Technische Universität Berlin

DAI-Labor
TU Berlin

Acoustic Self-Localization for Autonomous Soccer Robots

Masterarbeit

am Fachgebiet Agententechnologien in betrieblichen Anwendungen und der Telekommunikation (AOT)

Prof. Dr.-Ing. habil. Sahin Albayrak

Fakultät IV Elektrotechnik und Informatik

Technische Universität Berlin

written by

Wei Yang

Supervisor: Dr. Xu Yuan

Prof. Dr.-Ing. habil. Sahin Albayrak

Wei Yang
Register number: 352304
Fredericiastr. 3
14059 Berlin

Declaration

I, Wei Yang, declare that this thesis, Acoustic Self-Localization for Autonomous Soccer Robots and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Place, Date Signed

Abstract

An efficient yet precise position estimation result is always desired. Localization acts as key stone in many services and systems. The knowledge of current position affecting decision-making progress means the origin of everything. Not to human in daily life, also to the NAO humanoid robot in the worldwide competition RoboCup. Most implemented localization approaches in NAO robot are based on vision which is restricted by the site light circumstance also requests high computation power for successive image processing.

In this thesis, an acoustic approach for self-localization and communication is designed and implemented aiming at NAO humanoid robot under a real yet noise environment, i.e. RoboCup Soccer. This work contains two major parts. First actualising an audio communication system using Frequency-shift Keying. The NAO robots exchange position information using AFSK through the microphones and loudspeakers equipped then decode the position information embedded in the AFSK signal by using appropriate (Digital Signal Processing) DSP processes. The second part is using decoded position information and BeepBeep ranging method to estimate the location of robot that is calculated by Time Differences of Arrival (TDoA) algorithm. The orientation of robot is given by an audio sample measuring method. The implemented software solution for localization on NAO robots can provide loose localization within 1.5 meter error range for 6 robots within 1 minute.

Acknowledgements

As a master student, I'd like to express my gratefulness to my supervisor, Dr. Yuan Xu. He is a man of great wisdom who alway enlightens me when I get lost on the avenue of science, he is also a kind yet modest friend who alway gives me a hand who I need help when I climb the mount of science. With his company and guide the process of accomplishment of this master thesis is so joyful and splendid. I am also deeply grateful to esteemed team members Martin Berger and Erdene-Ochir Tuguldur, who are so friendly yet warm-hearted and help me reservelessly, advising and supporting.

Then I'd like to thank to myself, who is a perfectionist and put hard demands on herself and to my parents who raised me and supported me to finish my study abroad.

I do appreciate everything done by Mr. Dr. Cheng Li, whose love is endless and unconditional for me, who is standing by me all time encouraging and cheering me up.

I also want to express my huge gratefulness to all those people who devoted or are devoting their lives to science with making great contributions in every field. Especially to those scientists whose work and theory help me to solve problems and accomplish this thesis.

Contents

Chapter 1

Introduction

Robot Soccer World Cup, as known as RoboCup, is an annual international robotics competition conducted since 1997. Its goal is to foster research and development of robotics and Artificial Intelligence (AI), by tendering a public appearing but challenging competition. One of the subcategory of RoboCup aims soccer that all the teams use humanoid robot NAO, that operates fully automatically, e.g. motion, vision, behaviour, self-localization etc, during competition of RobotCup.

1.1 Motivation

Localization technologies as a comprehensive existence saturates our daily life. It is not only meaningful to human, but also plays an important role for robots as a fundamental in robotics. It has been broadly deployed in various applications. Position information directly impacts the behaviour and decision-making of robots when playing. The robot needs to be aware of where it is and where it should go to. By so far, most localization and navigation applications for NAO robots are the vision-based approach, such as [1, 2, 3, 4] as same as [5], however it is apparently a restricted solution per se.

A crucial drawback of vision based localization systems is that they cannot be used in many situations or environments due to the frequent blockage of the light by different obstacles and structures. In the RobCup case, the vision based solution for localization requires intensively computation due to the quality of images generated by embedded cameras. Otherwise using cameras for robot self-localization the ambiguity by image processing always occurs. Hence the capacity of the robot is limited often and the reliability can not be assured.

If the robots are able self-localizing via acoustic method through the air, or a self-localization function works like an aid compensating vision based localization, that dur-

ing a RoboCup soccer game the acoustic communication provides edge to the team as the data transfer via WLAN is limited. If so, the robot could behave better and make more reliable decision during the soccer competition via handing out a reasonable location. Otherwise the audio communication method can reduce the burden of data traffic on WLAN for information sharing during competition.

The Austrian Kangaroos, a humanoid soccer robot team of Vienna University of Technology and University of Applied Sciences Vienna, tackled this problem in an out-of-box way that the robots could talk with each other by taking advantage of digital acoustic communication [6]. They state that "experiments at laboratory conditions show good robustness even with presence of music and chatting students" in their report, which could partially prove that data exchange through acoustics is possible even in relatively harsh environment.

Acoustic positioning and communication have been applied, particularly for submarine purposes, and have a long-standing history, such as the LORAN [7] in World War II. Acoustic localization is more reliable and its estimation is more precise than others approaches for indoor, for example the Cricket Tracker System [8] and the HX-Series Positioning System from Hexamite.

However an acoustic localization system is able to control precisely the hardware, sending and receiving, they are usually commodity hardware aiming clear purposes. On NAO robot for acoustic experiments and Human-Robot interaction and communication researching, such as [9, 10], the researching results point out that the performance is limited by the hardware.

1.2 NAO Humanoid Robot

In [11] a NAO robot is equipped with a stereo broadcast system which is made up of 2 loudspeakers as shown in Figure 1.1. They, as sound transmitters, are responsible for generating sound. The left one locates at $(0.0038m, 0.0453m, 0.0526m)$, the right one is symmetric about y-axis and at $(0.0038m, -0.0453m, 0.0526m)$ [12], the origin is located at the neck gear of the NAO robot.

The NAO robot has 4 microphones, MicroFront and MicroRear, MicroRight and MicroLeft, as shown in Figure 1.2, their locations are in Table 1.1 respectively. The RoboCup edition comes with 2 microphones, MicroFront and MicroRear.

The electrical bandpass of the microphone's ways are from 300Hz to 8kHz.

The specification of [13] in NAO robot for V5 and V4 is:

- ATOM Z530 1.6 GHz CPU

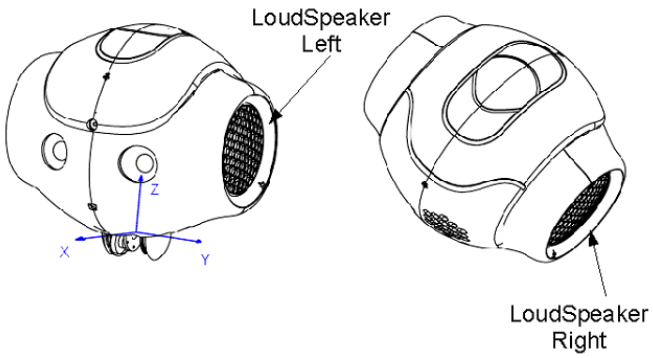

Figure 1.1: The locations of speackers in NAO humanoid robot V4.

Micro Name	X(m)	Y(M)	Z(m)
MicroFront	0.0489	0.0	0.076
MicroRear	-0.046	0.0	0.0814
MicroRight	-0.0195	-0.0606	0.0331
MicroLeft	-0.0195	0.0606	0.0331

Table 1.1: The positions of microphones in NAO robot

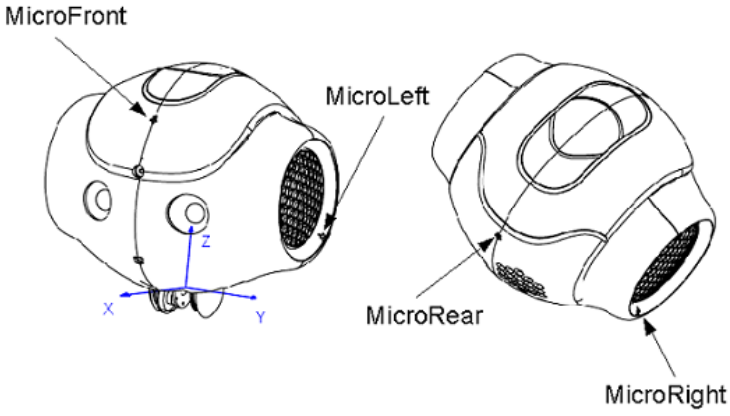

Figure 1.2: The locations of microphones in NAO humanoid robot V4

- 1 GB RAM
- 2 GB Flash memory
- 8 GB Micro SDHC

1.3 Problem Statement and Goals

Even sound localization- and communication systems have emerged for decades, but they are still isolated.

Particularly for humanoid robot a robust self-localization and communication based on audio is way more rarer, many trials are trapped in laboratory environment. Otherwise for implementing an accurate localization system or a suitable yet feasible localization algorithms are definitely a kernel component existing and supporting many services in many companies as the major pillar among others. Certainly those companies, who handle such as outdoor localization, are facing more complicated environments, e.g. Non-Line-of-Sight, multiple-path etc. However, for the humanoid soccer robot, the requirements for localization are loose and the environment conditions by RoboCup competition are geometrically tedious.

1.3.1 Problem Statement

- Audio Communication Process

  The humanoid robots communicate with each other using Audio Frequency-shift Keying (AFSK) to inform their present positions, when many audiences are standing on the outside of competition field and exchanging the thoughts on game while the host is commentating and the voice is amplified through a lot of loudspeakers, which is apparently more austere than the experiment in laboratory.

- Sound Localization Process

  The geometric relationship for humanoid robot self-localization during gaming stay mostly in 2D, since when game goes on, merely robots are on the competition field, therefore the Line-of-Sight (LOS) happens mostly by sound propagating. But if when a robot fells on ground and the coach robot sits on the table at the competition field margin, then the self-localization transforms from previously 2D into 3D. Besides, due to the limitations on the hardware in NAO robot and the complexity of Digital Signal Process (DSP) procedures and localization

calculation algorithms, the actualization of a real-time self-localization based on audio on NAO robot is difficult to achieved effortlessly, if the whole process costs too much time, then the result is useless, though accurate. Reaching a balance of complexity between preprocessing speed and localization accuracy level is also an important component.

1.3.2 Goals

The objectives of this master thesis are twofold: first, actualizing an audio based communication system for NAO robot for position exchanging. Second, implementing the acoustic localization system for position estimation. The characteristics of the to be accomplished acoustic self-localization for autonomous soccer robots will meet the following criteria.

- Robustness

  The audio communication for information exchanging among NAO robots is robust for differently environmental conditions, especially when the audio environment is varying and becoming complex.

- Loose Real-time

  A strict real-time system on NAO robot is very hard to actualized because of the limitation of hardware and complexity and computational requirements of Digital Signal Processing (DSP) procedures. Hence, a relatively loose real-time localization system is acceptable, but the potential delays have to be well handled as much as possible.

- Fully autonomous

  The robots are aware of the beginning and ending of audio self-localization process, acting spontaneously.

- Accurate localization

  The acoustic self-localization estimation result needs to be more accurate than vision based approaches and methods, excluding extra hardware in NAO robot.

1.4 Thesis Outlines

This master thesis is organised as following.

Section 2 introduces relative technologies and methods.

Section 3 provides an overview of work processes of acoustic self-localization on NAO, and describes each component.

Section 4 explores possible solutions for constructing the acoustic self-localization, determines the key process components.

Section 5 discusses the performances of acoustic self-localization on NAO about its accuracy, efficiency, drawbacks etc.

Section 6 summarises and possible future work.

Chapter 2

Related Technologies and Methods

This chapter gives an overview of related technologies involving in self-localization of autonomous soccer robots. The research focuses on tending reliabilities and accurate localization estimation involved in Audio Frequency-shift Key (AFSK) scheme, Digital Signal Processing (DSP) and localization algorithms. For increasing the possibility of decoding and preservation of desired information embedded in audio, a Forward Error Correction technique, the Hamming(7,4), is engaged to ensure correcting or detecting errors as much as possible in finite data length by receiving side, as well as the BeepBeep ranging method for an efficient yet accurate ranging estimation will be in this chapter presented.

2.1 Frequency-shift Keying (FSK) Scheme

Frequency-shift Keying (FSK) is one of the most common digital communication methods in our daily life, such as high-frequency radio spectrum and telephone. The information is insetted in frequency, by changing or shifting (increasing and decreasing) frequencies to communicate.

No doubt that Speech Synthesis and Speech Recognition technologies are close to human. Such as Apple Sire and Windows Cortana, they could understand nature human languages and their response is approximately or kind of like human communication most time which needs supporting by huge data warehouse to provide this service.

But considering the limited CUP capacity and storage memory in NAO robot with restricted data transferring during RoboCup per UDP within LAN, so a relatively high efficient AFSK communication seems appropriate than a semantic communication approach.

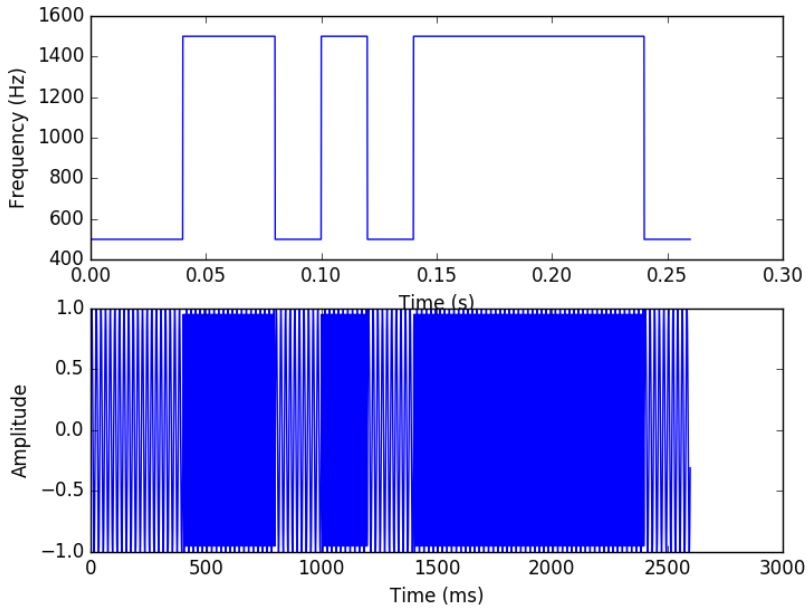

Figure 2.1: Binary Frequency-shift Keying example for a combination of 1100101000001

1. Binary FSK

   The simplest one among FSK scheme is Binary FSK (BFSK), the information is transmitted through one or two discrete frequencies to represent binary 1 and 0. The element length, i.e. the duration of sound ton for a single binary is between 5 and 22 milliseconds regularly that means, in 1 second 200 binary bits can be transported maximally, but there are also many exceptions, e.g. less than 1 microsecond or greater than 1 second, for a high resolution channel the element length to be greater than 0.5 millisecond is always expected [14].

2. Audio FSK

   Audio FSK (AFSK) uses sound wave as medium encoding digital data in audio tone. The transmission of AFSK is not capable for high speed data communication, because sound propagation in air is easily to be interfered by many elements and the performances are disturbed by quality of low-layer hardware in traducers, or by minor voltage changing in energy supplier, as well as depending on the software drivers and so on.

p1	p2	d3	p4	d5	d6	d7
p1	p2	1	p4	1	0	0

Table 2.1: Hamming(7,4) table for encoding

p1	p2	d3	p4	d5	d6	d7
0	1	1	1	1	0	0

Table 2.2: Hamming(7,4) table for encoding (cont.)

2.1.1 FSK Demodulator

The demodulation methods for FSK can be divided into two major categories. FM Detector-type demodulator and filter-type demodulator such as Goertzel algorithm [15]. The Implementation of this master thesis will put the emphasis on FM Detector-type demodulator. In [14] the FM Detector-type demodulator has been very comprehensively elaborated in an electronic point of view. In this paper, the process will not be introduced redundantly, but focusing on implementation in programmatic standpoint.

2.2 Forward Error Correction – Hamming(7,4)

Hamming(7,4), one of the Hamming Code, which "can detect up to two-bit errors or correct one-bit errors without detection of uncorrected errors" [1] in seven bits.

2.2.1 Hamming(7,4) Production

Assuming there is a four bits code 1100 that needs transporting. According to Hamming(7,4) the four bits locate in positions $d3$, $d5$, $d6$ and $d7$. Then adding three additional check bits into the message positioning at $p1$, $p2$, $p3$ which are also called parity bits illustrated in Table 2.1.

Hamming says that $p1$ is the parity bit for $d3$, $d5$ and $d7$ responsible, which are $(1, 1, 0)$, and $(1 + 1 + 0)\ |_2 = 0$ is even, so $p1 = 0$. Furthermore, $p2$ for $d3$, $d6$, $d7$, and $p4$ for $d5$, $d6$, $d7$. Consequently, $p1 = 0$, $p2 = 1$ and $p4 = 1$, so the seven-bits Hamming(7,4) code based on the sequence of 1100 is 0111100 as in Table 2.2.

[1] https://en.wikipedia.org/wiki/Hamming_code

p1	p2	d3	p4	d5	d6	d7
0	1	1	1	1	1	0

Table 2.3: Hamming(7,4) table for error detection

2.2.2 Hamming(7,4) Error Detection

In the error detection and correction process, first assuming the seven-bits Hamming(7,4) code 0111100 which is encoded by even parity in Table 2.2, go through a noise channel, neutral to 1 and 0. The received code is transmitted as 0111110 showing in Table 2.3, one error occurs at position $d6$.

The knowledge of $p1$ locates at $d3$, $d5$ and $d7$, then the parity check equation of $(p1, d3, d5, d7)$ is in equation 2.1.

$$(p1, d3, d5, d7) \mid_2 = (0, 1, 1, 0) \mid_2 = 0 \tag{2.1}$$

So $p1 = 0$ is correct. Likewise, which has been elaborated in the previous section, hence $p2 = 1$ and $p4 = 1$. Then a string from the sequence of $(p4p2p1)$ is (110). Hamming says that this particular string is in binary format to inform about the location of error bit which needs converting to decimal. So (110) in binary is equal to 6 in decimal. Then the position of error bit is sixth in the sequence, i.e. $d6$ in the Table 2.3.

2.2.3 Hamming(7,4) Limitation

Hamming(7,4) is an extraordinary method to detect or correct error in data piece. However, it has also limitations that it is able to either correct error one-bit error, nor detect one-bit or two-bit error, which means for instance in Hamming(7,4) that Hamming can merely handle less than or equal to two errors in 7 bits. If there are more than two error bits occurring, the Hamming(7,4) becomes worthless. Otherwise, it can not distinguish between single-bit errors and two-bit errors, so it will treat them identically, i.e. it will think that there is only one-bit error which is going to result incorrect decoding result in two-bit errors case.

2.3 Digital Signal Processing

Waves, like light, sound or heat, are invisible, but they can be felt, heard, or seen in some way. The preservation of a wave or a signal is a prerequisite before it is visualized for

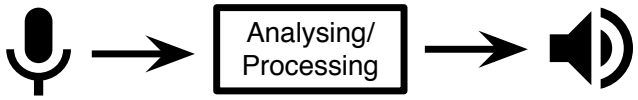

Figure 2.2: Analog-to-Digital conversion

such as analysing and researching purpose. Transducers are an important role converting the signal from a form to another which make the preservation process possible.

2.3.1 Analog-to-Digital Converting Process

Microphones and speakers are common transducers in audio context, they transduce, convert and transfer the sound wave.

Taking advantage of Pulse-code modulation (PCM) a sound wave through transducer can be represented in digital manner, an audio signal.

Under audio or voice generating, sampling and processing context, sampling frequency stands for the amount of samples per second, commonly from $300Hz$ to $348kHz$. The higher sampling frequency is, the better quality sound has, horizontally. Bit depth is the resolution of each sound sample vertically, which is commonly met at 8-, 16-, and 24-bit depth. Often the bit depth shows up with its data format together, e.g. integer, float, that is how the hardware processes and stores the input data of audio.

For instance, a piece of raw monotone with duration of 1 second is created in $11025Hz$ and has a 16-bit depth, then the amount of data volume is in the following calculated by equation 2.2 that represents a lower-quality PCM though.

$$\begin{aligned} duration \times samplingrate \times 2Bytes &= 1second \times 11025Hz \times 2Bytes \\ &= 22050Bytes \end{aligned} \tag{2.2}$$

In equation 2.2 monotone times 2 bytes for each sample, stereo times 4 bytes for each sample.

Aiming to this thesis, on the playing side a NAO robot plays the mono sound generated in $11025Hz$ that has a sample format of signed 16-bit in Little Endian byte order. On the other side one NAO robot records in this manner as well but in $44100Hz$ for avoiding Aliasing phenomenon due to Nyquist-Shannon Sampling Theorem.

2.3.2 Nyquist-Shannon Sampling Theorem

The most important theory of signal preservation or sampling is Nyquist-Shannon Sampling Theorem, also called Sampling Theorem. A continuous-time yet band-limited signal $x(t)$ and its equally spaced samples are in equation 2.3 where T is period.

$$x(nT), n = 0, \pm 1, \pm 2, \ldots \tag{2.3}$$

$$X(w) = 0 \tag{2.4}$$

and

$$|w| > w_{max} \tag{2.5}$$

The Fourier transform of this signal is zero in equation 2.4, where w_{max} is the highest frequency, which means out of some bands all are zero.

Then under the conditions, if the sampling frequency is higher than double original signal frequency, the original signal $x(t)$ is uniquely recoverable.

$$2\pi/T \doteq w_{sampling} > 2w_{max} \tag{2.6}$$

Sampling theorem is a mechanism for representing a band-limited yet continuous-time signal by a sequence of samples in time domain which is a discrete-time signal. It is also an approach for avoiding aliasing by sampling procedure, because if aliasing occurs then the original signal can not be restructured. Set in a simpler way, if a reconstruction of a signal is desired, for sampling it and reserving it a minimally twice higher sampling rate must be adopted. For instance, a sound is generated at sample rate of $11025Hz$, for a successful reconstruction it is sampled at least using a common $32000Hz$ rate that is bigger than $2 * 11025Hz$.

2.3.3 Filter Function and Butterworth Filters

Filter function is a significant component function in DSP, a well-designed filter can allow single or bunch of frequencies to pass through, while blocking or attenuating the other unwanted.There are many filter types, such as Low Pass Filter (LPF), which is only permitting low-frequency signal under a specific boundary to have the easement of the filter. High Pass Filter (HPS) blocks low-frequency signal than its cutoff frequency, as well as Band Pass Filter (BPF) that is capable to let signals within a certain frequency band or selected range which is named as Bandwidth fall in and through.

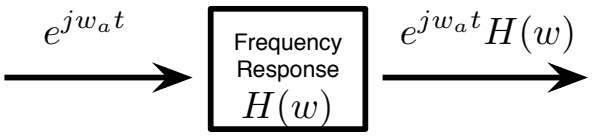

Figure 2.3: Generate Butterworth Filter Function. A frequency impulse e^{jw_at} as input imports to a frequency response $htb(w)$, the output is the convolution of them.

Butterworth filters are a relatively simple but also substantially useful filter class. Its response frequency is monotonic. An ideal Butterworth BPF can be formulated as the adjacency equation 2.7, N indicates the order.

$$|B(jw)|^2 = \frac{1}{1 + (jw/jw_{cutoff})^{2N}}. \tag{2.7}$$

In Figure 2.4 four obvious conclusions can be drew that

1. the bigger the N is, the sharper or steeper the filter edges are, and drop off more quickly and attenuates more in stop-bands,
2. the preserved frequency range is becoming narrow at the top of the filter curve,
3. all filters go through two same points,
4. high order filter, such as $order = 9$, becomes unstable and also requests high computation.

By design a Butterworth Band Filter the cutoff frequencies could be cut clean by rational high order Butterworth filter. In Figure 2.4, by increasing the order of filter, the edge is getting closer to the cutoff frequencies, which in this case are $1000Hz$ and $3000Hz$. But there is also a drawback as shown in Figure 2.4, when the order decreases, the passband is getting narrower in particular on the top of curve, then if broad stop bands are required, which implies that the accuracy on the pass band will lose, while the stop bands can be comprehensively covered at the same time. Assuming $1000Hz$ and $3000Hz$ are the corner frequencies, i.e. cutoff frequencies, so the frequencies in the range from $1000Hz$ up to $3000Hz$ remain which are in the bright area in Figure 2.5, others will be cut at the meanwhile.

In order to design a practical Butterworth filter which could maintain the specifications of high order Butterworth filter, i.e. the accuracy on the passband and the sharper cutoff frequencies, and also has the characteristics of low order Butterworth filter, e.g. broad stop bands, hence a transition area is introduced. The transition area works like

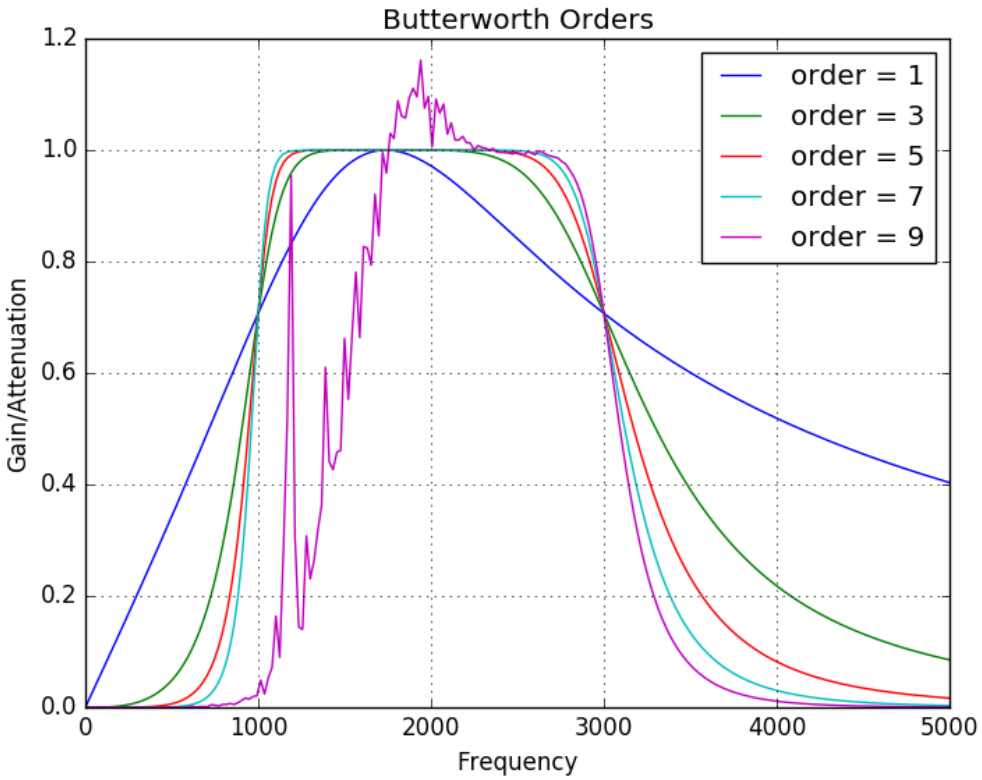

Figure 2.4: A Butterworth Band Pass Filter with $1000Hz$ and $3000Hz$ corner frequencies

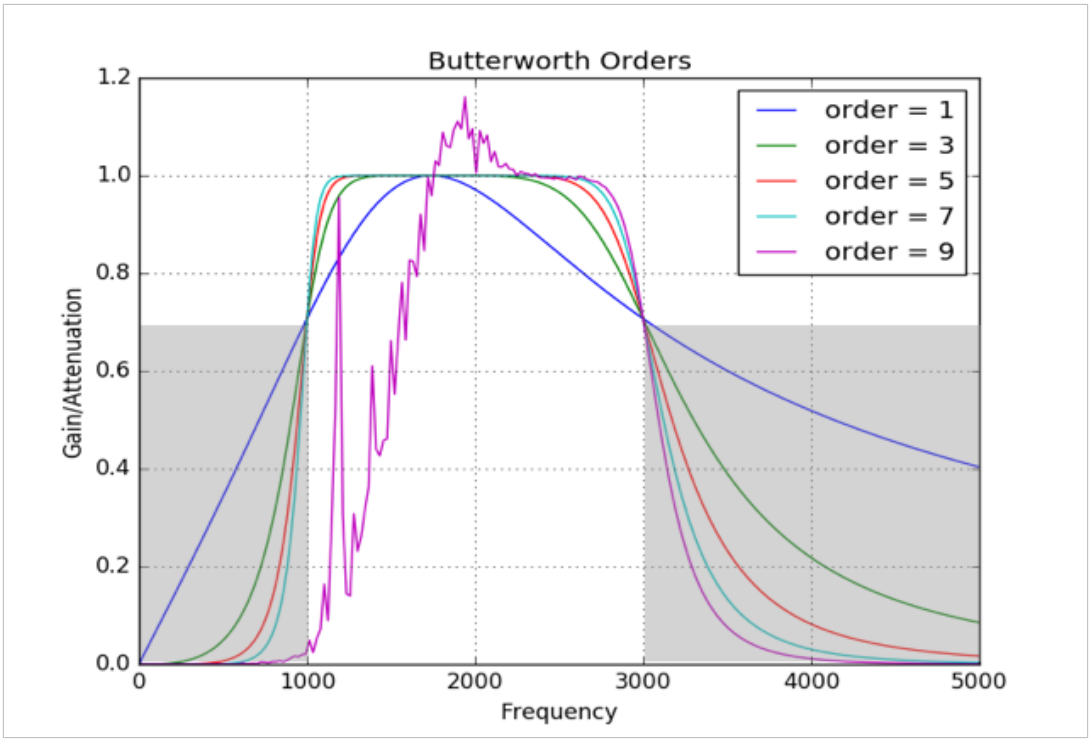

Figure 2.5: A Butterworth Band Pass Filter with $1000Hz$ and $3000Hz$ corner frequencies (cont.)

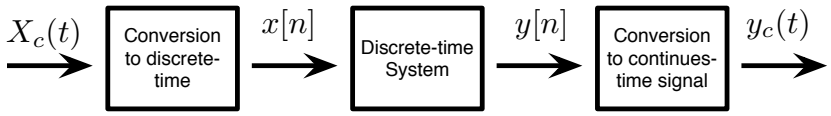

Figure 2.6: The processes of the construct of a continuous-time signal from a discrete-time signal [16]

a buffer between pass bands and stop bands. When $1000Hz$ and $3000Hz$ are the desired corner frequencies, then the stop bands are smaller than $1000Hz$ and bigger than $3000Hz$ respectively, the stop frequencies have to be a bit wider than pass bands.

2.3.4 Windows and Window Function

In the reconstruction process of a discrete-time signal and as well as in design for Finite Impulse Response (FIR) filters the window function is essential. The window handles signal in time domain, and the signal responds the window function in frequency domain.

A signal with finite duration on the time domain comes out after sampling procedure, however Fourier Transform is a periodic function and valid from $-\infty$ to $+\infty$. The basic idea to analyze a discrete-time signal is that, a finite discrete-time signal acts as a periodic signal and repeats in at least Nyquist frequency to avoiding aliasing. Then it can be treated as an infinite signal and analyzed in frequency domain using Fourier Transform. The window transforms the finite signal into periodic, then the window function generates coefficients of a desired frequency response and evaluates, at last filters out unwanted, that makes observation and analysis of an infinite signal at a particular time spot possible. Figure 2.7 shows different window types.

Among them Hann Window [2] is a versatile window, so it is chosen for assisting with LPF for signal recovering. As the Signal-to-Noise Ratio (SNR) is high, the better filtered result is gained, so the window function will be deployed after Butterworth Filter when the sound environment is intricate.

2.4 BeepBeep Ranging Method

The first present of BeepBeep Ranging System [17] appeared in 2007 using very basic hardware, two commercial cell phones at that time, for audio-based measuring approach

[2] in Figure 2.7 Hann Window is also known for Hanning Window.

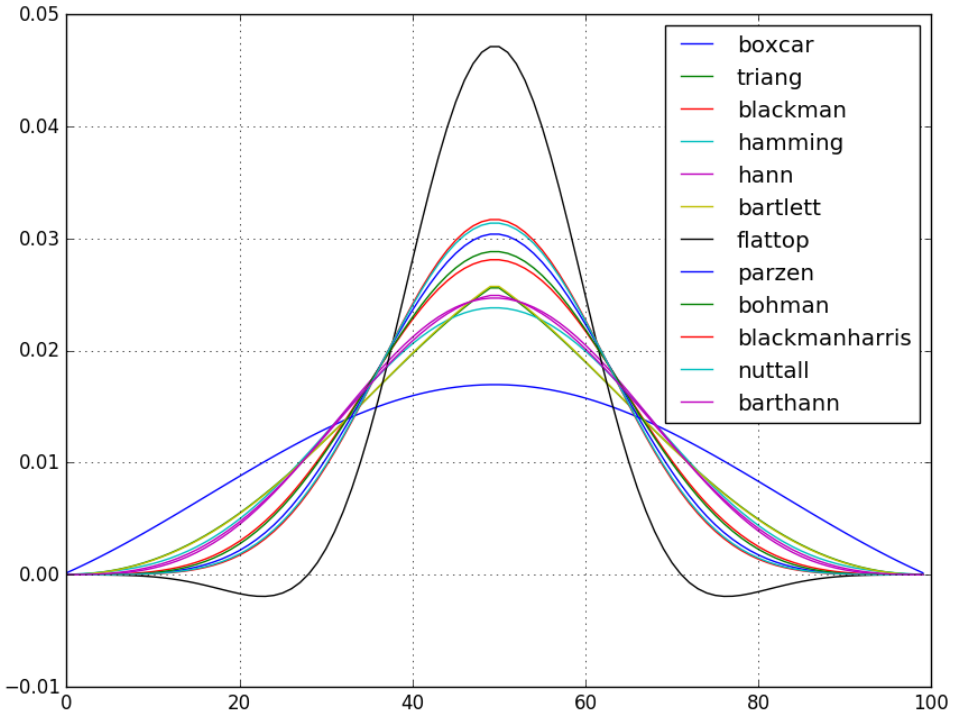

Figure 2.7: Different windows with 100 samples length

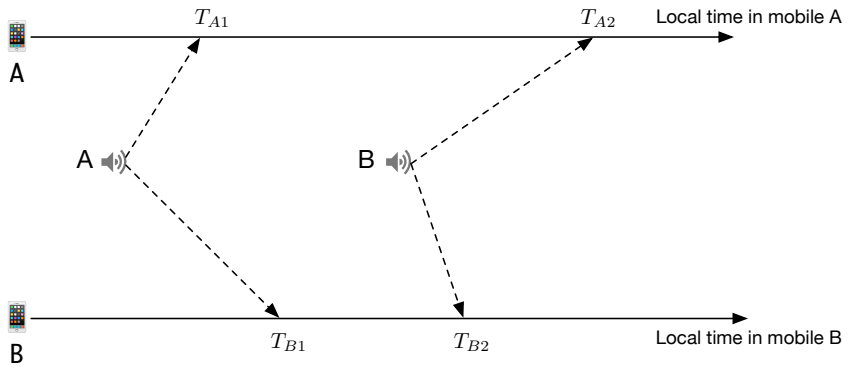

Figure 2.8: Event sequence of BeepBeep Ranging Method

to calculate the distance between them. This approach is not only isolated from hardware characteristics and also gives out high-accurate ranging result. It covers three techniques, two-way sensing, self- recording and sample counting. Two mobile phones play a specially designed sound piece with an obvious peak in it that refers as the "beep" sound, one after one. The arbitrary time span between two "beep"s can not affect the ranging accuracy at all.

The BeepBeep ranging process can be simplified as in Figure 2.8 that mobile phone A plays a "beep" and receives it at T_{A1} according to the internal time of mobile phone A, and after an arbitrary time elapsing mobile phone A receives another "beep" emitted from mobile phone B at local time T_{A2}. The mobile phone B receives a "beep" from A at time T_{B1}, and after a while it plays a "beep" sound and receives it at T_{B2}. Besides both have the same sample rate of R sample per second, the sound propagating speed is C meter per second. After exchanging the information about T, the distance between mobile phone A and B is computed by the equation 2.8 where K is the sum of the distances for both mobile phones from microphones to speaker.

$$D = C \times \frac{1}{2} \times ((T_{A2} - T_{A1}) - (T_{B2} - T_{B1}))/R + K \tag{2.8}$$

The most substantial advantage of this ranging method is that BeepBeep is totally independent of synchronisation, time stamp and all potential delays.

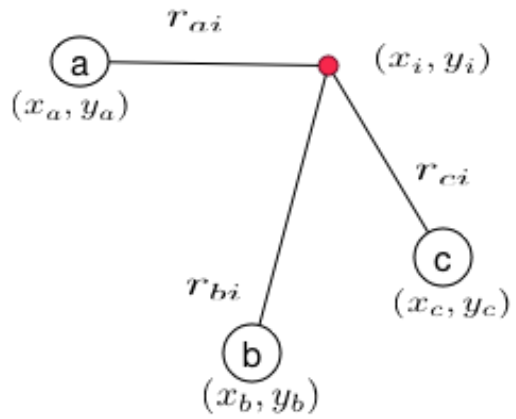

Figure 2.9: An illustration of TDoA model

2.5 Time Difference of Arrival (TDoA) Localization Algorithm

Time Difference of Arrival (TDoA) localization technique, also known as hyperbolic localization algorithm, is one of the most common localization techniques. Thanks to its high accuracy and facilely implementable method it has been broadly employed in military and marine affairs such as LORAN [7] in World War II. Nowadays it is also for commercial applications, thereof the benchmark localization solution, the world-wide navigation system Global Positioning System (GPS) using TDoA of signals from multiple synchronised satellite-based transmitters is the most well-known instance.

In humanoid robot GPS is a reachable solution as well, only a extra module can take all charges for localization task, while in RoboCup hardware modifying and changing in NAO robot are not allowed. Regarding to the limitations on the rule of SPL [18], an appropriate localization algorithm, Time Difference of Arrival (TDoA), for matching the default NAO robot hardware configurations and actualising the acoustic self-localization for RobotCup is presented.

2.5.1 General Time Difference of Arrival (TDoA) Algorithm Model

Assuming that an unknown node marked as i in colour red in Figure 2.9 is positioning at (x_i, y_i) in a 2D coordinates system. The node i transmits a signal which is received by multiple fixed nodes or vice versa, in this case three nodes stay steady, namely a, b and c. Figure 2.9 illustrates the TDoA model.

Due to time from node i to these three nodes a, b and c, the arriving time are separately known, t_{ai}, t_{bi} and t_{ci}. Besides a constant v is the signal propagation speed, so a formula derived from all conditions is 2.9.

$$\Delta t_{ab} = |t_{ai} - t_{bi}| = |r_{ai} - r_{bi}|/v \tag{2.9}$$

So:

$$\Delta t_{ab} \times v = \Delta r \tag{2.10}$$

Because:

$$r_{ai} = \sqrt{(x_a - x_i)^2 + (y_a - y_i)^2} \tag{2.11}$$

$$r_{bi} = \sqrt{(x_b - x_i)^2 + (y_b - y_i)^2} \tag{2.12}$$

This Δr gives the hyperbola which the node i must lie on, according to the hyperbolic function can then gain an expression 2.13.

$$\Delta r = \sqrt{(x_a - x_i)^2 + (y_a - y_i)^2} - \sqrt{(x_b - x_i)^2 + (y_b - y_i)^2}. \tag{2.13}$$

Due to equation 2.10 then the difference between a and b is calculable in equation 2.14.

$$\Delta t_{ab} = \frac{1}{v}[\sqrt{(x_a - x_i)^2 + (y_a - y_i)^2} - \sqrt{(x_b - x_i)^2 + (y_b - y_i)^2}] \tag{2.14}$$

Similarly, we have:

$$\Delta t_{bc} = \frac{1}{v}[\sqrt{(x_b - x_i)^2 + (y_b - y_i)^2} - \sqrt{(x_c - x_i)^2 + (y_c - y_i)^2}] \tag{2.15}$$

Now the x_i and y_i are calculable in combining 2.14 and 2.15.

2.5.2 Limitation of Time Differences of Arrival (TDoA)

Every algorithm has its limitations, TDoA is not an exception. These prerequisites in List 2.5.2 beneath have to be met before TDoA calculation for an unique localization estimation.

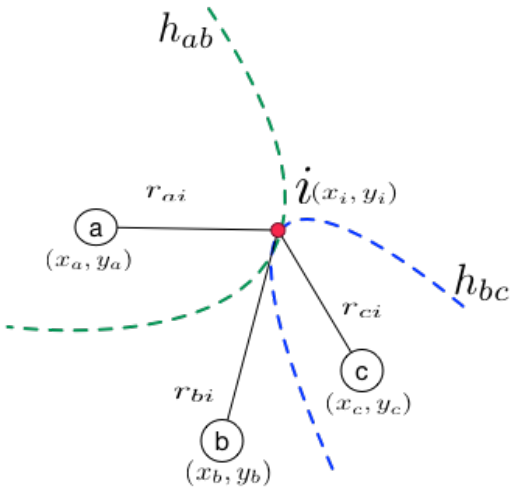

Figure 2.10: Hyperbola positioning

- 3 robots with known positions are required for 2D at least, whose relationship has not to be collinear,
- time synchronisation between signal receivers,
- at least one direct propagation path exists from unknown to the knowns [19], otherwise the refections or/and echoes arise a bias estimation result.

2.6 Chapter Summary

This chapter referrals mainly the correlative methods and techniques within the scope of this master thesis, which are in the list beneath.

- Introduced Binary Frequency-shift Keying (BFSK) and Audio Frequency-shift Keying (AFSK).
- Introduced Forward Error Correction Technique, provided an overview encoding process and error detection process of Hamming(7,4) Code.
- Elaborated how to sample an infinite yet band-limited signal, especially an audio signal and two common Digital Signal Processing functions, the Filter Function and Window Function.
- BeepBeep Ranging Method, the functional equation and the advatage of this method .

- Introduced the TDoA localization algorithm with the solving processes and listed the preconditions for a successful TDoA localization estimation.

Chapter 3

Design of Acoustic Self-localization

A robot broadcasts UDP packets embedded its ID number via WLAN, as the robots receive the UDP packet and begin reading audio raw data from steam immediately. The robots compose their positions in the AFSK signal, then the robots begin recording a specific number of audio samples and play AFSK in sequence. Those recorded audio samples are sliced into small pieces by sound detecting when the sound is loud enough contrasted with the background noise. Robot determines the peak of each slice of audio, demodulates the AFSK and decodes the information insetted in audio, afterwards sends its positions and the detected onset information per UDP package.

The major functions of audio localization in Figure 3.1 will be brought out in timeline in this chapter.

3.1 Audio Frequency-shift Keying (AFSK) Generation

Figure 3.2 illustrates how to generate a sound piece embedded location coordinates information using FSK. To increase the efficiency of information transmission with audio, utilising limited duration of the sound piece to send more data, as well as a high resolution are wished for RoboCup utility, thus using short information encoding approach is more reasonable. Conventionally an ASCII character is represented by seven-bit code, that is the most well-known character encoding standard in the world. But it is quite redundant for audio communication during RoboCup gaming, due to that it contains many symbols and distinguishing letter in lower case or upper case. Especially for actualising a robot self-localization system that there is no necessary to send many characters. Only few things we do care about for the position calculation, such as the position coordinates.

[1]ETOA: Elapse Time of Arrival [17]

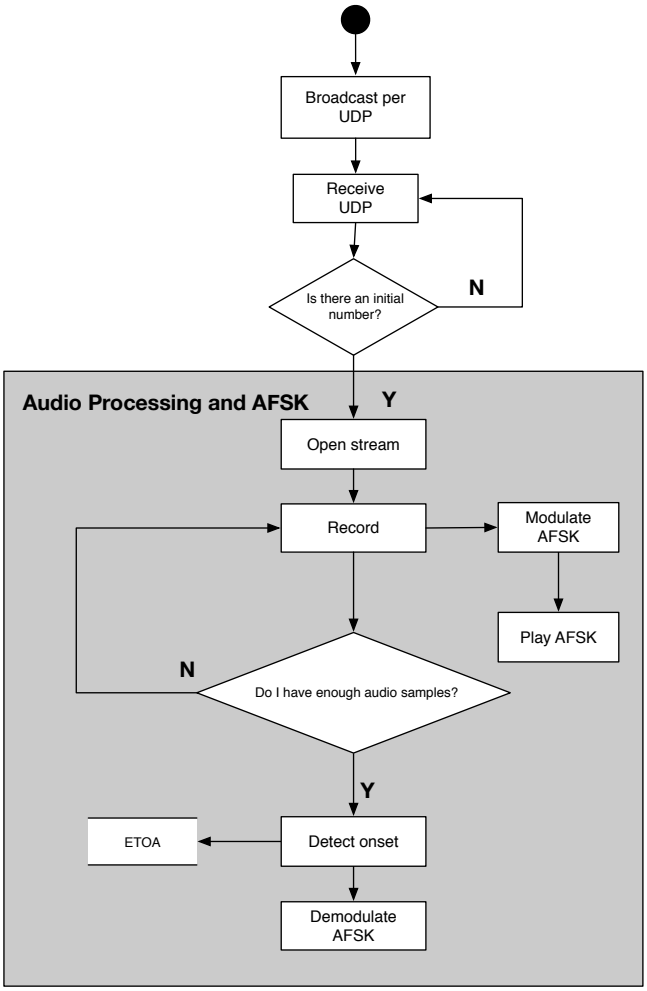

Figure 3.1: Processes of acoustic self-localization for NAO robot in time sequence[1]

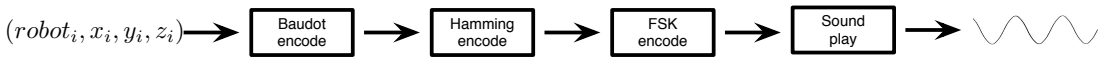

Figure 3.2: Audio FSK generating process

The information sent by AFSK is organised by the sequence of $(robot_i, x_i, y_i, z_i)$ in Figure 3.2. Since the error occurs in playing, propagating also in recording processes, the tail part information could not be wholly decoded sometime. Therefor more interested information should be assigned forward. If the AFSK is decoded faulty, a information exchanging per UDP package during the RoboCup is still possible. If z_i, the information about the height of robot according to different postures, such as lying, crunching, standing, is somehow undecode-able, which means that the information about height is missing, at least using the 2D position information a localization estimation is still feasible, although the accuracy is decreased when the microphones and loudspeakers in robots are not on a same plane.

3.1.1 Baudot Code

Baudot Code [20] is a fixed-length character encoding invented in 1870. It needs five-bit to represent a letter or a number, the most important is that there is no prepend and append. Whereas in Baudot Code the lower case and the upper case are not distinguishable, that means that Baudot Code is able to represent only one case, either uppercase or lowercase. It has two shift keys, Letter Shift and Figure Shift originally, and they are also treated as Space. For instance, when Letter Shift is activated, the following bits will be processed in Letter Domain, until the Figure Shift is met.

3.1.2 Audio Frequency-shift Keying (AFSK)

The microphones of NAO robot have a frequency range from $150Hz$ to $12kHz$ [21], the appropriate frequency range in AFSK should also belong to part of it. The nominal carrier frequency is "halfway between the mark and space frequencies" which is brought out in reference [14]. Carrier frequency is calculated in the equation 3.1.

$$f_{carrier} = (f_{mark} + f_{space}) \div 2 \tag{3.1}$$

Regards to the characteristics of sound and to pass most peoples' feelings in order to deploying this feature in RobCup competition obeying the rule book [18], $2000Hz$ as the carrier frequency is chosen, and the deviation is $1000Hz$, an element length of 20 milliseconds i.e. 50 Baud Rate is taken to guarantee a productive AFSK communication thusly.

3.2 Sound Onset Detection

A sound wave or a sinusoidal wave can be featured by four properties, they are frequency, amplitude, speed and direction mathematically. In reality, to sound wave detecting, the speed of sound is a certain constant in a loose medium. Besides to detect the direction of sound only if the density of microphones is high. Among them the easiest way to detect a sound wave is setting an appropriate threshold according to different detecting strategies and finding the pitch, such as in the frequency domain or amplitude or power changing in the time domain. Basically there are four roads to detecting the onset of sound.

- Calculating the Root Mean Square (RMS).

  In statistic science the RMS is generating a mean of a series number in small chunk to approximate the tendency of data. Under electronics or acoustics the RMS involves power and is a method for calculating the sound pressure level (SPL) or representing the voltage of a signal with the following formula 3.2.

$$x_{rms} = \sqrt{\frac{1}{n}(x_1^2 + x_2^2 + ... + x_n^2)}, n \in (x_1, x_2, ..., x_n). \tag{3.2}$$

- Calculating the peak value for sound amplitude or the Sound Pressure Level (SPL).

  For instance, a piece of signed 16-bit integer monophonic sound. On the recording side the signed 16 bit integer monophonic sound range is between $[-32768, +32767]$, so that the highest and the lowest number which can be represented in 16 bit monophonic sound are $+32767$ and -32768. The bigger the number is, the higher the power of pressure of sound is, hence the integer describes the amplitude of the sound or audio inherently. The normalised amplitude can be produced by the equation 3.3, that reveals the ratio of magnitude changing.

$$ratio = |sample| \div 32768 \times 100\% \tag{3.3}$$

  This method is also capable of setting environment noise level, acting as a basic threshold to detect the desired sound.

  The Figure 3.3 shows a sudden sound happening after a slice of silence, a series acute energy/power changing happening along the Y-axis.

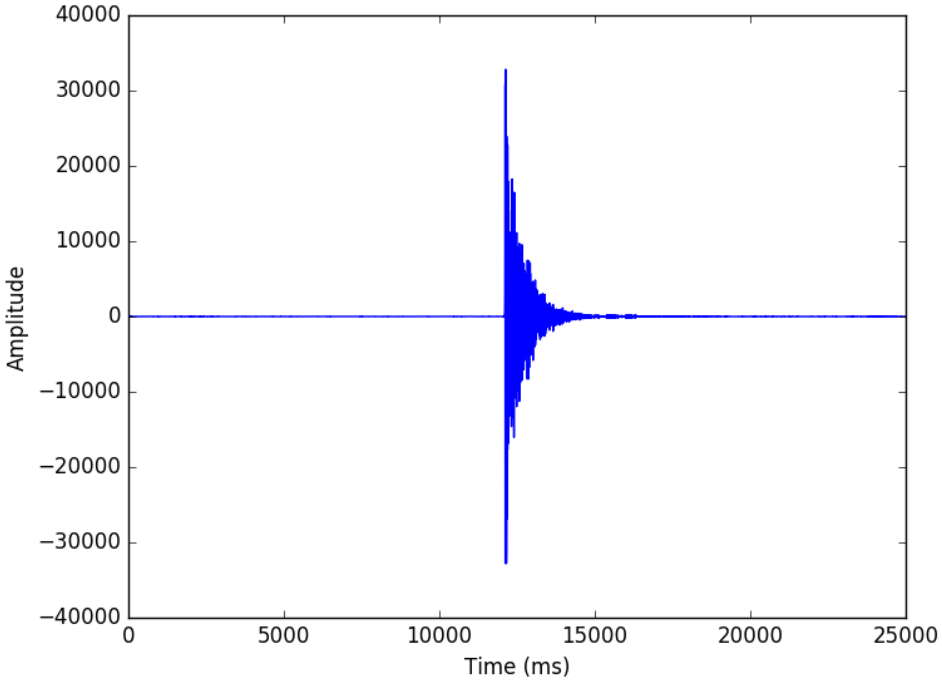

Figure 3.3: Plotting of a mono sound

- Calculating the frequency.

  In a noiseless environment condition or analysing an original soundtrack, an appropriate frequency threshold is exactly the sound frequency itself, which is ideal. Nevertheless the spectral leakage takes place in FFT process, that the wanted frequency is mixed with other lower or higher frequencies, so it shifts in reality. Hence the frequency detection prefers a range than a signal point.

- Calculating the cross-correlation for time shift of a signal.

  Measuring similarity of a signal to the other is an effective approach using cross-correlation when the detected signal is foreknown. The onset of known signal in an unknown series is located at their peak convolution. But this effective approach is inefficient with limited computation power.

As mentioned in the Chapter 1, an ATOM Z530 1.6 GHz CPU [13] planted in NAO robot has really limited capacity, and RMS requires relatively intensive computation for long audio data current as well as the Sound Pressure Level (SPL) calculation when the audio I/O is heavy. Thus, a peak detector for sound detection and cross-correlation

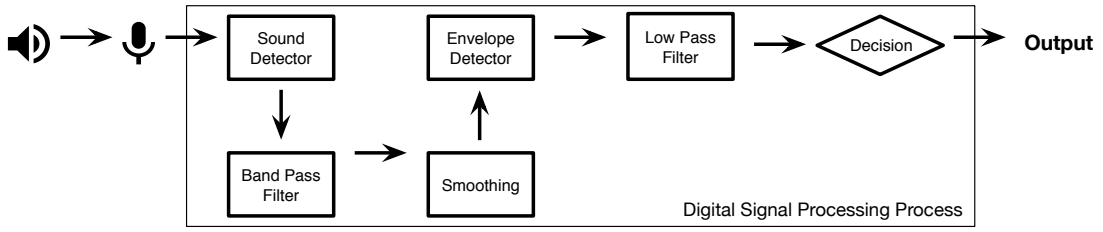

Figure 3.4: Digital Signal Processing

algorithm as a matched filter to calculate the real sound onset in the detected sound piece will be deployed.

3.3 Recording and FSK Demodulator

This section is going to present recording, signal reconstruction and actualising a FSK demodulator.

The NAO robot opens microphone streams and begins detecting sound after setting an amplitude threshold of environmental noise. As long as it hear a sound it starts recording. At the meanwhile all 6 robot play AFSK signal one by one.

- Sampling/Recording

  The main theoretic part of sampling theorem and PCM data format etc has been in Chapter 2 exhibited. The bit depth is the resolution of sound alike, the sample rate represents the quality of sound. The sample rate on the recording side needs to be higher than twice of sample rate on the playing side, which ensures no aliasing phenomenon.

  Those upper lines reach a conclusion that if on NAO the playing side plays the signal in sample format of signed 16 bit mono sound in Little Endian and $11025Hz$, on the other side one NAO robot records in this manner as well but in higher sample rate. In the thesis the AFSK signal is played at $11025Hz$, on the recording side the recorded AFSK signal is sampled at $44100Hz$.

- Butterworth Filters

  In previous section 2.3.3 the main features of Butterworth BPF have been broadly filed. In the Figure 2.5 all the filters in different orders pass through two same points, which make two orthogonal lines to the $X - axis$ standing right on the

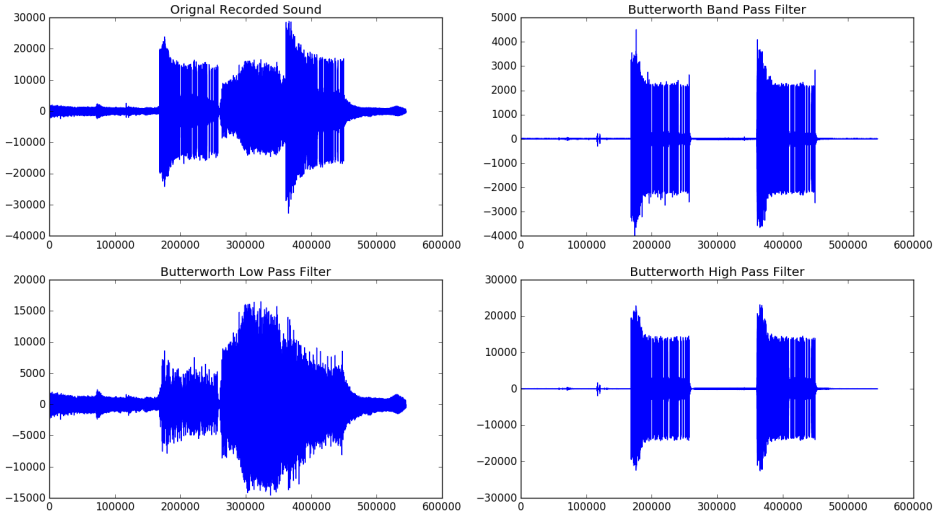

Figure 3.5: Testing results of different Butterworth Filers under complicated sound environment

corner frequency points. It can be determined that the two special points are affecting immediately the performances of Butterworth filters in essence.

For an ideal Butterworth BPF with the cutoff frequencies $1000Hz$ and $3000Hz$ and a sample rate of $44100Hz$ the two dots which refer the maximum loss in the pass bands (dB) and the minimum attenuation in the stop bands respectively, can be calculated by the next two equations 3.4 and 3.5.

$$\delta_{passband}(dB) \cong 20\log_{10}(1+2\omega), \omega = 2\pi f_{passband} \tag{3.4}$$

$$\delta_{stopband}(dB) \cong -20\log_{10}(2\omega), \omega = 2\pi f_{stopband} \tag{3.5}$$

Not only the BPF, also the HPF and LPF are profitable depending on the environment. The testing results of Butterworth filters under a real yet noisy condition are shown in the Figure 3.5, the original signal frequencies are $1000Hz$ and $3000Hz$, playing in $11025Hz$ and recorded in $44100Hz$.

In the right column in Figure 3.5 comparing to the original sound locating on the left upper corner, the filtered signal shows very clear edge and intervals. But the plotting lying on the left bottom shows differently, it has thorny waveform and

the noise apparently is magnified. Therefore, different filters perform differently depending on the noise types.

- Smoothing

  Smoothing is another regular noise reduction process. Many smoothing functions are available that can filter out environment noise or other rapid phenomena, e.g. minor distorted sound. The most common algorithms, for example moving average calculating the average value of successive values and LPF. In this thesis the Savitzky-Golay filter [22] is engaged which ought to hand out more reasonable result in DSP without making big damage in signal for increasing the SNR.

- Reconstruction

  This part could be split into three subprocesses. First, differentiating and then envelope detecting, finally a LPF with window function.

  Differentiating can construct a signal which approximates the waveform of the original in the time domain. If the SNR is high, after differentiating a rectangle waveform in the frequency domain can be obtained.

  In the envelope detecting process the Hilbert Transform is usually engaged in order to detect the signal envelope, in another word, the main goal of this process is aiming to draw the upper outline of the amplitude of the sampled signal in the time domain, and it only responds to the positive part.

  In the FSK demodulating or signal recovering the final step is a LPF to recover or reconstruct the original signal, which is designed to correspond to the original signal's baud rate to remove noise at frequency domain.

3.4 Localization

According to SPL Rules [18], each team is allowed maximally 6 players, which means that ideally, there are 5 known positions available for single unknown, including a coach robot seating on a table locating out of the soccer field.

For a successful localization by involving TDoA algorithm, 3 known positions for planar and 4 positions for stereo are required which is also a prerequisite. Otherwise, during gaming, the robots could behave diversely due to the decision they make or other issues, then its variant posture can be such as stand, walk, lie, crouch or fall or these postures in progress. Whereas sometimes the number of gainable known positions is

smaller than five, even smaller than three. To increase the robustness and reliabilities of localization the algorithm should be capable to handle both 2D and 3D situations.

Almost all existing localization algorithms contain two steps:

1. measuring the geographic or geometric data from limited known positions,
2. computing the unknown location according to the measured information.

At the first step various approaches are developed to be capable to inform the distance between unknown and known, e.g. signal-strength-based approach Received Signal Strength (RSS); time-based approach Time of Arrival (ToA) and Time Difference of Arrival (TDoA); angle-based approach Angle of Arrival (AoA); frequency-based approach Frequency Difference of Arrival (FDoA) etc. They could match diverse technologies and applying scenarios.

At the second step, to produce an unambiguous estimation result of an unknown position a lot of mathematic and statistic methods have been developed in calculation processes to produce a trustworthy localization estimation result, for instance conventional triangles approaches, Least Squares (LS) and its variable forms, Gauss-Newton Interpolation approaches (i.e. Taylor Series), Maximum Likelihood, Kalman Filters and so on, to address varying relationships between knowns and unknown in different scenarios. But all of them require that the amount of known is greater than the amount of unknown.

Time-based approach is the a feasible method for acoustic self-localization for autonomous soccer robots. Comparing to ToA, TDoA algorithm works without time synchronising between sender and receivers to inform of the precise sending and receiving time, it needs only acquiring the receiving time. However, as in Chapter 2 mentioned that it requires the synchronisation among receivers side, so there is no sending time information contained in the AFSK sound, as a result the duration in creating and playing even decoding AFSK is shortened, and also the number of errors on AFSK demodulating could be decreased properly.

In the Chapter 2 the TDoA algorithm is introduced. However the hyperbolic equation group is nonlinear and difficult to be solved conventionally. Many foregoing researchers have devoted to providing various ideas and methods for providing solution of the equations in different complexity and accuracy level to cover different demands.

Every algorithm has its own features, e.g. the Taylor Series [23] offers more accurate location estimation when the noise level is high, otherwise it is able to cooperate with other approaches and to achieve more reliable estimations. On the other hand it is an iterative method which is intensive computing and requires an initial guess iterated in the

calculating process, thus the quality of the initial guess is affecting the estimation result. Fang's TDoA algorithm [24] is limited within 2D and an extra piece of information could not improve the final estimation result. The LS group approaches, such as Least Square (LS), Total Least Square (TLS), Two Step Least Square (TSLS) and Weighted Least Square (WLS) are comprehensively deployed in position estimation algorithms due to the easy implementation with reasonable accuracy and cheap computation cost. So does the Maximum Likelihood (ML) approach.

Hence the TLS and one of the famous ML approaches are shown in this section, both of them are capable to deal with 2D and 3D position estimation.

3.4.1 TDoA Estimation based on Total Least Square (TLS)

From the paper [25] the formula 3.6 is known from which the position estimation results come out.

$$p = \frac{1}{2}(A^T A)^{-1} A^T k \tag{3.6}$$

where

$$p = \begin{bmatrix} x \\ y \\ r_1 \end{bmatrix} \tag{3.7}$$

$$A = \begin{bmatrix} x'_2 & y'_2 & \frac{r_{2,1}}{2} \\ x'_3 & y'_3 & \frac{r_{3,1}}{2} \\ \vdots & \vdots & \vdots \\ x'_n & y'_n & \frac{r_{n,1}}{2} \end{bmatrix} \tag{3.8}$$

$$k = \begin{bmatrix} k'_2 - r^2_{2,1} \\ k'_3 - r^3_{3,1} \\ \vdots \\ k'_n - r^n_{n,1} \end{bmatrix} \tag{3.9}$$

In 3.7, (x, y) is the position of the unknown node and r_1 is the distance from the unknown to the reference node marked by number 1, and $r_{n,1}$ can be produced by equation 2.13 in Chapter 2. The equation 3.8 about A contains known information relating to the unknown.

$$\begin{aligned} x'_n &= x_n - x_1 \\ y'_n &= y_n - y_1 \\ k'_n &= x_n^2 - x_1^2 - y_n^2 - y_1^2 \\ n &= 2, 3, \ldots, n. \end{aligned} \tag{3.10}$$

3.4.2 TDoA Estimation based on Maximum Likelihood (ML)

The ML algorithm from reference [26] for TDoA is a closed-form non-iterative solution, for 3 arbitrary nodes with known positions and an unknown in a system, so the estimation of unknown with 3 nodes given by Chan is calculated by equation 3.11.

$$\begin{bmatrix} x \\ y \end{bmatrix} = -\begin{bmatrix} x_{2,1} & y_{2,1} \\ x_{3,1} & y_{3,1} \end{bmatrix}^{-1} \times \{ \begin{bmatrix} r_{2,1} \\ r_{3,1} \end{bmatrix} r_1 + \frac{1}{2} \begin{bmatrix} r_{2,1}^2 - K_2 + K_1 \\ r_{3,1}^2 - K_3 + K_1 \end{bmatrix} \} \tag{3.11}$$

The information about $r_{i,1}$ is calculated by equation 2.9. The item r_1 is informed by equation 3.12.

$$\begin{aligned} r_i^2 &= (x_i - x)^2 + (y_i - y)^2 \\ &= K_i - 2x_i x - 2y_i y + x^2 + y^2 \end{aligned} \tag{3.12}$$

Where

$$\begin{aligned} K_i &= x_i^2 + y_i^2 \\ i &= 1, 2, \ldots, n. \end{aligned} \tag{3.13}$$

When the system obtains more than 3 known positions, then it becomes overdetermined which is the number of known information much greater than the unknown i.e. as the known nodes are 4 at least. This ML is contracted by the next two steps.

1. Per weighted linear LS an initial solution is produced.

2. By taking the former result as a piece of extra information and substituted into WLS an ameliorated outcome is brought out.

Letting the vector about the unknown be $z_a = [z_p^T, r_1]^T$ where $z_p = [x, y]^T$, and r_1 is the prior knowledge.

$$\begin{aligned} z_a &= \arg\min\{(h - G_a z_a)^T \Theta^{-1} (h - G_a z_a)\} \\ &\cong (G_a^T Q^{-1} G_a)^{-1} G_a^T Q^{-1} h \end{aligned} \tag{3.14}$$

The first estimation is given by equation 3.14, where

$$h = \frac{1}{2}\begin{bmatrix} r_{2,1}^2 - K_2 + K_1 \\ r_{3,1}^2 - K_3 + K_1 \\ \vdots \\ r_{M,1}^2 - K_M + K_1 \end{bmatrix} \tag{3.15}$$

$$G_a = -\begin{bmatrix} x_{2,1} & y_{2,1} & r_{2,1} \\ x_{3,1} & y_{3,1} & r_{3,1} \\ \vdots & \vdots & \vdots \\ x_{M,1} & y_{M,1} & r_{M,1} \end{bmatrix} \tag{3.16}$$

.

The K in equation 3.15 is equal to the equation 3.13. The covariance matrix Q, for example a system containing 5 nodes, 4 known and 1 unknown which is proposed in [26] can be constructed by matrix 3.17.

$$Q = \begin{bmatrix} 1 & 0.5 & 0.5 & 0.5 \\ 0.5 & 1 & 0.5 & 0.5 \\ 0.5 & 0.5 & 1 & 0.5 \\ 0.5 & 0.5 & 0.5 & 1 \end{bmatrix} \tag{3.17}$$

And the equation 3.18 is a LS equation about z_a, then to proceed to second WLS process by using the perturbation approach which is similar to the first part, but the covariance matrix contains information about the first outcome.

$$z_a^{'} = (G_a^{'T}\Theta^{'-1}G_a^{'})^{-1}G_a^{'T}\Theta^{'-1}h^{'} \tag{3.18}$$

$$\Theta = c^2BQB \tag{3.19}$$

c is signal propagation speed.

$$B = diag\{r_2^0, r_3^0, \ldots, r_M^0\} \tag{3.20}$$

This algorithm computes with respect to the first arrival, which means that all data compare themselves with the first one.

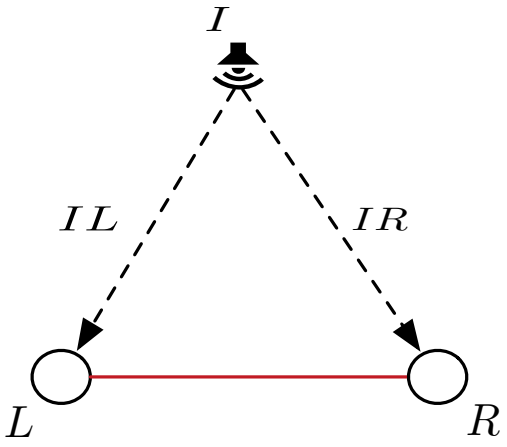

Figure 3.6: Minimum of differentials peak index on right and left microphones

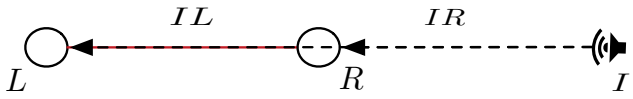

Figure 3.7: Maximum of differential peak index on right and left microphones

3.5 Robot Orientation Estimation

Robot's orientation belongs to a part of robot localization task. Nevertheless through mono track the orientation of robot can not be determined. Taking two channels of different amount samples information, the orientation of robot becomes calculable.

3.5.1 Sample Measurement Method

Assuming a sound is emitted by speaker of I. R and L denote the both side microphones of the NAO robot. The positions of I is known. And existing a node A is the real location of a robot, it is calculable or known, then the positions of two microphones are accessible as well with the orientation of robot.

Figure 3.6 shows that the robot faces to a sound emitter straightly, then robot records the sound from I, the arrival time to microphone L equals to the time to microphone R, hence the differential arrival time between microphone L and microphone R is 0. Figure 3.7 illustrates the maximal difference between sound arrival time of R and L, and if microphone R is closer to sound emitter, then the arrival time to R is smaller than L's.

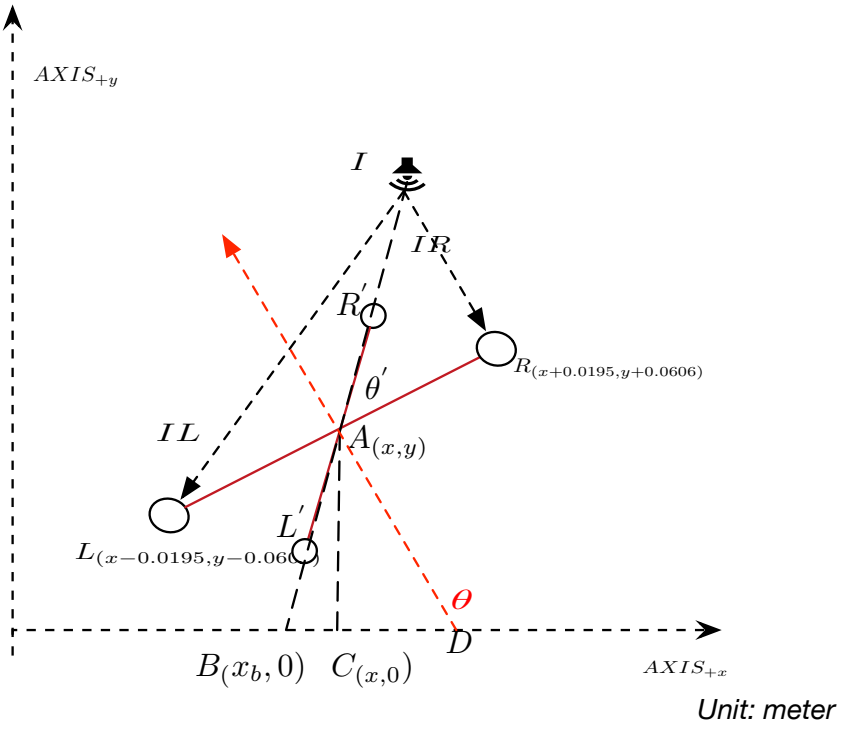

Figure 3.8: Simplified robot orientation calculation model

So,

$$\Delta t = (IR - IL)/v = RL/v \tag{3.21}$$

where v is the speed of sound, and RL is in Table 1.1. In case of Figure 3.6 LR is perpendicular of the LR in Figure 3.7. The distance of R and L is 0.1212 meter, which is roughly equal to 15 samples by sampling rate at $44100Hz$.

$$RL = 0.0606 + 0.0606 = 0.1212 \tag{3.22}$$

$$samples = 0.1212 \times \frac{1}{v} \times 44100 \approx 15 \tag{3.23}$$

Hence the angle difference of each sample is $\pm 6^\circ$ computed by equation 3.24.

$$90^\circ \div 15 samples = 6^\circ / sample \tag{3.24}$$

If θ is the desired angle for representing the orientation of robot. Then the simplified robot orientation calculation model is transferred to the Figure 3.8.

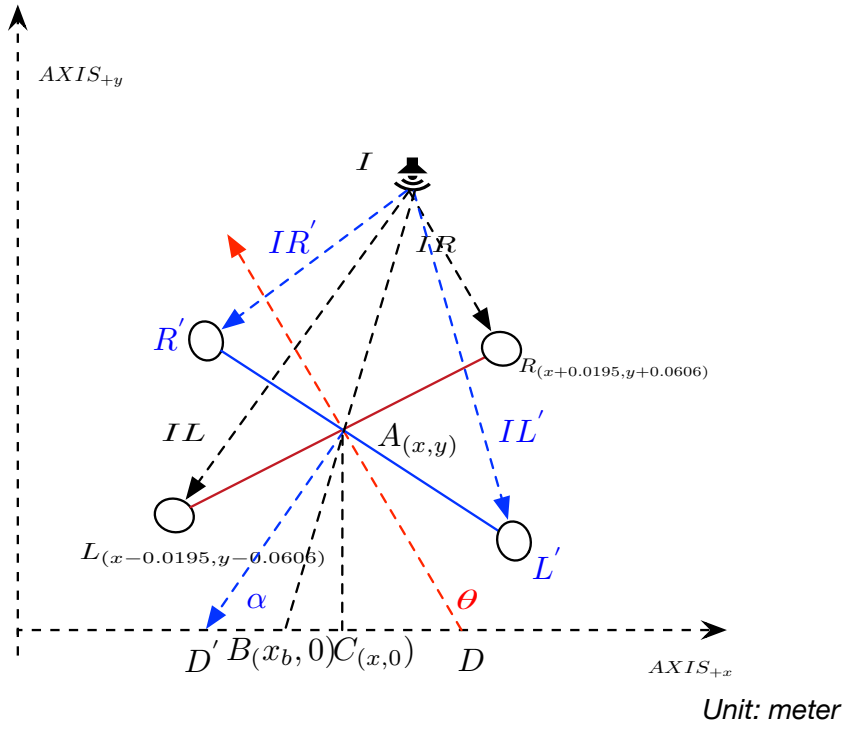

Figure 3.9: Simplified robot orientation calculation model (cont.)

In Figure 3.8 position information about I and A are given, additionally $|AL| = |AR|$, otherwise $|AD| \perp |LR|$ and $|AC| \perp |BD|$. So the triangle ABC is fully known. The $|LR|$ form $|L^{'}R^{'}|$ changed $\angle\theta^{'}$.

$$\begin{aligned}\angle\theta &= 180^\circ - \angle ADC \\ &= 180^\circ - (180^\circ - 90^\circ - \angle CAD) \\ &= 180^\circ - \{180^\circ - 90^\circ - (90^\circ - \angle\theta^{'} - \angle BAC)\}\end{aligned} \tag{3.25}$$

$$\angle\theta = \angle\theta^{'} + \angle BAC \tag{3.26}$$

$$\sin\angle BAC = \frac{|AC|}{|AB|} \tag{3.27}$$

Now the $\angle\theta$ between $X\text{-}axis$ and the line through points A and D is computable, but a line has no direction, hence the orientation of robot at point A owns two possibilities, $\angle\theta$ and $\angle\theta + 180^\circ$. Due to $|IR| < |IL|$, the orientation of robot is $\angle\theta$.

Another parallel situation shown in Figure 3.9. The difference of samples between $L^{'}$ and $R^{'}$ in colour and the original case in colour red is same, but the $\angle\alpha$ is not equal

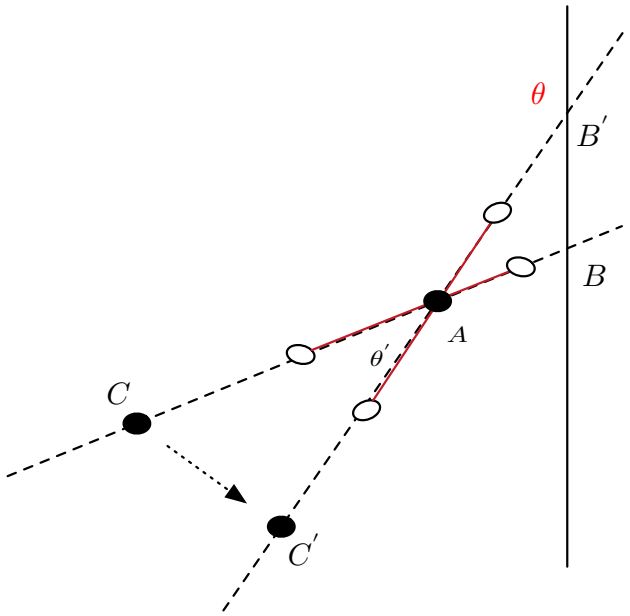

Figure 3.10: Simplified robot orientation calculation model for sample measurement method

to $\angle\theta$. So the orientation of robot has two solutions, one is from equation 3.26, the other is from equation 3.28.

$$\begin{aligned}\angle\alpha &= 180^\circ - \angle D^{'}AB - \angle D^{'}BA \\ &= 180^\circ - (90^\circ - \angle\theta^{'}) - (180^\circ - \angle ABC) \\ &= \angle\theta^{'} + \angle BAC - 90^\circ\end{aligned} \tag{3.28}$$

While this ritualistic method is strongly depending on the geometric relationship between the sound receiver and the sound emitter. Hence the following solution is proposed.

Two arbitrary nodes are existing in a system. Lining up the two points which is referring the case when the difference peak index in sample is zero between two microphones R and L. The line has minimally one intersection $(x_A, 0)$ on $X\text{-}axis$ or $(0, y_A)$ on $Y\text{-}axis$. The $\angle\theta^{'}$ in Figure 3.10 is same as the $\angle\theta^{'}$ in Figure 3.8. The line CB is around point A rotating by a angle of $\angle\theta$, then point $C^{'}$ is determinable and $B^{'}$ as well. Now $\angle\theta$ is estimable. Because the direction of robot is orthogonal of the line of CB, the orientation of robot is known. If the right microphone receives signal first, then the orientation of robot is counterclockwise of 90° to the angle θ, otherwise clockwise.

Considering the microphone sensibility of NAO robot the range of angle needs enlarging corresponding to each sample.

3.6 Chapter Summary

This chapter included the main process of acoustic self-localization and introduction of many techniques involved in this process.

The compact process contains basically five subprocesses.

1. Making the decision for sound sampling on both sides, playing and recording to gain better audio quality for the AFSK demodulation process.

2. Comparing the strategies of sound detecting. Since the localization algorithm is called TDOA, which is taking advantage of differences of arrival time of sound.

3. Generating AFSK with a 5-bit encoding method, Baudot Code with a forward error correction technique, Hamming(7,4) for self-able-correction due to the inherent characteristics of sound propagating in the air with short duration but more information.

4. AFSK demodulating process, involving various DSP components for denoising, filtering and reconstructing processes, such as windowing function, Butterworth Filters.

5. TDoA localization technique and its associating algorithms, Total Least Square (TLS) and Maximum Likelihood (ML), they provide a computational advantage with reliable estimation results as the noise level is relatively high.

6. An audio sample measurement method for robot orientation detection.

Chapter 4

Implementation of Acoustic Self-localization

This chapter documents every indispensable piece which might impact the main process immediately and lead to failure or error.

4.1 Audio Frequency Shift Keying (AFSK) Modulation and Demodulation

The first important process of acoustic self-localization for autonomous soccer robots lays down to the audio communication exchanging information between robots. These robots exchange information about their present position coordinates via AFSK signal. Different languages are analogous to different encoding and decoding methods to human being, while AFSK signal, a communicating method aiming to soccer robots, requires encoding and decoding processes too. The techniques involved in the information encoding and decoding processes are discussed in Chapter 3 slightly. The AFSK signal generating processes are illustrated in Figure 3.2, and among them the Baudot, Hamming(7,4) techniques and FSK encoding process are briefly presented as well. In this section more details about implementing AFSK demodulating processes and localization are informed.

4.1.1 Hardware Driver and Python Audio I/O

As mention in previous chapter that $2000Hz$ as carrier frequency and $1000Hz$ as discrete frequency, since the inherent noise frequency generating by

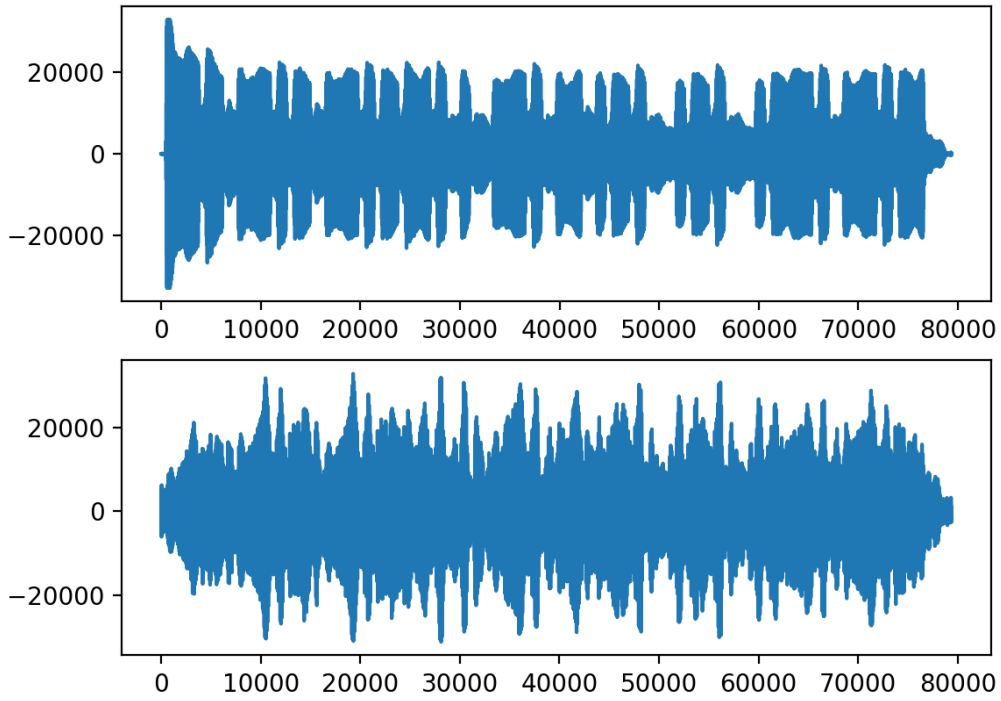

Figure 4.1: Recoding results contrast between normal and distorted sound

the fan in the head of NAO robot is $800Hz$. By an overlap of signal and noise, the original signal is hardly to be recovered. Although the high range frequencies are more easily distinguished from environmental noise essentially, on the other hand high audio signal can arise uncomfortable feeling or prickling in ears which is unbearable.

In report [6] the Austrian Kangaroos built a whistle detection system for controlling game controller to begin or stop the game. In that a well-known audio framework named Advanced Linux Sound Architecture (ALSA) is taken to drive the sound device in NAO robot that is native embedded in NAO robot. There are also several "ALSA for Python" [27] projects available, e.g. PyAlsaAudio. But the testing of AFSK demodulating turns out that the PyAlsaAudio causes distorted sound by playing AFSK signal in range from $1000Hz$ to $3000Hz$. The received audio signal is impossible to be recovered, because the original signal is destroyed on the sending side while playing which would have been or closed to be noiseless.

In Figure 4.1 both plots shows a recorded AFSK signal sampled at $11025Hz$ and $2000Hz$ as carrier frequency and $1000Hz$ as discrete frequency. The first plotting represents the recording result on a MacBook Pro without any peripheral audio I/O devices

using Audacity in $44100Hz$ sample rate and the original AFSK signal is played also by Audacity. The second plotting shows the recording result on a V4 NAO robot with ALSA in $44100Hz$ and 100% microphone sensitivity, besides the original AFSK signal is played by a V5 NAO robot with 35% playback volume and holding a LOS distance of $230cm$ between them. Both robots played and recorded signal monophonically.

Although this poor example might be unapproved in a thesis due to no restricted attitude of scientific reaching, it is only in the charge of that we can be visually informed from Figure 4.1 that the first plotting has a smoother envelope than the plotting beneath it, and a lot of thorns around the sound track are in the second plotting figure, otherwise by playing human hears cracking sound.

This phenomenon occurs very often, such as unstable electricity currents or because of faulty speaker or microphone, or of other software or program issues, for instance, driver or data format is unmatched to the hardware requirements.

Obviously on the hardware aspect, modifying and changing are not allowed according to the clause in RoboCup Standard Platform League (NAO) Rule Book [18], au contraire the software or program aspect, where the attentions are worthy to be paid for. That is the reason why another audio I/O library is applied, PortAudio, in combination with a Python package bindings for PortAudio audio input and output called PyAudio, are recommend and imported in this master thesis, also implemented in the sound localization prototype. AlsaAudio [27] is engaged at the same time.

A general signal can be formulated by:

$$y = Asin(2\pi ft) \tag{4.1}$$

where A is amplitude, f is frequency, t is time. Generating a sinusoid with Python with matched data type of the signal is fundamental because "Sound format controls how the PCM device interpret data for playback, and how data is encoded in captures" [27]. There are two data formats in PCM, which are shown up with the bit depth, such as 16-bit integer, 32-bit float for signed or unsigned etc, evidently that float data type is able to represent more details with endless number after the decimal point than integer data type, especially meaningful for music or high-quality synthesised audio. If the bit depth is too small, such as 4-bit and 8-bit that is also known as Chip tune which sounds tedious and electric. Additionally, the high bit depth requires more memory and CPU capacity to process the raw audio data. A tradeoff in this thesis is that the 16-bit unsigned integer as playing and recording data format is used. Consequently the sinusoid signal needs resetting the range of sample from $[-1, +1]$ to others associating with the configuration

of PCM before playing if it is required. For instance, resetting the range from $[-1, +1]$ to $[-32768, +32767]$.

Listing 4.1: Denormalization for 16-bit data

```
import numpy as np

def denormalize(data):
  return np.int16(data/np.max(np.abs(data)) * 32767)
```

4.1.2 AFSK Modulation and Demodulation

The AFSK modulation processes are illustrated in Figure 3.2, it is simulating a Voltage Controlled Oscillator (VOC) that changes frequencies with regards to the input data. The constructed signal is a cosine instead of sine. Corresponding to the Nyquist-Shannon Sampling Theorem in Chapter 2, the sampling frequency has to be at least twice higher than the carrier frequency, e.g. $11025Hz$ as carrier frequency, then the sampling frequency is at least more than $22050Hz$.

An example of a piece of information modulated in AFSK is shown in Figure 2.1, the FSK signal is generated by Python Script locating in Appendix A.

The FM type demodulation process is discussed in Chapter 3 engaging Butterworth Filters and window function. The recommended processes for a FM type demodulating in references [14, 28] are in the underlying list in sequence.

1. Band Pass Filter
2. Limiter
3. FM Discriminator
4. Low Pass Filter
5. Slicer

In this thesis every function mentioned in upper list is implemented and supplemented with other necessary functions, working in the sequence as following.

1. Limiter
   Setting an environmental threshold for sound amplitude. In Chapter 3 four approaches for sound onset detection functions are present. i.e. RMS, amplitude-based, frequency-based with their algorithms respectively and the matched filter. And also pointed out that due to the limitation of hardware on NAO two of four are implemented.

Listing 4.2: Peak detection

```
import numpy as np

def getAmplitude(snddata):
  # reading data from stream.
  piece = np.fromstring(snddata, np.int16)
  peak = np.max(piece)
  amplitude = peak / 32768.
  return amplitude
```

But first the audio data coming from stream need forming from raw audio data to Int16. But the outcome cannot be used as a threshold for sound onset detecting, because a single number represents nothing, while the arithmetic mean could be referred to the characteristic of a series of numbers. Otherwise if keeping detecting sound or detecting nothing in a row, then the threshold is too high or too low. Hence the setting environment threshold must be processed periodically.

The process is modelled in Figure 4.2,

2. Butterworth Band Pass Filter
   In Section 2.3.3 an introduction about Filter function and Butterworth BPF in different orders is given. In Figure 2.5 all functional Butterworth BPFs go through 2 points ideally. Sequentially in Section 3.3 the formulas 3.4 and 3.5 of calculating the two points are brought out with testing results of corresponding designed Butterworth BPFs in extraordinary complicated environment plotted in Figure 3.5.

3. Smoother
   Savitzky-Golay filter.

4. FM Discriminator
   A FM discriminator is usually made by a differentiator followed by envelope detector. Nevertheless by testing and evaluating of this particular combination of DSP functions, a poor performance of time-consuming aspect led the entire process into numbing statue or system halted.

   Profiling the AFSK demodulation process gives an overview of time consuming by execution, that indicates that the program spends too much time on the envelope detecting function using Hilbert Transform which demands intensive computation on account of fast Fourier transform (FFT) and convolution, especially when the length of signal is odd.

   The strategy of solving this issue is taking a replacement function, instead of Hilbert Transform an alternative approach for envelope detection is, first calculat-

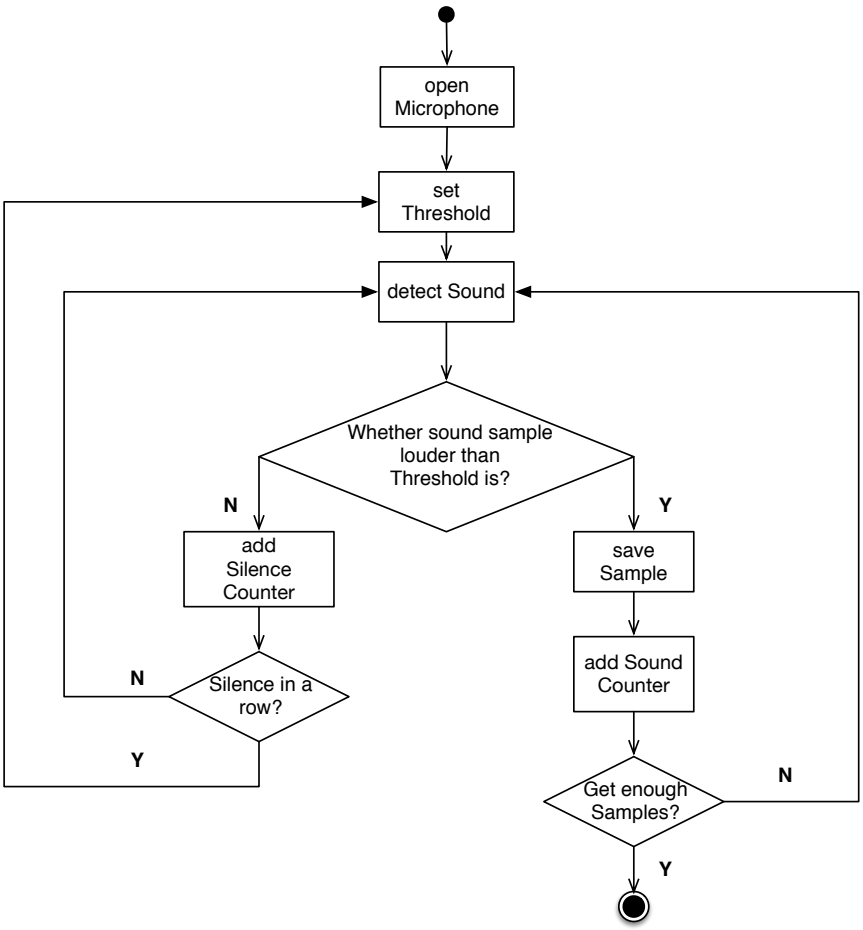

Figure 4.2: Recording process

ing the absolute part of sampled signal and making the result goes through a LPF which has an extreme cutoff frequency to gain the envelope, that is exactly the final step forward to the reconstructed signal.

5. Low Pass Filter
   LPF with Hann window.

6. Slicer
   The mean of filtered signal as the indicator for making decision.

Exclusive of Slicer and Limiter, which is introduced in Chapter 3 in the section about Sound Onset Detection, all functions have been wrapped in Python scientific packages Scipy and Numpy.

4.2 TDoA Algorithms Implementation

A general TDoA algorithm model is established in Chapter 2 with calculating step, an overview of different localization algorithms and particularly the time-based localization algorithms for comparison is shown in Chapter 3. Among them two algorithms for TDoA for solving a group of non-linear equations of hyperbola, TSL and ML, are considered as implementation components for acoustic self-localization for autonomous soccer robots.

4.2.1 TDoA Total Least Square (TLS) Algorithm Implementation

Beginning to form the TLS equation 3.6, p of equation 3.7 is desired and referring to the position of the unknown node, matrix A in equation 3.8 has two parts, one part is about the differences of distance between one node and others, the other part is about the differences of distance of arrival.

Let the node named 1 be the reference node, because paper of [25] announced that "all the TDoAs are measured with reference to the first reference node", which needs pointing out that the "the first reference node" is not the first arriving node or set in another way, essentially, any node can be this reference node.

The first part is that all positions subtract the same reference node, i.e. the node named as 1 located in (x_1, y_1), from equation 5.2.

$$rnr = \begin{bmatrix} x1 & y1 \end{bmatrix} \tag{4.2}$$

$$rn = \begin{bmatrix} x2 & y2 \\ x3 & y3 \\ x4 & y4 \end{bmatrix} \tag{4.3}$$

In the second part r^2 is contained by equation 4.4, where v is the speed of sound.

$$rdoa = \begin{bmatrix} toa_2 - toa_1 \\ toa_3 - toa_1 \\ toa_4 - toa_1 \end{bmatrix} \times v \tag{4.4}$$

Hence the final constructed matrix of A is in the next Python script.

Listing 4.3: Matrix A in equation 3.8

```
dim = np.shape(ary)-1
tdoa = ary[1:, dim] - ary[0, dim]
A = np.hstack((rn.T-rnr.T, 5e-1*rdoa.reshape(np.shape(tdoa)[0],1)))
```

Another important piece of equation 3.6 is the k, that is the sum of each position squared and then subtracts the sum of the reference node's position squared, and their difference subtracts distance between them squared. Due to the equation 4.4, the squared $rdoa$ is early calculated.

Listing 4.4: rdoa and K in equation 3.9 for Total Least Square

```
rdoa = tdoa * v
rdoa_squared = rdoa * rdoa
K = (np.sum(rn * rn,axis=0) - np.sum(rnr * rnr, axis=0)) - rdoa_squared
```

The p is calculated only when the matrix of A is invertible in equation 3.6, which is, only if the $rk(A) = rk(A^T)$, such as in 2D localization scenario, the matrix A is (3×3), likewise for 3D (4×4), otherwise it could not hand out an estimated result.

But in many cases, an extra node could provide useful information that products a more trustful estimation for localization algorithm. When matrix A is $(m \times n)$, and all elements belong to a field, then the matrix A is decomposed as in equation 4.6 due to Singular Value Decomposition (SVD) which is proposed in reference [25], hence instead of inverting a non-invertible matrix in sometimes, calculating its pseudoinverse of matrix A instead.

$$A = U\Sigma V^T \tag{4.5}$$

$$A^+ = V\Sigma^+ U^T \tag{4.6}$$

In equations 4.5 and 4.6, Σ^{+} is the pseudoinverse of Σ, and $*^{T}$ indicates the transpose of matrix.

So this complicated numerical computation process in Python is shown as followed Python snippet and the estimation result is gained.

Listing 4.5: Calculating estimation of equation 3.6

```
estimation = 5e-1 * np.dot(np.linalg.pinv(A),K)
```

4.2.2 TDoA Maximum Likelihood (ML) Algorithm Implementation

In paper [29] the ML algorithms for 3 nodes and for more than 3 have been implemented with detailed information and in a very traditional manner. There is no reason for a redundancy in parroting. As well as in the Chapter 3 the elaboration of Chan's ML algorithm, many steps were omitted in order to draw a clear outline for the adjacent improved implementation of the ML TDoA algorithm for more than 4 nodes.

As mentioned in Chapter 3, the ML TDoA algorithm contains 2 steps, first LS and then WLS is involving the outcome from the fist LS that the process is very similar to Two-Stage LS.

Usually the LS equation has the form of equation 3.6 and the implementing is presented in the pervious section for LS which is also pointed out that the pseudoinverse could solve the least square problem, because sometime the LS can not hand out a result if the matrix is not invertible. Therefor this implementing is also taking pseudoinverse as first calculating for the LS about the constructed TDoA matrix, and then calculating the WLS with convenience vector in a normal way. This improved algorithm in Python shows faster feature.

Depending on the array named ary from Section 5.2.1.1, according to [26], the array should be sorted by ToA first.

Listing 4.6: rdoa for ML

```
import numpy as np

n, dim = np.shape(ary)[0], np.shape(ary)[1]
group = ary[ary[:, dim].argsort()]
tdoa = group[:, dim] - group[:, dim][0]
rdoa = tdoa * v
```

Gathering necessary knowledge for matrix 3.16.

Listing 4.7: Matrix in equation 3.16

```
D = np.hstack((self.group[:, :dim], rdoa.reshape((self.n,1))))
M = D[:, :dim]
G = D[1:] - D[0]
```

Constructing h in equation 3.15 and computing the first estimation from LS, and extracting the error vector $R0$.

Listing 4.8: First LS estimation in ML and the error vector R0

```
r_squared = rdoa[1:] * rdoa[1:]
K = np.sum((np.multiply(M, M)),axis=1)
h = 5e-1 * (r_squared - K[1:] + K[0])
first = np.dot(np.linalg.pinv(-G),h)
R0 = first[-1]
```

Then constructing the covariance matrix $Theta$ of equation 3.19.

Listing 4.9: Covariance matrix

```
theta = np.mat(np.diag((rdoa[1:] + R0) * v ** 2))
```

Substituting the matrix $Theta$ in to the WLS equation 3.18 to get an ameliorated localization estimation result.

Listing 4.10: Weighted Least Square for ML

```
EST = ((-G.T * theta.I * -G).I * (-G.T * theta.I) * np.mat(h).T).A.squeeze()
```

4.3 Chapter Summary

In this chapter all necessary details for fulfilling a successful implementation of AFSK modulating and demodulating and code scripts for both TDoA algorithms, TLS and ML, are completed. Many scientists and programmers give their wisdom and effort building a solid foundation which is also within the researching scope of this thesis. While the specific design and implementation for NAO robot with a significant limitation of hardware capacity, has its advantage, overwhelming the limited capacity on NAO and adapting the Rules of RoboCup Standard Platform League. Many processes are redefined and reimplemented with attentions on the feature of technologies compromising between accuracy, reliabilities and time consuming.

Chapter 5

Integration and Evaluation of Acoustic Self-Localization

This chapter elaborates experimental testing of the acoustic self-localization on NAO robots, discuses different strategies for implementation and their performances about accuracy, efficiency, drawbacks etc.

5.1 Audio Frequency-shift Keying Demodulating

The audio FSK demodulating shows satisfied robustness and reliability even under very harsh conditions. The ameliorated AFSK demodulating process in Chapter 4 is specified. Due to engaging of $2000Hz$ as carrier frequency, the desired audio piece could be free from being contaminated by human voice environmental noise and music. However, if the environmental noise level is low, i.e. high SNR, the Butterwort filter can be omitted, since the followed Savitzky-Golay filter is capable to filter out the environmental noise in order to accelerate the demodulating process.

The difficult part is located after AFSK demodulating about how to extract interested information from a large amount. Set in another way that the robot should know where is or how to find the beginning of the real information to initialise decoding process. Because of the BFSK combining the AFSK, there are only two possibilities, nether 1 nor 0. The simplest way is defining a specific pattern that the robot is able to explore the pattern at first before decoding.

Otherwise the decision for cut duration/length of audio data for future processing and analysing is also tricky. Since if the length of audio data is too long, the information is easily to extract, while the robot is be overwhelmed by intensive computation in DSP.

Or if the length of audio data is too short, the robot might miss important information which is leading to non-decodable data.

An effective AFSK communication method is strongly depending on the distance from signal emitter to signal receiver, although the DSP is capable to filter out interferences and amplify the filtered signal. Nevertheless the operational range of AFSK is limited. If the volume is high, the AFSK is distorted, so that the signal is unrecoverable, while the volume is low, the sound can not be detected. In testing, when two NAO robots halt a distance within 2 meters with 70% volume the AFSK signal is demodulated. As the distance increasing, the AFSK signal becomes useless. The demodulation of AFSK is complicated in NAO robot V5 due to the EGO-noise generated by the fan in robot's head, since the SNR is tremendously low. If the signal is not strong enough in the frequency domain, it is distorted by the inherent noise easily, if it has no strong energy for propagating, it can not be detectable.

5.2 NAO Robot Self-Localization

To estimate an unknown location, two procedures must be completed which are also pointed out in Section 3.4 that first measuring the geographic or geometric information of known, and second computing the unknown with regards to the measured data. For a normal TDoA estimation, as mentioned in List 2.5.2, one of the three preconditions for a successful TDoA localization estimation is time synchronising on receiving side which is addressing for the measuring task. The privilege of this approach is that the distance between the unknown and known is not indispensable, only the differences of distances are significant. Many uncertainties will contribute to this TDoA estimation process negatively. Apparently there could be or is delay on sending and receiving side or in synchronising process. When the sum of delays reaches millisecond stage, it transforms an erroneous localization estimation more than $34cm$.

5.2.0.1 Synchronisation and Sound Onset Detection

Sound onset detection is the most compulsory and important component before TDoA localization calculating. Apparently synchronisation is an obstacle that has to be overcome first.

There are several strategies to make robots synchronise together. Such as dancing robots [30] used Server/Client topology to synchronise the movements of a swarm of NAO robots. They cut the timeline into small spans for oscillator speed adjustment

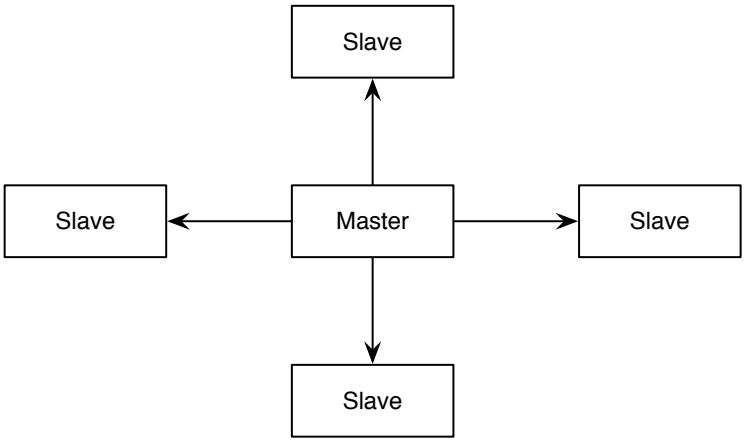

Figure 5.1: Master/Slave networking

which makes sure that all robots could dance simultaneously, even if some of them are newly added or fell and then still could catch up later on. In that paper also mentioned that "this kind of experiment has already been conducted by Aldebaran for the Universal Exposition in Shanghai in 2010 with 20 robots. Their approach was to synchronise every robots clock by using the NTP protocol, which is currently used by every computer to synchronise its clock on the Internet." Both cases are using NTP for clock synchronising between robots, alike the local time synchronisation method of computer to get precise local time. But the NTP protocol requires the connection to the Internet which is impossible during the RoboCup. Alternatively using synchronising algorithms, such as Berkley Algorithm or Round-Trip time measuring approach. They are workable on this scenario, but both also take time to accomplish an algorithm round.

An adhoc time synchronising network, Master/Slave networking in Figure 5.1 is deployed on trial. The coach robot as master broadcasts very frequently UDP packet via Wi-Fi network with timestamp generated by its internal clock and the others receive and take it as initial timing and then begin sound onset detecting.

Results and Conclusions

By testing the model illustrated in Figure 5.2 the limited capacity of NAO has been fully taken in everything at a glance. The data processing speed is strongly constrained, so that audio I/O can not be constantly digested. It causes I/O blocking often. Otherwise receiving time stamp from master suffers long delay if the network has too much to load, however clock synchronising needs frequent data exchanging. The sensitivity of embedded microphones restricts the accuracy and performance. Thus a stringent estimation for sound signal arrival time is difficult to achieve. The inaccuracy level re-

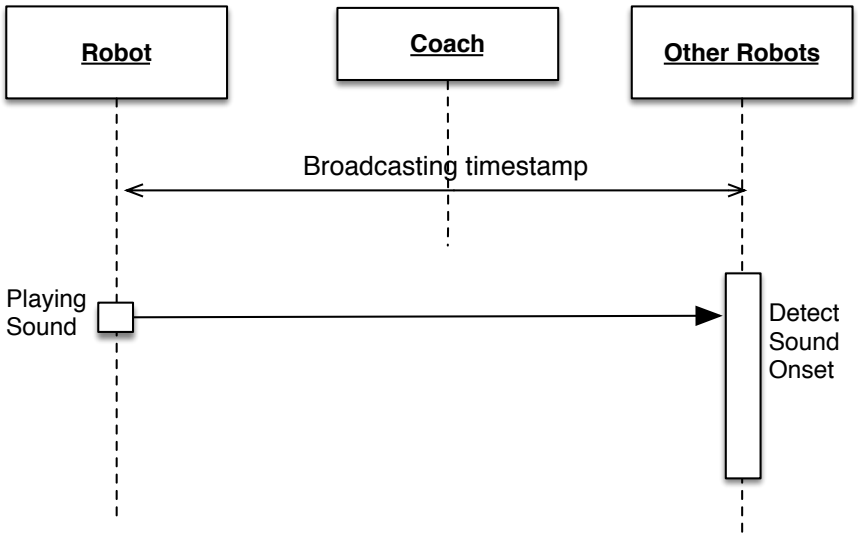

Figure 5.2: Implemented acoustic self-localization process with an adhoc time synchronising networking

mains in milliseconds which is not precise enough for localization estimation within the RoboCup field.

5.2.0.2 BeepBeep Acoustic Ranging System

The first BeepBeep Acoustic Ranging System is built in 2007 using commercial off-the-shelf (COTS) devices like mobile phones, MP3 players. According to the paper [17] the error level is controlled within 2 centimetres in range of more than 10 meters [31]. Besides the BeepBeep Acoustic Ranging System is a pure software solution and capable to handle the three aforementioned delay sources in Section 2.4 well, because it does not ask for synchronisation and timestamps and the "elapsed time between the two time-of-arrivals (ETOA)" is calculated by relative time clocks to estimate the ranging value.

This ranging measurement method needs minimally 2 devices and both in recording and playing statues, exchanging the detecting results for ranging calculation. The system used peak detection and Cross-Correlation algorithm comparing the original signal to the detected signal to evaluate the ETOA between two sound pieces emitted by two mobile phones. If the SNR is high, a successful detection could be met.

Results and Conclusions

By testing this approach, two NAO robots hold a distance of $120cm$ roughly and play signal one after one and record separately before they save the raw recorded WAV

file locally. After analysing the two raw WAV files taking advantage of RMS mentioned in Section 3.2, the error range remains within $2cm$. The BeepBeep method performs high accuracy ranging in static testing.

Narrowing down to this thesis, because the BeepBeep system is not relative to signal reconstruction, the peak detection or Cross-Correlation works analogously as a matched filter, so the Nyquist-Shannon Sampling Theorem is irrelevant in the entire process. The testing scenario in the paper [17] is recording and playing sides select $44100Hz$, so the data amount is unchanged. Nevertheless in signal reconstruction the Nyquist-Shannon Sampling Theorem is a rigidity criterion.

To integrate the BeepBeep method into the acoustic self-localization for NAO robots, the entire process is shown in Figure 5.9. All robots open audio stream and make a long enough window length to record more data as possible for later sound detecting and AFSK demodulating. To ensure the accuracy level of sound detecting, the memory is sacrificed. If the window is 60 seconds long and the sampling rate is $44100Hz$ for stereo channel, then according to equation 2.2 the data volume is $10584000Bytes$. Otherwise Cross-Correlation algorithm to matched filter is too expensive on NAO robot. So there is a compromise between accuracy and computation power that high ranging accuracy needs the a long recording window open which makes the memory and CPU suffer from the huge burden, so the CPU can not process the Audio I/O immediately, that the outcome is often overdue.

$$60sec. \times 44100Hz \times 2Bytes \times 2Channel = 10584000Bytes \tag{5.1}$$

Hence the main philosophy of BeepBeep Rang System is remaining in the prototype. Instead of Cross-Correlation algorithm a tedious peak detecting function in the time domain is deployed first, after the sampled signal go through a Butterworth Band Pass Filter (BPF) for increasing the SNR. It shows more reliable real-time feature than the Cross-Correlation algorithm and is also able to hand out a trustful peak detection result. The ranging error is usually within $150cm$.

In the Figure 5.3 the first plotting is the recorded sound in NAO V5. The sound is noisy, although the separate signal has bigger amplitude but because of wave superposition it becomes turbid. The middle plotting shows the gist of the real sound after setting an amplitude threshold. The last plotting is the sound filtered out by a Butterworth Band Pass Filter (BPF), the separate signal towers over and has an obvious peak.

The duration of the separate signal is even 0.1 second which is composed of 4410 samples on both recording and playing side.

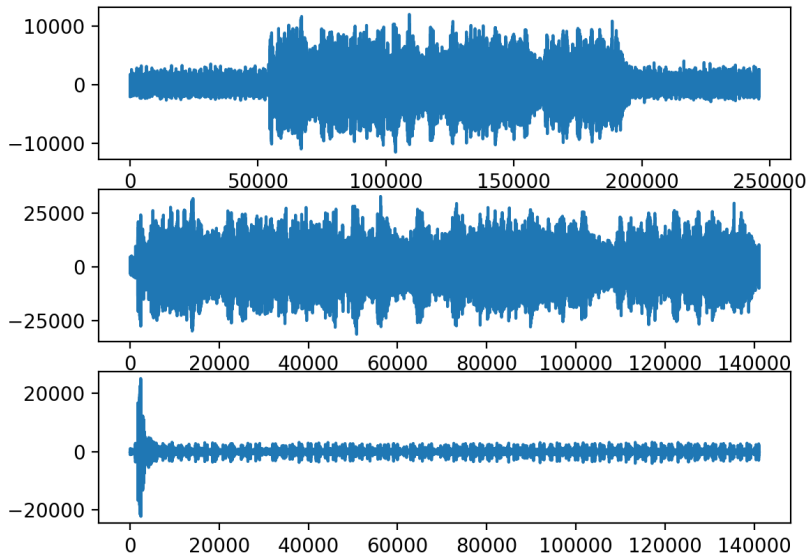

Figure 5.3: Plotting of peak detection process

5.2.1 NAO Robot Self-Localization

The two deployed TDoA associating algorithms work significantly fast and accurate in static testing. Because during RoboCup most audio propagating paths are LOS between robots, the TDoA localization algorithms fulfil all demands for a reasonable localization estimation if the ToA of audio and the known position information are accurate.

5.2.1.1 Simulating Scenario Statement

The TLS algorithm demands a reference node and others compare to it, while the ML estimates the unknown with regards to the first arriving as the reference node.

To ease the calculation of demonstration and testing the algorithm process, then a scenario for TDoA with four reference positions of robots in 2D is established.

An arbitrary locating at (x, y) in a planar surface. Generating a random 2D node in Python snippet beneath.

Listing 5.1: Generating one unknown position

```
1 import numpy as np
2 bn = L * np.random.rand(2, 1)
```

L is an integer defining the area of planar surface. It is in charge to create the ToA information.

With the known positions of robots as reference nodes with their ToA an array with $n * 4$ shape can be constructed, and n is the dimension of estimation.

$$rn = \begin{bmatrix} x1 & y1 \\ x2 & y2 \\ x3 & y3 \\ x4 & y4 \end{bmatrix} \tag{5.2}$$

Listing 5.2: Generating reference nodes

```
rn = L * np.random.rand(2, 4)
```

The corresponding toa array for the matrix 3.8 about the known positions of robots is equation 5.3.

$$toa = \begin{bmatrix} toa_1 \\ toa_2 \\ toa_3 \\ toa_4 \end{bmatrix} \tag{5.3}$$

Containing the toa array is not immediate. First step, calculating the distance between two points a and b in the Cartesian Coordinate System according to the equation 5.4. For testing purpose the unknown is informed, so the distances between the references and the unknown are calculable. Consequently the ToA is also gainable.

$$d_{ab}^2 = a^2 + b^2 \tag{5.4}$$

Listing 5.3: Calculating ToA

```
toa = np.sqrt(np.sum((rn-bn)**2, axis=0)) / v
```

Where v in the Python snippet above is the speed of sound. Adding Gaussian Noise to the toa term.

Listing 5.4: Extra noise

```
noise = np.zeros(0)
for i in range(len(toa)):
  noise = np.append(noise, np.random.randn()) * 1e-6
toa = toa + noise
```

An array constructed for testing TDoA algorithm is in equation 5.5 and correspondingly in Python code for its constructing.

$$array = \begin{bmatrix} x_1 & y_1 & toa_1 \\ x_2 & y_2 & toa_2 \\ x_3 & y_3 & toa_3 \\ x_4 & y_4 & toa_4 \end{bmatrix} \tag{5.5}$$

Listing 5.5: Array for localization estimation

```
ary = (np.vstack((rn, toa))).T
```

5.2.1.2 Simulation and Testing Results, Conclusions

- Static Testing of Localization Algorithms

  The improved TDoA associating algorithms, TLS and ML, are relatively cheap computation cost and able to hand out reasonable localization estimation results in static testing.

- Testing Results of NAO Robot Self-Localization in Reality

  The RMS method is introduced in Section 3.2 for sound detecting that is also used measure of positioning accuracy [32]. The estimated coordinates of unknown are evaluated by RMS method by following equations 5.6, 5.7, 5.8. X and Y denote the true coorinates, X_i and Y_i are estimated coordinates.

$$RMS_X = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - X)^2} \tag{5.6}$$

$$RMS_Y = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - Y)^2} \tag{5.7}$$

$$RMS_{XY} = \sqrt{RMS_X^2 + RMS_Y^2} \tag{5.8}$$

  – The first test scenario is the ALSA volume setting in 80% volume. Each robot plays a separate signal in 0.05 second in $44100Hz$ in 100% of the ALSA volume followed by an AFSK signal sampled at $11025Hz$ with 80% of the ALSA volume in mono track[1], records them at $44100Hz$ sample rate, ML algorithm for less than 4 nodes and TLS are engaged using

[1] Left loudspeaker and right loudspeaker play sound together

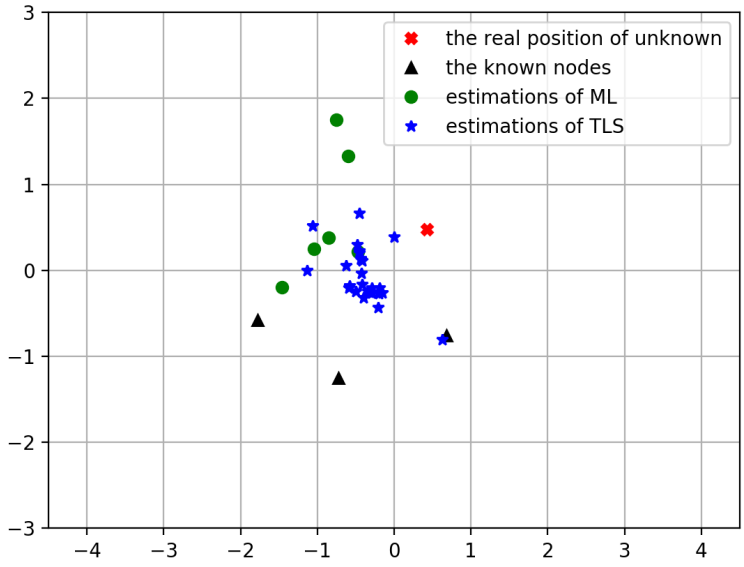

Figure 5.4: Estimations of unknown with 3 nodes

Algorithm	Source Location		Type of Errors		
	X(m)	Y(m)	RMS_X	RMS_Y	RMS_{XY}
ML for ≤ 4	0.425	0.479	1.32778089803	0.697739717866	1.4999476081
TLS	0.425	0.479	0.866803976238	0.625770647707	1.06908280164

Table 5.1: Comparison of RMS rrror for Maximum Likelihood and Total Least Square using peak detection

peak detection. Three robots with known position locate at $(0.684, -0.755)$, $(-1.780, -0.578)$ and $(-0.723, -1.249)$, the estimated results of the interest point at $(0.425, 0.479)$ which is denoted in red cross are shown in Figure 5.4.

In Figure 5.4 $blue\ stars$ denote the estimations of TLS and the $green\ circles$ are the results from ML for less than 4 nodes. Although, theoretically the estimated results of ML for less than 4 known nodes have high accuracy when the SNR is high, it gives out the localization estimation only when a positive root in equation 3.12 exists. And as the noise level increasing, the testing in reality turns out that the estimation results from TLS are closer to the real position of unknown comparing to the ML algorithm shown in Table 5.1.

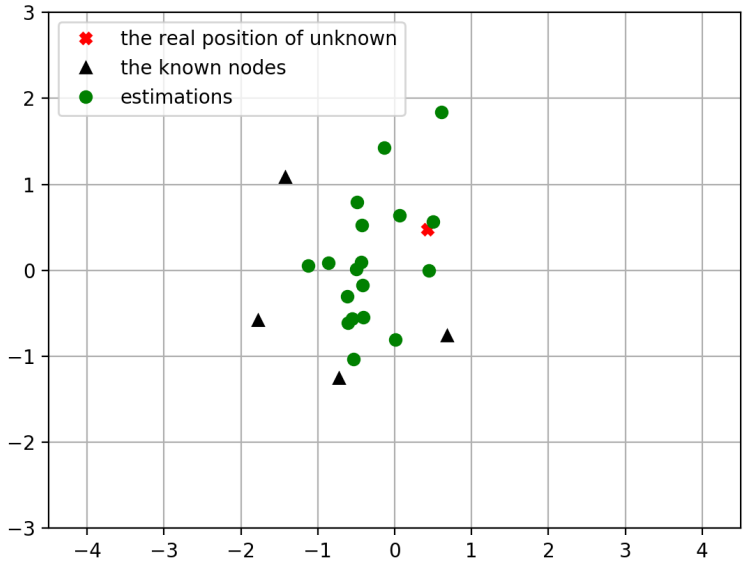

Figure 5.5: Estimations of unknown with 4 nodes

So the ML for more than 4 nodes and TLS are involved in the final implementation, both of them are able to compute 2D and 3D. The audio buffers missing issue is in [33] mentioned, besides the distorted sound interferes in peak detection process, both cases impact the localization accuracy. Even though the robots play signal in single track, the two loudspeakers as sound sources are also disturbing the signal arrival time.

– The second testing scenario is the ALSA volume in 80% volume. Each robot plays a separate signal in 0.05 second in $44100Hz$ in 100% of ALSA volume with mono track followed by an AFSK signal sampled at $11025Hz$ with 80% of ALSA volume, while records them at $44100Hz$ sample rate. ML algorithm for more than 4 nodes and TLS are engaged using peak detection.

  Three robots with known position as same as the first testing scenario locate at $(0.684, -0.755)$, $(-1.780, -0.578)$ and $(-0.723, -1.249)$, adding an extra robot at $(-1.424, 1.087)$, the estimated results in $green$ dots of interest point at $(0.425, 0.479)$, which is denoted in red cross, are shown in Figure 5.5.

  In Table 5.2 the error value is big comparing to using 3 robots, because the robot at $(-1.424, 1.087)$ is V5 which has fan in its head, the inherent noise leads into numbness.

Source Location		Type of errors		
X(m)	Y(m)	RMS_x	RMS_y	RMS_{xy}
0	0	0.859264923262	0.819786886259	1.1875970475

Table 5.2: The RMS of errors with 4 nodes using peak detection

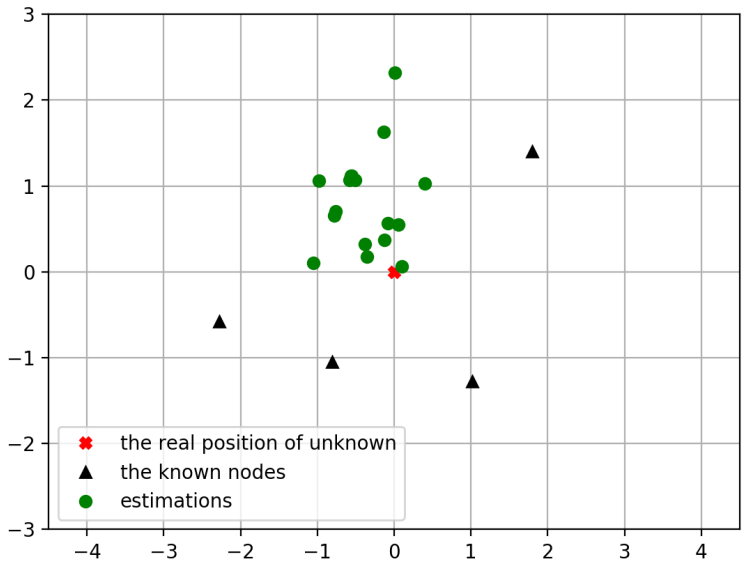

Figure 5.6: Estimations of unknown with 4 nodes using Cross-Correlation algorithm

– The third testing scenario is the ALSA volume setting in 70% volume. Each robot plays a separate signal with duration of 0.1 second in $44100Hz$ and 100% of ALSA volume single channel [2] followed by an AFSK signal sampled at $11025Hz$ with 80% volume, while records them at $44100Hz$ sample rate. ML algorithm for more than 4 nodes and TLS are engaged using cross-correlation algorithm.

  Four robots with known position locate at $(1.80, 1.40)$, $(1.018, -1.280)$, $(-2.277, -0.581)$ and $(-0.803, -1.050)$, the interest point is positioning at $(0, 0)$ which is denoted in red cross in Figure 5.6. The estimated results about position $(0, 0)$ are plotting in Figure 5.6 with $green$ dots.

– The fourth testing scenario has the same software configurations and algorithms of third testing. Setting four robots on positions at $(1.012, 1.232)$, $(0.998, -1.188)$, $(-2.032, -1.160)$ and $(-1.597, 1.444)$, the interest robot located at $(0, 0)$ and which is denoted in red cross in Figure 5.7, in addition

[2] only one loudspeaker plays, the other one is mute.

Source Location		Type of errors		
X(m)	Y(m)	RMS_x	RMS_y	RMS_{xy}
0	0	0.539754248427	0.995224720645	1.13216910984

Table 5.3: The RMS of errors with 4 nodes using Cross-Correlation algorithm

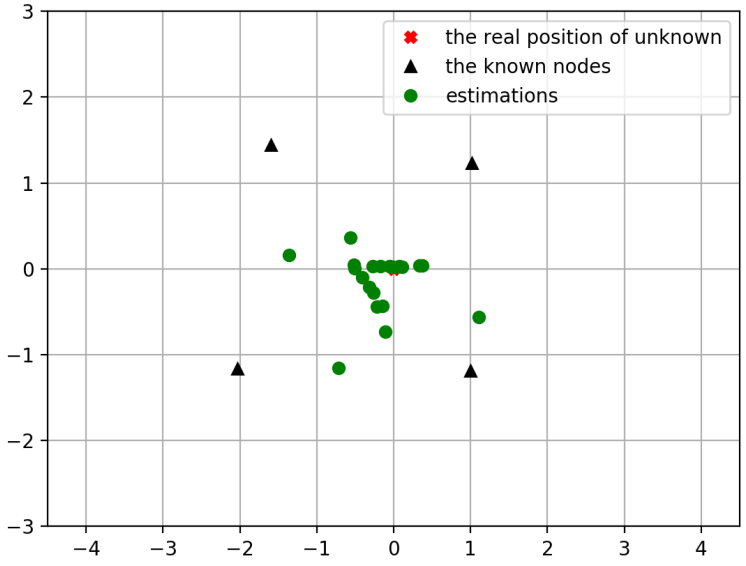

Figure 5.7: Estimations of unknown with 4 nodes using Cross-Correlation Algorithm

its orientation is $180°$ to the positive direction of $X\text{-}axis$. The estimated results about robot at $(0, 0)$ are plotting in Figure 5.7 in $green$ dots.

The Figure 5.8 in scatter plotting represents the estimated orientations of robot locating at $(0, 0)$ by the proposed robot orientation algorithm. Robot orientation estimation works differently on different NAO robots, since the two microphones on both sides have individual features. However the proposed robot orientation algorithm gives trustworthy estimation, as long as its localization estimation is available and meaningful peak detections on both microphones.

Source Location		Type of errors		
X(m)	Y(m)	RMS_x	RMS_y	RMS_{xy}
0	0	0.498759201536	0.368872028762	0.620344512928

Table 5.4: The RMS of errors with 4 nodes using Cross-Correlation algorithm

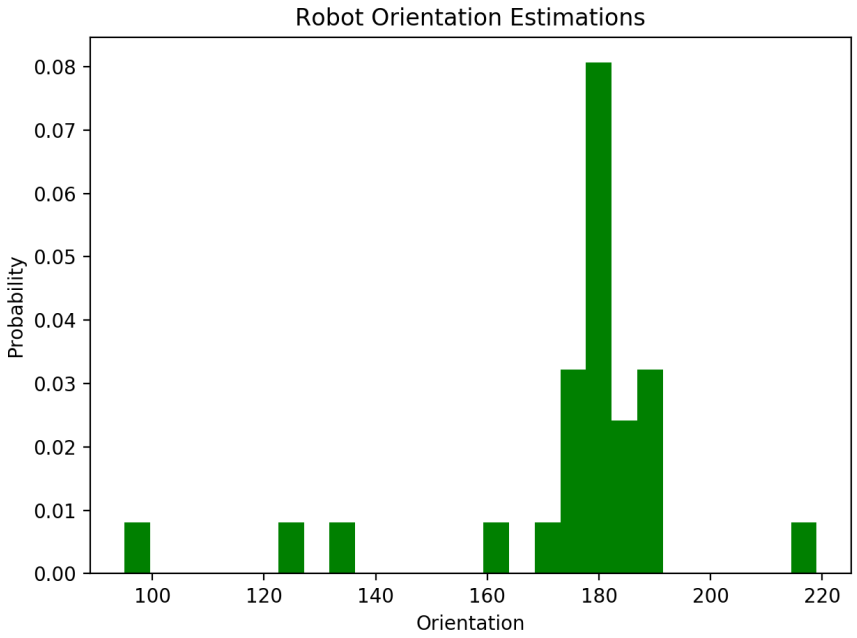

Figure 5.8: Estimations of the proposed robot orientation algorithm

5.3 Integration

The original design of integrating acoustic self-localization for NAO robot is in Figure 3.1.

The compact processes pertaining to the subject of acoustic self-localization for autonomous soccer robots is illustrated in Figure 5.9.

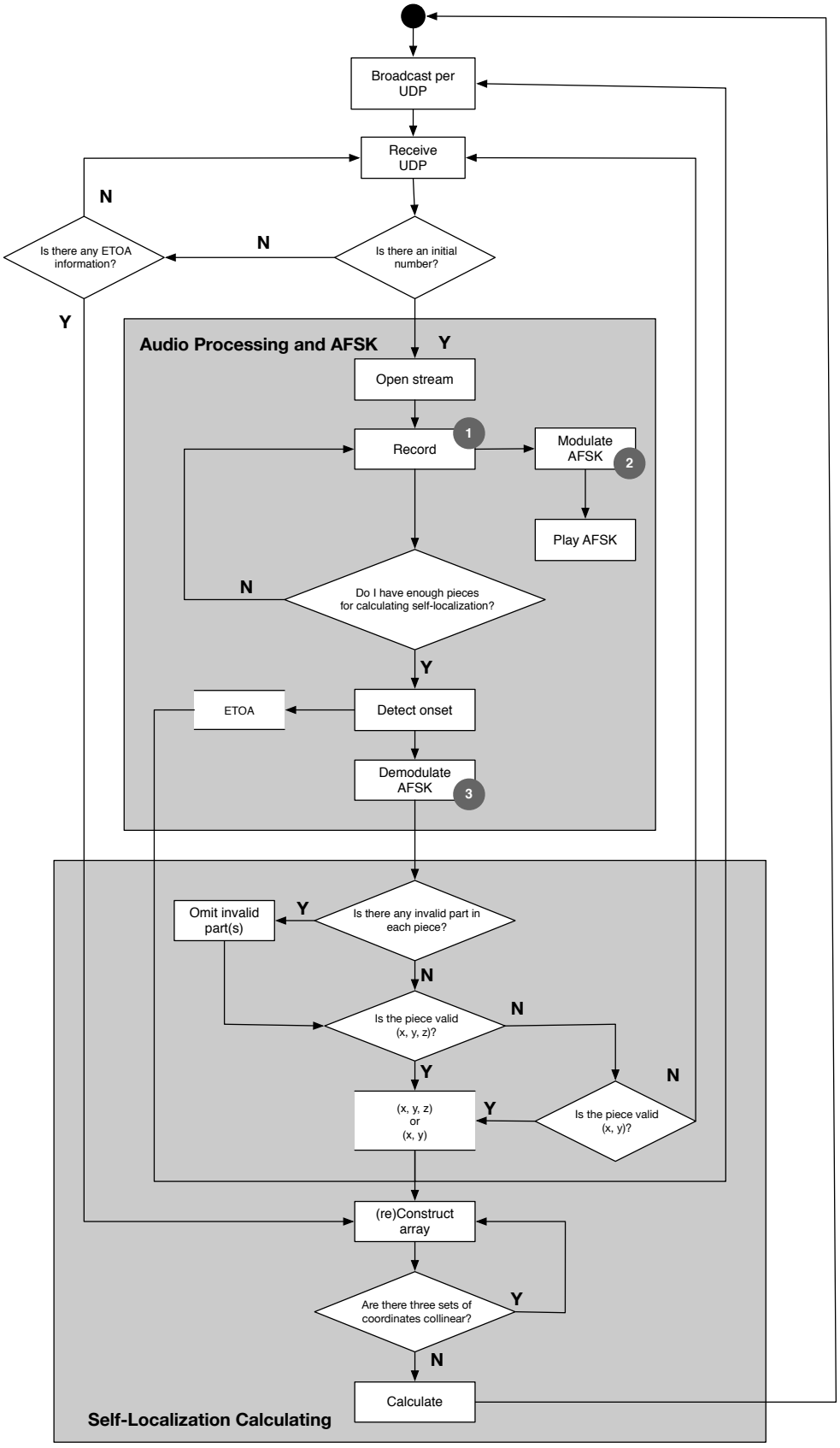

Figure 5.9: An illustration for integration
1 contains subprocesses in Figure 4.2
2 contains subprocesses in Figure 3.2
3 contains subprocesses in Figure 3.4

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This thesis has combined Audio Frequency-shift Keying (AFSK) as robot communicating method to exchange location information and Time Difference of Arrival (TDoA) algorithm for self-localization, and attends to the characteristics of NAO robot and the SPL rule book [18] in tending to explore a novel approach for data exchanging and self-localization during RoboCup game to overcome the limitation of conventional UDP communicating and be free from the vision-based localization and estimate more reliable position results.

In the Audio Frequency-shift Keying (AFSK) modulating as well as demodulating procedure, a Forward Error Correction technique named Hamming(7,4) is involved in both to provide a robust decoding procedure due to the interference during audio signal propagating through the air, particularly by RoboCup gaming. Moreover in AFSK demodulating many Digital Signal Processing (DSP) functions are utilised, they enforce the AFSK demodulating process which shows robustness under complicated sound environment. But a meaningful AFSK demodulating result is restricted by the sensitivity of the transducers involved and the characteristics of sound wave itself. The energy decay is inevitable in reality as the length propagation path increasing.

A time-based localization algorithm, Time Difference of Arrival (TDoA), for acoustic self-localization aiming at autonomous soccer robots is implemented in this thesis with two associating algorithms, Total Least Square and Maximum Likelihood, to achieve different accuracy level under 2D or/and 3D condition. The simulation results estimate loose localization and perform robustness. But a precise localization result of unknown is merely lying on the known information, aka. the known positions and the range measuring accurate level. In this thesis, two methods, synchronization with

sound onset detection in a Master/Slave networking and the BeepBeep Ranging System, are deployed and tested. By testing the first approach, the high density of information exchanging per Wi-Fi among NAO robots shows uncertainty due to the network congestion, thusly a successful ToA of acoustic signal can not be achieved. In the testing of ranging method BeepBeep, the static testing results are accurate as same as they announced in paper and the proposed algorithm for peak-detection, Cross-Correlation, gives a better estimation of ranging measuring than a tedious peak detection on amplitude. But the Cross-Correlation requires high computation which is not processed in real-time on NAO robot. In order to reach a balance of complexity between preprocessing and detection function, first a simplified peak detection is deployed on the signal receiver side to determine the onset and a Cross-Correlation algorithm followed to seek a specific signal in a small range. The ranging results are limited because the distorted sound occurs in playing or recording in NAO robots.

The acoustic localization and communication based on AFSK method perform poorly, the hardware obstacles are a gap between theoretical and practical feasibility. Besides it also restricted by the characteristics of sound wave itself. Therefore, to improve the loose localization based on acoustics, a better transducers set in NAO robot is the first prerequisite.

However instead of accurate acoustic localization, the implemented self-localization system for autonomous soccer robots can hand out loose or block localization estimations within 1 minute for entire 6 robot players including AFSK information encoding and decoding, also self-localization process.

6.2 Future Work

In this thesis many practical and experimental methods for comparing, testing and implementing a real-time acoustics localization system on NAO robot have been trailed and adopted, to find an appropriate solution matching the limitation in hardware of NAO robot and also achieving as best as possible localization estimation results. These trails and experiments are not covering Artificial Intelligence (AI) fields. But there are many possibilities introducing Artificial Intelligence (AI) into the self-localization process, such as during preparing TDoA localization the geographic or geometric measuring is unavoidable, in this thesis only two methods have been mentioned in Section 5.2. The physical distance measurement can be accomplished by machine learning algorithm too. The robots can study first about the relationship between distance changing and Sound Pressure Level (SPL) varying [34], and then by using regression and linearisation the

distance between sound source and itself can be evaluated. This method also does not require time synchronisation and hardware calibration.

All scientific approaches are using known to explore and deduce unknown. The estimations of unknown position are based on the known positions which is given out by manual measuring during laboratory stage and by the existing Particle Filter and Kalman Filter during RobotCup competition in the future. Weighting the estimated positions from different localization methods and taking advantage of prior knowledge to gain a more trustful and accurate estimation for location is another challenge.

Hence future work will focus on improving localization estimation using prior knowledge and introducing AI algorithms into self-localization process.

Bibliography

[1] WEI, CHANGYUN, JUNCHAO XU, CHANG WANG, PASCAL WIGGERS and KOEN HINDRIKS: *An approach to navigation for the humanoid robot nao in domestic environments*. In *Conference Towards Autonomous Robotic Systems*, pages 298–310. Springer, 2013. 1

[2] FOJTUU, SIMON, MICHAL HAVLENA and TOMAS PAJDLA: *Nao robot localization and navigation using fusion of odometry and visual sensor data*. In *International Conference on Intelligent Robotics and Applications*, pages 427–438. Springer, 2012. 1

[3] ELMOGY, MOHAMMED and JIANWEI ZHANG: *Robust real-time landmark recognition for humanoid robot navigation*. In *Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference*, pages 572–577. IEEE, 2009. 1

[4] DAVISON, ANDREW J: *Real-time simultaneous localisation and mapping with a single camera*. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference*, pages 1403–1410. IEEE, 2003. 1

[5] OSSWALD, STEFAN, ARMIN HORNUNG and MAREN BENNEWITZ: *Learning reliable and efficient navigation with a humanoid*. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2375–2380. IEEE, 2010. 1

[6] HAMBOECK, THOMAS and DIETMAR SCHREINER: *Austrian Kangaroos Team Research Report 2014 [Report about acoutic trail Whistle in NAO]*. Technical Report, Vienna University of Technology and University of Applied Sciences Vienna, 2015. 2, 42

[7] FRANK, ROBERT L: *History of Loran-C*. Navigation, 1982. 2, 19

[8] PRIYANTHA, NISSANKA BODHI: *The cricket indoor location system*. PhD thesis, Massachusetts Institute of Technology, 2005. 2

[9] CECH, JAN, RAVI MITTAL, ANTOINE DELEFORGE, JORDI SANCHEZ-RIERA, XAVIER ALAMEDA-PINEDA and RADU HORAUD: *Active-speaker detection and localization with microphones and cameras embedded into a robotic head*. In *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on*, pages 203–210. IEEE, 2013. 2

[10] LI, XIAOFEI, LAURENT GIRIN, FABIEN BADEIG and RADU HORAUD: *Reverberant sound localization with a robot head based on direct-path relative transfer function*. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 2819–2826. IEEE, 2016. 2

[11] ALDEBARAN-ROBOTICS: *NAO humanoid robot Software 1.14.5 documentation*. `http://doc.aldebaran.com/1-14/`. 2

[12] SOFTBANK ROBOTICS EUROPE: *hardware_speakerposition.png*. `http://doc.aldebaran.com/1-14/_images/hardware_speakerposition.png`, 2016. 2

[13] ALDEBARAN-ROBOTICS: *Motherboard*. `http://doc.aldebaran.com/2-1/family/robots/motherboard_robot.html`. 2, 27

[14] WATSON, BOB: *FSK: signals and demodulation*. Tech Notes, 1980. 8, 9, 25, 44

[15] GOERTZEL, GERALD: *An Algorithm for the Evaluation of Finite Trigonometric Series*. The American Mathematical Monthly, 65(1):34–35, 1958. 9

[16] OPPENHEIM, ALAN: *RES.6-008 Digital Signal Processing*. Spring 2011. Massachusetts Institute of Technology: MIT OpenCourseWare. 16, 75

[17] PENG, CHUNYI, GUOBIN SHEN, YONGGUANG ZHANG, YANLIN LI and KUN TAN: *Beepbeep: a high accuracy acoustic ranging system using cots mobile devices*. In *Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 1–14. ACM, 2007. 16, 23, 54, 55

[18] COMMITTEE, ROBOCUP TECHNICAL: *RoboCup Standard Platform League (NAO) Rule Book*. `http://www.tzi.de/spl/pub/Website/Downloads/Rules2017.pdf`, 2016. 19, 25, 30, 43, 65

[19] EL GEMAYEL, NOHA, SEBASTIAN KOSLOWSKI, FRIEDRICH K JONDRAL and JOACHIM TSCHAN: *A low cost tdoa localization system: Setup, challenges and*

results. In *Positioning Navigation and Communication (WPNC), 2013 10th Workshop on*, pages 1–4. IEEE, 2013. 21

[20] OFFICE, ENGINEER-IN-CHIEF'S: *Technical Pamphlet for Workmen - The Baudot Multiplex Printing-Type System*, 1919. 25

[21] EUROPE, SOFTBANK ROBOTICS: *hardware_microposition.png*. `http://doc.aldebaran.com/1-14/_images/hardware_microposition.png`, 2016. 25

[22] SAVITZKY, ABRAHAM and MARCEL JE GOLAY: *Smoothing and differentiation of data by simplified least squares procedures.* Analytical chemistry, 36(8):1627–1639, 1964. 30

[23] FOY, WADE H: *Position-location solutions by Taylor-series estimation*. IEEE Transactions on Aerospace and Electronic Systems, (2):187–194, 1976. 31

[24] FANG, BERTRAND T: *Simple solutions for hyperbolic and related position fixes*. IEEE transactions on aerospace and electronic systems, 26(5):748–753, 1990. 32

[25] LAARAIEDH, MOHAMED, STEPHANE AVRILLON and BERNARD UGUEN: *Overcoming singularities in TDoA based location estimation using total least square*. In *Signals, Circuits and Systems (SCS), 2009 3rd International Conference on*, pages 1–4. IEEE, 2009. 32, 47, 48

[26] CHAN, YIU TONG and KC HO: *A simple and efficient estimator for hyperbolic location*. IEEE Transactions on signal processing, 42(8):1905–1915, 1994. 33, 34, 49

[27] CASPER WILSTRUP, LARS IMMISCH: *alsaaudio documentation*. `https://larsimmisch.github.io/pyalsaaudio/`. 42, 43

[28] FAGERNESS, TRAVIS: *FSK Explained with Python*. `https://www.allaboutcircuits.com/technical-articles/fsk-explained-with-python/`, 2015. 44

[29] ROTTERDAM, V2_LAB: *A Python Implementation of Chan's TDoA algorithm for Ultrasonic Positioning and Tracking*. Technical Report, Stock, V2_Lab Rotterdam, 2008. 49

[30] BECHON, PATRICK and JEAN-JACQUES SLOTINE: *Synchronization and quorum sensing in a swarm of humanoid robots*. arXiv preprint arXiv:1205.2952, 2012. 52

[31] LIU, YUNHAO, ZHENG YANG, XIAOPING WANG and LIRONG JIAN: *Location, localization, and localizability*. Journal of computer science and technology, 25(2):274–297, 2010. 54

[32] STEFAŃSKI, JACEK: *Hyperbolic position location estimation in the multipath propagation environment*. Wireless and Mobile Networking, pages 232–239, 2009. 58

[33] ALDEBARAN-ROBOTICS: *NAO KEY FEATURE Audio Signal Processing*. https://www.generationrobots.com/media/NAO%20Next%20Gen/FeaturePaper(AudioSignalProcessing)%20(1).pdf. 60

[34] ALEXANDER SENGPIEL, SOHN VON EBERHARD SENGPIEL.: *Damping of sound level (decibel dB) vs. distance*. http://www.sengpielaudio.com/calculator-distance.htm. 66

[35] SMITH, ADAM, HARI BALAKRISHNAN, MICHEL GORACZKO and NISSANKA PRIYANTHA: *Tracking moving devices with the cricket location system*. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 190–202. ACM, 2004.

[36] SEO, KISUNG and ALDEBARAN ROBOTICS: *Using NAO: introduction to interactive humanoid robots*. Aldebaran Robotics.

[37] PUNSKAYA, ELENA: *Design of FIR Filters*. http://www.vyssotski.ch/BasicsOfInstrumentation/SpikeSorting/Design_of_FIR_Filters.pdf.

[38] LYONS, RICHARD G.: *Understanding Digital Signal Processing*. Prentice Hall, 2010.

[39] BEITER, MIKE, BRIAN COLTIN and SOMCHAYA LIEMHETCHARAT: *AN INTRODUCTION TO ROBOTICS WITH NAO*, volume 204 of *A STEM INTEGRATED, PROJECT BASED APPROACH TO LEARNING ROBOTICS AND COMPUTER SCIENCE*. Aldebaran Robotics, 2012.

[40] GARVEY, BOB: *Link Budget Analysis: Digital Modulation, Part 2*.

[41] DOWNEY, ALLEN B.: *Think DSP – Digital Signal Processing in Python*, volume 164. Green Tea Press, 2016.

[42] LI, XINYA, ZHIQUN DANIEL DENG, LYNN T RAUCHENSTEIN and THOMAS J CARLSON: *Contributed Review: Source-localization algorithms and applications using time of arrival and time difference of arrival measurements*. Review of Scientific Instruments, 87(4):041502, 2016.

[43] GLOVER, JOHN C, VICTOR LAZZARINI and JOSEPH TIMONEY: *Python for audio signal processing*, 2011.

[44] AMIOT, N., M. LAARAIEDH and B. UGUEN: *PyLayers: An open source dynamic simul ator for indoor propagation and localization*. In *Communications Workshops (ICC), 2013 IEEE Inter national Conference on*, pages 84–88, June 2013.

[45] HO, KC and YT CHAN: *Solution and performance analysis of geolocation by TDOA*. IEEE Transactions on Aerospace and Electronic Systems, 29(4):1311–1322, 1993.

[46] GILLETTE, MATTHEW D and HARVEY F SILVERMAN: *A linear closed-form algorithm for source localization from time-differences of arrival*. IEEE Signal Processing Letters, pages 1–4, 2008.

[47] IONESCU, RADU, RICCARDO CAROTENUTO, FABIO URBANI et al.: *3D localization and tracking of objects using miniature microphones*. Wireless Sensor Network, 2011.

[48] HECTOR, DIBIASE JOSEPH: *A high-accuracy, low-latency technique for talker localization in reverberant environments using microphone arrays*. PhD thesis, Brown University, 2000.

[49] BELLO, JUAN PABLO, LAURENT DAUDET, SAMER ABDALLAH, CHRIS DUXBURY, MIKE DAVIES and MARK B SANDLER: *A tutorial on onset detection in music signals*. IEEE Transactions on speech and audio processing, 13(5):1035–1047, 2005.

List of Figures

List of Tables

Abbreviations

TDoA	**T**ime **D**ifference **o**f **A**rrival
AoA	**A**ngle **o**f **A**rrival
FDoA	**F**requency **D**ifference **o**f **A**rrival
RSS	**R**adio **S**ignal **S**trength
ToA	**T**ime **o**f **A**rrival
LoS	**L**ine **o**f **S**ight
NLoS	**N**on **L**ine **o**f **S**ight
FSK	**F**requency **S**hift **K**eying
FFT	**F**ast **F**ourier **T**ransform
BFSK	**B**inary **F**requency **S**hift **K**eying
AFSK	**A**udio **F**requency **S**hift **K**eying
DSP	**D**igital **S**ignal **P**rocessing
VCO	**V**oltage **C**ontrolled **O**scillator
SPL	**S**ound **P**resure **L**evel
ADC	**A**nalog-to-**D**igital **C**onverter
FIR	**F**inite **I**mpulse **R**esponse
IIR	**I**nfinite **I**mpulse **R**esponse
LPF	**L**ow **P**ass **F**ilter
HPF	**H**igh **P**ass **F**ilter
BPF	**B**and **P**ass **F**ilter
SNR	**S**ignal-to-**N**oise **R**atio
RMS	**R**oot **M**ean **S**quare
ML	**M**aximum **L**ikelihood
LS	**L**east **S**quare
TLS	**T**otal **L**east **S**quare
TSLS	**T**wo-**S**tage **L**east **S**quare
FEC	**F**orward **E**rror **C**orrection
PCM	**Pulse**-**C**ode **M**odulation
SVD	**S**ingular **V**alue **D**ecomposition
ALSA	**A**udio **L**inux **S**ound **A**rchitecture

Appendix A

Python Script For FSK Signal Generating

Python Script For FSK Signal Generating

soundencoder.py

```
1  import numpy as np
2
3  SAMPLE_RATE = 50
4  MAX_FREQUENCY = 10000
5  DEV_FREQUENCY = 1000
6  CARRIER_FREQUENCY = 2000
7  AMPL = 1
8
9
10 def encodeSound(data):
11
12     nbits = len(data)
13
14     t = np.arange(0, float(nbits)/float(SAMPLE_RATE), 1/float(MAX_FREQUENCY), dtype=
       np.float)
15     m = np.zeros(0).astype(float)
16
17     for bit in data:
18         if bit == 0:
19             m = np.hstack((m, np.multiply(np.ones(MAX_FREQUENCY/SAMPLE_RATE),
                   CARRIER_FREQUENCY+DEV_FREQUENCY)))
20         else:
21             m = np.hstack((m, np.multiply(np.ones(MAX_FREQUENCY/SAMPLE_RATE),
                   CARRIER_FREQUENCY-DEV_FREQUENCY)))
22
23     sig = AMPL * np.cos(2 * np.pi * np.multiply(m, t))
24
25     return sig
```