_mean_value=mean(abs(d));
    variance=var(d);%variance
    standard_deviation=std(d);%standard deviation
    kurto = kurtosis(d);
    skew = skewness(d);
    root_mean_square = rms(d);
    %factors
    sinfactor = root_mean_square/abs_mean_value;
    crestfactor = vp/root_mean_square;
    impulsefactor = vp/abs_mean_value;
    xr = mean(sqrt(abs(d)))^2;
    Lfactor = vp/xr;
```

190

```matlab
%feature in FD
[~,f_range,fft_sonar] = positiveFFT(d,fs);
%%
f=fft_sonar;
fft_amp=abs(f);
amp_max = max(fft_amp);
amp_min = min(fft_amp);
amp_median = median(fft_amp);
amp_mean = mean(fft_amp);
amp_pk = amp_max - amp_min;
amp_mph = amp_pk * 0.75;
N=length(fft_amp);
amp_pks = [];
amp_pkfs = [];
for i1 = 1:N
    if amp_mph < fft_amp(i1)
        amp_pks = [amp_pks,fft_amp(i1)];
        amp_pkfs = [amp_pkfs,i1 / N * fs/ 2];
    end
end
N1=length(d);
avg_fs = fs*(1:N) / N1;
avg_fft =  fft_amp.^2 / fs;
amp_fc = sum(avg_fs .* avg_fft) / sum(avg_fft);
amp_msf =sum(avg_fs.^2 .* avg_fft) / sum(avg_fft);
amp_rmsf = sqrt(amp_msf);
amp_vf = sum((avg_fs - amp_fc).^2 .* avg_fft) / sum(avg_fft);
amp_rvf = sqrt(amp_vf);
[pbutt,fbutt] = periodogram(d,[],[],fs);
d1=d';
[~,~,~,fc,~,~] = kurtogram(d1,fs);
power_max = max(power2);
power_min = min(power2);
power_median = median(power2);
power_mean = mean(power2);
power_snr = 10*log10(power_max / (sum(power2) - power_max));
power_obw = obw(d,fs);
%cepstrum

cepstrum = real(ifft(log(abs(fft(d)))));
during_time = 1/fs:1/fs:length(d)/fs;
% old features
signal_length=length(d);
zerorate = zero_crossings(d);
```

```matlab
    [dd,dd1]=mapminmax(d);%normalization
    energy_t = sum(abs(d).^2);%energy of each segmented signal
    pitch=self_correlation_pitch(d,fs,20,fs/2);
    svdvalue=svd(d);
    amdf=zeros(N1,1);
    for k = 1:length(d)
        amdf(k) = sum(abs(d(k:end)-d(1:end-k+1)));
    end

    audioIn=d;
    audioBuffered =
buffer(audioIn,round(fs*0.03),round(fs*0.02),'nodelay');
    [p,cf] = poctave(audioBuffered,fs);%Calculate the centroid of the
octave power spectrum over time. Plot the results.
    centroid = spectralCentroid(p,cf);
    t = linspace(0,size(audioIn,1)/fs,size(centroid,1));
    Formantband=formant(d,fs);
    powerentropy=gonglvshang(d);
    entropy=yyshang(dd,10);% instant entropy
    pe=permutationentropy(d');
    %fe=Fuzzy_Entropy(4,0.2*standard_deviation,d,1);
    %???????
    specentropy=spectrum_entropy(d);
    D=[D;signal_length;mean_value;abs_mean_value;
        variance;standard_deviation;abs_sum;
        peak_value;valley_value;vp;
        median_value;
        kurto;skew;root_mean_square;
        sinfactor;crestfactor;impulsefactor;
        Lfactor;zerorate;amp_max;
        amp_min;amp_median;amp_mean;
        amp_pk;amp_fc;amp_msf;
        amp_rmsf;amp_vf;amp_rvf;
        fc;power_max;power_min;
        power_median;power_mean;power_snr;
        power_obw;pitch;svdvalue;
        specentropy;energy_t;FC
        entropy;powerentropy;pe;
        %fe;
        ];
    DA=[DA,D];
    D=[];
End
```

```matlab
function [maxAmp,freq,X]=positiveFFT(x,Fs)
N=length(x); %get the number of points
k=0:N-1; %create a vector from 0 to N-1
T=N/Fs; %get the frequency interval
freq=k/T; %create the frequency range
X=fft(x); % normalize the data
%only want the first half of the FFT, since it is redundant
cutOff = ceil(N/2);
%take only the first half of the spectrum
X = X(1:cutOff);
freq = freq(1:cutOff);
X = abs(X);
%X = mag2db(X);
maxAmp = max(X);


function d=FrequencyCal(x,nw,ni)
n=nw;                               %窗长度
h=ni;                               %窗移量?
s0=length(x);
win=hamming(n);                     %取窗,hamming窗为例
c=1;
ncols=1+fix((s0-n)/h);              %判断信号被分段的次数
d=zeros((1+n/2),ncols);
for b=0:h:(s0-n)
    u=win.*x((b+1):(b+n));
    t=fft(u);
    d(:,c)=t(1:(1+n/2))';
    c=c+1;
end
function varargout = instfreq1(x, varargin)
narginchk(1,7);
nargoutchk(0,2);

opts = parseAndValidateInputs(x,varargin{:});

if strcmp(opts.Method,'tfmoment')

    [instfreq, Time, opts ] =
computeInstantaneousFrequencyTfmoment(opts);
elseif strcmp(opts.Method,'hilbert')

    [instfreq,Time,opts] =
computeInstantaneousFrequencyHilbert(opts);
end
```

```matlab
%Adjust TT variable names
if strcmp(opts.InputType,'TimeTable')
    varlabel = strcat(opts.VarNames,'_instfreq');
    IF = x(1:size(instfreq,1),:);
    IF.Properties.RowTimes = Time;
    IF.Properties.DimensionNames{1} = 'Time';
    for idx = 1:length(opts.VarNames)
        IF.(opts.VarNames{idx}) = instfreq(:,opts.VarColIndex == idx);
    end
    IF.Properties.VariableNames = varlabel;

else
    IF = instfreq;
end
switch nargout
    case 0
        if strcmp(opts.Method,'tfmoment') && opts.NumChannels == 1

displayInstFreqSpectrum(Time,instfreq,opts.Power,opts.Frequency,opts.
Time)
        else
            displayInstFreq(Time,instfreq,opts)
        end
    case 1
        varargout{1} = IF;
    case 2
        varargout{1} = IF;
        varargout{2} = Time;
end


end

function [IF,T,opts] = computeInstantaneousFrequencyTfmoment(opts)
%Compute Instantaneous Frequency using the tfmoment method

for idx = 1:opts.NumChannels

    if ~strcmp(opts.InputType,'Spectrum')
        inputData = opts.Data(:,idx);
        [opts.Power,opts.Frequency,opts.Time] =
pspectrum(inputData,opts.TimeInfo,'spectrogram');
    end
```

```matlab
    IF(:,idx) =
signal.internal.tfmoment.tfsmomentCompute(opts.Power,opts.Frequency,1
,false,opts.FrequencyLim); %#ok<AGROW>

end

if opts.IsSingle
    IF = single(IF);
    if isnumeric(opts.Time)
        opts.Time = single(opts.Time);
    end
end
T = opts.Time;

end


function [IF,T,opts] = computeInstantaneousFrequencyHilbert(opts)
%Compute Instantaneous Frequency using the hilbert method

IF = zeros(size(opts.Data,1)-1,size(opts.Data,2));
for idx = 1:opts.NumChannels

    inputData = opts.Data(:,idx);
    z = hilbert(inputData);
    IF(:,idx) = opts.SamplingFrequency/(2*pi)*diff(unwrap(angle(z)));
end

if numel(opts.TimeInfo)>1
    T = opts.TimeInfo(1:end-1);
    if size(T,1) == 1
        T = T';
    end

else
    if isnumeric(opts.TimeInfo)
        temp = 0:1/opts.SamplingFrequency:(length(opts.Data)-
1)/opts.SamplingFrequency;
        T = temp(1:end-1)';
        if opts.IsSingle
            T = single(T);
        end
    else
        temp = 0:opts.TimeInfo:(length(opts.Data)-1)*opts.TimeInfo;
        T = temp(1:end-1)';
```

```matlab
        end
    end

    %adjusting Time array to be at center of each period
    if isnumeric(T)
        T = T+ ((1/opts.SamplingFrequency)/2);
    else
        T = T+ seconds((1/opts.SamplingFrequency)/2);
    end


    if opts.IsSingle
        IF = single(IF);
    end

end

function displayInstFreq(T,IF,opts)
%Display the instantaneous frequency as a line

[~,freqScale,uf] = ...
signalwavelet.internal.convenienceplot.getFrequencyEngUnits(max(abs(I
F(:))));
IF = IF*freqScale;
freqlbl = [getString(message('signal:instfreq:Frequency')) ' (' uf
')'];

if isnumeric(T)
    [~,timeScale,ut] = ...
signalwavelet.internal.convenienceplot.getTimeEngUnits(max(abs(T)));
    T = T*timeScale;
    timelbl = [getString(message('signal:instfreq:Time')) ' (' ut
')'];
else
    timelbl = getString(message('signal:instfreq:Time'));
end

xlbl = timelbl;
ylbl = freqlbl;
h = newplot;

ifhndl = plot(h,T,IF,'LineWidth',1);

% Disable AxesToolbar
```
196

```matlab
ax = ancestor(ifhndl,'axes');
if iscell(ax)
    cellfun(@(hAx) set(hAx,'Toolbar',[]),ax,'UniformOutput',false);
elseif ~isempty(ax) && ~isempty(ax.Toolbar)
    ax.Toolbar = [];
end


ylabel(ylbl);
xlabel(xlbl);
title(getString(message('signal:instfreq:InstFreqEstimate')));

if opts.NumChannels >1
    if strcmp(opts.InputType,'TimeTable')
        legendNames = [];
        for idx = 1:length(opts.VarNames)
            if sum(opts.VarColIndex==idx) ~= 1
                temp = strcat( opts.VarNames{idx}, '\_instfreq(:,',
num2str((1:sum(opts.VarColIndex==idx))'), ')');
            else
                temp = strcat( opts.VarNames{idx}, '\_instfreq');
            end
            legendNames = [legendNames; cellstr(temp)]; %#ok<AGROW>
        end
    else
        legendNames = cellstr(num2str((1:opts.NumChannels)', 'instfreq
(:,%d)'));
    end
    legend(h,ifhndl,legendNames,'Location','best')
end

ylim(h,[0 (opts.SamplingFrequency/2)*freqScale])

end

function displayInstFreqSpectrum(TF,IF,P,F,T)
%Display the instantaneous frequency as a line over the spectrogram

plotOpts.title =
getString(message('signal:instfreq:InstFreqEstimate'));
plotOpts.legend = getString(message('signal:instfreq:InstFreq'));
plotOpts.isFsNormalized = false;

signalwavelet.internal.convenienceplot.plotTFR(T,F,10*log10(abs(P)+ep
s),TF,IF,plotOpts);
```

197

```matlab
    end

function opts = parseAndValidateInputs(x,varargin)
opts = struct(...
    'Data',[],...
    'SamplingFrequency',[],...
    'Time',[],...
    'TimeInfo',[],...
    'Frequency',[],...
    'Power',[],...
    'DataLength',[],...
    'NumChannels',1,...
    'FrequencyLim',[],...
    'IsTimeTable',istimetable(x),...,
    'VarNames',[],...
    'VarColIndex',[],...
    'InputType','Signal',...
    'Method','tfmoment',...
    'IsSingle',0);


if ~isempty(varargin)
    numValueOnlyInput =
signal.internal.tfmoment.numOfRequiredInput(varargin{:});
else
    numValueOnlyInput = 0;
end

if opts.IsTimeTable
    %Input is Timetable
    opts.InputType = 'TimeTable';

    if numValueOnlyInput > 0
        error(message('signal:instfreq:SampleRateAndTimetableInput'));
    end
    if (height(x) < 2)
        error(message('signal:instfreq:InvalidInputLength'));
    end

    if ~all(varfun(@(x) isa(x,'double'),x,'OutputFormat','uniform'))
&& ~all(varfun(@(x) isa(x,'single'),x,'OutputFormat','uniform'))

error(message('signal:instfreq:InvalidNonHomogeneousDataType'));
```

198

```matlab
        end

    %Validate Data

signal.internal.utilities.validateattributesTimetable(x,{'sorted'},'i
nstfreq','Timetable');
    [Data, ~, TimeInfo] =
signal.internal.utilities.parseTimetable(x);
    opts.VarNames = x.Properties.VariableNames;
    [opts.DataLength, opts.NumChannels] = size(Data);
    validateattributes(Data,
{'single','double'},{'nonempty','finite','2d','nonnan'},'instfreq','T
imetable variables');
    opts.Data = Data;
    opts.IsSingle = isa(Data,'single');
    for idx = 1:length(opts.VarNames)
        opts.VarColIndex = [opts.VarColIndex
idx*ones(1,size(x.(opts.VarNames{idx}),2))];
    end

    %Validate Time
    opts.TimeInfo = TimeInfo;

validateattributes(TimeInfo,{'numeric','duration','datetime'},{'vecto
r','real','nonempty'},'instfreq','Timetable row times ');
    nvpair = {varargin{1:end}};
    if isnumeric(TimeInfo)
        TimeInfo = double(TimeInfo);
    end

    %Compute Fs
    Fs = signal.internal.utilities.computeFs(TimeInfo, 'instfreq');
    if isreal(Data)
        defaultFrequencyLimits = [0,Fs/2];
    else
        defaultFrequencyLimits = [-Fs/2,Fs/2];
    end
    defaultMethod = 'tfmoment';
    opts.SamplingFrequency = Fs;
else
    if isempty(varargin)
        error(message('signal:instfreq:InvalidInputDataTypeVector'));
    end
```

```matlab
    if numValueOnlyInput < 2
        %Input is Signal
        opts.InputType = 'Signal';

        %Validate Data
        validateattributes(x,
{'single','double'},{'nonempty','finite','2d','nonnan'},'instfreq','I
nput signal');
        if isrow(x)
            Data = x(:);
        else
            Data = x;
        end
        [opts.DataLength, opts.NumChannels] = size(Data);

        if (opts.DataLength < 2)
            error(message('signal:instfreq:InvalidInputLength'));
        end
        opts.Data = Data;
        opts.IsSingle = isa(Data,'single');

        %Validate Time
        TimeInfo = varargin{1};

validateattributes(TimeInfo,{'numeric','duration','datetime'},{'vecto
r','real','nonempty'},'instfreq','Time values');

        if isnumeric(TimeInfo)
            TimeInfo = double(TimeInfo);
        end

        len = length(x);
        if (~isscalar(TimeInfo))
            if length(TimeInfo) ~= len
                error(message('signal:instfreq:TimeNotMatchVector'));
            end
        elseif isdatetime(TimeInfo)
            error(message('signal:instfreq:TimeNotMatchVector'));
        end

        opts.TimeInfo = TimeInfo;

        if numValueOnlyInput == numel(varargin)
            nvpair = {};
```

```
        else
            nvpair = {varargin{numValueOnlyInput+1:end}};
        end

        %Compute Fs
        Fs = signal.internal.utilities.computeFs(TimeInfo,
'instfreq');
        if isreal(Data)
            defaultFrequencyLimits = [0,Fs/2];
        else
            defaultFrequencyLimits = [-Fs/2,Fs/2];
        end

        defaultMethod = 'tfmoment';
        opts.SamplingFrequency = Fs;
    else
        %Input is a TFD
        if numValueOnlyInput > 2
            error(message('signal:instfreq:TooManyValueOnlyInputs'));
        end
        opts.InputType = 'Spectrum';

        %Validate Data
        validateattributes(x,
{'single','double'},{'nonnegative','nonempty','finite','real','2d','n
onnan'},'instfreq','TFD');
        opts.Power = x;
        opts.IsSingle = isa(x,'single');

        %Validate Frequency
        F = varargin{1};
        validateattributes(F, {'numeric'},{'vector', 'nonempty',...

'nondecreasing','finite','real','nonnan'},'instfreq','Frequency
values');

        F = double(F);

        len = size(x,1);
        if length(F) ~= len
            error(message('signal:instfreq:FrequencyNotMatch'));
        end

        opts.Frequency = F;
```

```matlab
        %Validate Time
        TimeInfo = varargin{2};

validateattributes(TimeInfo,{'numeric','duration','datetime'},{'vecto
r','nonempty','real'},'instfreq','Time values');

        len = size(x,2);
        if length(TimeInfo) ~= len
            error(message('signal:instfreq:TimeNotMatchTFD'));
        end

        if isnumeric(TimeInfo)
            TimeInfo = double(TimeInfo);
        end

        [t, td]=
signal.internal.tfmoment.parseTime(TimeInfo,len,'instfreq');
        validateattributes(t,
{'single','double'},{'nonnegative','increasing',...
            'finite','real','vector'},'instfreq','time values');
        opts.Time = td;

        if numValueOnlyInput == numel(varargin)
            nvpair = {};
        else
            nvpair = {varargin{numValueOnlyInput+1:end}};
        end
        defaultFrequencyLimits = [F(1),F(end)];
        defaultMethod = 'tfmoment';
    end
end

opts =
parseAndValidateNVPair(opts,defaultMethod,defaultFrequencyLimits,
nvpair);
end

function opts = parseAndValidateNVPair(opts,
defaultMethod,defaultFrequencyLimits,nvpair)
%PARSEANDVALIDATENVPAIR parse and validate the name-value pair for
%INSTFREQ functions.
%
%   This function is for internal use only. It may be removed.
```

```matlab
%
%   Copyright 2017 The MathWorks, Inc.

p=inputParser;

addParameter(p,'FrequencyLimits',defaultFrequencyLimits);
addParameter(p,'Method',defaultMethod);

parse(p,nvpair{:});

%Validate Frequency Limits
FrequencyLim = p.Results.FrequencyLimits;
validateattributes(FrequencyLim, {'single','double'},...
    {'nondecreasing','finite',...

'real','vector','>=',defaultFrequencyLimits(1),'<=',defaultFrequencyL
imits(2),'numel',2},...
    'instfreq','FrequencyLim');
opts.FrequencyLim = FrequencyLim;

%Validate Method
Method = p.Results.Method;
if any(strcmp(Method,{'tfmoment','hilbert'}))
    opts.Method = Method;
    if (strcmp(Method,'hilbert')&& strcmp(opts.InputType,'Spectrum'))
        error(message('signal:instfreq:MethodAndTFDInput'))
    end

    if (strcmp(Method,'hilbert')&&
~any(strcmp(p.UsingDefaults,'FrequencyLimits')))
        error(message('signal:instfreq:MethodAndFreqLim'))
    end

    if (strcmp(Method,'hilbert')&& numel(opts.TimeInfo) >1)
        if isdatetime(opts.TimeInfo)
            temp = seconds(opts.TimeInfo - opts.TimeInfo(1));
        elseif isduration(opts.TimeInfo)
            temp = seconds(opts.TimeInfo);
        else
            temp = opts.TimeInfo;
        end

        err = max(abs(temp(:).'-
```

```
            linspace(temp(1),temp(end),numel(temp)))./max(abs(temp)));
        nonUniformSampling = err > 3*eps(class(err));

        if nonUniformSampling

error(message('signal:instfreq:MethodAndNonUniformSampling'))
        end
    end
else
    error(message('signal:instfreq:InvalidMethod'))
end

end


function Hx=sampleentropy(y,duan)
%²»ÒÔÔ-ÐÅºÅÎ²¼ø¼µÊ±¼äÓòµÄÐÅºÅìØ
%duan:´ýÇóÐÅ¢ìØµÄÐÁÐÒª±»·Ö¡éµÄ¿éÊý
%Hx:yµÄÐÅ¢ìØ
%duan=10;%½«ÐòÁÐ°´duanÊýµ·Ö£¬Èç¹ûduan=10,¾Í½«ÐòÁÐ·ÖÎª10µ·Ý
x_min=min(y);
x_max=max(y);
maxf(1)=abs(x_max-x_min);
maxf(2)=x_min;
duan_t=1.0/duan;
jiange=maxf(1)*duan_t;
% for i=1:10
% pnum(i)=length(find((y_p>=(i-1)*jiange)&(y_p<i*jiange)));
% end
pnum(1)=length(find(y<maxf(2)+jiange));
for i=2:duan-1
    pnum(i)=length(find((y>=maxf(2)+(i-
1)*jiange)&(y<maxf(2)+i*jiange)));
end
pnum(duan)=length(find(y>=maxf(2)+(duan-1)*jiange));
%sum(pnum)
ppnum=pnum/sum(pnum);%Ã¿¶Î³öÏÖµ¸ÅÂÊ
%sum(ppnum)
Hx=0;
for i=1:duan
    if ppnum(i)==0
        Hi=0;
    else
        Hi=-ppnum(i)*log2(ppnum(i));
    end
```

```matlab
    Hx=Hx+Hi;
  end
end

function spl = SPLCal( x,fs,flen )
Length = length(x);
M = flen*fs/1000;
if Length~=M
    error('è¾“å…¥ä¿¡å?·é•¿åº¦ä¸Žæ‰€å®šä¹‹å¸§é•¿ä¸?ç‰ï¼?');
end
pp = 0;
for i = 1:M
    pp = pp + x(i)^2;
end
pa = sqrt(pp/M);
p0 = 2*10^-5;
spl = 20*log10(pa/p0);
end

function d=FrequencyCal(x,nw,ni)
n=nw;
h=ni;
s0=length(x);
win=hamming(n);
c=1;
ncols=1+fix((s0-n)/h);
d=zeros((1+n/2),ncols);
for b=0:h:(s0-n)
    u=win.*x((b+1):(b+n));
    t=fft(u);
    d(:,c)=t(1:(1+n/2))';
    c=c+1;
end

function [F,Bw]=formant(x,fs)
u=filter([1 -.99],1,x);
wlen=length(u);
p=12;
a=lpc(u,p);
U=lpcar2pf(a,255);
freq=(0:256)*fs/512;
df=fs/512;
U_log=10*log10(U);
[Loc,Val]=findpeaks(U);
```

205

```matlab
ll=length(Loc);
for k=1 : ll
    m=Val(k);
    m1=m-1; m2=m+1;
    p=Val(k);
    p1=U(m1); p2=U(m2);
    aa=(p1+p2)/2-p;
    bb=(p2-p1)/2;
    cc=p;
    dm=-bb/2/aa;
    pp=-bb*bb/4/aa+cc;
    m_new=m+dm;
    bf=-sqrt(bb*bb-4*aa*(cc-pp/2))/aa;
    F(k)=(m_new-1)*df;
    Bw(k)=bf*df;
end


function [pe]=permutationentropy(X)
X=X';
 [m,~]=size(X);
eLag = 1;
eDim = 4;
pe=zeros(1,m);
for i=1:m
    [pe(i),~] = pec(X(i,:),eDim,eLag);
end


function basepitch=self_correlation_pitch(d,fs,low,high)
d=d';
nw=length(d);
r2=xcorr(d);
r2 = r2./(d'*d);
rhalf = r2(nw-1:end);
lmin = ceil((low/fs)*nw);
lmax = round((high/fs)*nw);
[~,tloc] = findpeaks(rhalf(lmin:lmax));
basepitch=1./(mean(diff(tloc))/fs);


function [maxAmp,freq,X]=positiveFFT(x,Fs)
N=length(x);
k=0:N-1;
T=N/Fs;
freq=k/T;
X=fft(x);
```

```
cutOff = ceil(N/2);
X = X(1:cutOff);
freq = freq(1:cutOff);
X = abs(X);
maxAmp = max(X);
```

## A.2 Acoustic signal images for CNN-based LTPM

*A.2.1 Signal transformations*

Signal transformation is similar to feature extraction, but the output of signal transformation is the signal image. 17 types of signal images are defined, and three types are introduced in Chapter 6. The remaining four types (11 signal images) are introduced in this section.

Note: the horizontal and vertical axis labels in the training images are erased for the training of CNN-based LTPM.

Amplitude to time:

Assume that a signal sequence is:

$$y \text{ (Sampling Value or Amplitude)} = x(i), i = 1,2, \dots, m \qquad A - 32$$

**Waveform**: $x(i)$ is a one-dimensional array (signal sequence), each value within the array is a discrete signal sampling value. The length of the array is the number of sampling points. The sampling time is calculated by dividing the number of sampling points by the sampling rate.

The waveform of a time-domain signal is acquired by setting the sampling time as the horizontal axis and the sampling value as the vertical axis, as shown in Figure A-1.
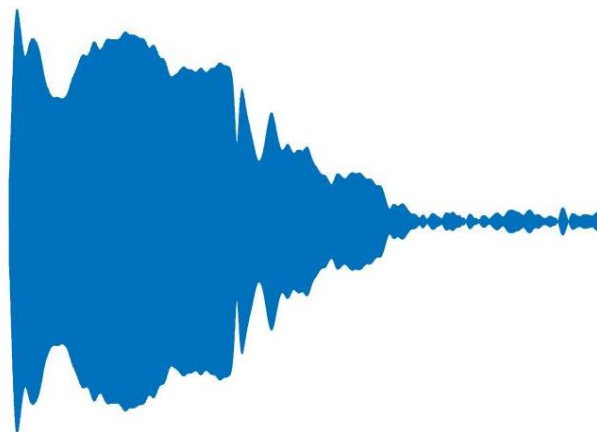


Figure A-2 Signal waveform. The X-axis is the sampling point (time), and the Y-axis is the amplitude of the signal

Amplitude to frequency:

$$Y_{amp} = |fft(x(i))|, i = 1, 2, \ldots, m \qquad (A - 33)$$

$$\text{If } Y_{amp} > 0.01 * \text{MAX}(|fft(x(i))|),$$

$$\{\text{Peaks of Amplitude}\} = Y_{amp}, \{m_{sum}\} = i \qquad (A - 34)$$

**Peaks of Amplitude**: Firstly, the FFT is applied to a time-domain acoustic signal. Then, transformed signal components $Y_{amp}$ that are greater than 1% of the maximum transformed amplitude value are stored in the summary {Peaks of Amplitude}. The vertical axis represents the peak value, and the horizontal axis represents the number of points.
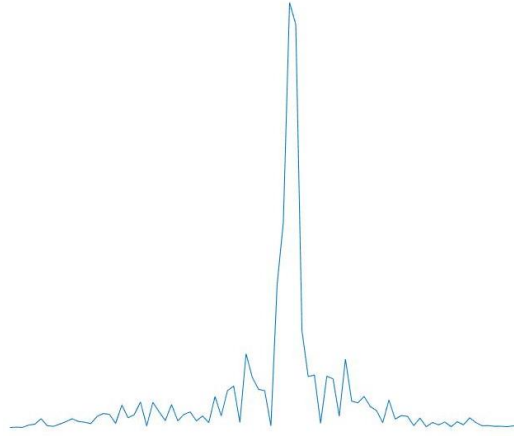


Figure A-2 The summary of peaks of amplitude in the frequency domain. The X-axis is the number of points, and the Y-axis is the amplitude of the signal

$$F_{peak} = \frac{m_{sum}}{N} * \frac{fs}{2} \qquad (A - 35)$$

**Frequencies of Peak Values**: $m_{sum}$ is the summary of signal points corresponding to the peak values extracted with (Eq. A-34), $N$ is the signal length, and $fs$ is the sampling frequency. The processing is similar to the processing of amplitude peaks, but the output is frequency.
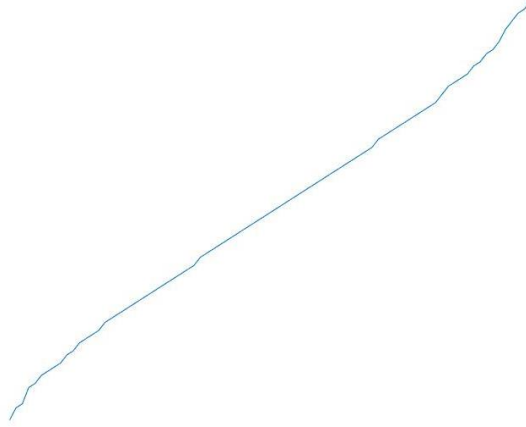
Figure A-3 Corresponding frequencies of peak values. The X-axis is the number of points, and the Y-axis is the frequency

$$\text{Signal Envelop} = \sqrt{hht(x(i))^2 + x(i)^2} \qquad A - 36$$

**Formant**: $hht$ is the Hilbert-Huang Transform (HHT). The Formant is included in the spectrum envelope of the acoustic signal. Therefore, it is necessary to estimate the spectrum envelope of an acoustic signal with the HHT to extract the Formant frequencies. Peak values in the spectrum envelope are the Formant frequencies. The spectrum envelope of the acoustic signal can be calculated with (Eq. A-36). This project estimates the formant band with an interpolated LPC-based channel model.
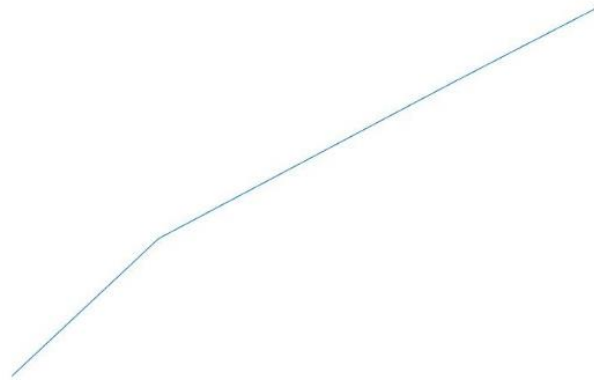


Figure A-4 Formant band. The X-axis is the number of points, and the Y-axis is the frequency

**Spectrum**: The spectrum is calculated with the FFT.
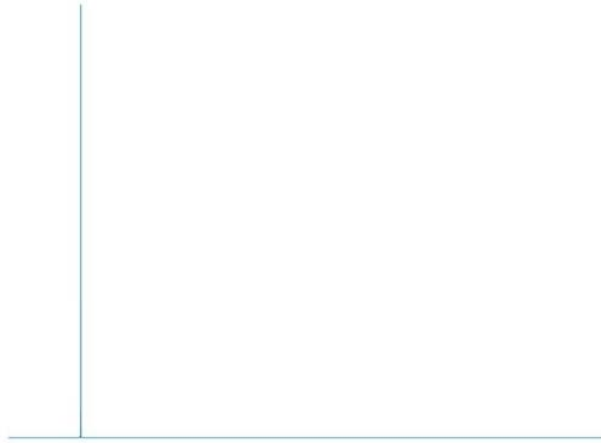
Figure A-5 Signal spectrum. The frequency components of an acoustic signal are displayed clearly. The X-axis is the frequency, and the Y-axis is the amplitude of the signal

$$Cepstrum = real(ifft(\log{(abs(fft(x(i)))))))} \qquad A - 37$$

**Cepstrum**: $real$ stands for the real number calculation. The Cepstrum is symmetrically distributed. The low-frequency components become significant in comparison to the signal spectrum.
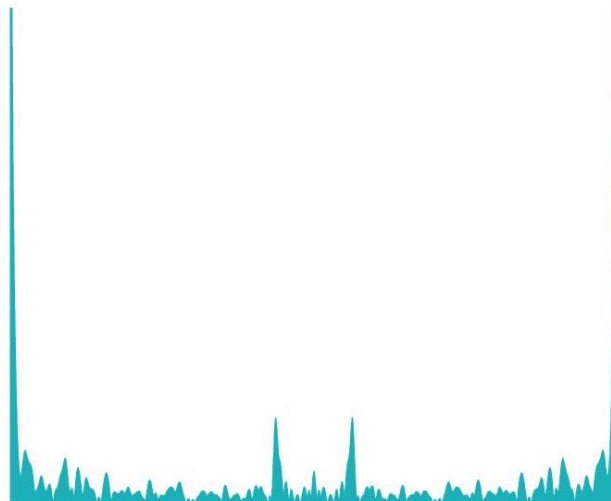


Figure A-6 Signal Cepstrum. The X-axis is the time, and the Y-axis is the amplitude of the signal (logarithmic amplified)

Frequency to time:

$$X_n(e^{j\omega}) = \sum_{m=1}^{m} X(m)w(n-m)e^{j\omega m} \qquad\qquad A-38$$

**Spectrogram**: $m$ is the frame length, $n$ is the number of frames. One acoustic signal is transformed into a multi-dimensional signal after framing: $x(i) \to X(m,n)$. Next, $fft$ is applied to each framed signal as shown in (Eq. A-32). Then the square of the absolute value of each transformed and framed signal is calculated. Finally, the spectrogram is acquired by connecting all the signal frames.
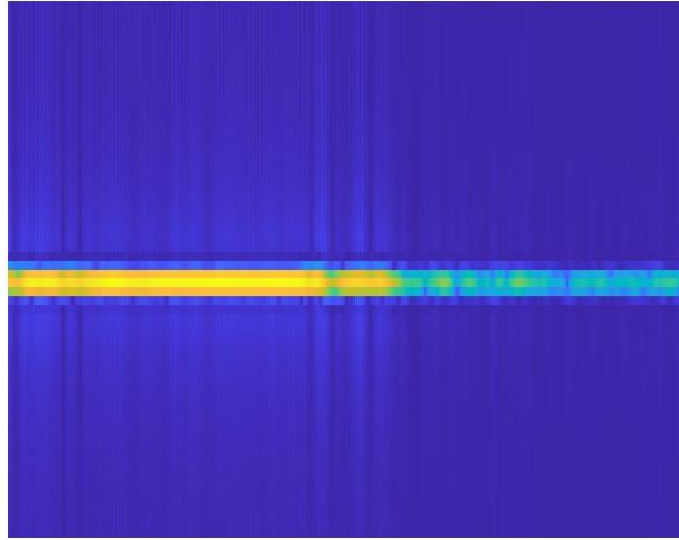


Figure A-7 Signal Spectrogram. The X-axis is the time, and the Y-axis is the frequency

**Change of Centroid Frequency**: An acoustic signal is divided into multiple frames. Then the centroid frequency of each signal frame is calculated according to (Eq. 5-1). The change of centroid frequency is acquired by connecting all signal frames.
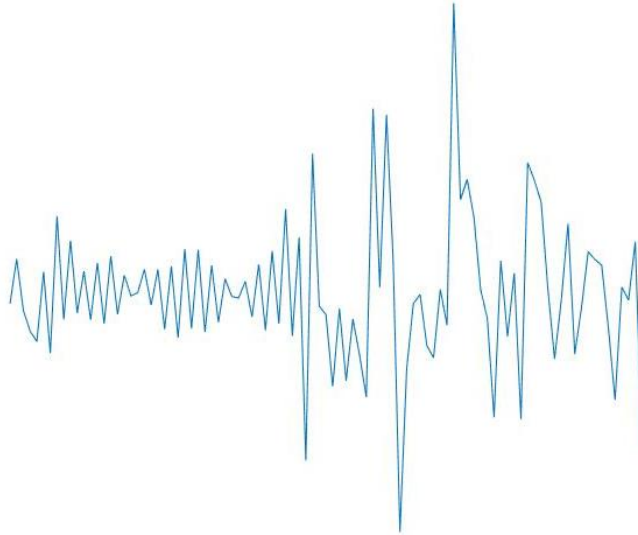
Figure A-8 Change of centroid frequency. The X-axis is the time, and the Y-axis is the frequency

$$f_{inst}(t) = \frac{\int_0^\infty f P(t,f)df}{\int_0^\infty P(t,f)df} \qquad\qquad A-39$$

**Instant Frequency**: $P(t,f)$ is the power spectrum of $x(i)$. The instantaneous frequency of an acoustic signal is a time-varying parameter that relates to the average of the frequencies present in the signal as it evolves.
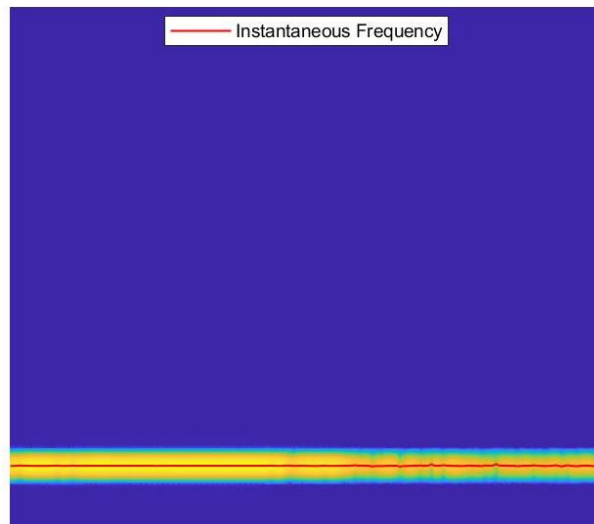


Figure A-9 Instant frequency change of a signal. The X-axis is the time, and the Y-axis is the frequency

Kurtosis to Frequency:

$$S(t,f) = \int_{-\infty}^{+\infty} x(i)w(i-\tau)\,e^{-2\pi fi}di \qquad\qquad A-40$$

$$K(f) = \frac{\langle|S(t,f)|^4\rangle}{\langle|S(t,f)|^2\rangle^2} - 2 \qquad\qquad A-41$$

**Spectral Kurtosis**: w(t) is the window function for the Short Time Fourier Transform (STFT), and $\langle\cdot\rangle$ is the time-average operator. An acoustic signal $x(i)$ is transformed into $S(t,f)$ with STFT first. Next, the Kurtosis in frequency distribution $K(f)$ is calculated based on $S(t,f)$. The spectral kurtosis is used to examine whether a signal is a stationary white noise signal. In practical cases, a white noise signal is always within the confidence interval at all frequency bands.
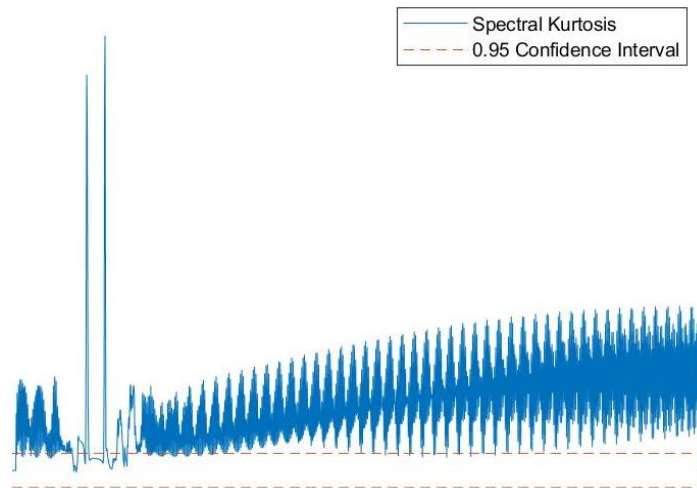


Figure A-10 Spectral kurtosis of an acoustic signal. The X-axis is the frequency, and the Y-axis stands for the value of the spectral Kurtosis

**Kurtosis Diagram**: Spectral Kurtosis uses one window function to calculate $S(t,f)$, while in the Kurtosis diagram, multiple windows are used to calculate $S(t,f)$ of each frame. Compared to spectral Kurtosis, the Kurtosis diagram displays changes in Kurtosis in frequency distribution under different window lengths.

**K**$_{max}$ **= 9.9203 at level 9, Optimal Window Length = 1024,**
**Center Frequency = 9.1553 kHz, Bandwidth = 0.048828 kHz**
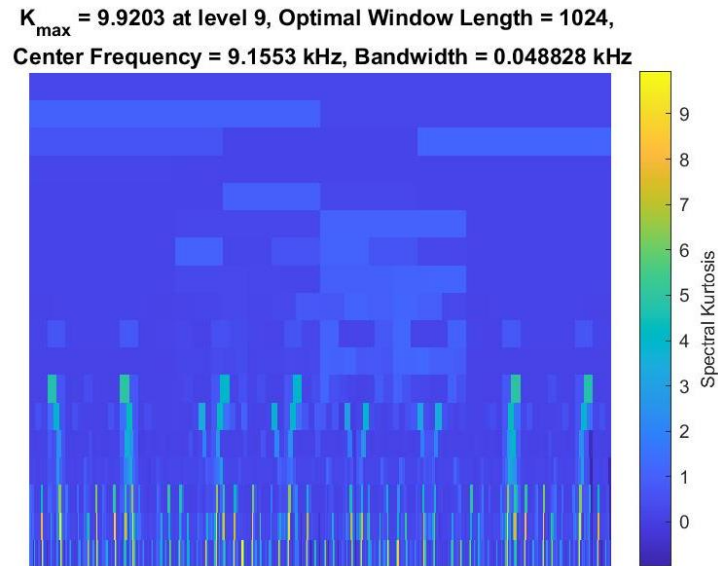
Figure A-11 Kurtosis diagram with a window length of 1024. The X-axis is frequency, and the Y-axis is the window length (points)

*A.2.2 Signal transformation codes (MATLAB)*

```matlab
clear all;clc;
tic;
signalsper=5;
points=8;
signals=signalsper*points;
signalset=zeros(signals,100000);
Lenset=zeros(signals,1);
maindir = 'D:\data for localisation 750mm';
subdir  = dir( maindir );
i1=1;
load('D:\data for localisation 750mm\errorlog.mat','berrorall');
%
for i = 1 : length( subdir )
   if( isequal( subdir( i ).name, '.' )||...
       isequal( subdir( i ).name, '..')||...
       ~subdir( i ).isdir)
       continue;
   end
   subdirpath = fullfile( maindir, subdir( i ).name, '*.mat' );
   dat = dir( subdirpath );
   % ??
   nameCell = cell(length(dat)-2,1);
   for i2 = 1:length(dat)
       %disp(dat(i2).name);
       nameCell{i2} = dat(i2).name;
   end
```

```matlab
    dat1 = sort_nat(nameCell);
    % ?error??
    [m,n]=size(berrorall);
    nameCell1 = cell(m,1);
    for i3 = 1 : m
        nameCell1{i3}=cell2mat(berrorall(i3,1));
    end
    for i4 = 1 : 5%length(dat)
        i5 = 1:1:m;
        jg=strcmp(dat1(i4),nameCell1(i5));
        if sum(jg)>0
            dat1(i4)=[];
        end
    end
    %??????????
    section=0000;
    for j = 1+section : section + signalsper%length( dat )
        datpath = fullfile( maindir, subdir( i ).name, dat1( j ));
        datpath = cell2mat(datpath);
        a=cell2mat(struct2cell(load( datpath )));
        Len=length(a);
        Lenset(i1)=Len;
        signalset(i1,1:Len)=a;
        i1=i1+1;

    end
end
%% ????
clearvars -except signalset Lenset signalsper
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
namedelay=0;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    plot(d);
    axis off
    saveas(gcf,['C:\image750\Time domain\',num2str(i1),'.jpg']);
end
% ????
clearvars -except signalset Lenset signalsper namedelay fs
```

216

```matlab
data=signalset;
[m,~]=size(data);
D=[];
DA=[];
fs1=50000;
for i=1:1:m
    s0=Lenset(i);
    d=data(i,1:s0);
    FFT_Data1 = fft(d);
    power2 = abs(FFT_Data1).^2/fs1;   % power spectrum
    power2 = power2(1:floor(s0/2));
    [B1,~] = mapminmax(power2',0,1);
    ff2=(1:floor(s0/2))*fs1/s0;
    i1=i+namedelay;
    plot(ff2,B1)
    axis off
    saveas(gcf,['C:\image750\Frequency domain\',num2str(i1),'.jpg']);
end
% ???????
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
D=[];
DA=[];
fs1=50000;
for i=1:1:m
    s0=Lenset(i);
    d=data(i,1:s0);
    %[popen,fopen] = periodogram(xs1,[],[],fs1);
    [pbutt,fbutt] = periodogram(d,[],[],fs1);
    i1=i+namedelay;
    plot(fbutt,20*log10(abs(pbutt)))
    axis off
    saveas(gcf,['C:\image750\Power spectrum
density\',num2str(i1),'.jpg']);
end
% ?????
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
D=[];
DA=[];
Fs=50000;
for i=1:1:m
```

217

```matlab
    s0=Lenset(i);
    d=data(i,1:s0);
    signal = d';
    nw=512;ni=nw/4;
    c=1;
    d=FrequencyCal(signal,nw,ni);

    framerate=1:60; %1:257
    d=d(framerate,1:end);

    tt=(0:ni:(length(signal)-nw))/Fs;
    ff=(0:(nw/2))*Fs/nw;
    ff=ff(framerate);
    imagesc(tt,ff,20*log10(c+abs(d)));
    i1=i+namedelay;
    axis off
    saveas(gcf,['C:\image750\Power spectrum\',num2str(i1),'.jpg']);
end
% power spectrum II
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
%namedelay=0;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    s0=length(d);
    FFT_Data1 = fft(d);
    power2 = abs(FFT_Data1).^2/fs;   % power spectrum
    power2 = power2(1:floor(s0/2));
    plot(power2);
    axis off
    saveas(gcf,['C:\image750\power spectrumII\',num2str(i1),'.jpg']);
end

% periodogram
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
```

218

```matlab
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    [pbutt,fbutt] = periodogram(d,[],[],fs);
    plot(fbutt,20*log10(abs(pbutt)),'--');
    axis off
    saveas(gcf,['C:\image750\periodogram\',num2str(i1),'.jpg']);
end
% pkurtosis
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    d1=d';
    pkurtosis(d1,fs);
    axis off
    saveas(gcf,['C:\image750\pkurtosis\',num2str(i1),'.jpg']);
end
% kurtogram
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    d1=d';
    kurtogram(d1,fs);
    axis off
    saveas(gcf,['C:\image750\kurtogram\',num2str(i1),'.jpg']);
end
%% instfreq
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
```

```matlab
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    d1=d';
    instfreq1(d1,fs);
    axis off
    saveas(gcf,['C:\image750\instfreq\',num2str(i1),'.jpg']);
end
% hht
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    d1=d';
    hht(d1,fs);
    axis off
    saveas(gcf,['C:\image750\hht\',num2str(i1),'.jpg']);
end
% ???
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    cepstrum = real(ifft(log(abs(fft(d)))));
    during_time = 1/fs:1/fs:length(d)/fs;
    plot(during_time,cepstrum,'color',[29/255 176/255
184/255]);ylim([0 0.01]);
    axis off
    saveas(gcf,['C:\image750\???\',num2str(i1),'.jpg']);
end
% ??????
```

```matlab
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    audioIn=d;
    audioBuffered =
buffer(audioIn,round(fs*0.03),round(fs*0.02),'nodelay');
    [p,cf] = poctave(audioBuffered,fs);%Calculate the centroid of the
octave power spectrum over time. Plot the results.
    centroid = spectralCentroid(p,cf);
    t = linspace(0,size(audioIn,1)/fs,size(centroid,1));
    plot(t,centroid);
    axis off
    saveas(gcf,['C:\image750\??????\',num2str(i1),'.jpg']);
end
% %% ??
% clearvars -except signalset Lenset signalsper namedelay
% data=signalset;
% [m,n2]=size(data);
% cd 'D:\localisation system + machine learning algorithm\laser
welding analysis'
% fs=50000;
% for i=1:1:m
%     d=data(i,1:Lenset(i));
%     i1=i+namedelay;
%
%     x=d;
%     Length=length(x);
%     framlen = 100;
%     M=fs*framlen/1000;
%     m = mod(Length,M);
%     if m >= M/2 % ??
%         % ??????
%         x = [x,zeros(1,M-m)];
%         % ????????
%         Length = length(x);
%     else   % ?m < M/2???????????
%         % l?Length/M?????
%         l = floor(Length/M);
```

```matlab
%         % ??????
%         x=x';
%         x = x(1:M*l,1);
%         % ????????
%         Length = length(x);
%     end
%     % ??????????
%     N = Length/M;
%     %%----------------------?????-------------
%     s = zeros(1,N);
%     % N??????????spl???
%     spl = zeros(1,N);
%     for k = 1:N
%         % ???k???
%         s =x((k-1)*M + 1:k*M);
%         % ???k????????
%         spl(1,k) = SPLCal(s,fs,framlen);
%     end
%     %%------------??------
%     t = 1:Length;
%     SPL = zeros(1,Length);
%     for r = 1:N
%         SPL(1,(r-1)*M+1:r*M) = spl(r);
%     end
%
%     plot(t/fs,x);
%     axis off
%     saveas(gcf,['D:\image\??\',num2str(i1),'.jpg']);
% end
%
% ???
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;

      x=d;
    Length=length(x);
    framlen = 100;
```

```matlab
    M=fs*framlen/1000;
    m = mod(Length,M);
    if m >= M/2 % ??
        % ??????
        x = [x,zeros(1,M-m)];
        % ????????
        Length = length(x);
    else   % ?m < M/2???????????
        % l?Length/M?????
        l = floor(Length/M);
        % ??????
        x=x';
        x = x(1:M*l,1);
        % ????????
        Length = length(x);
    end
    % ??????????
    N = Length/M;
    %%----------------------?????-------------
    s = zeros(1,N);
    % N???????????spl???
    spl = zeros(1,N);
    for k = 1:N
        % ???k???
        s =x((k-1)*M + 1:k*M);
        % ???k????????
        spl(1,k) = SPLCal(s,fs,framlen);
    end
    %%------------??------
    t = 1:Length;
    SPL = zeros(1,Length);
    for r = 1:N
        SPL(1,(r-1)*M+1:r*M) = spl(r);
    end

    stairs(t/fs,SPL,'r');
    axis off
    saveas(gcf,['C:\image750\???\',num2str(i1),'.jpg']);
end
% amp_pks
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
```

```matlab
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;

    [~,f_range,fft_sonar] = positiveFFT(d,fs);
    f=fft_sonar;
    fft_amp=abs(f);
    amp_pks = [];    %????
    amp_pkfs = [];  %?????????
    amp_max = max(fft_amp);      %?????
    amp_min = min(fft_amp);      %?????
    amp_median = median(fft_amp);   %?????
    amp_mean = mean(fft_amp);       %?????
    amp_pk = amp_max - amp_min;      %????
    amp_mph = amp_pk * 0.75;
    N=length(fft_amp);
    for i2 = 1:N
        if amp_mph < fft_amp(i2)
            amp_pks = [amp_pks,fft_amp(i2)];
            amp_pkfs = [amp_pkfs,i2 / N * fs/ 2];
        end
    end

    plot(amp_pks);
    axis off
    saveas(gcf,['C:\image750\amp_pks\',num2str(i1),'.jpg']);
end
% amp_pkfs
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;

    [~,f_range,fft_sonar] = positiveFFT(d,fs);
    f=fft_sonar;
    fft_amp=abs(f);
```

```matlab
    amp_pks = [];    %????
    amp_pkfs = [];  %?????????
    amp_max = max(fft_amp);     %?????
    amp_min = min(fft_amp);     %?????
    amp_median = median(fft_amp);   %?????
    amp_mean = mean(fft_amp);       %?????
    amp_pk = amp_max - amp_min;     %????
    amp_mph = amp_pk * 0.75;
    N=length(fft_amp);
    for i2 = 1:N
        if amp_mph < fft_amp(i2)
            amp_pks = [amp_pks,fft_amp(i2)];
            amp_pkfs = [amp_pkfs,i2 / N * fs/ 2];
        end
    end

    plot(amp_pkfs);
    axis off
    saveas(gcf,['C:\image750\amp_pkfs\',num2str(i1),'.jpg']);
end
%% amdf
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m %1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;

    N1=length(d);
    amdf=zeros(N1,1);
    for k = 1:length(d)
        amdf(k) = sum(abs(d(k:end)-d(1:end-k+1)));%????????????
    end

    plot(amdf);
    axis off
    saveas(gcf,['C:\image750\amdf\',num2str(i1),'.jpg']);
end

for i=1:1:m %1:1:m
    d=data(i,1:Lenset(i));
```

```matlab
        i1=i+namedelay;

        N1=length(d);
        amdf=zeros(N1,1);
        for k = 1:length(d)
            amdf(k) = sum(abs(d(k:end)-d(1:end-k+1)));%?????????????
        end

        plot(amdf);
        axis off
        saveas(gcf,['C:\image750\amdf\',num2str(i1),'.jpg']);
end
% formantband
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;

    Formantband=formant(d,fs);

    plot(Formantband);
    axis off
    saveas(gcf,['C:\image750\formantband\',num2str(i1),'.jpg']);
end
%%
clear all;clc;
tic;
signalsper=5;
points=8;
signals=signalsper*points;
signalset=zeros(signals,100000);
Lenset=zeros(signals,1);
maindir = 'D:\data for localisation 900mm';
subdir  = dir( maindir );
i1=1;
load('D:\data for localisation 900mm\errorlog.mat','berrorall');
%
for i = 1 : length( subdir )
    if( isequal( subdir( i ).name, '.' )||...
```
226

```matlab
        isequal( subdir( i ).name, '..')||...
        ~subdir( i ).isdir)
        continue;
    end
    subdirpath = fullfile( maindir, subdir( i ).name, '*.mat' );
    dat = dir( subdirpath );
    % ??
    nameCell = cell(length(dat)-2,1);
    for i2 = 1:length(dat)
        %disp(dat(i2).name);
        nameCell{i2} = dat(i2).name;
    end
    dat1 = sort_nat(nameCell);
    % ?error??
    [m,n]=size(berrorall);
    nameCell1 = cell(m,1);
    for i3 = 1 : m
        nameCell1{i3}=cell2mat(berrorall(i3,1));
    end
    for i4 = 1 : 5%length(dat)
        i5 = 1:1:m;
        jg=strcmp(dat1(i4),nameCell1(i5));
        if sum(jg)>0
            dat1(i4)=[];
        end
    end
    %??????????
    section=0000;
    for j = 1+section : section + signalsper%length( dat )
        datpath = fullfile( maindir, subdir( i ).name, dat1( j ));
        datpath = cell2mat(datpath);
        a=cell2mat(struct2cell(load( datpath )));
        Len=length(a);
        Lenset(i1)=Len;
        signalset(i1,1:Len)=a;
        i1=i1+1;

    end
end
%% ????
clearvars -except signalset Lenset signalsper
data=signalset;
[m,n2]=size(data);
```

```matlab
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
namedelay=0;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    plot(d);
    axis off
    saveas(gcf,['C:\image900\Time domain\',num2str(i1),'.jpg']);
end
% ????
clearvars -except signalset Lenset signalsper namedelay fs
data=signalset;
[m,~]=size(data);
D=[];
DA=[];
fs1=50000;
for i=1:1:m
    s0=Lenset(i);
    d=data(i,1:s0);
    FFT_Data1 = fft(d);
    power2 = abs(FFT_Data1).^2/fs1;   % power spectrum
    power2 = power2(1:floor(s0/2));
    [B1,~] = mapminmax(power2',0,1);
    ff2=(1:floor(s0/2))*fs1/s0;
    i1=i+namedelay;
    plot(ff2,B1)
    axis off
    saveas(gcf,['C:\image900\Frequency domain\',num2str(i1),'.jpg']);
end
% ???????
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
D=[];
DA=[];
fs1=50000;
for i=1:1:m
    s0=Lenset(i);
    d=data(i,1:s0);
    %[popen,fopen] = periodogram(xs1,[],[],fs1);
    [pbutt,fbutt] = periodogram(d,[],[],fs1);
    i1=i+namedelay;
```

228

```matlab
    plot(fbutt,20*log10(abs(pbutt)))
    axis off
    saveas(gcf,['C:\image900\Power spectrum
density\',num2str(i1),'.jpg']);
end
% ?????
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
D=[];
DA=[];
Fs=50000;
for i=1:1:m
    s0=Lenset(i);
    d=data(i,1:s0);
    signal = d';
    nw=512;ni=nw/4;
    c=1;
    d=FrequencyCal(signal,nw,ni);

    framerate=1:60; %1:257
    d=d(framerate,1:end);

    tt=(0:ni:(length(signal)-nw))/Fs;
    ff=(0:(nw/2))*Fs/nw;
    ff=ff(framerate);
    imagesc(tt,ff,20*log10(c+abs(d)));
    i1=i+namedelay;
    axis off
    saveas(gcf,['C:\image900\Power spectrum\',num2str(i1),'.jpg']);
end
% power spectrum II
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
%namedelay=0;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    s0=length(d);
    FFT_Data1 = fft(d);
```

229

```matlab
    power2 = abs(FFT_Data1).^2/fs;   % power spectrum
    power2 = power2(1:floor(s0/2));
    plot(power2);
    axis off
    saveas(gcf,['C:\image900\power spectrumII\',num2str(i1),'.jpg']);
end


% periodogram
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    [pbutt,fbutt] = periodogram(d,[],[],fs);
    plot(fbutt,20*log10(abs(pbutt)),'--');
    axis off
    saveas(gcf,['C:\image900\periodogram\',num2str(i1),'.jpg']);
end
% pkurtosis
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    d1=d';
    pkurtosis(d1,fs);
    axis off
    saveas(gcf,['C:\image900\pkurtosis\',num2str(i1),'.jpg']);
end
% kurtogram
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
```

```matlab
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    d1=d';
    kurtogram(d1,fs);
    axis off
    saveas(gcf,['C:\image900\kurtogram\',num2str(i1),'.jpg']);
end
%% instfreq
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    d1=d';
    instfreq1(d1,fs);
    axis off
    saveas(gcf,['C:\image900\instfreq\',num2str(i1),'.jpg']);
end
% hht
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    d1=d';
    hht(d1,fs);
    axis off
    saveas(gcf,['C:\image900\hht\',num2str(i1),'.jpg']);
end
% ???
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
```

```matlab
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    cepstrum = real(ifft(log(abs(fft(d)))));
    during_time = 1/fs:1/fs:length(d)/fs;
    plot(during_time,cepstrum,'color',[29/255 176/255
184/255]);ylim([0 0.01]);
    axis off
    saveas(gcf,['C:\image900\???\',num2str(i1),'.jpg']);
end
% ??????
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    audioIn=d;
    audioBuffered =
buffer(audioIn,round(fs*0.03),round(fs*0.02),'nodelay');
    [p,cf] = poctave(audioBuffered,fs);%Calculate the centroid of the
octave power spectrum over time. Plot the results.
    centroid = spectralCentroid(p,cf);
    t = linspace(0,size(audioIn,1)/fs,size(centroid,1));
    plot(t,centroid);
    axis off
    saveas(gcf,['C:\image900\??????\',num2str(i1),'.jpg']);
end
% %% ??
% clearvars -except signalset Lenset signalsper namedelay
% data=signalset;
% [m,n2]=size(data);
% cd 'D:\localisation system + machine learning algorithm\laser
welding analysis'
% fs=50000;
% for i=1:1:m
%     d=data(i,1:Lenset(i));
%     i1=i+namedelay;
%
%     x=d;
```

```matlab
%     Length=length(x);
%     framlen = 100;
%     M=fs*framlen/1000;
%     m = mod(Length,M);
%     if m >= M/2 % ??
%         % ??????
%         x = [x,zeros(1,M-m)];
%         % ????????
%         Length = length(x);
%     else   % ?m < M/2???????????
%         % l?Length/M?????
%         l = floor(Length/M);
%         % ??????
%         x=x';
%         x = x(1:M*l,1);
%         % ????????
%         Length = length(x);
%     end
%     % ??????????
%     N = Length/M;
%     %%----------------------?????-------------
%     s = zeros(1,N);
%     % N???????????spl???
%     spl = zeros(1,N);
%     for k = 1:N
%         % ???k???
%         s =x((k-1)*M + 1:k*M);
%         % ???k????????
%         spl(1,k) = SPLCal(s,fs,framlen);
%     end
%     %%------------??------
%     t = 1:Length;
%     SPL = zeros(1,Length);
%     for r = 1:N
%         SPL(1,(r-1)*M+1:r*M) = spl(r);
%     end
%
%     plot(t/fs,x);
%     axis off
%     saveas(gcf,['D:\image\??\',num2str(i1),'.jpg']);
% end
%
% ???
clearvars -except signalset Lenset signalsper namedelay
```

```matlab
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;

      x=d;
    Length=length(x);
    framlen = 100;
    M=fs*framlen/1000;
    m = mod(Length,M);
    if m >= M/2 % ??
        % ??????
        x = [x,zeros(1,M-m)];
        % ????????
        Length = length(x);
    else   % ?m < M/2???????????
        % l?Length/M?????
        l = floor(Length/M);
        % ??????
        x=x';
        x = x(1:M*l,1);
        % ????????
        Length = length(x);
    end
    % ??????????
    N = Length/M;
    %%----------------------?????-------------
    s = zeros(1,N);
    % N??????????spl???
    spl = zeros(1,N);
    for k = 1:N
        % ???k???
        s =x((k-1)*M + 1:k*M);
        % ???k????????
        spl(1,k) = SPLCal(s,fs,framlen);
    end
    %%-----------??------
    t = 1:Length;
    SPL = zeros(1,Length);
    for r = 1:N
```

```matlab
        SPL(1,(r-1)*M+1:r*M) = spl(r);
    end

    stairs(t/fs,SPL,'r');
    axis off
    saveas(gcf,['C:\image900\???\',num2str(i1),'.jpg']);
end
% amp_pks
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;

    [~,f_range,fft_sonar] = positiveFFT(d,fs);
    f=fft_sonar;
    fft_amp=abs(f);
    amp_pks = [];   %????
    amp_pkfs = [];  %?????????
    amp_max = max(fft_amp);     %?????
    amp_min = min(fft_amp);     %?????
    amp_median = median(fft_amp);   %?????
    amp_mean = mean(fft_amp);       %?????
    amp_pk = amp_max - amp_min;     %????
    amp_mph = amp_pk * 0.75;
    N=length(fft_amp);
    for i2 = 1:N
        if amp_mph < fft_amp(i2)
            amp_pks = [amp_pks,fft_amp(i2)];
            amp_pkfs = [amp_pkfs,i2 / N * fs/ 2];
        end
    end

    plot(amp_pks);
    axis off
    saveas(gcf,['C:\image900\amp_pks\',num2str(i1),'.jpg']);
end
% amp_pkfs
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
```

235

```matlab
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;

    [~,f_range,fft_sonar] = positiveFFT(d,fs);
    f=fft_sonar;
    fft_amp=abs(f);
    amp_pks = [];   %????
    amp_pkfs = [];  %?????????
    amp_max = max(fft_amp);     %?????
    amp_min = min(fft_amp);     %?????
    amp_median = median(fft_amp);   %?????
    amp_mean = mean(fft_amp);       %?????
    amp_pk = amp_max - amp_min;     %????
    amp_mph = amp_pk * 0.75;
    N=length(fft_amp);
    for i2 = 1:N
        if amp_mph < fft_amp(i2)
            amp_pks = [amp_pks,fft_amp(i2)];
            amp_pkfs = [amp_pkfs,i2 / N * fs/ 2];
        end
    end

    plot(amp_pkfs);
    axis off
    saveas(gcf,['C:\image900\amp_pkfs\',num2str(i1),'.jpg']);
end
%% amdf
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m %1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;

    N1=length(d);
    amdf=zeros(N1,1);
```

```matlab
    for k = 1:length(d)
        amdf(k) = sum(abs(d(k:end)-d(1:end-k+1)));%????????????
    end

    plot(amdf);
    axis off
    saveas(gcf,['C:\image900\amdf\',num2str(i1),'.jpg']);
end

for i=1:1:m %1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;

    N1=length(d);
    amdf=zeros(N1,1);
    for k = 1:length(d)
        amdf(k) = sum(abs(d(k:end)-d(1:end-k+1)));%????????????
    end

    plot(amdf);
    axis off
    saveas(gcf,['C:\image900\amdf\',num2str(i1),'.jpg']);
end
% formantband
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;

    Formantband=formant(d,fs);

    plot(Formantband);
    axis off
    saveas(gcf,['C:\image900\formantband\',num2str(i1),'.jpg']);
end
%%
clear all;clc;
tic;
signalsper=5;
```

237

```matlab
points=8;
signals=signalsper*points;
signalset=zeros(signals,100000);
Lenset=zeros(signals,1);
maindir = 'D:\data for localisation 1050mm';
subdir  = dir( maindir );
i1=1;
load('D:\data for localisation 1050mm\errorlog.mat','berrorall');
%
for i = 1 : length( subdir )
    if( isequal( subdir( i ).name, '.' )||...
        isequal( subdir( i ).name, '..')||...
        ~subdir( i ).isdir)
        continue;
    end
    subdirpath = fullfile( maindir, subdir( i ).name, '*.mat' );
    dat = dir( subdirpath );
    % ??
    nameCell = cell(length(dat)-2,1);
    for i2 = 1:length(dat)
        %disp(dat(i2).name);
        nameCell{i2} = dat(i2).name;
    end
    dat1 = sort_nat(nameCell);
    % ?error??
    [m,n]=size(berrorall);
    nameCell1 = cell(m,1);
    for i3 = 1 : m
        nameCell1{i3}=cell2mat(berrorall(i3,1));
    end
    for i4 = 1 : 5%length(dat)
        i5 = 1:1:m;
        jg=strcmp(dat1(i4),nameCell1(i5));
        if sum(jg)>0
            dat1(i4)=[];
        end
    end
    %??????????
    section=0000;
    for j = 1+section : section + signalsper%length( dat )
        datpath = fullfile( maindir, subdir( i ).name, dat1( j ));
        datpath = cell2mat(datpath);
        a=cell2mat(struct2cell(load( datpath )));
        Len=length(a);
```

238

```matlab
        Lenset(i1)=Len;
        signalset(i1,1:Len)=a;
        i1=i1+1;


    end
end
%% ????
clearvars -except signalset Lenset signalsper
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
namedelay=0;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    plot(d);
    axis off
    saveas(gcf,['C:\image1050\Time domain\',num2str(i1),'.jpg']);
end
% ????
clearvars -except signalset Lenset signalsper namedelay fs
data=signalset;
[m,~]=size(data);
D=[];
DA=[];
fs1=50000;
for i=1:1:m
    s0=Lenset(i);
    d=data(i,1:s0);
    FFT_Data1 = fft(d);
    power2 = abs(FFT_Data1).^2/fs1;   % power spectrum
    power2 = power2(1:floor(s0/2));
    [B1,~] = mapminmax(power2',0,1);
    ff2=(1:floor(s0/2))*fs1/s0;
    i1=i+namedelay;
    plot(ff2,B1)
    axis off
    saveas(gcf,['C:\image1050\Frequency
domain\',num2str(i1),'.jpg']);
end
% ???????
clearvars -except signalset Lenset signalsper namedelay
```

239

```matlab
data=signalset;
[m,n2]=size(data);
D=[];
DA=[];
fs1=50000;
for i=1:1:m
    s0=Lenset(i);
    d=data(i,1:s0);
    %[popen,fopen] = periodogram(xs1,[],[],fs1);
    [pbutt,fbutt] = periodogram(d,[],[],fs1);
    i1=i+namedelay;
    plot(fbutt,20*log10(abs(pbutt)))
    axis off
    saveas(gcf,['C:\image1050\Power spectrum
density\',num2str(i1),'.jpg']);
end
% ?????
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
D=[];
DA=[];
Fs=50000;
for i=1:1:m
    s0=Lenset(i);
    d=data(i,1:s0);
    signal = d';
    nw=512;ni=nw/4;
    c=1;
    d=FrequencyCal(signal,nw,ni);

    framerate=1:60; %1:257
    d=d(framerate,1:end);

    tt=(0:ni:(length(signal)-nw))/Fs;
    ff=(0:(nw/2))*Fs/nw;
    ff=ff(framerate);
    imagesc(tt,ff,20*log10(c+abs(d)));
    i1=i+namedelay;
    axis off
    saveas(gcf,['C:\image1050\Power spectrum\',num2str(i1),'.jpg']);
end
% power spectrum II
clearvars -except signalset Lenset signalsper namedelay
```

```matlab
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
%namedelay=0;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    s0=length(d);
    FFT_Data1 = fft(d);
    power2 = abs(FFT_Data1).^2/fs;   % power spectrum
    power2 = power2(1:floor(s0/2));
    plot(power2);
    axis off
    saveas(gcf,['C:\image1050\power
spectrumII\',num2str(i1),'.jpg']);
end

% periodogram
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    [pbutt,fbutt] = periodogram(d,[],[],fs);
    plot(fbutt,20*log10(abs(pbutt)),'--');
    axis off
    saveas(gcf,['C:\image1050\periodogram\',num2str(i1),'.jpg']);
end
% pkurtosis
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
```

```matlab
    d1=d';
    pkurtosis(d1,fs);
    axis off
    saveas(gcf,['C:\image1050\pkurtosis\',num2str(i1),'.jpg']);
end
% kurtogram
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    d1=d';
    kurtogram(d1,fs);
    axis off
    saveas(gcf,['C:\image1050\kurtogram\',num2str(i1),'.jpg']);
end
%% instfreq
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    d1=d';
    instfreq1(d1,fs);
    axis off
    saveas(gcf,['C:\image1050\instfreq\',num2str(i1),'.jpg']);
end
% hht
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
```

```matlab
    i1=i+namedelay;
    d1=d';
    hht(d1,fs);
    axis off
    saveas(gcf,['C:\image1050\hht\',num2str(i1),'.jpg']);
end
% ???
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    cepstrum = real(ifft(log(abs(fft(d)))));
    during_time = 1/fs:1/fs:length(d)/fs;
    plot(during_time,cepstrum,'color',[29/255 176/255
184/255]);ylim([0 0.01]);
    axis off
    saveas(gcf,['C:\image1050\???\',num2str(i1),'.jpg']);
end
% ??????
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    audioIn=d;
    audioBuffered =
buffer(audioIn,round(fs*0.03),round(fs*0.02),'nodelay');
    [p,cf] = poctave(audioBuffered,fs);%Calculate the centroid of the
octave power spectrum over time. Plot the results.
    centroid = spectralCentroid(p,cf);
    t = linspace(0,size(audioIn,1)/fs,size(centroid,1));
    plot(t,centroid);
    axis off
    saveas(gcf,['C:\image1050\??????\',num2str(i1),'.jpg']);
end
```

```matlab
% %% ??
% clearvars -except signalset Lenset signalsper namedelay
% data=signalset;
% [m,n2]=size(data);
% cd 'D:\localisation system + machine learning algorithm\laser
welding analysis'
% fs=50000;
% for i=1:1:m
%     d=data(i,1:Lenset(i));
%     i1=i+namedelay;
%
%     x=d;
%     Length=length(x);
%     framlen = 100;
%     M=fs*framlen/1000;
%     m = mod(Length,M);
%     if m >= M/2 % ??
%         % ??????
%         x = [x,zeros(1,M-m)];
%         % ????????
%         Length = length(x);
%     else  % ?m < M/2???????????
%         % l?Length/M?????
%         l = floor(Length/M);
%         % ??????
%         x=x';
%         x = x(1:M*l,1);
%         % ????????
%         Length = length(x);
%     end
%     % ??????????
%     N = Length/M;
%     %%----------------------?????-------------
%     s = zeros(1,N);
%     % N??????????spl???
%     spl = zeros(1,N);
%     for k = 1:N
%         % ???k???
%         s =x((k-1)*M + 1:k*M);
%         % ???k????????
%         spl(1,k) = SPLCal(s,fs,framlen);
%     end
%     %%------------??------
%     t = 1:Length;
```

244

```matlab
%     SPL = zeros(1,Length);
%     for r = 1:N
%         SPL(1,(r-1)*M+1:r*M) = spl(r);
%     end
%
%     plot(t/fs,x);
%     axis off
%     saveas(gcf,['D:\image\??\',num2str(i1),'.jpg']);
% end
%
% ???
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;


     x=d;
    Length=length(x);
    framlen = 100;
    M=fs*framlen/1000;
    m = mod(Length,M);
    if m >= M/2 % ??
        % ??????
        x = [x,zeros(1,M-m)];
        % ????????
        Length = length(x);
    else  % ?m < M/2???????????
        % l?Length/M?????
        l = floor(Length/M);
        % ??????
        x=x';
        x = x(1:M*l,1);
        % ????????
        Length = length(x);
    end
    % ??????????
    N = Length/M;
    %%----------------------?????-------------
    s = zeros(1,N);
```

245

```matlab
    % N???????????spl???
    spl = zeros(1,N);
    for k = 1:N
        % ???k???
        s =x((k-1)*M + 1:k*M);
        % ???k????????
        spl(1,k) = SPLCal(s,fs,framlen);
    end
    %%-----------??------
    t = 1:Length;
    SPL = zeros(1,Length);
    for r = 1:N
        SPL(1,(r-1)*M+1:r*M) = spl(r);
    end

    stairs(t/fs,SPL,'r');
    axis off
    saveas(gcf,['C:\image1050\???\',num2str(i1),'.jpg']);
end
% amp_pks
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;

    [~,f_range,fft_sonar] = positiveFFT(d,fs);
    f=fft_sonar;
    fft_amp=abs(f);
    amp_pks = [];   %????
    amp_pkfs = [];  %?????????
    amp_max = max(fft_amp);      %?????
    amp_min = min(fft_amp);      %?????
    amp_median = median(fft_amp);   %?????
    amp_mean = mean(fft_amp);       %?????
    amp_pk = amp_max - amp_min;     %????
    amp_mph = amp_pk * 0.75;
    N=length(fft_amp);
    for i2 = 1:N
        if amp_mph < fft_amp(i2)
```

```matlab
            amp_pks = [amp_pks,fft_amp(i2)];
            amp_pkfs = [amp_pkfs,i2 / N * fs/ 2];
        end
    end

    plot(amp_pks);
    axis off
    saveas(gcf,['C:\image1050\amp_pks\',num2str(i1),'.jpg']);
end
% amp_pkfs
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;

    [~,f_range,fft_sonar] = positiveFFT(d,fs);
    f=fft_sonar;
    fft_amp=abs(f);
    amp_pks = [];   %????
    amp_pkfs = [];  %?????????
    amp_max = max(fft_amp);     %?????
    amp_min = min(fft_amp);     %?????
    amp_median = median(fft_amp);   %?????
    amp_mean = mean(fft_amp);        %?????
    amp_pk = amp_max - amp_min;     %????
    amp_mph = amp_pk * 0.75;
    N=length(fft_amp);
    for i2 = 1:N
        if amp_mph < fft_amp(i2)
            amp_pks = [amp_pks,fft_amp(i2)];
            amp_pkfs = [amp_pkfs,i2 / N * fs/ 2];
        end
    end

    plot(amp_pkfs);
    axis off
    saveas(gcf,['C:\image1050\amp_pkfs\',num2str(i1),'.jpg']);
end
%% amdf
```

```matlab
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m %1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;

    N1=length(d);
    amdf=zeros(N1,1);
    for k = 1:length(d)
        amdf(k) = sum(abs(d(k:end)-d(1:end-k+1)));%????????????
    end

    plot(amdf);
    axis off
    saveas(gcf,['C:\image1050\amdf\',num2str(i1),'.jpg']);
end

for i=1:1:m %1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;

    N1=length(d);
    amdf=zeros(N1,1);
    for k = 1:length(d)
        amdf(k) = sum(abs(d(k:end)-d(1:end-k+1)));%????????????
    end

    plot(amdf);
    axis off
    saveas(gcf,['C:\image1050\amdf\',num2str(i1),'.jpg']);
end
% formantband
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
```

```matlab
    i1=i+namedelay;

    Formantband=formant(d,fs);

    plot(Formantband);
    axis off
    saveas(gcf,['C:\image1050\formantband\',num2str(i1),'.jpg']);
end
%%
clear all;clc;
tic;
signalsper=5;
points=8;
signals=signalsper*points;
signalset=zeros(signals,100000);
Lenset=zeros(signals,1);
maindir = 'D:\data for localisation 1100mm';
subdir  = dir( maindir );
i1=1;
load('D:\data for localisation 1100mm\errorlog.mat','berrorall');
%
for i = 1 : length( subdir )
   if( isequal( subdir( i ).name, '.' )||...
       isequal( subdir( i ).name, '..')||...
       ~subdir( i ).isdir)
       continue;
   end
   subdirpath = fullfile( maindir, subdir( i ).name, '*.mat' );
   dat = dir( subdirpath );
   % ??
   nameCell = cell(length(dat)-2,1);
   for i2 = 1:length(dat)
      %disp(dat(i2).name);
      nameCell{i2} = dat(i2).name;
   end
   dat1 = sort_nat(nameCell);
   % ?error??
   [m,n]=size(berrorall);
   nameCell1 = cell(m,1);
   for i3 = 1 : m
      nameCell1{i3}=cell2mat(berrorall(i3,1));
   end
   for i4 = 1 : 5%length(dat)
      i5 = 1:1:m;
```

249

```matlab
            jg=strcmp(dat1(i4),nameCell1(i5));
            if sum(jg)>0
                dat1(i4)=[];
            end
        end
        %??????????
        section=0000;
        for j = 1+section : section + signalsper%length( dat )
            datpath = fullfile( maindir, subdir( i ).name, dat1( j ));
            datpath = cell2mat(datpath);
            a=cell2mat(struct2cell(load( datpath )));
            Len=length(a);
            Lenset(i1)=Len;
            signalset(i1,1:Len)=a;
            i1=i1+1;

        end
    end
end
%% ????
clearvars -except signalset Lenset signalsper
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
namedelay=0;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    plot(d);
    axis off
    saveas(gcf,['C:\image1100\Time domain\',num2str(i1),'.jpg']);
end
% ????
clearvars -except signalset Lenset signalsper namedelay fs
data=signalset;
[m,~]=size(data);
D=[];
DA=[];
fs1=50000;
for i=1:1:m
    s0=Lenset(i);
    d=data(i,1:s0);
    FFT_Data1 = fft(d);
```

250

```matlab
    power2 = abs(FFT_Data1).^2/fs1;   % power spectrum
    power2 = power2(1:floor(s0/2));
    [B1,~] = mapminmax(power2',0,1);
    ff2=(1:floor(s0/2))*fs1/s0;
    i1=i+namedelay;
    plot(ff2,B1)
    axis off
    saveas(gcf,['C:\image1100\Frequency
domain\',num2str(i1),'.jpg']);
end
% ???????
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
D=[];
DA=[];
fs1=50000;
for i=1:1:m
    s0=Lenset(i);
    d=data(i,1:s0);
    %[popen,fopen] = periodogram(xs1,[],[],fs1);
    [pbutt,fbutt] = periodogram(d,[],[],fs1);
    i1=i+namedelay;
    plot(fbutt,20*log10(abs(pbutt)))
    axis off
    saveas(gcf,['C:\image1100\Power spectrum
density\',num2str(i1),'.jpg']);
end
% ?????
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
D=[];
DA=[];
Fs=50000;
for i=1:1:m
    s0=Lenset(i);
    d=data(i,1:s0);
    signal = d';
    nw=512;ni=nw/4;
    c=1;
    d=FrequencyCal(signal,nw,ni);

    framerate=1:60; %1:257
```
251

```matlab
    d=d(framerate,1:end);


    tt=(0:ni:(length(signal)-nw))/Fs;
    ff=(0:(nw/2))*Fs/nw;
    ff=ff(framerate);
    imagesc(tt,ff,20*log10(c+abs(d)));
    i1=i+namedelay;
    axis off
    saveas(gcf,['C:\image1100\Power spectrum\',num2str(i1),'.jpg']);
end
% power spectrum II
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
%namedelay=0;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    s0=length(d);
    FFT_Data1 = fft(d);
    power2 = abs(FFT_Data1).^2/fs;   % power spectrum
    power2 = power2(1:floor(s0/2));
    plot(power2);
    axis off
    saveas(gcf,['C:\image1100\power
spectrumII\',num2str(i1),'.jpg']);
end

% periodogram
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    [pbutt,fbutt] = periodogram(d,[],[],fs);
    plot(fbutt,20*log10(abs(pbutt)),'--');
    axis off
```

```matlab
    saveas(gcf,['C:\image1100\periodogram\',num2str(i1),'.jpg']);
end
% pkurtosis
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    d1=d';
    pkurtosis(d1,fs);
    axis off
    saveas(gcf,['C:\image1100\pkurtosis\',num2str(i1),'.jpg']);
end
% kurtogram
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    d1=d';
    kurtogram(d1,fs);
    axis off
    saveas(gcf,['C:\image1100\kurtogram\',num2str(i1),'.jpg']);
end
%% instfreq
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    d1=d';
    instfreq1(d1,fs);
```

```matlab
    axis off
    saveas(gcf,['C:\image1100\instfreq\',num2str(i1),'.jpg']);
end
% hht
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    d1=d';
    hht(d1,fs);
    axis off
    saveas(gcf,['C:\image1100\hht\',num2str(i1),'.jpg']);
end
% ???
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;
    cepstrum = real(ifft(log(abs(fft(d)))));
    during_time = 1/fs:1/fs:length(d)/fs;
    plot(during_time,cepstrum,'color',[29/255 176/255
184/255]);ylim([0 0.01]);
    axis off
    saveas(gcf,['C:\image1100\???\',num2str(i1),'.jpg']);
end
% ??????
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
```

```matlab
    i1=i+namedelay;
    audioIn=d;
    audioBuffered =
buffer(audioIn,round(fs*0.03),round(fs*0.02),'nodelay');
    [p,cf] = poctave(audioBuffered,fs);%Calculate the centroid of the
octave power spectrum over time. Plot the results.
    centroid = spectralCentroid(p,cf);
    t = linspace(0,size(audioIn,1)/fs,size(centroid,1));
    plot(t,centroid);
    axis off
    saveas(gcf,['C:\image1100\??????\',num2str(i1),'.jpg']);
end
% %% ??
% clearvars -except signalset Lenset signalsper namedelay
% data=signalset;
% [m,n2]=size(data);
% cd 'D:\localisation system + machine learning algorithm\laser
welding analysis'
% fs=50000;
% for i=1:1:m
%     d=data(i,1:Lenset(i));
%     i1=i+namedelay;
%
%     x=d;
%     Length=length(x);
%     framlen = 100;
%     M=fs*framlen/1000;
%     m = mod(Length,M);
%     if m >= M/2 % ??
%         % ??????
%         x = [x,zeros(1,M-m)];
%         % ????????
%         Length = length(x);
%     else  % ?m < M/2??????????
%         % l?Length/M?????
%         l = floor(Length/M);
%         % ??????
%         x=x';
%         x = x(1:M*l,1);
%         % ????????
%         Length = length(x);
%     end
%     % ??????????
%     N = Length/M;
```

255

```matlab
%     %%----------------------?????-------------
%     s = zeros(1,N);
%     % N???????????spl???
%     spl = zeros(1,N);
%     for k = 1:N
%         % ???k???
%         s =x((k-1)*M + 1:k*M);
%         % ???k????????
%         spl(1,k) = SPLCal(s,fs,framlen);
%     end
%     %%------------??------
%     t = 1:Length;
%     SPL = zeros(1,Length);
%     for r = 1:N
%         SPL(1,(r-1)*M+1:r*M) = spl(r);
%     end
%
%     plot(t/fs,x);
%     axis off
%     saveas(gcf,['D:\image\??\',num2str(i1),'.jpg']);
% end
%
% ???
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;

     x=d;
    Length=length(x);
    framlen = 100;
    M=fs*framlen/1000;
    m = mod(Length,M);
    if m >= M/2 % ??
        % ??????
        x = [x,zeros(1,M-m)];
        % ????????
        Length = length(x);
    else  % ?m < M/2???????????
```

256

```matlab
        % l?Length/M?????
        l = floor(Length/M);
        % ??????
        x=x';
        x = x(1:M*l,1);
        % ????????
        Length = length(x);
    end
    % ??????????
    N = Length/M;
    %%----------------------?????-------------
    s = zeros(1,N);
    % N???????????spl???
    spl = zeros(1,N);
    for k = 1:N
        % ???k???
        s =x((k-1)*M + 1:k*M);
        % ???k????????
        spl(1,k) = SPLCal(s,fs,framlen);
    end
    %%------------??------
    t = 1:Length;
    SPL = zeros(1,Length);
    for r = 1:N
        SPL(1,(r-1)*M+1:r*M) = spl(r);
    end

    stairs(t/fs,SPL,'r');
    axis off
    saveas(gcf,['C:\image1100\???\',num2str(i1),'.jpg']);
end
% amp_pks
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;

    [~,f_range,fft_sonar] = positiveFFT(d,fs);
    f=fft_sonar;
```

257

```matlab
    fft_amp=abs(f);
    amp_pks = [];   %????
    amp_pkfs = []; %?????????
    amp_max = max(fft_amp);     %?????
    amp_min = min(fft_amp);     %?????
    amp_median = median(fft_amp);   %?????
    amp_mean = mean(fft_amp);       %?????
    amp_pk = amp_max - amp_min;     %????
    amp_mph = amp_pk * 0.75;
    N=length(fft_amp);
    for i2 = 1:N
        if amp_mph < fft_amp(i2)
            amp_pks = [amp_pks,fft_amp(i2)];
            amp_pkfs = [amp_pkfs,i2 / N * fs/ 2];
        end
    end

    plot(amp_pks);
    axis off
    saveas(gcf,['C:\image1100\amp_pks\',num2str(i1),'.jpg']);
end
% amp_pkfs
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;

    [~,f_range,fft_sonar] = positiveFFT(d,fs);
    f=fft_sonar;
    fft_amp=abs(f);
    amp_pks = [];   %????
    amp_pkfs = []; %?????????
    amp_max = max(fft_amp);     %?????
    amp_min = min(fft_amp);     %?????
    amp_median = median(fft_amp);   %?????
    amp_mean = mean(fft_amp);       %?????
    amp_pk = amp_max - amp_min;     %????
    amp_mph = amp_pk * 0.75;
    N=length(fft_amp);
```

258

```matlab
    for i2 = 1:N
        if amp_mph < fft_amp(i2)
            amp_pks = [amp_pks,fft_amp(i2)];
            amp_pkfs = [amp_pkfs,i2 / N * fs/ 2];
        end
    end

    plot(amp_pkfs);
    axis off
    saveas(gcf,['C:\image1100\amp_pkfs\',num2str(i1),'.jpg']);
end
%% amdf
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m %1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;

    N1=length(d);
    amdf=zeros(N1,1);
    for k = 1:length(d)
        amdf(k) = sum(abs(d(k:end)-d(1:end-k+1)));%????????????
    end

    plot(amdf);
    axis off
    saveas(gcf,['C:\image1100\amdf\',num2str(i1),'.jpg']);
end

for i=1:1:m %1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;

    N1=length(d);
    amdf=zeros(N1,1);
    for k = 1:length(d)
        amdf(k) = sum(abs(d(k:end)-d(1:end-k+1)));%????????????
    end

    plot(amdf);
```

259

```matlab
    axis off
    saveas(gcf,['C:\image1100\amdf\',num2str(i1),'.jpg']);
end
% formantband
clearvars -except signalset Lenset signalsper namedelay
data=signalset;
[m,n2]=size(data);
cd 'D:\localisation system + machine learning algorithm\laser welding
analysis'
fs=50000;
for i=1:1:m
    d=data(i,1:Lenset(i));
    i1=i+namedelay;

    Formantband=formant(d,fs);

    plot(Formantband);
    axis off
    saveas(gcf,['C:\image1100\formantband\',num2str(i1),'.jpg']);
end
```

# Appendix B DAQ-2010 configuration and codes

## B.1 Signal acquisition with DAQ-2010

*B.1.1 The flow chart of the DAQ-2010 data acquisition card*



Figure B-1 DAQ-2010 configurations

*B.1.2 API functions*

The DAQ-2010 has a powerful sampling functionality. The data acquisition card samples acoustic signals in the two-dimensional and three-dimensional localisation tests.

The data acquisition card is controlled with MATLAB Application Programming Interface (API) functions. The operational code is as follows:

```
function datacollectionforlocalisation
clc;
clear all;
close all;
% Profile on
addpath('D:\MATLAB.m');
loadlibrary('D:\MATLAB.m\D2KDask64.dll','D:\MATLAB.m\D2KDask64.h')
LIB = 'D2KDask64';
libfunctions D2KDask64 -full;
% running counter
load('D:\DAQ-2010 MATLAB MEX\rounds_of_rawdata','rounds_of_rawdata');
sequence=find(rounds_of_rawdata==0);
rounds_of_rawdata(sequence(1))=1;
save('D:\DAQ-2010 MATLAB
MEX\rounds_of_rawdata.mat','rounds_of_rawdata');
% variable name
global name
name=['sample_40dB_testID_',int2str(sequence(1))];
%%
card_type = D2KDASK.DAQ_2010;
card_num = uint16(0);
Channel_Index = uint16(D2KDASK.All_Channels);
Channel = uint16(4);
Last_Channel = Channel - 1;
AdRange_RefGnd = D2KDASK.AD_B_10_V;
SyncMode = D2KDASK.ASYNCH_OP;
ConfigCtrl =
bitor(D2KDASK.DAQ2K_AI_ADSTARTSRC_Int,D2KDASK.DAQ2K_AI_ADCONVSRC_Int)
;
TrigCtrl = D2KDASK.DAQ2K_AI_TRGMOD_POST|D2KDASK.DAQ2K_AI_TRGSRC_SOFT;
MidOrDlyScans = uint32(0);
MCnt = uint32(0);
ReTrgCnt = uint32(1);
P2010_TIMEBASE = D2KDASK.P2010_TIMEBASE;
```

```matlab
% Sampling rate control/Scan for single channel while sample for
multiple
ScanIntrv = uint32(200)*uint32(Channel);
SampIntrv = uint32(200);
ScanningRate = double(P2010_TIMEBASE / ScanIntrv);
SamplingRate = double(P2010_TIMEBASE / SampIntrv);
% Size per upload
AI_ReadCount = uint32(ScanningRate);
colum =
14500; %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Data matrix
voltageArray = zeros(AI_ReadCount*uint32(Channel),1,'double');
buffer0 = zeros(AI_ReadCount*uint32(Channel),1,'uint16');
buffer1 = zeros(AI_ReadCount*uint32(Channel),1,'uint16');
AutoReset = uint32(1);
StartPos = uint32(0);
AccessCnt = uint32(0);
bEnable = uint32(1);
Stopped = uint32(0);
HalfReady = uint32(0);
bufferID0 = uint16(0);
bufferID1 = uint16(0);
% matfile setting
v1=zeros(AI_ReadCount,colum);
%%
card = calllib(LIB,'D2K_Register_Card',card_type,card_num);
if card < 0
      unloadlibrary(LIB);
      error = card;
      fprintf('D2K_Register_Card failed with error
code %d\n',error);
      return;
end
error =
calllib(LIB,'D2K_AI_CH_Config',card,Channel_Index,AdRange_RefGnd);
   if error < 0
      calllib ('dasklib','D2K_Release_Card',card);
      unloadlibrary(LIB);
      fprintf('D2K_AI_CH_Config failed with error code %d\n',error);
      return;
   end
```

```matlab
error =
calllib(LIB,'D2K_AI_Config',0,ConfigCtrl,TrigCtrl,MidOrDlyScans,MCnt,
ReTrgCnt,AutoReset);
    if error < 0
        calllib ('dasklib','D2K_Release_Card',card);
        unloadlibrary(LIB);
        fprintf('D2K_AI_Config failed with error code %d\n',error);
        return;
    end
error = calllib(LIB,'D2K_AI_AsyncDblBufferMode',0,bEnable);
    if error < 0
        calllib(LIB,'D2K_Release_Card',card);
        unloadlibrary(LIB);
        fprintf('D2K_AI_AsyncDblBufferMode failed with error
code %d\n',error);
        return;
    end
%%
pbuffer0 = libpointer('uint16Ptr',buffer0);
[error,tpbuffer0,bufferID0] =
calllib(LIB,'D2K_AI_ContBufferSetup',card,pbuffer0,AI_ReadCount*uint3
2(Channel),bufferID0);
if error < 0
        calllib(LIB,'D2K_AI_ContBufferReset',card);
        calllib(LIB,'D2K_Release_Card',card);
        unloadlibrary(LIB);
        fprintf('D2K_AI_ContBufferSetup failed with error
code %d\n',error);
        return;
end

pbuffer1 = libpointer('uint16Ptr',buffer1);
[error,tpbuffer1,bufferID1] =
calllib(LIB,'D2K_AI_ContBufferSetup',card,pbuffer1,AI_ReadCount*uint3
2(Channel),bufferID1);
if error < 0
        calllib(LIB,'D2K_AI_ContBufferReset',card);
        calllib(LIB,'D2K_Release_Card',card);
        unloadlibrary(LIB);
        fprintf('D2K_AI_ContBufferSetup failed with error
code %d\n',error);
        return;
end
```

```matlab
error = 
calllib(LIB,'D2K_AI_ContScanChannels',card,Last_Channel,bufferID0,AI_
ReadCount,ScanIntrv,SampIntrv,SyncMode);
if error < 0
        calllib(LIB,'D2K_AI_AsyncClear',card,StartPos,AccessCnt);
        calllib(LIB,'D2K_AI_ContBufferReset',card);
        calllib(LIB,'D2K_Release_Card',card);
        unloadlibrary(LIB);
        fprintf('D2K_AI_ContScanChannel failed with error
code %d\n',error);
        return;
end
%%
margin = 10.0;
TimeOut = double(AI_ReadCount*uint32(Channel)) / SamplingRate +
margin;
tic;
index = 0;
TimeLeft = TimeOut - toc;
fprintf('Start AI, press anykey on figure to stop\n');
figh = figure('keypressfcn',@(obj,ev) set(obj,'userdata',1));
bcounting=0;
while isempty(get(figh,'userdata')) && TimeLeft >= 0
    pause(0.1)
    [error,HalfReady,Stopped] =
calllib(LIB,'D2K_AI_AsyncDblBufferHalfReady',card,HalfReady,Stopped);
    if error < 0
        calllib(LIB,'D2K_AI_AsyncClear',card,StartPos,AccessCnt);
        calllib(LIB,'D2K_AI_ContBufferReset',card);
        calllib(LIB,'D2K_Release_Card',card);
        unloadlibrary(LIB);
        fprintf('D2K_AI_AsyncDblBufferHalfReady failed with error
code %d\n',error);
        return;
    end
    if HalfReady == true
        TimeLeft = TimeOut-toc;
        error = calllib(LIB,'D2K_AI_AsyncDblBufferHandled',card);
        if error < 0
            calllib(LIB,'D2K_AI_AsyncClear',card,AccessCnt);
            calllib(LIB,'D2K_Release_Card',card);
            unloadlibrary(LIB);
            fprintf('D2K_AI_AsyncDblBufferHandled failed with error
code %d\n',error);
```

```matlab
            return;
        end
        if index == 0
            buffer0 = pbuffer0.Value;
            [error,buffer0,voltageArray] =
calllib(LIB,'D2K_AI_ContVScale',card,AdRange_RefGnd,buffer0,voltageAr
ray,AI_ReadCount*uint32(Channel));
            index = 1;
            fprintf('Buffer 0 HalfReady, press anykey on figure to
stop\n');
        else
            buffer1 = pbuffer1.Value;
            [error,buffer1,voltageArray] =
calllib(LIB,'D2K_AI_ContVScale',card,AdRange_RefGnd,buffer1,voltageAr
ray,AI_ReadCount*uint32(Channel));
            index = 0;
            fprintf('Buffer 1 HalfReady, press anykey on figure to
stop\n');
        end
        % v storage
        data=stack(voltageArray);
        tic
        bcounting=bcounting+1;
        fprintf('Writing %d of %d\n',bcounting,colum);
        v1(:,bcounting) = data;
        toc
    end
    if bcounting == colum
        fprintf('voltageArray full\n');
        break;
    end
    if Stopped == true
        break;
    end
    TimeLeft = TimeOut - toc;
    pause(0.1)
end
fprintf('Stop AI\n');
if TimeLeft < 0
        calllib(LIB,'D2K_AI_AsyncClear',card,StartPos,AccessCnt);
        calllib(LIB,'D2K_AI_ContBufferReset',card);
        calllib(LIB,'D2K_Release_Card',card);
        unloadlibrary(LIB);
        fprintf('D2K_AI_AsyncCheck time out.\n');
```

266

```matlab
        return;
end
[error,StartPos,AccessCnt] =
calllib(LIB,'D2K_AI_AsyncClear',card,StartPos,AccessCnt);
if error < 0
        calllib(LIB,'D2K_AI_ContBufferReset',card);
        calllib(LIB,'D2K_Release_Card',card);
        unloadlibrary(LIB);
        fprintf('D2K_AI_AsyncClear failed with error
code %d\n',error);
        return;
end
if ~AutoReset
    error = calllib(LIB,'D2K_AI_ContBufferReset',card);
    if error < 0
        calllib(LIB,'D2K_Release_Card',card);
        unloadlibrary(LIB);
        fprintf('D2K_AI_ContBufferReset failed with error
code %d\n',error);
        return;
    end
end
calllib (LIB,'D2K_Release_Card',card);
unloadlibrary(LIB);
clearvars -except v1 bcounting name
v1=v1(:,(1:bcounting));
if bcounting > 48
save(['D:\DAQ-2010 MATLAB MEX\samples\',name],'v1','-v7.3');
end
%%
```

## B.2 Signal isolation

Acoustic signals in the collected one-dimensional signal sequence must be separated so acoustic samples from different locations can be processed independently. In this research, a signal waveform-based separation algorithm is compiled, and it achieved desired results. It separates acoustic signals according to the detected peak values and the prediction of sampling points. The code is as follows:

```matlab
function v1 = automaticfilling(v1)
v2=v1;
v3=v1;
stdlen=2555000;
margin=50000;
```

267

```matlab
margin2=30000;
interval=180000;
interval2=2400000;
interval3=300000;
Highvoltagecontroll =0.12;
Highvoltagecontrol2 =max(v1)-0.4;%0.4 for 1200mm, 0.30 for 1000mm
v2(v2<Highvoltagecontroll)=0;
v3(v3<Highvoltagecontrol2)=0;
%% ??????-maxl,????????-maxl1
[~,maxl]=findpeaks(v2,'minpeakdistance',interval);
[maxv1,maxl1]=findpeaks(v3,'minpeakdistance',interval3);
global index_of_maxl
index_of_maxl=isempty(maxl);
%% ??????
if index_of_maxl==0
    %% ?????????-??maxl1????????
    %??
    highestsig = find(maxv1 == max(maxv1));
    redun = 1000;
    while maxl1(highestsig)<180000000-interval2-750000
        Dbound = maxl1(highestsig)+interval2+redun;
        maxl1(maxl1 > maxl1(highestsig) & maxl1 < Dbound)=[];
        highestsig=highestsig+1;
        if highestsig > length(maxl1)
            break
        end
    end
    %??
    highestsig = find(maxv1 == max(maxv1));%???
    while maxl1(highestsig)>interval2+750000
        Dbound = maxl1(highestsig)-interval2-redun;
        maxl1(maxl1 < maxl1(highestsig) & maxl1 > Dbound)=[];
        highestsig=highestsig-1;
        if highestsig>length(maxl1)
            highestsig=length(maxl1);
        end
        if highestsig < 1
            break
        end
    end
    %????
    maxl1(maxl1<interval2) = [];
    Len1 = length(maxl1);
    maxl1(maxl1>length(v1)-stdlen)=[];
```

```matlab
%% ??????+?????
%maxl2=maxl1;%sigseq1=2448000;
load('D:\DAQ-2010 MATLAB MEX\sigseqB','sigseqB')
%load('D:\DAQ-2010 MATLAB MEX\sigseq_1000mm','sigseq_1000mm')
load('D:\DAQ-2010 MATLAB MEX\sigseq_800mm','sigseq_800mm')
%sigseq=sigseq_1000mm;
sigseq=sigseq_800mm;
missingpoints=zeros(1,1);
newpoints=zeros(1000,1);
pointer=1;
while isempty(missingpoints) == 0
    origseq=diff(maxl1);
    missingpoints=find(origseq > stdlen+margin);
    i = 1:1:length(missingpoints);
    newpoint=maxl1(missingpoints(i))+stdlen;
    if isempty(newpoint) == 0
        newpoints(pointer,1)=newpoint;
    end
    pointer=pointer+1;
    maxl1=[maxl1',newpoint']';
    maxl1=sort(maxl1);
end
% new points = 1.2 and v1 splitting
newpoints(newpoints==0)=[];
v1(newpoints)=1.2;
i11=length(newpoints);
fprintf('%d high points have been added\n',i11);
%% v1??-??startpoint?endpoint
% 1 ???????????????
sigseqA=v1(maxl1(2):maxl1(4));
sigseqA(sigseqA<Highvoltagecontroll)=0;
[~,maxl2]=findpeaks(sigseqA,'minpeakdistance',interval);
diffvalue=diff(maxl2);
Len=8;%figure; plot(sigseqA)
%%
for i1=1:Len
    compareseq1=diffvalue(i1:Len+i1-1);
    compareseq1(1)=1;
    diffvalue1=compareseq1-sigseq;
    index(i1)=sum(abs(diffvalue1));
end
index=abs(index(1:7));
startbit=find(index==min(index));
startbit1=find(maxl==maxl1(2));
```

269

```matlab
    sigseqA=v1(maxl(startbit1+startbit):maxl(startbit1+startbit+7));
    sigseqA(sigseqA<Highvoltagecontroll)=0;
    %figure;plot(sigseqA)
%    [maxv,~]=findpeaks(sigseqA,'minpeakdistance',interval);
%    highestsignal=find(maxv==max(maxv));
%    sigseqC=sigseqB+abs(sigseqB(1));
%    sigseqC1=sigseqC-sigseqC(highestsignal);
    %borderstart = maxl1(2)+sigseqC1(1)-margin;% 1000mm
    borderstart = maxl(startbit1+startbit)-margin;
    %% 2 ???????????????
    Lenomaxl1=length(maxl);
    sigseqA=v1(maxl1(Lenomaxl1-4):maxl1(Lenomaxl1-2));
    sigseqA(sigseqA<Highvoltagecontroll)=0;
    [~,maxl2]=findpeaks(sigseqA,'minpeakdistance',interval);
    diffvalue=diff(maxl2);
    Len=8;%figure; plot(sigseqA)
    %%
    for i1=1:Len
        compareseq1=diffvalue(i1:Len+i1-1);
        compareseq1(1)=1;
        diffvalue1=compareseq1-sigseq;
        index(i1)=sum(abs(diffvalue1));
    end
    index=abs(index(1:7));
    startbit=find(index==min(index));
    startbit1=find(maxl==maxl1(Lenomaxl1-3));
    sigseqA=v1(maxl(startbit1+startbit):maxl(startbit1+startbit+7));
    sigseqA(sigseqA<Highvoltagecontroll)=0;
    %figure;plot(sigseqA)
    borderend = maxl(startbit1+startbit-1)+margin;% -
1???A????????????A
    v1=v1(borderstart:borderend);% new v1-?????8???
    %%???
    v2=v1;
    v3=v1;
    v2(v2<Highvoltagecontroll)=0;
    v3(v3<Highvoltagecontrol2)=0;
    [~,maxl]=findpeaks(v2,'minpeakdistance',interval); % 100%
accurate highest detection
    [~,maxl1]=findpeaks(v3,'minpeakdistance',interval2);% discrete
dispersion of points.
    cycle=8;
    lenmaxl=length(maxl);
    lenmaxl1=length(maxl1);
```

```matlab
        sequence=zeros(lenmaxl1+1,1);
        sequence(lenmaxl1+1)=lenmaxl+1;
        pointer2=1;
        newpoints2=zeros(600,1);
        for pointer1=1:1:lenmaxl1
            sequence(pointer1,1)=find(maxl==maxl1(pointer1))-1;
        end
        while lenmaxl/cycle < lenmaxl1
            diffvalue=diff(sequence);
            missinggrp=(find(diffvalue<8));
            for i1=1:1:length(missinggrp)
                stdset=(sigseqB+maxl1(missinggrp(i1)))';%standard grp
                startpoint=stdset(1)-margin2;
                endpoint=stdset(8)+margin2;
                compset1=find(maxl>startpoint & maxl<endpoint);
                compset=[maxl(compset1)',endpoint]';
                for i2 = 1:1:length(sigseq)
                    diffvalue = abs(stdset(i2) - compset(i2));
                    if diffvalue > margin2-1
                        newpoint2=stdset(i2);
                        compset=[compset',newpoint2]';
                        compset=sort(compset);
                        newpoints2(pointer2,1)=newpoint2;
                    end
                    pointer2=pointer2+1;
                end
            end
            maxl=[maxl',newpoints2']';
            sort(maxl);
            lenmaxl=length(maxl);
        end
        newpoints2(newpoints2==0)=[];
        i12=length(newpoints2);
        fprintf('%d low points have been added\n',i12);
        v1(newpoints2)=0.25;
else
    fprintf('no signals detected\n');
end
```

## B.3 Machine learning algorithm compilation

### *B.3.1 Random Forest code in Python*

```
# Multiple Inputs
import keras
from keras.utils.vis_utils import plot_model
from keras.models import Model
from keras.layers import Input
from keras.layers import Dense
from keras.layers import Flatten
from keras.layers.convolutional import Conv2D
from keras.layers.pooling import MaxPooling2D
from keras.layers.merge import concatenate
from keras.preprocessing.image import ImageDataGenerator
from sklearn import metrics
import os
import pandas as pd
import os.path
import numpy as np
from random import randint
import tensorflow as tf
from keras.backend import set_session
import keras.backend as KTF
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Flatten, Conv2D,
MaxPooling2D
from keras.utils import np_utils
from keras.datasets import mnist
import cv2
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import time
from keras.models import load_model
# import seaborn as sns
# from PIL import Image
# os.environ["CUDA_VISIBLE_DEVICES"] = "-1"
# os.environ["CUDA_VISIBLE_DEVICES"] = "0"
# os.environ['KERAS_BACKEND']='tensorflow'
config = tf.compat.v1.ConfigProto()
config.gpu_options.allow_growth = True
sess = tf.compat.v1.Session(config=config)
KTF.set_session(sess)

cycle = 30
```

```python
np.random.seed(123)
X, y = np.arange(10).reshape((5, 2)), range(5)
labels = ['Change of centre frequency', 'Frequency domain', 'Power
spectrum', 'Power spectrum density', 'Time domain',
          'amdf', 'amp_pkfs', 'cepstrum', 'formantband', 'hht',
'instfreq', 'kurtogram', 'periodogram', 'pkurtosis',
          'power spectrumII', 'sound pressure level', 'amp_pks']
img_size = 60


def get_data(data_dir):
    data = []
    ydata = []
    for label in labels:
        path = os.path.join(data_dir, label)
        class_num = labels.index(label)
        print(path)
        for img in os.listdir(path):
            try:
                img_arr = cv2.imread(os.path.join(path,
img))[..., ::-1]  # convert BGR to RGB format
                resized_arr = cv2.resize(img_arr, (img_size,
img_size))  # Reshaping images to preferred size
                # data.append([resized_arr, class_num])
                data.append(resized_arr)
                ydata.append(class_num)
                # print ("class", class_num)
            except Exception as e:
                print(e)
    return np.array(data), np.array(ydata)


Xdata, Ydata = get_data("C:\CNNdata3(CNNcode-mode-10)\\1")
X_train, X_test, y_train, y_test = train_test_split(Xdata, Ydata,
test_size=0.2, random_state=42)
print(y_test)
# load pre-shuffled MNIST data
train = get_data("C:\CNNdata3(CNNcode-mode-10)\\1")
X_train1, X_test1, y_train1, y_test1 = train_test_split(Xdata, Ydata,
test_size=0.2, random_state=42)
Y_train1 = np_utils.to_categorical(y_train1, 17)
Y_test1 = np_utils.to_categorical(y_test1, 17)
train = get_data("C:\CNNdata3(CNNcode-mode-10)\\2")
```

```python
X_train2, X_test2, y_train2, y_test2 = train_test_split(Xdata, Ydata,
test_size=0.2, random_state=42)
Y_train2 = np_utils.to_categorical(y_train2, 17)
Y_test2 = np_utils.to_categorical(y_test2, 17)
train = get_data("C:\CNNdata3(CNNcode-mode-10)\\3")
X_train3, X_test3, y_train3, y_test3 = train_test_split(Xdata, Ydata,
test_size=0.2, random_state=42)
Y_train3 = np_utils.to_categorical(y_train3, 17)
Y_test3 = np_utils.to_categorical(y_test3, 17)
train = get_data("C:\CNNdata3(CNNcode-mode-10)\\4")
X_train4, X_test4, y_train4, y_test4 = train_test_split(Xdata, Ydata,
test_size=0.2, random_state=42)
Y_train4 = np_utils.to_categorical(y_train4, 17)
Y_test4 = np_utils.to_categorical(y_test4, 17)
train = get_data("C:\CNNdata3(CNNcode-mode-10)\\5")
X_train5, X_test5, y_train5, y_test5 = train_test_split(Xdata, Ydata,
test_size=0.2, random_state=42)
Y_train5 = np_utils.to_categorical(y_train5, 17)
Y_test5 = np_utils.to_categorical(y_test5, 17)
train = get_data("C:\CNNdata3(CNNcode-mode-10)\\6")
X_train6, X_test6, y_train6, y_test6 = train_test_split(Xdata, Ydata,
test_size=0.2, random_state=42)
Y_train6 = np_utils.to_categorical(y_train6, 17)
Y_test6 = np_utils.to_categorical(y_test6, 17)
train = get_data("C:\CNNdata3(CNNcode-mode-10)\\7")
X_train7, X_test7, y_train7, y_test7 = train_test_split(Xdata, Ydata,
test_size=0.2, random_state=42)
Y_train7 = np_utils.to_categorical(y_train7, 17)
Y_test7 = np_utils.to_categorical(y_test7, 17)
train = get_data("C:\CNNdata3(CNNcode-mode-10)\\8")
X_train8, X_test8, y_train8, y_test8 = train_test_split(Xdata, Ydata,
test_size=0.2, random_state=42)
Y_train8 = np_utils.to_categorical(y_train8, 17)
Y_test8 = np_utils.to_categorical(y_test8, 17)
# first input model
visible1 = Input(shape=(60, 60, 3))
conv11 = Conv2D(28, kernel_size=4, activation='relu')(visible1)
pool11 = MaxPooling2D(pool_size=(2, 2))(conv11)
conv12 = Conv2D(14, kernel_size=4, activation='relu')(pool11)
pool12 = MaxPooling2D(pool_size=(2, 2))(conv12)
flat1 = Flatten()(pool12)
# second input model
visible2 = Input(shape=(60, 60, 3))
conv21 = Conv2D(28, kernel_size=4, activation='relu')(visible2)
```

```python
pool21 = MaxPooling2D(pool_size=(2, 2))(conv21)
conv22 = Conv2D(14, kernel_size=4, activation='relu')(pool21)
pool22 = MaxPooling2D(pool_size=(2, 2))(conv22)
flat2 = Flatten()(pool22)
# third input model
visible3 = Input(shape=(60, 60, 3))
conv31 = Conv2D(28, kernel_size=4, activation='relu')(visible3)
pool31 = MaxPooling2D(pool_size=(2, 2))(conv31)
conv32 = Conv2D(14, kernel_size=4, activation='relu')(pool31)
pool32 = MaxPooling2D(pool_size=(2, 2))(conv32)
flat3 = Flatten()(pool32)
# forth input model
visible4 = Input(shape=(60, 60, 3))
conv41 = Conv2D(28, kernel_size=4, activation='relu')(visible4)
pool41 = MaxPooling2D(pool_size=(2, 2))(conv41)
conv42 = Conv2D(14, kernel_size=4, activation='relu')(pool41)
pool42 = MaxPooling2D(pool_size=(2, 2))(conv42)
flat4 = Flatten()(pool42)
# 17th input model
visible5 = Input(shape=(60, 60, 3))
conv51 = Conv2D(28, kernel_size=4, activation='relu')(visible5)
pool51 = MaxPooling2D(pool_size=(2, 2))(conv51)
conv52 = Conv2D(14, kernel_size=4, activation='relu')(pool51)
pool52 = MaxPooling2D(pool_size=(2, 2))(conv52)
flat5 = Flatten()(pool52)
# 6th input model
visible6 = Input(shape=(60, 60, 3))
conv61 = Conv2D(28, kernel_size=4, activation='relu')(visible6)
pool61 = MaxPooling2D(pool_size=(2, 2))(conv61)
conv62 = Conv2D(14, kernel_size=4, activation='relu')(pool61)
pool62 = MaxPooling2D(pool_size=(2, 2))(conv62)
flat6 = Flatten()(pool62)
# 7th input model
visible7 = Input(shape=(60, 60, 3))
conv71 = Conv2D(28, kernel_size=4, activation='relu')(visible7)
pool71 = MaxPooling2D(pool_size=(2, 2))(conv71)
conv72 = Conv2D(14, kernel_size=4, activation='relu')(pool71)
pool72 = MaxPooling2D(pool_size=(2, 2))(conv72)
flat7 = Flatten()(pool72)
# 8th input model
visible8 = Input(shape=(60, 60, 3))
conv81 = Conv2D(28, kernel_size=4, activation='relu')(visible8)
pool81 = MaxPooling2D(pool_size=(2, 2))(conv81)
conv82 = Conv2D(14, kernel_size=4, activation='relu')(pool81)
```

```python
pool82 = MaxPooling2D(pool_size=(2, 2))(conv82)
flat8 = Flatten()(pool82)
# merge input models
merge = concatenate([flat1, flat2, flat3, flat4, flat5, flat6, flat7,
flat8])
# interpretation model
hidden1 = Dense(544, activation='relu')(merge)
hidden2 = Dense(272, activation='relu')(hidden1)
hidden3 = Dense(136, activation='relu')(hidden2)
hidden4 = Dense(68, activation='relu')(hidden3)
hidden5 = Dense(34, activation='relu')(hidden4)
output1 = Dense(17, activation='softmax')(hidden5)
output2 = Dense(17, activation='softmax')(hidden5)
output3 = Dense(17, activation='softmax')(hidden5)
output4 = Dense(17, activation='softmax')(hidden5)
output5 = Dense(17, activation='softmax')(hidden5)
output6 = Dense(17, activation='softmax')(hidden5)
output7 = Dense(17, activation='softmax')(hidden5)
output8 = Dense(17, activation='softmax')(hidden5)
model = Model(inputs=[visible1, visible2, visible3, visible4,
visible5, visible6, visible7, visible8],
              outputs=[output1, output2, output3, output4, output5,
output6, output7, output8])
# summarize layers
print(model.summary())
# plot graph
model.compile(loss='categorical_crossentropy', optimizer='adam',
metrics=['accuracy'])
print(model.metrics_names)
A = []
A0 = []
A1 = []
A2 = []
A3 = []
A4 = []
A5 = []
A6 = []
A7 = []
A8 = []
A9 = []
A10 = []
A11 = []
A12 = []
A13 = []
```

```
A14 = []
A15 = []
A16 = []
A17 = []
A18 = []
A19 = []
A20 = []
A21 = []
A22 = []
A23 = []
A24 = []
C = []
C1 = []
C2 = []
C3 = []
C4 = []
C5 = []
C6 = []
C7 = []
C8 = []
C9 = []
C10 = []
C11 = []
C12 = []
C13 = []
C14 = []
C15 = []
C16 = []
C17 = []
C18 = []
C19 = []
C20 = []
C21 = []
C22 = []
C23 = []
C24 = []
D = []
for i in range(cycle):
    a = model.fit([X_train1, X_train2, X_train3, X_train4, X_train5,
X_train6, X_train7, X_train8],
                  [Y_train1, Y_train2, Y_train3, Y_train4, Y_train5,
Y_train6, Y_train7, Y_train8],
                  batch_size=40, epochs=1, verbose=1)
    # print(a.history['dense_5_loss'])
```

```python
    # print(a.history['dense_5_accuracy'])
    # b = abs(float(a.history['dense_5_loss'][0]))
    # c = abs(float(a.history['dense_5_accuracy'][0]))
    b = abs(float(a.history['loss'][0]))
    b1 = abs(float(a.history['dense_5_loss'][0]))
    b2 = abs(float(a.history['dense_6_loss'][0]))
    b3 = abs(float(a.history['dense_7_loss'][0]))
    b4 = abs(float(a.history['dense_8_loss'][0]))
    b5 = abs(float(a.history['dense_9_loss'][0]))
    b6 = abs(float(a.history['dense_10_loss'][0]))
    b7 = abs(float(a.history['dense_11_loss'][0]))
    b8 = abs(float(a.history['dense_12_loss'][0]))

    c1 = abs(float(a.history['dense_5_accuracy'][0]))
    c2 = abs(float(a.history['dense_6_accuracy'][0]))
    c3 = abs(float(a.history['dense_7_accuracy'][0]))
    c4 = abs(float(a.history['dense_8_accuracy'][0]))
    c5 = abs(float(a.history['dense_9_accuracy'][0]))
    c6 = abs(float(a.history['dense_10_accuracy'][0]))
    c7 = abs(float(a.history['dense_11_accuracy'][0]))
    c8 = abs(float(a.history['dense_12_accuracy'][0]))

    A.append(b)
    A1.append(b1)
    A2.append(b2)
    A3.append(b3)
    A4.append(b4)
    A5.append(b5)
    A6.append(b6)
    A7.append(b7)
    A8.append(b8)

    C1.append(c1)
    C2.append(c2)
    C3.append(c3)
    C4.append(c4)
    C5.append(c5)
    C6.append(c6)
    C7.append(c7)
    C8.append(c8)

A0 = np.array(A1)
C = np.array(C1)
B = np.array(range(len(A0)))
```

278

```python
D = [A, A1, A2, A3, A4, A5, A6, A7, A8,
     C1, C2, C3, C4, C5, C6, C7, C8]
np.savetxt("history.txt", D, delimiter=',')


plt.plot(B, A0, 'o-', label='loss')
plt.plot(B, C, 'o-', label='accuracy')
plt.title('model accuracy')
plt.xlabel('epoch')
plt.ylabel('accuracy')
plt.legend(loc='best')
plot_model(model, to_file='multiple_inputs.png')
model.save('final_model.h5')
print("model saved")
print('waiting for model save')
time.sleep(60)
print('model save complete')
model = load_model('final_model.h5')
print("Loaded model from disk")
predict = model.predict([X_test1, X_test2, X_test3, X_test4, X_test5,
X_test6, X_test7, X_test8])
l0 = []
for m in range(0, len(predict)):
    l1 = predict[m]
    l1 = l1.tolist()
    # print(l1)
    for i in range(0, len(l1)):
        # print(l1[i])
        l0.append(l1[i].index(max(l1[i])))
np.set_printoptions(threshold=np.inf)
pd.set_option('display.width', 300)  #
pd.set_option('display.max_rows', None)  #
pd.set_option('display.max_columns', None)  #
l0 = np.array(l0)
l2 = np.array([y_test1, y_test2, y_test3, y_test4, y_test5, y_test6,
y_test7, y_test8])
l3 = []
for m in range(0, len(l2)):
    for i in l2[m]:
        l3.append(i)
l3 = np.array(l3)
accuracy = metrics.accuracy_score(l3, l0)
Accuracy = [accuracy]
print("Accuracy of model using test data:", Accuracy)
np.savetxt("Accuracy of model using test data.txt", Accuracy)
```

279

```
print(plt.axis([0, cycle, 0, 1.9]))
plt.show()
```

### B.3.2 CNN code in Python

```python
# Multiple Inputs
import keras
from keras.utils.vis_utils import plot_model
from keras.models import Model
from keras.layers import Input
from keras.layers import Dense
from keras.layers import Flatten
from keras.layers.convolutional import Conv2D
from keras.layers.pooling import MaxPooling2D
from keras.layers.merge import concatenate
from keras.preprocessing.image import ImageDataGenerator
from sklearn import metrics
import os
import pandas as pd
import os.path
import numpy as np
from random import randint
import tensorflow as tf
from keras.backend import set_session
import keras.backend as KTF
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Flatten, Conv2D,
MaxPooling2D
from keras.utils import np_utils
from keras.datasets import mnist
import cv2
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import time
from keras.models import load_model
# import seaborn as sns
# from PIL import Image
# os.environ["CUDA_VISIBLE_DEVICES"] = "-1"
# os.environ["CUDA_VISIBLE_DEVICES"] = "0"
# os.environ['KERAS_BACKEND']='tensorflow'
config = tf.compat.v1.ConfigProto()
config.gpu_options.allow_growth = True  # RAM distribution
sess = tf.compat.v1.Session(config=config)
KTF.set_session(sess)

cycle = 30
```

```python
np.random.seed(123)
X, y = np.arange(10).reshape((5, 2)), range(5)
labels = ['Change of centre frequency', 'Frequency domain', 'Power
spectrum', 'Power spectrum density', 'Time domain',
          'amdf', 'amp_pkfs', 'cepstrum', 'formantband', 'hht',
'instfreq', 'kurtogram', 'periodogram', 'pkurtosis',
          'power spectrumII', 'sound pressure level', 'amp_pks']
img_size = 60


def get_data(data_dir):
    data = []
    ydata = []
    for label in labels:
        path = os.path.join(data_dir, label)
        class_num = labels.index(label)
        print(path)
        for img in os.listdir(path):
            try:
                img_arr = cv2.imread(os.path.join(path,
img))[..., ::-1]  # convert BGR to RGB format
                resized_arr = cv2.resize(img_arr, (img_size,
img_size))  # Reshaping images to preferred size
                # data.append([resized_arr, class_num])
                data.append(resized_arr)
                ydata.append(class_num)
                # print ("class", class_num)
            except Exception as e:
                print(e)
    return np.array(data), np.array(ydata)


Xdata, Ydata = get_data("C:\CNNdata3(CNNcode-mode-23)\\1")
X_train, X_test, y_train, y_test = train_test_split(Xdata, Ydata,
test_size=0.2, random_state=42)
print(y_test)
# load pre-shuffled MNIST data
train = get_data("C:\CNNdata3(CNNcode-mode-23)\\1")
X_train1, X_test1, y_train1, y_test1 = train_test_split(Xdata, Ydata,
test_size=0.2, random_state=42)
Y_train1 = np_utils.to_categorical(y_train1, 17)
Y_test1 = np_utils.to_categorical(y_test1, 17)
train = get_data("C:\CNNdata3(CNNcode-mode-23)\\2")
```

```python
X_train2, X_test2, y_train2, y_test2 = train_test_split(Xdata, Ydata,
test_size=0.2, random_state=42)
Y_train2 = np_utils.to_categorical(y_train2, 17)
Y_test2 = np_utils.to_categorical(y_test2, 17)
train = get_data("C:\CNNdata3(CNNcode-mode-23)\\3")
X_train3, X_test3, y_train3, y_test3 = train_test_split(Xdata, Ydata,
test_size=0.2, random_state=42)
Y_train3 = np_utils.to_categorical(y_train3, 17)
Y_test3 = np_utils.to_categorical(y_test3, 17)
train = get_data("C:\CNNdata3(CNNcode-mode-23)\\4")
X_train4, X_test4, y_train4, y_test4 = train_test_split(Xdata, Ydata,
test_size=0.2, random_state=42)
Y_train4 = np_utils.to_categorical(y_train4, 17)
Y_test4 = np_utils.to_categorical(y_test4, 17)
train = get_data("C:\CNNdata3(CNNcode-mode-23)\\5")
X_train5, X_test5, y_train5, y_test5 = train_test_split(Xdata, Ydata,
test_size=0.2, random_state=42)
Y_train5 = np_utils.to_categorical(y_train5, 17)
Y_test5 = np_utils.to_categorical(y_test5, 17)
train = get_data("C:\CNNdata3(CNNcode-mode-23)\\6")
X_train6, X_test6, y_train6, y_test6 = train_test_split(Xdata, Ydata,
test_size=0.2, random_state=42)
Y_train6 = np_utils.to_categorical(y_train6, 17)
Y_test6 = np_utils.to_categorical(y_test6, 17)
train = get_data("C:\CNNdata3(CNNcode-mode-23)\\7")
X_train7, X_test7, y_train7, y_test7 = train_test_split(Xdata, Ydata,
test_size=0.2, random_state=42)
Y_train7 = np_utils.to_categorical(y_train7, 17)
Y_test7 = np_utils.to_categorical(y_test7, 17)
train = get_data("C:\CNNdata3(CNNcode-mode-23)\\8")
X_train8, X_test8, y_train8, y_test8 = train_test_split(Xdata, Ydata,
test_size=0.2, random_state=42)
Y_train8 = np_utils.to_categorical(y_train8, 17)
Y_test8 = np_utils.to_categorical(y_test8, 17)
# first input model
visible1 = Input(shape=(60, 60, 3))
conv11 = Conv2D(28, kernel_size=4, activation='relu')(visible1)
pool11 = MaxPooling2D(pool_size=(2, 2))(conv11)
conv12 = Conv2D(14, kernel_size=4, activation='relu')(pool11)
pool12 = MaxPooling2D(pool_size=(2, 2))(conv12)
flat1 = Flatten()(pool12)
# second input model
visible2 = Input(shape=(60, 60, 3))
conv21 = Conv2D(28, kernel_size=4, activation='relu')(visible2)
```

```python
pool21 = MaxPooling2D(pool_size=(2, 2))(conv21)
conv22 = Conv2D(14, kernel_size=4, activation='relu')(pool21)
pool22 = MaxPooling2D(pool_size=(2, 2))(conv22)
flat2 = Flatten()(pool22)
# third input model
visible3 = Input(shape=(60, 60, 3))
conv31 = Conv2D(28, kernel_size=4, activation='relu')(visible3)
pool31 = MaxPooling2D(pool_size=(2, 2))(conv31)
conv32 = Conv2D(14, kernel_size=4, activation='relu')(pool31)
pool32 = MaxPooling2D(pool_size=(2, 2))(conv32)
flat3 = Flatten()(pool32)
# forth input model
visible4 = Input(shape=(60, 60, 3))
conv41 = Conv2D(28, kernel_size=4, activation='relu')(visible4)
pool41 = MaxPooling2D(pool_size=(2, 2))(conv41)
conv42 = Conv2D(14, kernel_size=4, activation='relu')(pool41)
pool42 = MaxPooling2D(pool_size=(2, 2))(conv42)
flat4 = Flatten()(pool42)
# 17th input model
visible5 = Input(shape=(60, 60, 3))
conv51 = Conv2D(28, kernel_size=4, activation='relu')(visible5)
pool51 = MaxPooling2D(pool_size=(2, 2))(conv51)
conv52 = Conv2D(14, kernel_size=4, activation='relu')(pool51)
pool52 = MaxPooling2D(pool_size=(2, 2))(conv52)
flat5 = Flatten()(pool52)
# 6th input model
visible6 = Input(shape=(60, 60, 3))
conv61 = Conv2D(28, kernel_size=4, activation='relu')(visible6)
pool61 = MaxPooling2D(pool_size=(2, 2))(conv61)
conv62 = Conv2D(14, kernel_size=4, activation='relu')(pool61)
pool62 = MaxPooling2D(pool_size=(2, 2))(conv62)
flat6 = Flatten()(pool62)
# 7th input model
visible7 = Input(shape=(60, 60, 3))
conv71 = Conv2D(28, kernel_size=4, activation='relu')(visible7)
pool71 = MaxPooling2D(pool_size=(2, 2))(conv71)
conv72 = Conv2D(14, kernel_size=4, activation='relu')(pool71)
pool72 = MaxPooling2D(pool_size=(2, 2))(conv72)
flat7 = Flatten()(pool72)
# 8th input model
visible8 = Input(shape=(60, 60, 3))
conv81 = Conv2D(28, kernel_size=4, activation='relu')(visible8)
pool81 = MaxPooling2D(pool_size=(2, 2))(conv81)
conv82 = Conv2D(14, kernel_size=4, activation='relu')(pool81)
```

```python
pool82 = MaxPooling2D(pool_size=(2, 2))(conv82)
flat8 = Flatten()(pool82)
# merge input models
merge = concatenate([flat1, flat2, flat3, flat4, flat5, flat6, flat7,
flat8])
# interpretation model
hidden1 = Dense(544, activation='relu')(merge)
hidden2 = Dense(272, activation='relu')(hidden1)
hidden3 = Dense(136, activation='relu')(hidden2)
hidden4 = Dense(68, activation='relu')(hidden3)
hidden5 = Dense(34, activation='relu')(hidden4)
output1 = Dense(17, activation='softmax')(hidden5)
output2 = Dense(17, activation='softmax')(hidden5)
output3 = Dense(17, activation='softmax')(hidden5)
output4 = Dense(17, activation='softmax')(hidden5)
output5 = Dense(17, activation='softmax')(hidden5)
output6 = Dense(17, activation='softmax')(hidden5)
output7 = Dense(17, activation='softmax')(hidden5)
output8 = Dense(17, activation='softmax')(hidden5)
model = Model(inputs=[visible1, visible2, visible3, visible4,
visible5, visible6, visible7, visible8],
              outputs=[output1, output2, output3, output4, output5,
output6, output7, output8])
# summarize layers
print(model.summary())
# plot graph
model.compile(loss='categorical_crossentropy', optimizer='adam',
metrics=['accuracy'])
print(model.metrics_names)
A = []
A0 = []
A1 = []
A2 = []
A3 = []
A4 = []
A5 = []
A6 = []
A7 = []
A8 = []
A9 = []
A10 = []
A11 = []
A12 = []
A13 = []
```

```
A14 = []
A15 = []
A16 = []
A17 = []
A18 = []
A19 = []
A20 = []
A21 = []
A22 = []
A23 = []
A24 = []
C = []
C1 = []
C2 = []
C3 = []
C4 = []
C5 = []
C6 = []
C7 = []
C8 = []
C9 = []
C10 = []
C11 = []
C12 = []
C13 = []
C14 = []
C15 = []
C16 = []
C17 = []
C18 = []
C19 = []
C20 = []
C21 = []
C22 = []
C23 = []
C24 = []
D = []
for i in range(cycle):
    a = model.fit([X_train1, X_train2, X_train3, X_train4, X_train5,
X_train6, X_train7, X_train8],
                  [Y_train1, Y_train2, Y_train3, Y_train4, Y_train5,
Y_train6, Y_train7, Y_train8],
                  batch_size=40, epochs=1, verbose=1)
    # print(a.history['dense_5_loss'])
```

285

```python
    # print(a.history['dense_5_accuracy'])
    # b = abs(float(a.history['dense_5_loss'][0]))
    # c = abs(float(a.history['dense_5_accuracy'][0]))
    b = abs(float(a.history['loss'][0]))
    b1 = abs(float(a.history['dense_5_loss'][0]))
    b2 = abs(float(a.history['dense_6_loss'][0]))
    b3 = abs(float(a.history['dense_7_loss'][0]))
    b4 = abs(float(a.history['dense_8_loss'][0]))
    b5 = abs(float(a.history['dense_9_loss'][0]))
    b6 = abs(float(a.history['dense_10_loss'][0]))
    b7 = abs(float(a.history['dense_11_loss'][0]))
    b8 = abs(float(a.history['dense_12_loss'][0]))

    c1 = abs(float(a.history['dense_5_accuracy'][0]))
    c2 = abs(float(a.history['dense_6_accuracy'][0]))
    c3 = abs(float(a.history['dense_7_accuracy'][0]))
    c4 = abs(float(a.history['dense_8_accuracy'][0]))
    c5 = abs(float(a.history['dense_9_accuracy'][0]))
    c6 = abs(float(a.history['dense_10_accuracy'][0]))
    c7 = abs(float(a.history['dense_11_accuracy'][0]))
    c8 = abs(float(a.history['dense_12_accuracy'][0]))

    A.append(b)
    A1.append(b1)
    A2.append(b2)
    A3.append(b3)
    A4.append(b4)
    A5.append(b5)
    A6.append(b6)
    A7.append(b7)
    A8.append(b8)

    C1.append(c1)
    C2.append(c2)
    C3.append(c3)
    C4.append(c4)
    C5.append(c5)
    C6.append(c6)
    C7.append(c7)
    C8.append(c8)

A0 = np.array(A1)
C = np.array(C1)
B = np.array(range(len(A0)))
```

```python
D = [A, A1, A2, A3, A4, A5, A6, A7, A8,
     C1, C2, C3, C4, C5, C6, C7, C8]
np.savetxt("history.txt", D, delimiter=',')


plt.plot(B, A0, 'o-', label='loss')
plt.plot(B, C, 'o-', label='accuracy')
plt.title('model accuracy')
plt.xlabel('epoch')
plt.ylabel('accuracy')
plt.legend(loc='best')
# save the model
plot_model(model, to_file='multiple_inputs.png')
model.save('final_model.h5')
print("model saved")
print('waiting for model save')
time.sleep(60)
print('model save complete')
# load the model, test it with test dataset
# evaluate the model
model = load_model('final_model.h5')
print("Loaded model from disk")
predict = model.predict([X_test1, X_test2, X_test3, X_test4, X_test5,
X_test6, X_test7, X_test8])
l0 = []
for m in range(0, len(predict)):
    l1 = predict[m]
    l1 = l1.tolist()
    # print(l1)
    for i in range(0, len(l1)):
        # print(l1[i])
        l0.append(l1[i].index(max(l1[i])))
np.set_printoptions(threshold=np.inf)
pd.set_option('display.width', 300)  #
pd.set_option('display.max_rows', None)  #
pd.set_option('display.max_columns', None)  #
l0 = np.array(l0)
l2 = np.array([y_test1, y_test2, y_test3, y_test4, y_test5, y_test6,
y_test7, y_test8])
l3 = []
for m in range(0, len(l2)):
    for i in l2[m]:
        l3.append(i)
l3 = np.array(l3)
accuracy = metrics.accuracy_score(l3, l0)
```

```python
Accuracy = [accuracy]
print("Accuracy of model using test data:", Accuracy)
np.savetxt("Accuracy of model using test data.txt", Accuracy)
print(plt.axis([0, cycle, 0, 1.9]))
plt.show()
```