ABSTRACT

Title of Document: ROBOTIC SOUND SOURCE LOCALIZATION AND TRACKING USING BIO-INSPIRED MINIATURE ACOUSTIC SENSORS

Laith Sami Sawaqed, Doctor of Philosophy, 2013

Directed By: Professor Miao Yu
Department of Mechanical Engineering

Sound source localization and tracking using auditory systems has been widely investigated for robotics applications due to their inherent advantages over other systems, such as vision based systems. Most existing robotic sound localization and tracking systems utilize conventional microphone arrays with different arrangements, which are inherently limited by a size constraint and are thus difficult to implement on miniature robots. To overcome the size constraint, sensors that mimic the mechanically coupled ear of fly *Ormia* have been previously developed. However, there has not been any attempt to study robotic sound source localization and tracking with these sensors.

In this dissertation, robotic sound source localization and tracking using the miniature fly-ear-inspired sensors are studied for the first time. First, through investigation into the Cramer Rao lower bound (CRLB) and variance of the sound incident angle estimation, an enhanced understanding of the influence of the mechanical coupling on the performance of the fly-ear inspired sensor for sound localization is achieved. It is found that due to the mechanical coupling between the membranes, at its working frequency, the fly-ear inspired sensor can achieve an estimation of incident angle that is 100 time better than that of the conventional microphone pair with same signal-to-noise ratio in detection of the membrane deflection. Second, development of sound localization algorithms that can be used for robotic sound source localization and tracking using the fly-ear inspired sensors is carried out. Two methods are developed to estimate the sound incident angle based on the sensor output. One is based on model-free gradient descent method and the other is based on fuzzy logic. In the first approach, different localization schemes and different objective functions are investigated through numerical simulations, in which two-dimensional sound source localization is achieved without ambiguity. To address the slow convergence due to the iterative nature of the first approach, a novel fuzzy logic model of the fly-ear sensor is developed in the second approach for sound incident angle estimation. This model is studied in both simulations and experiments for localization of a stationary source and tracking a moving source in one dimension with a good performance. Third, nonlinear and quadratic-linear controllers are developed for control of the kinematics of a robot for sound source localization and tracking, which is implemented later in a mobile platform equipped with a microphone pair. Both homing onto a stationary source and tracking of a moving source with pre-defined paths are successfully demonstrated.

Through this dissertation work, new knowledge on robotic sound source localization and tracking using fly-ear inspired sensors is created, which can serve as a basis for future study of sound source localization and tracking with miniature robots.

ROBOTIC SOUND SOURCE LOCALIZATION USING BIO-INSPIRED MINIATURE ACOUSTIC SENSORS

By

Laith Sami Sawaqed

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in Partial fulfillment
of the requirements of the degree of
Doctor of Philosophy
2013

Advisor Committee:
Professor Miao Yu (Mechanical Engineering, Chair)
Professor Amr Baz, (Mechanical Engineering)
Professor Nikhil Chopra (Mechanical Engineering)
Professor Sarah Bergbreiter (Mechanical Engineering)
Professor Nuno Martins (Electrical Engineering, Dean's Representative)

DEDICATION

To…

my lovely wife, *Hala*

my parents, sisters and brother

… for their everlasting patience and support

and my son Ra’ad

... for the never-ending joy he brings to my life.

ACKNOWLEDGEMENTS

I am deeply grateful to my advisor, Professor Miao Yu, for her encouragement, insightful guidance. Her guidance through the whole research was valuable to me and opened the doors to new aspects that will be useful to my career future as a faculty member at the Jordan University of Science and Technology. I am also very grateful to Dr. Yu for providing me the chance to have a great experience by participating in the ASME 2012 conference.

I am also grateful to my Ph.D supervisory committee members, Professor Amr Baz, Professor Nikhil Chopra, Professor Sarah Bergbreiter, and Professor Nuno Martins for your willingness to serve on my committee, providing very important and useful suggestions on the proposal to come up with a well-established work, and for reviewing this dissertation.

Special thanks to Professor Amr Baz for his great help and support.

I am also thankful to Prof. Sean Humbert for his hospitality by using his facility to conduct my experiments.

My thanks also go to all the members in the Sensor and Actuators Lab (SAL) for their support and friendship. Special thanks to my friends and colleagues Hector Escobar, Nick Vlajic, and Rubyca Jaai for their support and very useful suggestions in my work. I also would like to thank my friends Professor Khalid Oweis and

Professor Sulaiman Sweis and their families for being extremely supportive during my study and stay in the DC area.

Last but not least, I would like to acknowledge the unlimited support received from my wife Hala, my son Ra'ad for their sacrifices. And my parents, sisters, and brother whose prayers, guidance and never-ending love have been a lifetime blessing.

Always, it is impossible to remember all, and I apologize to those I have inadvertently left out.

Table of Contents

List of Tables

List of Figures

Nomenclature

Acronyms

2D	Two dimensions
2PA	Two-point algorithm
3D	Three dimensions
3PA	Three-point algorithm
ADC	Analogue to Digital converter
ANFIS	Adaptive Neuro Fuzzy Inference System
AOA	Angle of arrival
CRLB	Cramer Rao lower bound
DOA	Direction of arrival
DOF	Degree of freedom
EKF	Extended Kalman filter
FIM	Fisher information matrix
FIS	Fuzzy Inference System
FL	Fuzzy Logic
GCC	Generalized cross correlation
GP	Grid Partitioning
HYC	Quadratic-linear Controller
ILD	Interaural level difference
IPD	Interaural phase difference
ITD	Interaural time difference
MAX	Maximum
mDS	Mechanical directional sensitivity
mIID	Mechanical interaural intensity difference

mILD	Mechanical interaural level difference
MIN	Minimum
mIPD	Mechanical interaural phase difference
mITD	Mechanical interaural time difference
ML	Maximum likelihood
NLC	Nonlinear Controller
NWA	Non-weighted algorithm
NW2PA	Non-weighted two-point algorithm
NW3PA	Non-weighted three-point algorithm
PD	Phase difference
PDF	Probability density function
SC	Subtractive Clustering
SNR	Signal to noise ratio
SRP	Steered response power
TD	Time difference
TDOA	Time difference of arrival
WA	Weighted algorithm
WTAVER	Weighted average
W2PA	Weighted two-point algorithm
W3PA	Weighted three-point algorithm

Chapter 1 Introduction and Background

1.1 Problem of Interest

Sound source localization is one of the most interesting problems, which have been used in many applications such as hearing aids [1], underwater wireless sensor networks, tactical surveillance systems, robotics [2–20], and audio/video conferences [21], [22]. To date, sound source localization in all above mentioned applications are achieved by using conventional microphone arrays. These arrays usually consist of a number of microphones ranging from 2 to 32 [2–20], which are arranged in two dimensions (2D) or three dimensions (3D) to localize and/or track sound sources [23]. These arrangements are typically chosen through conducting analytical analysis on improving the localization accuracy [24]. Among many different methods, the most widely used method to analyze the localization accuracy is based on the Cramer Rao lower bound of the array for sound localization based on the time difference of arrival (TDOA) information. It has been shown that the microphones arranged in one of the five symmetrical 3D platonic solids shapes, namely, tetrahed, ocahed, cube, icosahed, and dodecahed [23], can help achieve the best sound source localization accuracy. However, the drawbacks of using such arrays are the large sizes and high computational complexity for extracting the directional cues. The large size is attributed to the fact that the separation distance between any pair of microphones in the array should not exceed half the minimum wavelength that presents in the propagated sound wave [25]. This condition is necessary to ensure a detectable time/phase difference between the microphones signals that are utilized to estimate the sound direction.

To overcome these drawbacks, bio-inspired ideas have been investigated and used in developing sound localization sensors with different configurations, characteristics, and sizes [26–31]. Miniature directional microphones that mimic the ear of the parasitoid fly *Ormia* are one of these sound localization sensors. Micro-Electro-Mechanical-Systems (MEMS) fabrication technology has been used to fabricate these sensors. Due to the small size and high performance of these sensors, they can become excellent choices for miniature robots to perform sound localization and tracking tasks. However, there has not been any effort to use these sensors for such applications.

This dissertation work is aimed to achieve a fundamental understanding of the fly-ear inspired acoustic sensors for sound localization and tracking, and to develop localization algorithms and control strategies for robotic sound localization and tracking using the bio-inspired acoustic sensors.

1.2 Literature Review

1.2.1 Sound Source Localization Approaches

Currently acoustic localization systems that are used for sound localization in an environment are mainly based on two approaches (see Figure 1.1). The first approach is a direct approach, in which the most popular technique is referred to as the steered beamforming. The second approach is an indirect approach, in which the time difference of arrival (TDOA) of the sound wave received from two microphones in a microphone array is estimated and utilized to determine the angle of arrival (AOA) of the sound [32].

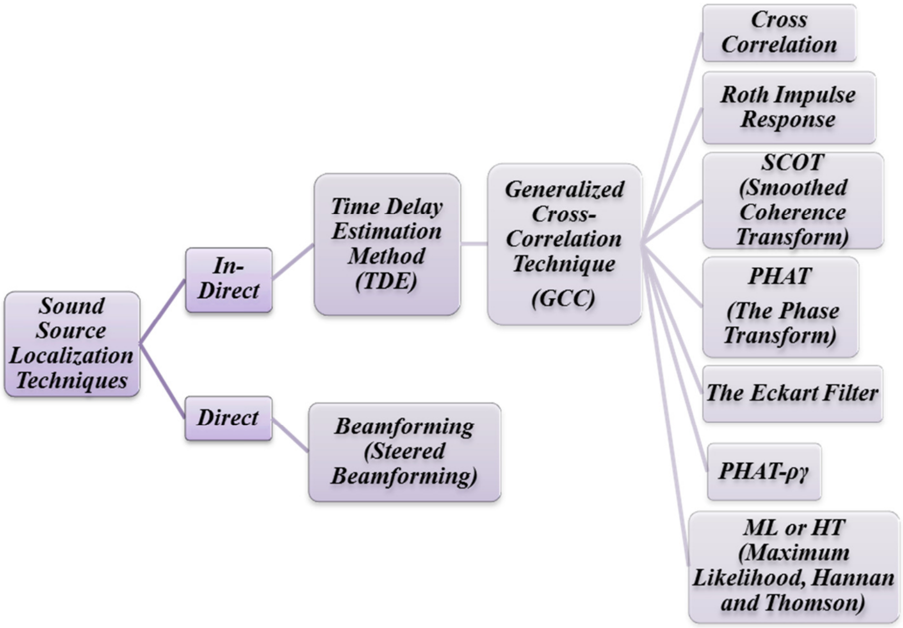

Sound Source Localization Techniques
In-Direct
Direct
Time Delay Estimation Method (TDE)
Beamforming (Steered Beamforming)
Generalized Cross-Correlation Technique (GCC)
Cross Correlation
Roth Impulse Response
SCOT (Smoothed Coherence Transform)
PHAT (The Phase Transform)
The Eckart Filter
PHAT-ργ
ML or HT (Maximum Likelihood, Hannan and Thomson)

Figure 1.1: Sound source localization techniques.

1.2.1.1 Direct Sound Source Localization

The most popular direct technique is the beamforming. Beamforming is based on a filter-and-sum process, which applies some temporal filters to the microphone signals before summing them to produce a single, focused signal [33]. The role of these filters is to enhance the signals from the desired sources and cancel or attenuate the signals from the undesired sources. This is achieved by performing time shifts to the detected signals then summing them up to localize the sound source. These filters can be chosen according to the nature of the sound source signal and the noise type [33]. Practically, the sound source location is usually unknown. Therefore, a beamformer can steer by following a predefined spatial region and by modifying the steering delays the source can be localized. This technique is called the steered beamforming, as illustrated in Figure 1.2.

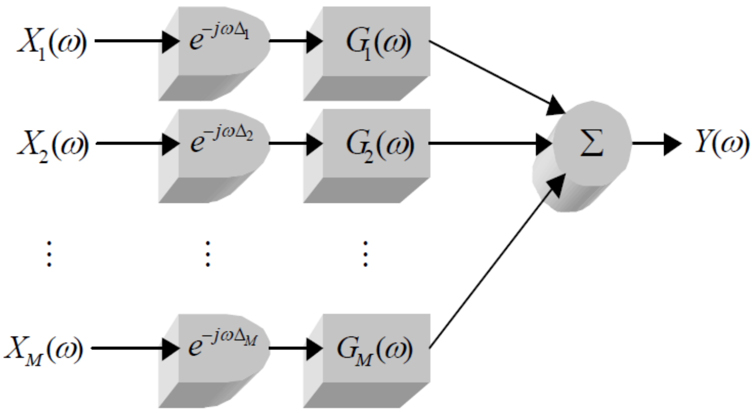


Figure 1.2: The structure of a filter-and-sum beamformer [33].

The frequency response of the M-element filter-and-sum beamformer is defined as:

$$Y\left(\omega, \Delta_1, \ldots, \Delta_M\right) = \sum_{m=1}^{M} G_m\left(\omega\right) X_m(\omega) e^{-j\omega\Delta_m} \quad , \tag{1.1}$$

and

$$\begin{aligned} x_1(t) &= s_1(t) + n_1(t) \\ x_i(t) &= \alpha s_1(t + D_i) + n_i(t), \quad i = 2, 3, \ldots, M \end{aligned} \tag{1.2}$$

where Δ_i is the determined steering delay to pinpoint the sound source, $s_1(t)$ is the time signal from the reference microphone, α is the spatial attenuation coefficient, D_i is the TDOA, $n_i(t)$ is the noise associated with the signal from sensor i, $X_i(\omega)$ is the Fourier transform of the microphones signals $x_i(t)$, and $G_i(\omega)$ is the Fourier transform of the temporal filter. For the case that the temporal filters are not used, the technique is called delay-and-sum beamformer. For the case that the temporal filters are used, the technique is called filter-and-sum beamformer. The output power of the filter-and-sum beamformer is called steered response power (SRP), which can be obtained as

$$P(\Delta_1 \ldots \Delta_M) \equiv \int_{-\infty}^{+\infty} Y(\omega, \Delta_1 \ldots \Delta_M) Y^*(\omega, \Delta_1 \ldots \Delta_M) d\omega \tag{1.3}$$,

where $Y(\omega,\Delta_1 \ldots \Delta_M)$ is the output of the filter-and-sum beamformer, $Y^*(\omega,\Delta_1 \ldots \Delta_M)$ is its complex conjugate, and $\Delta_1 \ldots \Delta_M$ are the steering delays that maximize the SRP. If the source location vector in space is $\vec{d}$, the SRP can be represented as:

$$P\left(\vec{d}\right) = P(\Delta_1 \ldots \Delta_M) \tag{1.4}$$.

Here, Δ_M is the propagation time delay between two sensors (microphones), which can be represented mathematically as:

$$\Delta_M = \tau_o - \frac{|\vec{d}_m - \vec{d}|}{c} \tag{1.5}$$.

For a far-field condition, the propagation delay can be expressed in terms of the assumed sound propagation direction $\vec{\zeta}_o$ as:

$$\Delta_M = \frac{-\vec{\zeta}_o . \vec{d}_m}{c} \tag{1.6}$$,

where $$-\vec{\zeta}_o \equiv \begin{bmatrix} cos\varphi sin\theta \\ cos\varphi cos\theta \\ sin\varphi \end{bmatrix} \tag{1.7}$$.

The angles (θ, φ) are the assumed direction of arrival, which can be used to find the propagation delays required to maximize Eq. (1.3) for localizing the source.

1.2.1.2 Indirect Sound Source Localization

Combining Eqs. (1.1) and (1.3) in the previous section with some manipulation, it can be shown that the SRP technique is basically a data averaging method. For the case of two microphone pairs, it is proved that SRP is equivalent to the generalized cross correlation (GCC), which is the most commonly used method for indirect sound source localization [33] that is represented as:

$$R_{lq}(\tau)=\frac{1}{2\pi}\int_{-\infty}^{+\infty}\Psi_{lq}(\omega)X_l(\omega)X'_q(\omega)e^{j\omega\tau}d\omega \quad , \tag{1.8}$$

where $R_{lq}(\tau)$ represents the GCC of the two microphone signals, and $\Psi_{lq}(\omega)$ represents the weighting function, which can be any of the weighting functions listed in Table 1-1. Figure 1.3 summarizes the different GCC methods used in the sound source localization.

Table 1-1: Generalized Cross Correlation (GCC) method weighting functions

GCC Method		
Name	Weight Ψ_{lq} (ω)	Reference
Cross Correlation	1	[34]
Roth Impulse Response	$\frac{1}{X_l(\omega)X'_q(\omega)}$	
SCOT	$\frac{1}{\sqrt{X_l(\omega)X'_l(\omega)X_q(\omega)X'_q(\omega)}}$	
PHAT	$\frac{1}{\lvert X_l(\omega)X'_q(\omega)\rvert}$	
Eckart	$\frac{S_l(\omega)S'_l(\omega)}{[N_l(\omega)N'_l(\omega)N_q(\omega)N'_q(\omega)]}$	
ML or HT	$\frac{\lvert\gamma_{lq}(\omega)\rvert^2}{\lvert X_l(\omega)X'_q(\omega)\rvert\left[1-\lvert\gamma_{lq}(\omega)\rvert^2\right]}$	
PHAT-ργ	$\frac{1}{\lvert X_l(\omega)X'_q(\omega)\rvert^{\rho}+\lvert\gamma_{lq}(\omega)\rvert^2}$	[35]

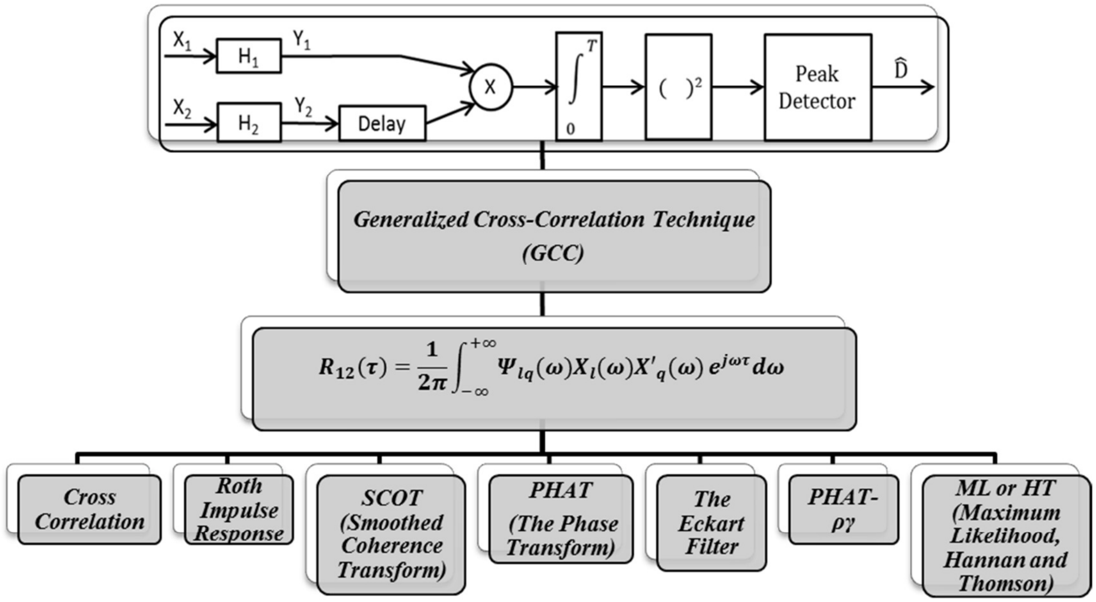

X_1
H_1
Y_1
X_2
H_2
Y_2
Delay
X
$\int_0^T$
$(\ )^2$
Peak Detector
$\hat{D}$
Generalized Cross-Correlation Technique (GCC)
$R_{12}(\tau) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \Psi_{lq}(\omega) X_l(\omega) X'_q(\omega)\, e^{j\omega\tau} d\omega$
Cross Correlation
Roth Impulse Response
SCOT (Smoothed Coherence Transform)
PHAT (The Phase Transform)
The Eckart Filter
PHAT-ργ
ML or HT (Maximum Likelihood, Hannan and Thomson)

Figure 1.3: Generalized Cross Correlation methods.

1.2.2 Localization of Single and Multiple Sound Sources

Robotic sound source localization problems have attracted much attention of many researchers due to the importance of audition systems for robotic perception, which does not need direct line of sight with the source [13]. Other systems, such as vision based systems, have limited capabilities that can lead to lack of information for robotic perception. The different methods discussed in the previous section have been used for robotic sound localization using different number of microphones and different microphone arrangements in audition systems. Localization of single and multiple, stationary and moving sound sources have been investigated intensively [2–22]. In the following subsections, detailed work on this aspect will be reviewed.

1.2.2.1 Localization of Stationary Single and Multiple Sound Sources

Huang *et al.* [8] studied a mobile robot equipped with a real time sound source localization system consisting of three microphones used in an echo-avoidance model to detect the echo–free portions of the coming sound. A sonar system was used to avoid the obstacles in the robotic navigation experiment, as shown in Figure 1.4. This system was shown to have great capability to localize the sound while avoiding the obstacles in an echoic environment.

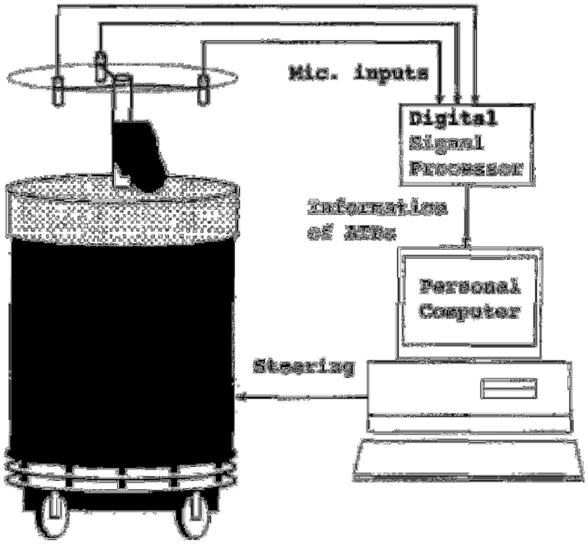


Figure 1.4: Schematic of a robot equipped with microphones for real time sound localization [8].

In another work, Valin *et al.* [13] presented a robust sound source localization system, using eight microphones arranged in three dimensions (3D) on a Pioneer 2 robot, as shown in Figure 1.5. The localization was based on time difference of arrivals and it was shown that the robot was capable to localizing different types of sound sources over a range of 3 meters with a localization precision of 3^{o}.

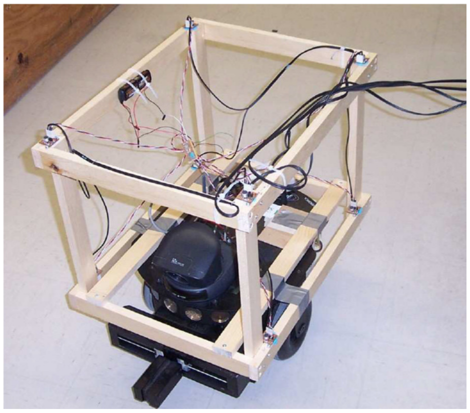
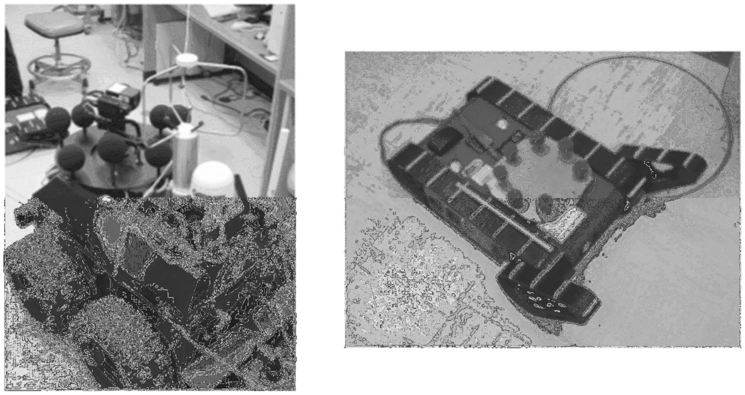


Figure 1.5: Microphone array installed on a Pioneer 2 robot for sound source localization [13].

In another work carried out by the U.S. Army Research Laboratory (ARL), acoustic sensors on robotic platforms were used to help the future combat systems to perform different types of military tasks, as shown in Figure 1.6 [20]. The robotic systems were tested for localizing acoustic sources such as gunshots in three different urban environments. The cross correlation method was used to extract the acoustic directions in azimuth and elevation to orient an infrared (IR) camera placed on top of the robot. Based on the measurements from different robots, triangulation was used to precisely locate the position of a sniper.



(a) (b)

Figure 1.6: (a) iRobot ATRV-2 robot with the acoustic array and (b) iRobot Urban Robot with 9" diameter microphone array [20].

In 2004, Andersson *et al.* reported a biomimetic sound source localization apparatus, as shown in Figure 1.7. The system was based on the principle that directional cues such as the interaural time difference (ITD) and the interaural level difference (ILD) obtained with two microphones are related to the diffraction about the head [5]. A performance metric was defined by using the directional cues. The angle indicating the sound source direction can be obtained by minimizing this metric using the least mean square between the measured and the theoretical directional cues. Localization of a single broadband sound source was achieved successfully with a localization error of ±2^o.

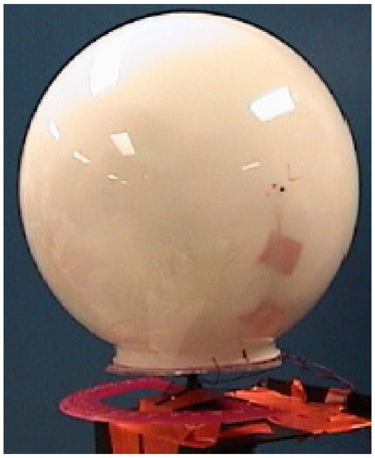


Figure 1.7:bio-inspired sound localization apparatus [5].

Moreover, Murray *et al.* demonstrated a robotic acoustic tracking system that used the interaural time difference, obtained by using the cross correlation method, to orient itself towards the sound source [9]. The robot used in this system was equipped with two ears (microphones) and sonar sensors, as shown in Figure 1.8. The sound was positioned at different angles relative to the head centerline and a high localization accuracy of better than 90% was achieved.

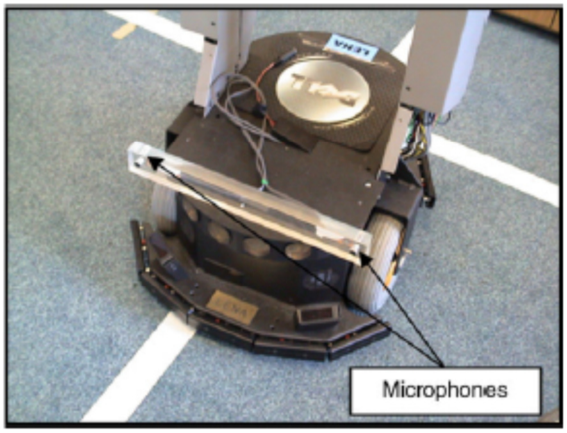


Figure 1.8: PeopleBOT with the two mounted microphones [9].

In 2005, Sasaki *et al.* reported a microphone array arrangement that is suitable for robotic navigation in an indoor environment [14]. The microphone array arrangement employed 32 microphones and the signals from these microphones showed small side lobes by using the sum and delay beamforming algorithm, discussed earlier, which can be used for sound localization. This arrangement led to a good localization performance and the capability of separation of multiple sources. The robot (Nomad-XR4000) equipped with this array (as shown in Figure 1.9) was capable of estimating the locations of three sound sources with a position estimation error of 20 cm.

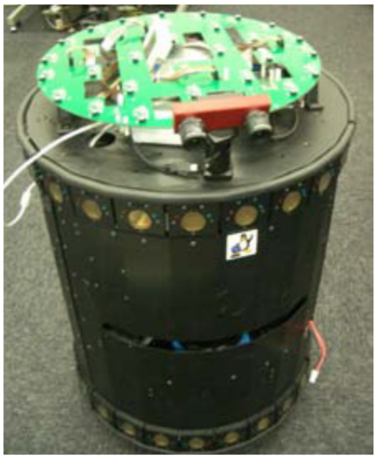


Figure 1.9: Nomad-XR4000 with the microphone array [14].

Another array (as shown in Figure 1.10) with 32 microphones arranged in a three ring configuration was developed by the same group to estimate the horizontal/vertical sound position [15]. The delay and sum beamforming method discussed earlier was used to localize the different sound sources. Separation of the source was achieved by using a frequency band selection algorithm. With this arrangement, localization of up to three sound sources can be achieved theoretically. Experimentally, two sound sources were localized with an angular error of 5^{o} and distance error of 20-30 cm at 1m distance.

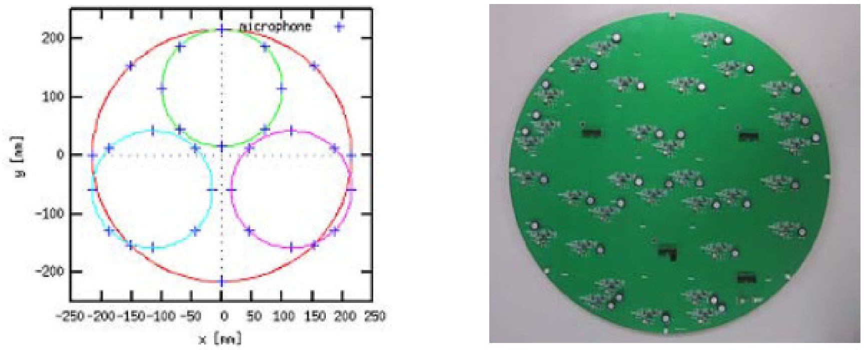


Figure 1.10: Microphone array arranged in three rings [15].

In 2009, the same research group developed another microphone array arrangement (as shown in Figure 1.11) so that a 2D sound source map can be obtained from the measured sound directions using the particle filtering algorithm [16]. The developed microphone array has 32 microphones with low side lobes and narrow main lobe. Localization of different sound sources was achieved by using the sum and delay beamforming method and the frequency band selection methods. The experimental results showed that the sound can be localized in 2D with an error of less than 50cm.

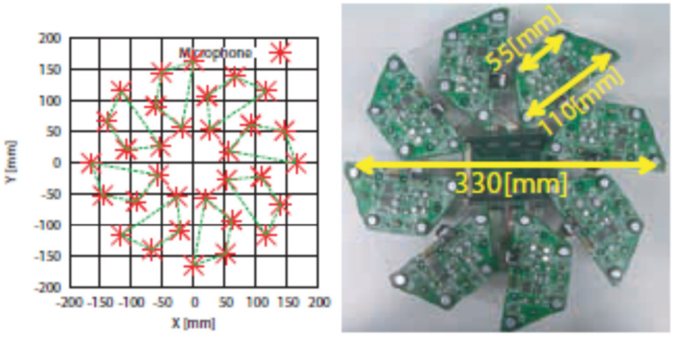


Figure 1.11: An array with 32 microphone arrangement [16].

In another work, Uchiyama *et al.* presented a mobile robot equipped with an auditory system for sound tracking and distance sensors for obstacles avoidance to follow a trajectory generated by a reference model [10]. Interaural time difference and sound level difference between the desired and the current position of the robot were used to control the robot. The system was tested in an indoor environment at different robot initial positions and different sound directions relative to the robot. The experiments showed that the robot (shown in Figure 1.12) was able to track the sound and avoid the obstacles during navigation and the navigation failed when the obstacle was high and the sound direction was 90 degrees since detecting the wall corner was difficult.

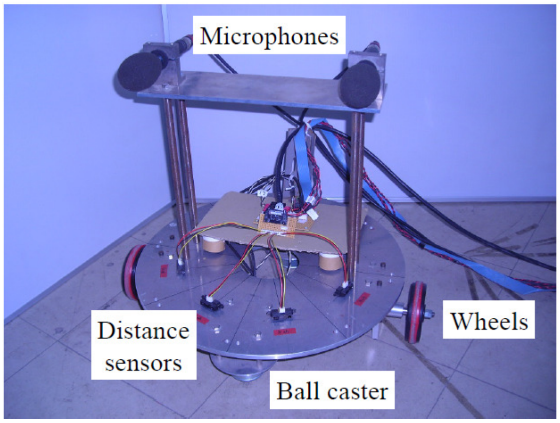


Figure 1.12: Two-wheeled robot with an auditory system [10].

In 2010, Arvin *et al.* presented a swarm robot (as shown in Figure 1.13) employed for exploring voice with a module that utilized four condenser microphones to capture the voice. The captured signals were processed to get the proportional signal strengths so as to estimate the sound orientation using a fuzzy logic approach [11]. Three separate experiments were conducted with different objectives. The first experiment aimed to test the developed module to localize the sound. The second experiment utilized one swarm robot to navigate randomly in the environment to measure the signal strengths and execute basic tasks like obstacle avoidance. In the third experiment, a number of swarm robots were used only one of them was served as the explorer for searching the sound source. The other robots would wait until the explorer found the source and transmit a message that the source is found to aggregate together after receiving the message.

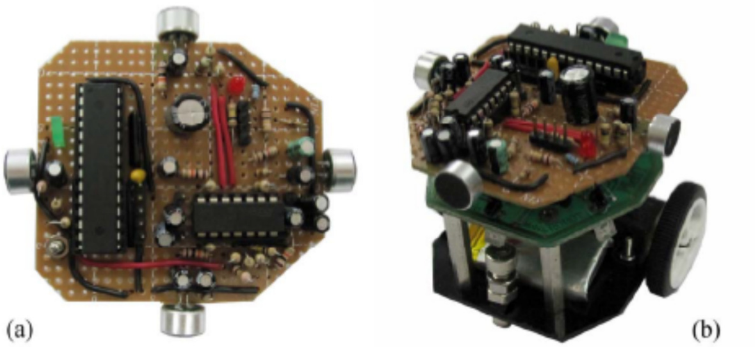


Figure 1.13: (a) Voice processor module and (b) robot equipped with the module [11].

More recently, in 2011, Sun *et al.* presented a rescue robot that can be used to search survivors in collapsed buildings after an earthquake occurs [6]. This robot was equipped with sonar sensors for obstacle avoidance and an auditory system that utilized three microphones, as shown in Figure 1.14. The sound direction is determined using the TDOA calculated for the array. Experiments showed a high accuracy in localizing the sound source.

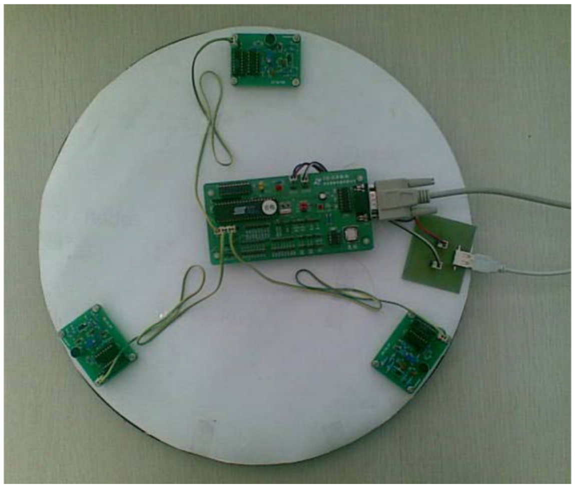


Figure 1.14: The experiment platform for searching survivors in collapsed buildings [6].

The platform, was improved later and a fourth microphone was added to have a tetrahedral shape arrangement of the microphones, as shown in Figure 1.15 [7]. The same localization algorithm was used with the new array to localize the sound with a priority to avoid the obstacles in the environment.

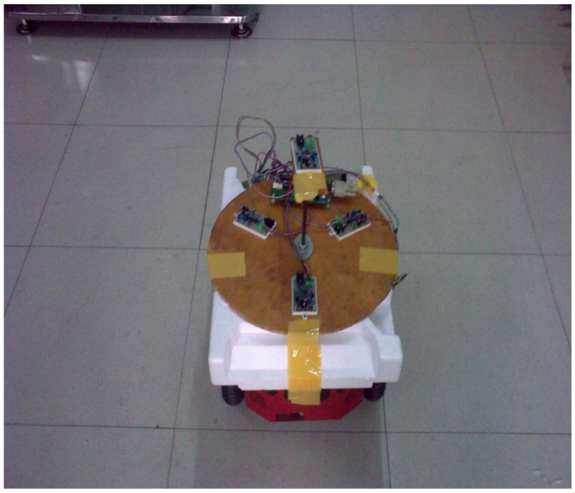


Figure 1.15: Improved survivor searching platform [7].

1.2.2.2 Localization of Moving Single and Multiple Sound Sources

Kumon and Uozumi presented a robot equipped with two microphones to provide binaural hearing, as shown in Figure 1.16 [4]. The calculated TDOA that takes into account the Doppler shift and the auditory system motion is used with the extended Kalman filter (EKF) to estimate the state of the dynamical system even with noisy measurements. Figure 1.17 shows the dynamics model of the system used.

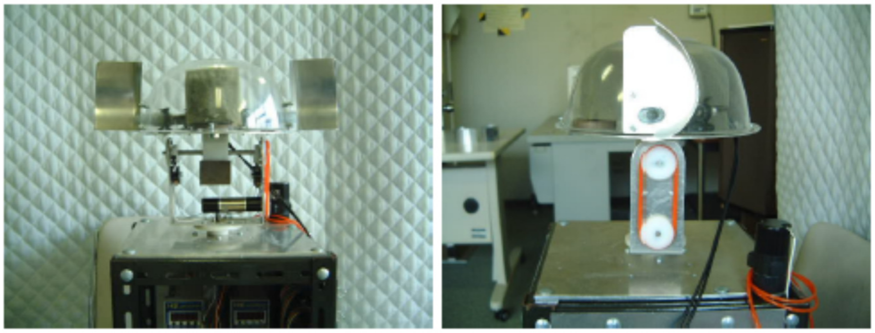


Figure 1.16: Robotic head with two microphones (front and side views) [4].

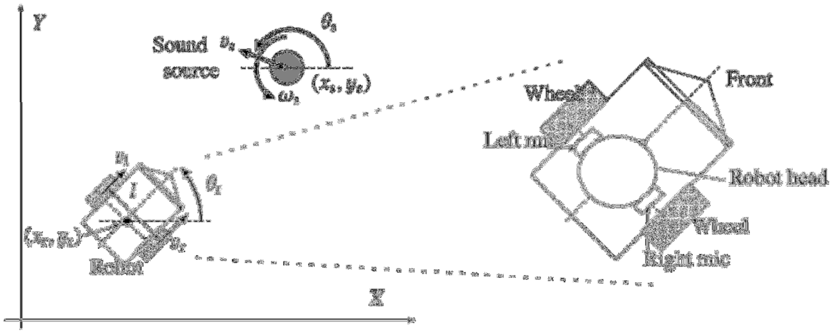


Figure 1.17: Dynamics model of the robot and the sound source (left) and detailed sketch of robot (right) [4].

Estimating the position of a moving sound source using the bearings-only measurements from a network of acoustical sensor arrays was investigated by Kaplan *et al.* [12]. The acoustical network geometry of the single node and multiple nodes are shown in Figure 1.18. Four maximum likelihood (ML) methods were developed, based on various simplifications of the target model, to estimate the moving target position.

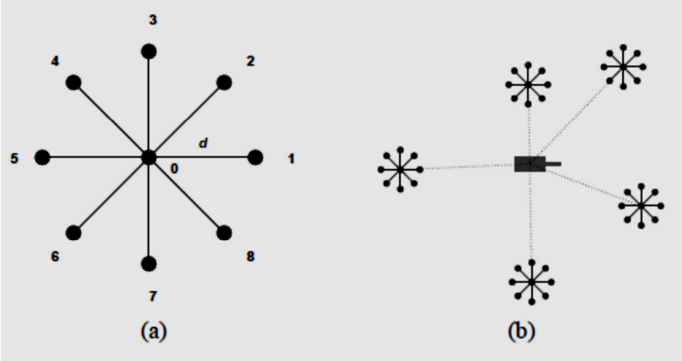


Figure 1.18: Geometry of the acoustical network: (a) The single node and (b) triangulation using multiple nodes [12].

More recently, in 2012, Han *et al.* presented a two wheeled robot equipped with three microphones that utilized the time delays and triangulation to track the sound of a moving object [36]. A Fuzzy Inference System was developed to drive the robot according to the estimated position and direction of the moving object.

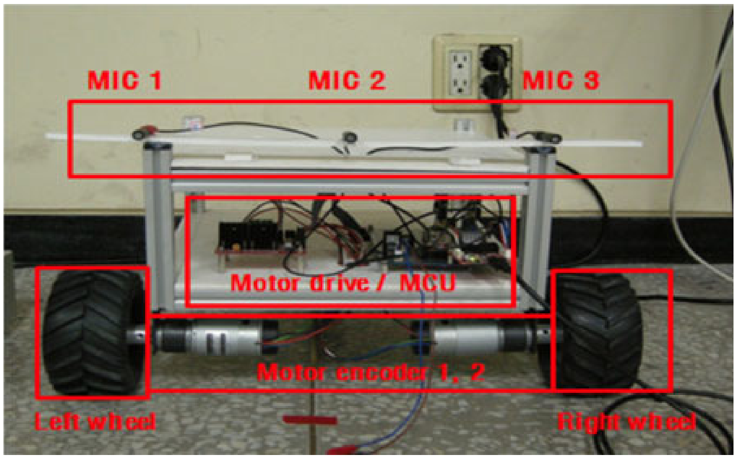


Figure 1.19: Mobile robot structure developed by Han *et al.* [36].

Although the above mentioned systems exhibited good capabilities to localize single and/or multiple, stationary and/or moving sound sources, the main limitation is the array size when applied to miniature robots. It should be noted that the separation distance between any pair of microphones in the array cannot be too small to obtain enough directional cues for sound localization even though this distance should be less than half the minimum wavelength present in the sound signal [25].

1.2.3 Bio-Inspired Acoustic Sensors

Bio-inspired sensors have received much attention during the last two decades. One type of sensors was inspired by the hearing organ (see Figure 1.20(a)) of the parasitoid fly Ormia Ochracea (tachinid family) which was first presented by Miles *et al.* [37], [38]. The fly-ear, as shown in Figure 1.20, consists of two membranes (Ipsilateral, and Contralateral) and a beam that connects the two membranes. The functionality of this beam is to provide a coupling between the two membranes, which will amplify the phase difference between the two membranes. The two membranes vibrate due to the acoustic pressure at a certain direction with an azimuth angle of θ from the midline of the two membranes. Although, the separation distance between the two ears is too small (~1.2 mm) to render enough directional cues at the acoustic stimulus level, the mechanical coupling between the two ears can help improve the directional hearing ability of the fly so that it can localize its host in azimuth and elevation directions. Studies also showed that fly's sound localization trajectory follows a saturation function, which suggests a localization/lateralization scheme. This means that the fly can only accurately localize the sound source if the sound is coming from orientations within ~20^{o} range. If the sound is beyond this range, the fly is only capable of distinguishing the direction of the sound (i.e. right or left).

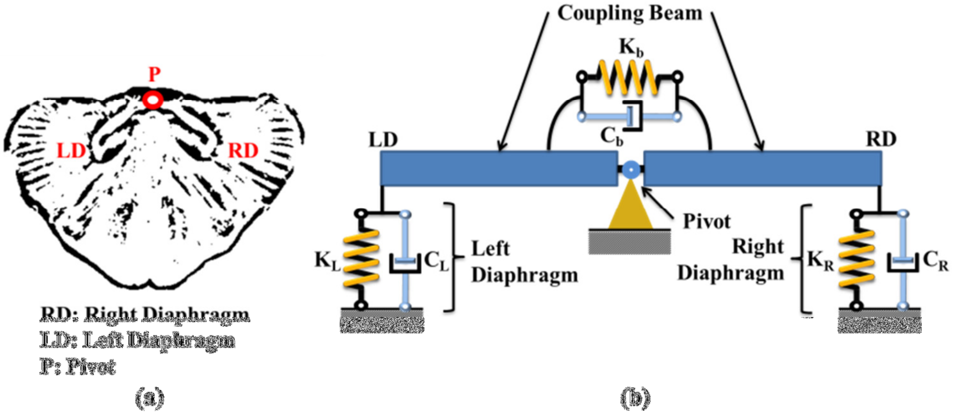


Figure 1.20: (a) Sketch of the fly-ear and (b) the 2-DOF model of the fly-ear.

Based on the two-degree-of-freedom (2-DOF) system model [37] of the fly-ear (see Figure 1.20(b)), a bio-inspired pressure gradient microphone was developed by Miles *et al.* which consists of two micro-machined polysilicon plates supported by a flexible pivot, as shown in Figure 1.21 . The acoustic pressure causes the two plates to deflect like a seesaw (rotational mode) or wings (translational mode). The deflections of the plates were detected by using an optical system demonstrated by Degertekin *et al.* [39] with a structure similar to the conventional capacitive acoustic sensor except that the back electrode has an optical diffraction grating that is used in the optical interferometer circuit. The measured light intensity change is a function of the deflection of the plates. The sensor has a small size (1 mm x 2 mm), which makes it possible to equip the sensor on miniature robots for sound source localization. However, there are several drawbacks of this sensor, which include the response amplitude dependence on the direction and the intensity of the sound, limited dynamic range, and thermal noise due to the squeeze film damping effect [39].

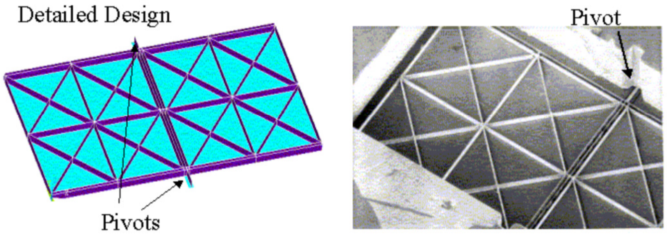


Figure 1.21: Bio-inspired pressure gradient microphone (Miles, *et al.* 2006) [26].

Another sensor design was presented by Saito *et al.,* which has a circular bronze diaphragm with a gimbal center support [27], as shown in Figure 1.22. The sensor structure with this design has three vibrational modes include one in-phase and two out-phase modes. Sound localization using this sensor was achieved in two-dimension (azimuth and elevation) using the information from the in-phase mode and the two out-phase modes of diaphragm deflections. The detection system used to detect the deflections of the diaphragm was a laser vibrometer. Although the size of this sensor is larger than the previous sensor (radius of 10.8 mm and thickness of 30 μm), it is still small enough for usage on small robots to localize a sound source. However, the problem in this sensor is the size of the detection system, which is even larger than the size of the robot.

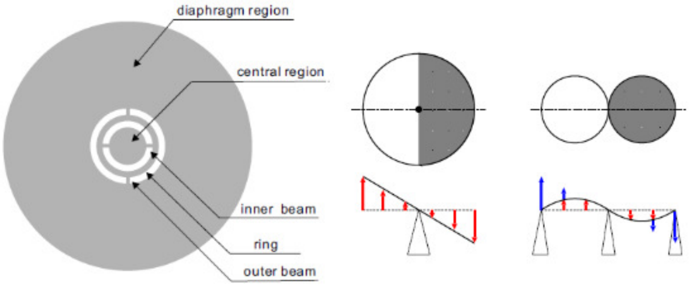


Figure 1.22: Directional microphone design presented by Saito *et al.* [27].

In another work, Currano *et al.* presented a design of a fly-ear inspired MEMS acoustic sensor that can be used to localize the sound in one dimension (azimuth) [28]. This sensor has two membranes coupled with a beam that amplifies the phase difference between the detected signals from the two membranes, as shown in Figure 1.23. The sensor structure with this design has also two vibrational modes. The first mode is the rocking mode in which the two membranes are 180^{o} out of phase and the second is the bending mode in which the two membranes are in phase, as shown in Figure 1.24. The deflections of the membranes were detected by using a low coherence fiber optic interferometer.

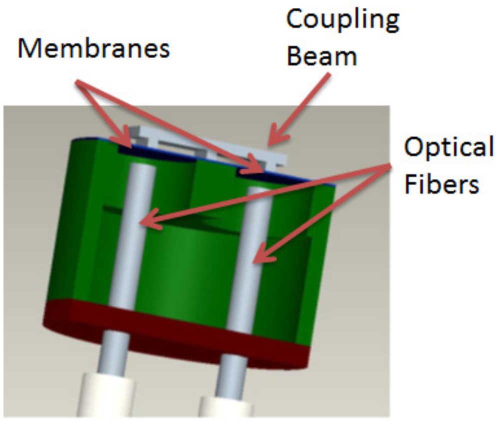
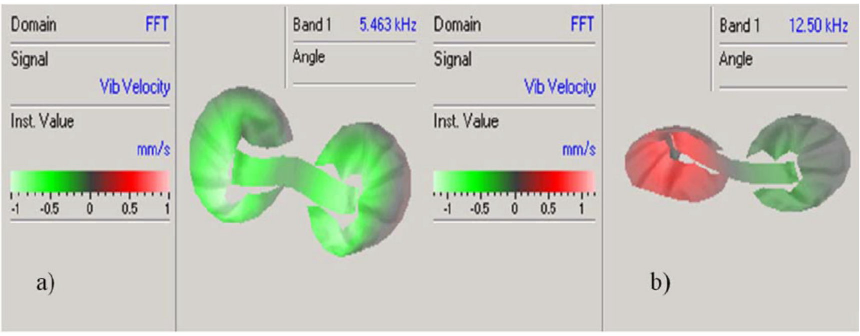


Figure 1.23: Bio-inspired MEMS optical acoustic sensor developed by Currano *et al.* [28].


Domain FFT
Signal Vib Velocity
Inst. Value mm/s
Band 1 5.463 kHz
Angle

Domain FFT
Signal Vib Velocity
Inst. Value mm/s
Band 1 12.50 kHz
Angle

Figure 1.24: Mode shapes of the fly-ear inspired sensor: (a) bending mode and (b) rocking mode [28].

An extension of the work of Currano *et al.,* another bio-inspired acoustic sensor design is presented by Lisiewski *et al.*, in which sound localization can be achieved in two dimensions (azimuth and elevation) [29]. This sensor has three membranes that are arranged in a triangular shape, as shown in Figure 1.25. The sensor structure has three vibrational modes: two rocking modes and one bending mode, as shown in Figure 1.26. The advantage of this sensor compared with the previous work is that the sound can be localized in 2-D (azimuth and elevation) and the power consumption due to measuring the deflections of three membranes is less than that of using two sensors with two-coupled membranes. The size of this device is about 2 mm x 2 mm. Owing to these advantages, a miniature robot can be equipped with this sensor to localize the sound in 2D.

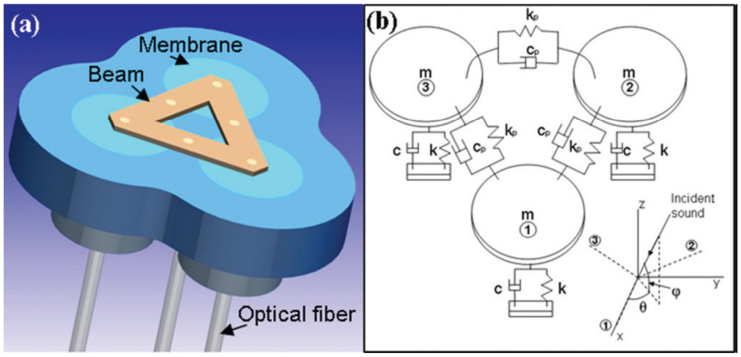


Figure 1.25: (a) Schematic of the fly-ear inspired sound localization sensor and (b) three-degree-of-freedom (3DOF) model of the sensor device [29].

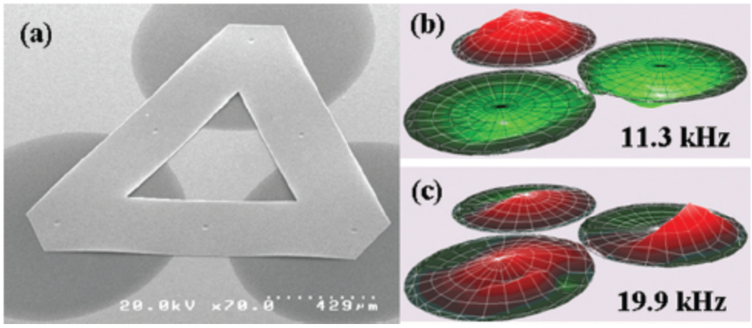


Figure 1.26: (a) SEM image of the micro-fabricated sensor device and mode shapes of the sensor device with (b) rocking mode at 11.3 kHz and (c) bending mode at 19.9 kHz [29].

Wang *et al.* presented a sensor that has three membranes that are coupled with rigid bars as shown in Figure 1.27 [30]. Laser displacement sensors were used to detect the membranes displacements. Specific device size was not mentioned in the paper but the separation distance is 50 mm, which gives an indication how big the device is compared with the MEMS devices presented by Currano *et al.* and Lisiewski *et al.* In this work, by using the same mechanics model, the stiffnesses and damping factors were investigated and the best values were chosen to increase the sensor sensitivity. Two localization methods of the fly were hypothesized. In the first method, the relation between the incident angle and the magnitude of transfer functions of any membrane pair displacements is used. The second approach is to find the incident angle in two separate steps. In the first step, the fly determines the side of the incident sound (ipsilateral or contralateral side). In the second step, the fly takes the advantage of the high sensitivity to the sound direction at the zero incident angle at a specific frequency.

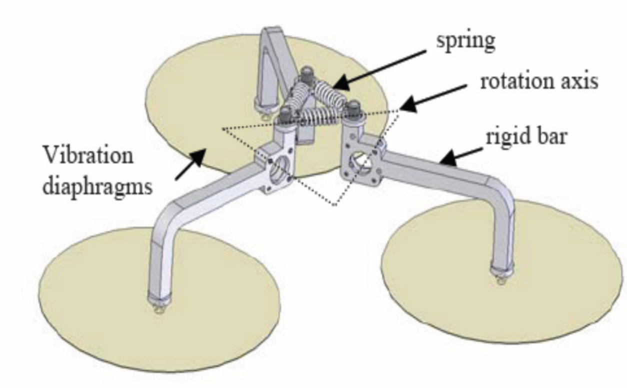


Figure 1.27: Schematic of the sound localization sensor developed by Wang *et al.* [30].

Chen and Cheng studied a biomimetic microphone designed with a central-supported (C-S) diaphragm, as shown in Figure 1.28 and Figure 1.29 [31]. The clover-stem-like C-S design was shown to achieve 47% improvement on net diaphragm displacement in comparison with the conventional C-S diaphragm design. Compared with the conventional C-S diaphragm, this design enables compensation of undesired deformation of sensing diaphragm due to gravity and residual stresses, and good diaphragm flexibility for better sound pressure sensitivity.

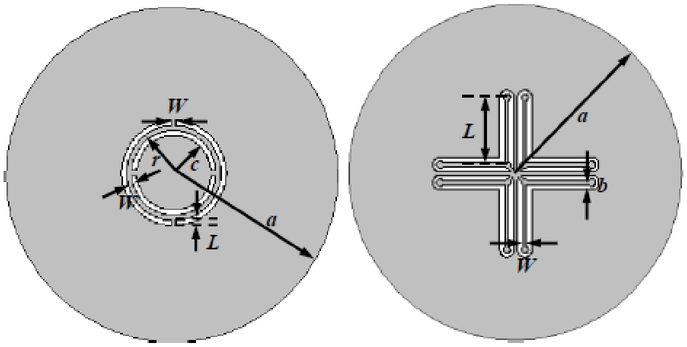


Figure 1.28: Conventional (left) and quadratic-linear (right) C-S diaphragms [31].

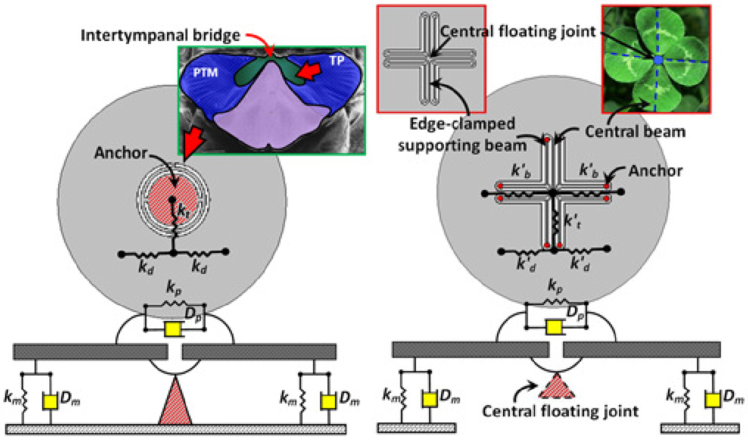


Figure 1.29: Schematic of central-supported gimbal circular biomimetic diaphragm (left), schematic of the quadratic-linear biomimetic microphone with central floating gimbal design achieving quadratic-linearized by the acoustic sensing mechanism of the parasitoid fly and the flexible clover-stem-like gimbal structure (right) [31].

1.3 Objectives and Scope of this Doctoral Research

The overall goal of the dissertation work is to investigate how to use the fly-ear inspired acoustic sensors for robotic sound source localization and tracking and develop a mobile robot platform along with localization algorithms and control schemes for sound source localization and tracking in an indoor environment. Specific objectives include the following:

1. Develop a fundamental understanding of using fly-ear inspired sensors in robotic sound localization and tracking.
2. Develop a localization algorithm based on model free gradient descent method to localize a stationary or moving sound source.
3. Develop a Fuzzy logic model of the fly-ear sensor to map the directional cues to the angle of arrival of the sound.
4. Investigate and develop nonlinear and quadratic-linear controllers to control the robot kinematics to localize a stationary and track a moving sound source.
5. Carry out experimental studies on a mobile robotic sound source localization platform for implementation and validation of the developed localization algorithms and controllers.

1.4 Organization of the Dissertation

The rest of the dissertation is organized as follows. In Chapter 2, detailed derivation of the CRLB for the coupled and uncoupled microphones is presented to show the effect of the coupling beam on improving the theoretical lower bound of the bearing angle estimation. In Chapter 3, making use of the fly-ear inspired sensor, different sound source localization algorithms are investigated, which utilize an optimization method, namely, model-free gradient descent method. In addition, a fuzzy logic model of the fly-ear inspired sensor is studied, which will be used to directly map the information extracted from the sensor, namely, the ITD to the angle of arrival (AOA) of the sound. Furthermore, experimental studies are conducted to compare the performance of localization and tracking of a single sound source using the developed fuzzy model to that using the conventional localization algorithms. In Chapter 4, nonlinear and quadratic-linear controllers are designed to control the robot kinematics to localize and track a stationary and moving single sound source in an indoor environment. In Chapter 5, a robotic platform equipped with a microphone pair, acting as a tracker, and another platform equipped with an omnidirectional speaker, acting as the sound source, are used to study robotic sound localization and tracking with a goal of demonstrating the developed algorithms and control schemes. In Chapter 6, a summary of the dissertation work is provided along with an outline for future work.

Chapter 2 Sensor Performance Analysis for Sound Source Localization

Arrangement of microphones in a microphone array is important in sound localization. Much effort has been made to find the best arrangement with a given number of microphones to achieve the desirable sound localization performance. The most popular criterion to determine the sound localization performance is based on the Cramer-Rao lower bound (CRLB) [12], [23–25] (see Appendix A for more details). The CRLB has been applied to evaluate many signal processing problems including range estimation (e.g. sonar, radar, and robotics) [40], [41], frequency estimation (e.g. sonar, radar, econometrics, and spectrometry) [42–44], bearing estimation (e.g. sonar and radar) [41], [45], [46], and autoregressive parameter estimation (e.g. speech and econometrics) [47]. The CRLB can be used in these estimation problems due to the fact that these problems are unbiased estimation problems [48].

In a previous study, Akcakaya and Nehorai presented a performance analysis of the fly ear using the CRLB [49]. In their work, the two-degree-of-freedom (2DOF) model of the fly ear was formulated in the state space and the frequency response was obtained accordingly. In the analysis, it was assumed that multiple sources were present with unknown incident angles. Even though the assumption of multiple sources represents a general scenario, this assumption is not necessary to show the influence of the mechanical coupling on the accuracy of incident angle estimation. In addition, the CRLB analysis was performed in the frequency domain by using the geometrical model of the time delay between the signals from the two ears for both the coupled and uncoupled diaphragm configurations. Even though the geometrical model can be used to obtain the time delay for two uncoupled microphone, it is not valid for the fly ear due to the coupling between the two eardrums. Furthermore, in their results, the phase difference between the two ear signals is shown to be almost constant over the entire frequency range. This is not consistent with the findings from the previous studies conducted by our group, in which the interaural phase difference is found to be amplified as the stimulus frequency reaches the rocking mode natural frequency when the signals are out of phase (i.e., IPD = 180^{o}) due to the mechanical coupling between the ear diaphragms. In this dissertation work, in order to achieve the fundamental understanding of the sound localization performance by using the fly-ear inspired sensors, the CRLB of the sensor with two coupled diaphragms is determined, which is compared with that obtained from two uncoupled diaphragms shown in Figure 2.1(a). As illustrated in Figure 2.1(b), the fly-ear inspired sensor has two circular diaphragms that are coupled by a beam pivoted in the middle, which can help amplify the directional cues. Here the phase difference between the two microphone signals [50] is used as the directional cue in the analysis. Furthermore, one acoustic source with single tone is assumed for the analysis.

In this this chapter, the CRLB of the fly-ear sensor and a conventional two-microphone array will be derived in detail and compared to study the effect of the mechanical coupling between the diaphragms on the variance of the incident angle estimation. In addition, numerical simulations will be conducted to compare the variance of the estimation by using a well-known real time estimator, namely, the cross correlation, for both the fly-ear sensor and the conventional array.

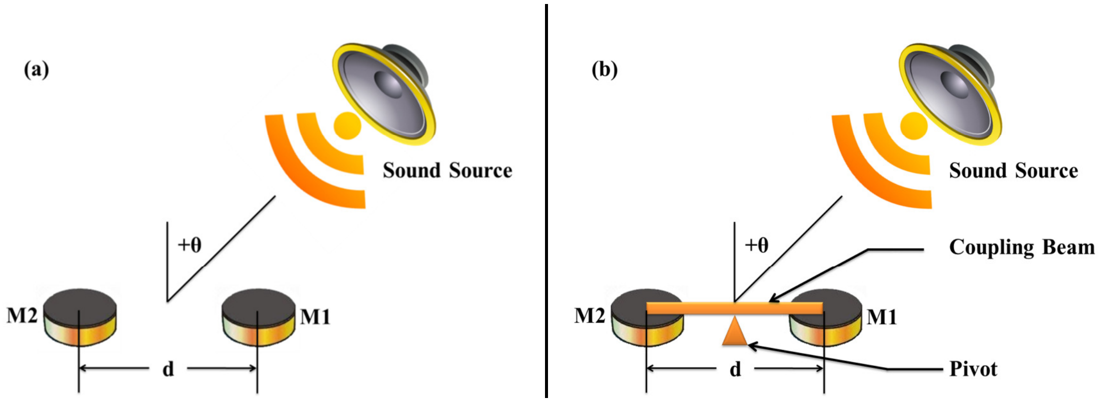


Figure 2.1: Schematic of (a) two uncoupled microphones and (b) fly-ear inspired sensor with two coupled diaphragms, d is the separation distance between the two microphones, and +θ represents a positive incident angle.

2.1 CRLB of Two Uncoupled Microphones

Here, two uncoupled microphones are considered, as shown in Figure 2.1 (a). For a pure tone source with a frequency *f*, the signal received from each microphone can be obtained as:

$$x_m(t) = A_m \cos(2\pi f t + \Phi_m) + w_m(t) \quad , \tag{2.1}$$

where A_m (m=1, 2) is the amplitude of the sound, w_m is the white Gaussian noise with zero mean and variance σ^2, Φ_m is the phase shift of the signal. It is assumed that the white noises from the two microphones are uncorrelated. The relationship between the two phases Φ_m (m=1, 2) is described by:

$$\Phi_1 = \Phi_2 + 2\pi \frac{d}{\lambda} \sin\theta \quad , \tag{2.2}$$

where d is the separation distance between the two microphones, λ is the sound wavelength, and θ is the incident angle.

Sampling the signal at N evenly spaced instants leads to:

$$x_m[n] = A_m \cos(2\pi f \Delta t \cdot n + \Phi_m) + w_m[n] , \tag{2.3}$$

where n is an integer from 0 to N-1, Δt is the sampling interval, and $w_m(n\Delta t)$ is simply denoted as $w_m[n]$. Accordingly, the signal-to-noise ratio (SNR) can be defined as:

$$\mathrm{SNR}_m = P_s / P_n = A_m{}^2 / (2\sigma^2) \tag{2.4}$$

where P_s and P_n are the signal and the noise power spectra respectively, and σ^2 is the noise variance.

In order to localize the acoustic source, it is necessary to estimate the following parameter vector:

$$\Theta = \, [A_1 \; A_2 \; \Phi_1 \; \Phi_2]^{\mathrm{T}} , \tag{2.5}$$

Also, to determine the CRLB, one needs to consider the following probability density function (PDF) [23], [48]:

$$p(x;\Theta) = \frac{1}{(2\pi\sigma^2)^{N/2}} \cdot \exp\left\{ -\frac{1}{2\sigma^2} \sum_{m=1}^{2} \sum_{n=0}^{N-1} \left[x_m[n] - A_m \cos(2\pi f \Delta t \cdot n + \Phi_m) \right]^2 \right\} . \tag{2.6}$$

Further, the corresponding Fisher Information Matrix (FIM) can be calculated by taking the following steps [48] (the details are given in Appendix A):

i) Take the logarithm of $p(x;\, \Theta)$ to remove the exponential part.

ii) Take the partial derivative with respect to the parameters: $\Theta = [A_1 \, A_2 \, \Phi_1 \, \Phi_2]^{\mathrm{T}}$

iii) If $x_m[n]$ still appears after step ii, the expectation (E{.}) of the obtained Eq. is calculated.

Accordingly, taking the natural log of $p(x;\,\Theta)$ leads to:

$$\ln\, p\left(x;\Theta\right) = -\frac{N}{2}\ln\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2}\sum_{m=1}^{2}\sum_{n=0}^{N-1}\left[x_m\left[n\right] - A_m\cos\left(2\pi f\,\Delta t\cdot n + \Phi_m\right)\right]^2 \quad (2.7)$$

.

Since there are four parameters, the FIM will be a 4x4 matrix, in which the elements of the matrix can be calculated as follows:

$$I(\Theta) = \begin{bmatrix} I_{11} & I_{12} & . & . & . & I_{1j} \\ I_{21} & . & & & & . \\ . & & . & & & . \\ . & & & . & & . \\ . & & & & . & . \\ I_{i1} & . & . & . & . & I_{ij} \end{bmatrix}, \quad (2.8)$$

where

$$\begin{aligned} &[I(\Theta)]_{ij} = -E\left[\frac{\partial^2}{\partial\Theta_i\partial\Theta_j} ln\ p(x;\Theta)\right] \\ &i, j = 1,2,3,4 \end{aligned} \quad (2.9)$$

.

The following trigonometric identities are needed for determining the FIM:

$$\sin\left(2\beta\right) \;=\; 2\sin\left(\beta\right)\cos\;\left(\beta\right) \quad (2.10)$$

,

and

$$\begin{aligned} \cos\left(2\beta\right) \;&=\; \cos^2\left(\beta\right) \;-\; \sin^2\left(\beta\right) \\ &=\; 2\cos^2\left(\beta\right) - 1 \\ &=\; 1 - \;2\sin^2\left(\beta\right) \end{aligned} \quad (2.11)$$

.

Starting with the first row in Eq. (2.8), the FIM can be defined as:

$$\frac{\partial}{\partial A_1}\left(\ln p\left(x;\Theta\right)\right) = \frac{1}{\sigma^2}\sum_{n=0}^{N-1}\left(x_1\left[n\right]\cos\left(2\pi f\,\Delta t\cdot n + \Phi_1\right) - A_1\cos^2\left(2\pi f\,\Delta t\cdot n + \Phi_1\right)\right) \quad (2.12)$$

$$\frac{\partial^2}{\partial {A_1}^2}\left(\ln p\left(x;\Theta\right)\right) = \frac{1}{\sigma^2}\sum_{n=0}^{N-1}\frac{1}{2}\left(1 + \cos\left(4\pi f\,\Delta t\cdot n + 2\Phi_1\right)\right) \quad (2.13)$$

Using the following approximation:

$$\sum_{n=0}^{N-1}\cos\left(4\pi f\Delta t\cdot n+2\Phi_1\right)\ll N\,, \tag{2.14}$$

then

$$[I[\Theta]]_{11}\simeq\frac{N}{2\sigma^2}\,. \tag{2.15}$$

Now using Eq. (2.12), taking the derivative with respect to A_2 and Φ_2, respectively, it can be obtained that:

$$\frac{\partial}{\partial A_2}\frac{\partial}{\partial A_1}\left(\ln p\left(x;\Theta\right)\right)=\frac{\partial}{\partial \Phi_2}\frac{\partial}{\partial A_1}\left(\ln p\left(x;\Theta\right)\right)=0\,, \tag{2.16}$$

which gives that

$$\left[I\left[\Theta\right]\right]_{12}=\left[I\left[\Theta\right]\right]_{14}=0\,. \tag{2.17}$$

Similarly, using Eq. (2.12), the third element of the first row is found to be

$$\begin{aligned}&\frac{\partial}{\partial \Phi_1}\frac{\partial}{\partial A_1}\left(\ln p\left(x;\Theta\right)\right)=\\&-\frac{1}{\sigma^2}\sum_{n=0}^{N-1}\left(x_1\left[n\right]\sin\left(2\pi f\Delta t\cdot n+\Phi_1\right)-A_1\sin\left(4\pi f\Delta t\cdot n+2\Phi_1\right)\right)\end{aligned}\,. \tag{2.18}$$

Since $x_m[n]$ still appears in the second derivative, taking the expectation of the Eq. (2.18)(2.18) gives

$$[I[\Theta]]_{13}=\frac{1}{\sigma^2}\sum_{n=0}^{N-1}\left(E\left\{x_1\left[n\right]\right\}\sin\left(2\pi f\Delta t\cdot n+\Phi_1\right)-A_1\sin\left(4\pi f\Delta t\cdot n+2\Phi_1\right)\right)\,, \tag{2.19}$$

$$\left[I\left[\Theta\right]\right]_{13}=-\frac{A_1}{\sigma^2}\sum_{n=0}^{N-1}\left(\sin\left(4\pi f\Delta t\cdot n+2\Phi_1\right)\right)\approx 0\,. \tag{2.20}$$

Note that

$$\left[I\left[\Theta\right]\right]_{12}=\left[I\left[\Theta\right]\right]_{21}, \tag{2.21}$$

and

$$\left[I\left[\Theta\right]\right]_{13}=\left[I\left[\Theta\right]\right]_{14}=0 \tag{2.22}$$

For the second row elements, taking the derivative of the PDF with respect to A_2 gives:

$$\begin{aligned}&\frac{\partial}{\partial A_2}\left(\ln p\left(x;\Theta\right)\right)= \\ &\frac{1}{\sigma^2}\sum_{n=0}^{N-1}\left(x_2\left[n\right]\cos\left(2\pi f\Delta t\cdot n+\Phi_2\right)-A_2\cos^2\left(2\pi f\Delta t\cdot n+\Phi_2\right)\right)\end{aligned} \tag{2.23}$$

.

Then taking the second derivative of Eq. (2.23) leads to

$$\frac{\partial^2}{\partial {A_2}^2}\left(\ln p\left(x;\Theta\right)\right)=-\frac{1}{\sigma^2}\sum_{n=0}^{N-1}-\cos^2\left(2\pi f\Delta t\cdot n+\Phi_2\right) \tag{2.24}$$

Following Step (iii) and using the identities defined earlier leads to:

$$[I[\Theta]]_{22}=\frac{1}{\sigma^2}\sum_{n=0}^{N-1}\frac{1}{2}\left(1+\cos\left(4\pi f\Delta t.n+2\Phi_2\right)\right) \tag{2.25}$$

.

By using Eq. (2.14), it gives:

$$[I[\Theta]]_{22}\simeq\frac{N}{2\sigma^2} \tag{2.26}$$

To obtain the first and third elements of the row, taking the derivative of Eq. (2.23) with respect to A_1 and Φ_1, respectively, gives:

$$\frac{\partial}{\partial A_1}\frac{\partial}{\partial A_2}\left(\ln p\left(x;\Theta\right)\right)=\frac{\partial}{\partial \Phi_1}\frac{\partial}{\partial A_2}\left(\ln p\left(x;\Theta\right)\right)=0\,. \tag{2.27}$$

Also note that:

$$\left[I\left[\Theta\right]\right]_{21}=\left[I\left[\Theta\right]\right]_{23}=0 \quad (2.28)$$

For the elements [I[Θ]]$_{24}$, [I[Θ]]$_{42}$, using Eq. (2.23) and taking the derivative with respect to Φ_2 leads to

$$\begin{aligned}&\frac{\partial}{\partial\Phi_2}\frac{\partial}{\partial A_2}\left(\ln p\left(x;\Theta\right)\right)=\\&-\frac{1}{\sigma^2}\sum_{n=0}^{N-1}\left(x_2\left[n\right]\sin\left(2\pi f\Delta t\cdot n+\Phi_2\right)-A_2\sin\left(4\pi f\Delta t\cdot n+2\Phi_2\right)\right)\end{aligned} \quad (2.29)$$

Similarly, it can be obtained that

$$\begin{aligned}&[I[\Theta]]_{24}=[I[\Theta]]_{42}=\\&\frac{1}{\sigma^2}\sum_{n=0}^{N-1}\left(E\left\{x_2\left[n\right]\right\}\sin\left(2\pi f\Delta t\cdot n+\Phi_2\right)-A_2\sin\left(4\pi f\Delta t\cdot n+2\Phi_2\right)\right)\end{aligned} \quad (2.30)$$

Using Eqs. (2.10) and (2.11), the following FIM elements are found

$$[I[\Theta]]_{24}=[I[\Theta]]_{42}=-\frac{A_2}{\sigma^2}\sum_{n=0}^{N-1}\left(\sin\left(4\pi f\Delta t\cdot n+2\Phi_2\right)\right)\approx 0 \quad (2.31)$$

For the third row of the FIM, taking the derivative of PDF with respect to Φ_1 leads to:

$$\begin{aligned}&\frac{\partial}{\partial\Phi_1}\left(\ln p\left(x;\Theta\right)\right)=\\&\frac{1}{\sigma^2}\sum_{n=0}^{N-1}\left(-A_1x_1\left[n\right]\sin(2\pi f\Delta t\cdot n+\Phi_1)-\frac{A_1}{2}\sin(4\pi f\Delta t\cdot n+2\Phi_1)\right)\end{aligned} \quad (2.32)$$

Then, taking the second derivative gives:

$$\begin{aligned}&\frac{\partial^2}{\partial{\Phi_1}^2}\left(\ln p\left(x;\Theta\right)\right)=\\&-\frac{A_1}{\sigma^2}\sum_{n=0}^{N-1}\left(x_1\left[n\right]\cos\left(2\pi f\Delta t\cdot n+\Phi_1\right)-A_1\cos\left(4\pi f\Delta t\cdot n+2\Phi_1\right)\right)\end{aligned} \quad (2.33)$$

Further, following Step (iii) , the third element of the third row can be obtained as:

$$\left[I\left[\Theta\right]\right]_{33}=\frac{A_1^{\;2}}{\sigma^2}\sum_{n=0}^{N-1}\cos^2\left(2\pi f\Delta t\cdot n+\Phi_1\right)-\cos\left(4\pi f\Delta t\cdot n+2\Phi_1\right) \quad (2.34)$$

Similarly, using Eqs. (2.10) and (2.11) leads to

$$\left[I\left[\Theta\right]\right]_{33}=\frac{A_1^{\;2}}{2\sigma^2}\sum_{n=0}^{N-1}\left(1-\cos\left(4\pi f\Delta t\cdot n+2\Phi_1\right)\right)\approx\frac{NA_1^{\;2}}{2\sigma^2} \quad (2.35)$$

Since

$$\frac{\partial}{\partial\Phi_1}\frac{\partial}{\partial A_1}\left(\ln p\left(x;\Theta\right)\right)=\frac{\partial}{\partial A_1}\frac{\partial}{\partial\Phi_1}\left(\ln p\left(x;\Theta\right)\right)=0 \quad (2.36)$$

and $\frac{\partial}{\partial A_2}\frac{\partial}{\partial\Phi_1}\left(\ln p\left(x;\Theta\right)\right)=\frac{\partial}{\partial\Phi_2}\frac{\partial}{\partial\Phi_1}\left(\ln p\left(x;\Theta\right)\right)=0$ (2.37) ,

it can be obtained that $\left[I\left[\Theta\right]\right]_{31}=0$ (2.38)

and $\left[I\left[\Theta\right]\right]_{32}=\left[I\left[\Theta\right]\right]_{34}=0$. (2.39)

Finally, for the fourth row of the FIM, taking the derivative of the PDF with respect to Φ2 gives:

$$\begin{aligned}&\frac{\partial}{\partial\Phi_2}\left(\ln p\left(x;\Theta\right)\right)=\\&\frac{1}{\sigma^2}\sum_{n=0}^{N-1}\left(-A_2x_2\left[n\right]\sin(2\pi f\Delta t\cdot n+\Phi_2)-\frac{A_2}{2}\sin(4\pi f\Delta t\cdot n+2\Phi_2)\right)\end{aligned} \quad (2.40)$$

Next, taking the second derivative leads to:

$$\frac{\partial^2}{\partial\Phi_2^{\;2}}\left(\ln p\left(x;\Theta\right)\right)=-\frac{A_2}{\sigma^2}\sum_{n=0}^{N-1}\left(x_2\left[n\right]\cos\left(2\pi f\Delta t\cdot n+\Phi_2\right)-A_2\cos\left(4\pi f\Delta t\cdot n+2\Phi_2\right)\right) \quad (2.41)$$

Similarly, following Step (iii), it can be shown that

$$\left[I\left[\Theta\right]\right]_{44}=\frac{A_2^{\;2}}{\sigma^2}\sum_{n=0}^{N-1}\cos^2\left(2\pi f\Delta t\cdot n+\Phi_2\right)-\cos\left(4\pi f\Delta t\cdot n+2\Phi_2\right) \quad (2.42)$$

,

which can be further simplified as $\left[I[\Theta]\right]_{44} = \frac{A_4^{\ 2}}{\sigma^2}\sum_{n=0}^{N-1}\sin^2\left(2\pi f\Delta t\cdot n+\Phi_4\right)$ (2.43)
.

Again, using the identities defined earlier leads to:

$$\left[I[\Theta]\right]_{44} = \frac{A_2^{\ 2}}{2\sigma^2}\sum_{n=0}^{N-1}\left(1-\cos\left(4\pi f\Delta t\cdot n+2\Phi_2\right)\right)\approx\frac{NA_2^{\ 2}}{2\sigma^2}. \quad (2.44)$$

It can be also shown that:

$$\frac{\partial}{\partial A_1}\frac{\partial}{\partial\Phi_2}\left(\ln p(x;\Theta)\right)=\frac{\partial}{\partial\Phi_1}\frac{\partial}{\partial\Phi_2}\left(\ln p(x;\Theta)\right)=0, \quad (2.45)$$

and
$$\frac{\partial}{\partial A_2}\frac{\partial}{\partial\Phi_2}\left(\ln p(x;\Theta)\right)=\frac{\partial}{\partial\Phi_2}\frac{\partial}{\partial A_2}\left(\ln p(x;\Theta)\right)=0. \quad (2.46)$$

Therefore, it can be obtain that $\left[I[\Theta]\right]_{41}=\left[I[\Theta]\right]_{43}=0$ (2.47)

and $\left[I[\Theta]\right]_{24}=\left[I[\Theta]\right]_{42}=0$. (2.48)

Accordingly, the FIM matrix is found to be:

$$FIM=\begin{bmatrix}\frac{N}{2\sigma^2} & 0 & 0 & 0\\ 0 & \frac{N}{2\sigma^2} & 0 & 0\\ 0 & 0 & \frac{N\cdot A_1^{\ 2}}{2\sigma^2} & 0\\ 0 & 0 & 0 & \frac{N\cdot A_2^{\ 2}}{2\sigma^2}\end{bmatrix}. \quad (2.49)$$

Since FIM is a diagonal matrix, the inverse of FIM is found to be:

$$\left(FIM\right)^{-1} = \begin{bmatrix} \frac{2\sigma^2}{N} & 0 & 0 & 0 \\ 0 & \frac{2\sigma^2}{N} & 0 & 0 \\ 0 & 0 & \frac{2\sigma^2}{N \cdot A_1^{\ 2}} & 0 \\ 0 & 0 & 0 & \frac{2\sigma^2}{N \cdot A_2^{\ 2}} \end{bmatrix} \tag{2.50}$$

The CRLB can be written as:

$$CRLB = (FIM)^{-1} = \frac{1}{\det(FIM)} \times Adjo\operatorname{int}(FIM) \tag{2.51}$$

To determine the variance of the following parameters:

$$\alpha = \left[A_1\ A_2\ \theta\right]^T = g\left(\Theta\right), \tag{2.52}$$

the variances are related to the CRLB as

$$\operatorname{var}\left(\alpha\right) \geq CRLB_\alpha = \left(\frac{\partial g\left(\Theta\right)}{\partial\Theta}\right) CRLB_\Theta \left(\frac{\partial g\left(\Theta\right)}{\partial\Theta}\right)^T, \tag{2.53}$$

where

$$\frac{\partial g(\Theta)}{\partial\Theta} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{1}{2\pi\left(\frac{d}{\lambda}\right)\cos(\theta)} & \frac{1}{2\pi\left(\frac{d}{\lambda}\right)\cos(\theta)} \end{bmatrix}, \tag{2.54}$$

and

$$\left(\frac{\partial g(\Theta)}{\partial\Theta}\right)^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -\dfrac{1}{2\pi\left(\dfrac{d}{\lambda}\right)\cos(\theta)} \\ 0 & 0 & \dfrac{1}{2\pi\left(\dfrac{d}{\lambda}\right)\cos(\theta)} \end{bmatrix}. \qquad (2.55)$$

Therefore, the variance of the incident angle estimation using Eq. (2.53) can be rewritten as

$$\operatorname{var}\left(\hat{\theta}\right) \geq \left(\frac{1}{SNR_1}+\frac{1}{SNR_2}\right)\frac{1}{N}\frac{1}{\left(2\pi\left(\dfrac{d}{\lambda}\right)\cos\left(\theta\right)\right)^2}, \qquad (2.56)$$

where SNR_1 and SNR_2 are the signal to noise ratios of the two microphone signals, which are defined as [48]:

$$\begin{aligned} SNR_1 &= \frac{A_1^{\ 2}}{2\sigma^2} \\ SNR_2 &= \frac{A_2^{\ 2}}{2\sigma^2}. \end{aligned} \qquad (2.57)$$

Let the equivalent SNR be defined as:

$$SNR_{eq} = \frac{SNR_1 \cdot SNR_2}{SNR_1 + SNR_2}, \qquad (2.58)$$

and the directional sensitivity *DS* be defined as the derivative of interaural phase difference (IPD) with respect to the incident angle, i.e.

$$\frac{\partial IPD}{\partial\theta} = 2\pi\left(\frac{d}{\lambda}\right)\cos\left(\theta\right) = DS, \qquad (2.59)$$

the variance can thus be determined as

$$\mathrm{var}\left(\hat{\theta}\right) \geq \frac{1}{SNR_{eq}} \frac{1}{N} \frac{1}{DS^2} \quad (2.60)$$
.

2.2 CRLB of the Fly-Ear Inspired Sensor

In the fly-ear inspired sensor, the two diaphragms are mechanically coupled by a beam as indicated in Figure 2.1(b). As a result, the IPD in the sound stimulus is amplified to the mechanical Interaural Phase Difference (mIPD). Hence, Eq. (2.2) needs to be replaced by:

$$\Phi_1 = \Phi_2 + mIPD \quad (2.61)$$
,

where mIPD can be obtained from the equivalent 2-DOF model of the acoustic sensor by (details are shown in Appendix B):

$$mIPD = \angle \frac{\Gamma + j\tan\left(IPD/2\right)}{\Gamma - j\tan\left(IPD/2\right)} \quad (2.62)$$
,

where

$$\Gamma = \frac{1 - \Omega^2 + j2\xi_1\Omega}{\eta^2 - \Omega^2 + j2\eta\xi_2\Omega} \quad (2.63)$$
,

Also, $\eta = f_2 / f_1$ denotes the ratio of the system first two natural frequencies, ξ_1, and ξ_2 are the damping ratios, Ω is the sound frequency f normalized by the first natural frequency of the fly-ear inspired sensor structure as $\Omega = f / f_1$. Following the same procedure used in obtaining the CRLB for the uncoupled microphones in the previous section and using Eqs. (2.61) - (2.63), the variance can be obtained for the fly-ear inspired sensor as

$$\mathrm{var}\left(\hat{\theta}\right) \geq \frac{1}{SNR_{eq}} \frac{1}{N} \frac{1}{mDS^2} \quad (2.64)$$
,

where *mDS* is the directional sensitivity of the fly-ear inspired sensor and

$$mDS = \frac{\partial mIPD}{\partial \theta} \quad . \tag{2.65}$$

Accordingly, to determine the *mDS*, let

$$\Gamma = R + Ij \quad . \tag{2.66}$$

Then, Eq. (2.62) can be rewritten as

$$mIPD = \angle \frac{R + j\left(I + \tan\left(\frac{\varphi}{2}\right)\right)}{R + j\left(I - \tan\left(\frac{\varphi}{2}\right)\right)} \quad , \tag{2.67}$$

where

$$\varphi = 2\pi \frac{d}{\lambda} \sin(\theta) \equiv IPD \quad , \tag{2.68}$$

and

$$\frac{\partial \varphi}{\partial \theta} = 2\pi \frac{d}{\lambda} \cos(\theta) \quad . \tag{2.69}$$

Multiplying Eq. (2.67) by the conjugate of the denominator, gives:

$$\tan(mIPD) = \frac{2R.\tan\left(\frac{\varphi}{2}\right)}{R^2 + I^2 - \tan^2\left(\frac{\varphi}{2}\right)} \quad . \tag{2.70}$$

Next, the derivative of *mIPD* with respect to the incident angle is determined to estimate the directional sensitivity as follows:

$$\frac{\partial mIPD}{\partial \theta} = mDS = \frac{2\pi(d/\lambda)R\cos\theta\left(|\Gamma|^2 + \Psi^2\right)\left(1 + \Psi^2\right)}{\left(|\Gamma|^2 - \Psi^2\right)^2 + 4R^2\Psi^2} \quad , \tag{2.71}$$

where *R* denotes the real part of *Γ* as

$$R = \frac{1 - \Omega^2 + 2\xi_1\Omega}{\eta^2 - \Omega^2 + 2\eta\gamma\xi_1\Omega} \quad , \tag{2.72}$$

and Ψ are defined as:

$$\Psi = \tan\left(\pi\frac{d}{\lambda}\sin\theta\right). \tag{2.73}$$

Based on Eqs. (2.60) and (2.64), it can be concluded that the variance of the azimuth estimation for both the conventional microphone pair and the fly-ear inspired sensor is inversely proportional to the equivalent *SNR*, number of samples, and the squares of the directional sensitivity. When the *SNR* and *N* are assumed to be the same for both the conventional microphone pair and the fly-ear inspired sensor, for the ease of comparison, the variance can be normalized as follows:

$$\mathrm{var}_n\left(\hat{\theta}\right) = \mathrm{var}\left(\hat{\theta}\right)\cdot SNR_{eq}\cdot N. \tag{2.74}$$

In Figure 2.2, comparison of the normalized CRLB for both the conventional microphone pair and the fly-ear inspired directional microphone is shown, in which for the simulation purposes, the fly-ear inspired sensor is characterized with the same calibrated parameters of the fly ear listed in Table 2-1 [37]. Two remarks can be made, based on this figure. First, for both configurations, the CRLB increases as the sound incident angle increases. This indicates that the incident angle estimation accuracy decreases when the sound source deviates from the midline (θ=0$^{\text{o}}$). Second, due to the amplification of the directional sensitivity by the mechanical coupling mechanism, the CRLB for the fly-ear inspired sensor can be reduced significantly. For example, at the midline, the directional sensitivity is amplified from 0.11 deg/deg in the sound stimulus input to 1.13 deg/deg in the mechanical response output. As a result, the CRLB of the fly-ear reduces to be less than 1/100 of that obtained for the conventional microphone pair with the same separation distance between the two microphones. This indicates that the mechanical coupling improved the incident angle estimation accuracy by 100 times compared with that of the conventional microphone pair.

Table 2-1: Parameters used in the 2-DOF model for the fly-ear [37]

Parameters	Values
Mass of membrane m	2.88×10^{-10} kg
Translational spring k_1,k_2	0.576 N/m
Translation dashpot c_1,c_2	1.15×10^{-5} N s/m
Torsional spring k_3	5.18 N/m
Torsional dashpot c_3	2.88×10^{-5} N s/m
Separation of force locations d	1.2×10^{-3} m
Tympanum area s	0.288×10^{-6} m^2
Excitation frequency ω	3.14×10^{4} rad/s (5 kHz)
Sound speed c	344 m/s

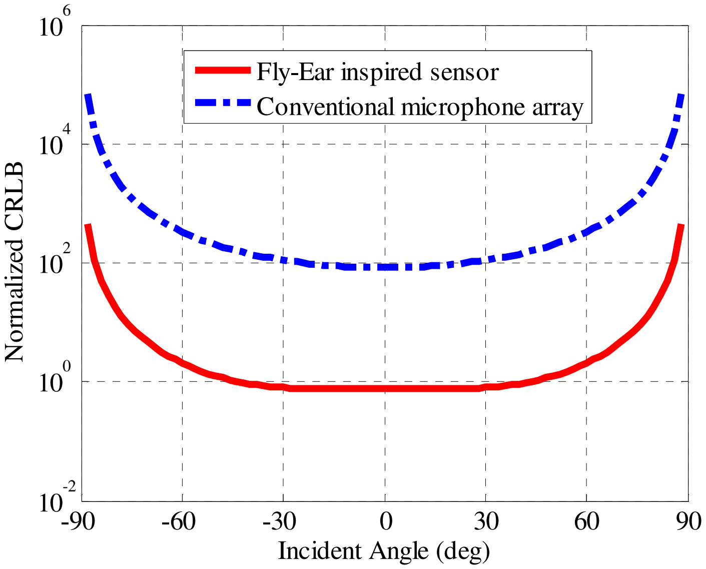


Figure 2.2: Cramer-Rao lower bound (CRLB) of azimuth estimation obtained from both the microphone pair and the fly-ear inspired sensor. The separation distance is 1.2mm for both cases, the frequency is 5kHz, and the parameters in [37] are used for the fly-ear inspired sensor.

A question needs to be addressed is whether an unbiased estimator that can be used to achieve the CRLB exists for the fly-ear inspired sensor and the conventional microphone array. Clearly, if the CRLB condition can be satisfied then there exists an unbiased estimator that attains the CRLB, which is the minimum variance unbiased (MVU) estimator. Otherwise, an unbiased estimator that attains the CRLB does not exist. But this does not necessarily mean that there may not be a minimum variance unbiased estimator. To further explore this, the CRLB theorem [48] for the scalar and vector parameters states that:

(i) Cramer-Rao lower bound theorem for scalar parameters

It is assumed that the PDF p(x; θ) satisfies the “regularity” condition

$$E\left[\frac{\partial}{\partial\theta}\ln p(x;\theta)\right]=0 \qquad \textit{for all } \theta \tag{2.75}$$
,

where the expectation is taken with respect to *p*(*x*; *θ*). Then, the variance of any unbiased estimator $\hat{\theta}$ must satisfy

$$\operatorname{var}(\hat{\theta})\geq\frac{1}{-E\left[\frac{\partial^2}{\partial\theta^2}\ln p(x;\theta)\right]} \tag{2.76}$$
,

where the derivative is evaluated at the true value of θ and the expectation is taken with respect to *p*(*x*; *θ*). Furthermore, an unbiased estimator can be found that attains the CRLB for all *θ* if and only if

$$\frac{\partial}{\partial\theta}\ln p(x;\theta)=I(\theta)\times(g(x)-\theta) \tag{2.77}$$
,

for some functions *g*(.) and *I* (.). The estimator, which is the MVU estimator, is $\hat{\theta}=g(x)$, and the minimum variance is 1/ I(θ).

(ii) Cramer-Rao lower bound theorem for vector parameters

It is assumed that the PDF *p*(*x*; *θ*) satisfies the “regularity” condition

$$E\left[\frac{\partial}{\partial\theta}\ln p(x;\theta)\right]=0 \qquad for\,all\;\theta \tag{2.78}$$

,

where the expectation is taken with respect to *p*(*x*; *θ*). Based on Eq.(2.77), the covariance matrix of any unbiased estimator θ̂ satisfies

$$C_{\hat{\theta}}-I^{-1}(\theta)\geq 0 \tag{2.79}$$

,

where larger or equal to zero is interpreted as meaning that the matrix is positive semi definite. The Fisher information matrix *I*(*θ*) is then given as

$$[I(\theta)]_{ij}=-E\left[\frac{\partial^2}{\partial\theta_i\partial\theta_j}\ln p(x;\theta)\right] \tag{2.80}$$

,

where the derivatives are evaluated at the true value of *θ* and the expectation is taken with respect to *p*(*x*; *θ*). Furthermore, an unbiased estimator can be found that attains the CRLB in that $C_{\hat{\theta}}=I^{-1}(\theta)$ if and only if Eq. (2.77) is satisfied. For a q-dimensional function *g* and a q x q matrix *I*, the estimator, which is the MVU estimator, is $\hat{\theta}=g(x)$, and its covariance matrix is $I^{-1}(\theta)$.

It is found that Eqs. (2.12), (2.23), (2.32), and (2.40) cannot be rewritten in the form shown in Eq. (2.77) in the CRLB theorem for the vector parameters. Therefore, there is no unbiased estimator that can be used to attain the CRLB. To find a possible MVU estimator, two approaches can be used, namely, applying the Rao-Blackwell - Lechman - Scheffe theorem [48], which is not common in practice, or applying a linear unbiased estimator. The problem of using the second approach is that the sensor behavior is not linear and a linear approximation can only be made within a certain range of incident angles. Outside this linear region, the linear unbiased estimator is not valid.

Using the first approach, a single sufficient statistic for θ should be found by using the Neyman-Fisher factorization theorem [48]. Then, the sufficient statistic is tested for completeness. If it is not complete, this approach cannot be used. If it is complete a function g of the sufficient statistic that yields an unbiased estimator $\hat{\theta} = g(T(x))$ should be found which is the MVU estimator. An alternative implementation of the last step can be evaluating $\hat{\theta} = E(\breve{\theta}\ |T(x))$, where $\breve{\theta}$ is any unbiased estimator.

According to Neyman-Fisher factorization theorem, if the PDF $p(x;\theta)$ can be written as

$$p(x;\theta) \ = \ g(T(x),\theta)h(x)\ , \tag{2.81}$$

where g is a function depending on x through $T(x)$ with r x 1 statistic, and also on θ, h is a function depending only on x, and $T(x)$ is a sufficient statistic for θ.

The PDF in Eq. (2.81) can be re-written as the following:

$$\begin{aligned} &p(x;\Theta) = \\ &\frac{1}{\left(2\pi\sigma^2\right)^{N/2}}\exp\left\{-\frac{1}{2\sigma^2}\sum_{m=1}^{2}\left[\frac{A_m{}^2\cdot N}{2}-2A_mT_1(x)\cos(\varphi_m)+2A_mT_2(x)\sin(\varphi_m)\right]\right\} \\ &\cdot\exp\left\{-\frac{1}{2\sigma^2}\sum_{m=1}^{2}\sum_{n=0}^{N-1}x_m^2[n]\right\} \end{aligned}, \tag{2.82}$$

where the jointly sufficient statistic is represented as

$$T(x) = \begin{bmatrix} \sum_{n=0}^{N-1} x_m[n]\cdot\cos(2\pi f\Delta t\cdot n) \\ \sum_{n=0}^{N-1} x_m[n]\cdot\sin(2\pi f\Delta t\cdot n) \end{bmatrix}. \tag{2.83}$$

Since the PDF (Eq. (2.82)) belongs to the vector exponential family of PDFs, the sufficient statistic is complete.

Based on Rao-Blackwell – Lechman – Scheffe theorem, if $\breve{\theta}$ is an unbiased estimator of θ and $T(x)$ is an r x 1 sufficient statistic for θ, then $\hat{\theta} = E(\breve{\theta} \,|T(x))$ is

i) a valid estimator for θ (not dependent on θ),

ii) unbiased,

iii) of lesser or equal variance that of $\breve{\theta}$(each element of $\hat{\theta}$ has lesser or equal variance).

Additionally, if the sufficient statistic is complete, then $\hat{\theta}$ is the minimum variance estimator. However, the Rao-Blackwell - Lechman - Scheffe theorem cannot be applied without the estimators $g(T(x))$ which cannot be easily obtained. Alternatively, numerical simulations are conducted to find the angle estimation variance.

2.3 Numerical Simulations

Numerical simulations are conducted to find the angle estimation variance by using the variance of the phase difference between two signals, namely, the IPD in Eq. (2.68). The signals from the two microphones in the fly-ear sensor [37] or the conventional microphone pair are defines as follows

The fly-ear inspired sensor [37], the signals are

$$\begin{aligned} x_1(t) &= A_t \sin(\omega t + \phi_t) + A_r \cos(\omega t + \phi_r), \\ x_2(t) &= A_t \sin(\omega t + \phi_t) - A_r \cos(\omega t + \phi_r), \end{aligned} \tag{2.84}$$

where

$$
\begin{aligned}
A_r &= \frac{P \cdot s}{m}\left(\frac{\sin(\omega\tau/2)}{\sqrt{(\omega_r^2-\omega^2)^2+(2\omega_r\xi_r\omega)^2}}\right),\\
A_t &= \frac{P \cdot s}{m}\left(\frac{\cos(\omega\tau/2)}{\sqrt{(\omega_t^2-\omega^2)^2+(2\omega_t\xi_t\omega)^2}}\right),\\
\phi_r &= -\tan^{-1}\left(\frac{2\omega_r\xi_r\omega}{\omega_r^2-\omega^2}\right),\\
\phi_t &= -\tan^{-1}\left(\frac{2\omega_t\xi_t\omega}{\omega_t^2-\omega^2}\right),
\end{aligned}
\qquad (2.85)
$$

Here, *P* is the sound pressure level in dB, *s* is the sensor diaphragm cross sectional area, *m* is the mass of the diaphragm, ω_t and ω_r are the bending and the rocking mode natural frequencies, respectively, in rad/s, ξ_t and ξ_r are the bending (translational) and the rocking mode damping ratios, respectively, and ω is the stimulus frequency.

For a conventional microphone pair, the received signals can be written as:

$$x_m(t) = A_m \sin(\omega t + \phi_m), \qquad (m=1,2) \qquad (2.86)$$

where

$$\phi_m = (-1)^{m+1}\pi\frac{d}{\lambda}\sin(\theta) \qquad (2.87)$$

.

Here, *d* and λ are the separation distance between the microphones and the sound wavelength, respectively. The phase difference between the signals is obtained using one of the well-known methods, namely, the cross correlation explained earlier in Section 1.2.1.2.

The variance of the incident angle estimation cannot be evaluated directly, since the closed form calculation of the incident angle cannot be derived from Eq. (2.62). In this work a numerical approach is developed to evaluate the variance of the incident angle estimation.

To get the variance of estimating the sound incident angle, the variance of estimating the phase difference can be utilized since IPD is a function of the sound incident angle and frequency; that is,

$$IPD = F(\theta, f)\,. \tag{2.88}$$

Taking the Taylor series expansion of Eq. (2.88) at a specific known frequency and ignoring the higher order terms leads to

$$IPD \cong \frac{\partial F(\theta)}{\partial \theta} \cdot \theta\,. \tag{2.89}$$

The variance is found to be

$$\mathrm{var}(\theta) \cong \frac{\mathrm{var}(IPD)}{\left(\frac{\partial IPD}{\partial \theta}\right)^2}\,. \tag{2.90}$$

In the simulations, a sampling rate of 500 kHz is used to obtain 1000 data points for each signal with additive white Gaussian noise. Angle estimation was conducted 100 times at each hypothetical sound position. Since the phase difference estimation is sensitive to the separation distance and signal to noise ratio (SNR), these parameters are kept the same in the simulations for the conventional microphone array and the fly-ear sensor to have a fair comparison under the same conditions.

Based on Eq. (2.97), the variance of the simulated measurements is evaluated and compared with the CRLB obtained for the fly-ear inspired sensor and the conventional microphone array, as shown in Figure 2.3, in which the SNR is assumed to be 30 dB. As expected, a smaller variance is obtained for the fly-ear inspired sensor than that for the conventional microphone array due to the higher directional sensitivity resulted from the mechanical coupling between the diaphragms. Figure 2.4

shows the effect of the SNR on the variance of the incident angle estimation at a specific incident angle of 45°. Both the fly-ear and the conventional microphone array exhibit a smaller variance with increasing the SNR.

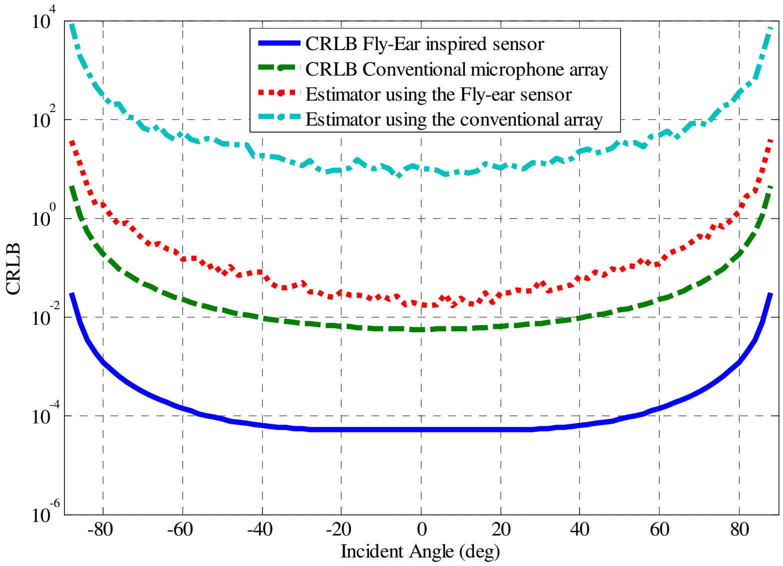


Figure 2.3: Cramer-Rao lower bound (CRLB) of azimuth estimation and the variance of azimuth estimation using Eq. (2.90). The diaphragms separation is 1.2mm, the frequency is 5 kHz, and the parameters in [33] are used for the fly-ear inspired sensor. The signal to noise ratio is 30 dB

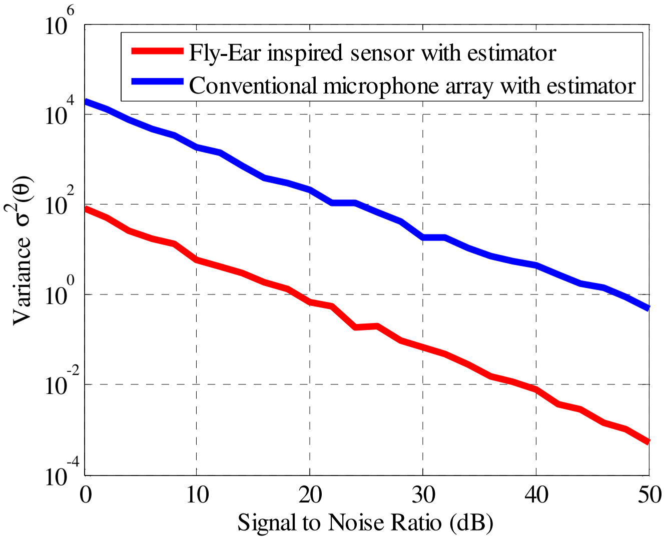


Figure 2.4: Variance of azimuth angle estimation at different SNRs. The diaphragms separation is 1.2mm, the frequency is 5 kHz, the parameters in [33] are used for the fly-ear inspired sensor, and the incident angle is 45°.

The above results indicate that using the fly-ear inspired sensor with mechanical coupling between the two microphone diaphragms, the accuracy of estimating the sound incident angle can be improved. This means that by using a microphone array with the fly-ear inspired sensor, a reduced size of the array can be obtained, without significantly compromising the localization performance. These results demonstrate that it is promising that the fly-ear inspired sensors can be applied to miniature robots for sound source localization.

2.4 Summary

In this chapter, the derivation of the *Cramer Rao Lower Bound* (CRLB) for the fly-ear inspired sensor and the conventional two-microphone array is presented. Comparing with that of a microphone pair with the same separation distance and same signal to noise ratio, it is found that the CRLB for the fly-ear inspired sensor is 100 times less due to the mechanical coupling between the diaphragms.

A numerical approach is used to obtain an estimation of the incident angle with the fly-ear inspired sensor and the two-microphone array. The simulation results demonstrate that the fly-ear sensor has a much lower variance (~500 times) in the sound incident angle estimation than that for the conventional two-microphone array. The improvement on the incident angle estimation variance for the fly-ear inspired sensor is related to the high directional sensitivity enabled by the mechanical coupling between the diaphragms in the sensor.

These results shown that it is promising to equip the fly-ear inspired sensors in miniature robots for sound source localization, which will help to address size restrictions associated with the conventional microphone arrays. These results can also enhance the understanding of the performance of the fly-ear inspired sensors.

Chapter 3 Sound Localization Algorithms

Although the fly-ear inspired sensor has been proven to have better performance in estimating the sound incident angle compared with a conventional microphone pair, using the fly-ear inspired acoustic sensor for localizing the sound source has several challenges. First, based on Eq. (2.62), the explicit analytical expression for the sound azimuth angle is difficult to obtain from the measurement results of mechanical Interaural Phase Difference (mIPD). Second, since mIPD to sound incident angle is a surjective (not one-to-one) mapping (i.e., inverse problem) for different frequencies, the incident angle cannot be obtained easily. Third, when the mIPD is used as the directional cues, differentiation between the sound propagating from the front and the back is challenging which results in the “ambiguity problem”.

In this chapter, two sound source localization algorithms that can be used along with the fly-ear inspired acoustic sensors are developed. The first sound source localization algorithm utilizes an optimization method, namely, the model-free gradient descent method. This algorithm utilizes the extracted phase difference information (i.e., mIPD) from the fly-ear inspired sensor, which is investigated for two dimensional sound localization. Instead of using an iterative optimization technique to localize the sound, the second algorithm employs a fuzzy logic method. This algorithm can be used to obtain a mapping between the sound azimuth angle and the information extracted from the fly-ear inspired sensor (e.g., mIPD or mechanical interaural time difference (mITD)).

3.1 Optimization Based Method

In optimization, a gradient descent/ascent method, which is a first-order optimization algorithm, is commonly used to find a local minimum/maximum of a function. Using this method, increments proportional to the negative/positive of the gradient of the function are taken, at the current point to improve the estimates of the state variable *y*, which can be written as [51]:

$$y(i+1) - y(i) = - \mu \nabla J(y(i)), \tag{3.1}$$

where μ is the gain such that

$$\begin{array}{ll} \mu > 0 & \text{(searching for local minimum)} \\ \mu < 0 & \text{(searching for local maximum)} \end{array} \tag{3.2}$$

and *J* is the objective function used to find a local minimum/maximum.

In this work, rather than calculating the analytical value for the gradient, the sensor measurement will be used to numerically evaluate the gradient in the azimuth and elevation directions. In addition, the gradient can be approximated as a finite difference as the following:

$$\nabla J(\Theta) = \frac{J(\Theta(i+1)) - J(\Theta(i))}{\Theta(i+1) - \Theta(i)}, \tag{3.3}$$

where Θ is the absolute position of the fly-ear sensor.

In order to illustrate this algorithm, a conventional microphone array with four microphones is used which satisfies the uniform angular distribution [24], as shown in Figure 3.1. In this work, the studies are carried out for the conventional microphone array, but the methodology can be readily applied to the fly-ear inspired sensor array. Two rotational stages are used to control the azimuth and elevation positions separately. The separation distance between each two microphones in the azimuth/elevation is assumed to be 1 inch (2.54 cm). Figure 3.2 illustrates the setup of two rotational stages equipped with two of the 2-diaphram fly-ear inspired devices or one of the 3-diaphragm device for sound localization in the azimuth and elevation directions.

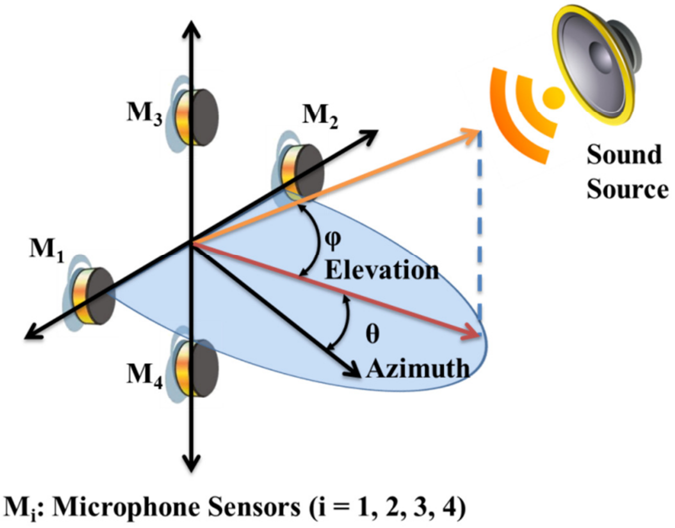


Figure 3.1: Conventional microphone array arrangement, where θ and φ are the azimuth and elevation angles.

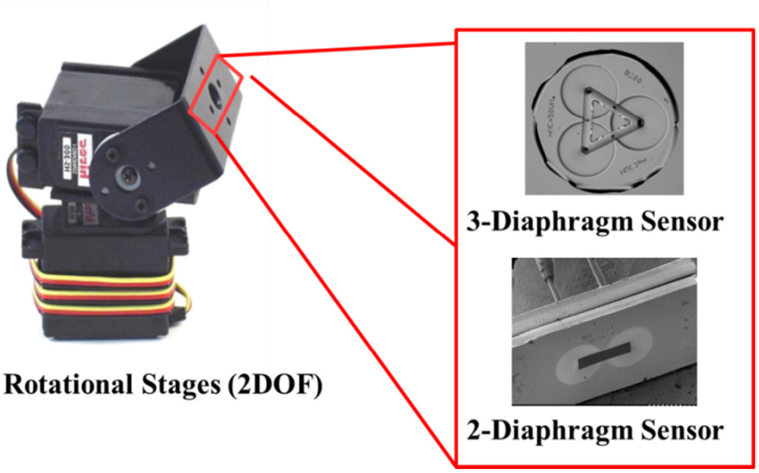


Figure 3.2: Sound localization with two rotational stages and the fly ear inspired sensors.

3.1.1 Objective Functions

In this section, two objective functions are developed and investigated with a goal of achieving the least number of iterations and smallest error in sound source angle determination. The first objective function, called the 3-Point Algorithm (3PA), is defined as:

$$J = -\partial ITD / \partial \theta - \partial ITD / \partial \varphi \, , \qquad (3.3)$$

where $\partial ITD/\partial\theta$, and $\partial ITD/\partial\varphi$ are the directional sensitivity of the sensor (or the microphone array) in the azimuth (θ) and elevation (φ) directions, respectively. Figure 3.3 shows the contour plot of the negative of objective function described in Eq. (3.3), which achieves the global minimum value at the origin ($\theta = \varphi = 0^{o}$), when the sensor is pointing to the sound source.

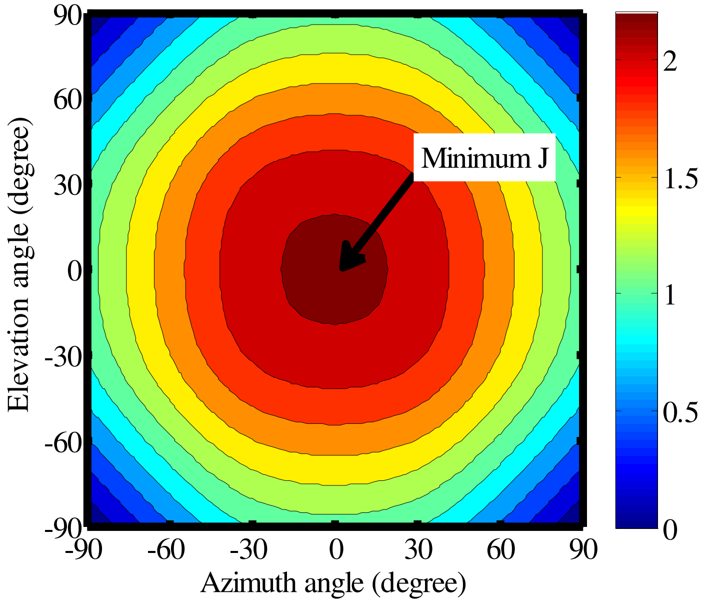


Figure 3.3: Contour plot of the objective function given by Eq. (3.3).

To evaluate the change in the objective function (Eq. (3.3)) via the finite difference approximation, three data points are needed in each direction, e.g., $ITD(\theta_i)$, i=1,2,3, in the azimuth direction, where θ_1, θ_2, and θ_3 represent the starting position, the end position after one moving step, and the end position after two moving steps, respectively. Let J_1 denote the objective function based on the first two data points, and J_2 to be based on the last two data points. Accordingly, the change in the objective function can be obtained as:

$$\delta J = J_2 - J_1 , \tag{3.4}$$

The second objective function, called 2-Point Algorithm (2PA), is defined as:

$$J = ITD_\theta^2 + ITD_\varphi^2 , \tag{3.4}$$

where ITD_θ, and ITD_φ are the ITD in the azimuth and elevation directions respectively.

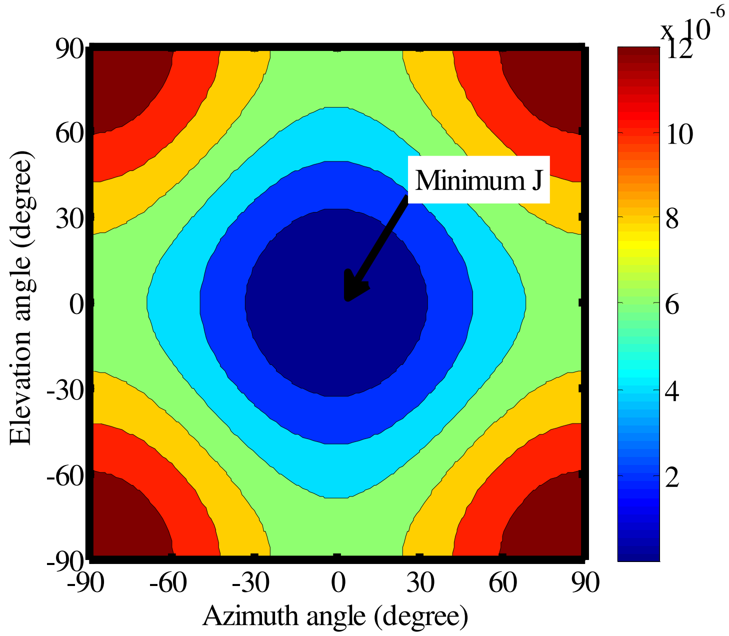


Figure 3.4: Contour plot of the objective function given by Eq. (3.4).

The objective function J defined in Eq. (3.4) has a global minimum value (*ITD* = 0) when the sensor is pointing to the sound source direction, as shown in Figure 3.4. To evaluate the change in the objective function given by Eq. (3.4), two data points in each direction are needed, namely, $ITD(\theta_1)$, and $ITD(\theta_2)$ where θ_1 represents the current position and θ_2 represents the end position after one moving step. The change in this objective function can similarly be calculated by Eq. (3.4).

3.1.2 Proposed Localization Schemes

After defining the objective functions a control function needs to be applied to relate the change in the objective function to the step size for moving the rotational stages using the gradient descent method. Here, four model-free localization schemes are proposed, which are the following:

$$\begin{bmatrix}\Delta\theta \\ \Delta\varphi\end{bmatrix} = -\mu\,\delta J \begin{bmatrix}\delta u_1 \\ \delta u_2\end{bmatrix}, \tag{3.4}$$

$$\begin{bmatrix} \Delta\theta \\ \Delta\varphi \end{bmatrix} = -\begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix} \delta J \begin{bmatrix} \delta u_1 \\ \delta u_2 \end{bmatrix}, \tag{3.5}$$

$$\begin{bmatrix} \Delta\theta \\ \Delta\varphi \end{bmatrix} = -\mu \begin{bmatrix} \delta J_\theta & 0 \\ 0 & \delta J_\varphi \end{bmatrix} \begin{bmatrix} \delta u_1 \\ \delta u_2 \end{bmatrix}, \tag{3.6}$$

and

$$\begin{bmatrix} \Delta\theta \\ \Delta\varphi \end{bmatrix} = -\mu\, \delta J \begin{bmatrix} w_{11} & 0 \\ 0 & w_{22} \end{bmatrix} \begin{bmatrix} \delta u_1 \\ \delta u_2 \end{bmatrix}. \tag{3.6}$$

Here, $\Delta\theta$ and $\Delta\varphi$ are the rotation angles in the azimuth and elevation directions respectively. μ, α, and β denote the gain values, and δJ is the change in the objective function, w_{11} and w_{22} are weighting factors that will be defined later depending on the objective function and δu_1 and δu_2 are the rotation directions (clockwise (+1) or counter clockwise (-1)) in the azimuth and elevation directions, respectively. The first three schemes given by Eqs. (3.4) - (3.6) are called Non Weighted Algorithm (NWA), while the last one given by Eq. (3.6) is called the Weighted Algorithm (WA). The characteristics of each scheme are investigated through simulations at different initial positions of the sound source.

The first localization scheme given by Eq. (3.4) uses the same gain value μ and objective function change δJ to control both azimuth and elevation rotational stages. The drawback of using this scheme is the large overshoot. This overshoot is related to the fact that when the azimuth and elevation angles are not close to each other, one term in the objective functions will be dominant. This will result in an overshoot in one of the directions until the two angles (azimuth and elevation) become close to each other.

The second localization scheme given by Eq. (3.5) uses the same δJ but different gain values (α and β) to control the two rotational stages. The drawback of using this scheme is the need for continuously tuning the gains at different sound positions, which is experimentally impractical unless the gain is changed adaptively.

The third scheme given by Eq. (3.6) uses the same gain value μ but different δJ to control the two rotational stages. This case limits the working range of the system, in which the sound position in the azimuth and the elevation directions should be equal or close to each other, otherwise the system performance will be undesirable. For example, since the two stages are controlled using the same gain with different changes in the objective function, using a larger gain will cause the direction with the smaller δJ to move with a large step. Furthermore, Since in the linear region, using the directional sensitivity, δJ will be constant and thus there will be no convergence toward the zero angle in that direction unless the azimuth and elevation angles are initially equal or close to each other. On the other hand, using a smaller gain may solve the problem but at the cost of convergence time.

Finally, the fourth localization scheme described by Eq. (3.6) uses the same gain value μ and the same δJ with weighting factors. For the objective function defined in Eq. (3.3), the weighting factors are defined as

$$w_{11} = \frac{(\partial ITD / \partial \theta)^2}{\sqrt{(\partial ITD / \partial \theta)^4 + (\partial ITD / \partial \varphi)^4}}, \tag{3.7}$$

$$w_{22} = \frac{(\partial ITD / \partial \varphi)^2}{\sqrt{(\partial ITD / \partial \theta)^4 + (\partial ITD / \partial \varphi)^4}}. \tag{3.8}$$

The localization algorithm based on these weighting factors and the control scheme defined in Eq. (3.6) is called the Weighted 3-Point Algorithm (W3PA). Using the same scheme, the weighting factors can also be defined for the objective function

(3.4) as:

$$w_{11} = \frac{ITD_{\theta}^{2}}{\sqrt{ITD_{\theta}^{4} + ITD_{\varphi}^{4}}}, \tag{3.9}$$

$$w_{22} = \frac{ITD_{\varphi}^{2}}{\sqrt{ITD_{\theta}^{4} + ITD_{\varphi}^{4}}}. \tag{3.10}$$

In this case the localization algorithm is called the Weighted 2-Point Algorithm (W2PA).

In Figure 3.5, the azimuth and elevation angles obtained with different algorithms, using the objective functions defined by Eqs. (3.3) and (3.4) and the localization schemes described in Eqs. (3.4) and (3.6), for an initial position with 75° azimuth and 20° elevation angles are compared. It can be seen that the number of iterations needed for the non-weighted two-point and weighted three-point algorithms is larger than those for the weighted two-point and non-weighted three-point algorithms. In addition, the overshoot happens in the non-weighted two-point, the non-weighted three-point, and the weighted three-point. The weighting did not help in reducing the overshoot because of the large gain that is chosen based on the criteria in terms of the number of iterations required to converge and the steady state error. To reduce the overshoot, a smaller gain can be used at the cost of increasing the number of iterations to localize the sound. This means that there is a tradeoff between the number of iterations and the level of overshoot when specific algorithm is used.

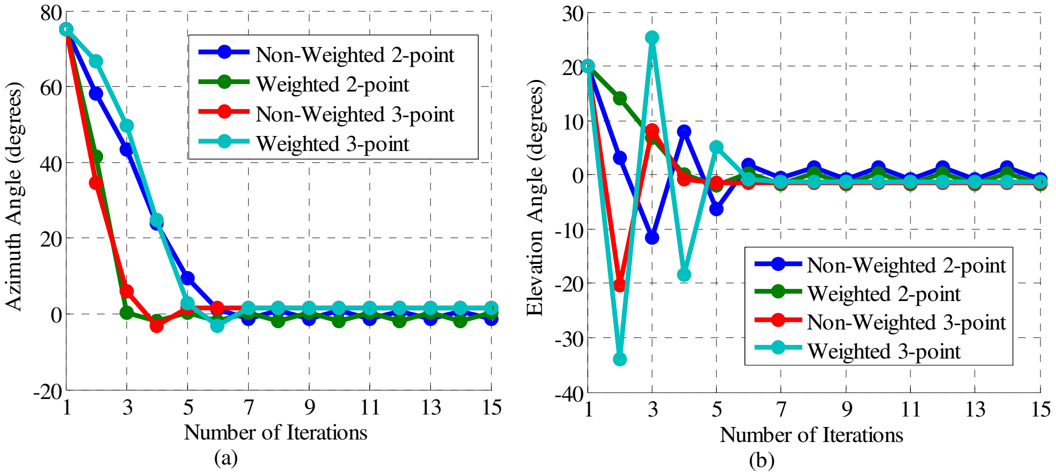


Figure 3.5: Time history of (a) azimuth and (b) elevation angles obtained with different algorithms for the initial sound source position θ =75^{o} and φ = 20^{o}.

In Figure 3.6, the directional sensitivity updates in the azimuth and elevation directions obtained with different algorithms are compared. The overshoot observed earlier for the weighted three-point algorithm (see Figure 3.5(b)) agrees with the overshoot shown in Figure 3.6(b). The overshoot in the non-weighted two-point and the non-weighted three-point algorithms does not affect the sensitivity plot due to the fact that the conventional microphone array or the fly-ear inspired sensor has linear response within the incident angle region of -30^{o} to 30^{o} azimuth and elevation directions. The non-weighted two-point and weighted three-point algorithms require more iterations to achieve convergence than the weighted two-point and non-weighted three-point algorithms.

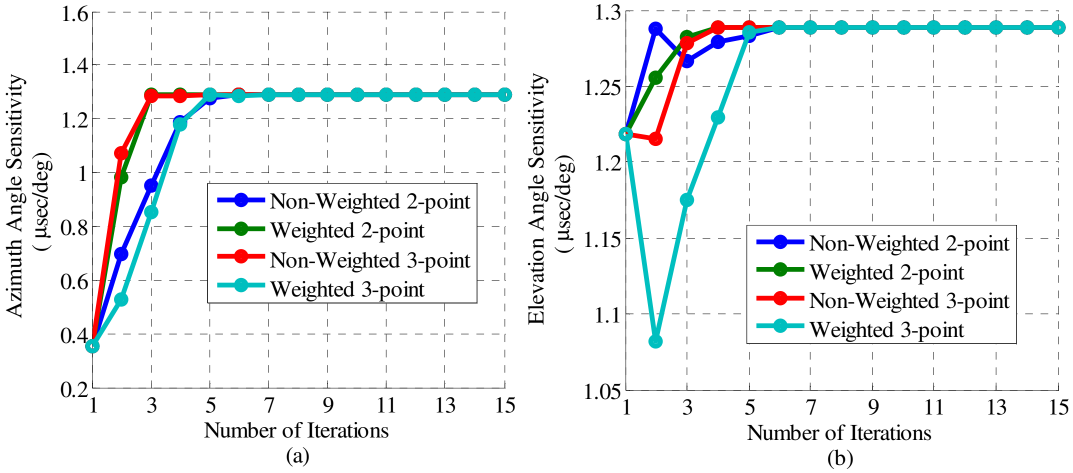


Figure 3.6: Time history of (a) azimuth and (b) elevation directions sensitivities obtained with different algorithm for the initial sound source position θ =75^o and φ = 20^o.

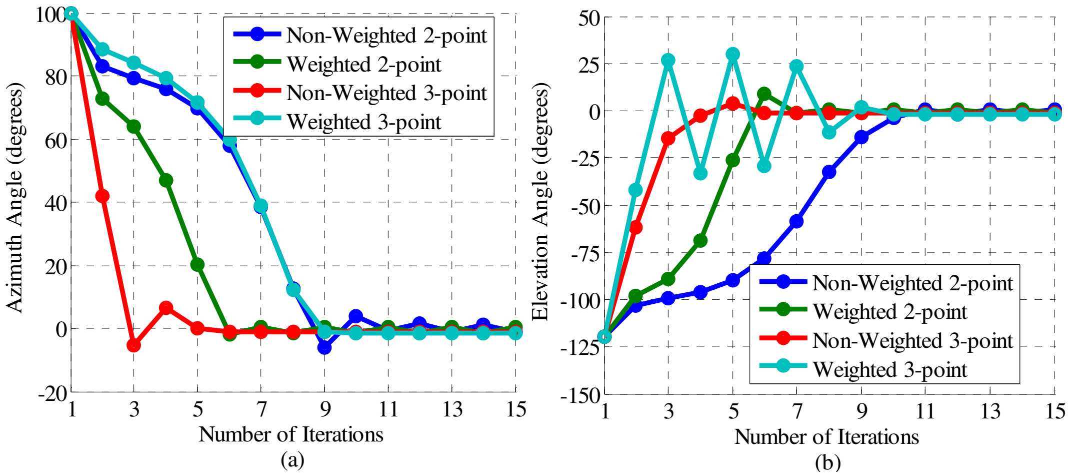
Table 3-1:Performance comparison of proposed algorithms

Localization algorithm	Number of iterations	Steady state error azimuth (deg)	Steady state error Elevation (deg)
Non-weighted 3-point	5	1.45	1.48
Weighted 3-point	7	1.44	1.35
Non-weighted 2-point	7	1.02	1.02
Weighted 2-point	5	1.00	1.00

In Table 3-1, the localization algorithm performance obtained with different algorithms for the initial positions of θ = 75^o, and φ = 20^o are compared. As concluded earlier the number of iterations required for the non-weighted three-point and the weighted 2-point algorithm are less than that for the weighted three-point and the non-weighted 2-point algorithms. The steady state error, which is defined as the average of the steady state azimuth/elevation angle with respect to zero, ranges from 1~1.5 degrees for all algorithms in both azimuth and elevation directions.

3.1.3 Ambiguity Problem

The algorithms discussed previously have been shown to be able to localize the sound source in two dimensions (2D), namely azimuth and elevation directions. Here an answer is sought for the following question: can the proposed algorithms be used to localize a sound propagating from the back (i.e., ambiguity problem)? To answer this question, the initial sound direction in both azimuth and elevation directions are changed to $\theta = 100^{o}$, and $\varphi = -120^{o}$ to represent a sound from the back of the sensor. The results are shown in Figure 3.7.



(a)

(b)

Figure 3.7: Time history of (a) azimuth and (b) elevation angles obtained with different algorithms for the initial sound source position $\theta = 100^{o}$ and $\varphi = -120^{o}$.

The results show that the numbers of iterations required to localize a back sound with the non-weighted 2-point and weighted 3-point algorithms are more than those obtained with the weighted two-point and non-weighted three-point algorithms. This is similar to the previous case for localizing a front source. The non-weighted two-point and the non-weighted three-point algorithms exhibit overshoot due to the gain value and the dominating *ITD* values associated with the non-weighted cases. To reduce the overshoot, a smaller gain can be used at the expense of increasing the number of iterations. The control performance is more chattering for the weighted three-point algorithm than that for the other three methods. The reason is that for the specific initial position chosen for the simulation, the derivative $\partial ITD/\partial\varphi$ is larger than $\partial ITD/\partial\theta$ so that the localization scheme has more effect on the elevation rotations. Furthermore, results indicate that all the algorithms can be used to localize a sound from the back of the sensor without the "*ambiguity problem*". To explain this, for simplicity, consider sound source localization in a 1D case (azimuth only). Assume that the sign of the measured *ITD* according to the configuration shown in Figure 3.1 is positive. Then, the stage will rotate clockwise, which causes an increase in the measured *ITD* and a decrease in the sensitivity until the measured *ITD* reaches the maximum and the sensitivity reaches the minimum. After this point, the sound source becomes in frontal position and the localization process will continue till the sound source is located. This process of localizing a back source needs more effort In Figure 3.8, the sensitivity histories obtained with different algorithms in the azimuth and elevation directions for a back sound source are compared. The sensitivity requires less number of iterations to stabilize than the angle, due to the fact that when the sound position is in the linear region the sensitivity becomes constant. It is also noted that the large fluctuations in the elevation angle history shown in Figure 3.7(b) for the weighted three-point did not show up in Figure 3.8.

compared with the front source case, which is reflected on the number of iterations as shown in Table 3-2.

Table 3-2: Proposed algorithms performance for localizing a back source

Localization algorithm	Number of iterations	steady state error azimuth (deg)	steady state error elevation (deg)
Non-weighted 3-point	6	1.03	1.00
Weighted 3-point	9	1.37	1.90
Non-weighted 2-point	11	1.24	1.02
Weighted 2-point	7	1.00	1.00

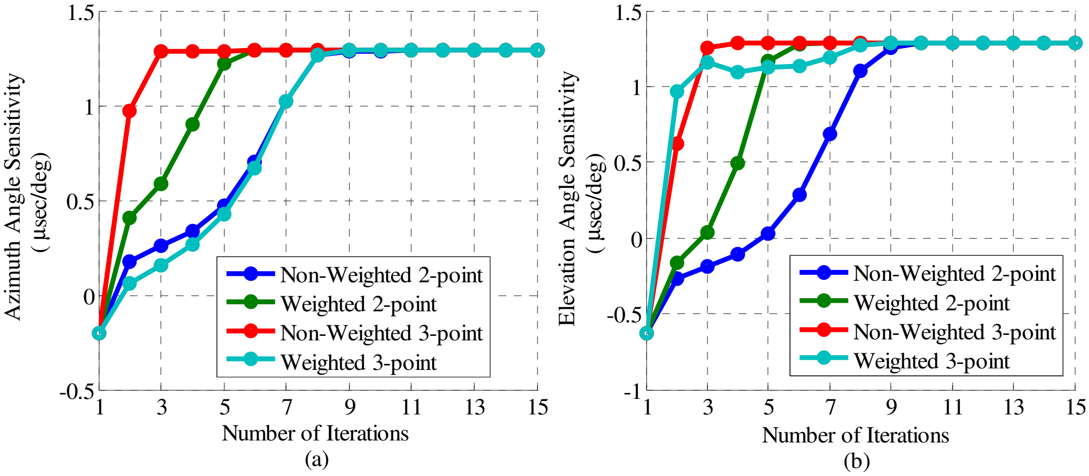


Figure 3.8: Time history of (a) azimuth and (b) elevation directional sensitivities obtained with different algorithms for the initial sound source θ = 100° and φ = -120°.

Table 3-2 summarizes also the localization algorithm performance obtained with different algorithms for the initial positions θ = 100° and φ = -120°. As concluded earlier, the number of iterations for the non-weighted three-point and the weighted two- point algorithm is less than that for the weighted three-point and the non-weighted two- point. The absolute localization error (steady state error) in all cases ranges from 1~1.9 degrees for both azimuth and elevation directions.

The gain values used in the simulations are listed in Table 3-3. These values are obtained by using a MATLAB code that searches for the gains leading to the smallest number of iterations and the smallest absolute error at steady state.

Table 3-3: Gain values used in different algorithms

Algorithm	Non-weighted 2-point	Weighted 2-point	Non-weighted 3-point	Weighted 3-point
gain μ	0.034	0.073	320	440

In summary, the model-free gradient descent method for the sound localization without the problem of ambiguity has been demonstrated in the simulations. However, the main drawback of this method is that it is time consuming especially when the sound source is moving in space following an unknown path since this method is based on physical motion iterations not numerical iterations. In the next section a non-iterative approach is investigated in which the sound localization is achieved by modeling the fly-ear sensor using fuzzy logic.

3.2 Fuzzy Logic Method

To carry out robotic sound source localization, the uncertainties in *ITD/IPD* measurements should be considered. In addition, as discussed earlier, for a fly-ear inspired sensor, the relation between the incident angle and the directional cue is linear within the -30^{o} to 30^{o} sound incident angle. Outside this linear region the measurement of directional cue behaves nonlinearly with respect to the sound incident angle. This nonlinear relationship also varies with respect to different stimulus frequencies. To address this, it is essential to study a localization algorithm with which one can make use of the entire range of incident angles regardless the frequency of the sound source. Beside the iterative method discussed in Section 3.1, several methods can be used to localize the sound source in general. The first method is based on the assumption that the mIPD has a constant amplification of the IPD obtained from a conventional microphone pair, which can be represented as

$$\begin{aligned} mIPD &= M \times IPD \\ IPD &= 2\pi \frac{d}{\lambda} \sin(\theta) \end{aligned}, \tag{3.11}$$

where d is the separation distance of the microphones, λ the wavelength, and θ the sound incident angle. However, the main drawback of this model is that this model can only be used for a specific frequency and within certain incident angle range, at which the *IPD* is amplified by a certain value M and the inverse of the model can take values within the range of -1 to 1.

In another method, the mathematical model of the sensor presented in Chapter 2 can be used to estimate the incident angle. The main drawback of the mathematical model is its complexity. Hence, a closed-form relationship that maps the sensor information to the incident angle can hardly be established. The least square error analysis can be used to determine the incident angle, but this method is iterative in nature and it requires a long execution time that increases dramatically with increasing the estimation accuracy and the number of unknowns.

One good method that can address the challenge of using the fly-ear inspired sensor for sound localization is the Fuzzy Logic (FL) method. The FL has been widely used because of its simplicity and flexibility. Furthermore, the FL method can be used to model nonlinear systems. In addition, for any set of input-output data, a fuzzy model can be built using some adaptive techniques such as adaptive Neuro-Fuzzy Inference Systems (ANFIS), which are provided in the Fuzzy Logic Toolbox of MATLAB. Furthermore, fuzzy models for the problems that involve human expert can be built using the human natural language according to the experience of the people who totally understand the system of interest. Therefore, fuzzy models can be built based on imprecise or uncertain information, which makes it suitable for applications that involve measurements uncertainties. The FL method can also be used with conventional control systems to make the system implementation simpler [52], [53].

FL is a logical system that extends the Boolean logic concept that has two values, (i.e., even completely on or off) to partial value concept that ranges between completely on and off which is totally different from the traditional multivalued logical systems. The basic idea of FL is that one can linguistically define some variables, such as temperature, age, length, etc. These variables are called "*linguistic variables*" which have values that can be defined linguistically as well. These values are called "*linguistic values*" of the linguistic variables. Imprecision mentioned above came from the fuzzy way of defining the system or problem of interest. Fuzzy rule is another concept in FL, which can be represented linguistically as:

IF A is a Then X is x,

where A is the linguistic variable that represents the input of the system that has a linguistic value a. The FL inputs in the fuzzy rules are called "*antecedents*". Every linguistic value is called a "*membership function*". These membership functions can be defined in many different shapes, including triangular, bell, Gaussian, trapezoidal, sigmoidal, etc. X represents the output variable that has a linguistic value x. The FL outputs in the fuzzy rules are called "*consequents*". To have a meaningful system there should be a translation mechanism that determines the relation between the fuzzy antecedents and consequents. Fuzzy calculus is the mechanism used to do the transformation from the human linguistic representation into a meaningful quantization. The whole system including the antecedents, consequents, the if-then rules, and the fuzzy calculus is called the Fuzzy Inference System (FIS). Fuzzy inference process consists of five steps: fuzzification of the input variables, application of the fuzzy operator (AND or OR) in the antecedent, implication from the antecedent to the consequent, aggregation of the consequents across the rules, and defuzzification [54]. In the fuzzification process, the linguistic values of the antecedents are defined. When the fuzzy rule has more than one input, the fuzzy operator is used to obtain a single value that represents the antecedent output of that rule. Generally in the FIS, each rule has a weight (ranging between 0 and 1) that should be considered in the implication process represented by the "THEN" word in the fuzzy rule discussed previously (default weight is 1). The implication process has two methods: min (minimum), which truncates the output, and prod (product), which scales the output. Since the FIS has many rules, the output of these rules should be combined to get a single fuzzy set so as to make a decision. This process is called "*aggregation"* of the consequents, which is applied across all the rules. Finally, defuzzification is the process of getting a single value from the aggregated single output fuzzy set which represents the decision. Five different methods can be used for defuzzification, namely, centroid, bisector, middle of maximum (the average of the maximum value of the output set), largest of maximum, and smallest of maximum. Figure 3.9 shows the components of the FIS and the flow of the five steps explained above.

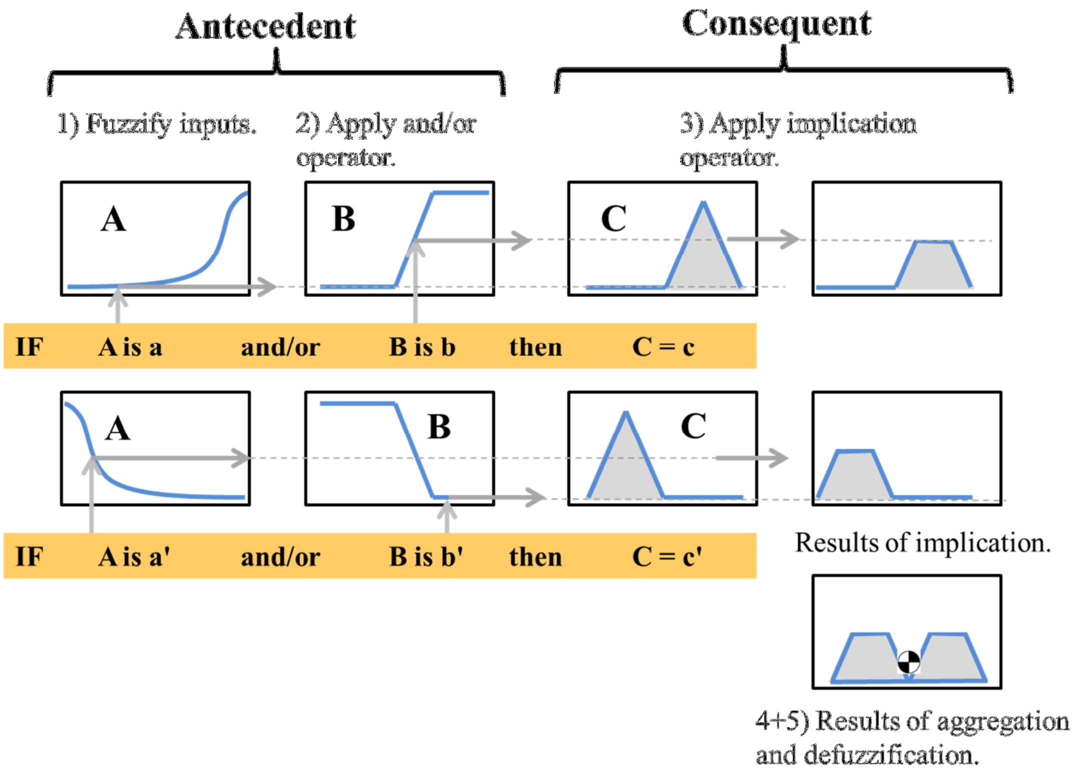

Antecedent
Consequent
1) Fuzzify inputs.
2) Apply and/or operator.
3) Apply implication operator.
A
B
C
IF A is a and/or B is b then C = c
A
B
C
IF A is a' and/or B is b' then C = c'
Results of implication.
4+5) Results of aggregation and defuzzification.

Figure 3.9: Flow chart for Fuzzy Inference System.

The most commonly used FIS types are the Mamdani-type and Sugeno-type [54]. Mamdani-type FIS, which was proposed by Ebrahim Mamdani in 1975, has been used more than Sugeno-type since it was the first control system that was implemented to control a steam engine and boiler combination by translating the experience of a human operator to a FIS. The main difference between Mamdani-type and Sugeno-type FIS is that the output membership functions in the Mamdani-type are fuzzy as well as the inputs [55]. The output membership functions in the Sugeno-type FIS are even constants, which are called the zero-order functions, or linear, which are called first-order functions of the inputs. Using such output simplifies the defuzzification step. The zero and first order functions for two inputs (x and y) case can be represented as the following:

$$z = c \quad \text{(constant)} \tag{3.12}$$

$$z = a \times x + b \times y + c\,. \tag{3.13}$$

To build a fuzzy model for the input-output data set, one of the soft computing techniques, namely neural network, can be combined with fuzzy logic so as to adaptively move and/or widen/narrow the membership functions used to match the input-output set. This neuro-fuzzy system is called *Adaptive Neuro-Fuzzy Inference System* (ANFIS) [54], which has been included in the fuzzy logic toolbox in MATLAB.

3.2.1 Neuro-Fuzzy Inference System (Grid Partitioning vs. Subtractive Clustering)

In Figure 3.10, the IPD of the fly-ear inspired acoustic sensor as a function of the incident azimuth angle at different sound frequencies are obtained by using Eq. (2.62) , which is nonlinear outside the region of -30^{o} to 30^{o}. This nonlinearity is a function of the stimulus frequencies. The parameters used to obtain the simulation results are based on the fly ear parameters listed in Table 2-1.

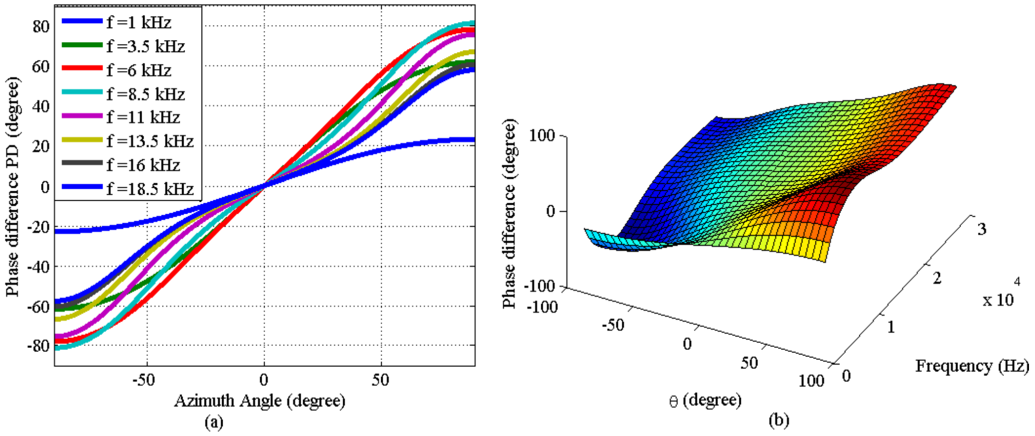


Figure 3.10: (a) IPD versus incident azimuth angle and (b) 3D surface of the mIPD at different incident angles θ and frequencies using the fly ear parameters listed in Table 2-1.

Based on these results, a normalized truth table can be generated that involves all the theoretical values of the measurements at different frequencies. The maximum absolution value of the theoretical measurement at each frequency is obtained when the incident angle is at $\pm 90^o$. These maximum values are used to normalize the response at each frequency. The obtained data is then used to model the sensor response by using fuzzy logic toolbox in MATLAB.

The generated truth table can be used in ANFIS editor to train the data and test Sugeno-type fuzzy systems to get the sensor fuzzy model. ANFIS GUI can be started to achieve the following four different tasks [56]:

1. loading, plotting, and clearing the data set,
2. generating or loading the Initial FIS structure,
3. training the FIS,
4. validating the trained FIS.

After loading the data to the ANFIS GUI, since the initial FIS structure has to be generated, one has to choose between the "*Grid Partition*" method and the "*Subtractive Clustering*" method. The grid partition (GP) method can be used to divide the input space into grids. A small number of membership functions for each input is needed so the problem of having a large number of fuzzy rules is not encountered, which will complicate the computation of the output [57]. For this reason, subtractive clustering (SC) is usually used to estimate the number of clusters and the clusters centers needed for a given input-output data set when the number of clusters and cluster centers are unknown since it is a fast, and one-pass algorithm [58].

Accordingly, a fuzzy model using each above-mentioned method is constructed and the results from each model are compared. The obtained data set has two inputs and one output. When the grid partitioning method is used, some parameters should be initialized to generate the FIS model, which includes the number of inputs and outputs, the number of membership functions, and type of the membership functions for the inputs and outputs. In this study, seven Gaussian membership functions for each input are initialized. For the output membership function, the linear type is chosen to get a smoother response.

After initializing the FIS structure, the data will be trained to obtain the best fuzzy model according to the initialization. The training error as a function of the number of epochs (i.e., the number of iterations needed to train the data) is shown in Figure 3.11. After training, the model can be evaluated to find out how well it matches the original input-output data set. Usually a different set of data called the "*checking data*" is used to test the model.

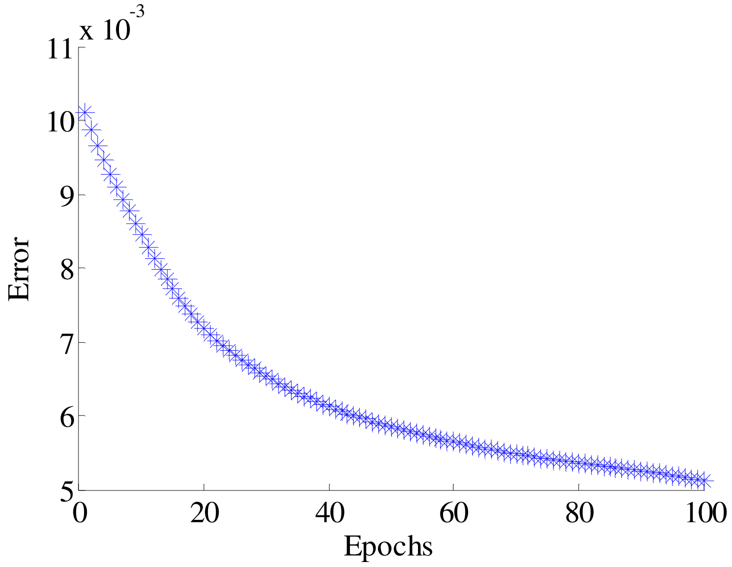


Figure 3.11: FIS training error history using the Grid Partitioning method.

The resulted model as illustrated in Figure 3.12 basically has two inputs and one output. The two inputs are the normalized frequency and the normalized mIPD, with seven membership functions for each input, which are specified in the fuzzy structure initialization (as shown in Figure 3.13). The one output is the normalized bearing angle which is normalized by the maximum estimated angle (90^o), with 49 membership functions. A total of 49 fuzzy rules are constructed by multiplying the number of membership functions in the inputs. The relationship between the two inputs and the output is shown in Figure 3.14. The fuzzy calculus used to achieve the mathematical operations in the fuzzy model include the AND method that is chosen to be the product operation and the OR method that is chosen as the "*probabilistic OR*" (PROBOR) method (also known as the algebraic sum), which is calculated according to the following equation:

$$probor(a,b) = a + b - ab. \tag{3.14}$$

For the implication and aggregation methods, MIN and MAX (minimum and maximum) operations are chosen respectively. For the defuzzification method, the "*weighted average*" (WTAVER) is chosen. WTAVER can be expressed as a function of the rule firing strength and the rule output; that is,

$$\text{Output} = \frac{\sum_{i=1}^{N} w_i \times z_i}{\sum_{i=1}^{N} w_i}, \tag{3.15}$$

where w_i is the rule firing strength (output of the antecedents after applying the AND/OR operator) and z_i represents the output from the zero or first order Sugeno functions.

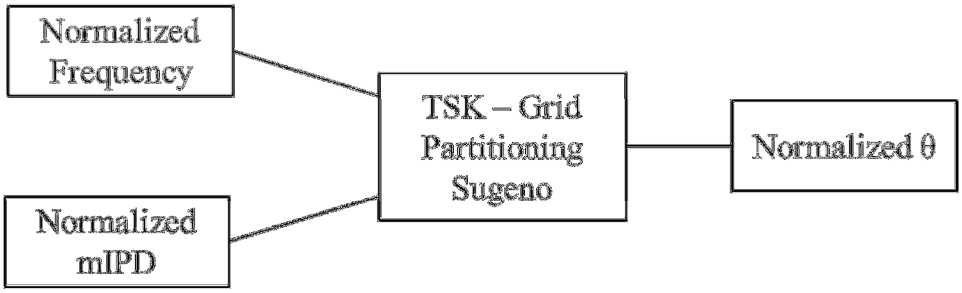


Figure 3.12: Final Fuzzy model using the Grid Partitioning method.

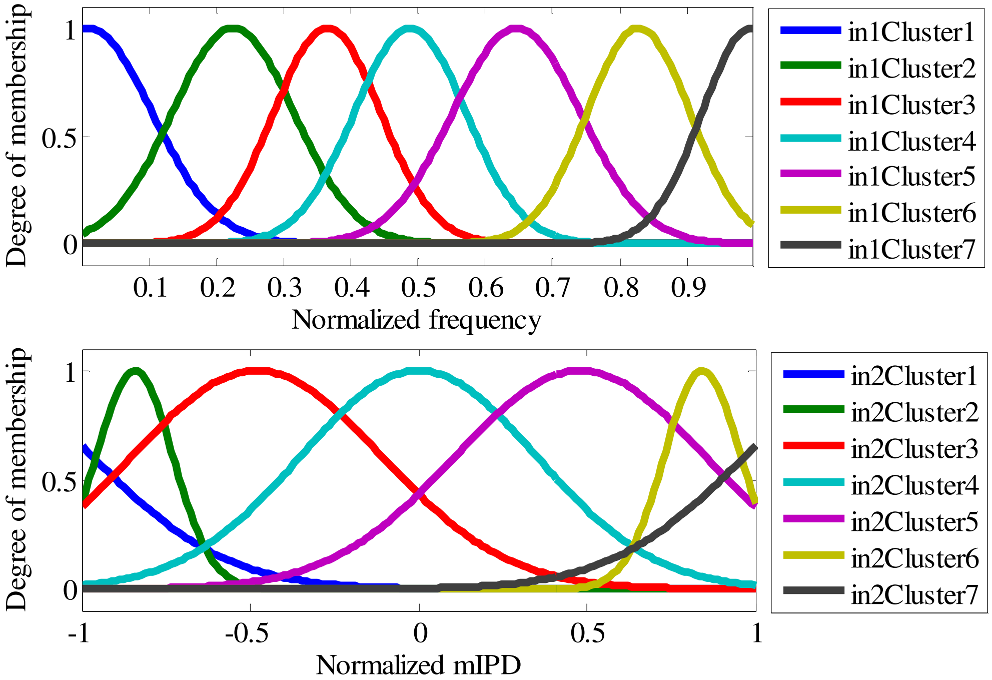


Figure 3.13: Normalized inputs membership functions using the grid partitioning method.

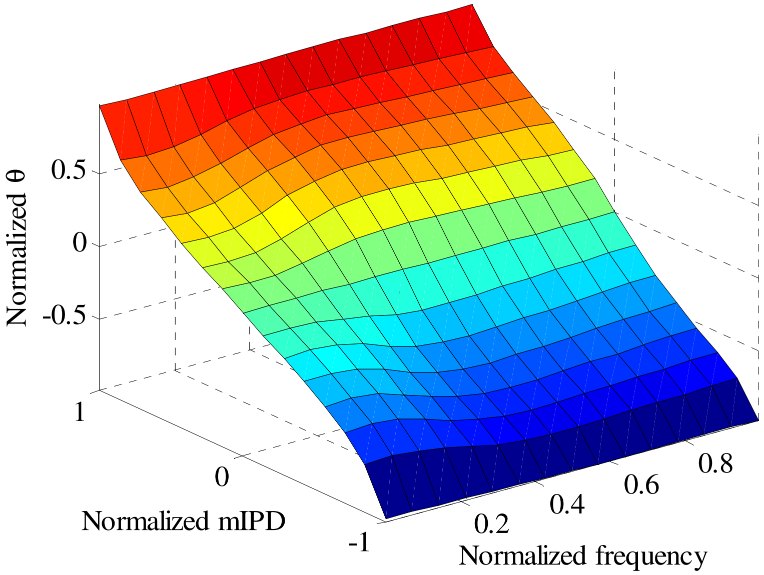


Figure 3.14: Fuzzy model surface using the grid partitioning method.

Next, the subtractive clustering algorithm is used to construct the fuzzy model. After loading the data set from the workspace, the FIS parameters are initialized. The parameters need to be initialized include the range of influence, squash factor, accept ratio, and reject ratio. The "*range of influence*" represents the cluster radius considering the data space as a unit hypercube. A small cluster radius will yield to large number of clusters, which results in more fuzzy rules. On the other hand, a large cluster radius yields to a smaller number of clusters, which results in less fuzzy rules. The "*squash factor*" is a factor that is multiplied with the radius values that determine the neighborhood of a cluster center, so as to reduce or even eliminate the outlying points belonging to that cluster (a default is: 1.25). The "*accept ratio*" is a threshold factor that is a fraction of the potential threshold of the first cluster center. Above this threshold another data point is accepted as a new cluster center (default value of 0.5). The "*reject ratio*" is another threshold factor that is a fraction of the potential threshold of the first cluster center. Below this threshold a data point will be rejected as the cluster center (default value of 0.15) [59]. The default values are used in the simulations. By confirming the FIS parameters the clustering process starts grouping the data points that share certain characteristics. Similar to the model obtained using the GP method, after initializing the structure the data will be trained to get the best fuzzy model according to the initialization. The obtained training error history as a function of the number of epochs needed to train the data is shown in Figure 3.15. In addition, after training, the model can be tested to check how well it matches the original input-output data set.

Compared with the GP algorithm based fuzzy model, the number of rules and the number of output membership functions used in the SC algorithm based model are less. Next, investigation is carried out to show whether this affects the fuzzy model performance.

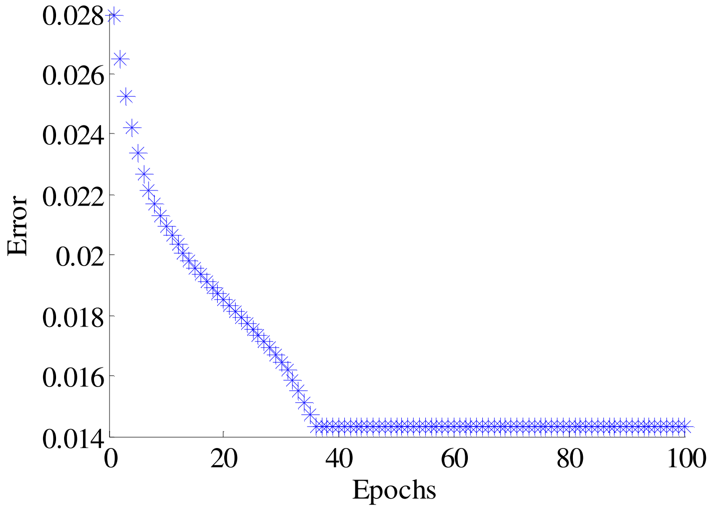


Figure 3.15: FIS training error history using the Subtractive Clustering method.

By using the checking data set, the model can be tested to investigate how the model output matches the original data set. The resulted model shown in Figure 3.16 has the same number of inputs (i.e., the normalized frequency and the normalized *mIPD* with seven Gaussian membership functions as shown in Figure 3.17) and one output. The output is the normalized bearing angle with only 7 linear membership functions which is much less than that for the GP method. This will be useful when the fuzzy model is implemented experimentally to reduce the processing time and needed memory. The relationship between the two inputs and the output of the SC method is shown in Figure 3.18, which appears to be similar to that of the GP method. The fuzzy calculus used to achieve the mathematical operations in this fuzzy model is the same as the ones used in the GP method.

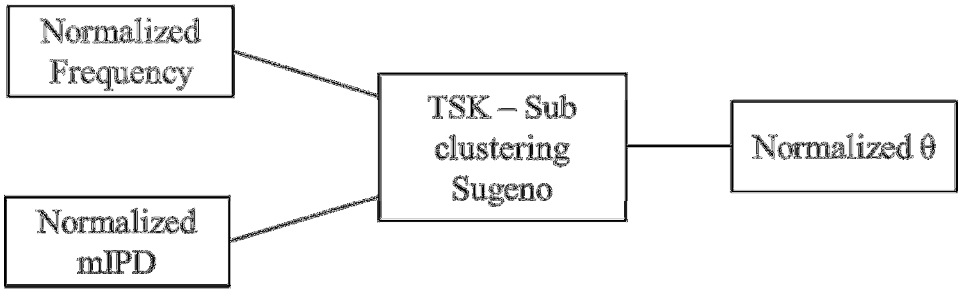


Figure 3.16: Schematic of fuzzy model using the subtractive clustering method.

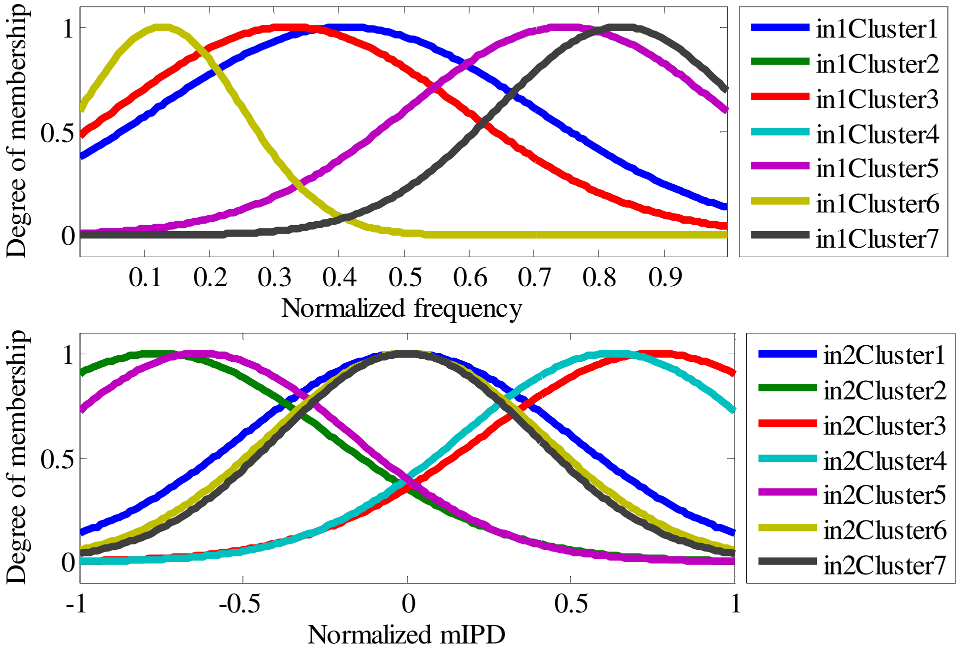


Figure 3.17: Normalized inputs membership functions used in the subtractive clustering method.

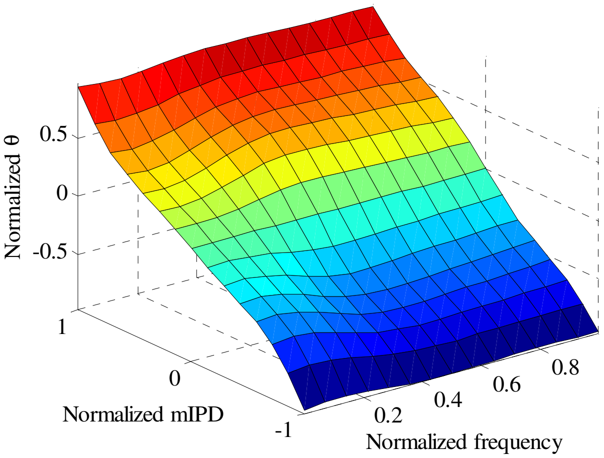
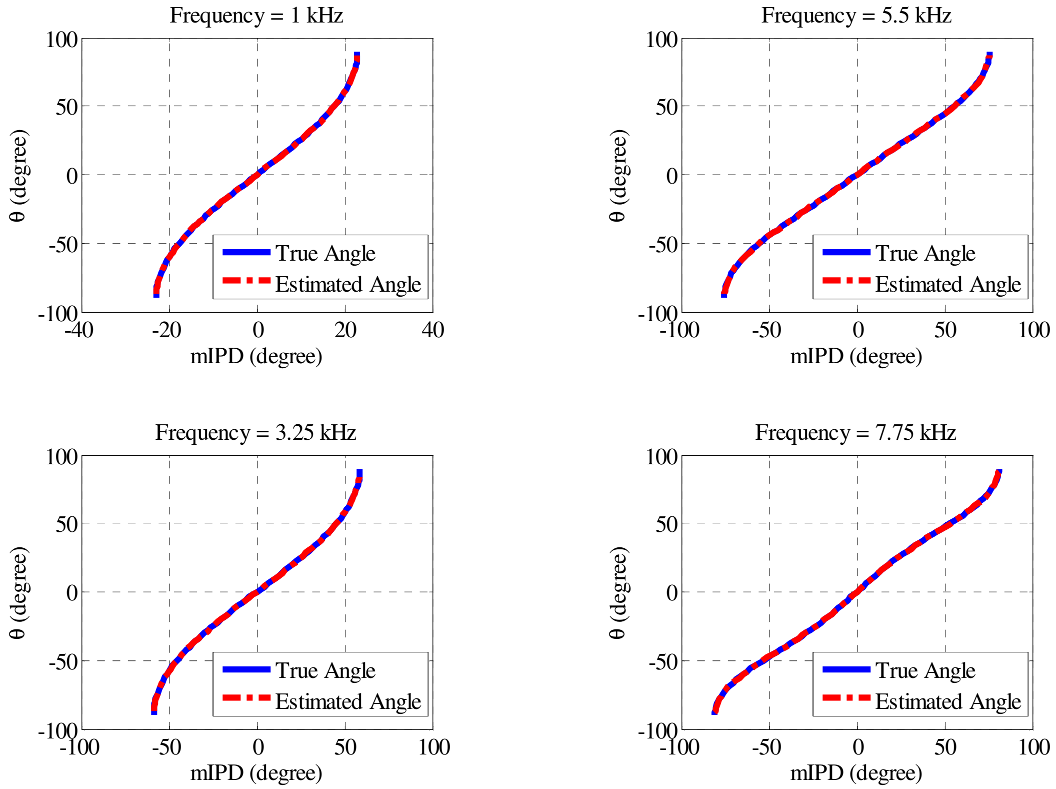


Figure 3.18: Fuzzy model surface obtained with the subtractive clustering method.

The bearing angle estimation obtained with both of the two fuzzy models are compared with that obtained by using the theoretical mIPD as shown in Figure 3.19 and Figure 3.20. It can be found that both models work properly with small estimation error. Since the model generated using the subtractive clustering method has less number of rules and output membership functions, it can help dramatically reduce the computational time. Therefore, this model will be used in the later studies.

In this subsection, two fuzzy models were developed for the fly-ear inspired sensor. The first model used the grid partition method and the second model used the subtractive clustering method to construct the fuzzy model. Although the simulation results showed that both models work properly, the model created using the subtractive clustering method has less number of rules which will be a critical issue when all the processing and decision making is conducted on board and online. To further validate the fuzzy models, previously collected experimental data will be used to test the fuzzy models which will be discussed in the next subsection.



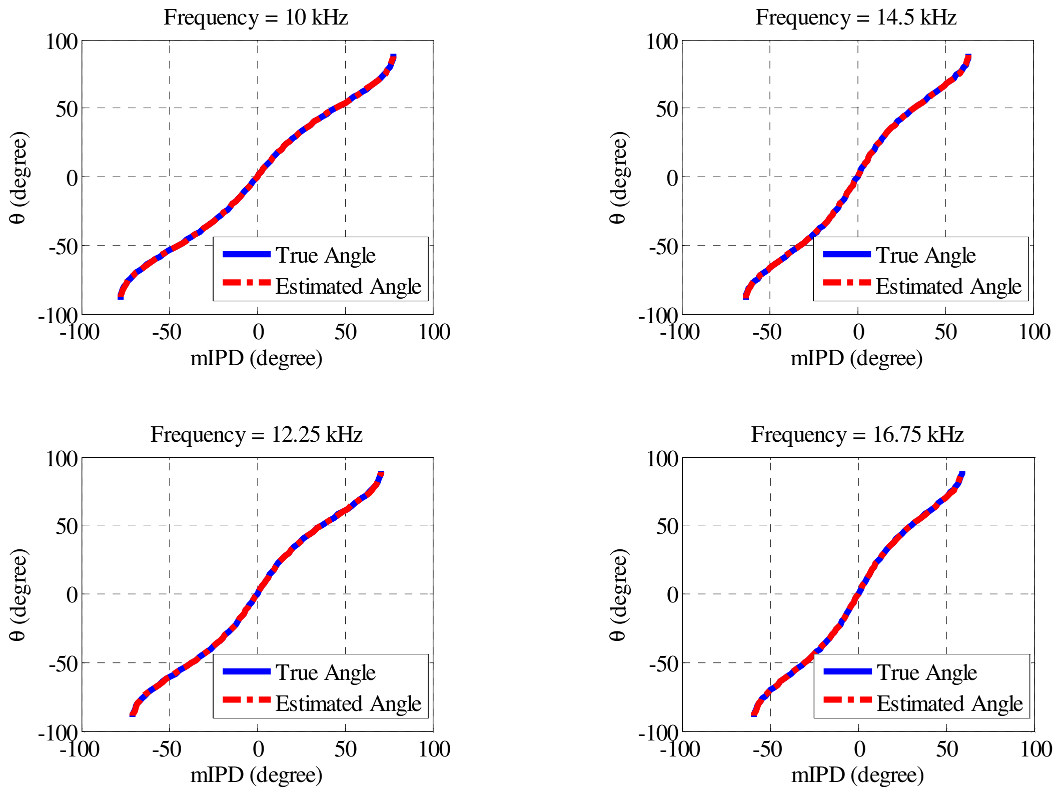
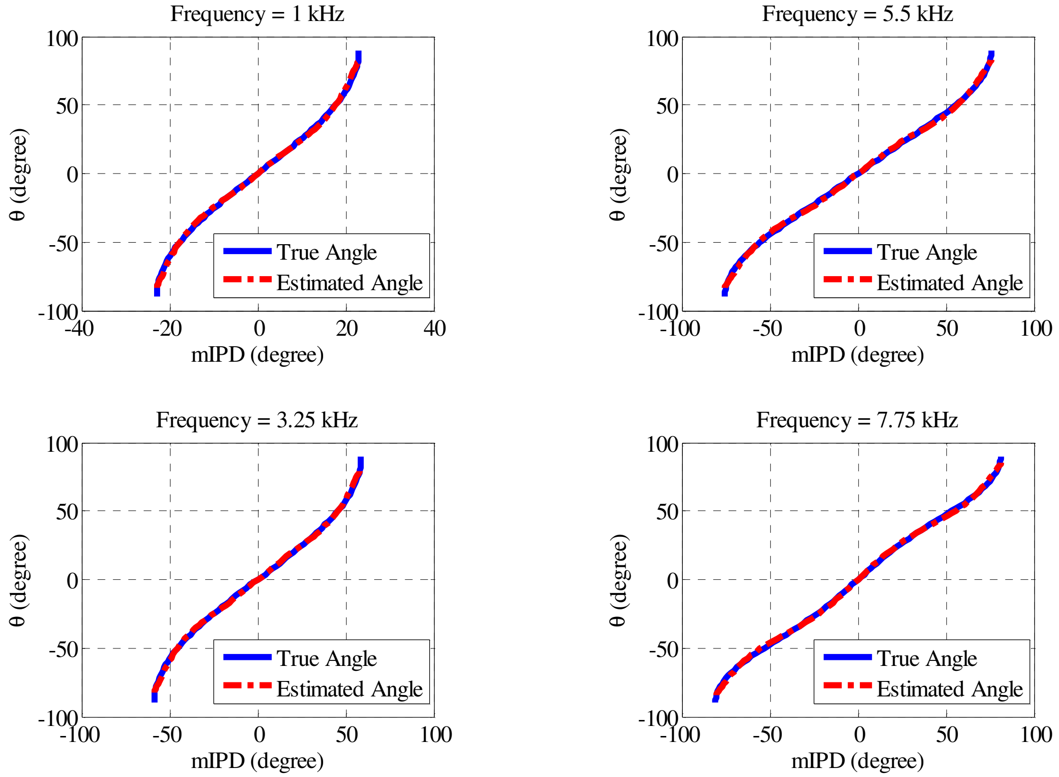


Figure 3.19: Fuzzy model testing results at different frequencies using the GP method.



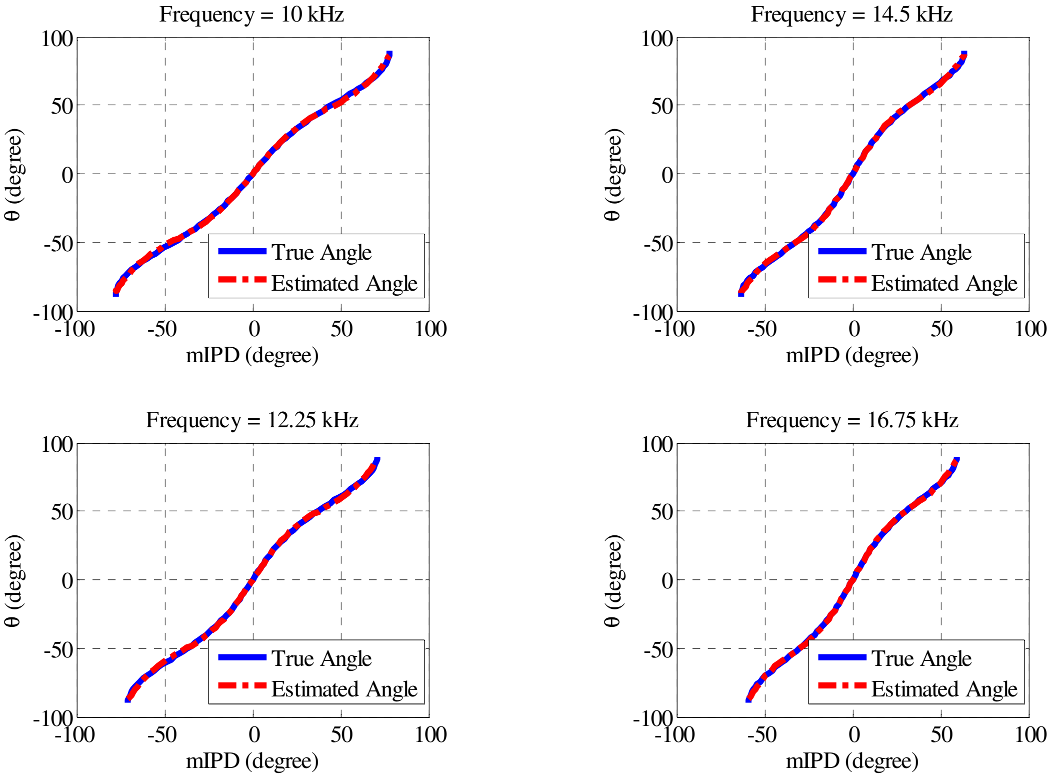


Figure 3.20: Fuzzy model testing results at different frequencies using the SC method.

3.2.2 Fuzzy Modeling of the Fly-Ear Inspired Sensor

A fuzzy model of the fly-ear inspired sensor can be constructed by using two different approaches. In the first approach, the theoretical model is used to generate theoretical input/output data set, which can be utilized to construct the fuzzy model. The fuzzy model will then be tested using a data set obtained from the experiment. The main drawback of this approach is that when the theoretical parameters used to generate the theoretical input/output data differ from the real experimental parameters, the model becomes not accurate in estimating the sound incident angle. In the second approach, an experimental data set is divided into training and checking data sets to construct and test the fuzzy model, respectively. However, the main challenge using this approach is that the training data should be accurate enough so that the obtained model can be used to estimate the incident angle accurately.

To construct the mapping (i.e. the fuzzy models) based on any of the two approaches, the training and checking data have two inputs, namely the normalized frequency and the normalized mIPD, and one output, namely the normalized azimuth angle. The frequency space is normalized by the maximum applied sound frequency (22.5 kHz). The azimuth angle space is normalized by the maximum estimated angle (90^o). The mIPD at each frequency is normalized by the maximum mIPD, which is obtained when the incident angle is 90^o at that frequency. Therefore, the normalization of the mIPD space is different with respect to frequency, which can help distinguish the sound incident angle at the different frequencies.

3.2.2.1 Experimental Setup

The experimental arrangement, and the fly-ear inspired sensor developed by another lab member, Haijun Liu, is used in the experimental studies, as shown in Figure 3.21. The fly-ear inspired sensor was used to collect the experimental data, which were utilized later to construct one of the fuzzy models of the sensor and to check the performance of the models. As shown in Figure 3.21, the sensor device is mounted on the tip of an aluminum rod with a diameter of 0.25", which is fixed on a motorized rotational stage (Newport, URS75BPP) connected to a motion controller (Newport, ESP 300). A pure tone or chirp sound of various frequencies is played through a speaker (ESS Heil air motion transformer). To simulate the movement of the source, another motorized rotational stage was placed under the stage mounting the sensor while the position of the speaker was fixed. The responses from the two diaphragms were acquired by using a fiber optic detection system for every 2.5 degrees in the range of -30 to 30 degrees (i.e. the linear range) and every 5 degrees outside this range. For each frequency, this process was repeated 5 times and the average value was obtained. Signals received from the photo detectors in the optical detection system were sampled at a rate of 500 kHz for each channel.

The time delay was calculated by finding the maximum cross-correlation of signals from the two channels and converted to phase difference mIPD. For the sampling rate used in the experiment (500 kHz), the time resolution of the cross-correlation is 2 μs (note that a center-to-center distance of 1.2 mm only renders a time difference of up to 3.53 μs). For a sound frequency of 8 kHz, a time delay of 2 μs is equivalent to a phase difference of 5.76^{o}. To improve the resolution, a second order polynomial is used to perform the curve fitting in the vicinity of the center peak of the cross-correlation [60].

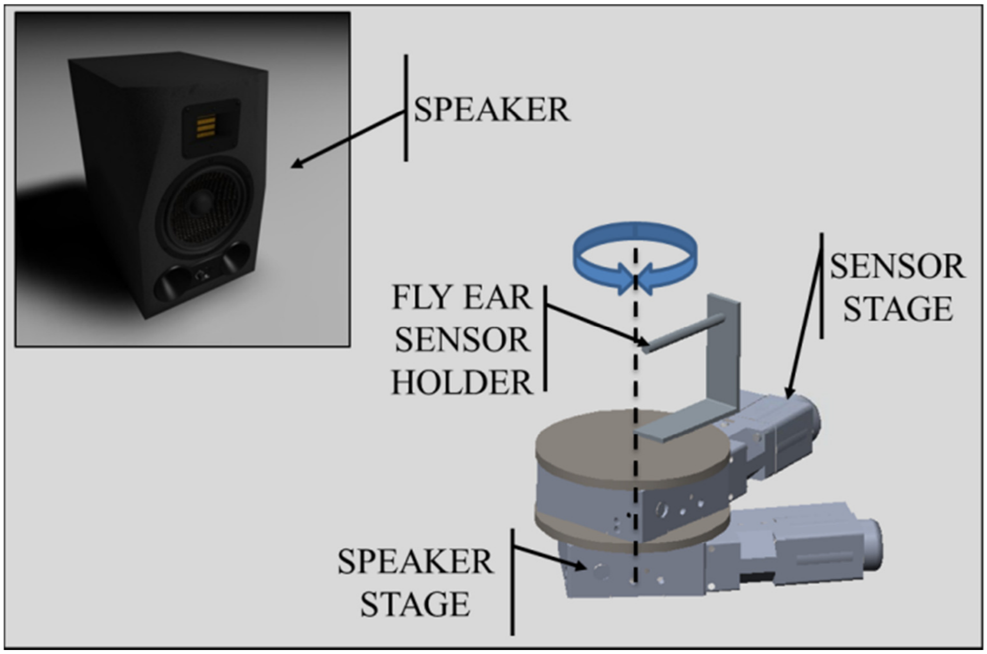


Figure 3.21: Experimental arrangement of the fly-ear inspired sensor mounted on two rotational stages for sound source localization.

3.2.2.2 Fuzzy Modeling Based on the Data Set Obtained with the 2-DOF Model

By using the 2-DOF model of the fly-ear inspired sensor (Eq. (2.62) and the parameters listed in Table 3-4), mIPDs at different incident angles θ and frequencies were calculated. The incident angle space was divided equally with 5^o increments within the region $|\theta|>30^o$ and 2.5^o increments within the region $|\theta| \leq 30^o$. On the other hand, the frequency space was evenly divided over the frequency ranges of 1 kHz to 22.5 kHz. The middle region from 5 kHz to 11 kHz was divided into 30 unevenly spaced segments 100 or 200 Hz increments. The constructed theoretical training data was used to get a Sugeno-type fuzzy model using ANFIS. The model was checked later by using the experimental data set obtained with the fly-ear inspired sensor. It should be mentioned that dividing the training data into two sets to get two separate fuzzy models can enhanced the incident angle estimation since the IPD changes the sign at the rocking mode natural frequency. The experimentally measured sensor parameters were used to obtain a theoretical model of the sensor, which are listed in Table 3-4. Figure 3.22 shows the theoretical mIPD of the sensor as a function of sound incident angle and frequency by using these parameters, which has a good agreement with the experimental results shown in Figure 3.27.

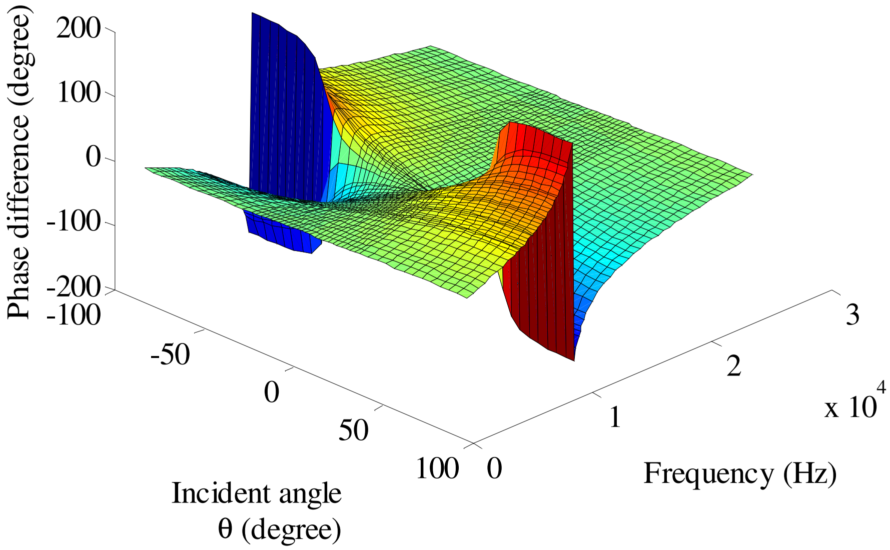


Figure 3.22: 3D surface of the mIPD at different incident angles θ and frequencies using the fly parameters shown in Table 3-4.

Table 3-4: Parameters for the fly-ear inspired sensor device.

Parameters	Values
Mass of bar *m*	2.88×10^{-10} kg
Rocking mode natural frequency	9750 Hz
Bending mode natural frequency	22970 Hz
Damping ratio 1	0.18
Damping ratio 2	0.05
Separation of force locations *d*	1.2×10^{-3} m
Tympanum area *s*	0.288×10^{-6} m^2
Sound speed *c*	344 m/s

Training with the theoretical data

The general structure of the fuzzy inference system for the fly-ear inspired sensor has two antecedents, i.e., the normalized frequency and the normalized mIPD, and one consequent, i.e., the normalized incident angle of the sound source as illustrated in Figure 3.23. The membership functions of the inputs for the two models in two different frequency regions are plotted in Figure 3.24 and Figure 3.25. In both models, 8 Gaussian membership functions for each input were used for the comparison purposes. The following first order Sugeno-type rule is used:

R_i: IF *nf* is NF_{1i} and *nmipd* is $NmIPD_{1i}$ THEN $n\theta_i = a_{0i} + a_{1i} \times nf + a_{2i} \times nmipd$

By using the subtractive clustering algorithm, the antecedent membership functions were identified with a radius of influence of 0.42 and the default values for the other factors (squash factor of 1.25, accept ratio of 0.5, and reject ratio of 0.15). The least square estimation was used to optimize the consequent parameters. Table 3-5 and Table 3-6 list the parameters of the antecedent Gaussian membership functions, shown in Figure 3.24 and Figure 3.25, which include the standard deviations σ_i the cluster centers c_i and the optimized consequent parameters (a_{0i} a_{1i} a_{2i}) after tuning by ANFIS. The weighted average defined by Eq. (3.15), of the outputs of the rules was calculated to obtain the model output.

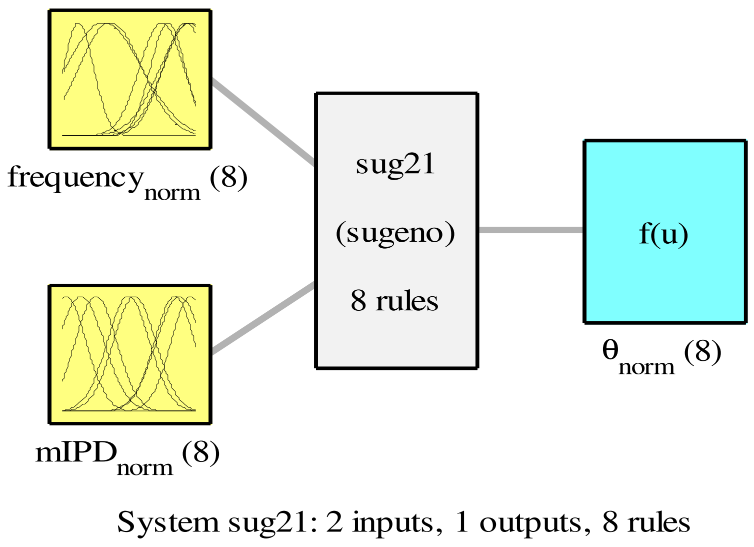


Figure 3.23: Schematic of fuzzy inference system structure for the fly-ear inspired sensor.

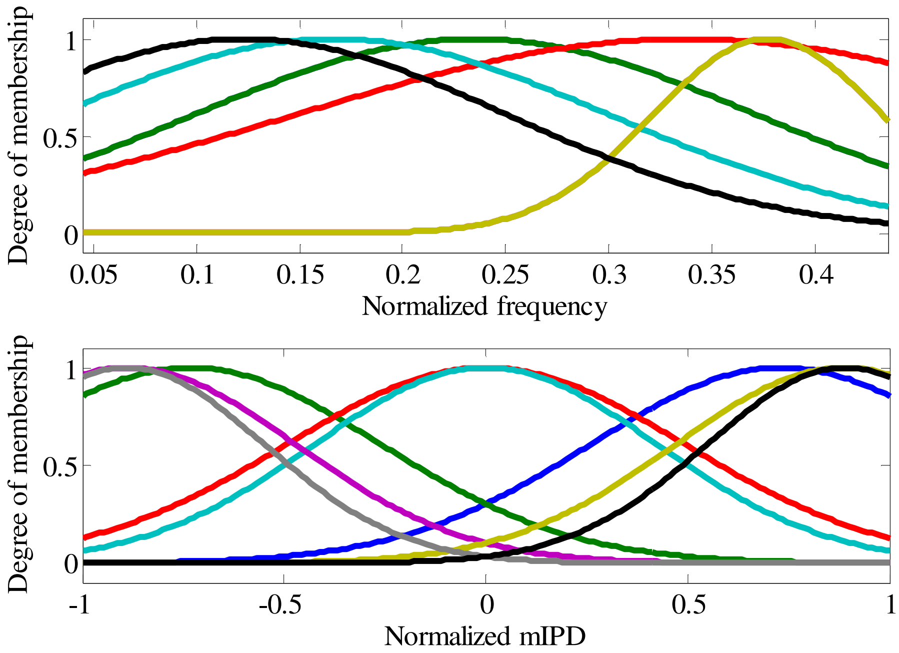


Figure 3.24: Antecedent membership functions for the 1st model based on theoretical results.

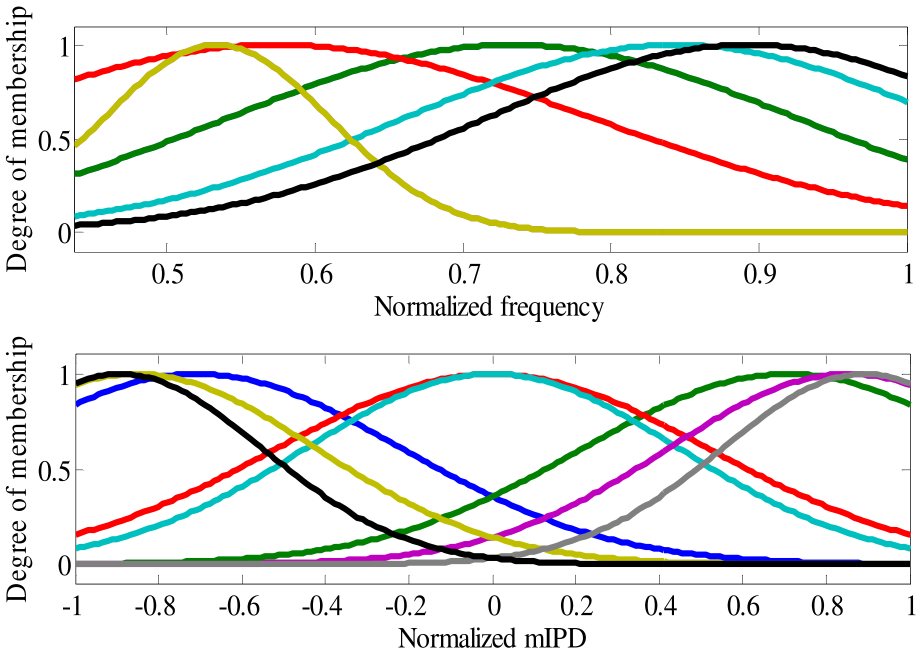


Figure 3.25: Antecedent membership functions for the 2nd model based on theoretical results.

Table 3-5: Antecedent and consequent parameters for the first model.

Rule #	Normalized Frequency $[\sigma_i\ c_i]$	Normalized mIPD $[\sigma_i\ c_i]$	Normalized θ $[a_{0i}\ a_{1i}\ a_{2i}]$
1	[0.137 0.235]	[0.475 0.733]	[-31.61 4.6527 8.168]
2	[0.137 0.235]	[0.475 -0.733]	[31.61 4.6527 -8.168]
3	[0.191 0.338]	[0.493 0.00]	[0.00 9.0295 0.00]
4	[0.134 0.167]	[0.423 0.00]	[0.00 -9.426 0.00]
5	[0.056 0.377]	[0.415 -0.886]	[1.902 2.6127 0.698]
6	[0.056 0.377]	[0.415 0.886]	[-1.902 2.6127 -0.698]
7	[0.128 0.123]	[0.342 -0.89]	[11.92 1.922 4.585]
8	[0.128 0.123]	[0.342 0.89]	[-11.92 1.922 -4.585]

Table 3-6: Antecedent and consequent parameters for the second model.

Rule #	Normalized Frequency $[\sigma_i\ c_i]$	Normalized mIPD $[\sigma_i\ c_i]$	Normalized θ $[a_{0i}\ a_{1i}\ a_{2i}]$
1	[0.194 0.733]	[0.493 -0.708]	[23.71 -5.159 -17.785]
2	[0.194 0.733]	[0.493 0.708]	[-23.71 -5.159 17.785]
3	[0.214 0.575]	[0.519 -0.00]	[0.00 -9.202 0.00]
4	[0.184 0.844]	[0.449 0.00]	[0.00 7.734 0.00]
5	[0.077 0.534]	[0.433 0.854]	[-1.02 -2.435 1.805]
6	[0.077 0.534]	[0.433 -0.854]	[1.02 -2.435 -1.805]
7	[0.178 0.893]	[0.341 0.892]	[-7.245 -1.129 11.15]
8	[0.178 0.893]	[0.341 -0.892]	[7.245 -1.129 -11.15]

<u>Validating the model using experimental data</u>

After the fuzzy models were constructed they were tested using experimental data set. In Figure 3.26, the estimated incident angle using the fuzzy logic models as a function of mIPD are compared with the theoretical and the experimental results at different frequencies. A complete list of results at all frequencies is provided in Appendix F. From these figures, the results obtained with the fuzzy models agree well with the theroretical data but poorly match with the experimental checking data. Figure 3.26(a) and (b) show the angle estimation at frequencies less than the rocking mode natural frequency and Figure 3.26(c) and (d) show the estimation for frequencies higher than the rocking mode natural frequency. Comparing the two regions, mIPD changes its sign at the rocking mode natural frequency, which is estimated to be 9.8 kHz, and the mIPD range increases as the stimulus frequency gets closer to the rocking mode natural frequency. The huge differnce in the mIPD range does not affect the capability of fuzzy model to estimate the incident angle due to the normalization of the data. The main advantage of this normalization in constructing the fuzzy model is the ability to use the sensor in a wider frequency range. In addition, note that the nonlinearity of the sensor increases with increasing the stimulus frequency. Even though the fuzzy model based on the theoretical data set can deal with the nonlinearity, the huge error can be observed between the estimated angle and the experimental data due to the accuracy of the constructed model. To improve the estimation accuracy, a fuzzy model based on experimental results is constructed, which will be discussed in the following subsection.

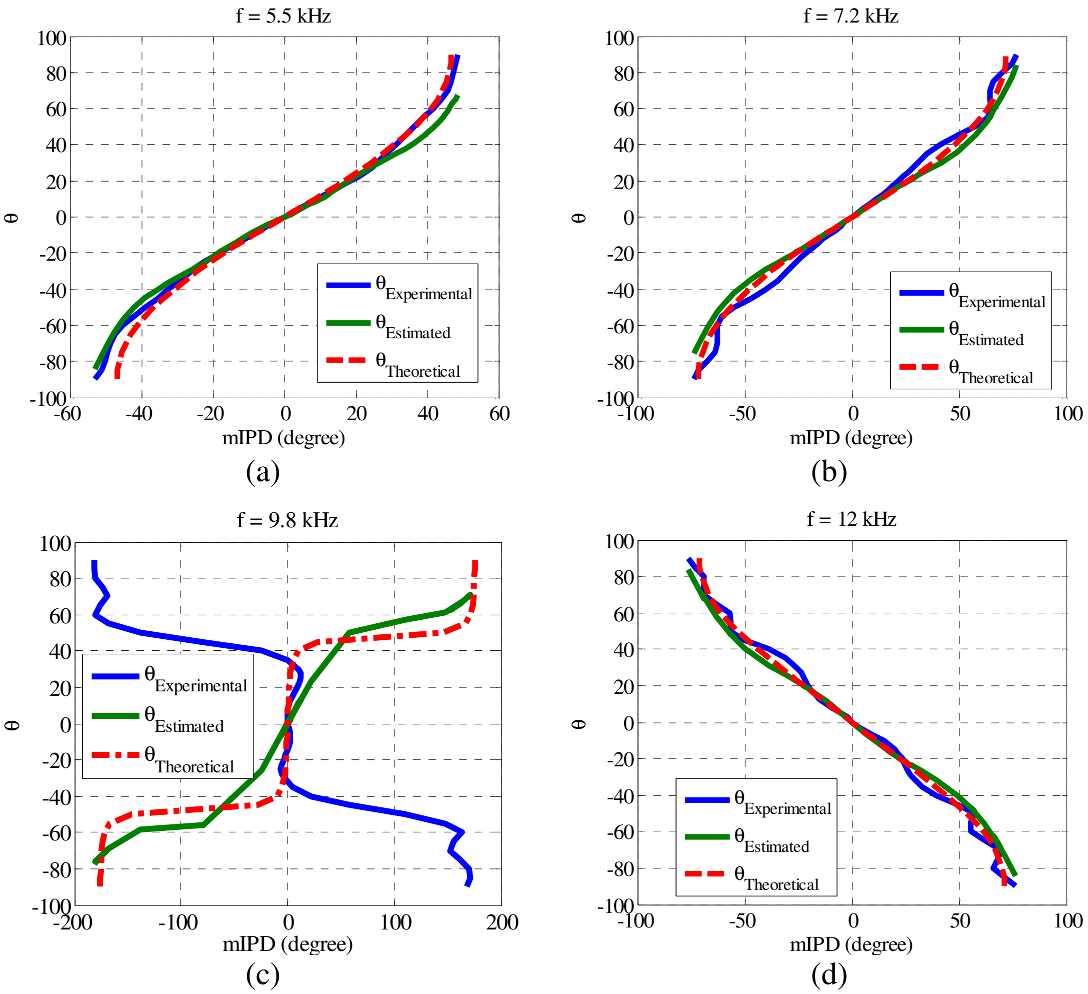


Figure 3.26: Estimated sound incident angle compared with the experimental and theoretical results θ at different frequencies (a) f = 5.5 kHz, (b) f = 7.2 kHz, (c) f = 9.8 kHz, and d) f = 12 kHz.

3.2.2.3 Fuzzy Model Based on Experimental Data Set

Here, the previously collected experimental data were used to construct and test the fuzzy models. The experimental results obtained from the sensor were divided into two equal sets, with the same space segmentation method explained in Section 3.2.2.2, for training and checking. It should be noted that the experimental map of the phase difference as a function of the incident angle and the stimulus frequency slightly differs from the map shown in Figure 3.10 even with using the calibrated sensor parameters listed in Table 3-4. Accordingly, new fuzzy models were created using the subtractive clustering method. Similarly, the mIPD values were normalized by the maximum value at each frequency so that the incident angle at the frequencies can be distinguished. Furthermore, two fuzzy models were created, one for frequencies lower than the rocking mode natural frequency and the other for the higher frequencies, which can help reduce the training error and make the offline training process faster, compared with constructing one model for the entire frequency range.

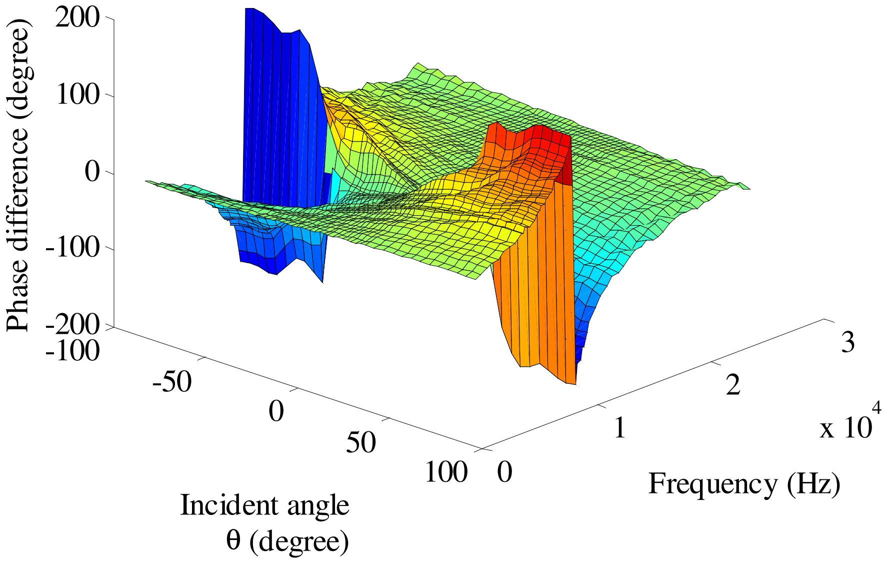


Figure 3.27: Experimental results of the mIPD at different incident angles θ and frequencies.

Training with the experimental data

Two fuzzy models were constructed using the training experimental data. As mentioned earlier, one model is for the frequencies smaller than the rocking mode frequency. The membership functions of the two inputs of the first and second models are plotted in Figure 3.28 and Figure 3.29, respectively. Table 3-7and Table 3-8 list the parameters of the antecedent Gaussian membership functions, shown in Figure 3.28 and Figure 3.29, which include the standard deviations σ_i, the cluster centers c_i , and the optimized consequent parameters (a_{0i} a_{1i} a_{2i}) after tuning by ANFIS. Similarly, the weighted average defined by Eq. (3.15), of the outputs of the rules was calculated to obtain the model output. Table 3-7 and Table 3-8 list the antecedents Gaussian membership function parameters, shown in Figure 3.28 and Figure 3.29, identified by the standard deviations σ_i, the cluster centers $c_{i,}$ and the optimized consequent parameters (a_{0i} a_{1i} a_{2i}) after tuning by ANFIS. Accordingly, the weighted average of the rule outputs is computed to get the model output.

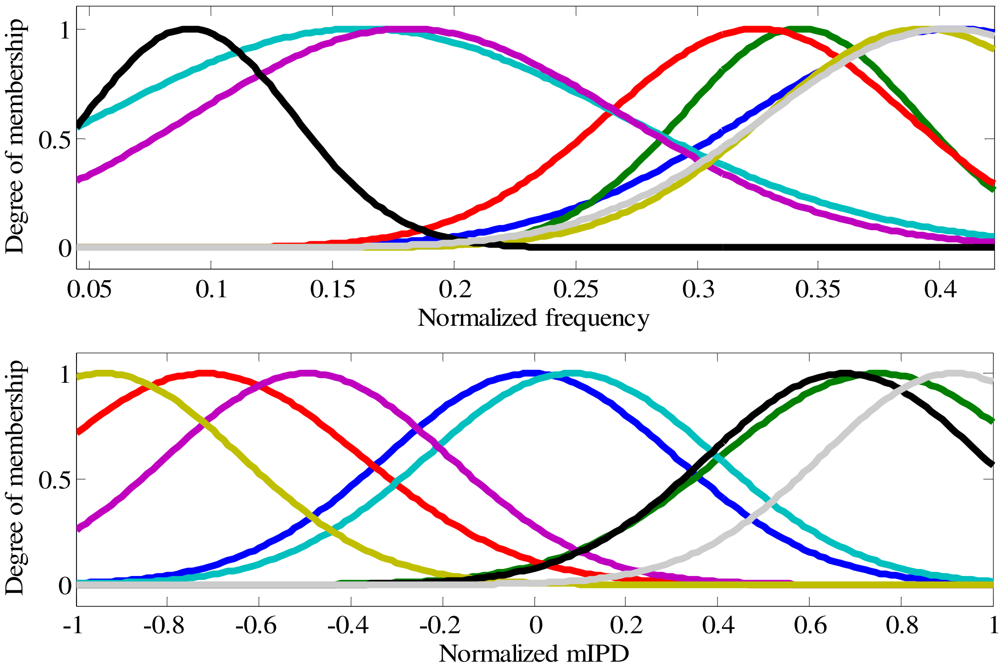


Figure 3.28: Antecedent membership functions for 1st model trained with experimental data.

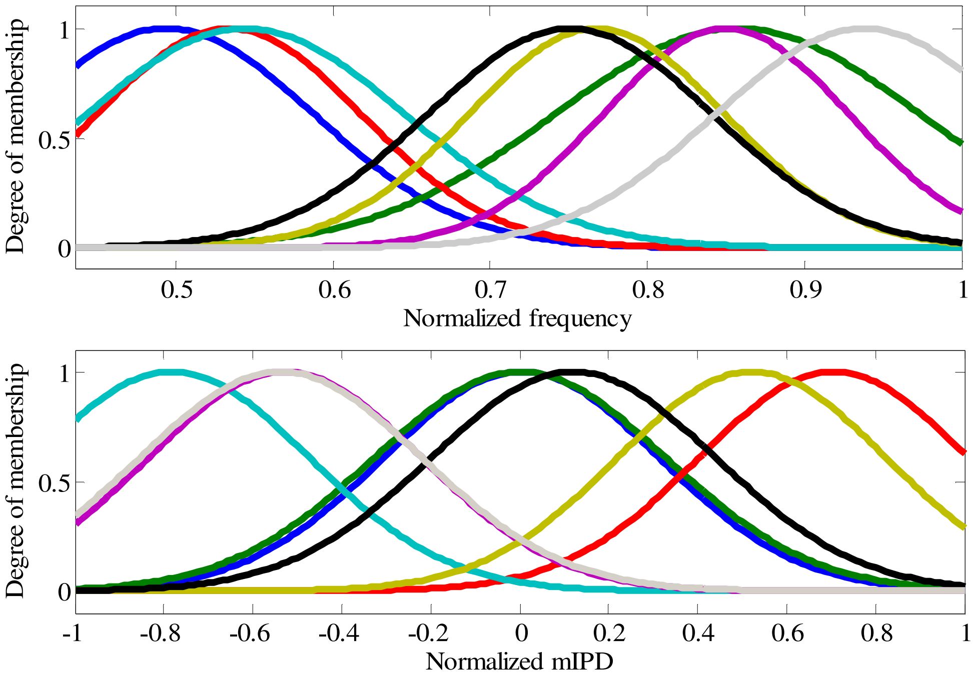


Figure 3.29: Antecedent membership functions for the 2nd model trained with experimental data.

Table 3-7: Antecedent and consequent parameters for the 1st model trained with experimental data.

Rule #	Normalized Frequency $[\sigma_i \; c_i]$	Normalized mIPD $[\sigma_i \; c_i]$	Normalized θ $[a_{0i} \; a_{1i} \; a_{2i}]$
1	[0.0847 0.4062]	[0.3159 -0.0090]	[-0.593 2.296 0.248]
2	[0.0487 0.3426]	[0.3434 0.7508]	[-1.253 1.789 -0.271]
3	[0.0618 0.3250]	[0.3432 -0.7190]	[1.61 1.075 0.041]
4	[0.1067 0.1618]	[0.3168 0.0817]	[-0.645 0.862 0.027]
5	[0.0884 0.1809]	[0.3079 -0.4941]	[-0.296 1.174 -2.762]
6	[0.0642 0.3931]	[0.3006 -0.9395]	[6.575 0.96 -2.762]
7	[0.0427 0.0912]	[0.3003 0.6779]	[0.673 1.301 -0.330]
8	[0.0736 0.4028]	[0.2911 0.9163]	[1.197 1.346 -0.933]

Table 3-8: Antecedent and consequent parameters for the 2nd model trained with experimental data.

Rule #	Normalized Frequency [σ_i c_i]	Normalized mIPD [σ_i c_i]	Normalized θ [a_{0i} a_{1i} a_{2i}]
1	[0.094 0.494]	[0.310 0.004]	[0.960 -1.565 -0.430]
2	[0.117 0.857]	[0.328 0.001]	[15.387 -0.3197 -15.333]
3	[0.0845 0.534]	[0.303 0.706]	[0.0542 -1.47 0.423]
4	[0.102 0.544]	[0.309 -0.781]	[0.193 -1.316 -0.503]
5	[0.079 0.850]	[0.308 -0.527]	[-12.250 1.083 6.768]
6	[0.082 0.768]	[0.303 0.522]	[-0.987 -0.1451 -0.0935]
7	[0.091 0.751]	[0.320 0.121]	[5.1547 -2.294 -1.204]
8	[0.095 0.938]	[0.318 -0.537]	[-51.778 -0.425 52. 49]

Figure 3.30 shows the training error history obtained with different number of the input membership functions used in the fuzzy model. When the number of membership functions is larger than 8 the error saturates at a value of 0.042 to 0.044 at about 50 epochs. This means that increasing the number of membership functions will not be useful for further improving the accuracy of azimuth angle estimation. To reduce the computation cost the smallest number of membership functions should be used (i.e., 8 membership functions) especially when all calculations are executed on a robotic platform in future.

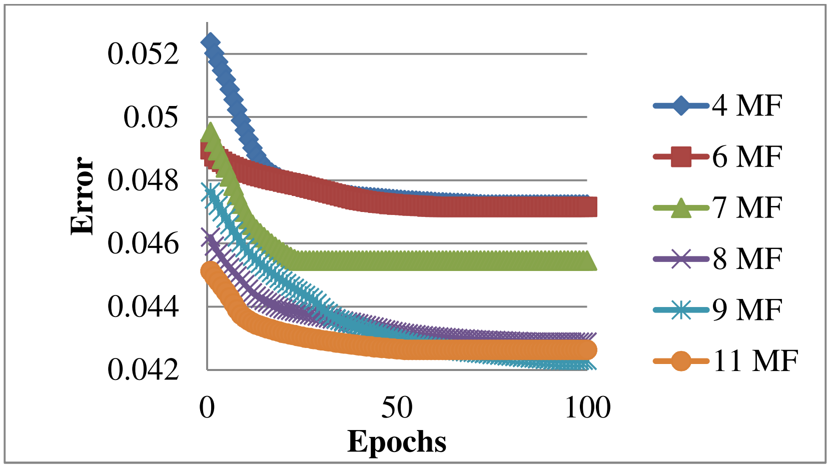


Figure 3.30: Training error history obtained with different number of inputs membership functions.

Validating the models with experimental data

After the fuzzy models were constructed, they were tested using the checking experimental data set. The estimated incident angle obtained with the fuzzy logic models at different frequencies are compared with the experimental checking data, as shown in Figure 3.31. It can be seen that the results obtained with the fuzzy model have a good agreement with the experimental checking data, when the sound frequency is not close to the rocking mode natural frequency. Figure 3.31(a) and (b) show the angle estimation for frequencies less than the rocking mode natural frequency which is found to be at 9.75 kHz, and Figure 3.31(c) and (d) show the estimation for frequencies higher than the rocking mode natural frequency. It can be seen that the results obtained with the fuzy model have a good agreement with the experimental checking data, when the sound frequency is not close to the rocking mode natural frequency. However, due to the abrupt change in mIPD at the rocking mode natural frequency, when the sound stimulus frequency is close to this frequency, a relatively large discrepancy can be found between the model estimation and the experimental data. A complete list of figures for incident angle estimation at all checking frequencies are provided in Appendix F.

The root mean square error (RMSE) in the estimated angles obtained with the fuzzy model constructed using the theoretical data set and that constructed using the experimental data set are compared in Figure 3.32. It can be observed that under the same conditions (same frequency, same number of fuzzy rules, same number of training/checking data points) the fuzzy model constructed from the experimental data is more accurate. This is due to the fact that the simple spring-mass-dashpot system used to model the fly-ear inspired sensor is not accurate enough. Another reason can be the estimation accuracy of the sensor parameters used in the 2-DOF model to map the sound incident angle to mIPD.

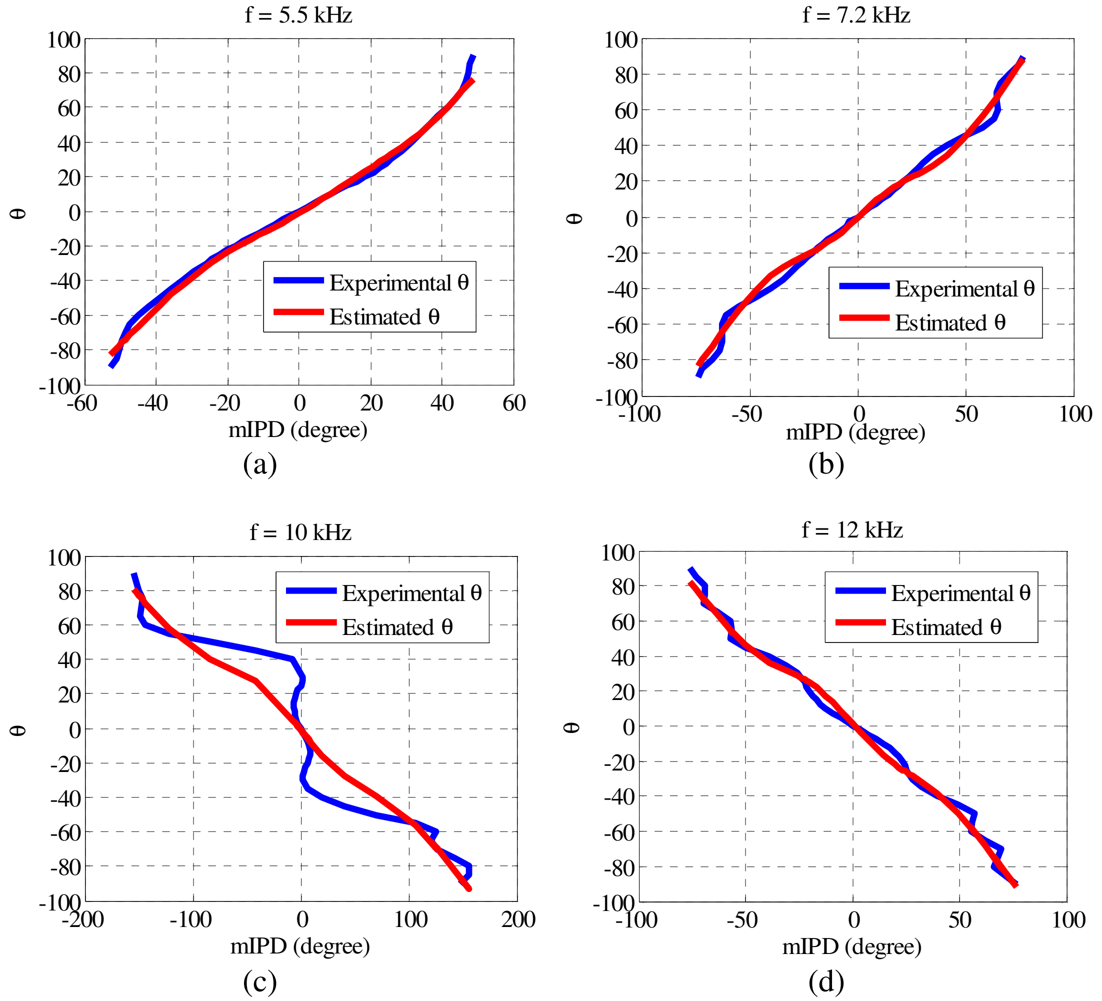


Figure 3.31: Estimated incident angle θ obtained using the model trained with an experimental data set compared with the experimental data at (a) f = 5.5 kHz (b) f = 7.2 kHz (c) f = 10 kHz, and d) f = 12 kHz.

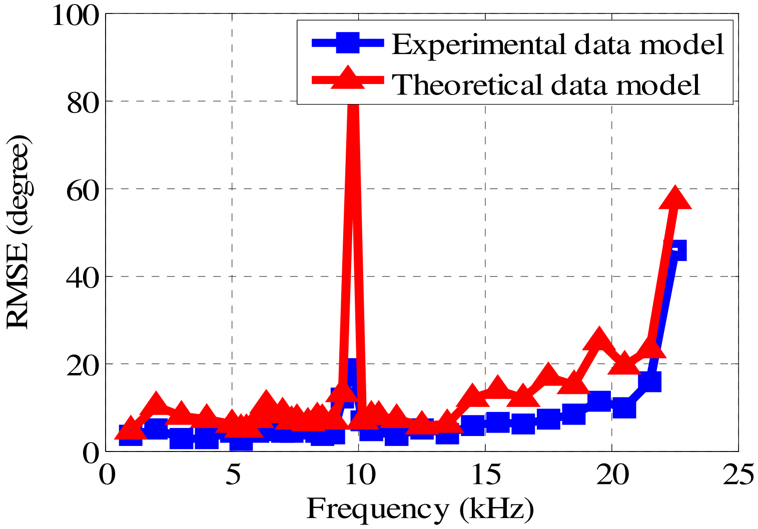


Figure 3.32: RMSE versus frequency for the two fuzzy models.

3.2.2.4 Experimental Validation

In this subsection, a reduced frequency range fuzzy model is constructed by following the procedure explained in the previous section using the experimental data for stimulus frequency range from 1 kHz up to 9 kHz. This model is then used in the experiment to localize a stationary and track a moving sound source, respectively. The experimental arrangement shown in Figure 3.21 was used. The localization performance with the fuzzy model is compared with that obtained with two conventional methods, namely, the least square error (LSE) method and the saturation function method. To compare the performance, the following four performance metrics are defined: the settling time, the over shoot, the steady state error, and the error band. The definitions of these metrics are listed in Table 3-9. Figure 3.33 illustrates the performance metrics by using a sample time history of localizing a stationary sound source.

Table 3-9 : Performance metrics definitions.

Name	Units	Definition
Settling time	[s]	The time required for the estimated sound incident angle to reach and remain within an error band regardless the initial sensor orientation with respect to the sound source.
Over shoot	[deg]	The maximum difference between the initial estimated sound source incident angle and the actual incident angle.
Steady state error	[deg]	The difference between the average of the estimated angles during the period after the settling time and the true value (zero in this case).
Error band	[deg]	The maximum range of error of the estimated angle after the settling time.
Root mean square error	[deg]	A measure of the differences between the estimated incident angle and the actual incident angle calculated over the localization trip time.

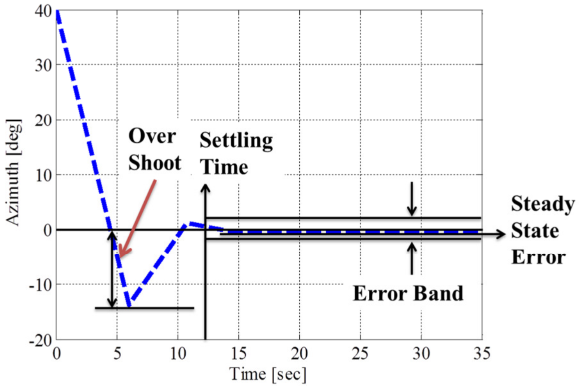


Figure 3.33: Illustration of performance metrics.

The importance of these performance metrics comes from the fact that one can use them to evaluate the performance of different methods and possibly choose a method with a better performance. Ideally, method that shows fastest convergence (i.e., shortest settling time) to a desired value with the smallest steady state error should be chosen. However, as will be shown later, a tradeoff is often found between the settling time and the error.

In the experiments, localization of a stationary source was first carried out. The sensor was initially oriented at different at an angle with respect to the source. The angle was changed from -180^o to 170^o with an increment of 10^o, including the case when the sound source was from the back. Without loss of generality, the sound frequency was chosen at 6.25 kHz for the experiment. The sound incident angle was estimated using the fuzzy model, the LSE method and saturation function at each angular position of the sensor. For the LSE, a predefined array of experimental phase difference results as a function of incident angle and sound frequency was used as a map for searching the incident angle at a specific frequency. If the exact angle cannot be found from the map, linear interpolation was used to estimate the angle using the data for the frequencies directly higher or lower than the frequency of interest. For the saturation function, the mIPD was assumed to be linearly proportional to the estimated angle within the range of -30^o to 30^o. The slope of the linear region was calculated at the beginning of the experiment by calibrating the sensor at the chosen sound frequency. For the sound localization using all methods, a LabView code was developed to collect the sensor signals and calculate the mIPD, using the cross-correlation method explained in Chapter 1. These measurements were averaged 15 times, and the averaged results were used for estimation of the sound incident angles according to the three different methods. The estimated angle was then sent by the LabView code through the serial port (USB) to control the rotational stages.

The localization histories obtained by using the different methods with initial sound positions of -20^o and 150^o are shown in Figure 3.34. As seen in Figure 3.34(a), the LSE and fuzzy logic methods exhibits a smaller overshoot in the incident angle estimation compared with the saturation function method. For a sound coming from the back, the saturation function method requires more time to pinpoint the sound compared with the other two methods as shown in Figure 3.34(b). This is due to the fixed moved steps used in the saturation function method when the incident angle is outside the linear region.

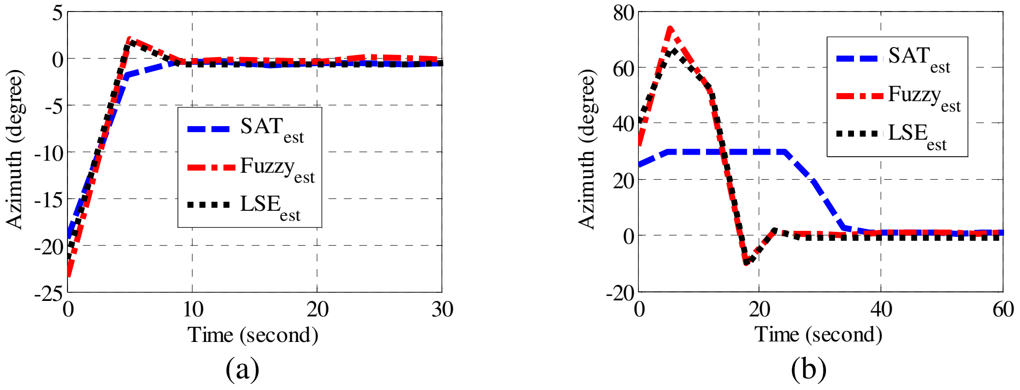


Figure 3.34: Sound localization histories obtained with different methods for initial sound incident angles of (a) -20° and (b) 150°.

The performance metrics obtained with the three methods used to localize a stationary source are compared in Figure 3.35. As can be seen from Figure 3.35(a), in terms of the settling time the fuzzy model performs slightly better than the LSE method, both of which exhibit much better performance than the saturation function method. The saturation function method requires more time to settle beyond the linear range since the sensor stage will rotate with a fixed step of 20° until it reaches the linear region. When the source is from the back of the sensor, it takes even longer to achieve convergence. For the LSE method, the settling time is almost the same as that for the fuzzy model. However, at some angles, the settling time shows some spikes which can be attributed to the measurement uncertainties that in turn will affect the linear interpolation in the LSE method. However, using the fuzzy logic method can eliminate these spikes since the uncertainties can be addressed in the fuzzy models. The reason for the long settling time for all the methods is due to the following: i) the processing time of the sensor signals to get 15 phase difference measurements, ii) the slow data transfer between the DAQ and the PC, and iii) the limitation of sensor stage movement speed (30 deg/sec) until it reaches the estimated position of the sound source.

The steady state error obtained from the three methods are compared in Figure 3.35(b). All of them have a steady state error within the range of -3° to 4°, where the LSE method shows better performance than the other two methods.

Figure 3.35(c) shows the error band for the three methods which can give an indication on the level of fluctuation about the steady state for each method. All the methods exhibit considerable level of fluctuation about the steady state. However, the LSE method exhibits better performance than the other two methods with the smallest range of fluctuations up to 2°.

Figure 3.35(d) shows the overshoot for the three methods. Again, the saturation function shows higher overshoot than the other two methods outside the linear region. However, overshoot for all methods starts to increase dramatically as the incident angle increases, especially when the sound is coming from the back due to the effort needed to solve the ambiguity problem. On the other hand, this figure also shows that the LSE method suffers from the overshoot spikes, which can be explained with the same reasons as those settling time spikes.

Figure 3.35(e) shows the root mean square error for the three methods, which were calculated over the entire localization process. The saturation function is shown to perform poorly again when the source is outside the linear region for the same reasons mentioned previously, while the other two methods show better performance in localizing the sound source.

Based on these results, for localization of a stationary source can be concluded that fuzzy logic method can overcome the linear range limitation the saturation function method and the settling time and overshoot spikes that LSE suffer from. However, the fuzzy logic method exhibits more steady state error compared to the LSE method.

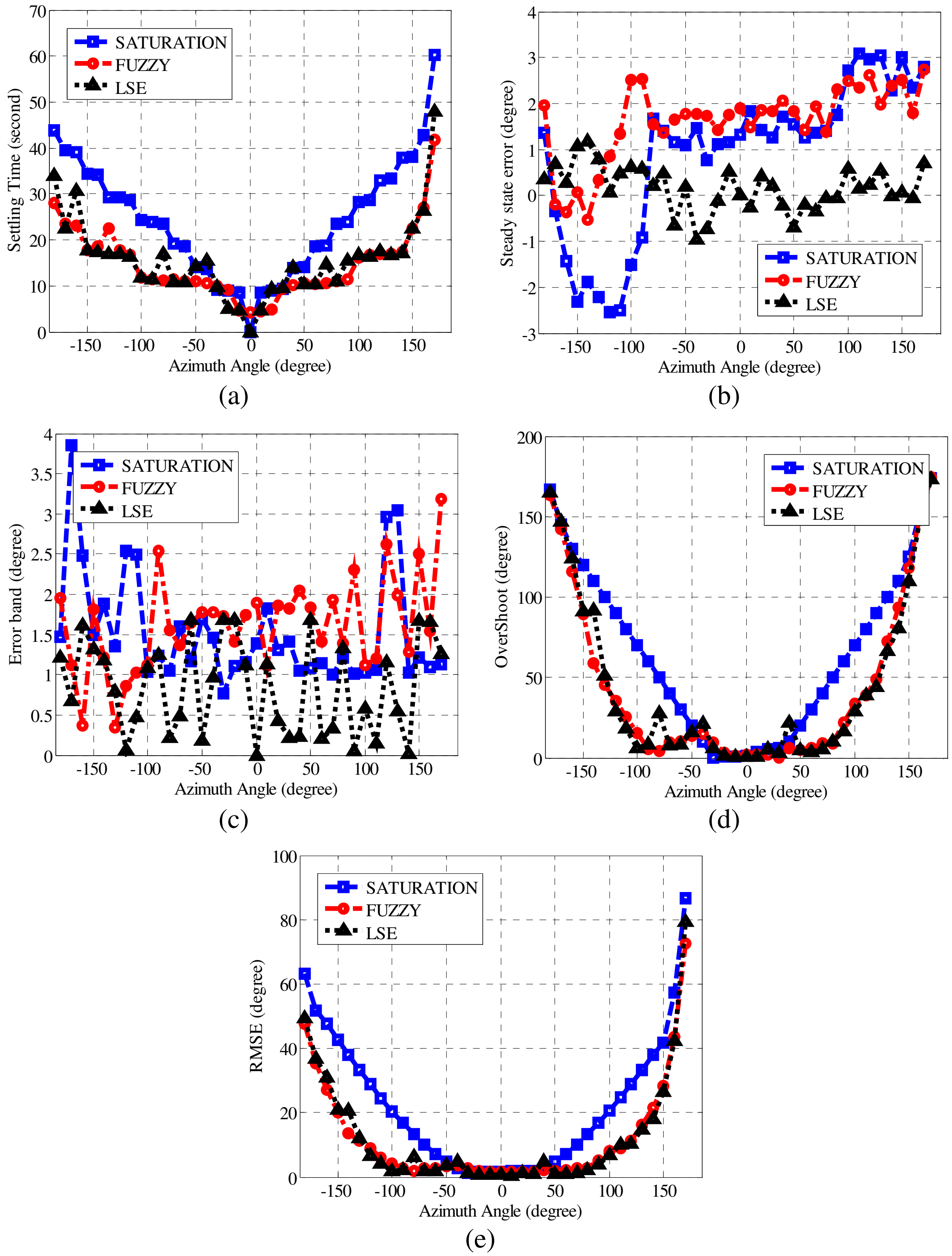


Figure 3.35: Sound localization performance metrics obtained for different methods: (a) settling time, (b) steady state error, (c) error band, (d) overshoot, and (e) root mean square error.

Further, tracking of a moving sound source was carried out. The tracking histories using the different methods at specific speed ratios (ratio of the speed of the sensor stage and that of the sound source stage) of 1 and 4 are shown in Figure 3.36. It is difficult to compare between the performance of different methods for tracking a moving source, since the tracking histories, especially at low speed ratios, do not look similar to those obtained for localization of a stationary source, as shown in Figure 3.36(a). Therefore, to compare the performance of the three methods, different performance metrics were used. The absolute positions of the sensor (θ_{sensor}) and sound source (θ_{sound}) stages were acquired throughout the tracking process, which was used to obtain the absolute RMSE, which is defined as the difference between the sound stage position (θ_{sound}) and the sensor stage position (θ_{sensor}). The relative RMSE was also determined, which was defined as the difference between the estimated incident angle (θ_{est}) and the actual sound incident angle (θ_{actual}). In this experiment, a sound from the back was not tested to protect the sensor since the optical fibers used to detect the sensor signals may be broken especially at a low speed ratio.

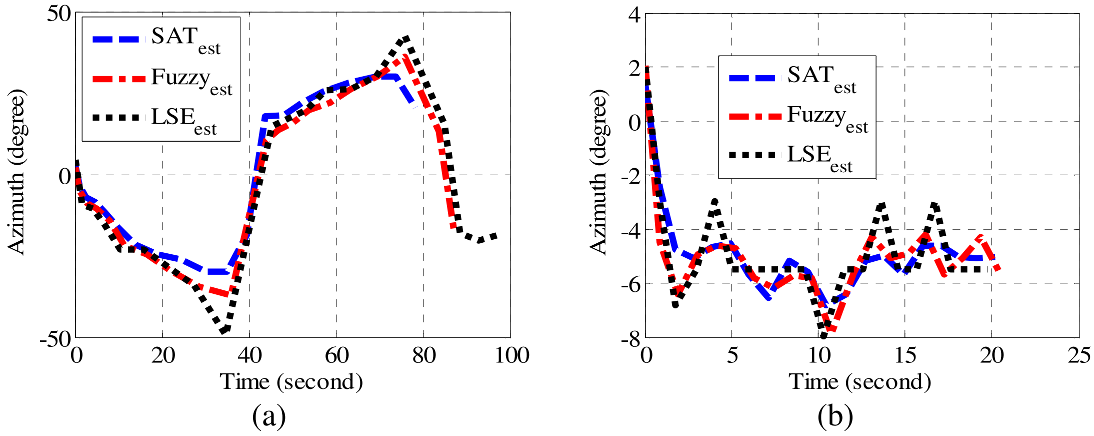


Figure 3.36: Sound tracking histories obtained with different methods at (a) speed ratio of 1 and (b) speed ratio of 4.

Figure 3.37 shows the relative and absolute RMSE for the three methods as a function of the speed ratio (relative angular speed between the sensor and source stages). In both cases, the fuzzy logic method is shown to have better performance than the other two methods with less RMSE errors. To get a relative RMSE of less than 2^o (the localization accuracy of the fly ear), the speed ratio should be at least 2 for the fuzzy logic, 4 for the LSE, and 5 for the saturation function. Due to the delays that can be attributed to the signals processing, angle estimation, and stage motion, a bias error that decreases with increasing the speed ratio is present as shown in Figure 3.37(b).

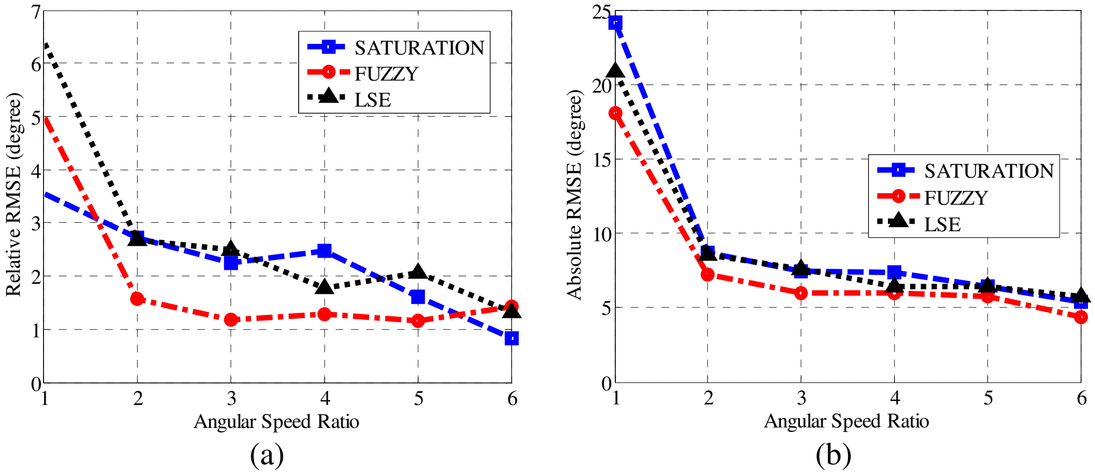


Figure 3.37: (a) Relative RMSE and (b) absolute RMSE versus angular speed ratio.

3.3 Summary

In this chapter, two methods are investigated and used for sound source localization with the fly-ear inspired sensor. The first method is based on the model-free gradient descent method where the sensor output is used directly to evaluate an objective function. Different objective functions are used to develop difference localization schemes. The performance of these schemes is evaluated via simulations.

Furthermore, this method is shown to have the capability of solving the ambiguity problem. However, the main drawback of this method is the iterative nature, which makes the localization of a moving sound source challenging.

A non-iterative method for sound localization with the fly-ear inspired sensor is also developed, namely the fuzzy logic method. The sensor response at different incident angles and stimulus frequencies is modeled using fuzzy logic. The first attempt is to select an appropriate method to build the fuzzy model based on training set generated from the 2DOF model of the fly-ear sensor. Two methods, namely, the grid partitioning and the subtractive clustering method, are investigated. The subtractive clustering is found to be the better choice due to its clustering nature that require less number of rules which leads to faster execution time especially when this system is implemented in a mobile platform that has limited processing capabilities. Furthermore, a previously collected experimental data set is used to conduct the fuzzy model. The experimental data set is divided into two groups, the training data set and the checking data set. As indicated by the names, the training data set is used to train the fuzzy model and the checking data is used to validate the constructed fuzzy model. It is found that the fuzzy model works properly in estimating the incident angle at different frequencies. The challenge of finding the maximum phase difference at the specific sound source frequency, which is needed for normalization of the fuzzy model input, is addressed by using a lookup table and linear interpolation.

Finally, the fly-ear inspired sensor with a reduced fuzzy model is used in experimental studies to localize and track a stationary and moving sound source. The performance of the system with the fuzzy model is compared to that using the least square error method and saturation function method. For sound source localization, it is found that the fuzzy model overcome the linear range limitation in localization using the saturation function method and the spikes that the LSE method suffers from. However, the fuzzy model method is found to have a larger steady state error than that of the LSE method.

The fuzzy modeling of the fly-ear sensor will open the door for localization of stationary and moving sound sources by using a mobile robot equipped with the fly-ear sensor.

Chapter 4 Mobile Robot Control for Localization and Tracking

In this chapter, nonlinear and quadratic-linear controllers are designed to control a differentially driven mobile platform, equipped with a pair of acoustic sensors for localization of a single stationary sound source and tracking a single moving sound source without any knowledge about the distance between the robot and the sound source or the trajectory of the moving sound source. Experimental implementation of this platform will be discussed in Chapter 5. As shown in Figure 4.1, the control loop has a plant (a differentially driven 2-wheeled robot kinematics in this case) to be controlled, a sensor with its own circuitry and the algorithms that calculate the time/phase difference between the acquired signals from the two microphones, an angle estimation block that estimates the incident angle from the sensor information by using any of the methods developed earlier in Chapter 3, and a controller. Here, different controllers are designed to drive the robot toward the sound source by using the acoustic sensor information. The well-known "*cross correlation*" method is applied to calculate the time/phase difference.

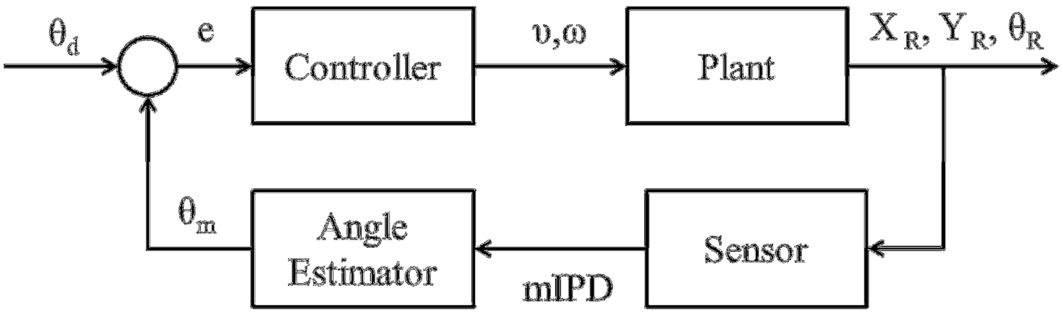


Figure 4.1: Mobile robot control loop; θ_d is the desired angle, θ_m is the measured error angle, e is the error signal (θ_d-θ_m), υ is the robot translational velocity, ω is the robot angular velocity, X_R, Y_R, θ_R are the robot position and heading angle.

In this work, since the goal is to track a moving sound source by only using the angle error information, the sound tracking problem is considered as a path tracking problem. In the literature, the tracking problem is considered as a highly nonlinear problem [61]. Many approaches have been developed to solve the general path tracking problem or the sound tracking problem by directly controlling the dynamics and kinematics of the tracking mobile robotic platform. For example, Zhang *et al.* developed a Lyapunov approach based nonlinear controller to steer vehicles automatically to track a curved path [61]. In this work, the small angle assumption was not used and the model nonlinear terms were not ignored. Similarly, following the same approach DeSantis *et al.* experimentally studied the controller performance compared with a fuzzy controller developed in the literature [63]. Furthermore, Zhang *et al.* developed a sliding mode nonlinear controller [64], [65]. In addition, Shaout *et al.* developed a fuzzy logic robust controller based on Sugeno fuzzy model, in which the tracking performance was improved even with the existence of noise and the vehicle parameter variations [66]. Han *et al.* developed a Mamdani type fuzzy controller to track a moving object based on the sound signal detected using a three-microphone array [36]. Triangulation was employed to calculate the distance and orientation of the object based on the time difference of arrival measurements between the microphone pairs. Gholipour and Yazdanpanah designed a tracking controller for non-holonomic mobile robot [67]. The Lyapunov based controller was designed in two steps. First, the kinematic stabilization was achieved by using nonlinear control laws. Second, translational and angular velocities were exponentially stabilized by controlling the acceleration rate. Bui *et al.* presented a simple and robust Lyapunov based nonlinear controller to track a reference welding path and velocity of a differential driven mobile robot [68]. Song *et al.* designed a PID controller for odor/sound tracking in a multi-robot system [19]. Olfactory robot used sensor fusion (gas and air flow sensors) to get information about the odor distribution in space and two hearing robots were used to estimate the time difference with a microphone pair to geometrically localize the sound source. Normey-Rico *et al.* developed a PID controller for the path tracking problem [69]. The mobile robot model is normalized to tune the PID gains easily keeping a robust and nominal performance of the system. Experiments demonstrated the controller performance and robustness.

Generally speaking, controllers that are used to control a robot dynamics by directly controlling the motor torques and traction forces that are not physically accessible, cannot be implemented regardless how good their performance is [61]. Therefore, in this dissertation work, different controllers are proposed, which can be used to control the error of the robot heading angle while the robot is moving by using the robot kinematics. In a robotic platform equipped with the fly-ear inspired sensor, the output from the fuzzy model will be used as the input of the control laws. However, here for proof of concept, the geometrical model of a microphone pair is used to estimate the sound incident angle. The control laws are derived using the Lyapunov approach for localization of a stationary sound source. The results are compared with the conventional control of the robot kinematics using a PID controller.

4.1 Robot Kinematics for Localization of a Stationary Sound Source

The objective of this section is to derive the robot kinematics for localization of a stationary sound source, which will be used later to derive the control laws using the Lyapunov approach. The problem of interest is basically a robot equipped with acoustic sensor, which is used to pinpoint the sound source by using the directional cue information obtained from the sensors. The distance between the sound source and the robot is unknown and cannot be estimated from the sensor measurement. In addition, there is no predefined path that the robot should follow to reach the source. Note that the robot position and the sound source position can be connected by a virtual line regardless the original robot (tracker) heading direction, as shown in Figure 4.2. If the measured robot heading angle relative to this line is zero, the robot should move along this line. Note that for a stationary sound source, while the tracking robot is moving, the measured sound incident angle not only changes due to not only the changed tracker heading angle but also the planar motion of the tracker. The problem here is therefore to control the robot kinematics according to the measured angle. Since the only obtainable information is the angle from the sensor, the decision making will only depends on the sensor measurements to control the robot angular and translational velocities needed to approach the sound source. Figure 4.2 shows the schematic of the robot and the stationary sound source. The non-holonomic constraints of a differentially driven robot are taken into consideration.

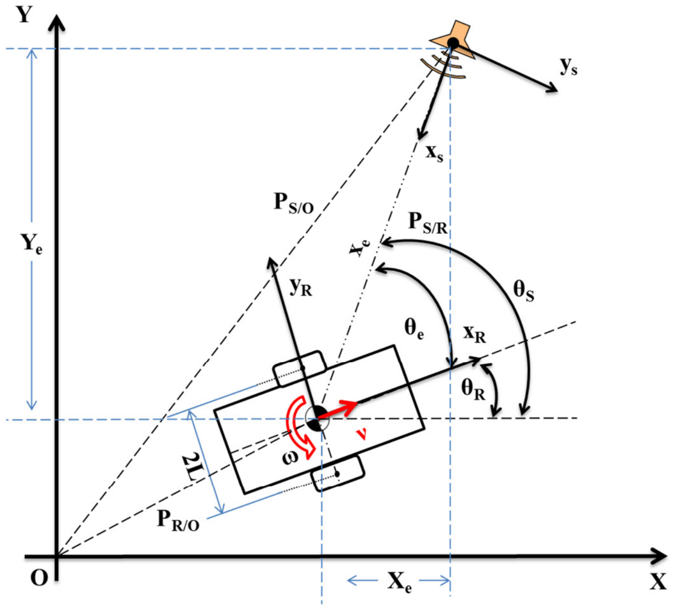


Figure 4.2: Schematic of the robot and sound source positions. θ_e is the heading error measurement, θ_R is the robot heading angle, $P_{S/O}$ is the position vector of the sound source relative to the origin, $P_{R/O}$ is the position vector of the robot relative to the origin, and $P_{S/R}$ is the position vector of the sound source relative to the robot, ω and υ are the robot angular and translational velocities, 2L is the width of the robot, and X_e and Y_e are the X and Y axis posture errors of the sound source relative to the robot in the global coordinates.

For the posture shown in Figure 4.2, the posture error can be expressed in the Polar coordinates as:

$$P_e = \begin{bmatrix} x_e \\ \theta_e \end{bmatrix}, \tag{4.1}$$

and

$$\begin{gathered} \theta_m = \theta_e + \eta \\ -\frac{\pi}{2} \le \theta_e \le \frac{\pi}{2} \ , \end{gathered} \tag{4.2}$$

where η is an additive measurement noise.

Accordingly, the posture velocity of the same configuration can be derived from the posture Eq. (4.1) as:

$$\dot{P}_e = \begin{bmatrix} \dot{x}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} -v_R \cdot \cos(\theta_e) \\ \omega_e \end{bmatrix}. \tag{4.3}$$

The main problem of using the polar coordinate is that the trajectory that the robot will follow is not clear, according to the control laws that will be derived later. The posture error can also be expressed in the Cartesian coordinates, as

$$P_e = \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = P_{S/O} - P_{R/O} = \begin{bmatrix} x_s \\ y_s \\ \theta_s \end{bmatrix} - \begin{bmatrix} x_R \\ y_R \\ \theta_R \end{bmatrix}. \tag{4.4}$$

The corresponding posture velocity error can be derived from Eq.(4.4) as

$$\dot{P}_e = \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} \dot{x}_s - \dot{x}_R \\ \dot{y}_s - \dot{y}_R \\ \dot{\theta}_s - \dot{\theta}_R \end{bmatrix}. \tag{4.5}$$

For simulation purposes, in addition to the stationary sound source assumption, it is also assumed that the sound source is a point source. Eq. (4.5) can thus be re-written as

$$\begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} -v_R \cdot \cos(\theta_R) \\ -v_R \cdot \sin(\theta_R) \\ \dot{\theta}_R - \dot{\theta}_S \end{bmatrix} = \begin{bmatrix} -v_R \cdot \cos(\theta_R) \\ -v_R \cdot \sin(\theta_R) \\ \omega_R - \omega_S \end{bmatrix} = \begin{bmatrix} -v_R \cdot \cos(\theta_R) \\ -v_R \cdot \sin(\theta_R) \\ \omega_e \end{bmatrix}, \tag{4.6}$$

where

$$\theta_R = \theta_e + \theta_s + \pi \tag{4.7}$$

and

$$\theta_s = \tan^{-1}\left(\frac{y_R}{x_R}\right). \tag{4.8}$$

Here, υ and ω_e are the translational and angular velocities (i.e., control laws) that will be derived in the following section by using the Lyapunov approach. However, as will be shown later, the control laws designed using the Cartesian coordinates require more information about the robot states so that the robot can be driven towards the sound source. For this reason, the control laws described by using the polar coordinates are used to control the robot kinematics represented in the Cartesian coordinates.

4.2 Lyapunov Based Controller for a Stationary Sound Source

In this work, the Lyapunov approach is used to derive the control laws needed to control the mobile robot kinematics. For the posture error using the polar coordinates, a Lyapunov candidate can be defined as follows:

$$V = \frac{1}{2} x_e^2 + \left(1 - \cos\left(\theta_e\right)\right) . \tag{4.9}$$

To ensure that this system is asymptotically stable according to the Lyapunov approach, the first time derivative of *V* in Eq. (4.9) (i.e.,$\dot{V}$) should be less than zero; that is:

$$\dot{V} \le 0 . \tag{4.10}$$

Accordingly, it can be obtained that

$$\dot{V} = x_e \ . \ \dot{x}_e + \sin\left(\theta_e\right).\dot{\theta}_e . \tag{4.11}$$

Substituting $\dot{x}_e$ and $\dot{\theta}_e$ from Eq. (4.3) into Eq. (4.11) yields

$$\dot{V} = x_e(-v \cos(\theta_e)) + \sin\left(\theta_e\right) \cdot \omega_e . \tag{4.12}$$

where

$$\begin{aligned} \omega_e &= \omega_R - \omega_S \\ \omega_S &= \frac{v_R \cdot \sin(\theta_e)}{x_e} \end{aligned} \tag{4.13}$$

To satisfy the Lyapunov approach condition, the following control laws are proposed:

$$v_R = K_v \cdot x_e \cdot \cos(\theta_e) \tag{4.14}$$

$$\omega_R = -K_\omega \sin\left(\theta_e\right) \quad , \tag{4.15}$$

where K_{xe} and $K_{\omega e}$ are the control laws gains. Since the range is always positive in a polar coordinate, the range term can be removed from Eq.(4.14) and it becomes

$$v_R = K_v \cos(\theta_e) \quad . \tag{4.16}$$

Therefore, one can find

$$\dot{V} = -K_v\, x_e^{\ 2}\ \cos^2(\theta_e) - K_\omega\ \sin\left(\theta_e\right)^2 - \frac{v_R \cdot \sin^2(\theta_e)}{x_e} \leq 0 \tag{4.17}$$

or

$$\dot{V} = -K_v\, x_e\ \cos^2(\theta_e) - K_\omega \sin\left(\theta_e\right)^2 - \frac{v_R \cdot \sin^2(\theta_e)}{x_e} \leq 0 \quad . \tag{4.18}$$

Since $\dot{V}$ in both cases is less than zero except when $x_e = \theta_e = 0$, the system is globally asymptotically stable.

On the other hand, for the posture error using the Cartesian coordinates a Lyapunov candidate can be defined as following

$$V = \frac{1}{2}\left(x_R^2 + y_R^2\right) + \left(1 - \cos\left(\theta_e\right)\right) \quad . \tag{4.19}$$

Similarly, it can be proved that the system satisfies the Lyapunov condition (Eq.(4.10)) and is thus asymptotically stable. The time derivative of V in Eq. (4.19) can be expressed as:

$$\dot{V} = x_R \cdot \dot{x}_R + y_R \cdot \dot{y}_R + \sin\left(\theta_e\right) \cdot \dot{\theta}_e \quad . \tag{4.20}$$

Substituting $\dot{x}_R, \dot{y}_R,$ and $\dot{\theta}_e$ from Eq. (4.6) yields

$$\dot{V} = -v_R(x_R \cdot \cos(\theta_R) + y_R \cdot \sin(\theta_R)) + \sin\left(\theta_e\right) \cdot \omega_e \quad . \tag{4.21}$$

Substituting θ_R from Eq. (4.7) yields

$$\dot{V} = -v_R\ (x_R \cdot \cos(\theta_e + \theta_s + \pi) + y_R \cdot \sin(\theta_e + \theta_s + \pi)) + \sin\left(\theta_e\right) \cdot \omega_e \quad . \tag{4.22}$$

To satisfy the Lyapunov stability condition, the translational and rotational velocities should follow the following laws:

$$v_R = K_v (x_R \cdot \cos(\theta_e + \theta_s + \pi) + y_R \cdot \sin(\theta_e + \theta_s + \pi)) \tag{4.23}$$

and

$$\omega_e = -K_\omega \sin\left(\theta_e\right) \quad . \tag{4.24}$$

Accordingly, the condition in Eq. (4.10) is satisfied

$$\dot{V} = -K_v \left(x_R.\cos\left(\theta_e + \theta_s + \pi\right) + y_R.\sin\left(\theta_e + \theta_s + \pi\right)\right)^2 - K_{\omega_e}.\sin\left(\theta_e\right)^2 \leq 0 \quad , \tag{4.25}$$

where

$$\theta_s = \tan^{-1}\left(\frac{y_R}{x_R}\right) \quad . \tag{4.26}$$

Since $\dot{V}$ is less than zero except when $x_R = y_R = \theta_e = 0$, the system is globally asymptotically stable. The main problem of using these control laws (Eqs. (4.23) and (4.24)) is that the information of x_R, y_R,and ω_e should be made available to calculate the proposed control laws. Unfortunately, this information is not available since the only used sensor on the platform is the acoustic sensor. Accordingly, the control laws should only be a function of the estimated sound incident angle since it is the only measurable information. Therefore, the proposed control laws given by Eqs. (4.15) and (4.16) are used in localization of a stationary sound source and the tracking of a moving sound source.

After carefully examining the control laws (Eqs. (4.15) and (4.16)), the following remarks can be made. For the translational velocity law, the velocity ranges between zero and the maximum translational velocity depending on the heading angle error; that is,

$$v \in [0, v_{\max}] . \tag{4.27}$$

As the angle gets closer to zero, the translational velocity goes to the maximum. For the angular velocity law, the angular velocity ranges between the negative of the maximum angular velocity to the positive maximum angular velocity; that is,

$$\omega_R \in [-\omega_{R_{\max}}, \omega_{R_{\max}}] \tag{4.28}$$

As the heading angle error gets closer to zero, the angular velocity goes to zero.

Furthermore, when the heading angle error is approaching ±90°, the angular velocity reaches its maximum value with a sign related to the sign of the angle error. In addition, the gains used in the control laws are basically the maximum translational and angular velocities of the robot.

Accordingly, the control laws are modified to the following forms

$$\begin{aligned} v &= K_{x_e}\left(1-\left(\frac{\theta_e}{\pi/2}\right)^2\right) \\ \omega_e &= -K_{\omega_e}\left(\frac{\theta_e}{\pi/2}\right) . \end{aligned} \tag{4.29}$$

This controller is called “Quadratic-linear Controller”, which combines linear and quadratic functions.

4.3 Simulation Results

4.3.1 Localization of Stationary Sound Source

In a preliminary effort, simulations are carried out for locating a stationary sound source using the nonlinear and quadratic-linear controllers. Considering the polar coordinate case using the translation and angular velocities laws (i.e., nonlinear controller Eqs. (4.15) and (4.16)). Figure 4.3 shows the history of the translational and angular errors. For the angular error, it converges to zero at about 1 second. However, the translational error first decreases first, and then starts to increase. The increase in the translational error is due to the fact that the robot does not stop when the measured angle error becomes zero and the robot keeps moving in a straight line until it hits the target.

The control history of the translational and angular velocities is shown in Figure 4.4. The translational velocity saturates after about 0.5 second and the robot is moving with a constant maximum speed afterwards. The angular velocity converges to zero at about 1.2 seconds.

The change of translational error with respect to the orientation error is shown in Figure 4.5. The convergence of these velocities depends on the gains used in the control laws (i.e., K_v, and K_ω). The gain values used in this simulation are 0.68 m/s and 3.88 rad/s, respectively. These control gains are chosen according to the physical capabilities of the robot in terms of the maximum translational and angular speeds.

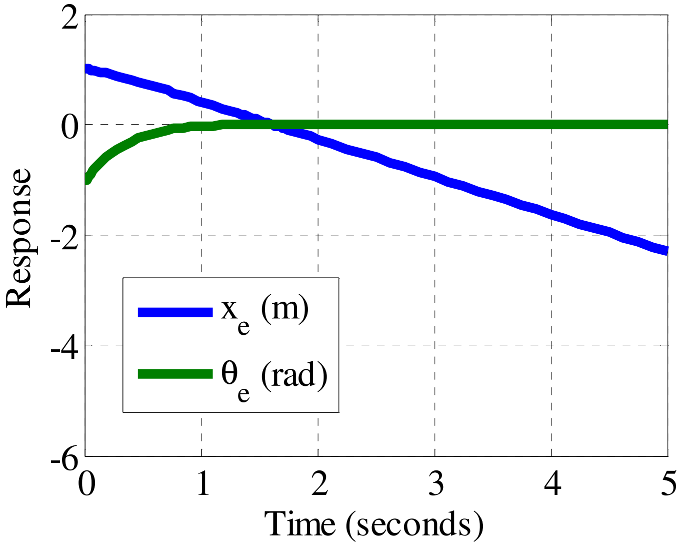


Figure 4.3: History of error obtained by using the nonlinear controller.

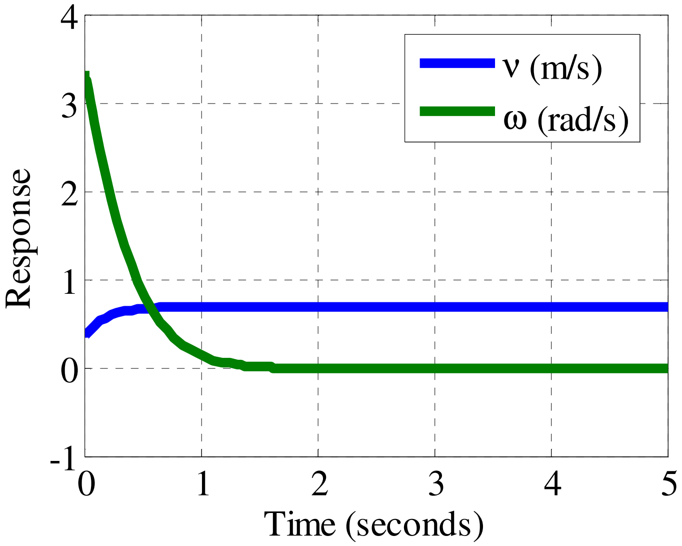


Figure 4.4: History of translational and Angular velocities by using the nonlinear controller.

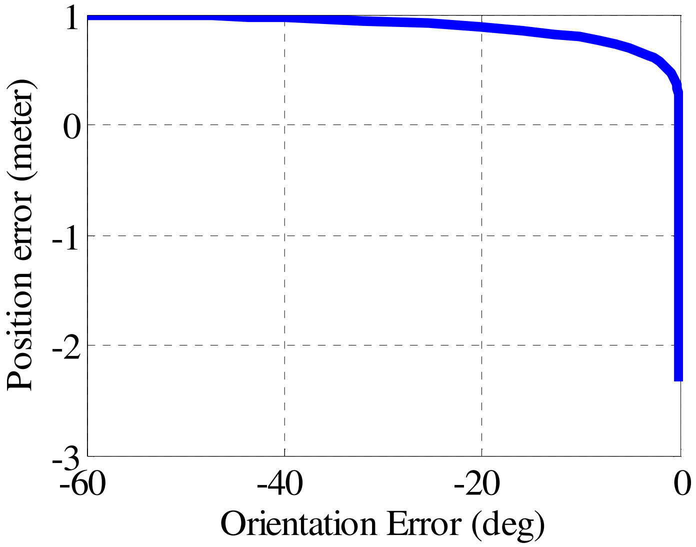


Figure 4.5: Translational error x_e as a function of θ_e using the nonlinear controller.

Next, the performance of the quadratic-linear controller (Eqs. (4.29)) is studied in the simulation.. Figure 4.6 shows the error tracking history, in which the angular error is shown to converge to zero at about 1.2 seconds. Again, the translational error is shown to decrease first, and then start to increase. Compared with the nonlinear controller, the angle error for the quadratic-linear controller reached zero slower because the angular control effort is less.

Figure 4.7 shows the history of translational and angular velocities (control efforts). The translational velocity is shown to saturate after about 1 second the robot is moving with a constant maximum speed afterwards. The angular velocity converges to zero at about 1.5 seconds. The translational error changes with respect to the orientation error are shown in Figure 4.8.

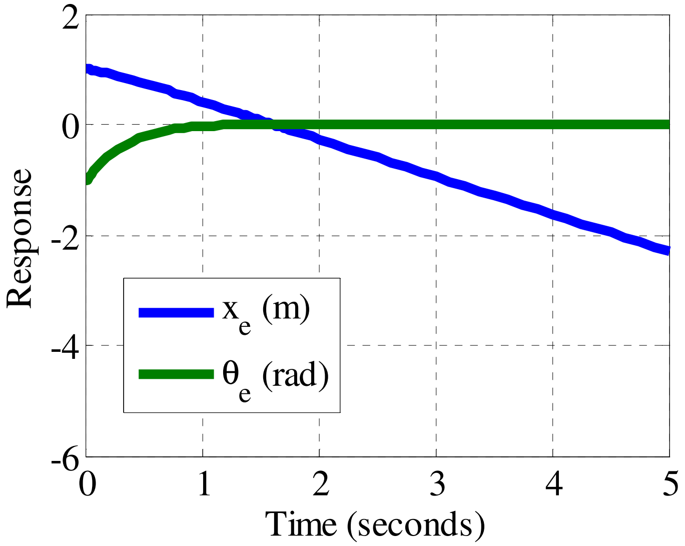


Figure 4.6: History of error obtained by using the quadratic-linear controller.

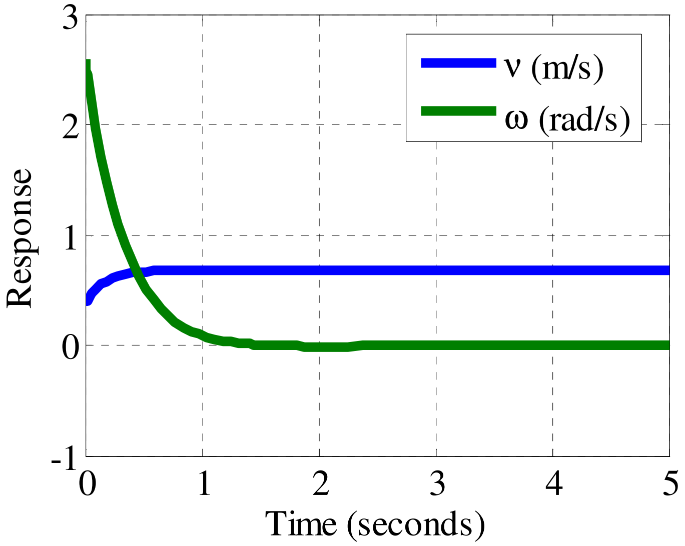


Figure 4.7: History of translational and angular velocities using the quadratic-linear controller.

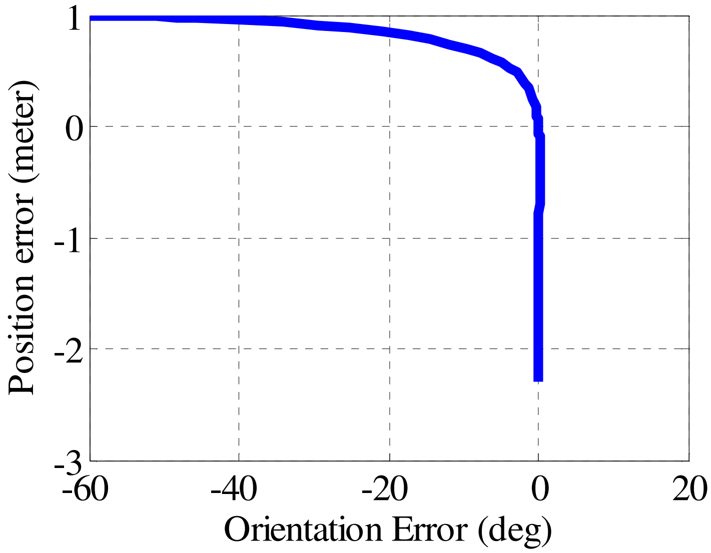


Figure 4.8: Tracking error history of x_e as a function of measured error θ_e using the Quadratic-linear controller.

The problem in the localization simulations is that since the polar coordinate is used, there is no information about the trajectory of the robot (i.e., if the robot reaches the sound source and how it reaches it). Therefore, in order to show the robot trajectory, the control laws derived using the polar coordinates are used along with the robot kinematics model in the Cartesian coordinates in the inertial frame. Again, no information about the robot posture can be attained except the robot heading angle error, which is defined with respect to the virtual line connecting the robot and the sound source. In this case, not only the errors can be tracked, but also the robot motion can be obtained.

A 3D animation of the robot in a virtual environment for localization of a stationary source by using the nonlinear and quadratic-linear controllers is used to gain more insight about the robot behavior. Figure 4.9 shows the virtual environment built to show the robot trajectory while localizing the sound source (pink box). The results are shown in Figure 4.10 and Figure 4.11. For each controller, six parameter are plotted, namely, angular velocity, translation velocity, robot heading angle, sensor measurements, and the robot position. The results show that by using either controller the robot was able to orient itself and move towards the source. Here, noise was not taken into consideration due to the limitation of MATLAB ode45 function (SIMULINK block diagrams are provided in Appendix G).

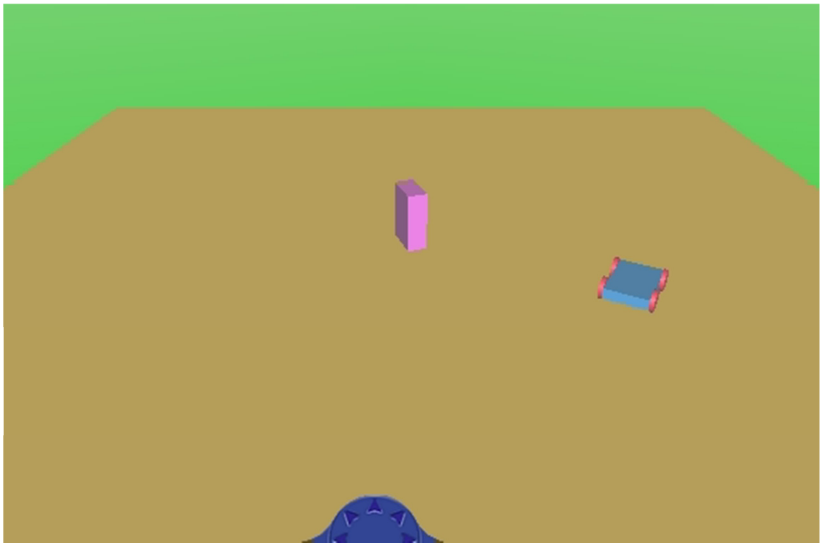


Figure 4.9: 3D virtual environment.

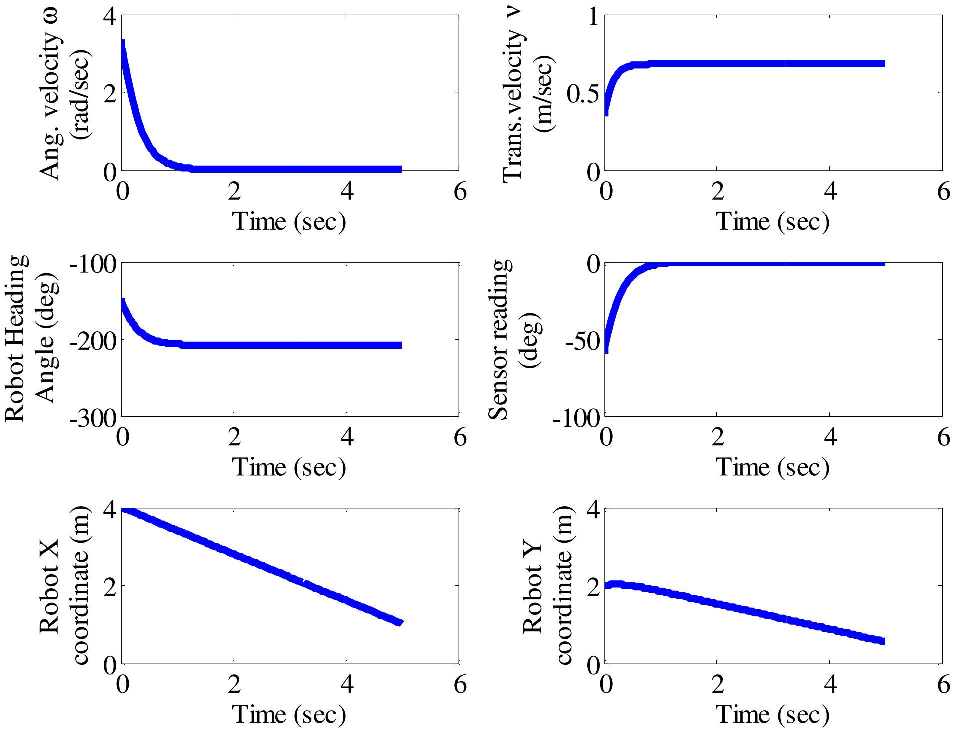


Figure 4.10: System responses obtained by using the nonlinear controller.

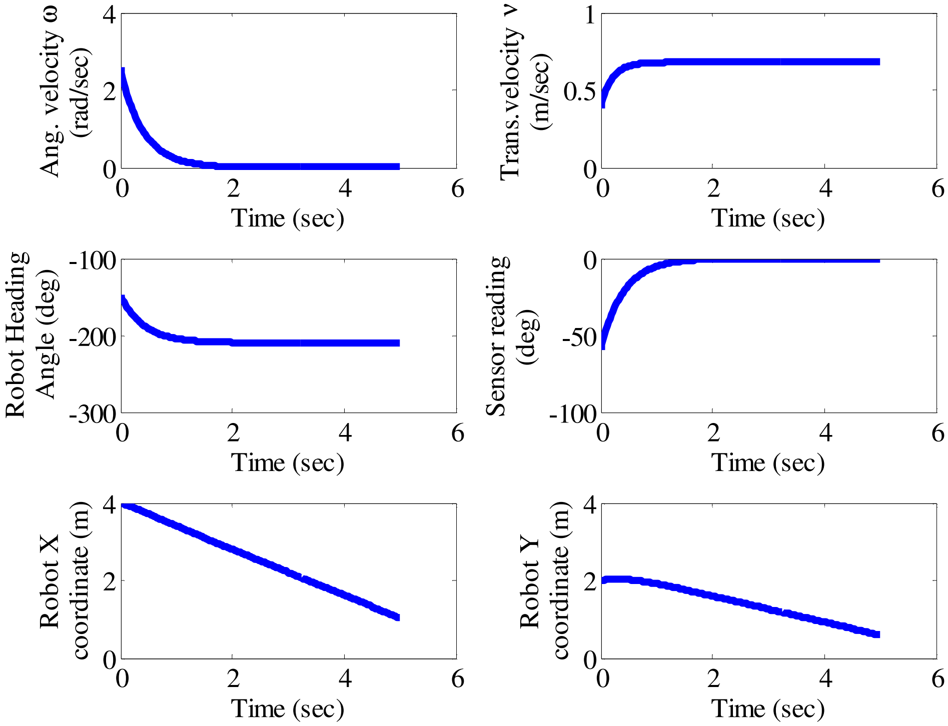


Figure 4.11: System responses obtained by using the quadratic-linear controller.

Furthermore, simulations are carried out to compare the performance of the developed controllers with the controllers reported in the literature, which include the proportional controller presented by Andersson *et al.* [5] and the PID controller presented by Song *et al.* [19] utilizing an Euler integrator. In the simulations, the sound source is assumed to be at the origin and the tracking robot is 5 meters away. The initial orientation of the tracking robot is assumed to be 90^o, which corresponds to an initial heading angle error of -90^o. The gains for the PID controller are determined by executing enumerative search in the integrator gain (K_I) and derivative gain (K_D) spaces and keeping the proportional gain (K_P) fixed. To find the optimal gains, some performance metrics, namely, the rising time, the settling time, the overshoot, the PID output, and length of the trajectory are evaluated and the gain values that enable better performance are chosen. Accordingly, the optimal gains (K_I, K_D, K_P) are found to be 0.01, 0.005, and 0.9, respectively. Figure 4.12 compares the step responses obtained with different controllers. As shown in Figure 4.12(a), the angle error reaches zero faster for the nonlinear controller than that for the other controllers. With a quadratic-linear and proportional controllers, the angle error converges to zero at about the same speed convergence with the PID controller is the slowest. The control effort used in different controllers are compared in Figure 4.12(c) and (d). For the nonlinear controller, the translation and angular velocities reach steady state faster than those for the other controllers. The trajectory of the tracker is shown in Figure 4.12(e). In all cases, the tracker robot is able to localize the source, but followed different trajectories. Using the nonlinear controller, not only the error converges faster but also the robot follows the shortest trajectory.

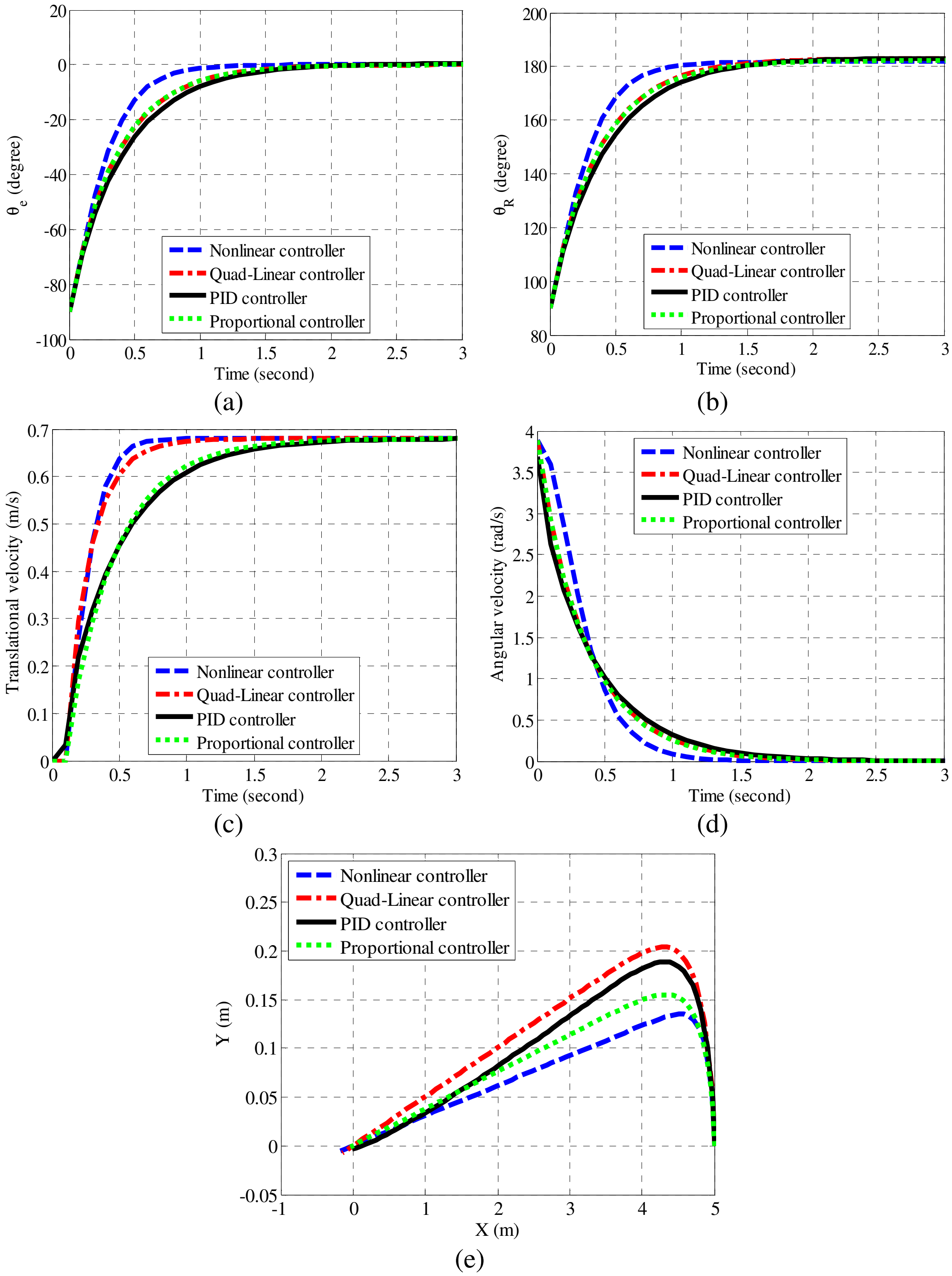


Figure 4.12: Response obtained with different controllers; (a) angle error history, (b) tracker heading angle history, (c) translational velocity history, (d) angular velocity history, and (e) tracker trajectory.

4.3.2 Effect of Measurement Error on Localization Performance

To investigate how the measurement error affects the sound localization performance, a random noise with a normal distribution was added to the measured sound incident angle and the step responses were obtained. Here, three metrics are defined to compare the performance of the different controllers with the additive noise, which are the settling time, the steady state error, and the error band of the estimated error angle. The settling time in these simulations is defined as the time at which the estimated incident angle becomes smaller than a threshold value. The steady state error is defined as the average of the estimated sound incident angle starting from the settling time until the tracking robot reaches the sound source. The error band is defined as the maximum range of error of the estimated angle starting from the settling time.

The settling time of the estimated sound incident angle using the different controllers is shown in Figure 4.13. The nonlinear controller exhibit the shortest settling time compared with the other methods. The steady state error of the estimated sound incident angle is shown in Figure 4.14. There is no clear relationship that can be observed between the measurement noise and the settling time as well as the steady state error for all the controllers. Figure 4.15 shows the error band of the estimated incident angle. It can be seen that for all the controllers the error band increases as the estimated incident angle noise increases.

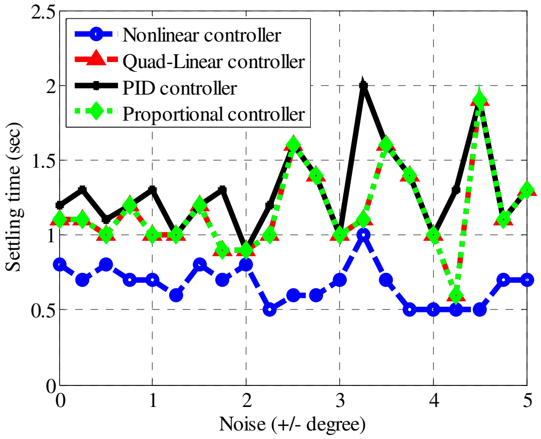


Figure 4.13: Settling time versus the measurement noise.

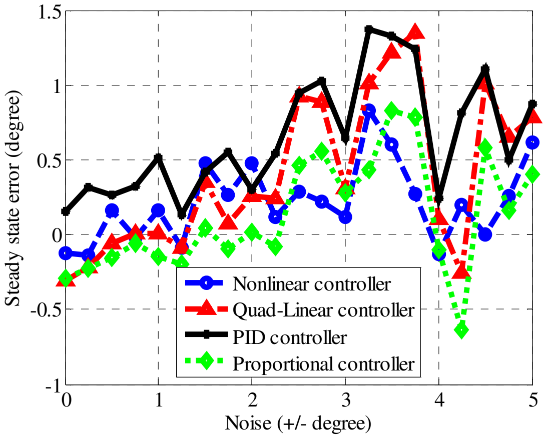


Figure 4.14: Steady state error versus the measurement noise.

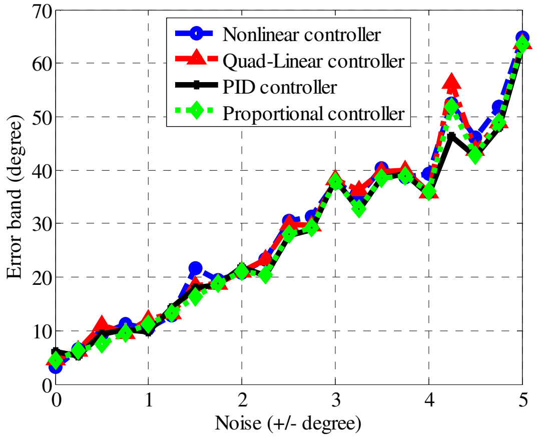


Figure 4.15: Error band versus the measurement noise.

4.3.3 Tracking of a Moving Sound Source

In this subsection, tracking of a moving sound source is investigated. The challenge as mentioned earlier is that the nonlinear and quadratic-linear control laws, that need to be only a function of the angle error. Figure 4.16 shows the schematic of the tracking robot and the trajectory of the sound source that the robot should follow using only information of the angle error θ_e.

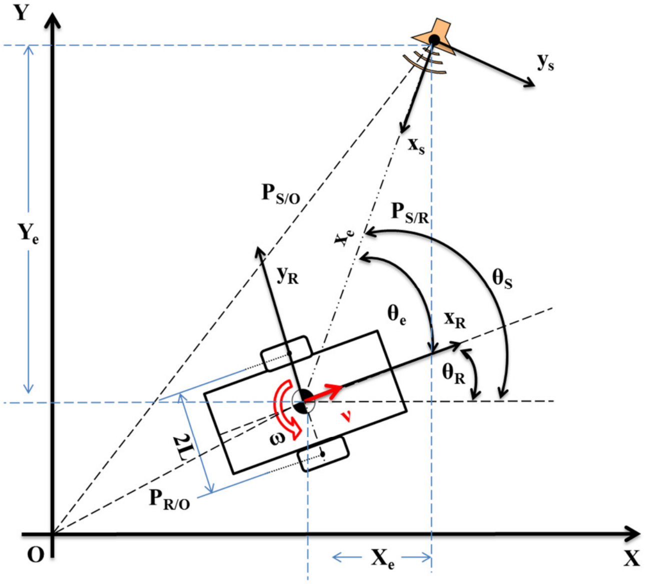


Figure 4.16: Schematic of the robot and a moving sound source positions. θ_e is the heading angle error, θ_R is the robot heading angle, $P_{S/O}$ is the position vector of the sound source relative to the origin, $P_{R/O}$ is the position vector of the robot relative to the origin, and $P_{S/R}$ is the position vector of the sound source relative to the robot, ω and υ are the robot angular and translational velocities, 2L is the width of the robot, and X_e and Y_e are the X and Y axis posture errors of the sound source relative to the robot in the global coordinate.

The kinematics of the robot in the robot body frame can be described as [70]

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos(\theta_R) & \sin(\theta_R) & 0 \\ -\sin(\theta_R) & \cos(\theta_R) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix}, \tag{4.30}$$

where

$$\begin{aligned} x_e &= x_R - x_S \\ y_e &= y_R - y_S \\ \theta_e &= \theta_R - \theta_S \end{aligned} \quad . \tag{4.31}$$

Taking the time derivative of Eq. (4.30) gives

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} v_R + \omega_R \cdot e_2 - v_S \cdot \cos(e_3) \\ -\omega_R \cdot e_1 + v_S \cdot \sin(e_3) \\ \omega_R - \omega_S \end{bmatrix} . \tag{4.32}$$

Here, the actual position and orientation of the robot are denoted by $x_R(t)$, $y_R(t)$, $\theta_R(t)$, respectively, and the time-varying sound position and orientation are denoted by $x_S(t)$, $y_S(t)$, $\theta_S(t)$, respectively. The sound source is assumed omnidirectional. Based on the sound and tracker position, θ_S can be defined as

$$\theta_S = \tan^{-1}\left(\frac{y_S - y_R}{x_S - x_R} \right) . \tag{4.33}$$

The time derivative of Eq. (4.33) is obtained as

$$\omega_S = \frac{\cos^2(\theta_S)}{(x_S - x_R)^2} \left((x_S - x_R)(\dot{y}_S - \dot{y}_R) - (y_S - y_R)(\dot{x}_S - \dot{x}_R) \right) . \tag{4.34}$$

Following the Lyapunov approach, let the Lyapunov candidate be

$$V = \frac{1}{2}\left(e_1^2 + e_2^2\right) + \left(1 - \cos\left(e_3\right)\right) . \tag{4.35}$$

Using the same control laws derived earlier for the stationary sound source (Eqs. (4.15) and (4.16)) and taking the time derivative of the candidate, one can obtain that

$$\dot{V} = e_1.\dot{e}_1 + e_2.\dot{e}_2 + \sin(e_3).\dot{e}_3 . \tag{4.36}$$

Substituting $\dot{e}_1$, $\dot{e}_2$, and $\dot{e}_3$ from Eq. (4.32) into Eq. (4.36) leads to

$$\dot{V} = (K_v - v_S) \cdot e_1.\cos(e_3) + v_S \cdot e_2 \cdot \sin(e_3) + \sin(e_3)(-K_\omega \cdot \sin(e_3) \text{-} \omega_S) , \tag{4.37}$$

where $$-\frac{\pi}{2} \le e_3 \le \frac{\pi}{2}\,. \tag{4.38}$$

To satisfy the Lyapunov condition, the following inequalities are found:

$$\begin{aligned}
&\cos(e_3) \ge 0 \\
&-K_\omega \sin^2(e_3) < 0 \\
&(K_v - v_S)\cdot e_1 \cdot \cos(e_3) + \sin(e_3)(v_S \cdot e_2 \text{ - } \omega_S) \ < \ 0 \\
&\tan(e_3) \ > \ \frac{v_S - K_{x_e}}{v_S \cdot e_2 \text{ - } \omega_S} \\
&or, \\
&\ (K_v - v_S)\cdot e_1 . \cos(e_3) \ < \ 0 \\
&and \\
&\ (v_S \cdot e_2 \text{ - } \omega_S)\sin(e_3) \ < \ 0 \qquad .
\end{aligned} \tag{4.39}$$

To gain more insight about the conditions that ensure a stable system, the localization and tracking problem is simulated by using the MatLab SIMULINK. In the simulations, a moving sound source that follows a circular path at a constant velocity is considered; that is,

$$\begin{aligned}
x_S &= -R \cdot \sin(-\alpha t) \\
y_S &= -R \cdot \cos(-\alpha t)\ ,
\end{aligned} \tag{4.40}$$

where *R* is the radius of the circular path and *α*=0.3 is the frequency of completing one revolution of the circular path.

In the simulations different control scenarios are studied with different types of controllers. Both the robot kinematics and the motors of the robot can be controlled. Two PID controllers are designed to control the left and right side motors. Furthermore, a PID controller is used to control the robot kinematics while tracking the sound source. Table 4-1 summarizes the different simulation scenarios (the SIMULINK block diagrams are provided in Appendix G).

Table 4-1: List of different simulation scenarios.

Controller type	**Control Laws**	**PID Motor Control**	**Initial conditions** **S: sound source** **R: robot**
Non-linear	$\nu = K_{x_e}\cos(\theta_e)$ $\omega_e = -K_{\omega_e}\sin\left(\theta_e\right)$	No	S: (2 m, 0 m) R: (3 m, 3 m, 0^o)
Non-linear	$\nu = K_{x_e}\cos(\theta_e)$ $\omega_e = -K_{\omega_e}\sin\left(\theta_e\right)$	Yes	S: (2 m, 0 m) R: (3 m, 3 m, 0^o)
Quadratic-linear	$\nu_R = K_{x_e}\left(1-\left(\frac{\theta_e}{\pi/2}\right)^2\right)$ $\omega_R = -K_\omega\left(\frac{\theta_e}{\pi/2}\right)$	No	S: (2 m, 0 m) R: (3 m, 3 m, 0^o)
Quadratic-linear	$\nu_R = K_{x_e}\left(1-\left(\frac{\theta_e}{\pi/2}\right)^2\right)$ $\omega_R = -K_\omega\left(\frac{\theta_e}{\pi/2}\right)$	Yes	S: (2 m, 0 m) R: (3 m, 3 m, 0^o)
Linear	PID	No	S: (2 m, 0 m) R: (3 m, 3 m, 0^o)
Linear	Proportional	No	S: (2 m, 0 m) R: (3 m, 3 m, 0^o)

Figure 4.17 - Figure 4.19 show the tracking results obtained by using the nonlinear controller with the sound and robot initial positions and orientations listed in Table 4-1. As shown in Figure 4.17, the robot was able to track the sound source with a small heading error. However, as shown in Figure 4.18, after about 13 seconds, the angle error starts to have large fluctuations. The reason for this is that the robot and the sound source positions coincide starting at that time (i.e., the tracking robot hit the moving source). This also leads to a considerable control effort as can be seen in Figure 4.19. When increasing the rotational speed of the sound source (i.e., increasing α) up to a certain level it is possible to let the tracker keep following the source without hitting it. This means that there should be a relative speed limit that depends

on the application. For example, a missile tracker may not only need track but also hit the target.

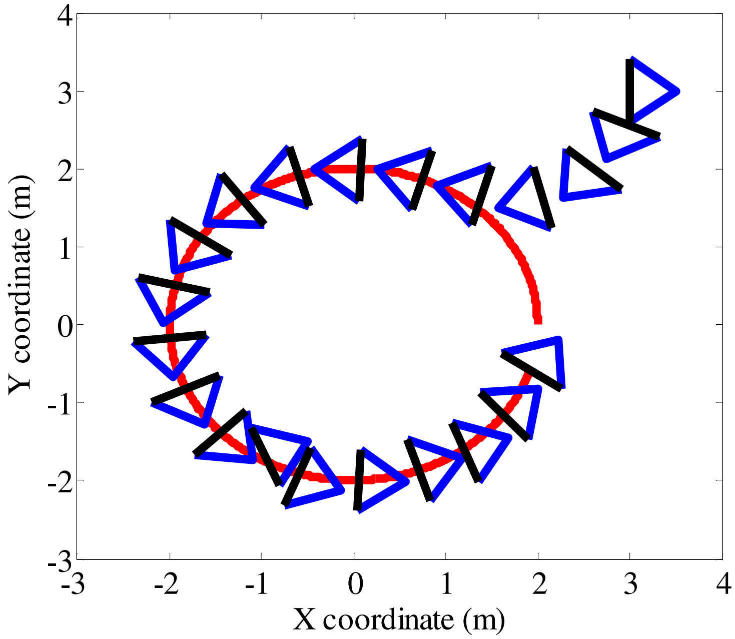


Figure 4.17: Trajectory of the sound source (red curve), and the robot (triangles) by using the nonlinear controller without PID wheel control.

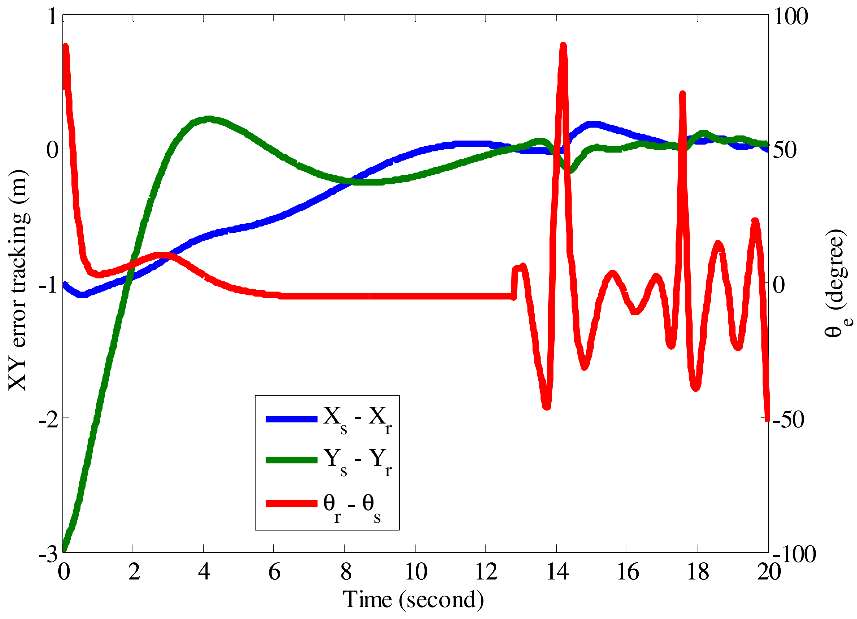


Figure 4.18: Tracking errors obtained with the nonlinear controller without PID wheel control.

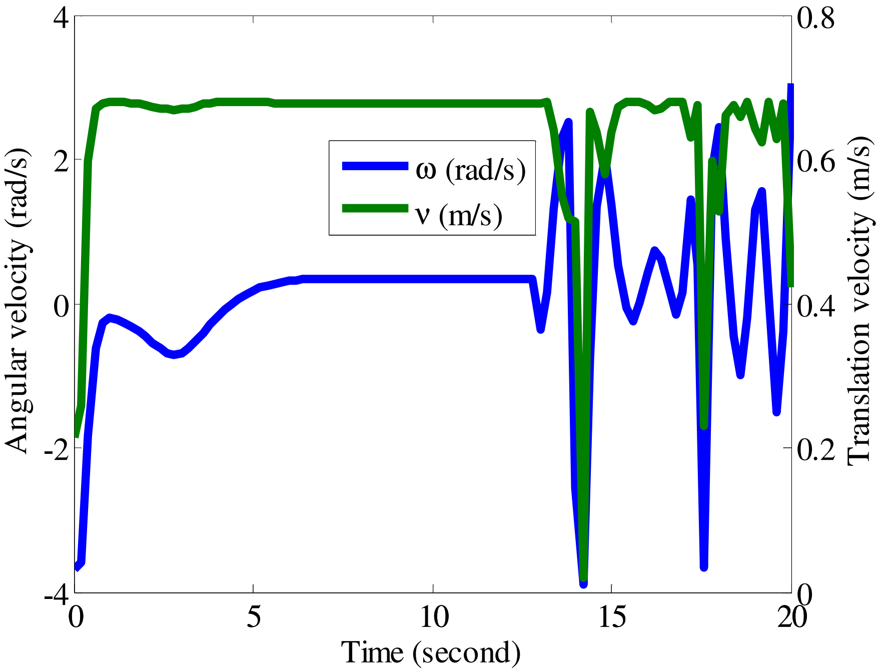


Figure 4.19: History of control effort obtained with the nonlinear controller without PID wheel control.

The sound source tracking results obtained with the nonlinear controller with PID wheel control are shown in Figures Figure 4.20 - Figure 4.25. Although the robot is able to track the sound source, but the angle error has considerable fluctuations. The reason for this is that the wheel control loop needs time until it reaches the steady state. During this time the angle error is not updated and the robot has to put considerable effort to reach the desired motor speeds based on the error measurements. This also limits the relative velocity between the robot and the sound source. Furthermore, as shown in Figure 4.22, the control efforts have considerable variations, especially the angular velocity. This will result in frequent rotation of robot, which can lead to high power consumption and shorten the life time of the electrical components of the robot. Figure 4.23 shows the history of torques of the motors based on the required motor angular velocities shown in Figure 4.24.

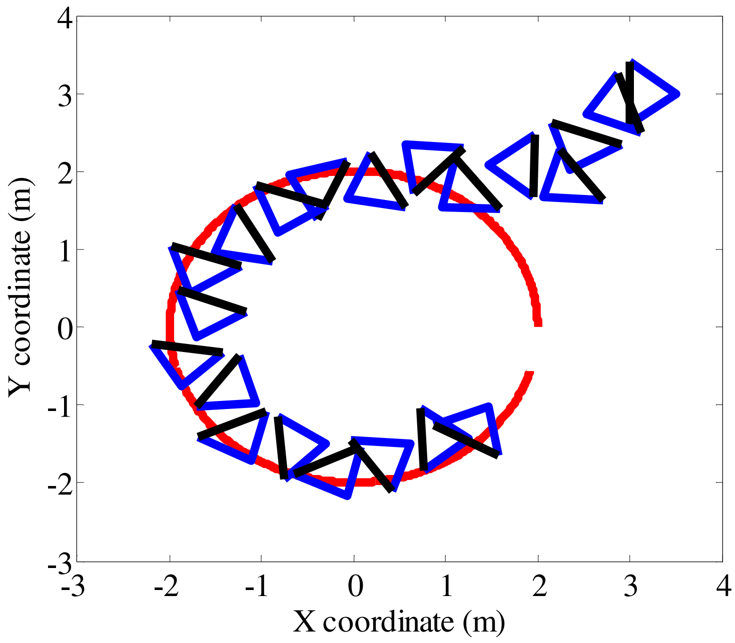


Figure 4.20: Trajectory of the sound source (red curve), and the robot (triangles) by using the nonlinear controller with PID wheel control.

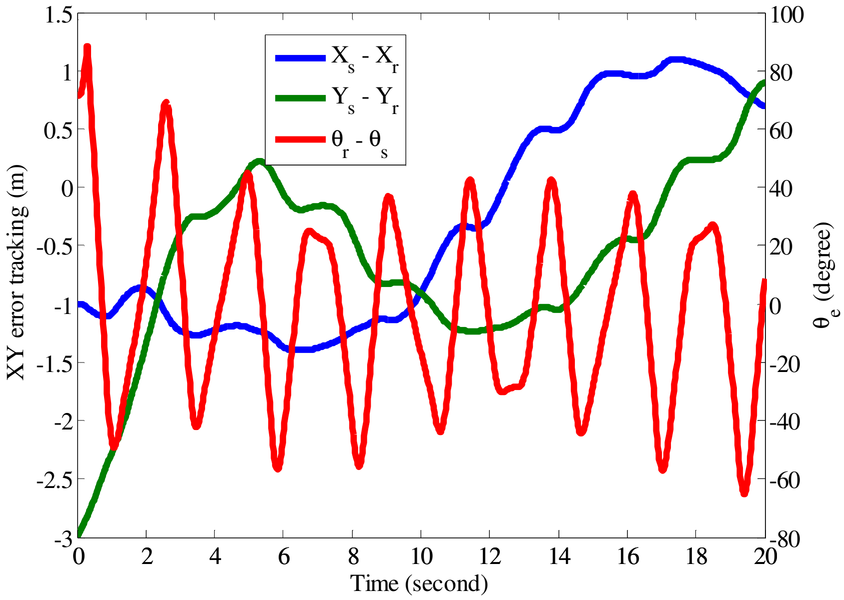


Figure 4.21: Tracking errors using the nonlinear controller with PID wheel control.

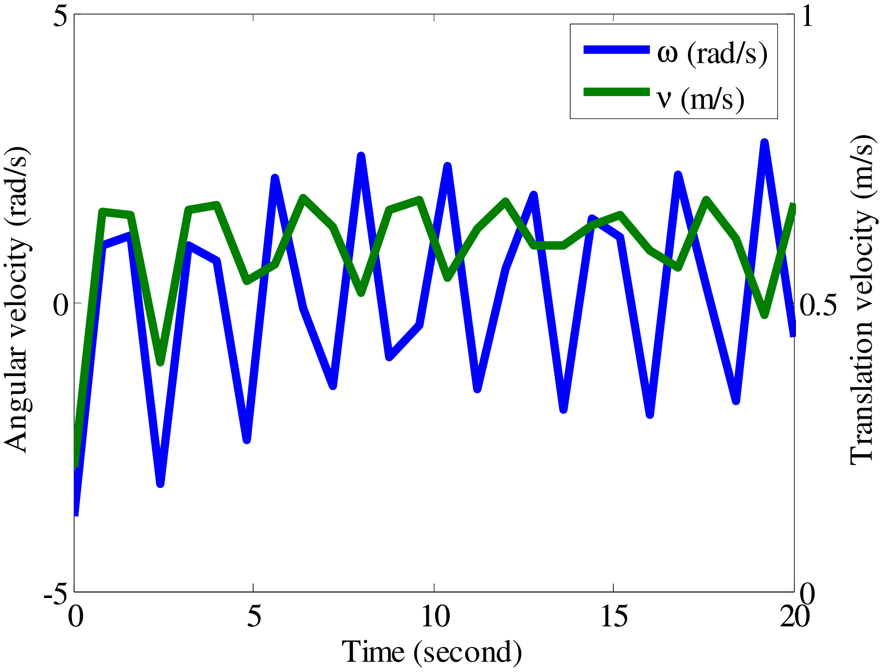


Figure 4.22: History of control efforts by using the nonlinear controller with PID wheel control.

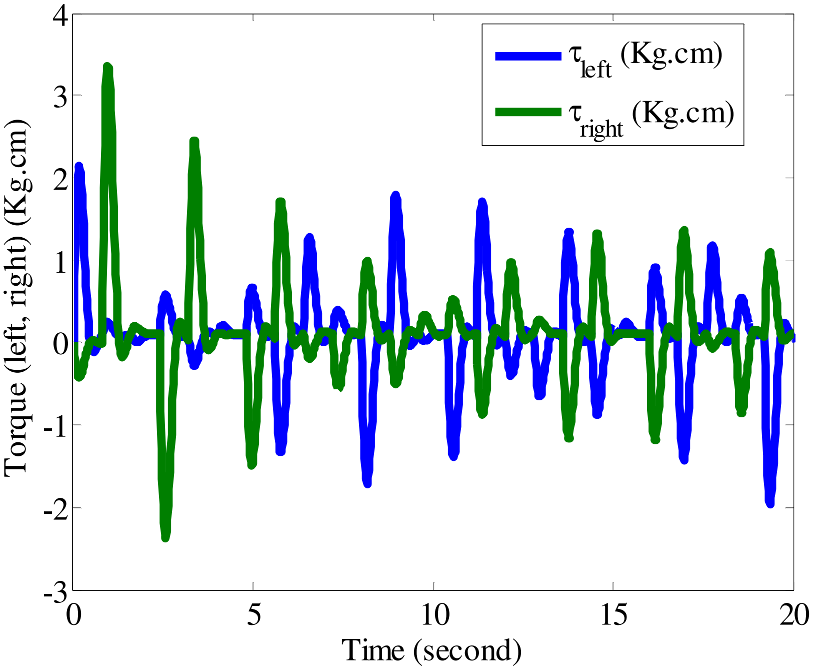


Figure 4.23: History of wheel torques by using the nonlinear controller with PID wheel control.

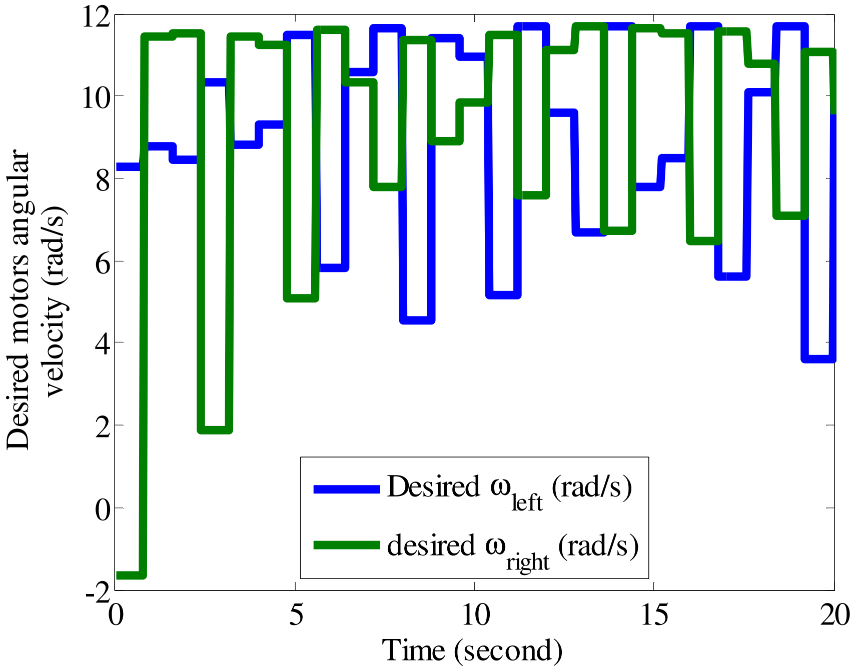


Figure 4.24: Required angular velocities of the wheels by using the nonlinear controller with PID wheel control.

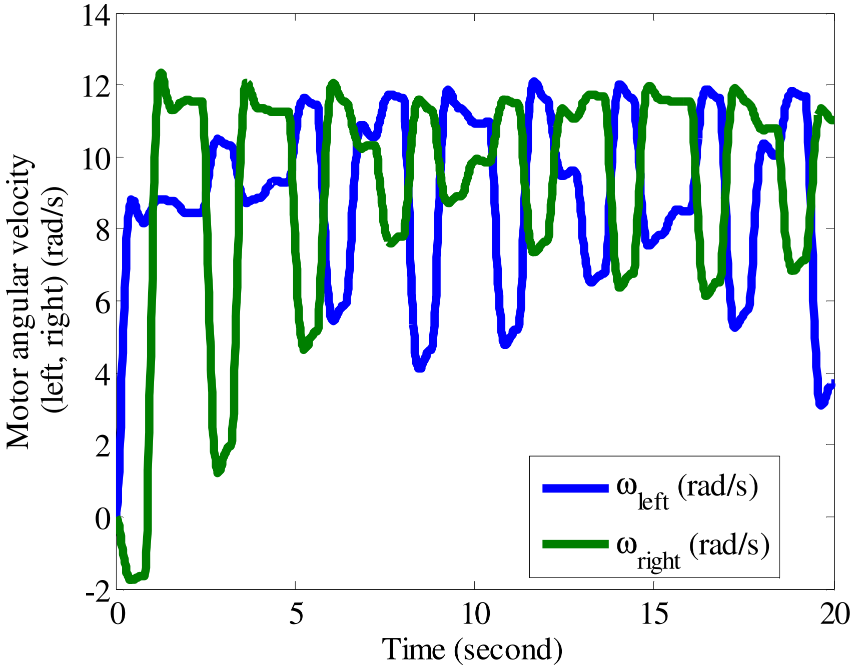


Figure 4.25: Wheels angular velocities obtained by using the nonlinear controller with PID wheel control.

The tracking results obtained by using the quadratic-linear controller without the PID wheel control by using the same sound and robot initial positions and orientations (listed in Table 4-1) are shown in Figures Figure 4.26 - Figure 4.28. Similarly, as shown in Figure 4.27 the angle error starts to have large fluctuations at about 5 seconds. This happens slower than that using the nonlinear controller. The reason for the angle error fluctuations again is due to position coincidence of the robot and the sound source. This also leads to a considerable control effort as shown in Figure 4.28, which is smaller than the one shown in the nonlinear controller.

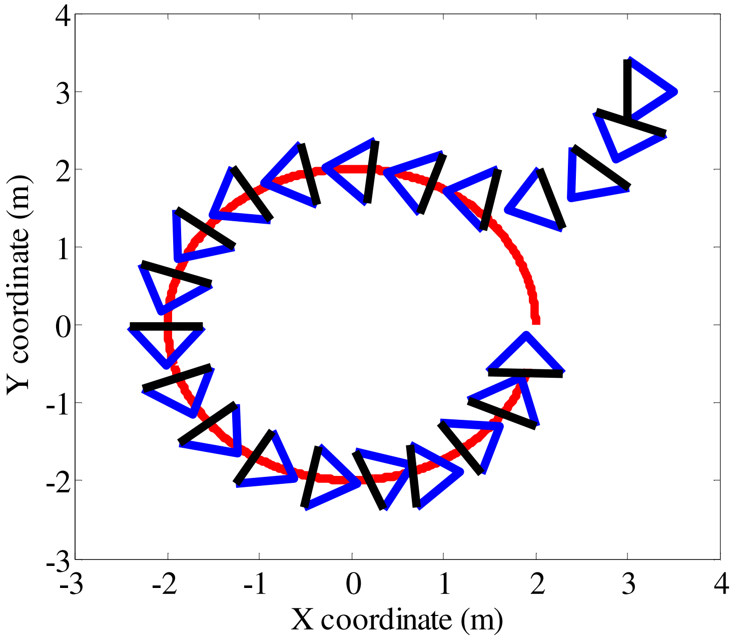


Figure 4.26: Trajectories of the sound source (red curve), and the robot (triangles) obtained by using the quadratic-linear controller without PID wheel control.

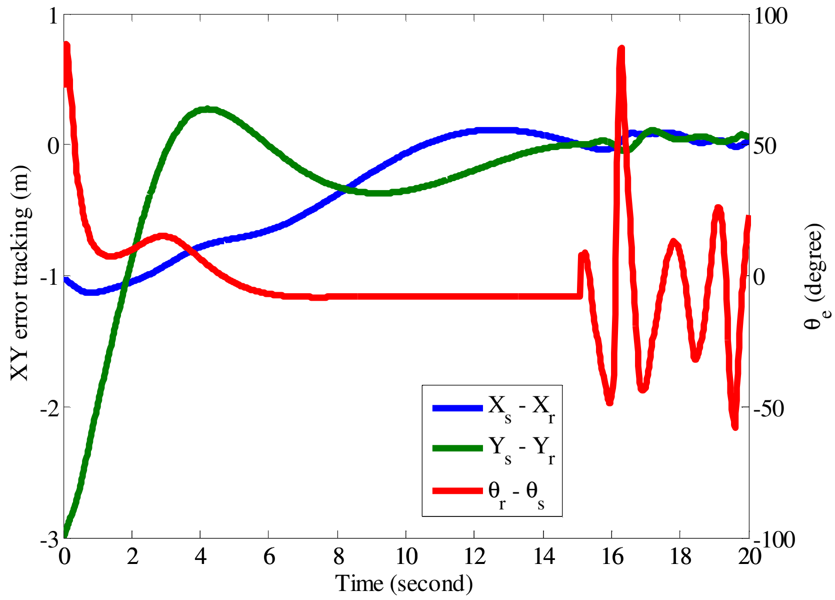


Figure 4.27: Tracking errors using the quadratic-linear controller without PID wheel control.

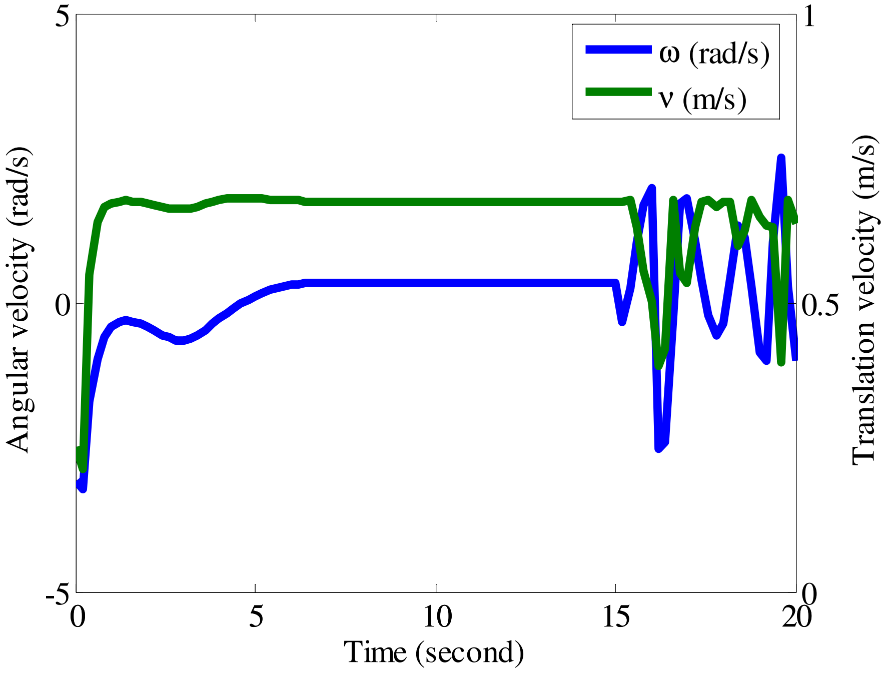


Figure 4.28: Control efforts using the quadratic-linear controller without PID wheel control.

The tracking results obtained by using the quadratic-linear controller with the wheel control as shown in Figure 4.29 - Figure 4.34. The robot is shown to be able to track the sound source with less angle error variations compared with those obtained by using the nonlinear controller with the wheel control. The reason for the variations is again due to the slow wheel control loop. On the other hand, compared with the results obtained with the nonlinear controller, the variations are less because the quadratic-linear controller has less control effort, as shown in Figure 4.31. With less variation in angle error, the power consumption will be less and the life time of the electrical components will be longer. Figure 4.32 shows the corresponding motor torques based on the required motors angular velocities shown in Figure 4.33.

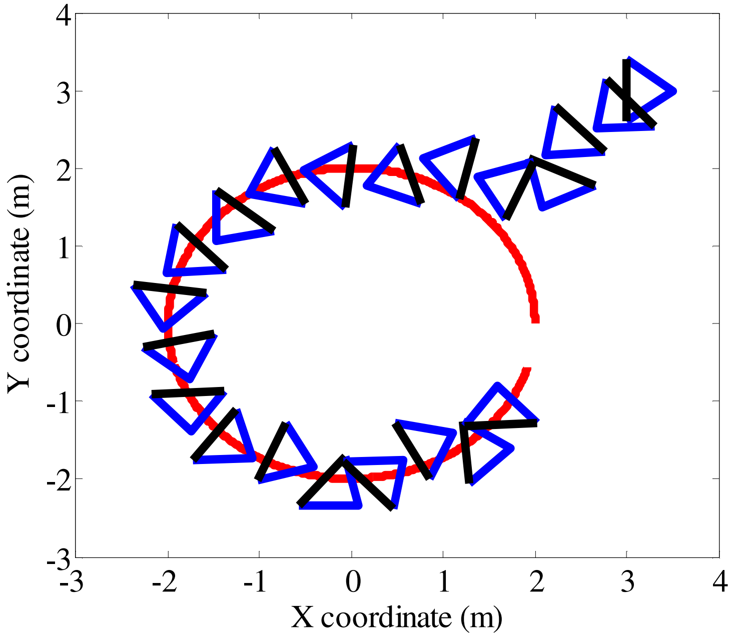


Figure 4.29: Trajectories of the sound source (red curve), and the robot (triangles) obtained by using the quadratic-linear controller with PID wheel control.

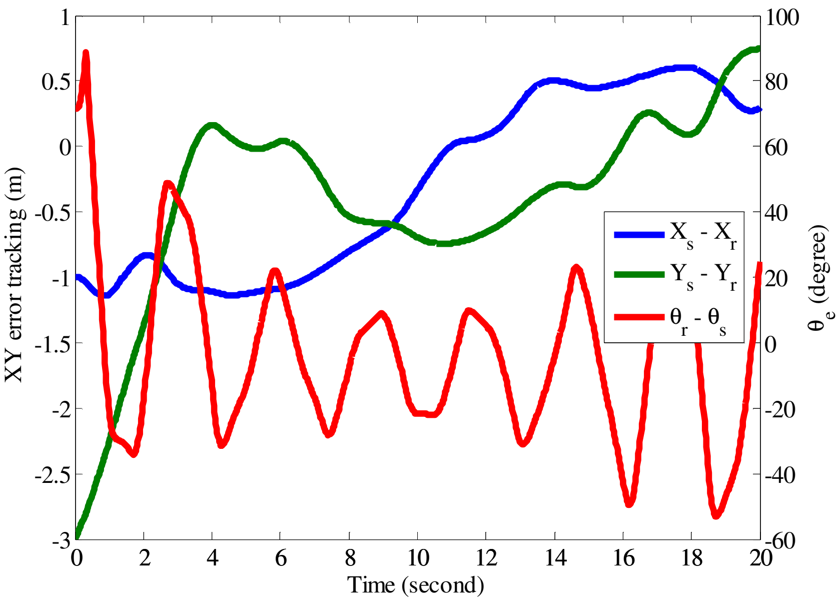


Figure 4.30: Tracking errors by using the quadratic-linear controller with PID wheel control.

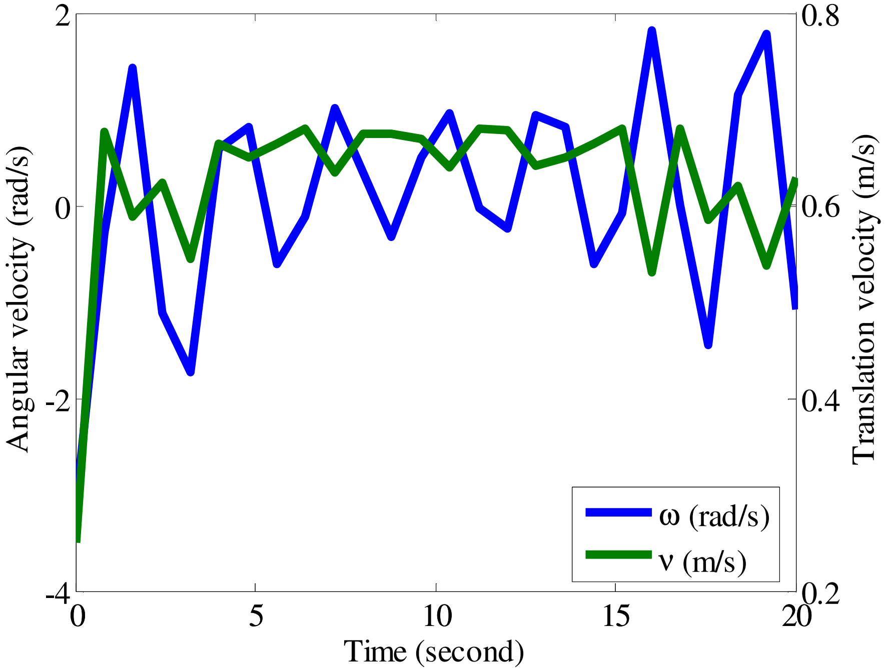


Figure 4.31: Control efforts by using the quadratic-linear controller with PID wheel control.

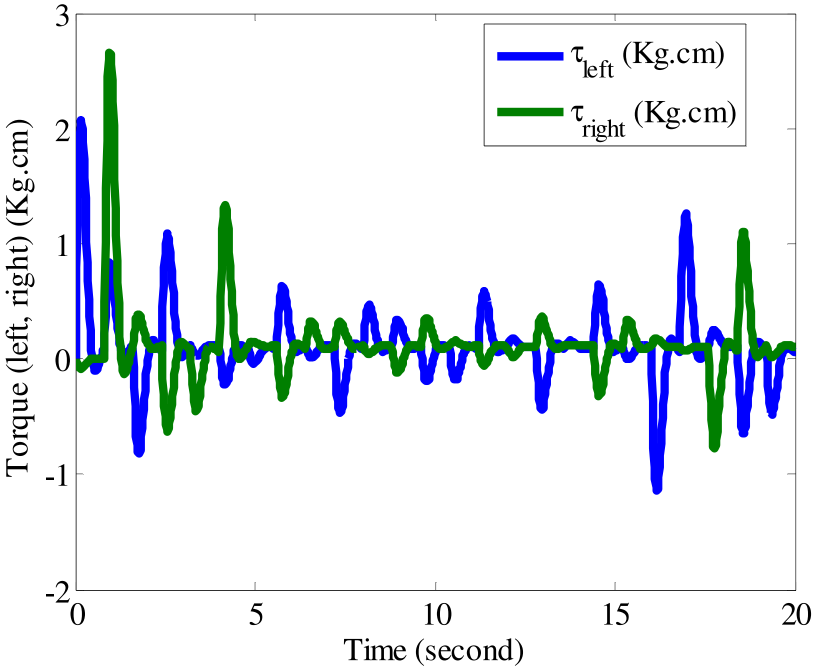


Figure 4.32: History of wheel torques by using the quadratic-linear controller with PID wheel control.

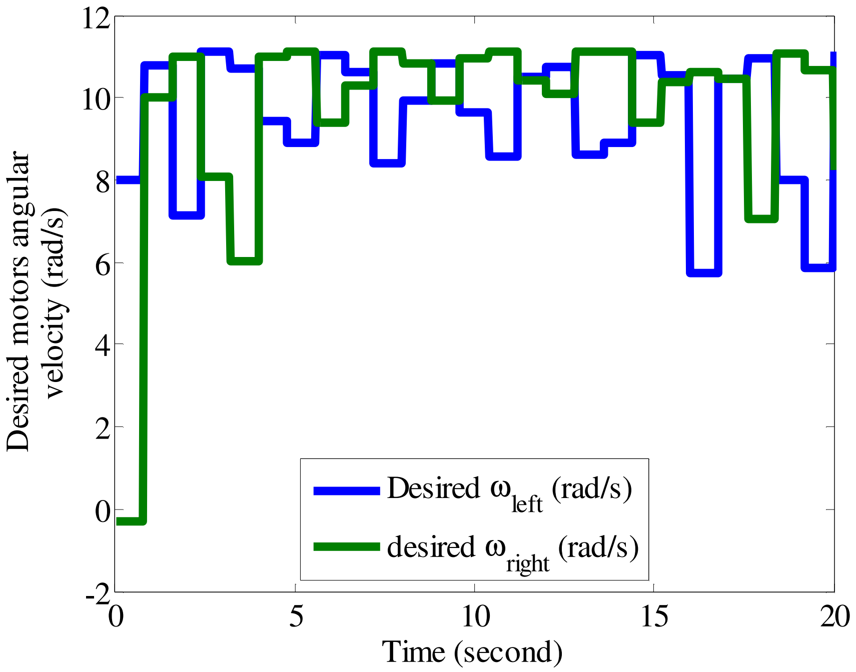


Figure 4.33: Required angular velocities of the wheels by using the quadratic-linear controller with PID wheel control.

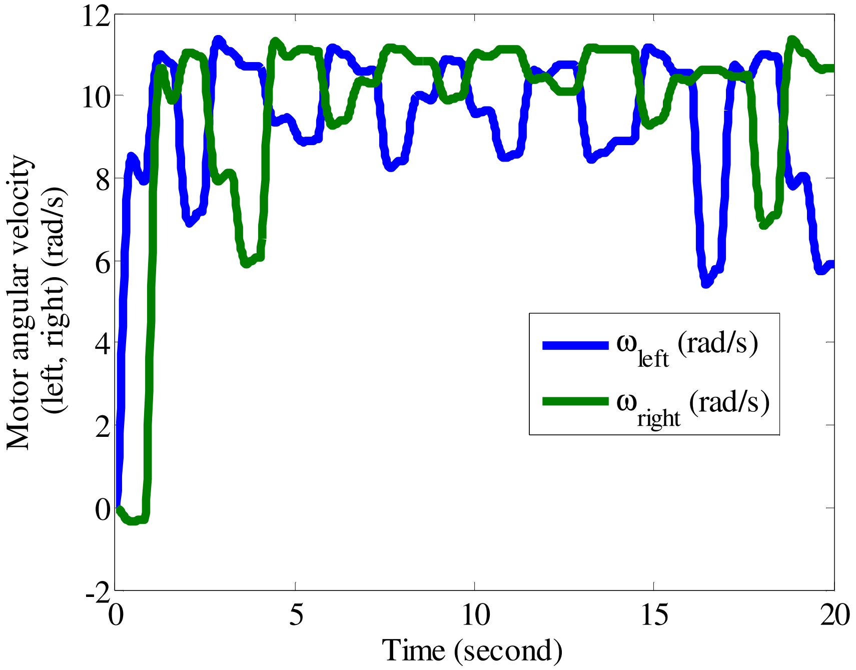


Figure 4.34: Wheels angular velocities obtained by using the quadratic-linear controller with PID wheel control.

The sound tracking results obtained by using a PID controller are shown in Figure 4.35 - Figure 4.37. Due to the slow convergence of the PID controller, which was discussed, the tracking robot does not hit the target during the tracking process compared with the nonlinear and quadratic-linear controller cases. On the other hand, Figure 4.37 shows the control efforts by using the PID controller.

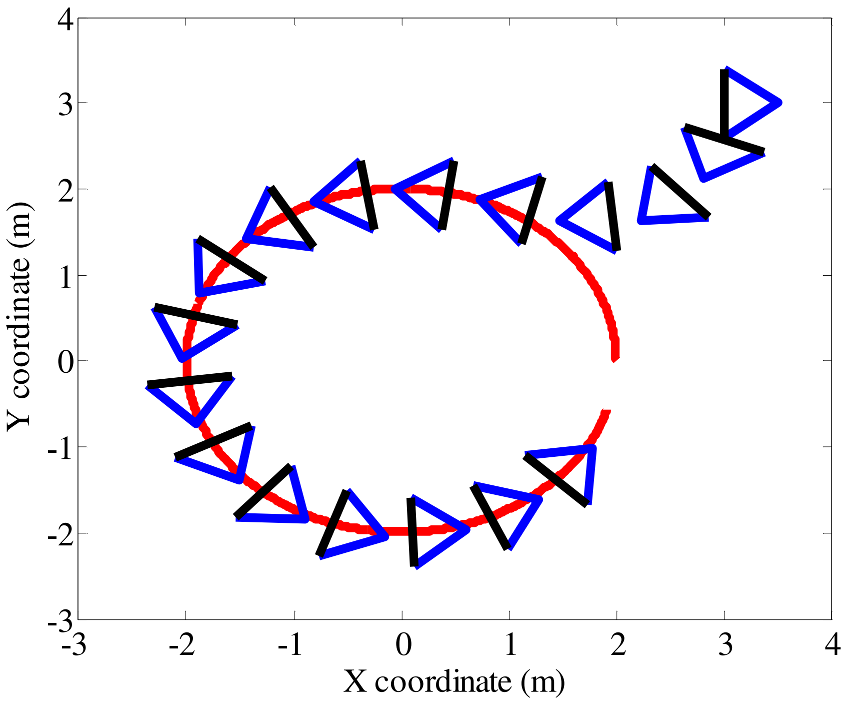


Figure 4.35: Trajectories of the sound source (red curve), and the robot (triangles) obtained by using the PID controller.

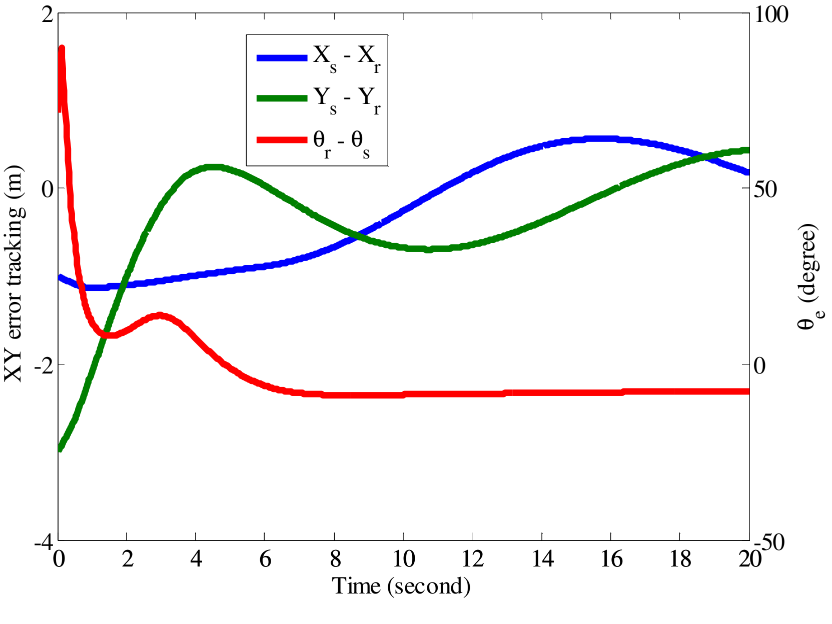


Figure 4.36: Tracking errors using the PID controller.

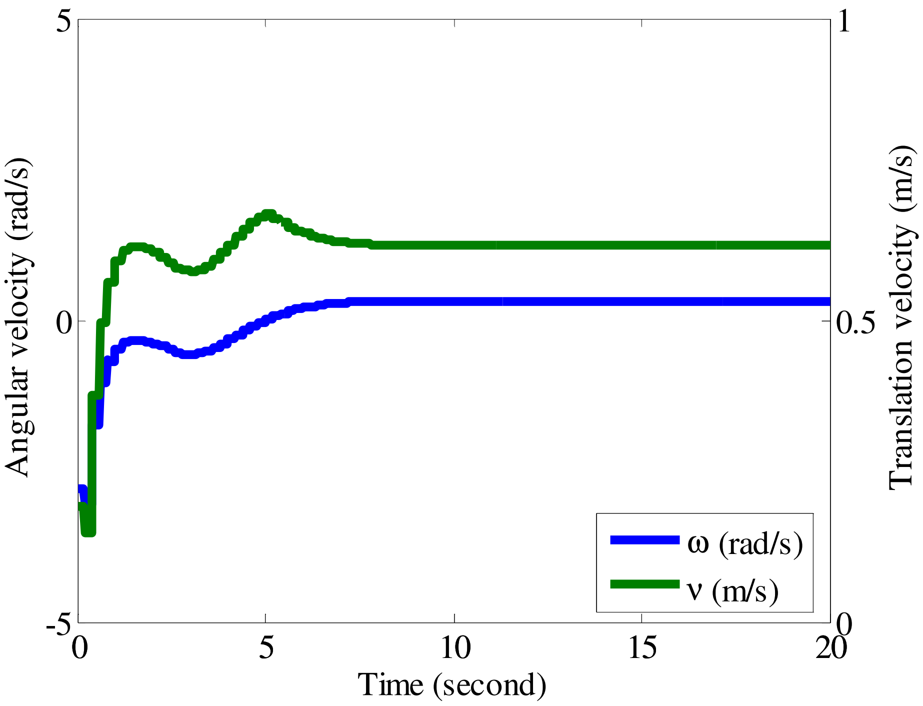


Figure 4.37: Control efforts by using the PID controller.

Finally, the results obtained by using the proportional (P) controller are shown in Figure 4.38 - Figure 4.40. Due to the absence of the integrator (I), convergence to steady state is faster than that by using the PID controller, but slower than that by using the Nonlinear controller. The convergence speed is the same compared with the quadratic-linear controller. Figure 4.40 shows the control efforts by using the P controller. Similar to using the PID controllers, small rapid oscillations exhibit the control efforts.

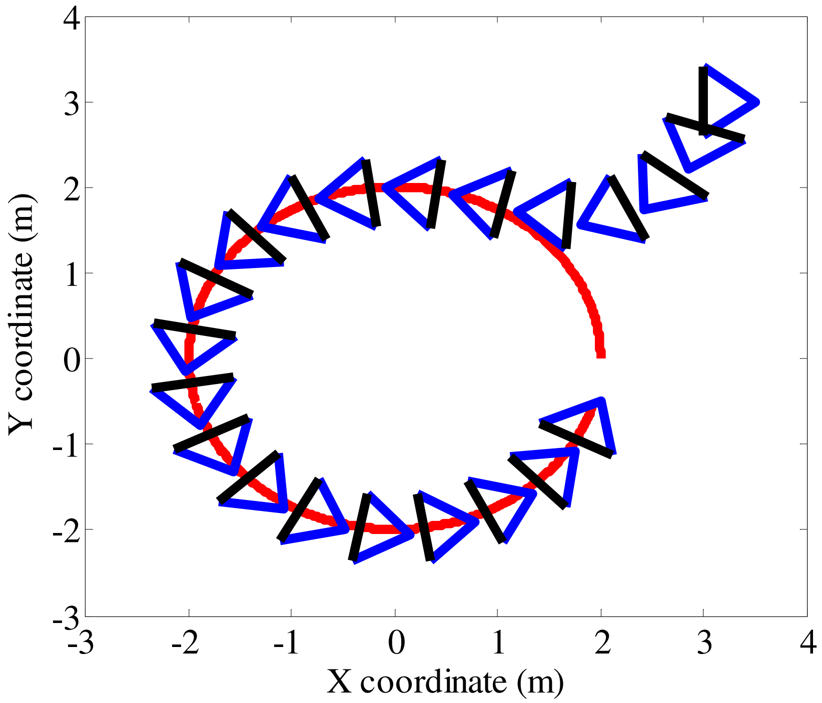


Figure 4.38: Trajectories of the sound source (red curve), and the robot (triangles) obtained by using the P controller.

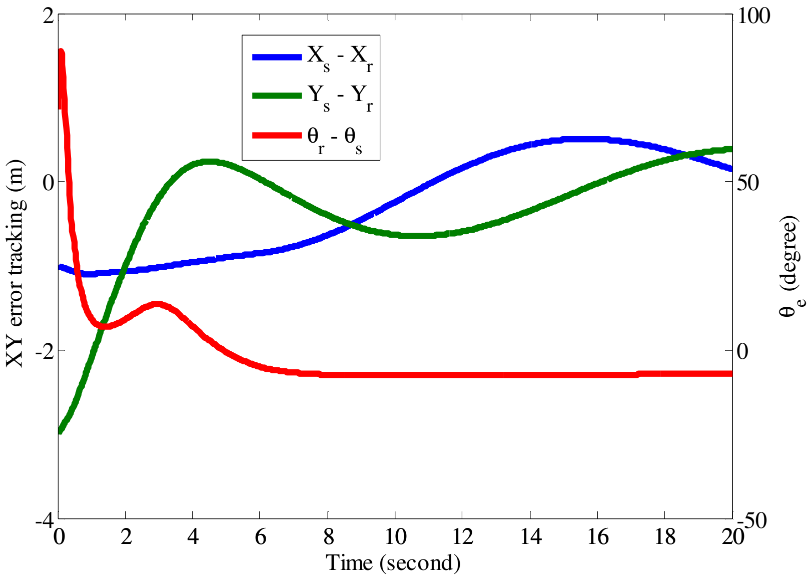


Figure 4.39: Tracking errors by using the P controller.

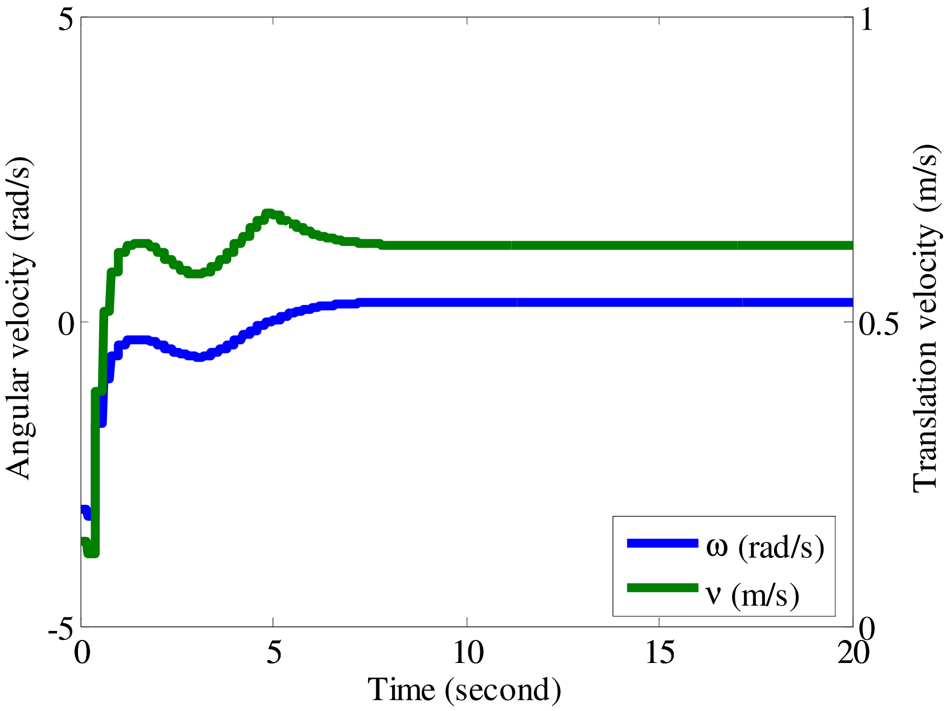


Figure 4.40: Control efforts by using the P controller.

4.4 Summary

In this chapter, different controllers are designed and investigated to control a robot equipped with acoustic sensor for localizing a single stationary sound source and tracking a single moving sound source. Simulations are carried out to compare the performance of these controllers to that of the conventional controllers, namely, Proportional and PID controllers, presented in the literature.

For localization of a single stationary sound source, the robot kinematics were modeled using both the polar and Cartesian coordinates. It is found that the control laws should only be a function of the estimated angle error since it is the only measurable information in the system of interest. In addition, the control laws derived in the polar coordinates were used in the Cartesian coordinate model. In terms of convergence speed of the angle error, the nonlinear controller was the fastest, followed by the quadratic-linear and proportional controllers, and the PID controller was found to be the slowest. In terms of the convergence speed of the X and Y trajectory errors, the nonlinear controller was found to be not only the fastest in driving the angle error to zero but also following the shortest path to arrive the origin.

For tracking moving sound source different controller configurations were investigated. For the control system using the nonlinear and quadratic-linear controllers, a PID controller that controls the robot wheel speeds is added to the control loop and performance is compared to the cases without the robot wheel control. It is found that adding the PID controller increased the control time for both the nonlinear and quadratic-linear controllers In addition, adding the PID controller also increased the angle error variation for the nonlinear controller due to the time needed for the PID controller to reach the steady state. However, the angle error variations are less for the quadratic-linear controller with the wheel control. In general, these variations can lead to more power consumption and shorter life time of the electrical components of the robot.

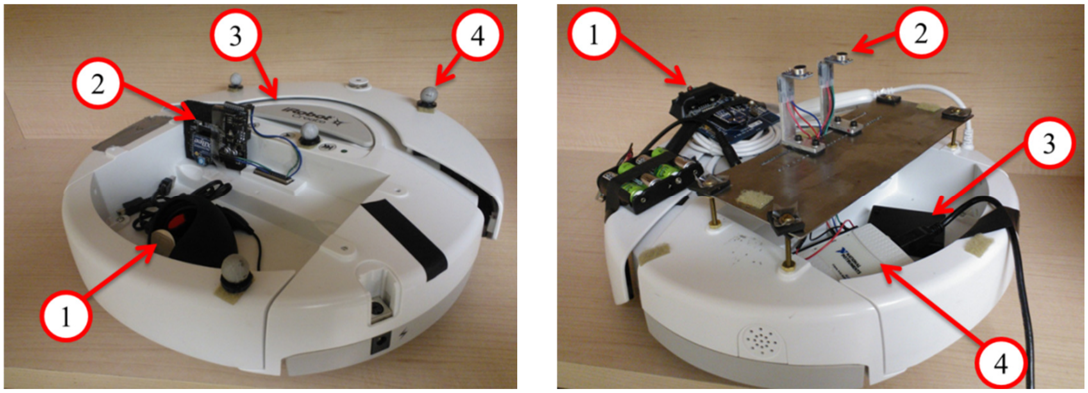
On the other hand, the performance of the nonlinear and quadratic-linear controllers is compared to two conventional controllers (P and PID controllers) presented in the literature for the same problem. When the P or PID controllers are used to control the robot kinematics, it is found that the angle error convergence speed is slower than that using the nonlinear controller. In general, for all the simulated cases, it is found that there is a limitation on the relative speed between the robot and the sound source, which should be properly chosen according to the application.

In the next chapter, the experimental implementation of the models and designed controllers will be carried out, which will help validate the simulation results obtained in the previous chapters.

Chapter 5 Experimental Studies of Robotic Sound Localization and Tracking Using Mobile Platforms

5.1 Experimental Setup

The objective of this chapter is to study robotic sound localization and tracking with two mobile platforms (iRobot - Create). One of the platforms is equipped with a conventional two-microphone array and the other one holds a portable speaker. The two robotic platforms are shown in Figure 5.1.



(a) **(b)**

Figure 5.1: a) Robotic platform serving as the moving source with the following components: 1) portable speaker (X-mini MAX II Capsule Speakers), 2) Xbee module for wireless communication, 3) iRobot – Create robot, and 4) markers. b) Tracking robot with the following components: 1) Xbee module for wireless communication, 2) electret microphone array separated by 5 cm with the conditioning circuit, 3) electret microphones power supply (battery set), and 4) data acquisition (DAQ) board (NI USB-6009).

The moving source platform has three main components: the iRobot-Create platform, the wireless communication module (Xbee), and the portable speaker (X-mini MAX II), as shown in Figure 5.1(a). The tracking platform has four main components: the iRobot-Create platform, the wireless communication module (Xbee), a microphone pair with a conditioning circuit, and a DAQ board (NI USB-6009), as shown in Figure 5.1(b).

The Xbee module in both platforms is used to receive the motion commands from a PC, which is used to define the path of the moving sound platform any feedback control. The path of the moving source can be defined in any arbitrary form. For simplicity in this work, a straight path and rectangular shaped path are chosen. The tracking platform receives the commands after extracting the angle of arrival information of the sound source, which is used in the control laws defined in the previous chapter. A list of commands with a brief description and the corresponding opcodes can are provided in Appendix H (more details can also be found in Ref. [71]). The signals from the microphones are acquired by using the DAQ. The signal processing and decision making is conducted by using a LabView code provided in Appendix H.

The microphone used in the tracking platform are electret microphones [72] with a frequency range of 20 Hz to 20 kHz. The microphone conditioning circuit is a compact fully assembled board from *Adafruit* [73], as shown in Figure 5.2. An op-amp (Maxim MAX4466) is used to amplify the microphone signal, which is designed to have excellent power supply noise rejection [74]. A small trimmer pot is added to adjust the gain (25x to 125x).

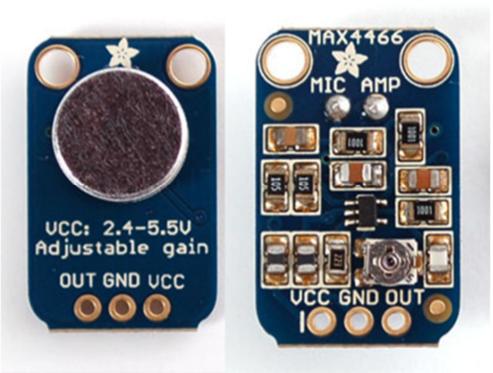


Figure 5.2: Front and back views of the microphone circuit board.

Figure 5.3 shows the experimental arrangement to obtain the microphone signals from two channel of the DAQ. The signals are sampled at a rate of 22 kHz for each channel. The acquired signals are transmitted through a USB cable from the DAQ to a PC, and then processed to obtain the phase difference between the two microphone signals. To determine the phase difference the time delay is first calculated by finding the maximum of the cross-correlation of the two signals, which is converted later to phase difference (PD) using the following relation:

$$PD=2\pi f\cdot TD. \tag{5.1}$$

A second order polynomial is used to perform the curve fitting around the center peak of the cross-correlation to improve the resolution [60].

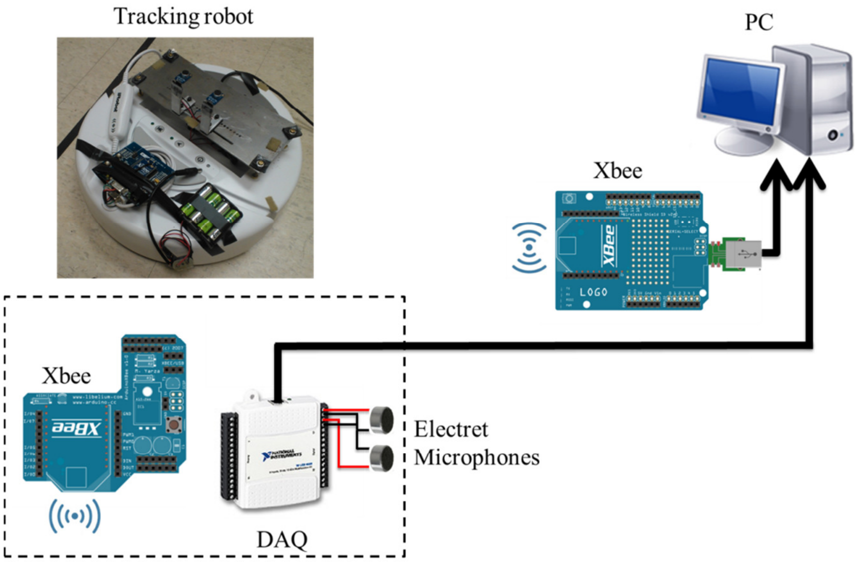


Figure 5.3: Experimental arrangement for data acquisition and wireless communication.

5.2 Results

The experiments were carried out in Prof. Humbert's lab in the Department of Aerospace, which is equipped with the VICON tracking system [75]. The markers shown in Figure 5.1 reflect the red light from the VICON system, which can be used to collect data about the states (position, orientation, translational and angular velocities) of the robot in 3D. The data is processed later and used to evaluate the localization and tracking performance of the tracking robot using the control laws derived in the previous chapter (i.e., nonlinear and quadratic-linear controllers (see Table 4-1)). For the moving sound source experiment, the source platform is controlled to move in a straight path and a square path. In all experiments, the sound source platform and the tracking robot are represented as green (referred to as (2)) and blue circles (referred to as (1)), respectively. The dashed line represents the trajectory of each robot during the localization and tracking missions. The gains are chosen to be K_{xe} = 0.10 (m/s), and $K_{\omega e}$ = 0.766 (rad/s), which represent the maximum translational and angular velocities that the robot can achieve. The tracking robot can be stopped manually by sending the stop command via the Xbee module.

5.2.1 Localization of a Stationary Sound Source

In this experiment, the robot holding the portable speaker was fixed in place. The speaker is located in the cargo bay, which is facing up to receive the sound waves. The nonlinear and quadratic-linear controllers were tested at an initial sound incident angle of 45^o. Snapshots of the localization of the stationary sound process are shown in Appendix I.

5.2.1.1 Nonlinear Controller

The trajectories of the robots in the homing sound source with initial angle of 45^o are shown in Figure 5.4. Figure 5.5 shows the history of the measured translational velocities at the initial sound incident angle. The history of the measured angular velocities at the initial incident angle is shown in Figure 5.6. The trajectory errors X_e, Y_e, Ψ_e are shown in Figure 5.7. It can be observed that the tracking robot hits the target eventually. As expected, the tracking robot needs more time to hit the target when the initial angle error increases. The trajectory errors in XY coordinate decreases as the tracker moves closer to the source. However, they will not converge to zero since the VICON system considers the origin of each robot at its center.

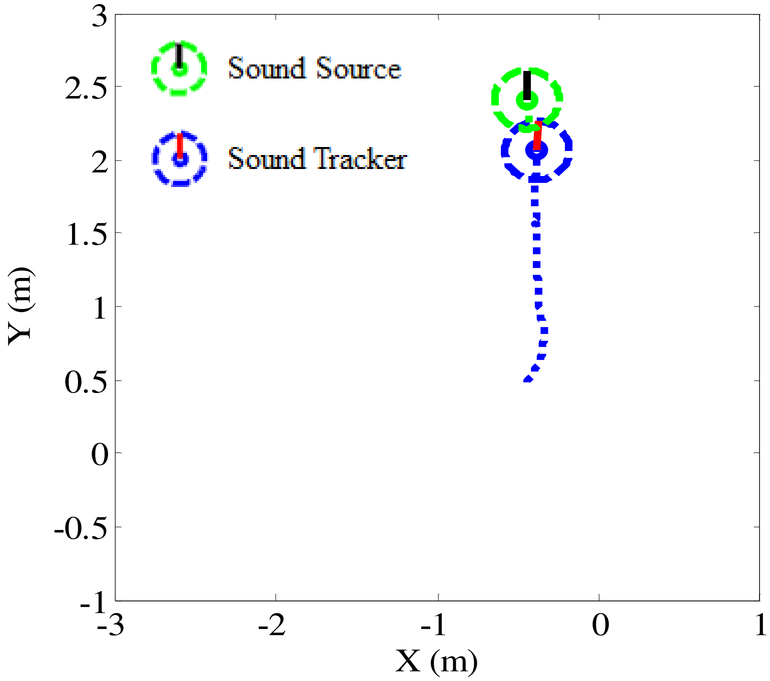


Figure 5.4: Trajectory of the tracking robot homing to a stationary source using the nonlinear controller with an initial angle error of 45°.

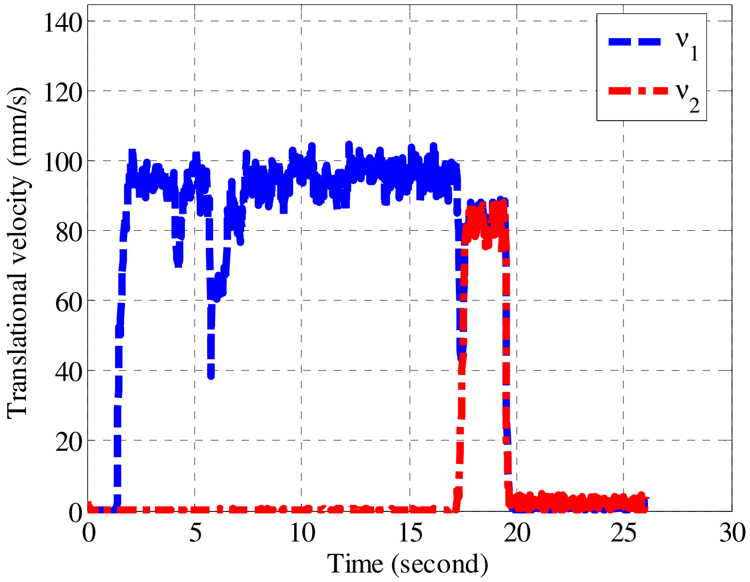


Figure 5.5: History of translational velocity using the nonlinear controller with initial angle error of 45°.

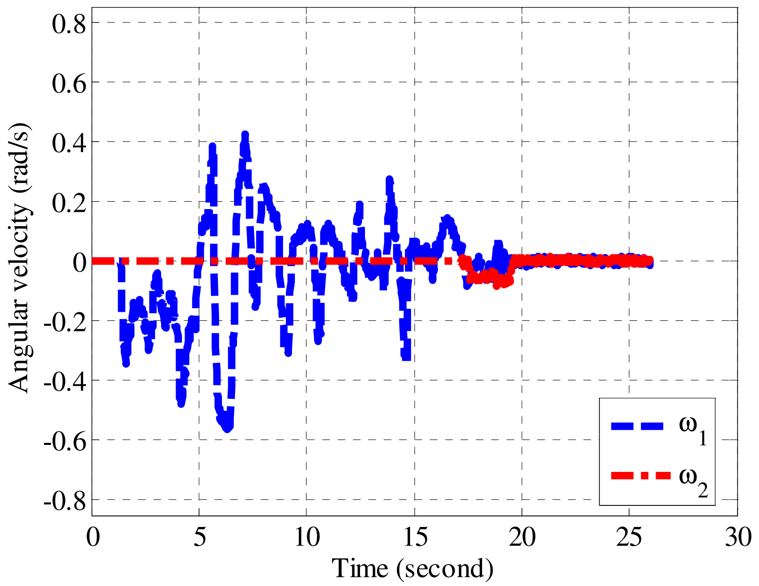


Figure 5.6: History of angular velocity by using nonlinear controller with an initial angle of 45^o.

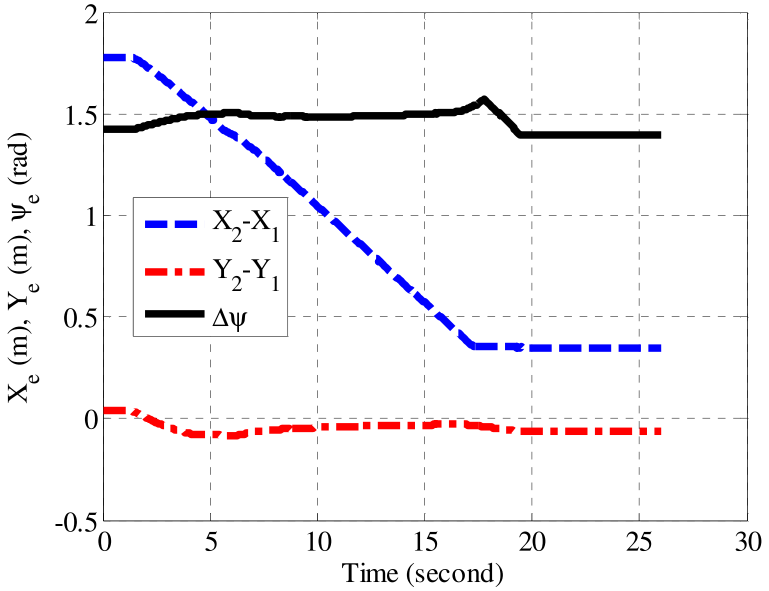


Figure 5.7: Tracking errors by using the nonlinear controller with an initial angle error of 45^o.

5.2.1.2 Quadratic-linear Controller

In Figure 5.8, the trajectory of the tracking robot in the homing to the stationary source by using the quadratic-linear controller is shown. Figure 5.9 and **Error! Reference source not found.** show the histories of the measured translational velocities of the tracking robot at initial angle of 45°. In Figure 5.10 and **Error! Reference source not found.**, the histories of the measured angular velocities are shown. The tracking errors X_e, Y_e, Ψ_e are shown in Figure 5.11 and **Error! Reference source not found.**. Similarly, the trajectory errors in XY coordinate decrease as the tracker gets closer to the source, but will not reach zero.

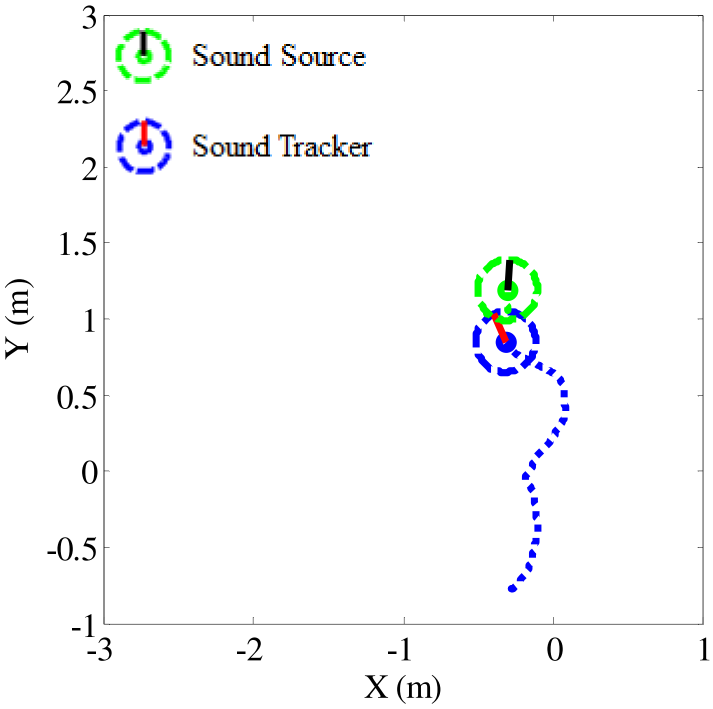


Figure 5.8: Trajectory of the tracking robot homing to a stationary source using the quadratic-linear controller with an initial angle error of 45°.

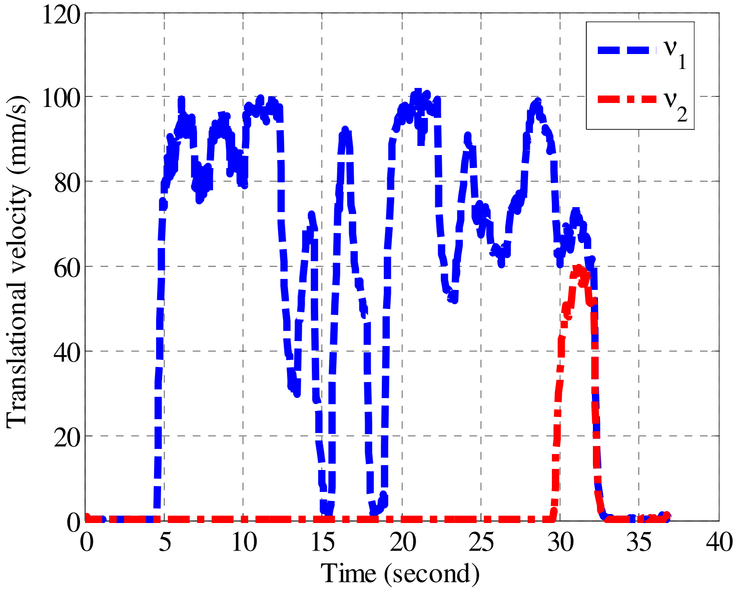


Figure 5.9: History of translational velocity using the quadratic-linear controller with initial angle error of 45^o.

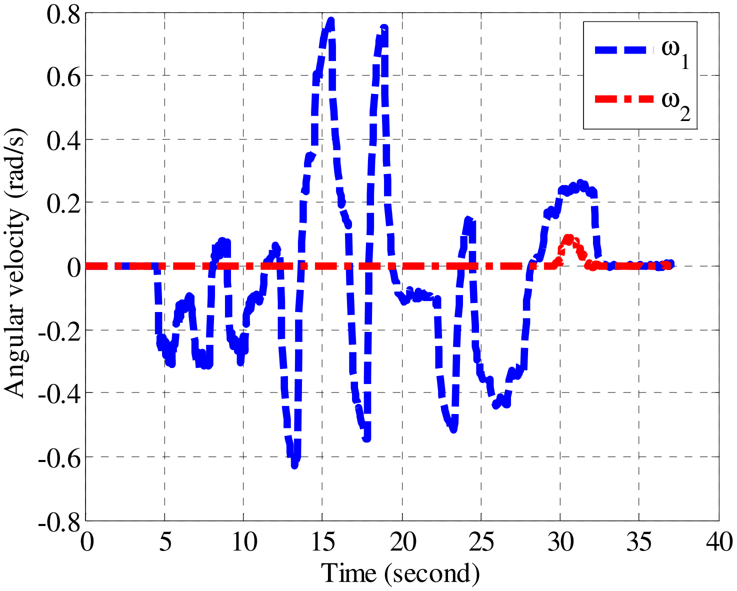


Figure 5.10: History of angular velocity by using quadratic-linear controller with an initial angle of 45^o.

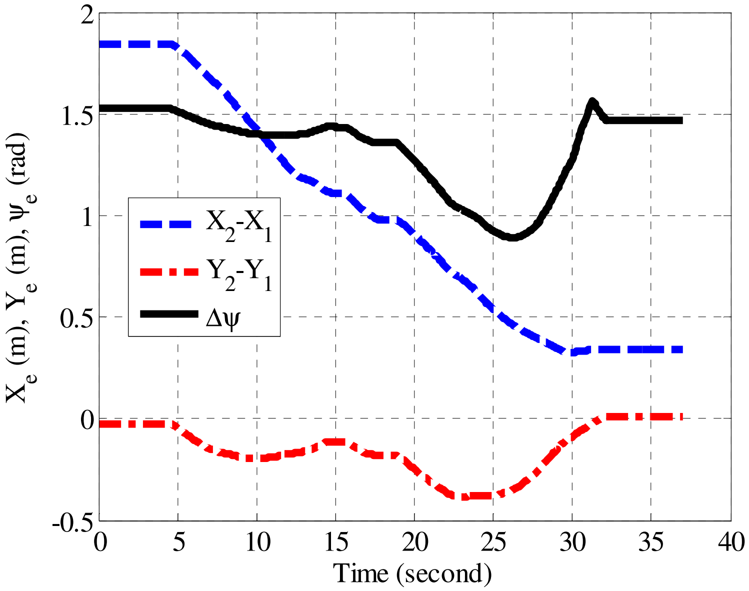


Figure 5.11: Tracking errors by using the quadratic-linear controller with an initial angle error of 45°.

5.2.2 Tracking of a Moving Sound Source

In the experiment, two paths were pre-defined for the moving sound source platform, a straight path and a square shaped path with sharp turns (90°) at the corners. Tracking of the moving source was carried out by using the nonlinear and quadratic-linear controllers for comparison. Snapshots of the tracking of the moving sound are shown in Appendix J.

5.2.2.1 Nonlinear Controller

Figure 5.12 shows the trajectory of the robot in tracking of a source moving in a straight path using the nonlinear controller. The histories of the measured translation and angular velocities of the source and tracker are shown in Figure 5.13 and Figure 5.14. The translational velocity of the source is kept constant most of the time, and the fluctuations at the end of the tracking process are due to the tracker hitting the target. After hitting, both robots were stopped manually. The tracking errors X_e, Y_e, Ψ_e are shown in Figure 5.15.

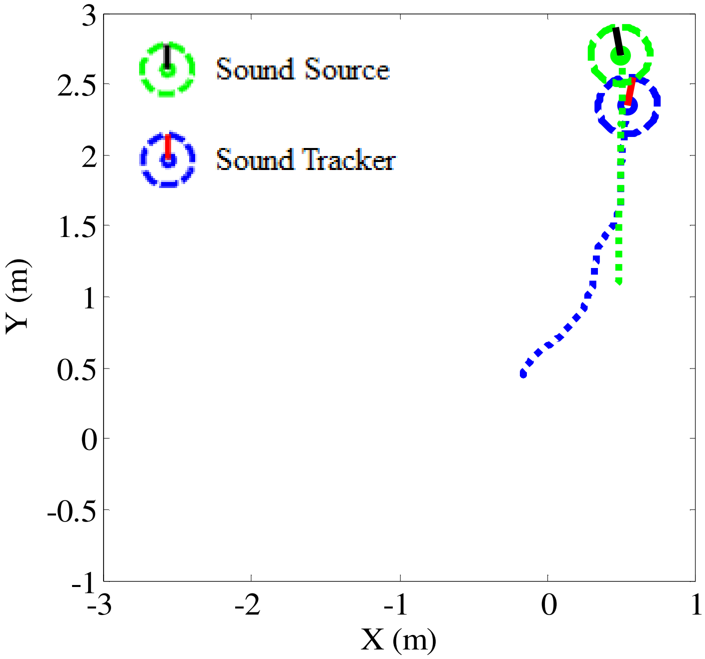


Figure 5.12: Trajectory of the robot tracking a source with a straight path by using the nonlinear controller.

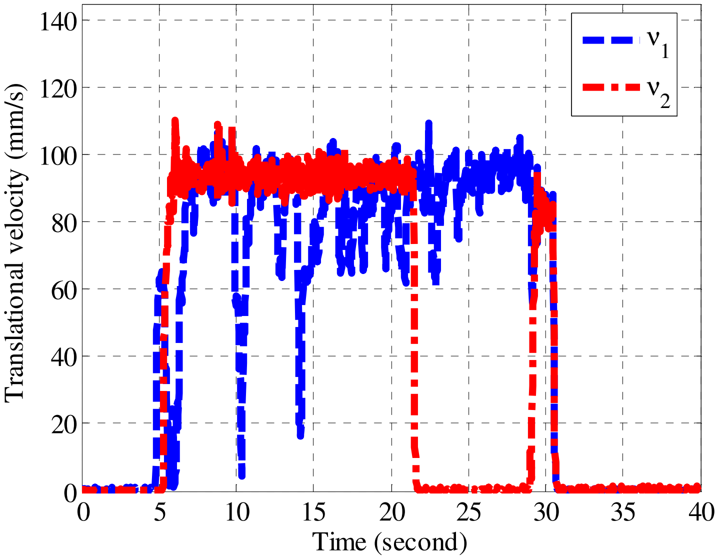


Figure 5.13: History of translational velocity for robot tracking a source with a straight path by using the nonlinear controller.

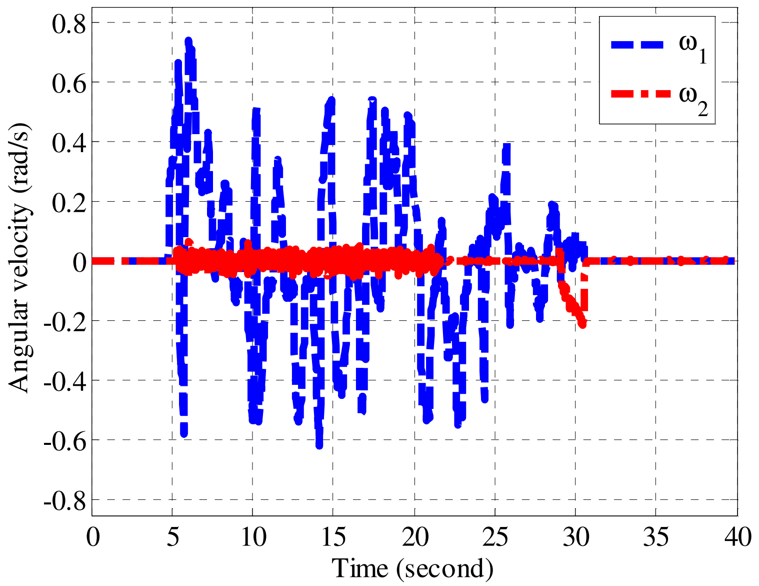


Figure 5.14: Angular velocity history for robot tracking a source with a straight path by using the nonlinear controller

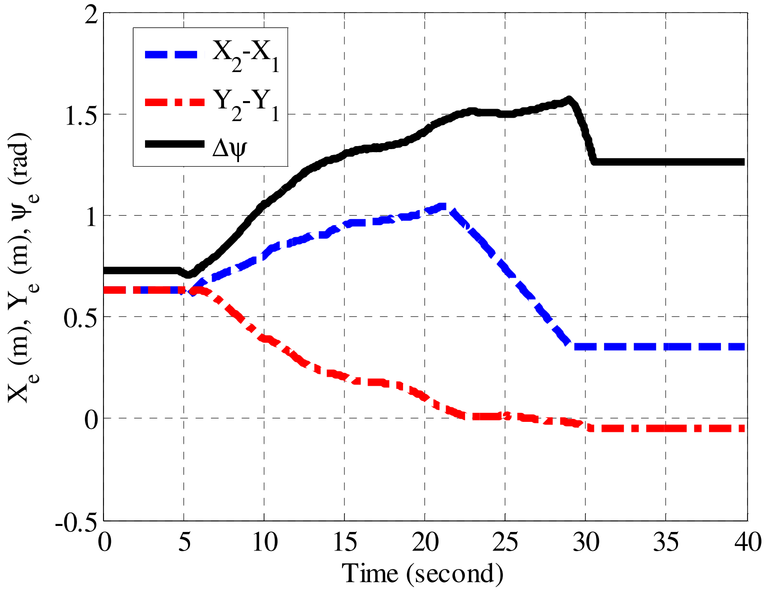


Figure 5.15: Tracking errors for robot tracking a source with a straight path by using the nonlinear controller.

Experiments were also carried out for tracking of a source moving along a square path. In Figure 5.12, the trajectory of the robot tracking the source moving along a square path by using the nonlinear controller is plotted. Note that the tracker robot hit the source at the lower right corner resulting in a drift in the source motion. The histories of the measured translational and angular velocities of the source and tracking robots are shown in Figure 5.13 and Figure 5.14. The translational speed of the source was kept constant. The square waveform signal indicates that the source has straight motion followed by turning at edges. The spikes present in Figure 5.17 are due to the discontinuous velocity at the edges resulted the phase wrapping. The tracker in this experiment was able to track the moving sound source all the time and eventually hit the target. Figure 5.15 shows the tracking errors X_e, Y_e, Ψ_e. The sign of Ψ_e flip at about 50 seconds, which is due to the phase wrapping when the angle gets larger than 180°. If the source speed is reduced, the tracker will be able to hit the target before the source finishes moving one cycle along the square path. The speed of the tracking robot can be adjusted according to the requirements of the application and the goal to be achieved.

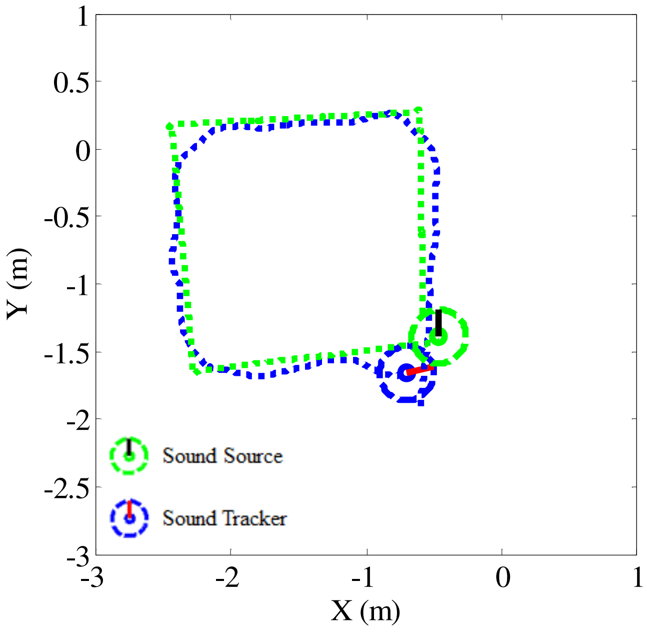


Figure Trajectory of the robot tracking a source with a square path by using the nonlinear controller.

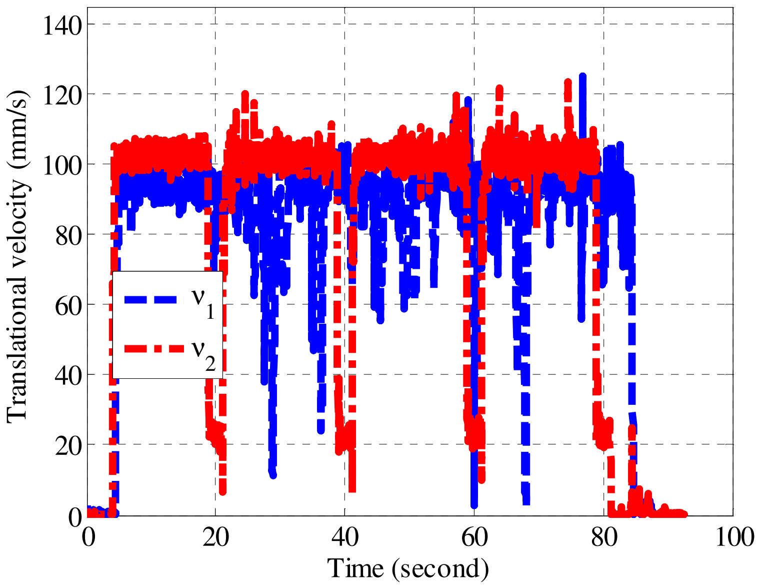


Figure 5.16: Translational velocity history of robot tracking a source with a square path by using the nonlinear controller.

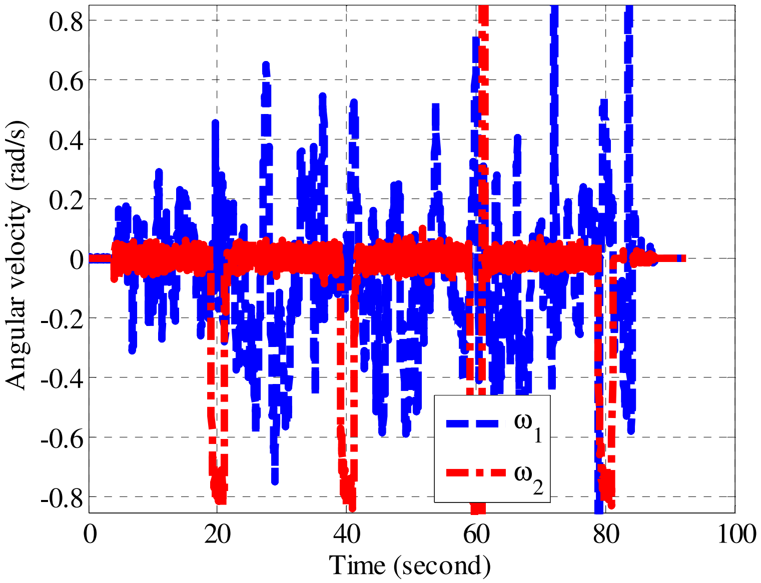


Figure 5.17: Angular velocity history of robot tracking a source with a square path by using the nonlinear controller.

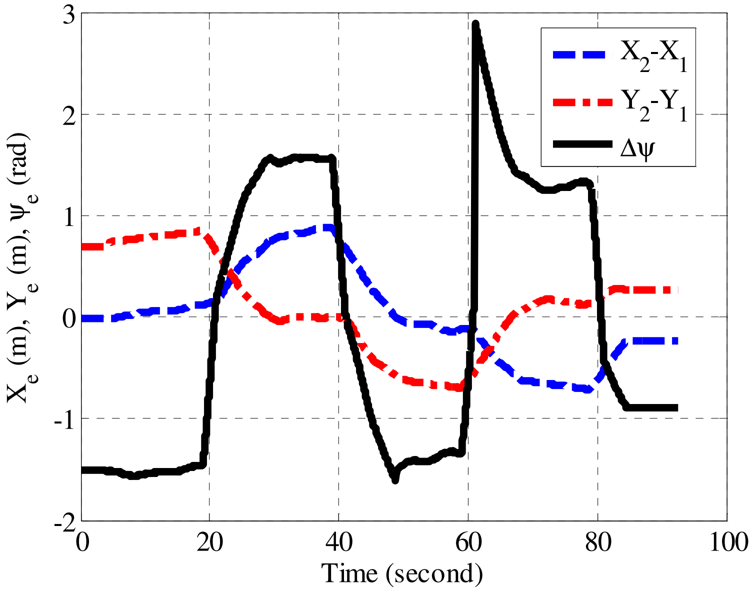


Figure 5.18: Tracking errors for robot tracking a source with a square path by using the nonlinear controller.

5.2.2.2 Quadratic-linear Controller

Using the quadratic-linear controller, the trajectory of the robot tracking a source moving in a straight path is obtained, as shown in Figure 5.19. The histories of the measured translational and angular velocities of the source and tracking robot are shown in Figure 5.20 and Figure 5.21, respectively. Again the translational speed of the source was kept constant during tracking. Compared with the same experiment by using the nonlinear controller, the control efforts here are less and the tracking is achieved in less time, which leads to less power consumption. Also note that the tracker was able to track the sound source and eventually hit the source. In Figure 5.22, the tracking errors X_e, Y_e, Ψ_e are plotted. The X and Y tracking errors do not go to zero due to the robot centers in the VICON system.

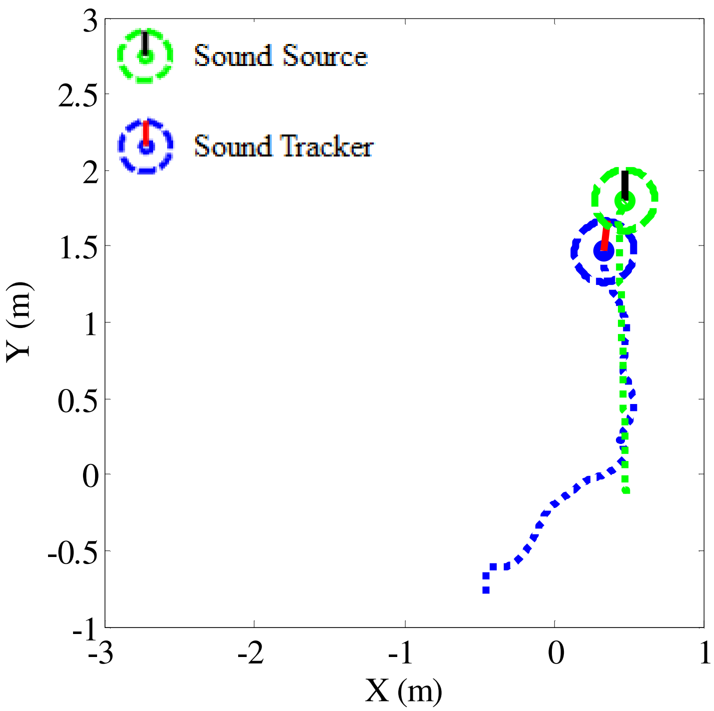


Figure 5.19: Trajectory of the robot tracking a source with a straight path by using the quadratic-linear controller.

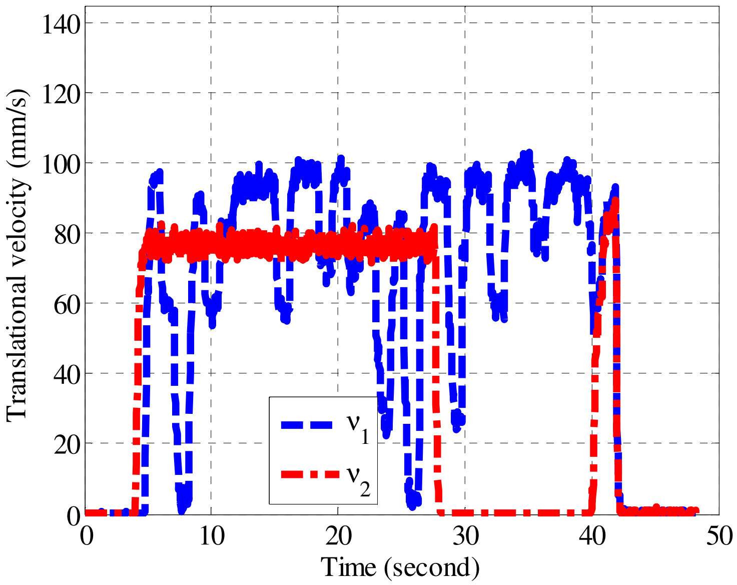


Figure 5.20: Translational velocity history of robot tracking a source with a straight path by using the quadratic-linear controller.

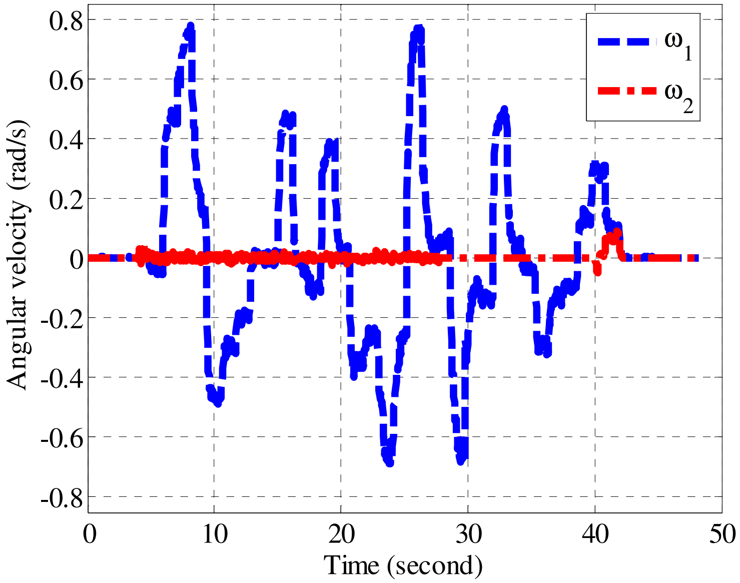


Figure 5.21: Angular velocity history of robot tracking a source with a straight path by using the quadratic-linear controller.

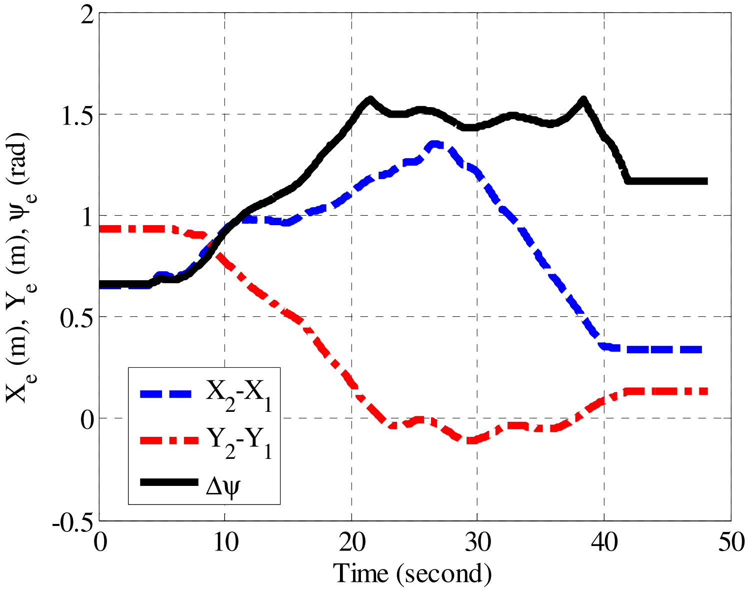


Figure 5.22: Tracking errors for robot tracking a source with a straight path by using the quadratic-linear controller.

For tracking moving source with square path by using the quadratic-linear controller, the trajectory of the tracking robot is shown in Figure 5.23. The histories of the measured translational and angular velocities of the source and tracker are provided in Figure 5.24 and Figure 5.25, respectively. Figure 5.26 shows the tracking errors X_e, Y_e, Ψ_e. These results are similar to those obtained with the nonlinear controller for tracking of a moving source with a square path.

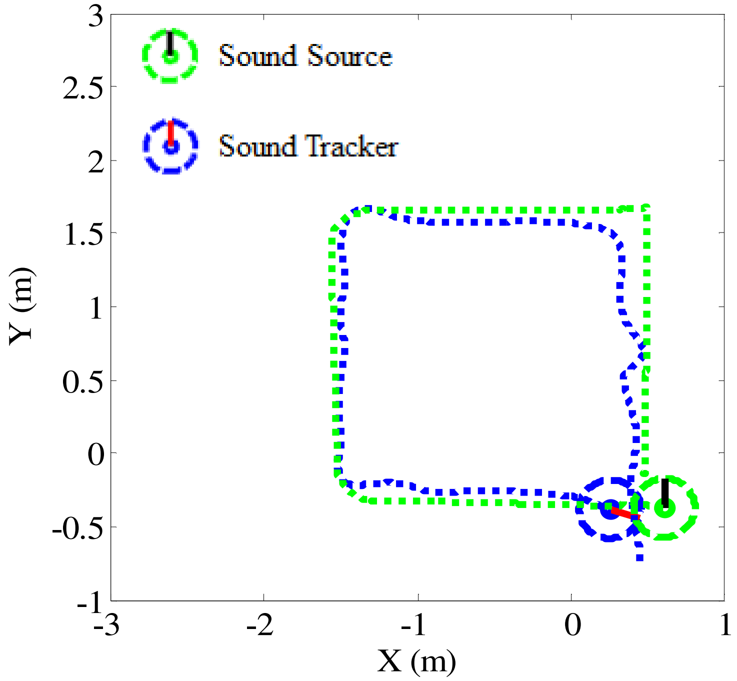


Figure 5.23: Trajectory of the robot tracking a source with a square path by using the quadratic-linear controller.

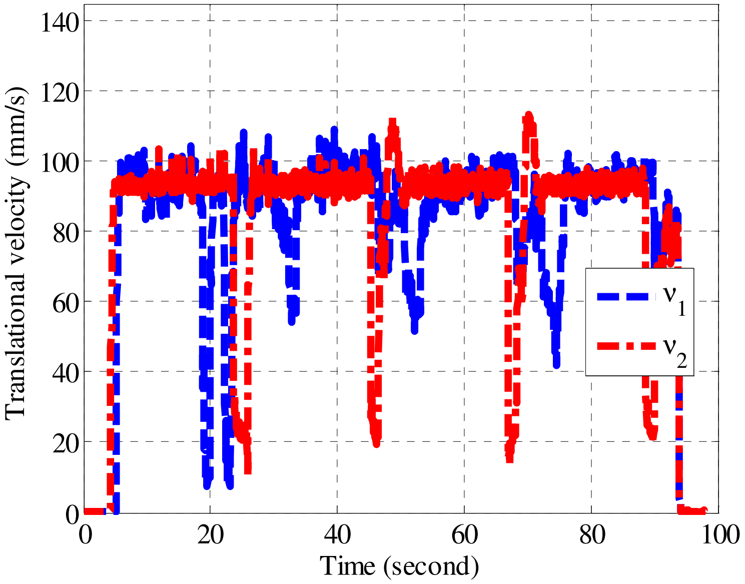


Figure 5.24: Translational velocity history of robot tracking a source with a square path by using the quadratic-linear controller.

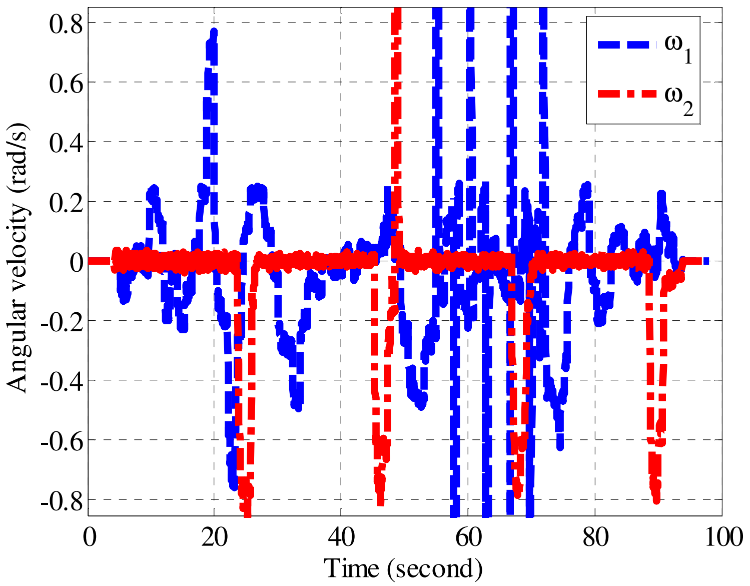


Figure 5.25: Angular velocity history of robot tracking a source with a square path by using the quadratic-linear controller.

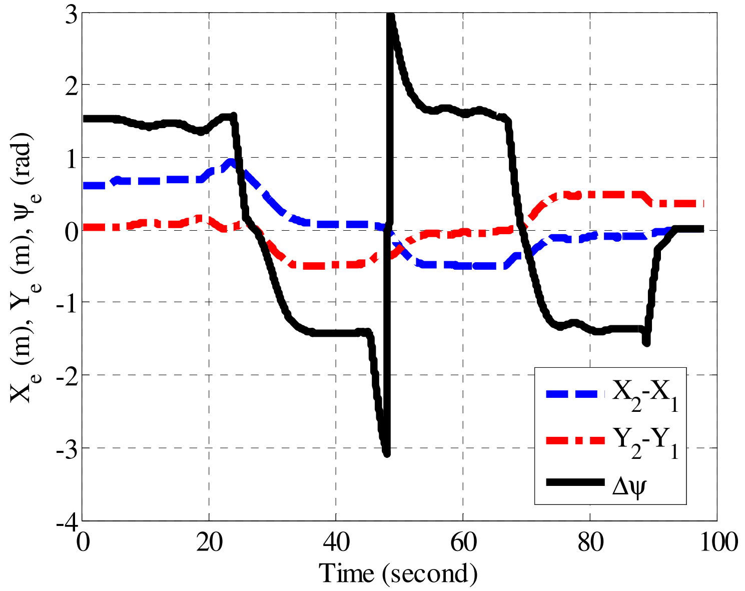


Figure 5.26: Tracking errors for robot tracking a source with a square path by using the quadratic-linear controller.

5.3 Summary

In this chapter, implementation of nonlinear and quadratic-linear controllers on a mobile robotic platform for localization of a single stationary and tracking of a moving sound source was carried out. The robot was equipped with a two microphone array. A LabView block diagram was constructed, which can be used to do the following: i) acquire the microphone signals, ii) calculate the phase difference between the signals for estimation of the angle of arrival of the sound source, and iii) control the tracking robot based on the estimated incident angle.

For localizing a stationary source (i.e., the homing problem), the source robot was fixed in an initial position and the tracking robot was oriented initially at 45^o and 90^o. By using both the nonlinear and quadratic-linear controllers developed in the previous chapter, the robot was able to localize the source by following a nonlinear path.

For tracking of a moving source, the source was controlled to move in a straight path and a square shaped path. In the tracking experiments, by using both the nonlinear and quadratic-linear controllers, the tracking robot was able to track the source and hit the target. For the square path experiment, the trajectory of the robot using the quadratic-linear controller showed less oscillation than that using the nonlinear controller. However, this may simply due to the testing conditions and angle measurement errors during the experiments.

The purpose of this chapter is to experimentally implement the different types of controllers and to validate the ability of using these controllers in robotic sound localization and tracking. Simulations in the previous chapter showed that using the quadratic-linear controller the control efforts are less than that using the nonlinear controller. However, in the experiments, since the acoustic measurements can be greatly affected by the surrounding environment, the testing conditions can hardly be kept the same to ensure a fair comparison between different experiments. It is suggested that future experiments should be conducted in quite room by using mobile platforms with less vibrations and noise, which can help improve the accuracy on the sound incident angle estimation.

Chapter 6 Conclusions and Future Work

6.1 Summary of the Dissertation Work

In this dissertation, theoretical and experimental investigations into sound source localization and tracking using bio-inspired acoustic sensor have been carried out. The sensor was previously developed, which was inspired by the tiny ear of the fly *Ormia*. The two ear membranes are mechanically coupled, which helps significantly amplify the directional cues that can be obtained by the eardrums. In this work, to achieve an enhanced understanding of the performance of the fly-ear inspired sensor for sound source localization, the performance of the fly ear sensor has been compared to a conventional two-microphone array by evaluating the Cramer Rao Lower Bound (CRLB). It is found that the fly-ear inspired sensor configuration can achieve a much lower (100 times lower) CRLB compared to the conventional two-microphone array. In addition, since it is found that an estimator that can achieve the CRLB does not exist, the variance of estimating the sound incident angle by using the mIPD information is evaluated through numerical. It is found that by using the fly-ear inspired sensor a lower variance in estimating the incident angle can be achieved, compared with that using a conventional microphone pair. These results indicate that the mechanical coupling not only helps amplify the directional cues, but in addition, it helps the fly to estimate the sound direction with a much improved accuracy.

Furthermore, in order to better use the fly-ear inspired sensors, since the directional cues as a function of sound source incident angle is nonlinear and frequency dependent, two novel approaches have been investigated to estimate the sound incident angle from directional cues of the fly-ear inspired sensor. The first one is an optimization approach based on model free gradient descent method. The directional cue from the sensor, namely the mIPD, is used to define an objective function and the sound incident angle is determined from minimizing the objective function in an iterative process. The main drawback of this approach is that a number of iterations are needed to pinpoint the sound source.

In another approach, a fuzzy logic model is constructed to map the directional cue at the different frequencies to the incident angle. Two different models have been constructed. One model makes use of the data obtained from the two-degree-of-freedom (2DOF) model as the training data set and while the other model utilizes the experimental data obtained from the sensor as the training data set. It has been found that the model trained by using the experimental data was able to model the senor more accurately. Later, in the experiment, this model has been used along with the sensor for localization of a stationary sound and tracking of a moving sound source. For localization of a stationary source, the experimental results have been compared with those obtained by using the least square error (LSE) method and the saturation function method in terms of several defined performance metrics, including settling time, overshoot, error band, and root mean square error (RMSE). The experiments lead to the conclusion that the fuzzy model has a better performance than the saturation function method in terms of the settling time, overshoot, and the RMSE. Compared to the LSE method, the fuzzy model exhibits less performance fluctuations due to the fact that fuzzy model takes into consideration the uncertainties in the measurements. In addition, LSE method requires searching a 2D array for determination of the incident angle. This means that the accuracy of the LSE method depends on the array size; the larger the array the better the accuracy. However, a larger array means a longer searching time, which can be a problem for a mobile platform with limited memory and computational capabilities. For tracking of a moving source, the experimental results have been compared with those obtained by using the least square error (LSE) method and the saturation function method at different speed ratios. Since the performance in this experiment differs from that for the localization of a stationary sound, the relative and absolute RMSEs are used as the performance metrics for comparison. It is found that using the fuzzy model, a smaller RMSE can be obtained, compared with that obtainable with the LSE and saturation function methods. The fuzzy model is also shown to overcome the performance fluctuations that the LSE method suffers from due to its capability of addressing uncertainties. In addition, the fuzzy model also overcomes the linear range limitation in the saturation function method.

In addition, two novel controllers, the nonlinear and quadratic-linear controllers have been designed and implemented to control a mobile robot equipped with acoustic sensors for localization of a stationary source as well as tracking a moving source. The nonlinear controller is designed based on the Lyapunov approach with only the information from the acoustic sensor for both localization and tracking. The quadratic-linear controller, motivated by the nonlinear controller nature and bounded values, is designed by inspection. In simulations, both controllers have been demonstrated to have good performance in controlling the robot kinematics for sound source localization and tracking. The simulations results have also shown that adding a PID controller to control the robot wheels, for both controllers, will result in delays in the homing and tracking the sound due to the speed limit of the PID controller loop. Further, the performance of sound localization and tracking using these two controllers has been compared to that using a PID controller. The PID controller has been shown to exhibit more oscillations in the control efforts, which may result in high power consumption for the robot platform, especially for miniature robots. Experimental investigation of nonlinear and quadratic-linear controllers have also been conducted for localizing and tracking a sound source by using two iRobot platforms (a source platform and a tracking platform). Successful localization and tracking have been demonstrated with both controllers.

6.2 Summary of Contributions

The original contributions of this dissertation work can be summarized as follows.

1. Enhanced understanding of the influence of the mechanical coupling on the fly-ear sensor performance for sound localization has been achieved based on investigating the Cramer Rao lower bound and estimation of the variance of the sound incident angle.
2. Two novel approaches have been developed to determine the sound incident angle from the output (i.e., interaural phase difference) of the fly-ear inspired sensor. The first one is an iterative optimization approach based on the model-free gradient descent method. Two dimensional localization of a sound source without ambiguity has been demonstrated. The second approach is based on

constructing a fuzzy logic model of the sensor. Simulations and experiments showed the capability to localize and track a single sound source.

3. Nonlinear and quadratic-linear controllers have been developed based on the Lyapunov approach to control the kinematics of a robotic platform. The controllers that make use of only the estimated source angle information to achieve homing and tracking of a single stationary or moving sound source, have been demonstrated in simulations to have superior performance than a PID controller.
4. The nonlinear controller has successfully been implemented with a mobile platform for sound source localization and tracking.

6.3 Future Work

6.3.1 Design of a Small Size, Standalone, and High Speed Data Acquisition Board

In this work, since the signal from fly-ear inspired sensor was detected by using a large scale optical system, implementation of the sensor on the mobile platform cannot be performed. In the future work, a small size, wireless data acquisition and signal processing device for the fly-ear inspired sensor should be developed, which can be integrated with a mobile robot for sound source localization. Some efforts have been undertaken to develop a small optical wireless sensor network node with a built-in optical system to detect the diaphragm deflections of the fly-ear inspired sensor, as shown in Figure 6.1.

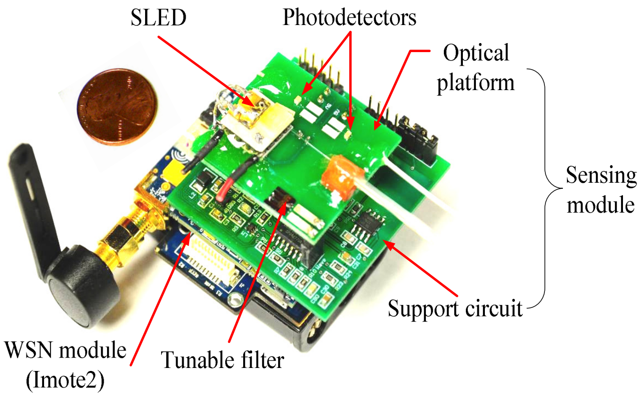


Figure 6.1: Fully integrated wireless sensor network node [76].

6.3.2 Robotic Sound Source Homing and Tracking Using Fuzzy Logic

Control of a robot equipped with the fly-ear inspired acoustic sensor using fuzzy logic has not been investigated yet. The fuzzy models presented in this dissertation work can be extended to a multi input multi output (MIMO) instead of multi input single output (MISO). Two separate models can be used in a cascaded system: one to estimate the sound incident angle and the other to decide the translational and angular velocities of the robot. In this way, the first fuzzy model will take care of the uncertainties of the measurements and the second model will consider the uncertainties of the estimated incident angle from the previous model. Furthermore, a Kalman filter will be used to reduce the incorrect measurements effect on the homing and tracking performance.

6.3.3 3D Robotic Sound Source Homing and Tracking

As an extension of this dissertation work, using a fly ear inspired sensor with three coupled diaphragms reviewed in Section 1.2.3, along with a quad rotor, 3D robotic sound source localization and tracking can be studied. The quad rotors equipped with this sensor can also be used to navigation and exploration, for example, explore hazardous environment and perform search and rescue of survivals. Fuzzy logic can also be used to control the quad rotor with the cascaded system explained in the previous suggested future work. The Kalman filter will be investigated to find out whether it can be used to address the issues associated with the measurement errors and help improve the localization and tracking performance.

Appendix A   Cramer-Rao Lower Bound (CRLB)

In this appendix, the background of the CRLB is explained briefly (form more information refer to [48]). CRLB is a lower bound on the variance of any unbiased estimator that allows to confirm that an estimator is the minimum variance unbiased (MVU) estimator if the estimator attains the bound for all the unknown parameters values. Furthermore, any unbiased estimator cannot be lower than this bound. Accordingly, the CRLB is the criteria to which the performance of any unbiased estimator can be compared. CRLB is usually used in signal processing applications such as sonar, radar, robotics, econometrics, spectrometry, and speech.

CRLB can be calculated by getting the inverse of the *Fisher Information Matrix* (FIM) which is derived from the *Probability Density Function* (PDF) in which the unknown parameters are embedded. The more the PDF depends on the unknown parameter the more accurate is the parameter estimation. To have more insight on what this does mean, consider observing a simple sample as follows:

$$x[0] = A + w[0] \tag{A.1}$$

where $w[0]$ is Gaussian noise with zero mean and σ^2 variance, and it is required to estimate the parameter A. A good unbiased estimator for this case is $\hat{A} = x[0]$ with a σ^2 variance. The estimation accuracy increases as the variance decreases. The PDF of this example is defined as

$$p(x; A) = \frac{1}{\sqrt{2\pi\sigma^2}}.\exp[-\frac{1}{2\sigma^2}(x[0] - A)^2]. \tag{A.2}$$

If this PDF is viewed as a function of the unknown parameter it can be called the likelihood function. The sharpness of this function indicates how accurate the parameter can be estimated. The sharpness is measured by the negative of the second derivative of the natural logarithm of the likelihood function which is also called the curvature of the likelihood function at the peak. Taking the natural logarithm of the PDF, gives

$$\ln p(x[0];A) = \text{-}\ln\sqrt{2\pi\sigma^2} - \frac{1}{2\sigma^2}(x[0]-A)^2 . \tag{A.3}$$

Taking the first derivative of Eq. (A.3), gives

$$\frac{\partial \ln p(x[0];A)}{\partial A} = \frac{1}{\sigma^2}(x[0]-A) . \tag{A.4}$$

Then taking the negative of the second derivative of Eq. (A.4), leads to

$$-\frac{\partial^2 \ln p(x[0];A)}{\partial A^2} = \frac{1}{\sigma^2} . \tag{A.5}$$

According to Eq. (A.5), the curvature decreases as σ^2 increases. Knowing that the estimator $\hat{A}$ = $x[0]$ has a variance σ^2, then the variance of estimating the unknown parameter A is

$$\mathrm{var}(\widehat{A}) = \frac{1}{-\frac{\partial^2 \ln p(x[0];A)}{\partial A^2}} , \tag{A.6}$$

where the variance of the estimator increases as the curvature decreases.

A better representation of the curvature is the average curvature of the log-likelihood function. Mathematically, the average can be taken as the expectation of the negative of the second derivative of the log-likelihood function which result in a function with the unknown parameter only. From this explanation the CRLB theorem for scalar and vector parameters are stated as follows:

(i) Cramer-Rao lower bound Theorem - Scalar parameter

It is assumed that the PDF p(x; θ) satisfies the “regularity” condition

$$E\left[\frac{\partial}{\partial\theta}\ln\, p(x;\theta)\right]=0 \qquad \textit{for all } \theta \tag{A.7}$$

where the expectation is taken with respect to *p*(*x; θ*). Then, the variance of any unbiased estimator $\hat{\theta}$ must satisfy

$$\operatorname{var}(\hat{\theta}) \geq \frac{1}{-E\left[\frac{\partial^2}{\partial\theta^2}\ln\, p(x;\theta)\right]} \tag{A.8}$$

where the derivative is evaluated at the true value of θ and the expectation is taken with respect to *p*(*x, θ*). Furthermore, an unbiased estimator can be found that attains the bound for all θ if and only if

$$\frac{\partial}{\partial\theta}\ln\, p(x;\theta)=I(\theta)\times(g(x)-\theta) \tag{A.9}$$

for some functions *g*(.) and *I* (.). The estimator, which is the MVU estimator, is $\hat{\theta}=g(x)$, and the minimum variance 1/ *I*(θ).

(ii) Cramer-Rao lower bound Theorem - Vector parameter

It is assumed that the PDF p(x; θ) satisfies the “regularity” condition

$$E\left[\frac{\partial}{\partial\theta}\ln\, p(x;\theta)\right]=0 \qquad \textit{for all } \theta \tag{A.10}$$

where the expectation is taken with respect to *p*(*x; θ*). Then, the covariance matrix of any unbiased estimator $\hat{\theta}$ satisfies

$$C_{\hat{\theta}} - I^{-1}(\theta) \geq 0 \tag{A.11}$$

where ≥ 0 is interpreted as meaning that the matrix is positive semi definite. The Fisher information matrix I(θ) is given as

$$[I(\theta)]_{ij} = -E\left[\frac{\partial^2}{\partial\theta_i \partial\theta_j} \ln p(x;\theta)\right] \tag{A.12}$$

where the derivatives are evaluated at the true value of θ and the expectation is taken with respect to p(x, θ). Furthermore, an unbiased estimator can be found that attains the bound in that $C_{\hat{\theta}} = I^{-1}(\theta)$ if and only if

$$\frac{\partial}{\partial\theta} \ln p(x;\theta) = I(\theta) \times (g(x) - \theta) \tag{A.13}$$

for some *p*-dimensional function *g* and some *p* x *p* matrix *I*. The estimator, which is the MVU estimator, is $\hat{\theta} = g(x)$, and its covariance matrix is $I^{1}(\theta)$.

From the CRLB - vector parameter theorem, the FIM is the term denoted as *I*(*θ*) where the matrix elements are defined by Eq. (A.12).

The procedure of evaluating the CRLB can be summarized as follows:

1. Taking the log of *p*(*x*; *Θ*) to get rid of the exponential part.
2. Taking the partial derivative with respect to the unknown parameters
3. If the observation parameter still appear in the second derivative, the expectation (*E*{.}) of the Eq. is calculated to get rid of it.

4. After evaluating all the FIM elements take the inverse of the matrix.
5. The diagonal of the inverse matrix are the variance of estimating the unknown parameters.

Appendix B Equivalent 2-DOF Model for the Fly-ear

The 2-DOF model was proposed originally by Miles et al., 1995 [37]. As shown in Figure B.1, the two tympana are modeled as two rigid bars connected by a torsion spring k_3. The outer end of each bar is supported by a translational spring k_1 or k_2, which is equivalent to the tympanum stiffness. Dashpots c_1, c_2 and sc_3 are added to account for the damping ratios of the system. All the parameters used to study the fly-ear structure are listed in Table 2-1.

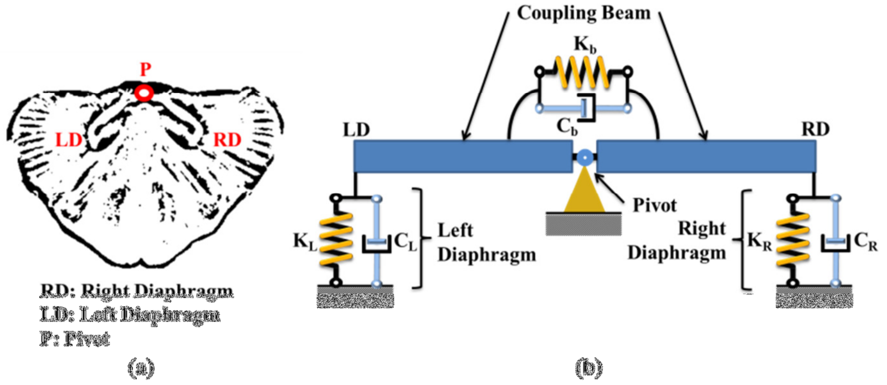


Figure B.1: Equivalent 2-DOF model of the fly-ear [37]

The governing Eqs. of the coupled system can be obtained as follows:

$$\mathbf{M}\begin{Bmatrix}\ddot{x}_1\\ \ddot{x}_2\end{Bmatrix}+\mathbf{C}\begin{Bmatrix}\dot{x}_1\\ \dot{x}_2\end{Bmatrix}+\mathbf{K}\begin{Bmatrix}x_1\\ x_2\end{Bmatrix}=\begin{Bmatrix}F_1\\ F_2\end{Bmatrix}, \qquad \text{(B.1)}$$

where

$$\mathbf{M}=\begin{bmatrix}m & 0\\ 0 & m\end{bmatrix}, \qquad \text{(B.2)}$$

$$\mathbf{C} = \begin{bmatrix} c_1 + c_3 & c_3 \\ c_3 & c_2 + c_3 \end{bmatrix}, \tag{B.3}$$

and
$$\mathbf{K} = \begin{bmatrix} k_1 + k_3 & k_3 \\ k_3 & k_2 + k_3 \end{bmatrix}. \tag{B.4}$$

In the case of free vibration, the natural frequencies and mode shapes of the system can be found as:

$$\omega_1 = \sqrt{k_1 / m}, \quad \omega_2 = \sqrt{(k_1 + 2k_3)/m}, \tag{B.5}$$

$$\mathbf{v}_1 = \begin{Bmatrix} 1 \\ -1 \end{Bmatrix}, \mathbf{v}_2 = \begin{Bmatrix} 1 \\ 1 \end{Bmatrix}. \tag{B.6}$$

The two diaphragms move out of phase in the first mode, and in phase in the second mode.

In terms of modal coordinates, the acting force vector can be written as

$$\begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix} = p_0 s \cdot e^{j\omega t} \begin{Bmatrix} e^{j\omega d \sin\theta/2c} \\ e^{-j\omega d \sin\theta/2c} \end{Bmatrix} = p_0 s \cdot e^{j\omega t} \left(p_1 \mathbf{v}_1 + p_2 \mathbf{v}_2 \right), \tag{B.7}$$

where

$$p_1 = j \sin\left(\omega d \sin\theta / 2c\right), p_2 = \cos\left(\omega d \sin\theta / 2c\right). \tag{B.8}$$

By using modal analysis (Meirovitch, 1996) [77], the modal coordinates of the response can be obtained as

$$u_1 = \frac{p_0 s}{k_1} \frac{p_1}{1 - \Omega^2 + 2j\Omega\xi_1}, u_2 = \frac{p_0 s}{k_1} \frac{p_2}{\eta^2 - \Omega^2 + 2j\eta\Omega\gamma\xi_1}, \tag{B.9}$$

where

$$\Omega = \omega / \omega_1, \eta = \omega_2 / \omega_1 = \sqrt{1 + 2\chi_k}, \chi_k = k_3 / k_1, \tag{B.10}$$

$$\text{and } \xi_1 = c_1/(\omega_1 m), \xi_2 = (c_1 + 2c_3)/(\omega_2 m), \gamma = \xi_2/\xi_1 . \tag{B.11}$$

In terms of physical coordinates of the two DOFs, the responses are

$$\begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{Bmatrix} A_1 \\ A_2 \end{Bmatrix} e^{j\omega t} = \frac{p_0 s}{k_1} \cdot \frac{\cos\phi}{1-\Omega^2 + j2\xi_1\Omega} e^{j\omega t} \begin{Bmatrix} \Gamma + j\tan(\phi/2) \\ \Gamma - j\tan(\phi/2) \end{Bmatrix} , \tag{B.12}$$

where

$$\Gamma = \frac{1-\Omega^2 + j2\xi_1\Omega}{\eta^2 - \Omega^2 + j2\eta\gamma\xi_1\Omega}, \phi = 2\pi\chi_\lambda \sin\theta, \chi_\lambda = \frac{d}{\lambda} . \tag{B.13}$$

The ratio of modal forces p_1 and p_2 defined in (B.8) can be written as

$$p_1 / p_2 = j\tan(\phi/2) . \tag{B.14}$$

Here, Γ represents the relative contribution of the two modes subject to unit modal force, which is dependent on the natural frequency ratio η (or equivalently the stiffness ratio χ_k), the damping factors (ξ_1 and ratio γ), and the normalized excitation frequency Ω. ϕ represents the phase difference of the incident sound field applied to the two mass-spring systems, which is related to the ratio of modal forces, as seen in Eq. (B.14).

Consequently, mIID and mIPD can be obtained as

$$mIID = 20\log_{10}\frac{|A_1|}{|A_2|} = 20\log_{10}\left|\frac{\Gamma + j\tan(\phi/2)}{\Gamma - j\tan(\phi/2)}\right| , \tag{B.15}$$

$$\text{and } mIPD = \angle\frac{A_1}{A_2} = \angle\frac{\Gamma + j\tan(\phi/2)}{\Gamma - j\tan(\phi/2)} , \tag{B.16}$$

where mIPD ranges from -π to π. mITD can be calculated via mIPD; that is,

$$mITD = \frac{mIPD}{2\pi\Omega} . \tag{B.17}$$

mIPD and mIID can also be interpreted in the complex plane. In Figure B.2, the trajectory of Γ is first drawn, and a point D is selected for any given frequency Ω. When the azimuth θ increases from 0° to 90°, point B and C moves along the vertical axis from the origin to the farthest point possible. Consequently, mIPD can be interpreted as the angle between vectors $\overline{DC}$ and $\overline{DB}$, and mIID as the magnitude ratio of them. It can be proved that as θ increases, mIPD increases monotonically. However, mIID achieves maximum when $\overline{DC}$ is perpendicular to $\overline{DB}$, i.e., mIPD is equal to 90°.

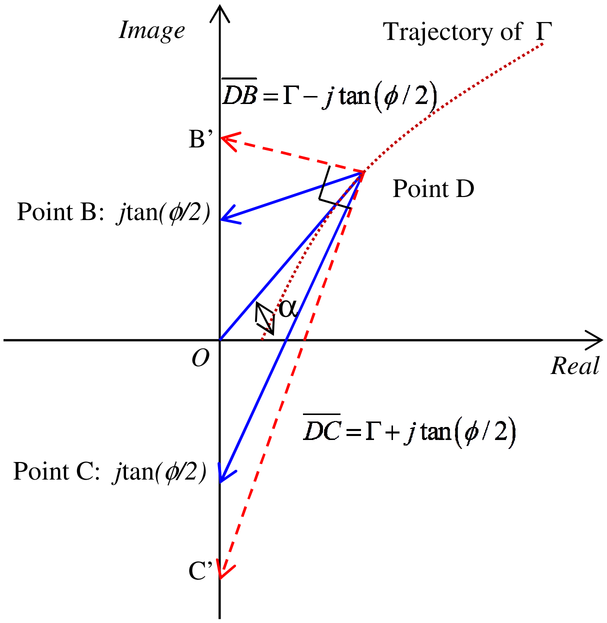


Image
Trajectory of Γ
$\overline{DB} = \Gamma - j\tan(\phi/2)$
B'
Point D
Point B: jtan(φ/2)
α
O
Real
$\overline{DC} = \Gamma + j\tan(\phi/2)$
Point C: jtan(φ/2)
C'

Figure B.2: Interpretation of mIPD and mIID in complex plane.

Let

$$\tau = \frac{\tan(\phi/2)}{|\Gamma|}, \alpha = \angle\Gamma \quad \text{(B.18)}$$

(B.15) and (B.16) can be rewritten as:

$$mIID = 20\log_{10}\sqrt{\frac{1+\tau^2+2\tau\sin\alpha}{1+\tau^2-2\tau\sin\alpha}} = 20\log_{10}\sqrt{1+\frac{4\sin\alpha}{\tau+\frac{1}{\tau}-2\sin\alpha}}\,, \tag{B.19}$$

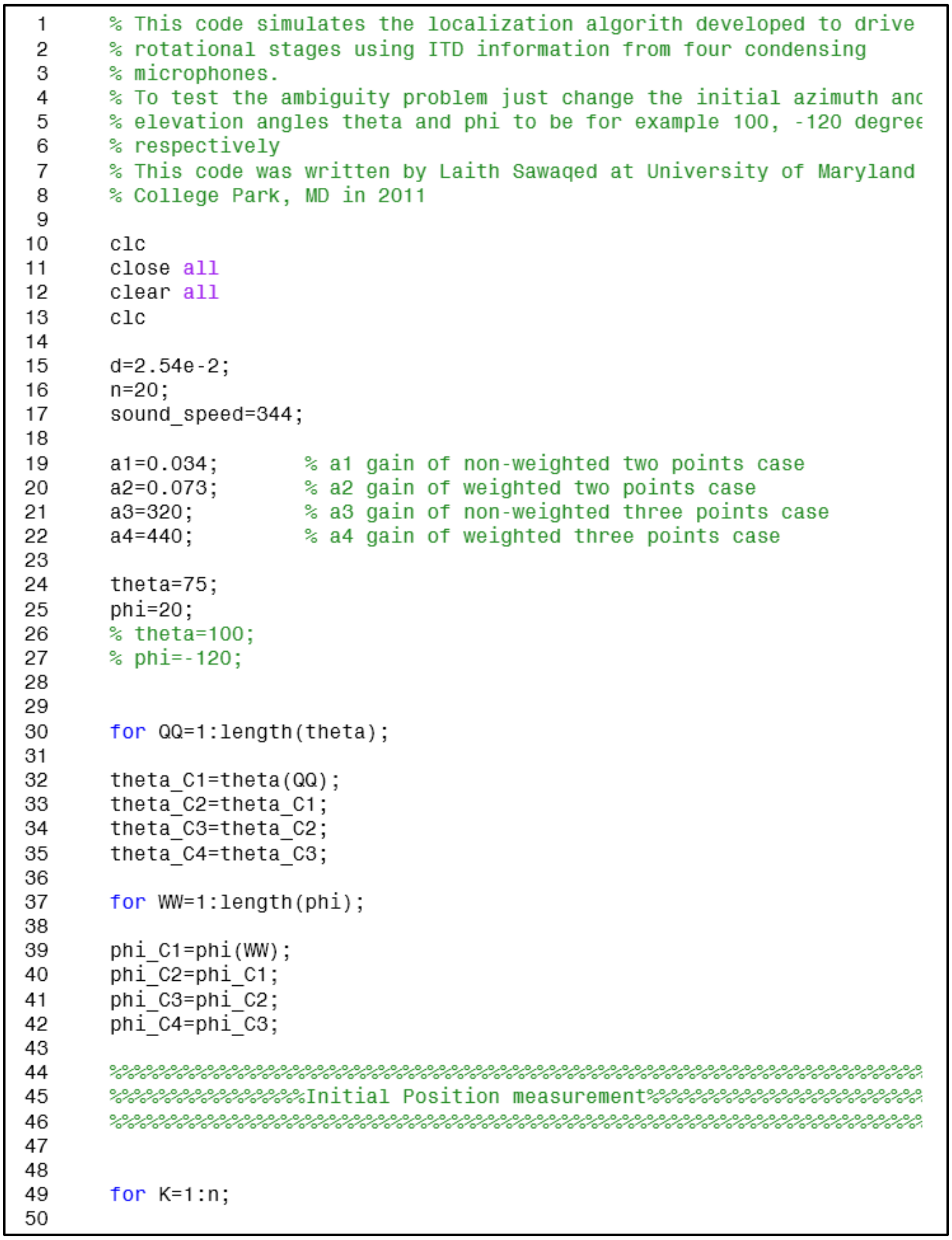
$$mIPD = \cos^{-1}\frac{1-\tau^2}{\sqrt{\left(1-\tau^2\right)^2+4\tau^2\cos^2\alpha}}\,. \tag{B.20}$$

It can be readily verified that when $\tau = 1$, mIPD is equal to 90° and $|mIID|$ reaches maximum:

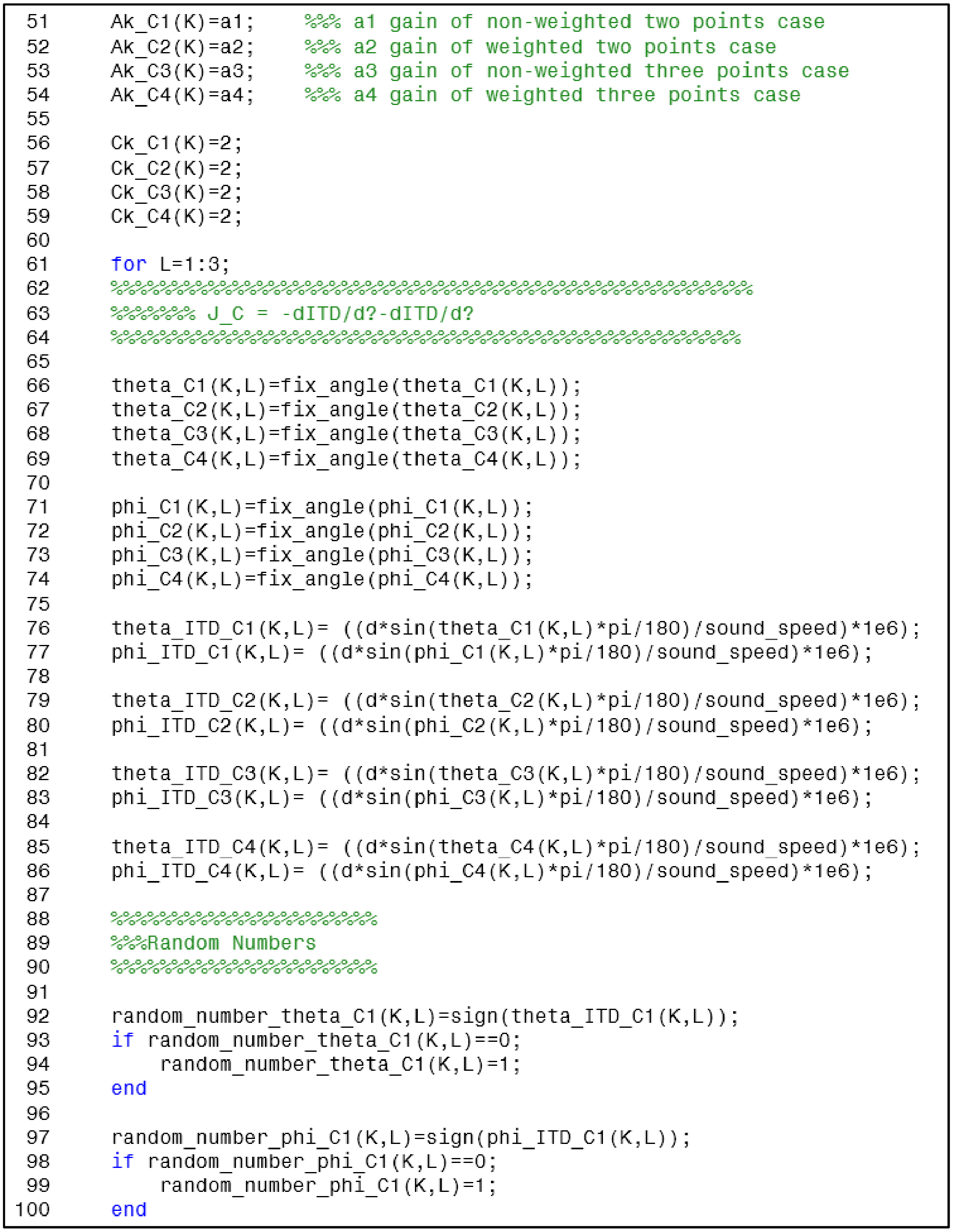
$$\max|mIID| = \left|10\log_{10}\frac{1+\sin\alpha}{1-\sin\alpha}\right|. \tag{B.21}$$

Appendix C MatLab Codes

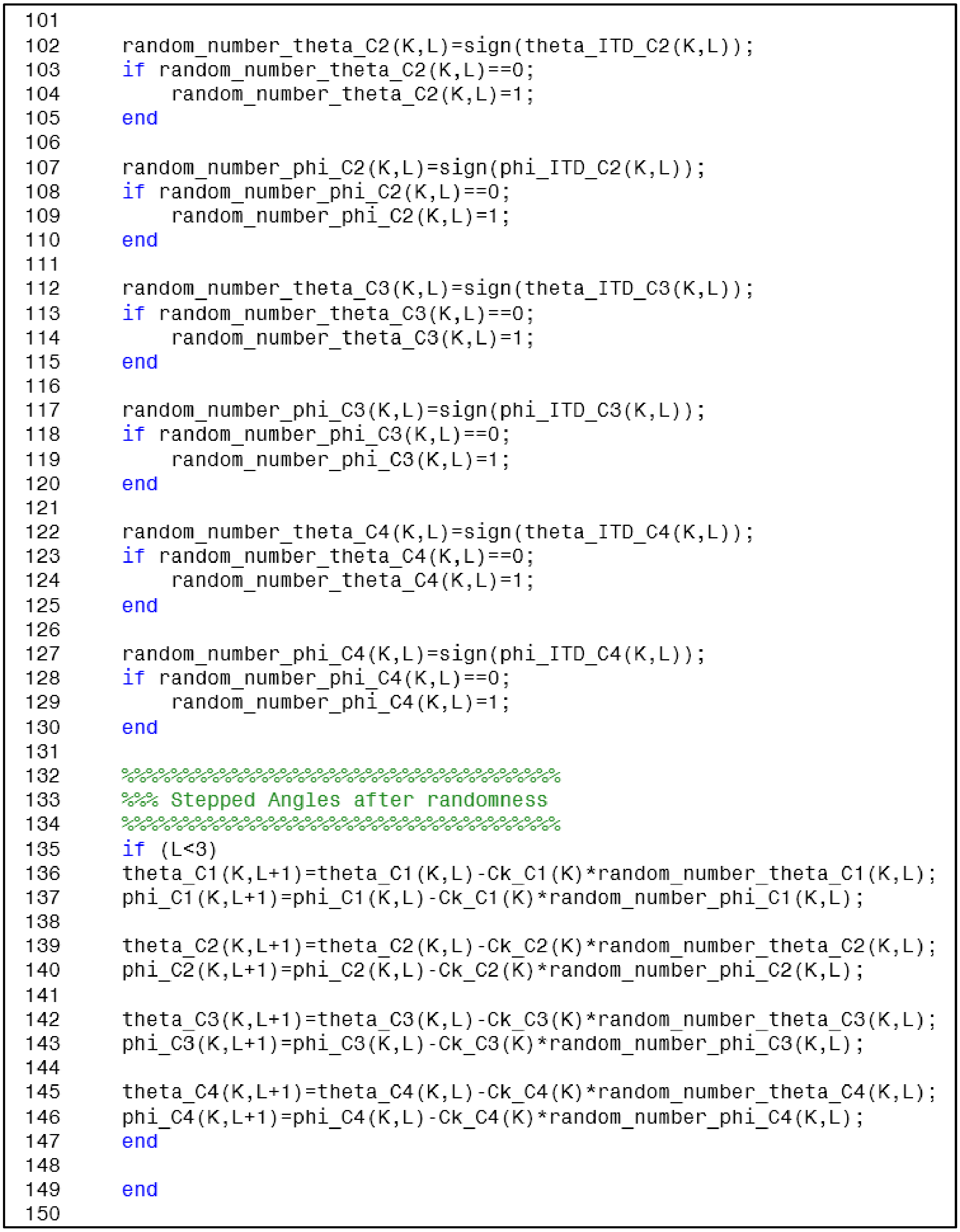
C.1 Sound Localization Using Two Rotational Stages



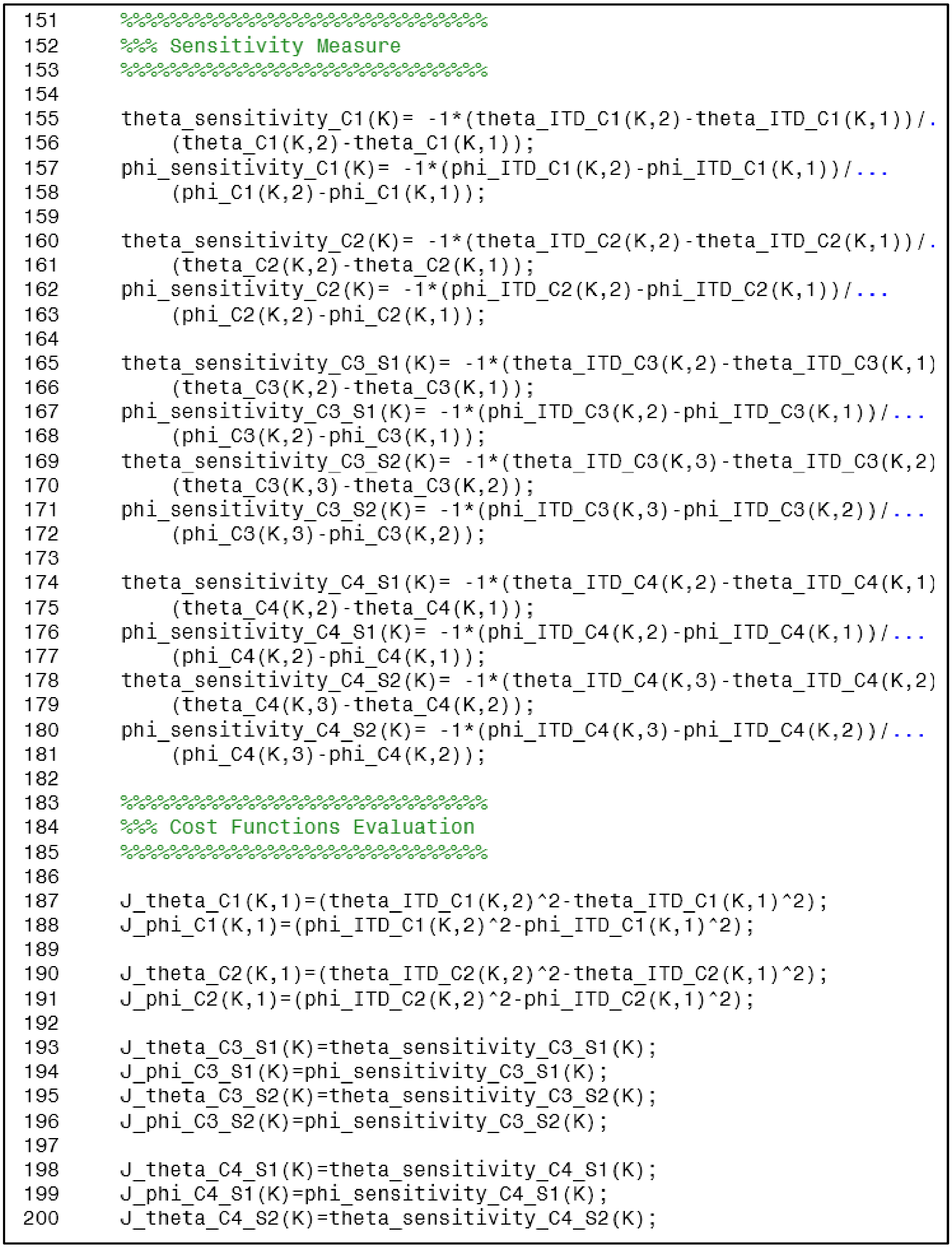
```
% This code simulates the localization algorith developed to drive
% rotational stages using ITD information from four condensing
% microphones.
% To test the ambiguity problem just change the initial azimuth anc
% elevation angles theta and phi to be for example 100, -120 degree
% respectively
% This code was written by Laith Sawaqed at University of Maryland
% College Park, MD in 2011

clc
close all
clear all
clc

d=2.54e-2;
n=20;
sound_speed=344;

a1=0.034;       % a1 gain of non-weighted two points case
a2=0.073;       % a2 gain of weighted two points case
a3=320;         % a3 gain of non-weighted three points case
a4=440;         % a4 gain of weighted three points case

theta=75;
phi=20;
% theta=100;
% phi=-120;


for QQ=1:length(theta);

theta_C1=theta(QQ);
theta_C2=theta_C1;
theta_C3=theta_C2;
theta_C4=theta_C3;

for WW=1:length(phi);

phi_C1=phi(WW);
phi_C2=phi_C1;
phi_C3=phi_C2;
phi_C4=phi_C3;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%Initial Position measurement%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


for K=1:n;

```



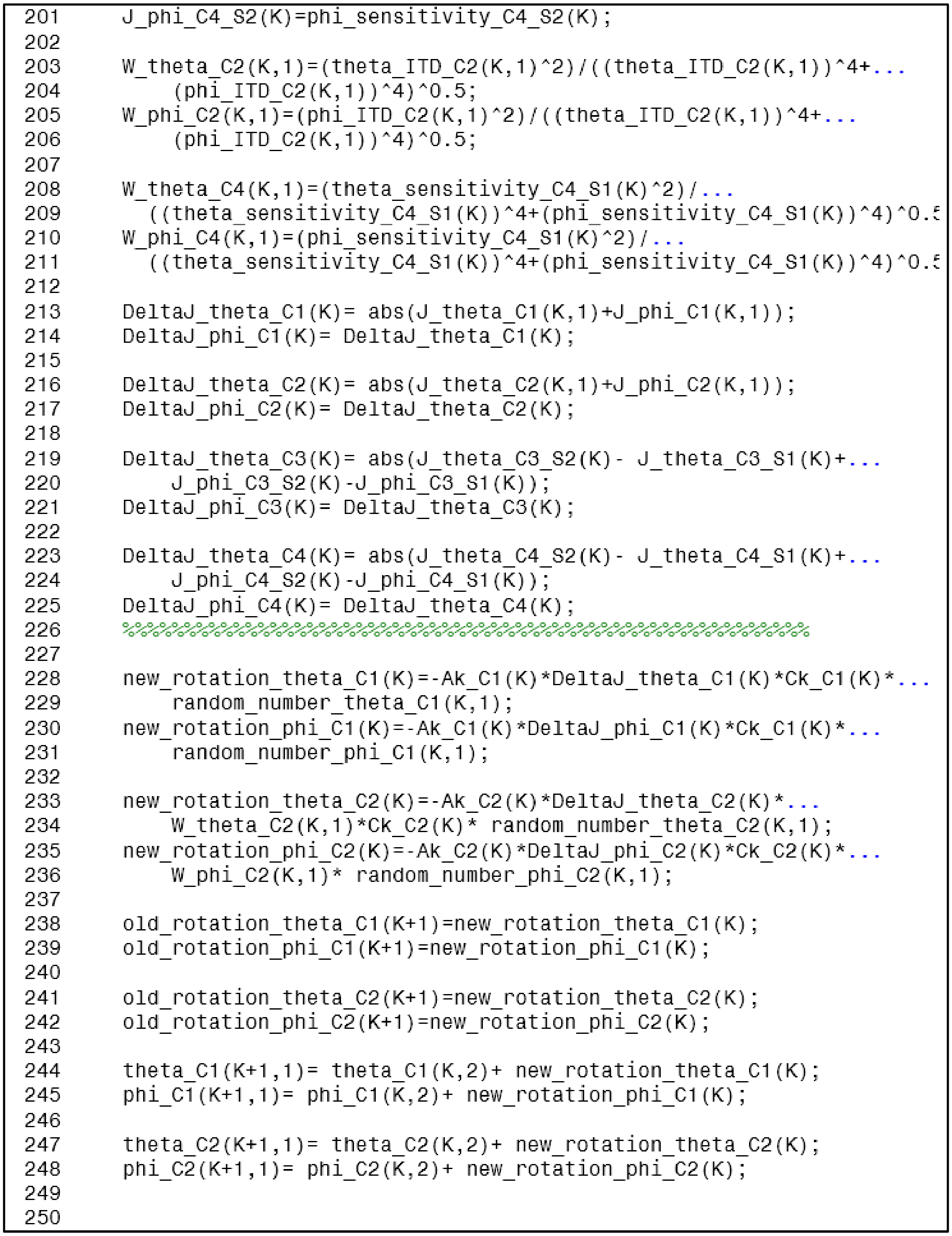
```
Ak_C1(K)=a1;    %%% a1 gain of non-weighted two points case
Ak_C2(K)=a2;    %%% a2 gain of weighted two points case
Ak_C3(K)=a3;    %%% a3 gain of non-weighted three points case
Ak_C4(K)=a4;    %%% a4 gain of weighted three points case

Ck_C1(K)=2;
Ck_C2(K)=2;
Ck_C3(K)=2;
Ck_C4(K)=2;

for L=1:3;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%% J_C = -dITD/d?-dITD/d?
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

theta_C1(K,L)=fix_angle(theta_C1(K,L));
theta_C2(K,L)=fix_angle(theta_C2(K,L));
theta_C3(K,L)=fix_angle(theta_C3(K,L));
theta_C4(K,L)=fix_angle(theta_C4(K,L));

phi_C1(K,L)=fix_angle(phi_C1(K,L));
phi_C2(K,L)=fix_angle(phi_C2(K,L));
phi_C3(K,L)=fix_angle(phi_C3(K,L));
phi_C4(K,L)=fix_angle(phi_C4(K,L));

theta_ITD_C1(K,L)= ((d*sin(theta_C1(K,L)*pi/180)/sound_speed)*1e6);
phi_ITD_C1(K,L)= ((d*sin(phi_C1(K,L)*pi/180)/sound_speed)*1e6);

theta_ITD_C2(K,L)= ((d*sin(theta_C2(K,L)*pi/180)/sound_speed)*1e6);
phi_ITD_C2(K,L)= ((d*sin(phi_C2(K,L)*pi/180)/sound_speed)*1e6);

theta_ITD_C3(K,L)= ((d*sin(theta_C3(K,L)*pi/180)/sound_speed)*1e6);
phi_ITD_C3(K,L)= ((d*sin(phi_C3(K,L)*pi/180)/sound_speed)*1e6);

theta_ITD_C4(K,L)= ((d*sin(theta_C4(K,L)*pi/180)/sound_speed)*1e6);
phi_ITD_C4(K,L)= ((d*sin(phi_C4(K,L)*pi/180)/sound_speed)*1e6);

%%%%%%%%%%%%%%%%%%%%%%
%%%Random Numbers
%%%%%%%%%%%%%%%%%%%%%%

random_number_theta_C1(K,L)=sign(theta_ITD_C1(K,L));
if random_number_theta_C1(K,L)==0;
    random_number_theta_C1(K,L)=1;
end

random_number_phi_C1(K,L)=sign(phi_ITD_C1(K,L));
if random_number_phi_C1(K,L)==0;
    random_number_phi_C1(K,L)=1;
end
```



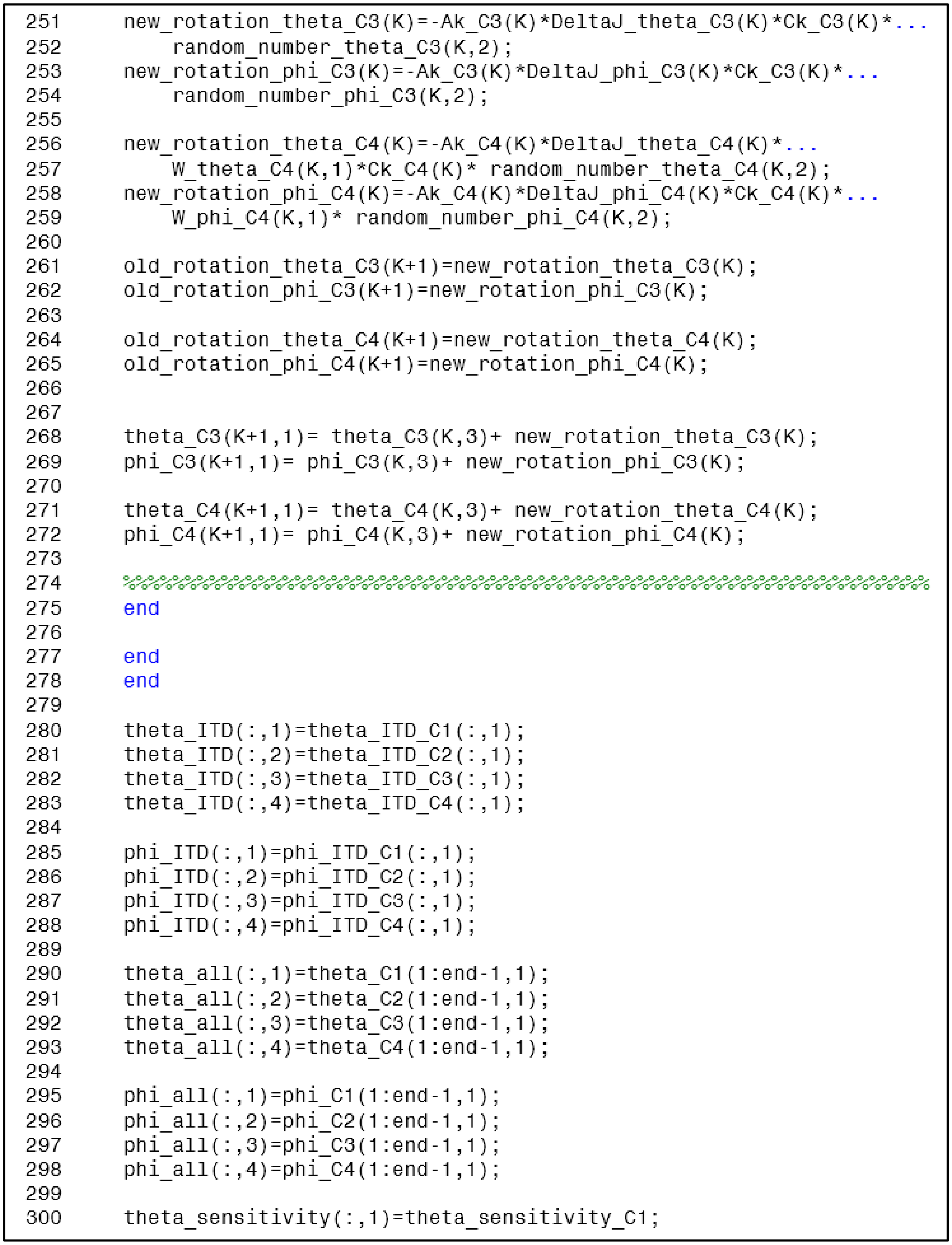
```
    random_number_theta_C2(K,L)=sign(theta_ITD_C2(K,L));
    if random_number_theta_C2(K,L)==0;
        random_number_theta_C2(K,L)=1;
    end

    random_number_phi_C2(K,L)=sign(phi_ITD_C2(K,L));
    if random_number_phi_C2(K,L)==0;
        random_number_phi_C2(K,L)=1;
    end

    random_number_theta_C3(K,L)=sign(theta_ITD_C3(K,L));
    if random_number_theta_C3(K,L)==0;
        random_number_theta_C3(K,L)=1;
    end

    random_number_phi_C3(K,L)=sign(phi_ITD_C3(K,L));
    if random_number_phi_C3(K,L)==0;
        random_number_phi_C3(K,L)=1;
    end

    random_number_theta_C4(K,L)=sign(theta_ITD_C4(K,L));
    if random_number_theta_C4(K,L)==0;
        random_number_theta_C4(K,L)=1;
    end

    random_number_phi_C4(K,L)=sign(phi_ITD_C4(K,L));
    if random_number_phi_C4(K,L)==0;
        random_number_phi_C4(K,L)=1;
    end

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%% Stepped Angles after randomness
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    if (L<3)
    theta_C1(K,L+1)=theta_C1(K,L)-Ck_C1(K)*random_number_theta_C1(K,L);
    phi_C1(K,L+1)=phi_C1(K,L)-Ck_C1(K)*random_number_phi_C1(K,L);

    theta_C2(K,L+1)=theta_C2(K,L)-Ck_C2(K)*random_number_theta_C2(K,L);
    phi_C2(K,L+1)=phi_C2(K,L)-Ck_C2(K)*random_number_phi_C2(K,L);

    theta_C3(K,L+1)=theta_C3(K,L)-Ck_C3(K)*random_number_theta_C3(K,L);
    phi_C3(K,L+1)=phi_C3(K,L)-Ck_C3(K)*random_number_phi_C3(K,L);

    theta_C4(K,L+1)=theta_C4(K,L)-Ck_C4(K)*random_number_theta_C4(K,L);
    phi_C4(K,L+1)=phi_C4(K,L)-Ck_C4(K)*random_number_phi_C4(K,L);
    end

    end
```



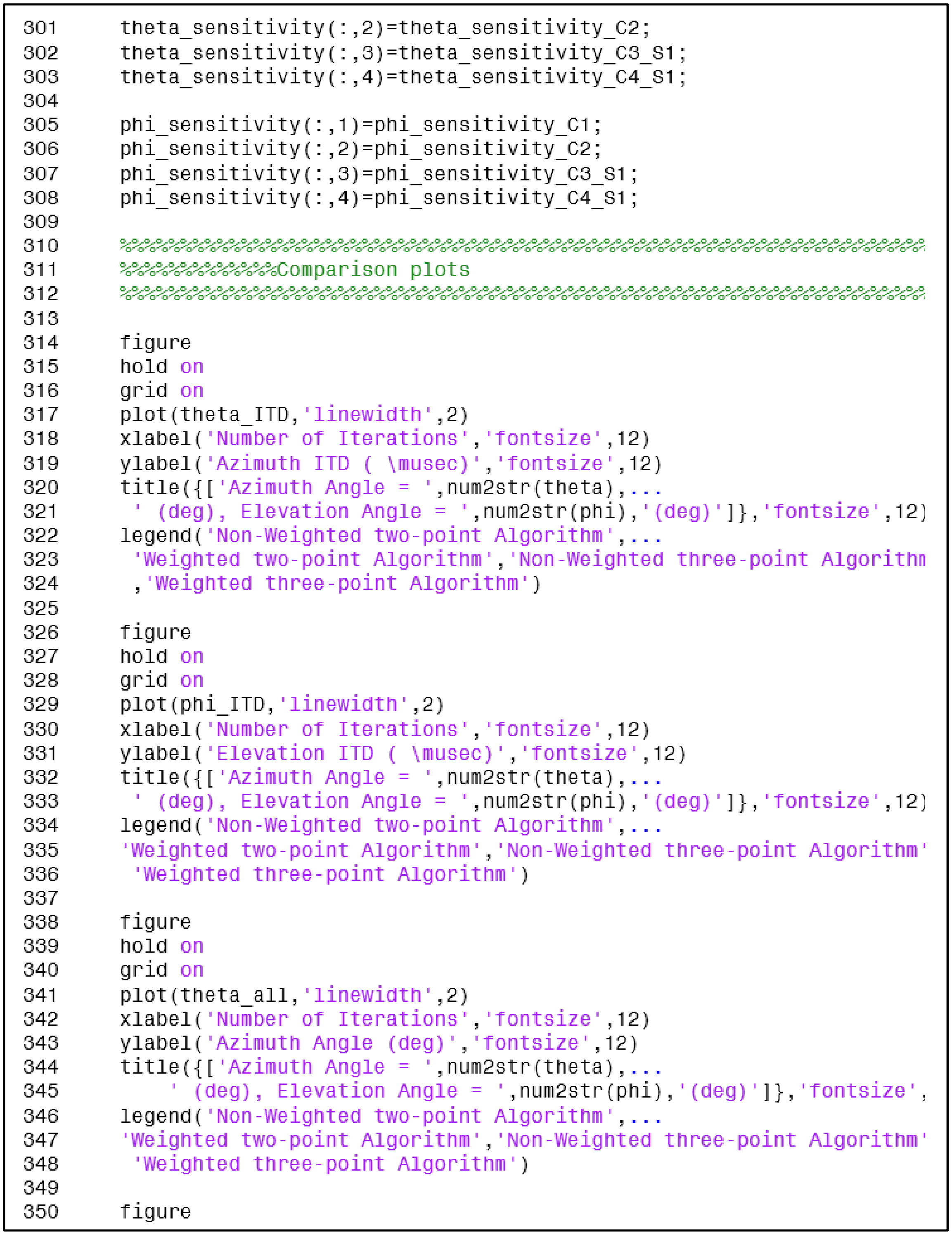
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Sensitivity Measure
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

theta_sensitivity_C1(K)= -1*(theta_ITD_C1(K,2)-theta_ITD_C1(K,1))/.
    (theta_C1(K,2)-theta_C1(K,1));
phi_sensitivity_C1(K)= -1*(phi_ITD_C1(K,2)-phi_ITD_C1(K,1))/...
    (phi_C1(K,2)-phi_C1(K,1));

theta_sensitivity_C2(K)= -1*(theta_ITD_C2(K,2)-theta_ITD_C2(K,1))/.
    (theta_C2(K,2)-theta_C2(K,1));
phi_sensitivity_C2(K)= -1*(phi_ITD_C2(K,2)-phi_ITD_C2(K,1))/...
    (phi_C2(K,2)-phi_C2(K,1));

theta_sensitivity_C3_S1(K)= -1*(theta_ITD_C3(K,2)-theta_ITD_C3(K,1)
    (theta_C3(K,2)-theta_C3(K,1));
phi_sensitivity_C3_S1(K)= -1*(phi_ITD_C3(K,2)-phi_ITD_C3(K,1))/...
    (phi_C3(K,2)-phi_C3(K,1));
theta_sensitivity_C3_S2(K)= -1*(theta_ITD_C3(K,3)-theta_ITD_C3(K,2)
    (theta_C3(K,3)-theta_C3(K,2));
phi_sensitivity_C3_S2(K)= -1*(phi_ITD_C3(K,3)-phi_ITD_C3(K,2))/...
    (phi_C3(K,3)-phi_C3(K,2));

theta_sensitivity_C4_S1(K)= -1*(theta_ITD_C4(K,2)-theta_ITD_C4(K,1)
    (theta_C4(K,2)-theta_C4(K,1));
phi_sensitivity_C4_S1(K)= -1*(phi_ITD_C4(K,2)-phi_ITD_C4(K,1))/...
    (phi_C4(K,2)-phi_C4(K,1));
theta_sensitivity_C4_S2(K)= -1*(theta_ITD_C4(K,3)-theta_ITD_C4(K,2)
    (theta_C4(K,3)-theta_C4(K,2));
phi_sensitivity_C4_S2(K)= -1*(phi_ITD_C4(K,3)-phi_ITD_C4(K,2))/...
    (phi_C4(K,3)-phi_C4(K,2));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Cost Functions Evaluation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

J_theta_C1(K,1)=(theta_ITD_C1(K,2)^2-theta_ITD_C1(K,1)^2);
J_phi_C1(K,1)=(phi_ITD_C1(K,2)^2-phi_ITD_C1(K,1)^2);

J_theta_C2(K,1)=(theta_ITD_C2(K,2)^2-theta_ITD_C2(K,1)^2);
J_phi_C2(K,1)=(phi_ITD_C2(K,2)^2-phi_ITD_C2(K,1)^2);

J_theta_C3_S1(K)=theta_sensitivity_C3_S1(K);
J_phi_C3_S1(K)=phi_sensitivity_C3_S1(K);
J_theta_C3_S2(K)=theta_sensitivity_C3_S2(K);
J_phi_C3_S2(K)=phi_sensitivity_C3_S2(K);

J_theta_C4_S1(K)=theta_sensitivity_C4_S1(K);
J_phi_C4_S1(K)=phi_sensitivity_C4_S1(K);
J_theta_C4_S2(K)=theta_sensitivity_C4_S2(K);
```



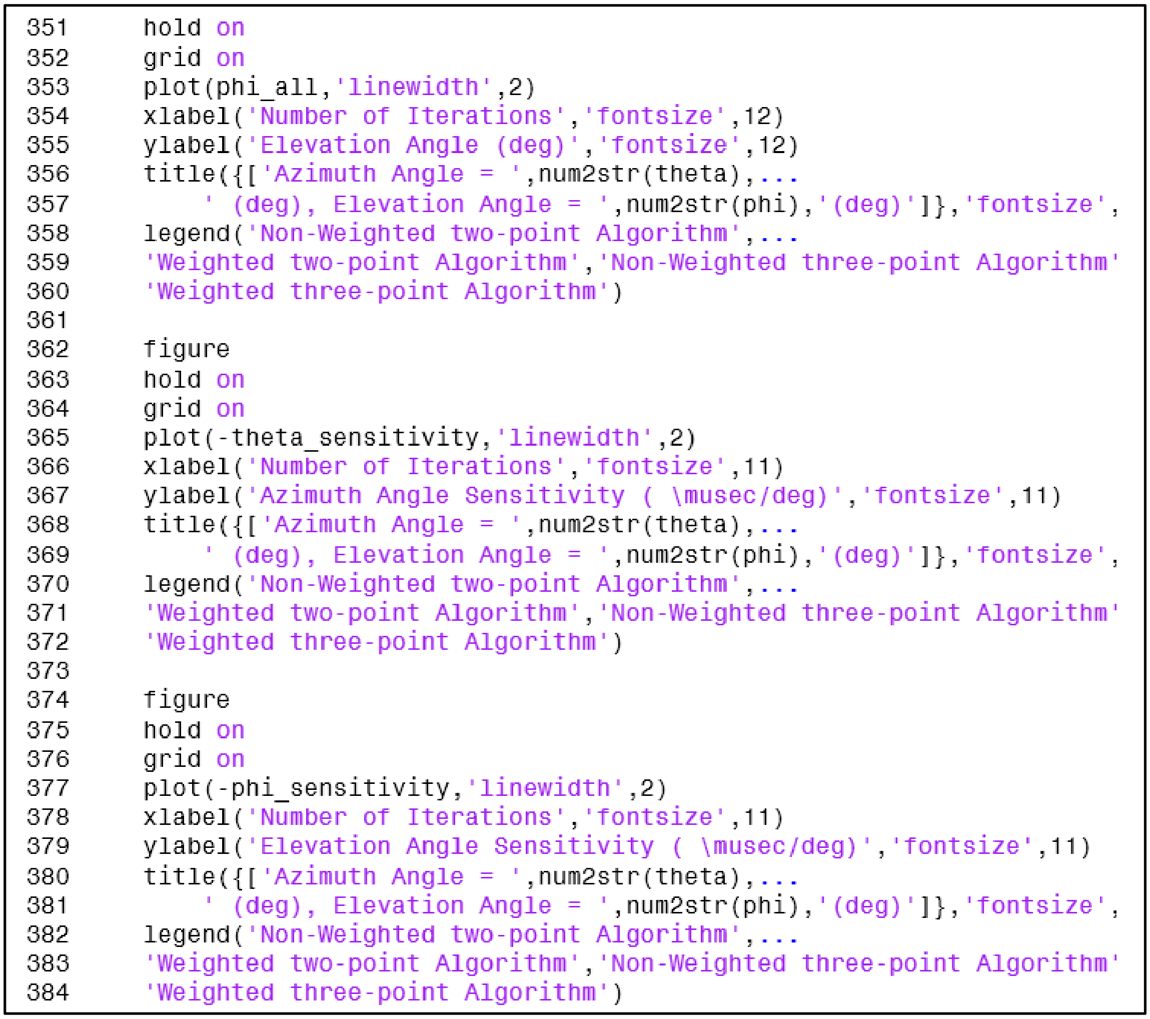
```
J_phi_C4_S2(K)=phi_sensitivity_C4_S2(K);

W_theta_C2(K,1)=(theta_ITD_C2(K,1)^2)/((theta_ITD_C2(K,1))^4+...
    (phi_ITD_C2(K,1))^4)^0.5;
W_phi_C2(K,1)=(phi_ITD_C2(K,1)^2)/((theta_ITD_C2(K,1))^4+...
    (phi_ITD_C2(K,1))^4)^0.5;

W_theta_C4(K,1)=(theta_sensitivity_C4_S1(K)^2)/...
  ((theta_sensitivity_C4_S1(K))^4+(phi_sensitivity_C4_S1(K))^4)^0.5
W_phi_C4(K,1)=(phi_sensitivity_C4_S1(K)^2)/...
  ((theta_sensitivity_C4_S1(K))^4+(phi_sensitivity_C4_S1(K))^4)^0.5

DeltaJ_theta_C1(K)= abs(J_theta_C1(K,1)+J_phi_C1(K,1));
DeltaJ_phi_C1(K)= DeltaJ_theta_C1(K);

DeltaJ_theta_C2(K)= abs(J_theta_C2(K,1)+J_phi_C2(K,1));
DeltaJ_phi_C2(K)= DeltaJ_theta_C2(K);

DeltaJ_theta_C3(K)= abs(J_theta_C3_S2(K)- J_theta_C3_S1(K)+...
    J_phi_C3_S2(K)-J_phi_C3_S1(K));
DeltaJ_phi_C3(K)= DeltaJ_theta_C3(K);

DeltaJ_theta_C4(K)= abs(J_theta_C4_S2(K)- J_theta_C4_S1(K)+...
    J_phi_C4_S2(K)-J_phi_C4_S1(K));
DeltaJ_phi_C4(K)= DeltaJ_theta_C4(K);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

new_rotation_theta_C1(K)=-Ak_C1(K)*DeltaJ_theta_C1(K)*Ck_C1(K)*...
    random_number_theta_C1(K,1);
new_rotation_phi_C1(K)=-Ak_C1(K)*DeltaJ_phi_C1(K)*Ck_C1(K)*...
    random_number_phi_C1(K,1);

new_rotation_theta_C2(K)=-Ak_C2(K)*DeltaJ_theta_C2(K)*...
    W_theta_C2(K,1)*Ck_C2(K)* random_number_theta_C2(K,1);
new_rotation_phi_C2(K)=-Ak_C2(K)*DeltaJ_phi_C2(K)*Ck_C2(K)*...
    W_phi_C2(K,1)* random_number_phi_C2(K,1);

old_rotation_theta_C1(K+1)=new_rotation_theta_C1(K);
old_rotation_phi_C1(K+1)=new_rotation_phi_C1(K);

old_rotation_theta_C2(K+1)=new_rotation_theta_C2(K);
old_rotation_phi_C2(K+1)=new_rotation_phi_C2(K);

theta_C1(K+1,1)= theta_C1(K,2)+ new_rotation_theta_C1(K);
phi_C1(K+1,1)= phi_C1(K,2)+ new_rotation_phi_C1(K);

theta_C2(K+1,1)= theta_C2(K,2)+ new_rotation_theta_C2(K);
phi_C2(K+1,1)= phi_C2(K,2)+ new_rotation_phi_C2(K);
```



```
new_rotation_theta_C3(K)=-Ak_C3(K)*DeltaJ_theta_C3(K)*Ck_C3(K)*...
    random_number_theta_C3(K,2);
new_rotation_phi_C3(K)=-Ak_C3(K)*DeltaJ_phi_C3(K)*Ck_C3(K)*...
    random_number_phi_C3(K,2);

new_rotation_theta_C4(K)=-Ak_C4(K)*DeltaJ_theta_C4(K)*...
    W_theta_C4(K,1)*Ck_C4(K)* random_number_theta_C4(K,2);
new_rotation_phi_C4(K)=-Ak_C4(K)*DeltaJ_phi_C4(K)*Ck_C4(K)*...
    W_phi_C4(K,1)* random_number_phi_C4(K,2);

old_rotation_theta_C3(K+1)=new_rotation_theta_C3(K);
old_rotation_phi_C3(K+1)=new_rotation_phi_C3(K);

old_rotation_theta_C4(K+1)=new_rotation_theta_C4(K);
old_rotation_phi_C4(K+1)=new_rotation_phi_C4(K);


theta_C3(K+1,1)= theta_C3(K,3)+ new_rotation_theta_C3(K);
phi_C3(K+1,1)= phi_C3(K,3)+ new_rotation_phi_C3(K);

theta_C4(K+1,1)= theta_C4(K,3)+ new_rotation_theta_C4(K);
phi_C4(K+1,1)= phi_C4(K,3)+ new_rotation_phi_C4(K);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end

end
end

theta_ITD(:,1)=theta_ITD_C1(:,1);
theta_ITD(:,2)=theta_ITD_C2(:,1);
theta_ITD(:,3)=theta_ITD_C3(:,1);
theta_ITD(:,4)=theta_ITD_C4(:,1);

phi_ITD(:,1)=phi_ITD_C1(:,1);
phi_ITD(:,2)=phi_ITD_C2(:,1);
phi_ITD(:,3)=phi_ITD_C3(:,1);
phi_ITD(:,4)=phi_ITD_C4(:,1);

theta_all(:,1)=theta_C1(1:end-1,1);
theta_all(:,2)=theta_C2(1:end-1,1);
theta_all(:,3)=theta_C3(1:end-1,1);
theta_all(:,4)=theta_C4(1:end-1,1);

phi_all(:,1)=phi_C1(1:end-1,1);
phi_all(:,2)=phi_C2(1:end-1,1);
phi_all(:,3)=phi_C3(1:end-1,1);
phi_all(:,4)=phi_C4(1:end-1,1);

theta_sensitivity(:,1)=theta_sensitivity_C1;
```

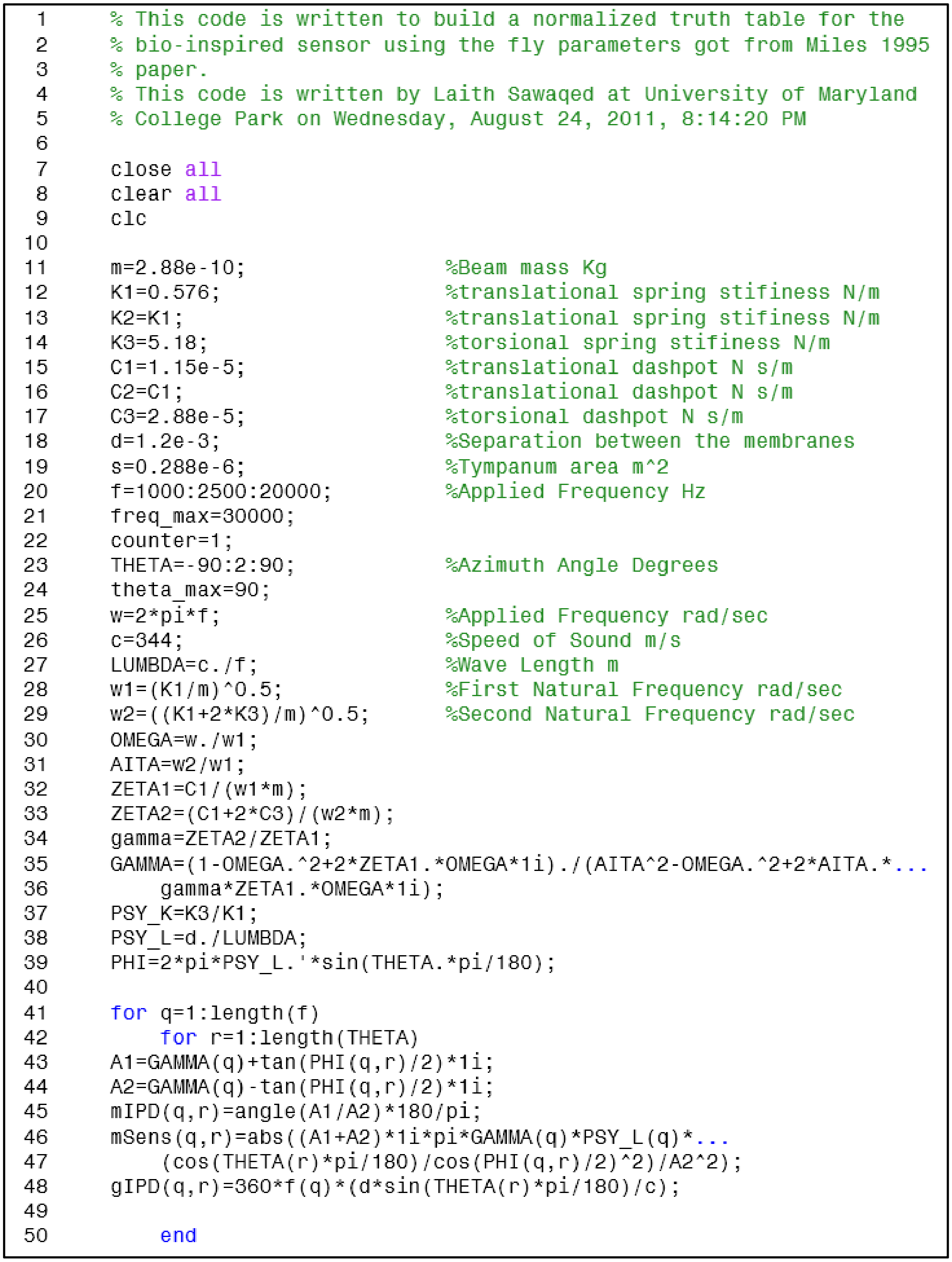


```
    theta_sensitivity(:,2)=theta_sensitivity_C2;
    theta_sensitivity(:,3)=theta_sensitivity_C3_S1;
    theta_sensitivity(:,4)=theta_sensitivity_C4_S1;

    phi_sensitivity(:,1)=phi_sensitivity_C1;
    phi_sensitivity(:,2)=phi_sensitivity_C2;
    phi_sensitivity(:,3)=phi_sensitivity_C3_S1;
    phi_sensitivity(:,4)=phi_sensitivity_C4_S1;

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%Comparison plots
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    figure
    hold on
    grid on
    plot(theta_ITD,'linewidth',2)
    xlabel('Number of Iterations','fontsize',12)
    ylabel('Azimuth ITD ( \musec)','fontsize',12)
    title({['Azimuth Angle = ',num2str(theta),...
     ' (deg), Elevation Angle = ',num2str(phi),'(deg)']},'fontsize',12)
    legend('Non-Weighted two-point Algorithm',...
     'Weighted two-point Algorithm','Non-Weighted three-point Algorithm
     ,'Weighted three-point Algorithm')

    figure
    hold on
    grid on
    plot(phi_ITD,'linewidth',2)
    xlabel('Number of Iterations','fontsize',12)
    ylabel('Elevation ITD ( \musec)','fontsize',12)
    title({['Azimuth Angle = ',num2str(theta),...
     ' (deg), Elevation Angle = ',num2str(phi),'(deg)']},'fontsize',12)
    legend('Non-Weighted two-point Algorithm',...
    'Weighted two-point Algorithm','Non-Weighted three-point Algorithm'
     'Weighted three-point Algorithm')

    figure
    hold on
    grid on
    plot(theta_all,'linewidth',2)
    xlabel('Number of Iterations','fontsize',12)
    ylabel('Azimuth Angle (deg)','fontsize',12)
    title({['Azimuth Angle = ',num2str(theta),...
        ' (deg), Elevation Angle = ',num2str(phi),'(deg)']},'fontsize',
    legend('Non-Weighted two-point Algorithm',...
    'Weighted two-point Algorithm','Non-Weighted three-point Algorithm'
     'Weighted three-point Algorithm')

    figure
```

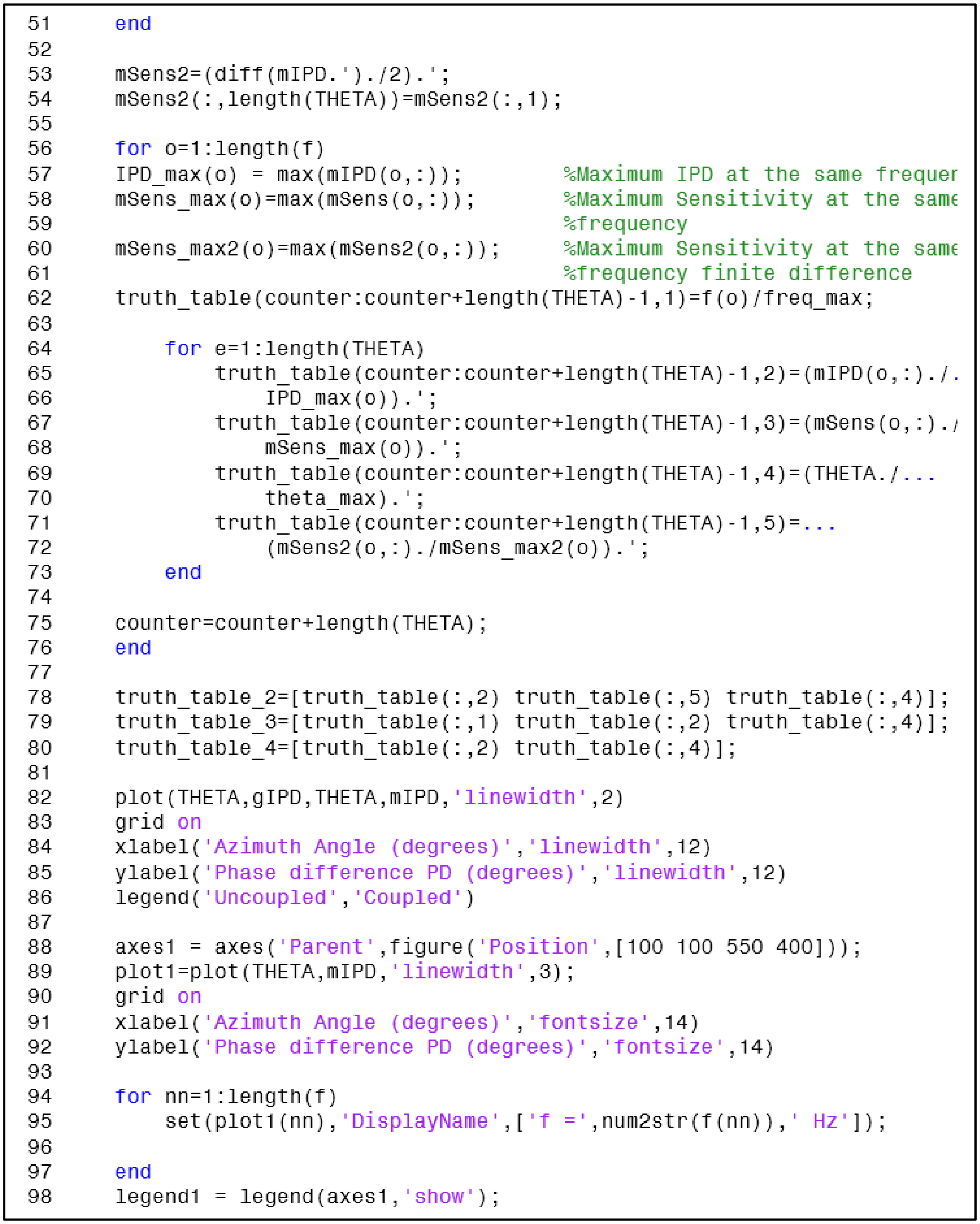


```
hold on
grid on
plot(phi_all,'linewidth',2)
xlabel('Number of Iterations','fontsize',12)
ylabel('Elevation Angle (deg)','fontsize',12)
title({['Azimuth Angle = ',num2str(theta),...
    ' (deg), Elevation Angle = ',num2str(phi),'(deg)']},'fontsize',
legend('Non-Weighted two-point Algorithm',...
'Weighted two-point Algorithm','Non-Weighted three-point Algorithm'
'Weighted three-point Algorithm')

figure
hold on
grid on
plot(-theta_sensitivity,'linewidth',2)
xlabel('Number of Iterations','fontsize',11)
ylabel('Azimuth Angle Sensitivity ( \musec/deg)','fontsize',11)
title({['Azimuth Angle = ',num2str(theta),...
    ' (deg), Elevation Angle = ',num2str(phi),'(deg)']},'fontsize',
legend('Non-Weighted two-point Algorithm',...
'Weighted two-point Algorithm','Non-Weighted three-point Algorithm'
'Weighted three-point Algorithm')

figure
hold on
grid on
plot(-phi_sensitivity,'linewidth',2)
xlabel('Number of Iterations','fontsize',11)
ylabel('Elevation Angle Sensitivity ( \musec/deg)','fontsize',11)
title({['Azimuth Angle = ',num2str(theta),...
    ' (deg), Elevation Angle = ',num2str(phi),'(deg)']},'fontsize',
legend('Non-Weighted two-point Algorithm',...
'Weighted two-point Algorithm','Non-Weighted three-point Algorithm'
'Weighted three-point Algorithm')
```

C.2 mIPD vs. the Incident Azimuth Angle

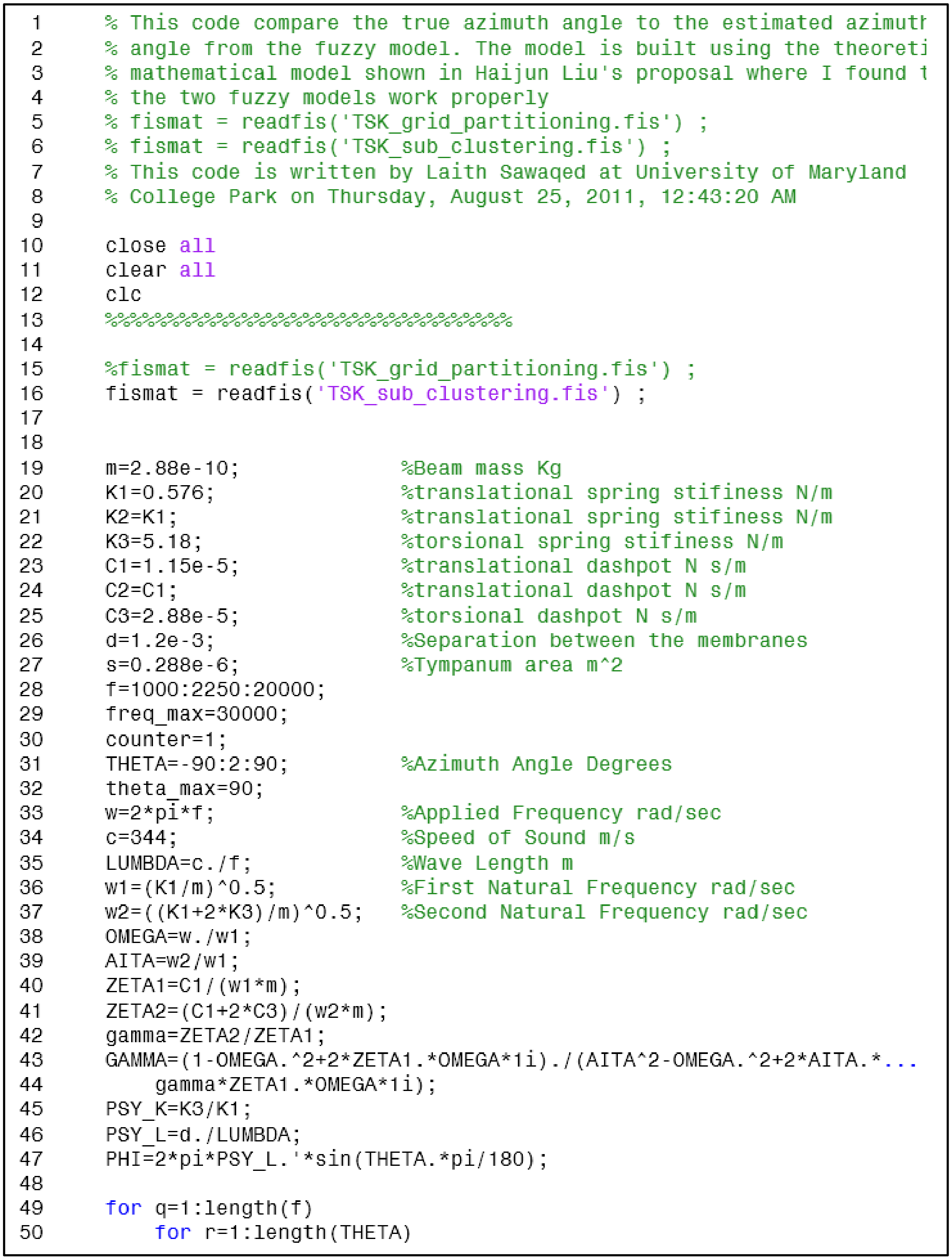


```
% This code is written to build a normalized truth table for the
% bio-inspired sensor using the fly parameters got from Miles 1995
% paper.
% This code is written by Laith Sawaqed at University of Maryland
% College Park on Wednesday, August 24, 2011, 8:14:20 PM

close all
clear all
clc

m=2.88e-10;                %Beam mass Kg
K1=0.576;                  %translational spring stifiness N/m
K2=K1;                     %translational spring stifiness N/m
K3=5.18;                   %torsional spring stifiness N/m
C1=1.15e-5;                %translational dashpot N s/m
C2=C1;                     %translational dashpot N s/m
C3=2.88e-5;                %torsional dashpot N s/m
d=1.2e-3;                  %Separation between the membranes
s=0.288e-6;                %Tympanum area m^2
f=1000:2500:20000;         %Applied Frequency Hz
freq_max=30000;
counter=1;
THETA=-90:2:90;            %Azimuth Angle Degrees
theta_max=90;
w=2*pi*f;                  %Applied Frequency rad/sec
c=344;                     %Speed of Sound m/s
LUMBDA=c./f;               %Wave Length m
w1=(K1/m)^0.5;             %First Natural Frequency rad/sec
w2=((K1+2*K3)/m)^0.5;      %Second Natural Frequency rad/sec
OMEGA=w./w1;
AITA=w2/w1;
ZETA1=C1/(w1*m);
ZETA2=(C1+2*C3)/(w2*m);
gamma=ZETA2/ZETA1;
GAMMA=(1-OMEGA.^2+2*ZETA1.*OMEGA*1i)./(AITA^2-OMEGA.^2+2*AITA.*...
    gamma*ZETA1.*OMEGA*1i);
PSY_K=K3/K1;
PSY_L=d./LUMBDA;
PHI=2*pi*PSY_L.'*sin(THETA.*pi/180);

for q=1:length(f)
    for r=1:length(THETA)
A1=GAMMA(q)+tan(PHI(q,r)/2)*1i;
A2=GAMMA(q)-tan(PHI(q,r)/2)*1i;
mIPD(q,r)=angle(A1/A2)*180/pi;
mSens(q,r)=abs((A1+A2)*1i*pi*GAMMA(q)*PSY_L(q)*...
    (cos(THETA(r)*pi/180)/cos(PHI(q,r)/2)^2)/A2^2);
gIPD(q,r)=360*f(q)*(d*sin(THETA(r)*pi/180)/c);

    end
```

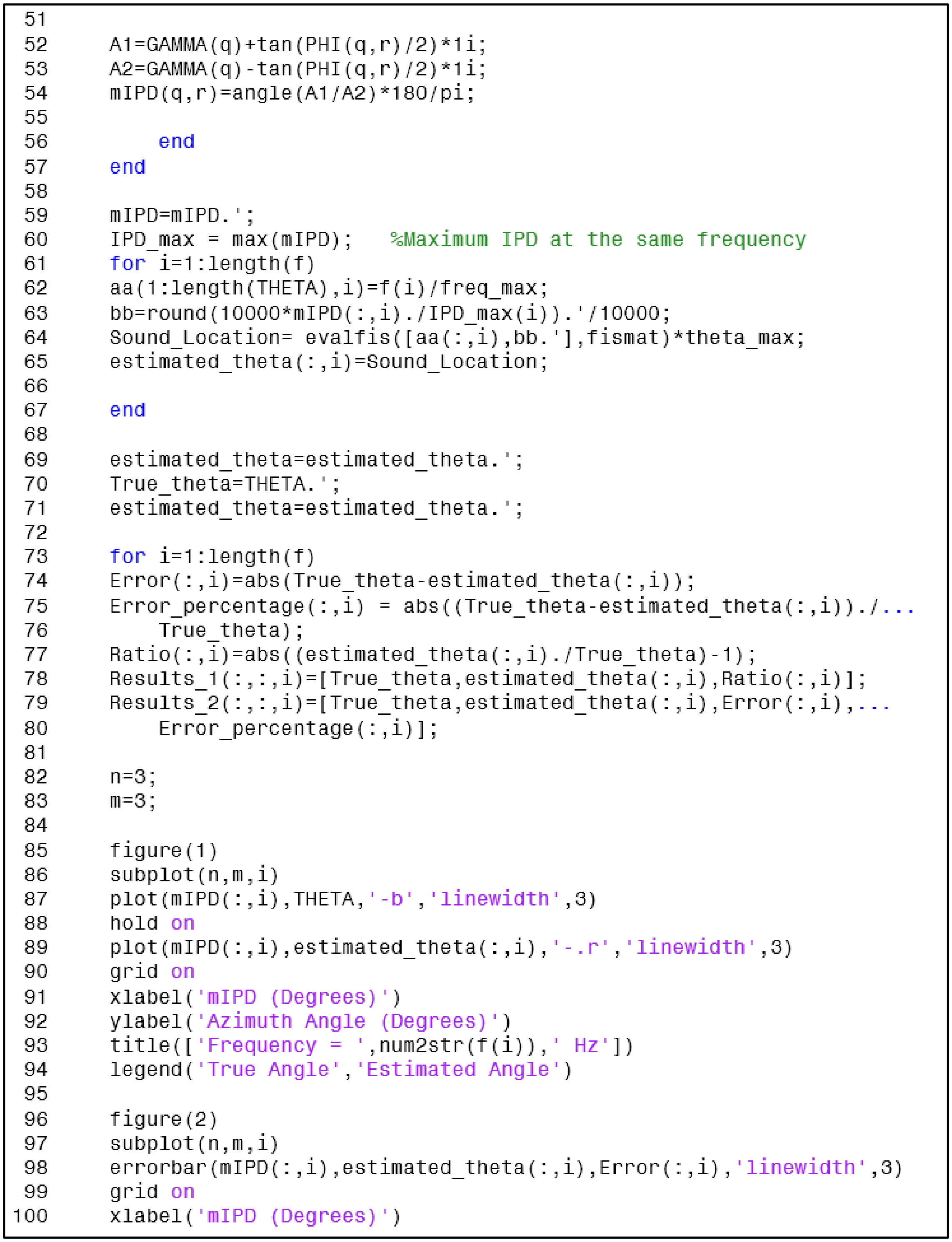


```
end

mSens2=(diff(mIPD.')./2).';
mSens2(:,length(THETA))=mSens2(:,1);

for o=1:length(f)
IPD_max(o) = max(mIPD(o,:));        %Maximum IPD at the same frequer
mSens_max(o)=max(mSens(o,:));       %Maximum Sensitivity at the same
                                    %frequency
mSens_max2(o)=max(mSens2(o,:));     %Maximum Sensitivity at the same
                                    %frequency finite difference
truth_table(counter:counter+length(THETA)-1,1)=f(o)/freq_max;

    for e=1:length(THETA)
        truth_table(counter:counter+length(THETA)-1,2)=(mIPD(o,:)./.
            IPD_max(o)).';
        truth_table(counter:counter+length(THETA)-1,3)=(mSens(o,:)./
            mSens_max(o)).';
        truth_table(counter:counter+length(THETA)-1,4)=(THETA./...
            theta_max).';
        truth_table(counter:counter+length(THETA)-1,5)=...
            (mSens2(o,:)./mSens_max2(o)).';
    end

counter=counter+length(THETA);
end

truth_table_2=[truth_table(:,2) truth_table(:,5) truth_table(:,4)];
truth_table_3=[truth_table(:,1) truth_table(:,2) truth_table(:,4)];
truth_table_4=[truth_table(:,2) truth_table(:,4)];

plot(THETA,gIPD,THETA,mIPD,'linewidth',2)
grid on
xlabel('Azimuth Angle (degrees)','linewidth',12)
ylabel('Phase difference PD (degrees)','linewidth',12)
legend('Uncoupled','Coupled')

axes1 = axes('Parent',figure('Position',[100 100 550 400]));
plot1=plot(THETA,mIPD,'linewidth',3);
grid on
xlabel('Azimuth Angle (degrees)','fontsize',14)
ylabel('Phase difference PD (degrees)','fontsize',14)

for nn=1:length(f)
    set(plot1(nn),'DisplayName',['f =',num2str(f(nn)),' Hz']);

end
legend1 = legend(axes1,'show');
```

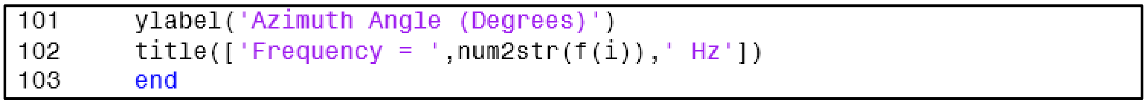
C.3 Testing the Two Proposed Fuzzy Models



```
% This code compare the true azimuth angle to the estimated azimuth
% angle from the fuzzy model. The model is built using the theoreti
% mathematical model shown in Haijun Liu's proposal where I found t
% the two fuzzy models work properly
% fismat = readfis('TSK_grid_partitioning.fis') ;
% fismat = readfis('TSK_sub_clustering.fis') ;
% This code is written by Laith Sawaqed at University of Maryland
% College Park on Thursday, August 25, 2011, 12:43:20 AM

close all
clear all
clc
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%fismat = readfis('TSK_grid_partitioning.fis') ;
fismat = readfis('TSK_sub_clustering.fis') ;


m=2.88e-10;             %Beam mass Kg
K1=0.576;               %translational spring stifiness N/m
K2=K1;                  %translational spring stifiness N/m
K3=5.18;                %torsional spring stifiness N/m
C1=1.15e-5;             %translational dashpot N s/m
C2=C1;                  %translational dashpot N s/m
C3=2.88e-5;             %torsional dashpot N s/m
d=1.2e-3;               %Separation between the membranes
s=0.288e-6;             %Tympanum area m^2
f=1000:2250:20000;
freq_max=30000;
counter=1;
THETA=-90:2:90;         %Azimuth Angle Degrees
theta_max=90;
w=2*pi*f;               %Applied Frequency rad/sec
c=344;                  %Speed of Sound m/s
LUMBDA=c./f;            %Wave Length m
w1=(K1/m)^0.5;          %First Natural Frequency rad/sec
w2=((K1+2*K3)/m)^0.5;   %Second Natural Frequency rad/sec
OMEGA=w./w1;
AITA=w2/w1;
ZETA1=C1/(w1*m);
ZETA2=(C1+2*C3)/(w2*m);
gamma=ZETA2/ZETA1;
GAMMA=(1-OMEGA.^2+2*ZETA1.*OMEGA*1i)./(AITA^2-OMEGA.^2+2*AITA.*...
    gamma*ZETA1.*OMEGA*1i);
PSY_K=K3/K1;
PSY_L=d./LUMBDA;
PHI=2*pi*PSY_L.'*sin(THETA.*pi/180);

for q=1:length(f)
    for r=1:length(THETA)
```



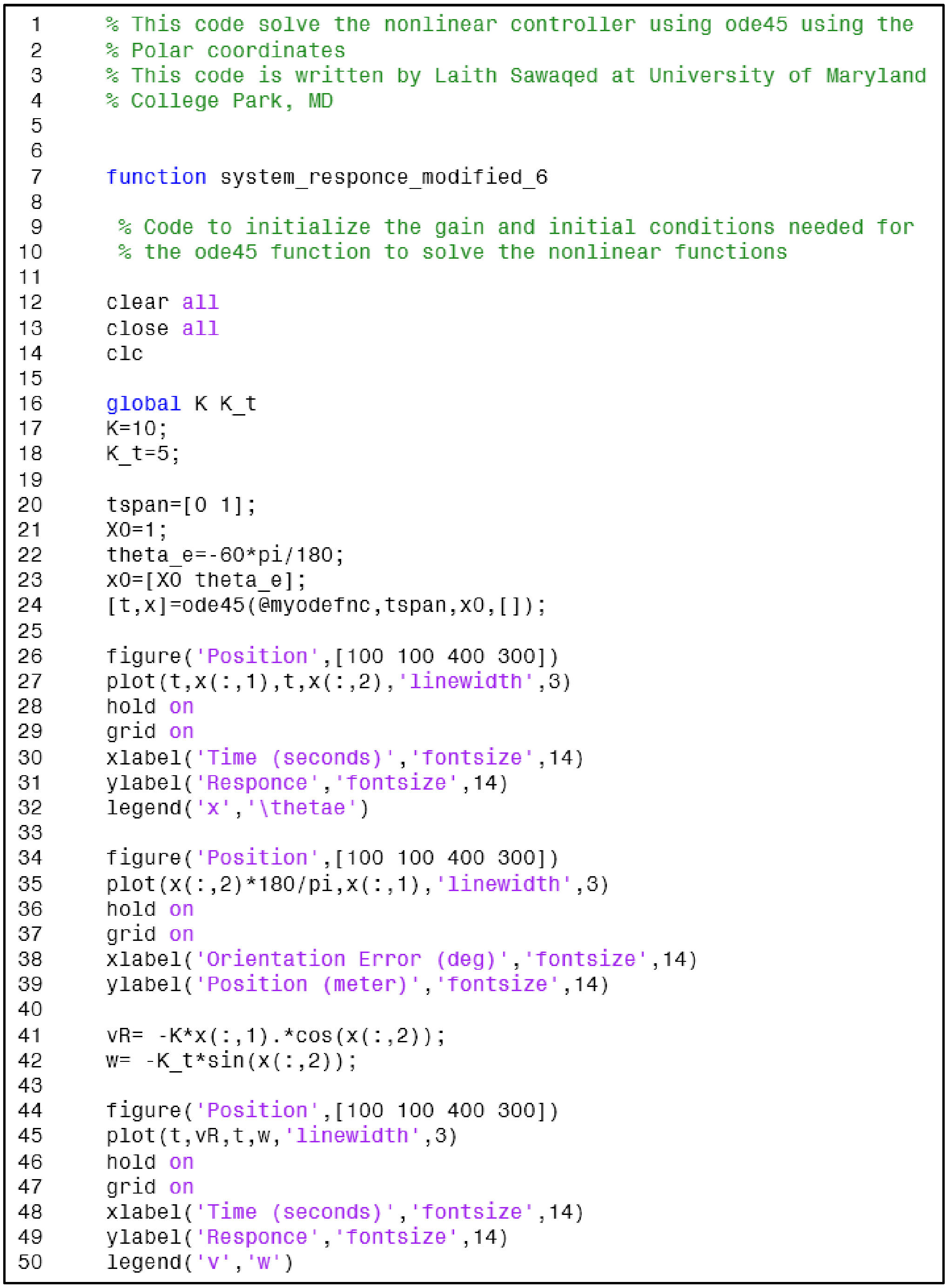
```matlab

A1=GAMMA(q)+tan(PHI(q,r)/2)*1i;
A2=GAMMA(q)-tan(PHI(q,r)/2)*1i;
mIPD(q,r)=angle(A1/A2)*180/pi;

    end
end

mIPD=mIPD.';
IPD_max = max(mIPD);   %Maximum IPD at the same frequency
for i=1:length(f)
aa(1:length(THETA),i)=f(i)/freq_max;
bb=round(10000*mIPD(:,i)./IPD_max(i)).'/10000;
Sound_Location= evalfis([aa(:,i),bb.'],fismat)*theta_max;
estimated_theta(:,i)=Sound_Location;

end

estimated_theta=estimated_theta.';
True_theta=THETA.';
estimated_theta=estimated_theta.';

for i=1:length(f)
Error(:,i)=abs(True_theta-estimated_theta(:,i));
Error_percentage(:,i) = abs((True_theta-estimated_theta(:,i))./...
    True_theta);
Ratio(:,i)=abs((estimated_theta(:,i)./True_theta)-1);
Results_1(:,:,i)=[True_theta,estimated_theta(:,i),Ratio(:,i)];
Results_2(:,:,i)=[True_theta,estimated_theta(:,i),Error(:,i),...
    Error_percentage(:,i)];

n=3;
m=3;

figure(1)
subplot(n,m,i)
plot(mIPD(:,i),THETA,'-b','linewidth',3)
hold on
plot(mIPD(:,i),estimated_theta(:,i),'-.r','linewidth',3)
grid on
xlabel('mIPD (Degrees)')
ylabel('Azimuth Angle (Degrees)')
title(['Frequency = ',num2str(f(i)),' Hz'])
legend('True Angle','Estimated Angle')

figure(2)
subplot(n,m,i)
errorbar(mIPD(:,i),estimated_theta(:,i),Error(:,i),'linewidth',3)
grid on
xlabel('mIPD (Degrees)')
```



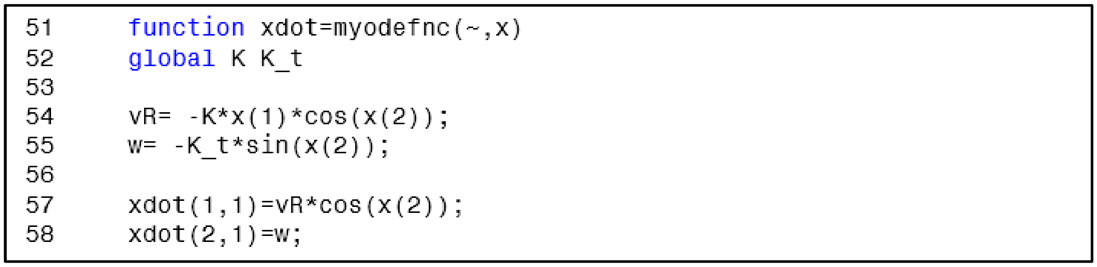
```matlab
ylabel('Azimuth Angle (Degrees)')
title(['Frequency = ',num2str(f(i)),' Hz'])
end
```

Appendix D Nonlinear Controllers Simulations

D.1 Testing the Nonlinear Controller Performance Using the Polar Coordinates



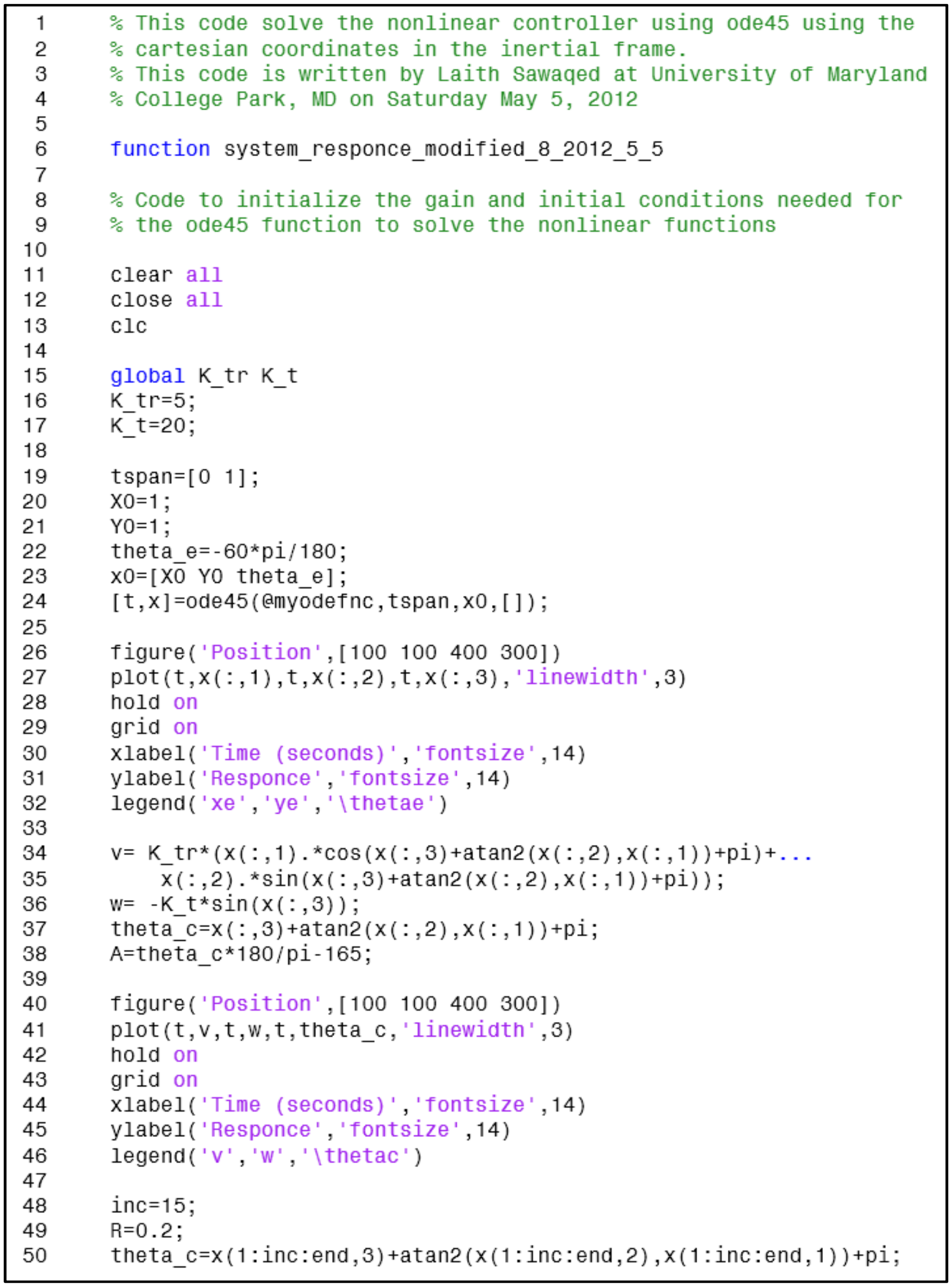
```
% This code solve the nonlinear controller using ode45 using the
% Polar coordinates
% This code is written by Laith Sawaqed at University of Maryland
% College Park, MD


function system_responce_modified_6

 % Code to initialize the gain and initial conditions needed for
 % the ode45 function to solve the nonlinear functions

clear all
close all
clc

global K K_t
K=10;
K_t=5;

tspan=[0 1];
X0=1;
theta_e=-60*pi/180;
x0=[X0 theta_e];
[t,x]=ode45(@myodefnc,tspan,x0,[]);

figure('Position',[100 100 400 300])
plot(t,x(:,1),t,x(:,2),'linewidth',3)
hold on
grid on
xlabel('Time (seconds)','fontsize',14)
ylabel('Responce','fontsize',14)
legend('x','\thetae')

figure('Position',[100 100 400 300])
plot(x(:,2)*180/pi,x(:,1),'linewidth',3)
hold on
grid on
xlabel('Orientation Error (deg)','fontsize',14)
ylabel('Position (meter)','fontsize',14)

vR= -K*x(:,1).*cos(x(:,2));
w= -K_t*sin(x(:,2));

figure('Position',[100 100 400 300])
plot(t,vR,t,w,'linewidth',3)
hold on
grid on
xlabel('Time (seconds)','fontsize',14)
ylabel('Responce','fontsize',14)
legend('v','w')
```



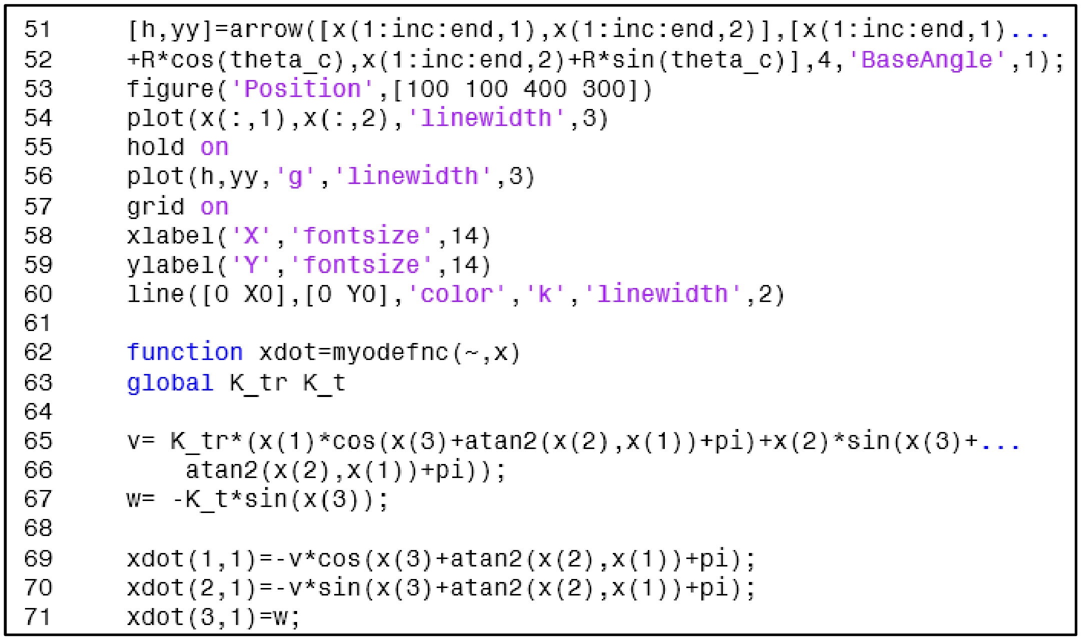
```
function xdot=myodefnc(~,x)
global K K_t

vR= -K*x(1)*cos(x(2));
w= -K_t*sin(x(2));

xdot(1,1)=vR*cos(x(2));
xdot(2,1)=w;
```

D.2 Testing the Nonlinear Controller Performance Using the Cartesian Coordinates

D.2.i) Using the ODE45 Function



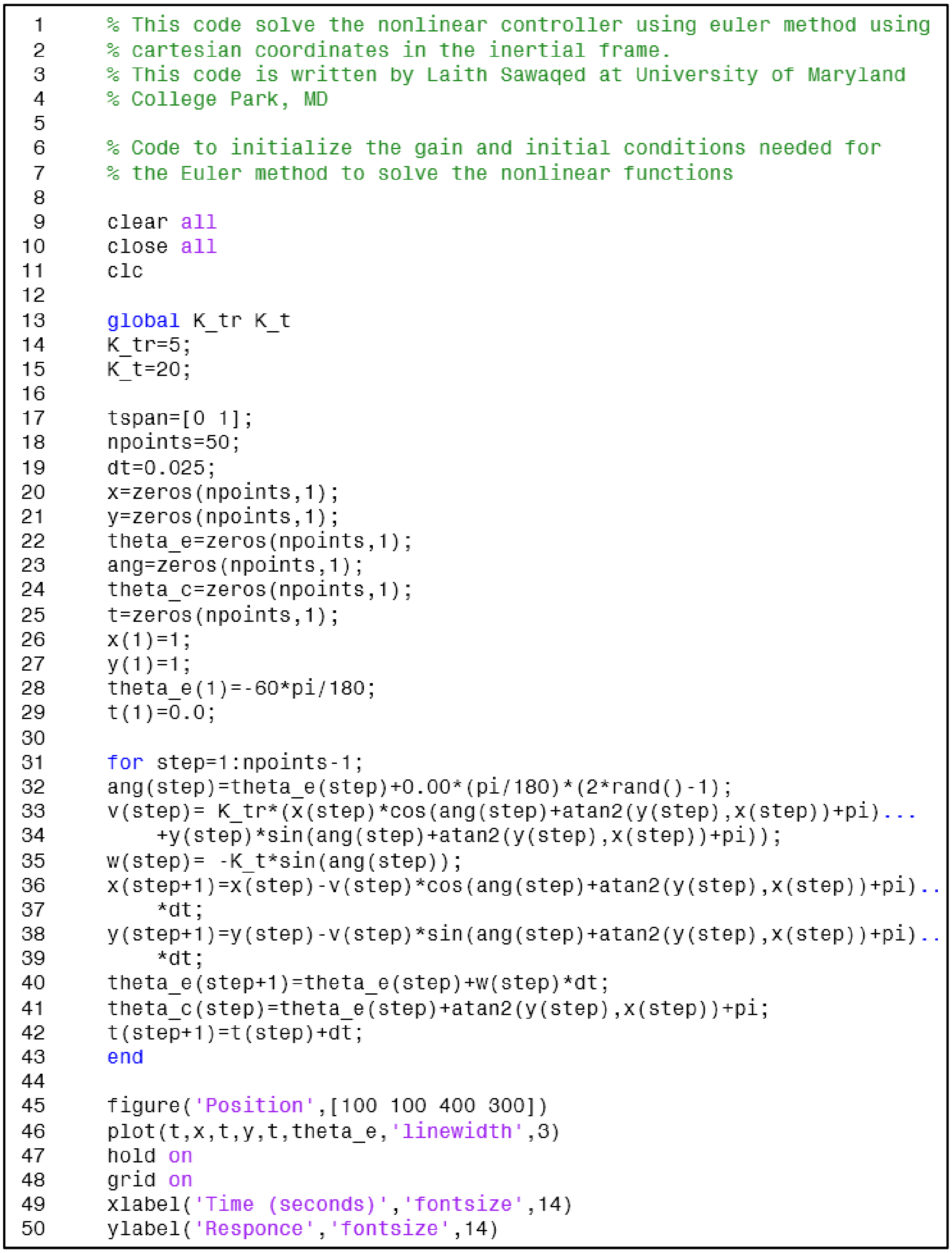
```matlab
% This code solve the nonlinear controller using ode45 using the
% cartesian coordinates in the inertial frame.
% This code is written by Laith Sawaqed at University of Maryland
% College Park, MD on Saturday May 5, 2012

function system_responce_modified_8_2012_5_5

% Code to initialize the gain and initial conditions needed for
% the ode45 function to solve the nonlinear functions

clear all
close all
clc

global K_tr K_t
K_tr=5;
K_t=20;

tspan=[0 1];
X0=1;
Y0=1;
theta_e=-60*pi/180;
x0=[X0 Y0 theta_e];
[t,x]=ode45(@myodefnc,tspan,x0,[]);

figure('Position',[100 100 400 300])
plot(t,x(:,1),t,x(:,2),t,x(:,3),'linewidth',3)
hold on
grid on
xlabel('Time (seconds)','fontsize',14)
ylabel('Responce','fontsize',14)
legend('xe','ye','\thetae')

v= K_tr*(x(:,1).*cos(x(:,3)+atan2(x(:,2),x(:,1))+pi)+...
    x(:,2).*sin(x(:,3)+atan2(x(:,2),x(:,1))+pi));
w= -K_t*sin(x(:,3));
theta_c=x(:,3)+atan2(x(:,2),x(:,1))+pi;
A=theta_c*180/pi-165;

figure('Position',[100 100 400 300])
plot(t,v,t,w,t,theta_c,'linewidth',3)
hold on
grid on
xlabel('Time (seconds)','fontsize',14)
ylabel('Responce','fontsize',14)
legend('v','w','\thetac')

inc=15;
R=0.2;
theta_c=x(1:inc:end,3)+atan2(x(1:inc:end,2),x(1:inc:end,1))+pi;
```



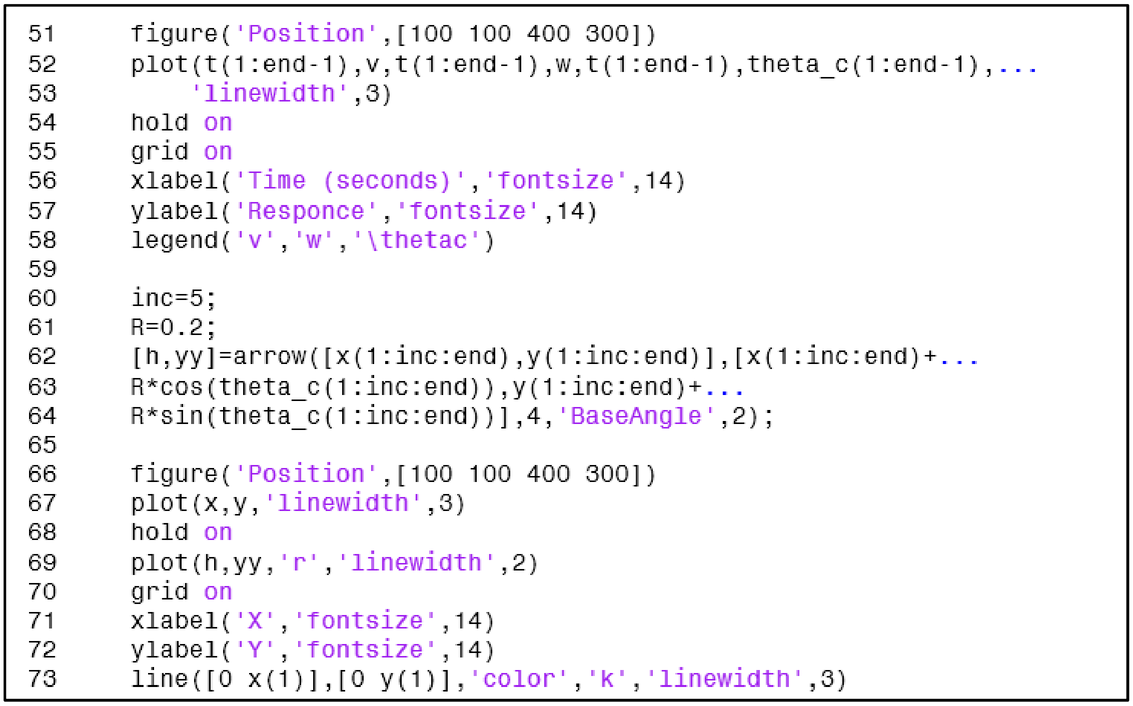
```
[h,yy]=arrow([x(1:inc:end,1),x(1:inc:end,2)],[x(1:inc:end,1)...
+R*cos(theta_c),x(1:inc:end,2)+R*sin(theta_c)],4,'BaseAngle',1);
figure('Position',[100 100 400 300])
plot(x(:,1),x(:,2),'linewidth',3)
hold on
plot(h,yy,'g','linewidth',3)
grid on
xlabel('X','fontsize',14)
ylabel('Y','fontsize',14)
line([0 X0],[0 Y0],'color','k','linewidth',2)

function xdot=myodefnc(~,x)
global K_tr K_t

v= K_tr*(x(1)*cos(x(3)+atan2(x(2),x(1))+pi)+x(2)*sin(x(3)+...
    atan2(x(2),x(1))+pi));
w= -K_t*sin(x(3));

xdot(1,1)=-v*cos(x(3)+atan2(x(2),x(1))+pi);
xdot(2,1)=-v*sin(x(3)+atan2(x(2),x(1))+pi);
xdot(3,1)=w;
```

D.2.ii) Using Euler Method

To add the measurement noise just change the zero multiplied by the random function in line 32

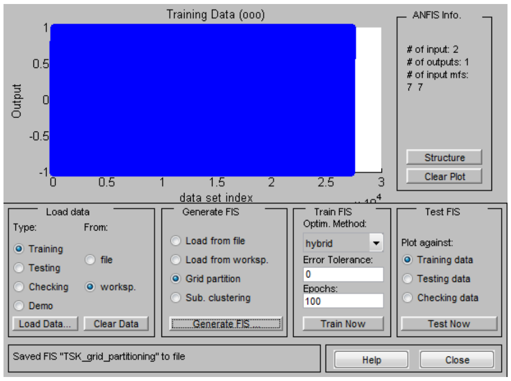


```
1      % This code solve the nonlinear controller using euler method using
2      % cartesian coordinates in the inertial frame.
3      % This code is written by Laith Sawaqed at University of Maryland
4      % College Park, MD
5
6      % Code to initialize the gain and initial conditions needed for
7      % the Euler method to solve the nonlinear functions
8
9      clear all
10     close all
11     clc
12
13     global K_tr K_t
14     K_tr=5;
15     K_t=20;
16
17     tspan=[0 1];
18     npoints=50;
19     dt=0.025;
20     x=zeros(npoints,1);
21     y=zeros(npoints,1);
22     theta_e=zeros(npoints,1);
23     ang=zeros(npoints,1);
24     theta_c=zeros(npoints,1);
25     t=zeros(npoints,1);
26     x(1)=1;
27     y(1)=1;
28     theta_e(1)=-60*pi/180;
29     t(1)=0.0;
30
31     for step=1:npoints-1;
32     ang(step)=theta_e(step)+0.00*(pi/180)*(2*rand()-1);
33     v(step)= K_tr*(x(step)*cos(ang(step)+atan2(y(step),x(step))+pi)...
34         +y(step)*sin(ang(step)+atan2(y(step),x(step))+pi));
35     w(step)= -K_t*sin(ang(step));
36     x(step+1)=x(step)-v(step)*cos(ang(step)+atan2(y(step),x(step))+pi)..
37         *dt;
38     y(step+1)=y(step)-v(step)*sin(ang(step)+atan2(y(step),x(step))+pi)..
39         *dt;
40     theta_e(step+1)=theta_e(step)+w(step)*dt;
41     theta_c(step)=theta_e(step)+atan2(y(step),x(step))+pi;
42     t(step+1)=t(step)+dt;
43     end
44
45     figure('Position',[100 100 400 300])
46     plot(t,x,t,y,t,theta_e,'linewidth',3)
47     hold on
48     grid on
49     xlabel('Time (seconds)','fontsize',14)
50     ylabel('Responce','fontsize',14)
```



```
figure('Position',[100 100 400 300])
plot(t(1:end-1),v,t(1:end-1),w,t(1:end-1),theta_c(1:end-1),...
    'linewidth',3)
hold on
grid on
xlabel('Time (seconds)','fontsize',14)
ylabel('Responce','fontsize',14)
legend('v','w','\thetac')

inc=5;
R=0.2;
[h,yy]=arrow([x(1:inc:end),y(1:inc:end)],[x(1:inc:end)+...
R*cos(theta_c(1:inc:end)),y(1:inc:end)+...
R*sin(theta_c(1:inc:end))],4,'BaseAngle',2);

figure('Position',[100 100 400 300])
plot(x,y,'linewidth',3)
hold on
plot(h,yy,'r','linewidth',2)
grid on
xlabel('X','fontsize',14)
ylabel('Y','fontsize',14)
line([0 x(1)],[0 y(1)],'color','k','linewidth',3)
```

Appendix E ANFIS and Fuzzy Toolbox GUI's



# of input: 2
# of outputs: 1
# of input mfs:
7 7

Type: ● Training ○ Testing ○ Checking ○ Demo

From: ○ file ● worksp.

○ Load from file
○ Load from worksp.
● Grid partition
○ Sub. clustering

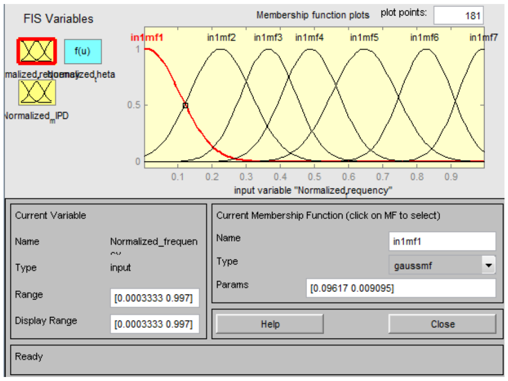
Optim. Method: hybrid

Error Tolerance: 0

Epochs: 100

Plot against: ● Training data ○ Testing data ○ Checking data

Saved FIS "TSK_grid_partitioning" to file



plot points: 181

Name: Normalized_frequency

Type: input

Range: [0.0003333 0.997]

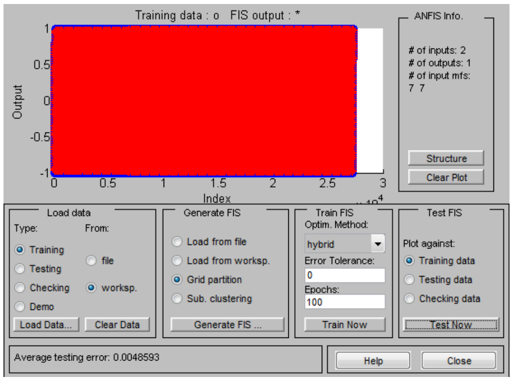
Display Range: [0.0003333 0.997]

Name: in1mf1

Type: gaussmf

Params: [0.09617 0.009095]

Ready



# of inputs: 2
# of outputs: 1
# of input mfs:
7 7

Type: ● Training ○ Testing ○ Checking ○ Demo

From: ○ file ● worksp.

○ Load from file
○ Load from worksp.
● Grid partition
○ Sub. clustering

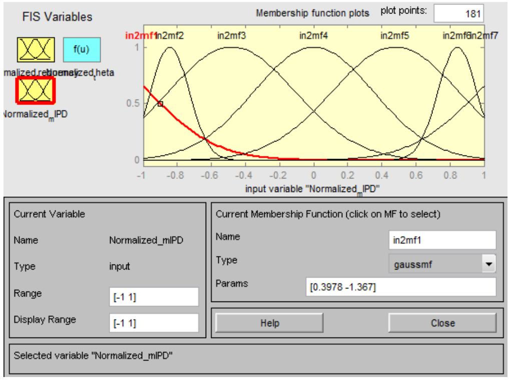
Optim. Method: hybrid

Error Tolerance: 0

Epochs: 100

Plot against: ● Training data ○ Testing data ○ Checking data

Average testing error: 0.0048593



plot points: 181

Name: Normalized_mIPD

Type: input

Range: [-1 1]

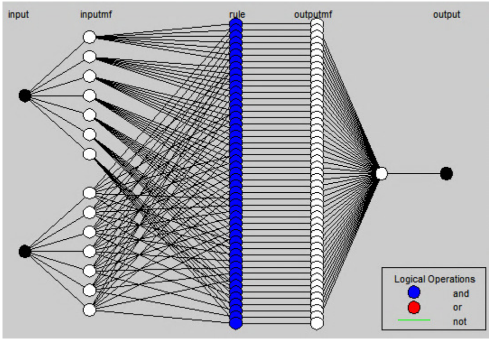
Display Range: [-1 1]

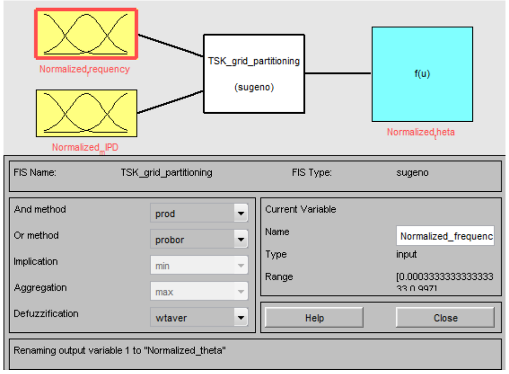
Name: in2mf1

Type: gaussmf

Params: [0.3978 -1.367]

Selected variable "Normalized_mIPD"





FIS Name: TSK_grid_partitioning

FIS Type: sugeno

And method: prod

Or method: probor

Implication: min

Aggregation: max

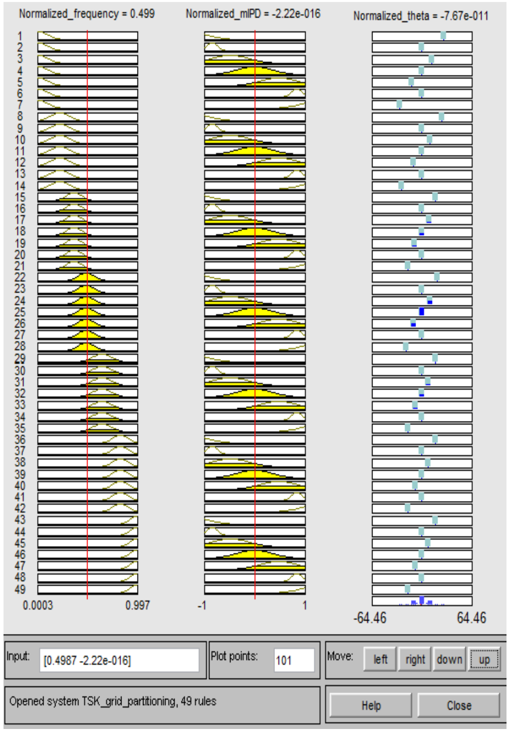
Defuzzification: wtaver

Name: Normalized_frequenc

Type: input

Range: [0.000333333333333333 0.9971

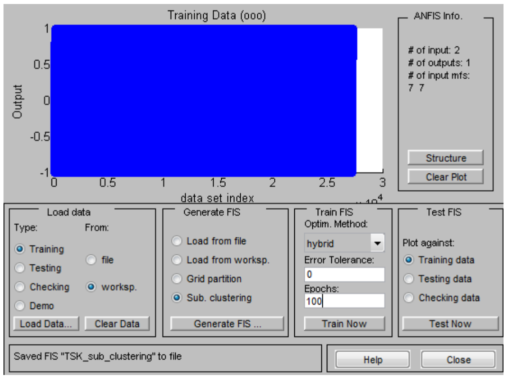
Renaming output variable 1 to "Normalized_theta"



Input: [0.4987 -2.22e-016]

Plot points: 101

Opened system TSK_grid_partitioning, 49 rules



# of input: 2
# of outputs: 1
# of input mfs:
7 7

Type: ● Training ○ Testing ○ Checking ○ Demo

From: ○ file ● worksp.

● Sub. clustering

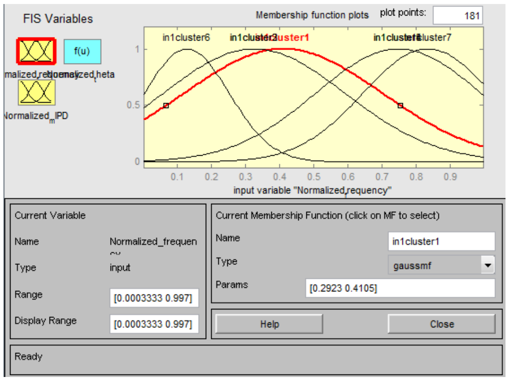
Optim. Method: hybrid

Error Tolerance: 0

Epochs: 100

Plot against: ● Training data

Saved FIS "TSK_sub_clustering" to file



plot points: 181

input variable "Normalized_frequency"

Current Variable

Name Normalized_frequency

Type input

Range [0.0003333 0.997]

Display Range [0.0003333 0.997]

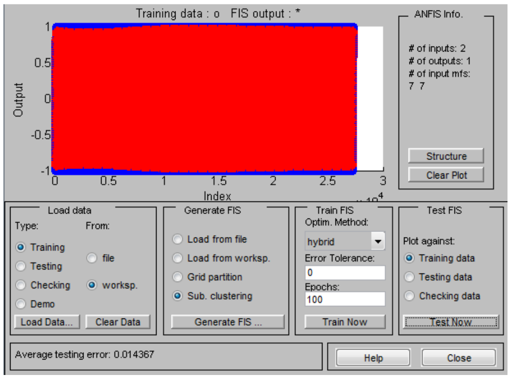
Current Membership Function (click on MF to select)

Name in1cluster1

Type gaussmf

Params [0.2923 0.4105]

Ready



# of inputs: 2
# of outputs: 1
# of input mfs:
7 7

Type: ● Training ○ Testing ○ Checking ○ Demo

From: ○ file ● worksp.

● Sub. clustering

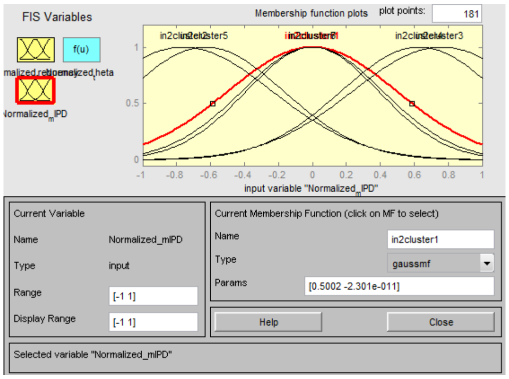
Optim. Method: hybrid

Error Tolerance: 0

Epochs: 100

Plot against: ● Training data

Average testing error: 0.014367



plot points: 181

input variable "Normalized_mIPD"

Current Variable

Name Normalized_mIPD

Type input

Range [-1 1]

Display Range [-1 1]

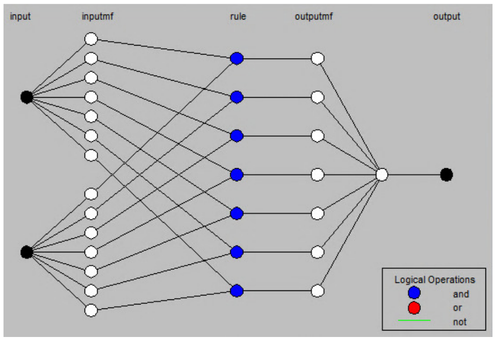
Current Membership Function (click on MF to select)

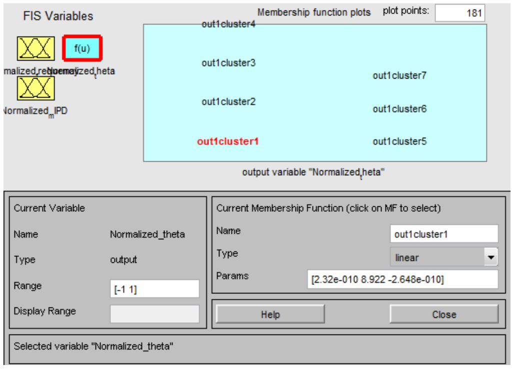
Name in2cluster1

Type gaussmf

Params [0.5002 -2.301e-011]

Selected variable "Normalized_mIPD"





plot points: 181

output variable "Normalized_theta"

Current Variable

Name Normalized_theta

Type output

Range [-1 1]

Display Range

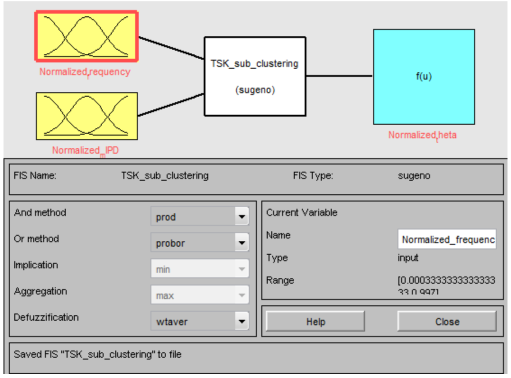
Current Membership Function (click on MF to select)

Name out1cluster1

Type linear

Params [2.32e-010 8.922 -2.648e-010]

Selected variable "Normalized_theta"



FIS Name: TSK_sub_clustering

FIS Type: sugeno

And method prod

Or method probor

Implication min

Aggregation max

Defuzzification wtaver

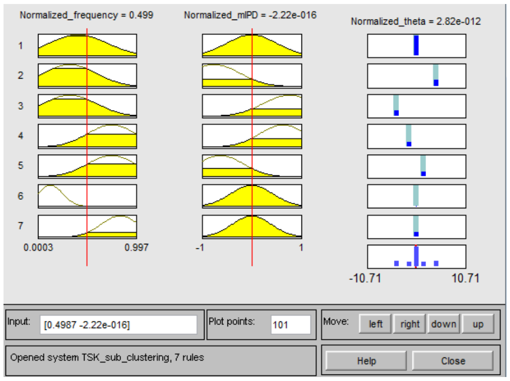
Current Variable

Name Normalized_frequenc

Type input

Range [0.000333333333333333 0.9971

Saved FIS "TSK_sub_clustering" to file


Normalized_frequency = 0.499
Normalized_mIPD = -2.22e-016
Normalized_theta = 2.82e-012
1
2
3
4
5
6
7
0.0003
0.997
-1
1
-10.71
10.71
Move:
left
right
down
up
Help
Close

Input: [0.4987 -2.22e-016]

Plot points: 101

Opened system TSK_sub_clustering, 7 rules

Appendix F   Fuzzy models incident angle estimation

F.1 Theoretical model

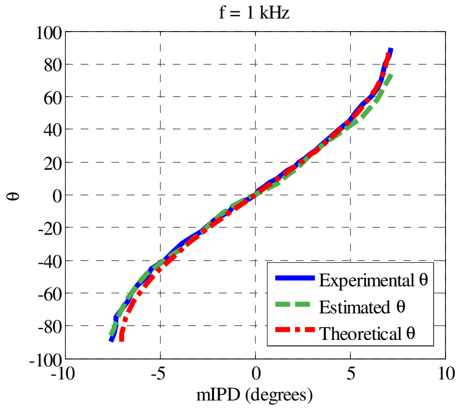


Figure F.1: Estimated vs. Experimental and Theoretical incident angle θ at f = 1 kHz

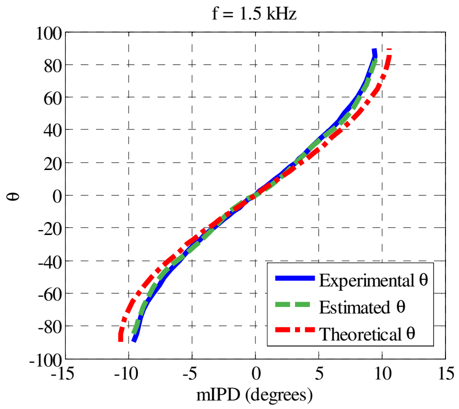


Figure F.2: Estimated vs. Experimental and Theoretical incident angle θ at f = 1.5 kHz

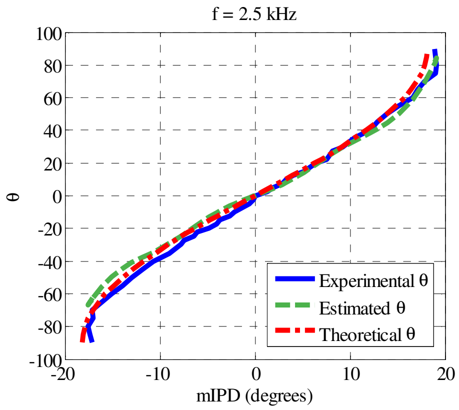


Figure F.3: Estimated vs. Experimental and Theoretical incident angle θ at f = 2.5 kHz

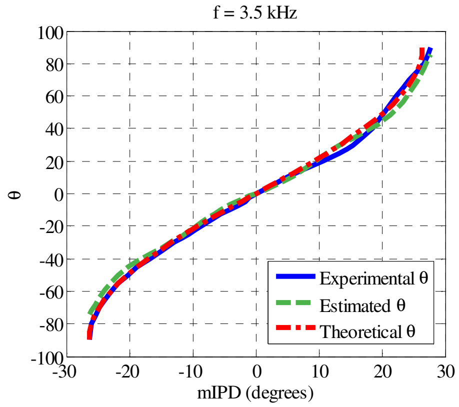


Figure F.4: Estimated vs. Experimental and Theoretical incident angle θ at f = 3.5 kHz

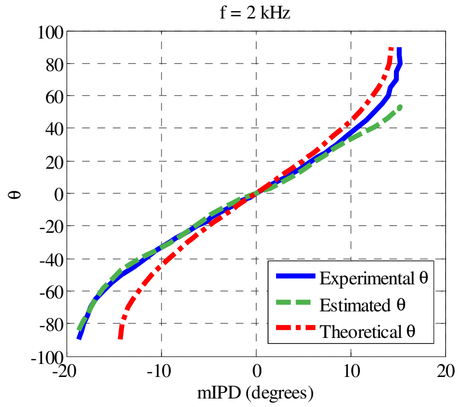


Figure F.5: Estimated vs. Experimental and Theoretical incident angle θ at f = 2 kHz

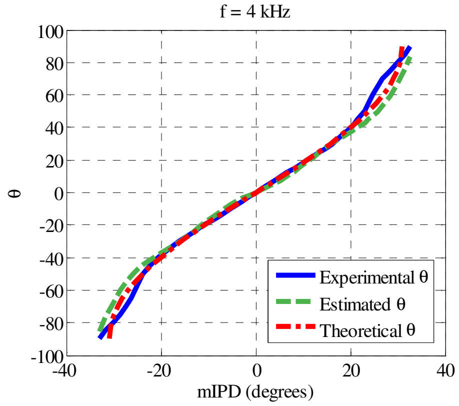


Figure F.6: Estimated vs. Experimental and Theoretical incident angle θ at f = 4 kHz

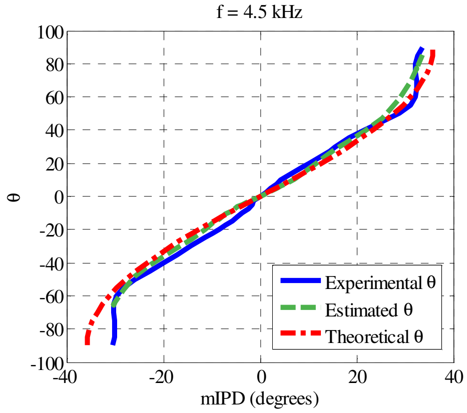


Figure F.7: Estimated vs. Experimental and Theoretical incident angle θ at f = 4.5 kHz

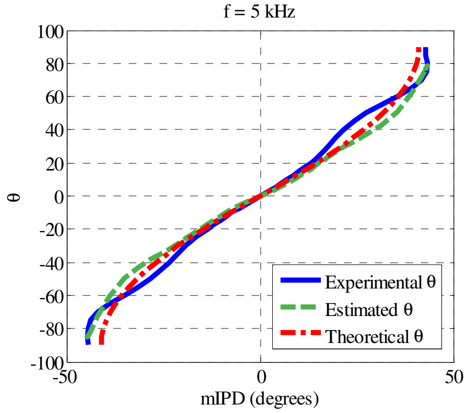


Figure F.8: Estimated vs. Experimental and Theoretical incident angle θ at f = 5 kHz

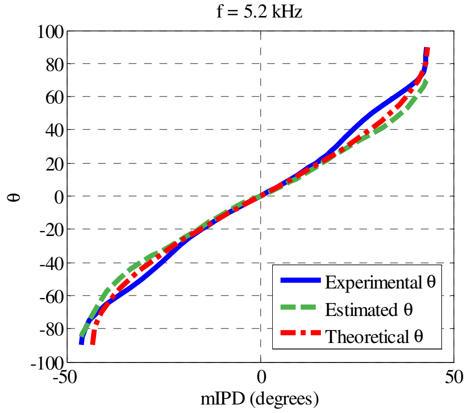


Figure F.9: Estimated vs. Experimental and Theoretical incident angle θ at f = 5.2 kHz

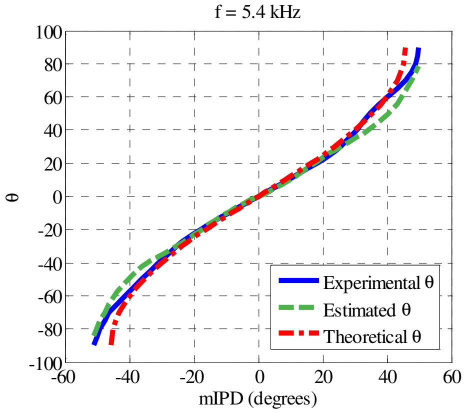


Figure F.10: Estimated vs. Experimental and Theoretical incident angle θ at f = 5.4 kHz

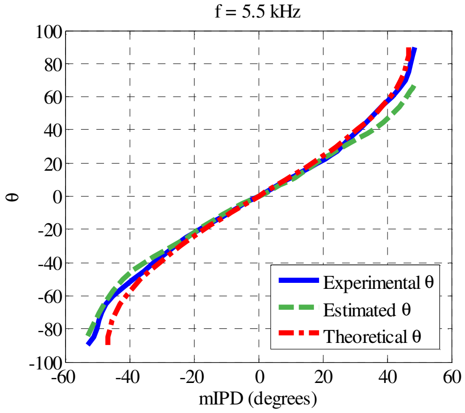


Figure F.11: Estimated vs. Experimental and Theoretical incident angle θ at f = 5.5 kHz

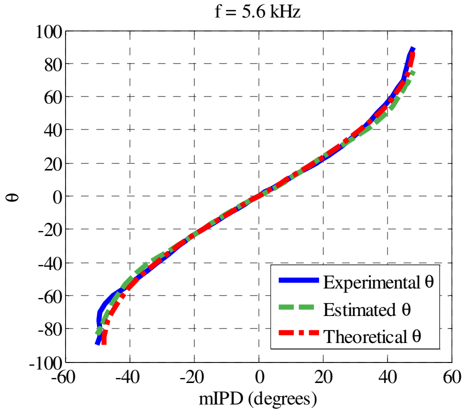


Figure F.12: Estimated vs. Experimental and Theoretical incident angle θ at f = 5.6 kHz

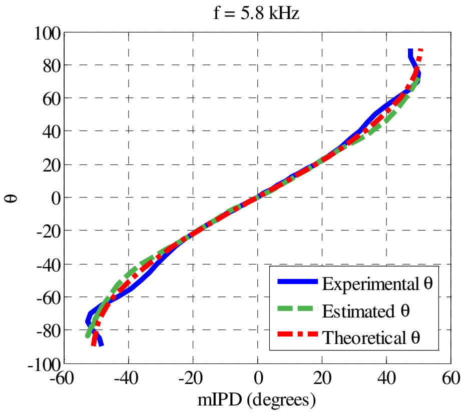


Figure F.13: Estimated vs. Experimental and Theoretical incident angle θ at f = 5.8 kHz

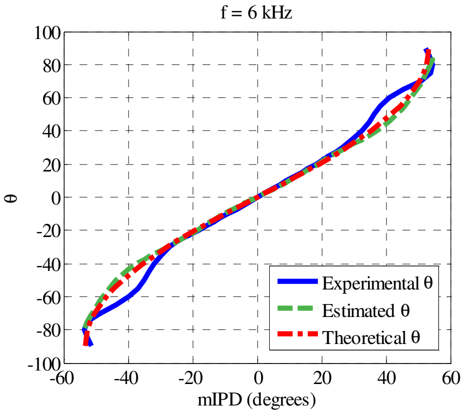


Figure F.14: Estimated vs. Experimental and Theoretical incident angle θ at f = 6 kHz

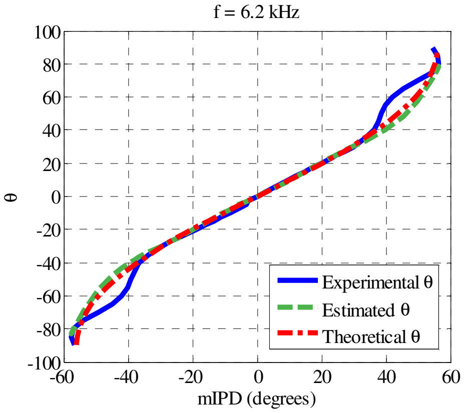


Figure F.15: Estimated vs. Experimental and Theoretical incident angle θ at f = 6.2 kHz

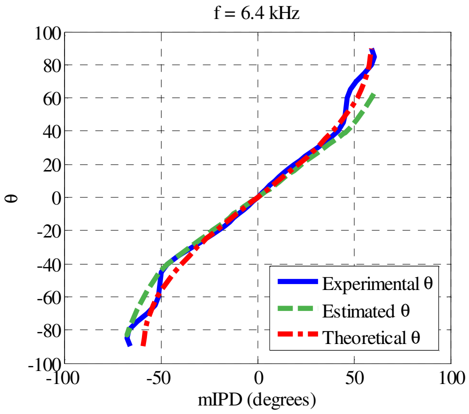


Figure F.16: Estimated vs. Experimental and Theoretical incident angle θ at f = 6.4 kHz

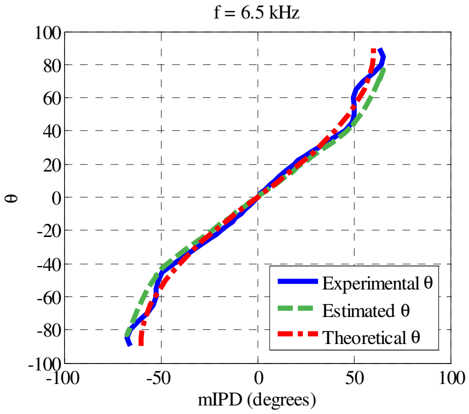


Figure F.17: Estimated vs. Experimental and Theoretical incident angle θ at f = 6.5 kHz

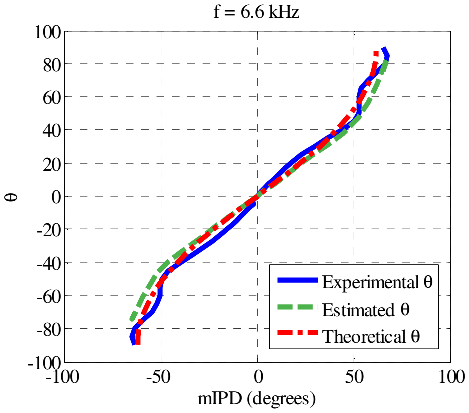


Figure F.18: Estimated vs. Experimental and Theoretical incident angle θ at f = 6.6 kHz

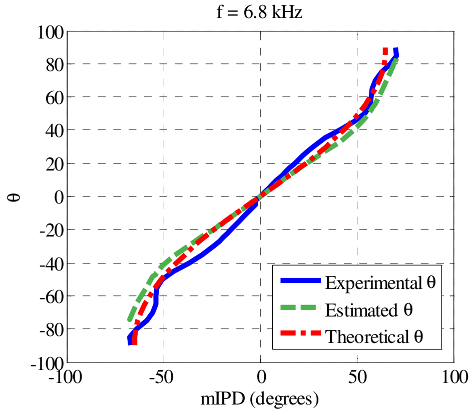


Figure F.19: Estimated vs. Experimental and Theoretical incident angle θ at f = 6.8 kHz

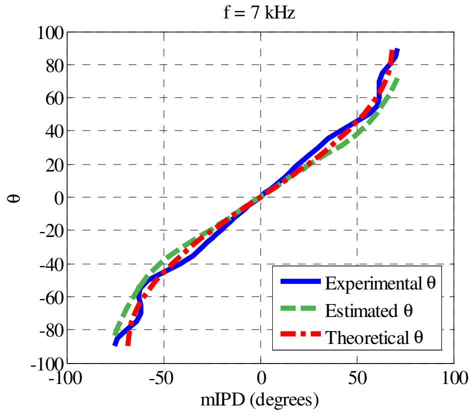


Figure F.20: Estimated vs. Experimental and Theoretical incident angle θ at f = 7 kHz

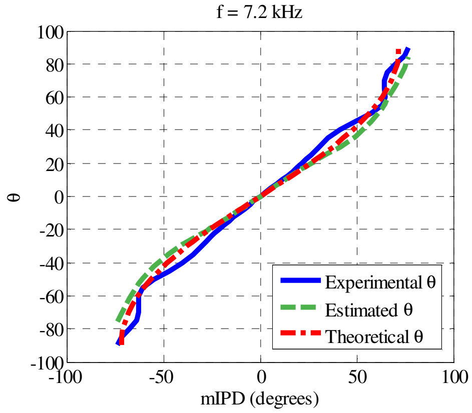


Figure F.21: Estimated vs. Experimental and Theoretical incident angle θ at f = 7.2 kHz

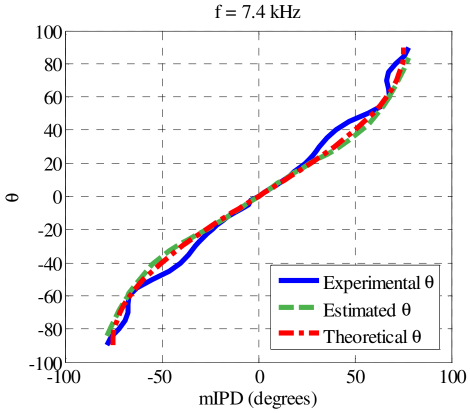


Figure F.22: Estimated vs. Experimental and Theoretical incident angle θ at f = 7.4 kHz

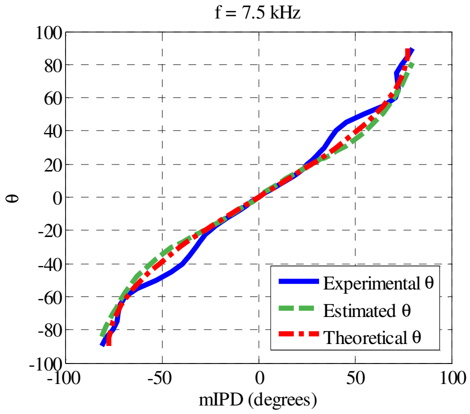


Figure F.23: Estimated vs. Experimental and Theoretical incident angle θ at f = 7.5 kHz

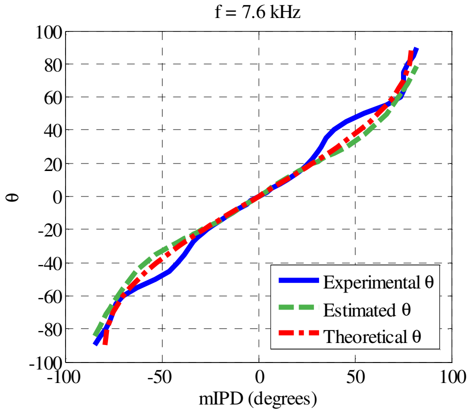


Figure F.24: Estimated vs. Experimental and Theoretical incident angle θ at f = 7.6 kHz

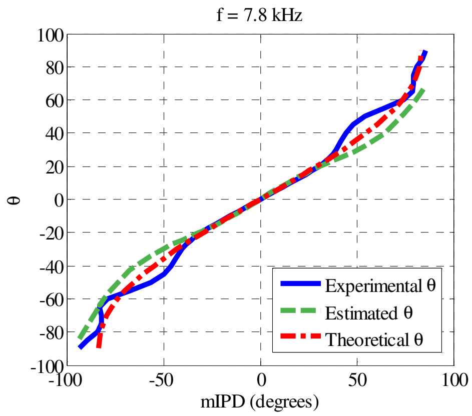


Figure F.25: Estimated vs. Experimental and Theoretical incident angle θ at f = 7.8 kHz

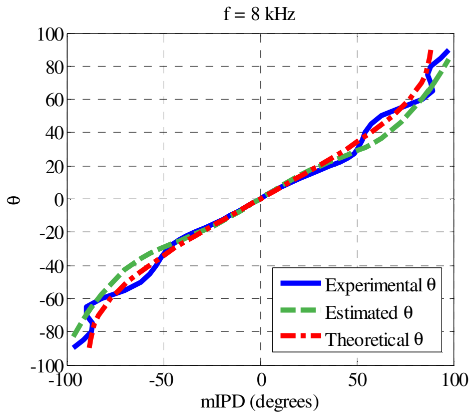


Figure F.26: Estimated vs. Experimental and Theoretical incident angle θ at f = 8 kHz

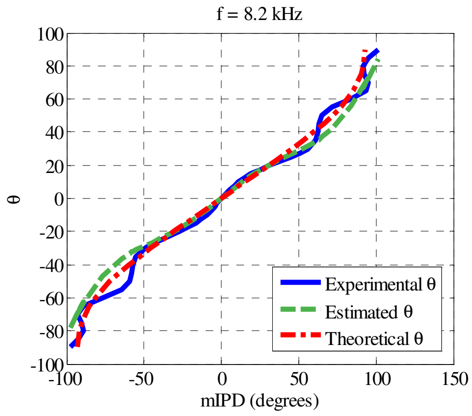


Figure F.27: Estimated vs. Experimental and Theoretical incident angle θ at f = 8.2 kHz

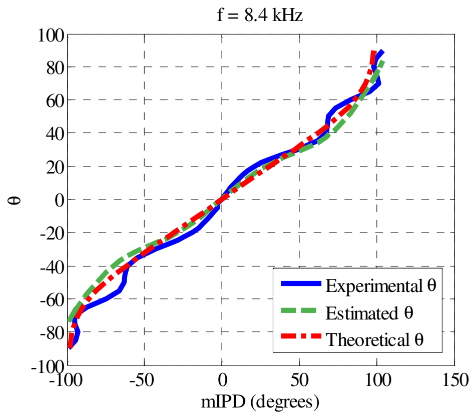


Figure F.28: Estimated vs. Experimental and Theoretical incident angle θ at f = 8.4 kHz

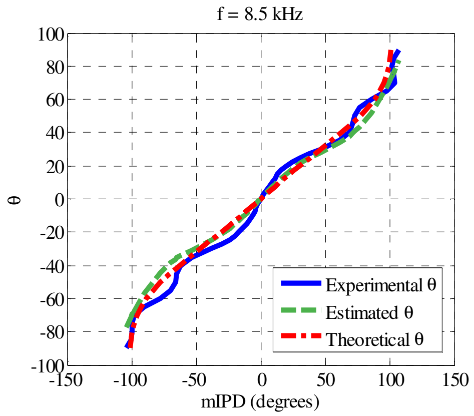


Figure F.29: Estimated vs. Experimental and Theoretical incident angle θ at f = 8.5 kHz

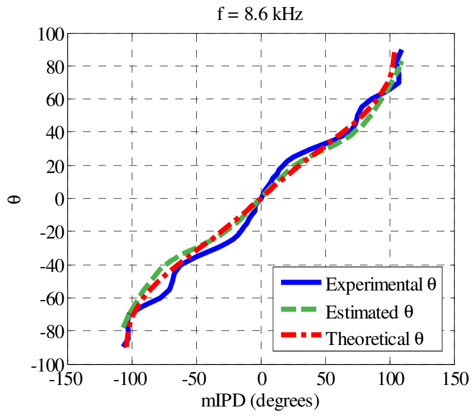


Figure F.30: Estimated vs. Experimental and Theoretical incident angle θ at f = 8.6 kHz

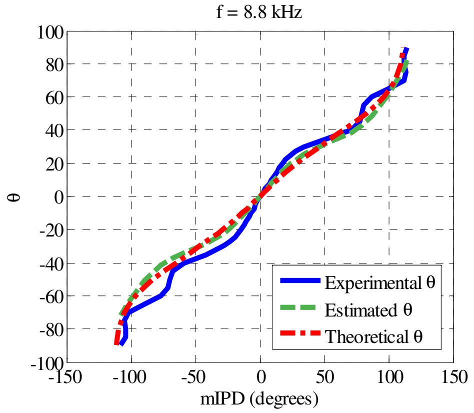


Figure F.31: Estimated vs. Experimental and Theoretical incident angle θ at f = 8.8 kHz

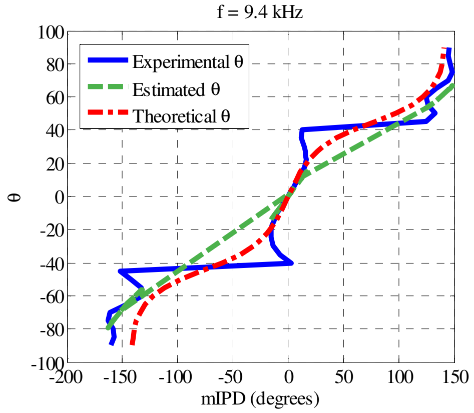


Figure F.34: Estimated vs. Experimental and Theoretical incident angle θ at f = 9.4 kHz

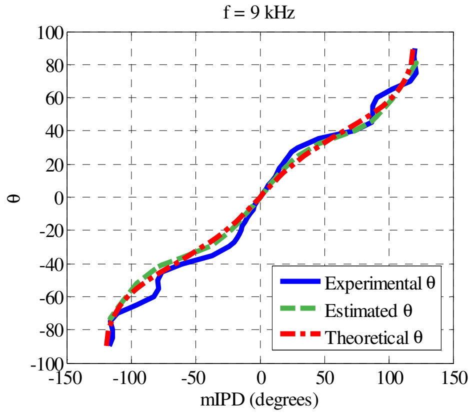


Figure F.32: Estimated vs. Experimental and Theoretical incident angle θ at f = 9 kHz

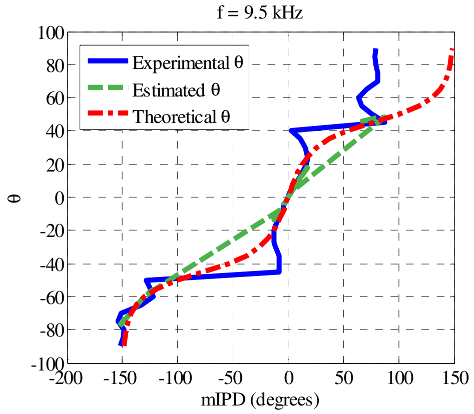


Figure F.35: Estimated vs. Experimental and Theoretical incident angle θ at f = 9.5 kHz

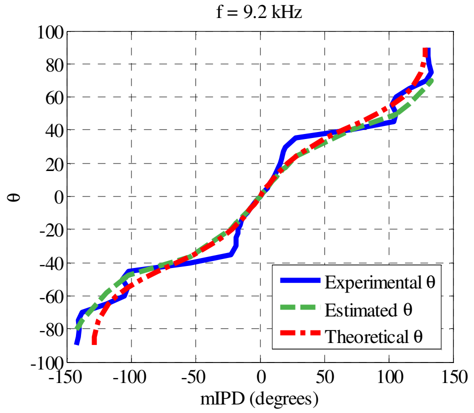


Figure F.33: Estimated vs. Experimental and Theoretical incident angle θ at f = 9.2 kHz

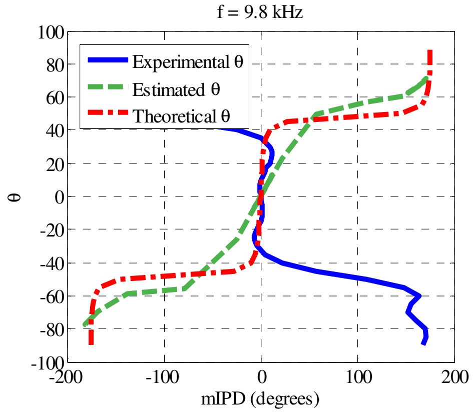


Figure F.36: Estimated vs. Experimental and Theoretical incident angle θ at f = 9.8 kHz

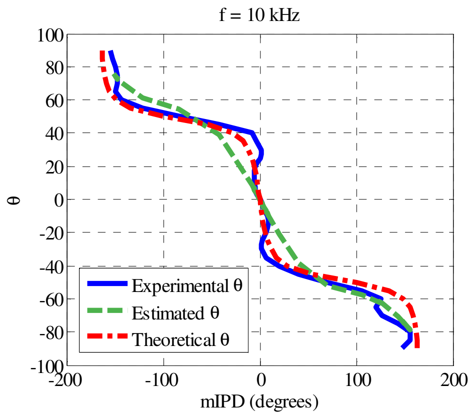


Figure F.37: Estimated vs. Experimental and Theoretical incident angle θ at f = 10 kHz

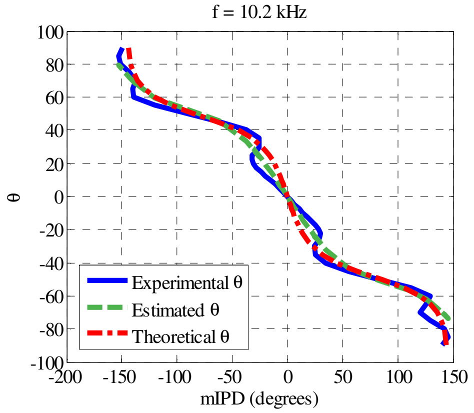


Figure F.38: Estimated vs. Experimental and Theoretical incident angle θ at f = 10.2 kHz

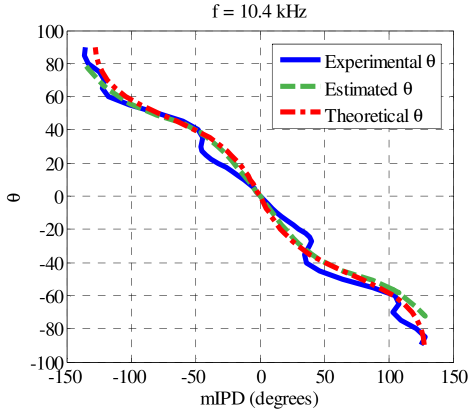


Figure F.39: Estimated vs. Experimental and Theoretical incident angle θ at f = 10.4 kHz

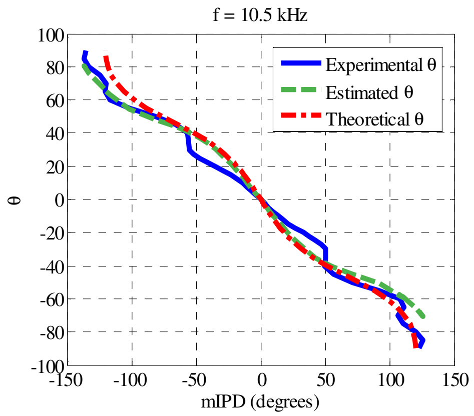


Figure F.40: Estimated vs. Experimental and Theoretical incident angle θ at f = 10.5 kHz

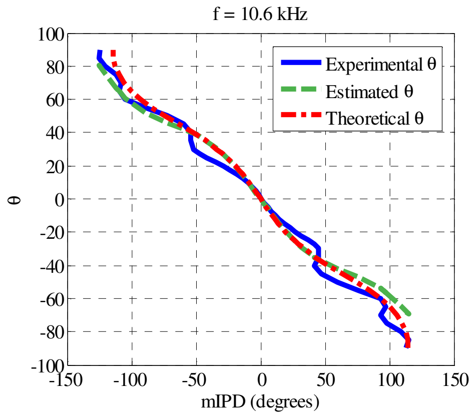


Figure F.41: Estimated vs. Experimental and Theoretical incident angle θ at f = 10.6 kHz

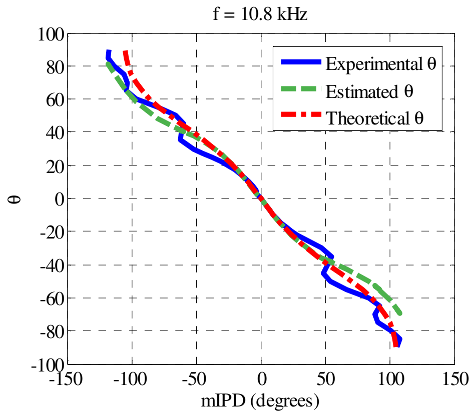


Figure F.42: Estimated vs. Experimental and Theoretical incident angle θ at f = 10.8 kHz

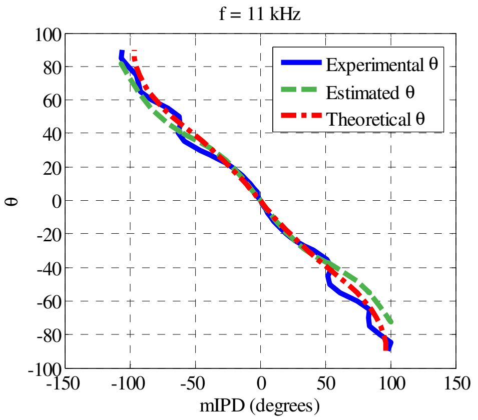


Figure F.43: Estimated vs. Experimental and Theoretical incident angle θ at f = 11 kHz

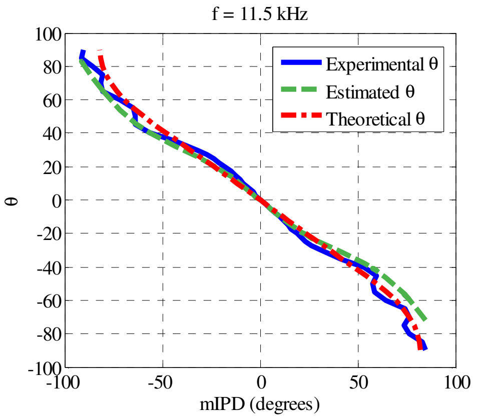


Figure F.44: Estimated vs. Experimental and Theoretical incident angle θ at f = 11.5 kHz

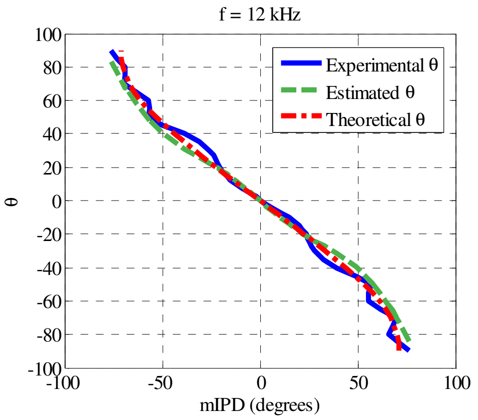


Figure F.45: Estimated vs. Experimental and Theoretical incident angle θ at f = 12 kHz

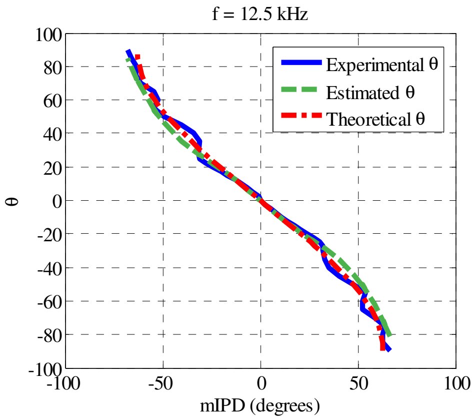


Figure F.46: Estimated vs. Experimental and Theoretical incident angle θ at f = 12.5 kHz

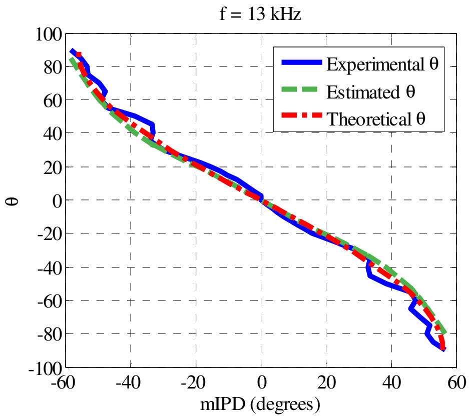


Figure F.47: Estimated vs. Experimental and Theoretical incident angle θ at f = 13 kHz

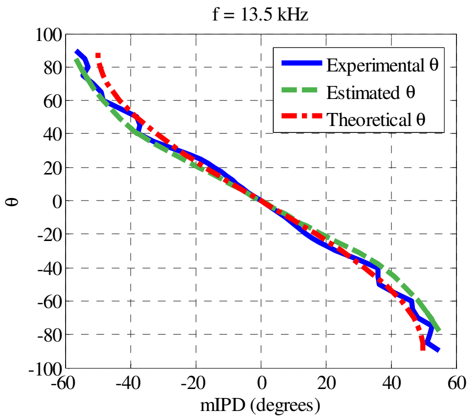


Figure F.48: Estimated vs. Experimental and Theoretical incident angle θ at f = 13.5 kHz

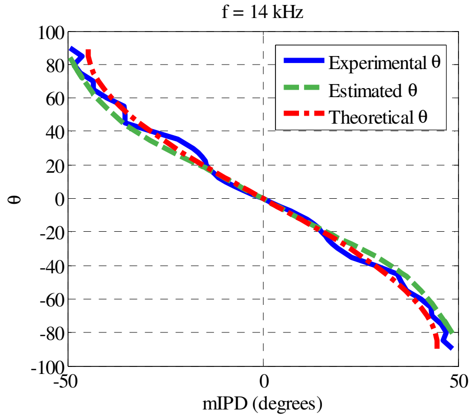


Figure F.49: Estimated vs. Experimental and Theoretical incident angle θ at f = 14 kHz

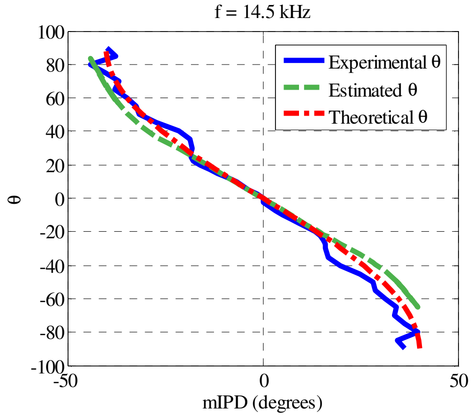


Figure F.50: Estimated vs. Experimental and Theoretical incident angle θ at f = 14.5 kHz

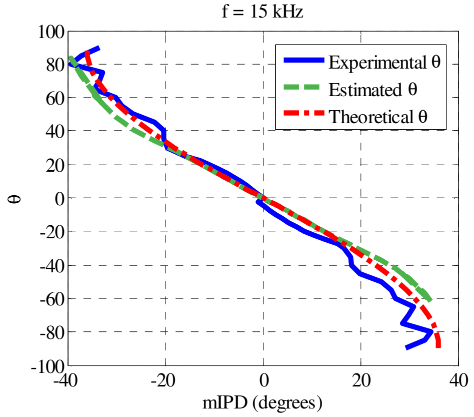


Figure F.51: Estimated vs. Experimental and Theoretical incident angle θ at f = 15 kHz

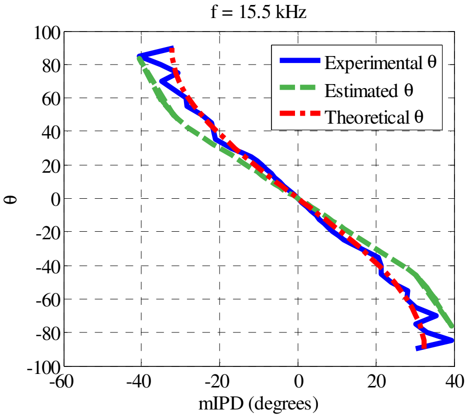


Figure F.52: Estimated vs. Experimental and Theoretical incident angle θ at f = 15.5 kHz

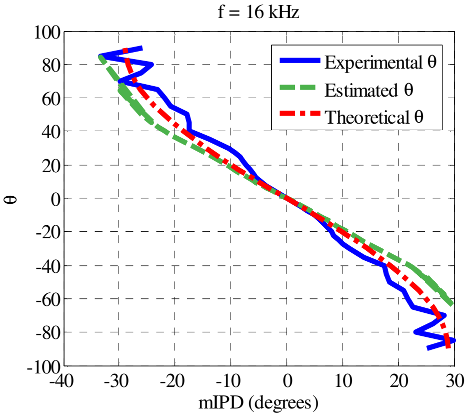


Figure F.53: Estimated vs. Experimental and Theoretical incident angle θ at f = 16 kHz

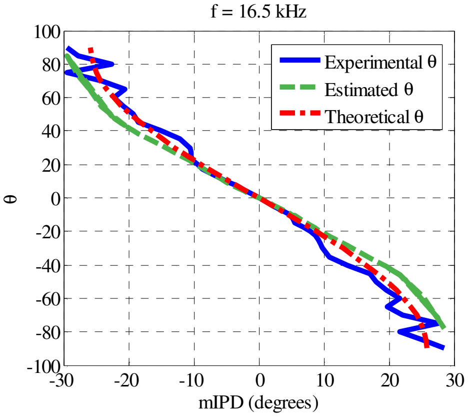


Figure F.54: Estimated vs. Experimental and Theoretical incident angle θ at f = 16.5 kHz

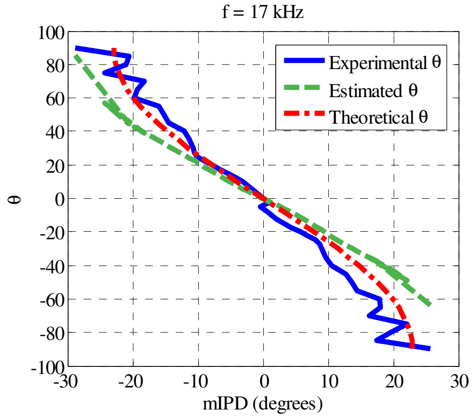


Figure F.55: Estimated vs. Experimental and Theoretical incident angle θ at f = 17 kHz

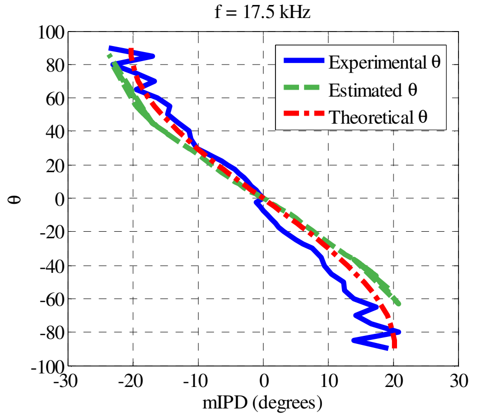


Figure F.56: Estimated vs. Experimental and Theoretical incident angle θ at f = 17.5 kHz

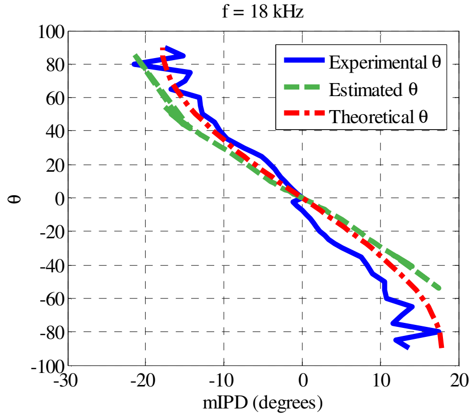


Figure F.57: Estimated vs. Experimental and Theoretical incident angle θ at f = 18 kHz

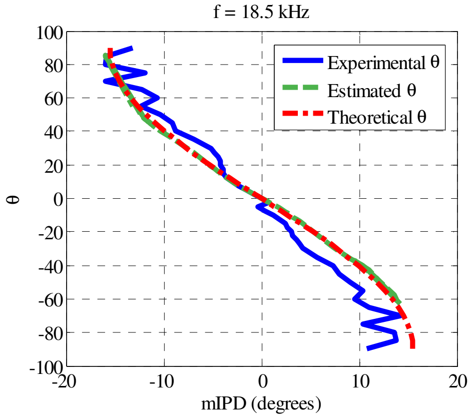


Figure F.58: Estimated vs. Experimental and Theoretical incident angle θ at f = 18.5 kHz

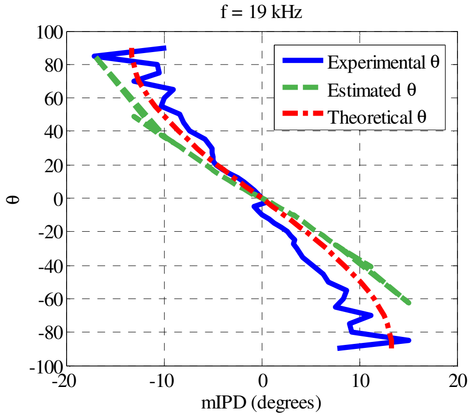


Figure F.59: Estimated vs. Experimental and Theoretical incident angle θ at f = 19 kHz

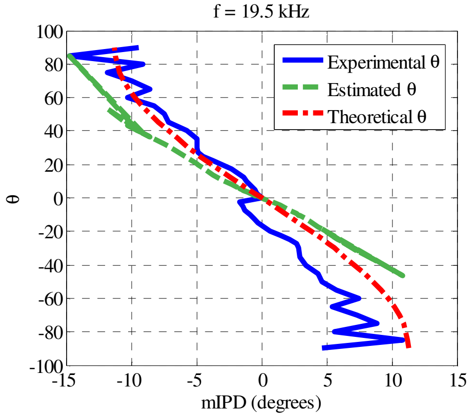
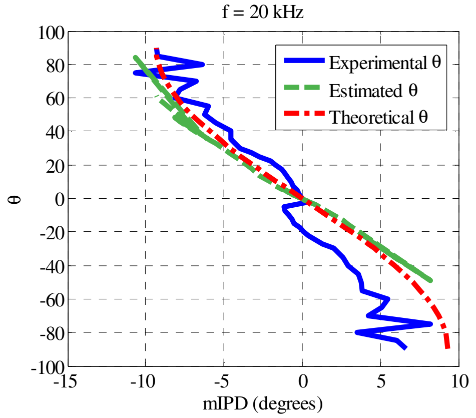


Figure F.60: Estimated vs. Experimental and Theoretical incident angle θ at f = 19.5 kHz



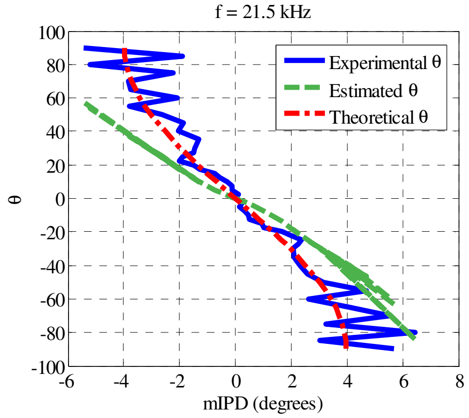
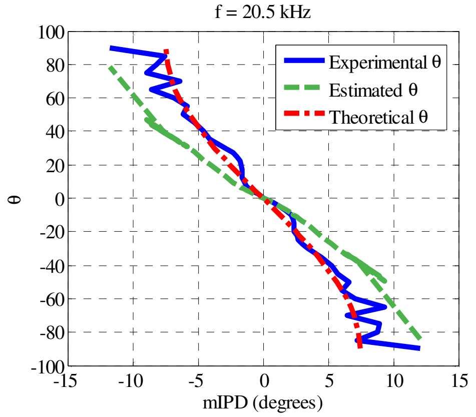


Figure F.61: Estimated vs. Experimental and Theoretical incident angle θ at f = 20 kHz

Figure F.64: Estimated vs. Experimental and Theoretical incident angle θ at f = 21.5 kHz



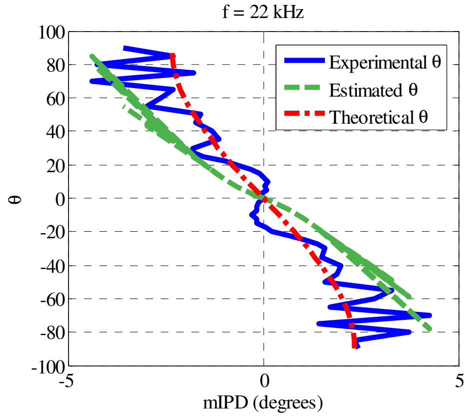
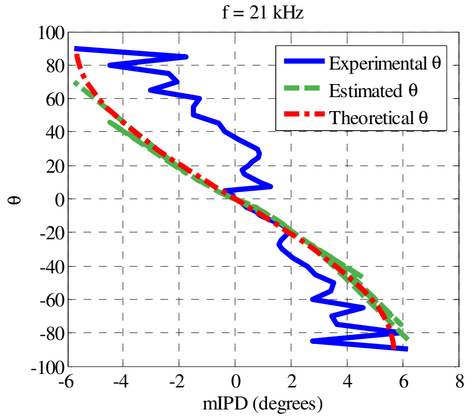


Figure F.62: Estimated vs. Experimental and Theoretical incident angle θ at f = 20.5 kHz

Figure F.65: Estimated vs. Experimental and Theoretical incident angle θ at f = 22 kHz



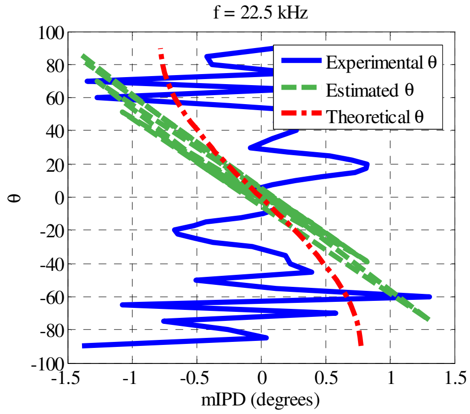


Figure F.63: Estimated vs. Experimental and Theoretical incident angle θ at f = 21 kHz

Figure F.66: Estimated vs. Experimental and Theoretical incident angle θ at f = 22.5 kHz

F.2 Experimental Model

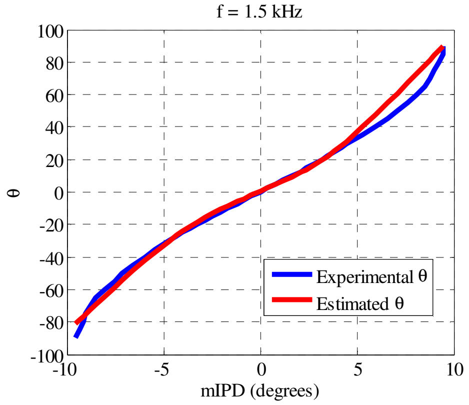


Figure F.67: Estimated vs. Experimental incident angle θ at f = 1.5 kHz

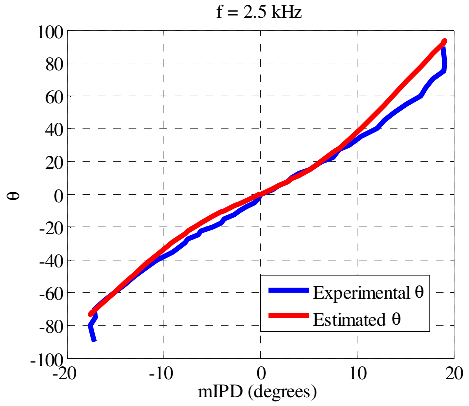


Figure F.68: Estimated vs. Experimental incident angle θ at f = 2.5 kHz

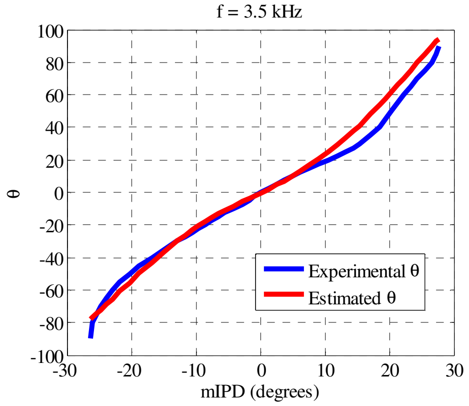


Figure F.69: Estimated vs. Experimental incident angle θ at f = 3.5 kHz

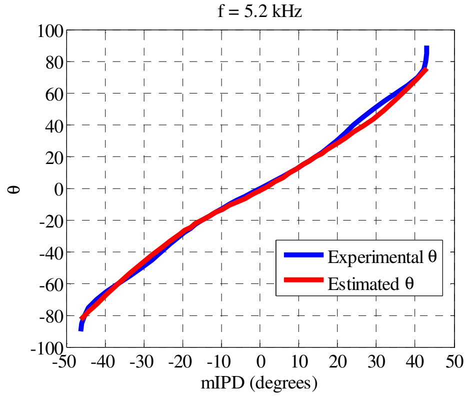


Figure F.70: Estimated vs. Experimental incident angle θ at f = 5.2 kHz

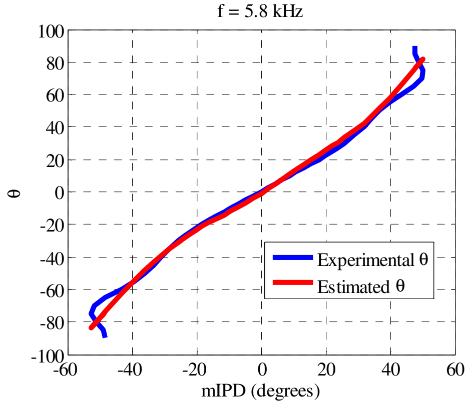


Figure F.71: Estimated vs. Experimental incident angle θ at f = 5.8 kHz

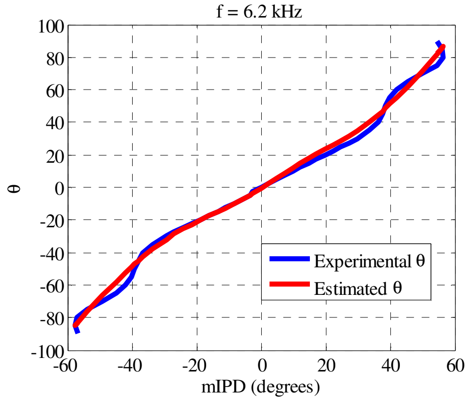


Figure F.72: Estimated vs. Experimental incident angle θ at f = 6.2 kHz

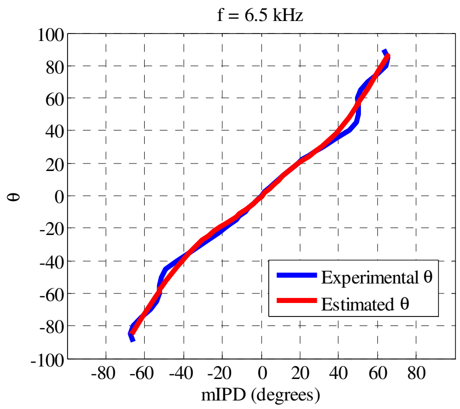


Figure F.73: Estimated vs. Experimental incident angle θ at f = 6.5 kHz

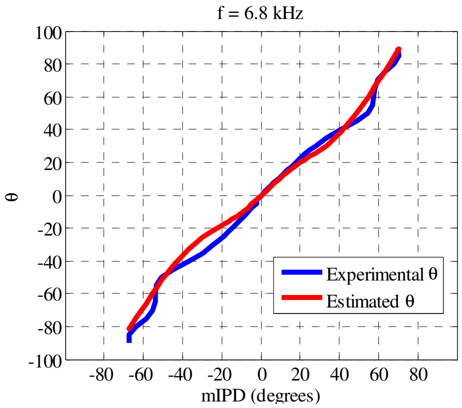


Figure F.74: Estimated vs. Experimental incident angle θ at f = 6.8 kHz

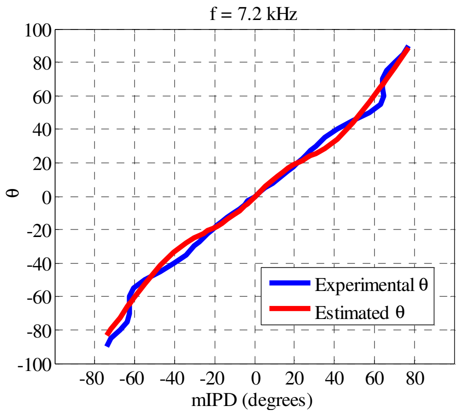


Figure F.75: Estimated vs. Experimental incident angle θ at f = 7.2 kHz

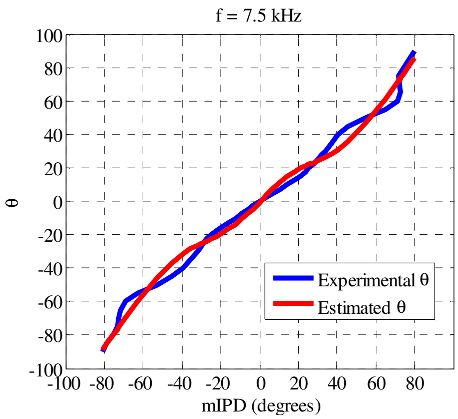


Figure F.76: Estimated vs. Experimental incident angle θ at f = 7.5 kHz

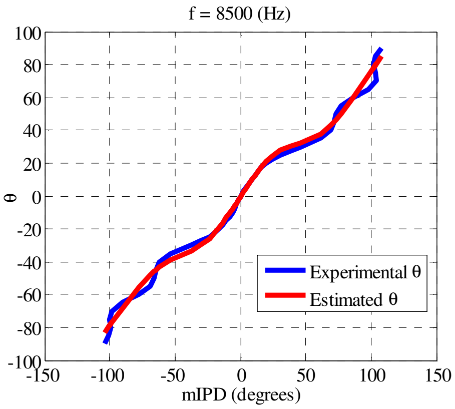


Figure F.77: Estimated vs. Experimental incident angle θ at f = 8.5 kHz

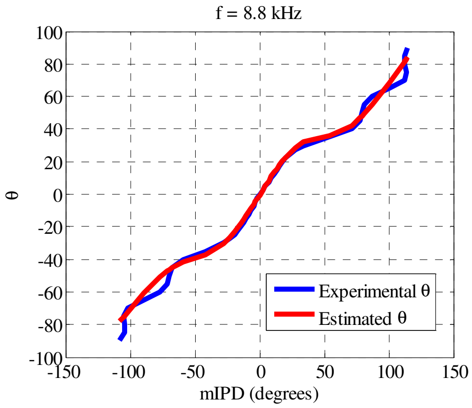


Figure F.78: Estimated vs. Experimental incident angle θ at f = 8.8 kHz

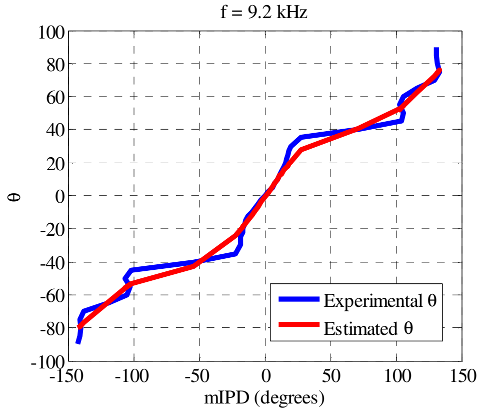


Figure F.79: Estimated vs. Experimental incident angle θ at f = 9.2 kHz

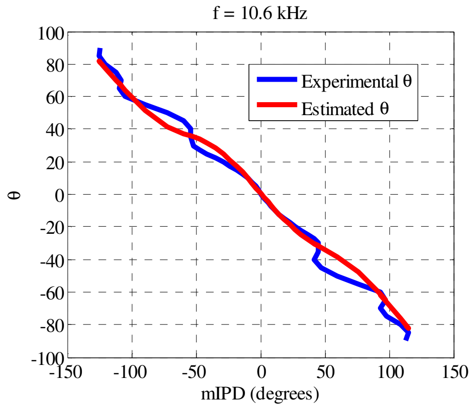


Figure F.80: Estimated vs. Experimental incident angle θ at f = 10.6 kHz

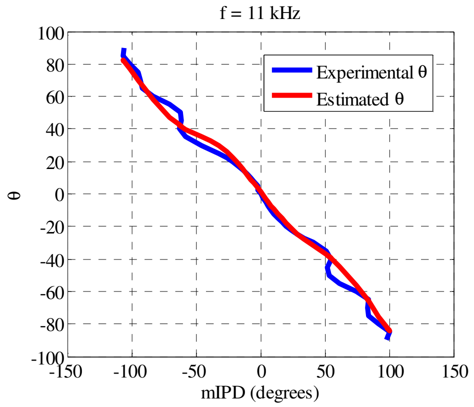


Figure F.81: Estimated vs. Experimental incident angle θ at f = 11 kHz

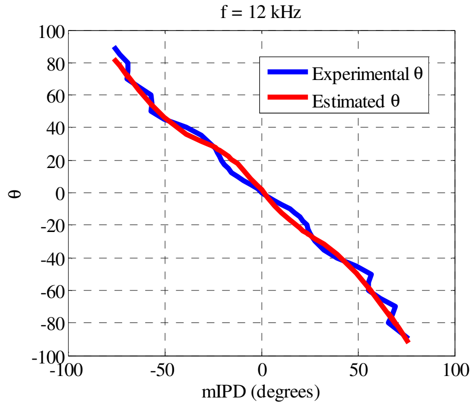


Figure F.82: Estimated vs. Experimental incident angle θ at f = 12 kHz

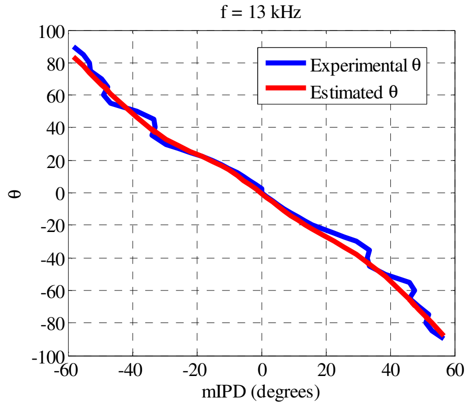


Figure F.83: Estimated vs. Experimental incident angle θ at f = 13 KHz

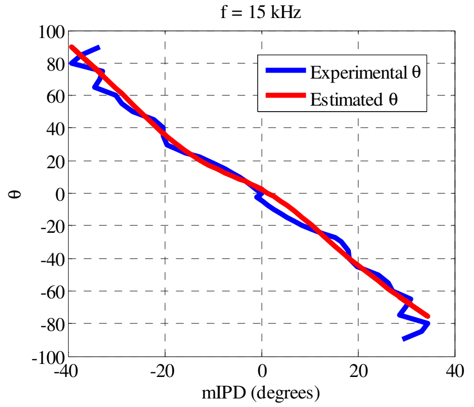


Figure F.84: Estimated vs. Experimental incident angle θ at f = 15 kHz

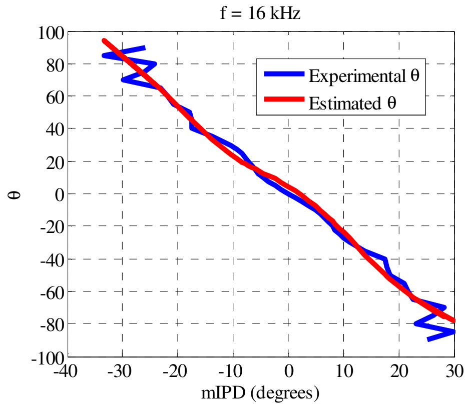


Figure F.85: Estimated vs. Experimental incident angle θ at f = 16 kHz

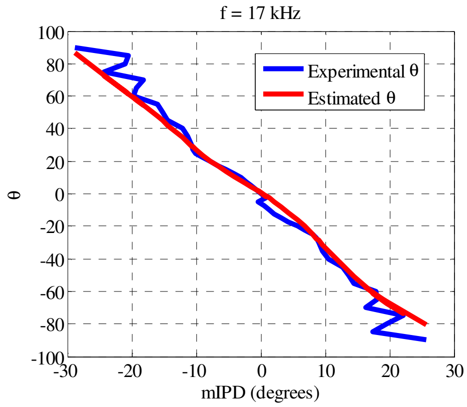


Figure F.86: Estimated vs. Experimental incident angle θ at f = 17 kHz

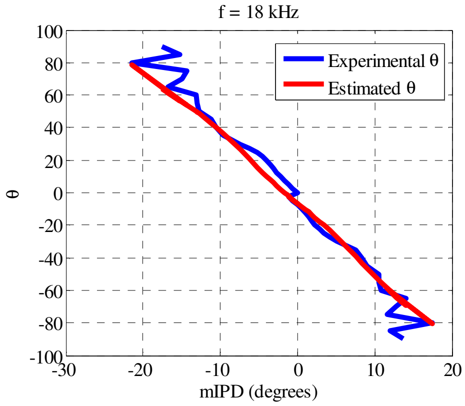


Figure F.87: Estimated vs. Experimental incident angle θ at f = 18 kHz

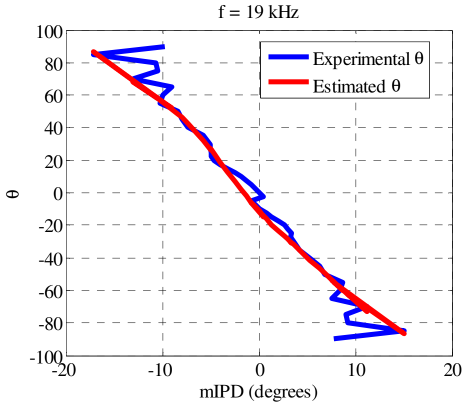


Figure F.88: Estimated vs. Experimental incident angle θ at f = 19 kHz

Appendix G SIMULINK Codes Simulating the Moving Sound Localization and Tracking Problem

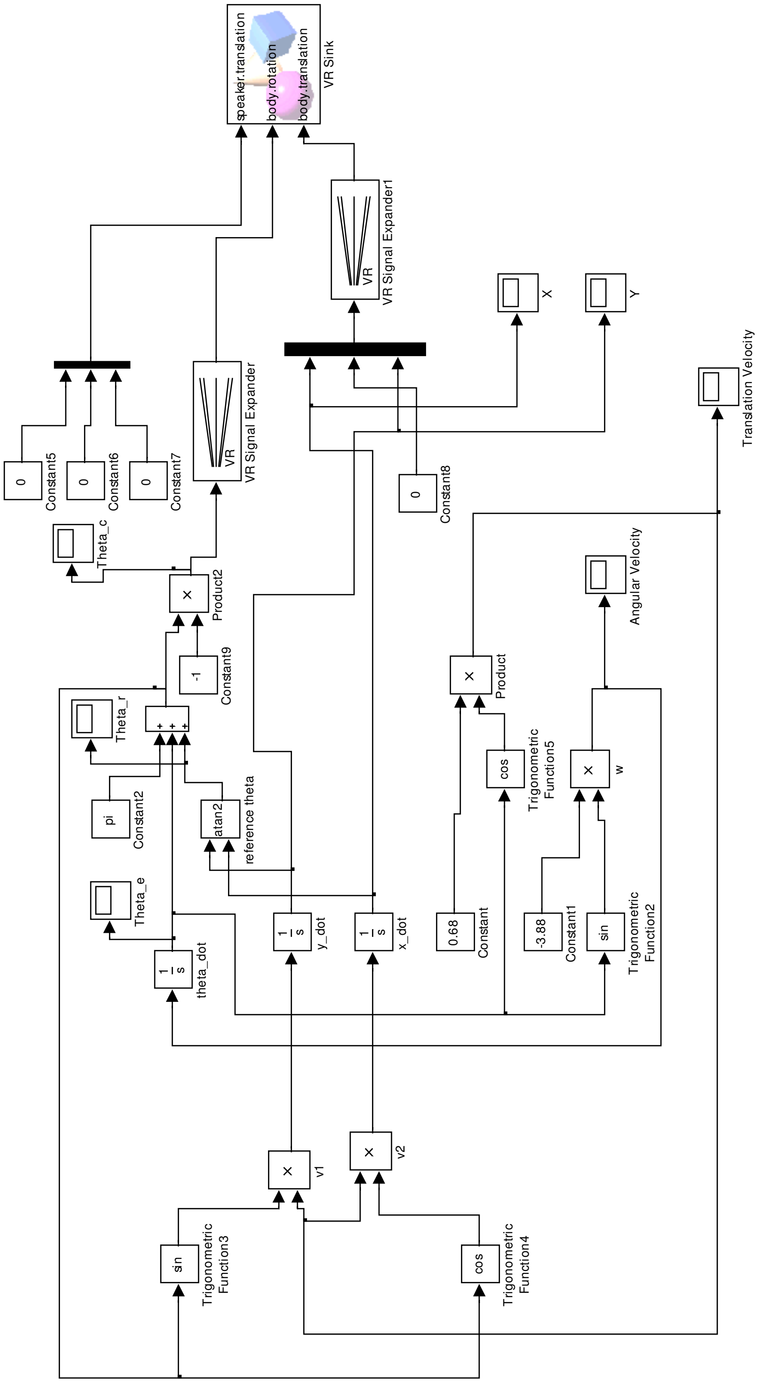


Figure G.1: Stationary sound localization using the Nonlinear Controller.

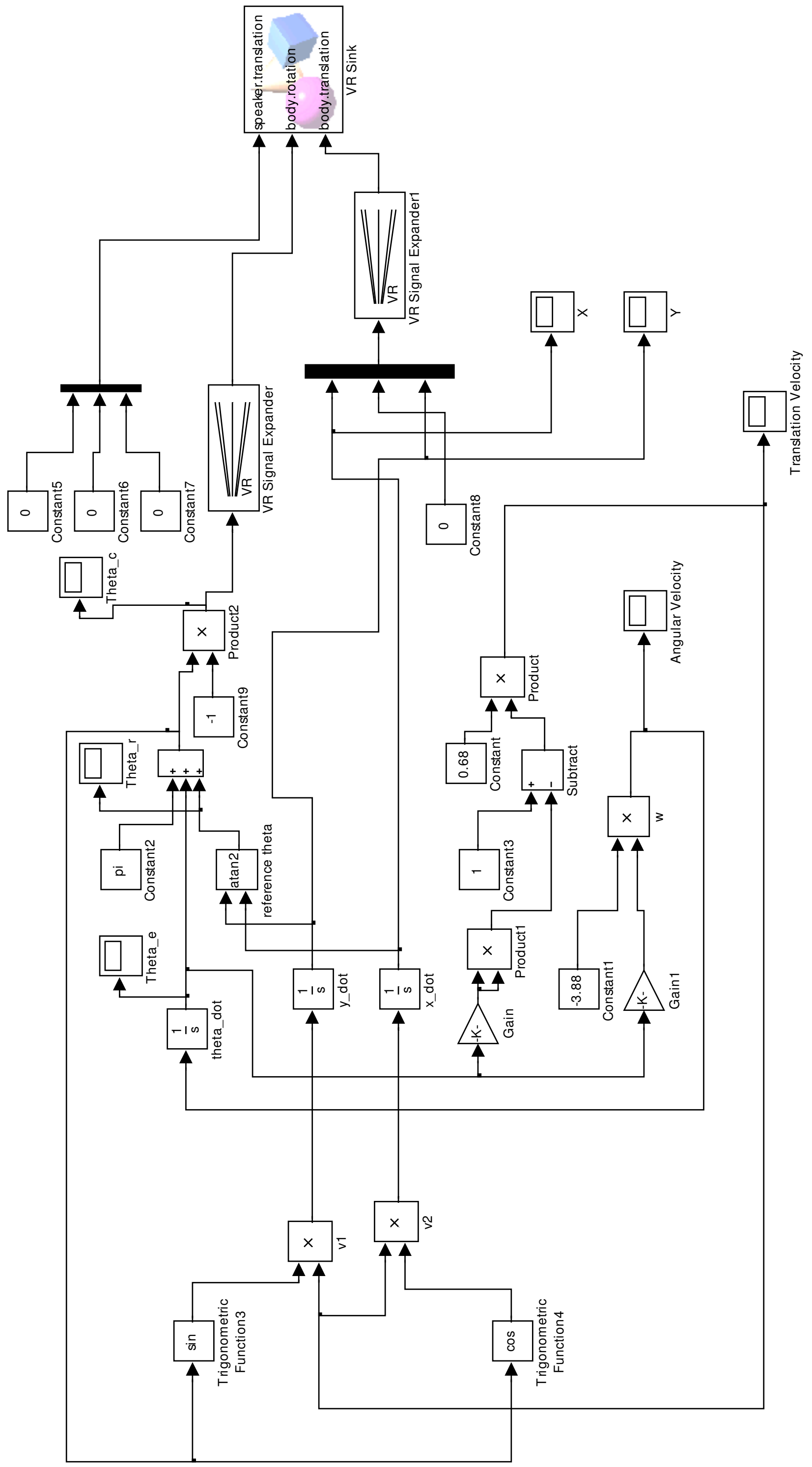


Figure G.2: Stationary sound localization using the Quadratic-linear Controller.

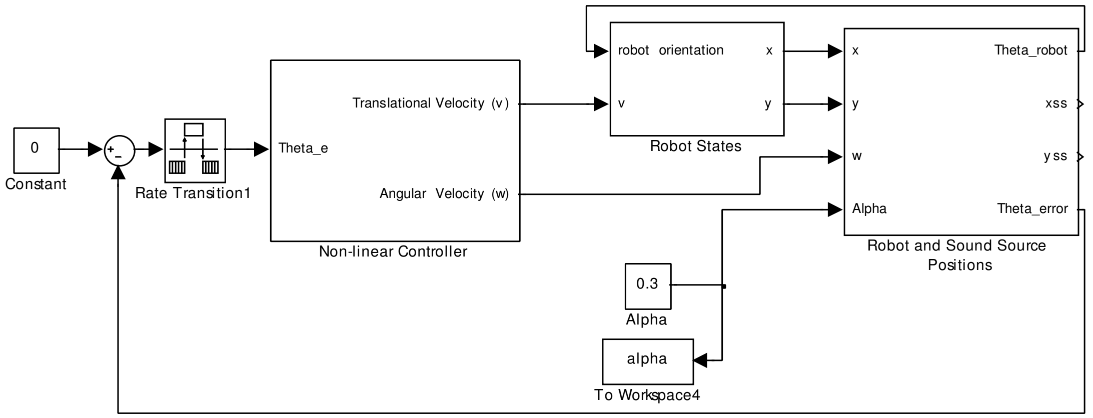


Figure G.3: Moving sound tracking using the Nonlinear Controller.

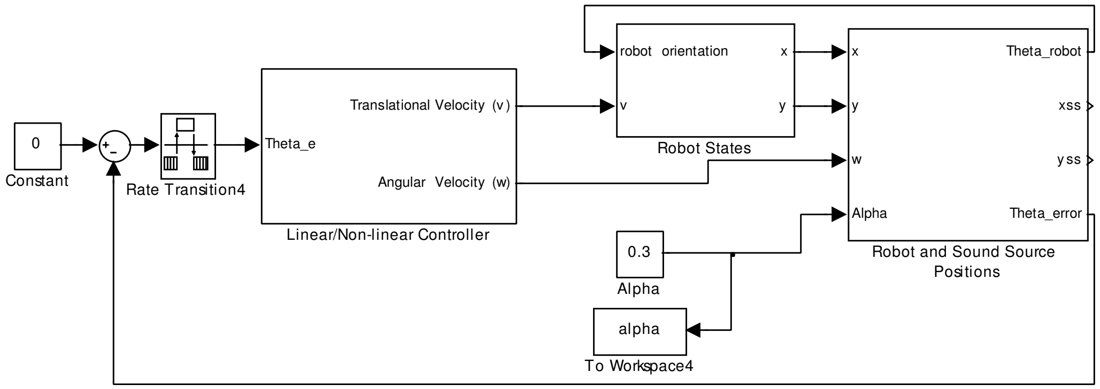


Figure G.4: Moving sound tracking using the Quadratic-linear Controller.

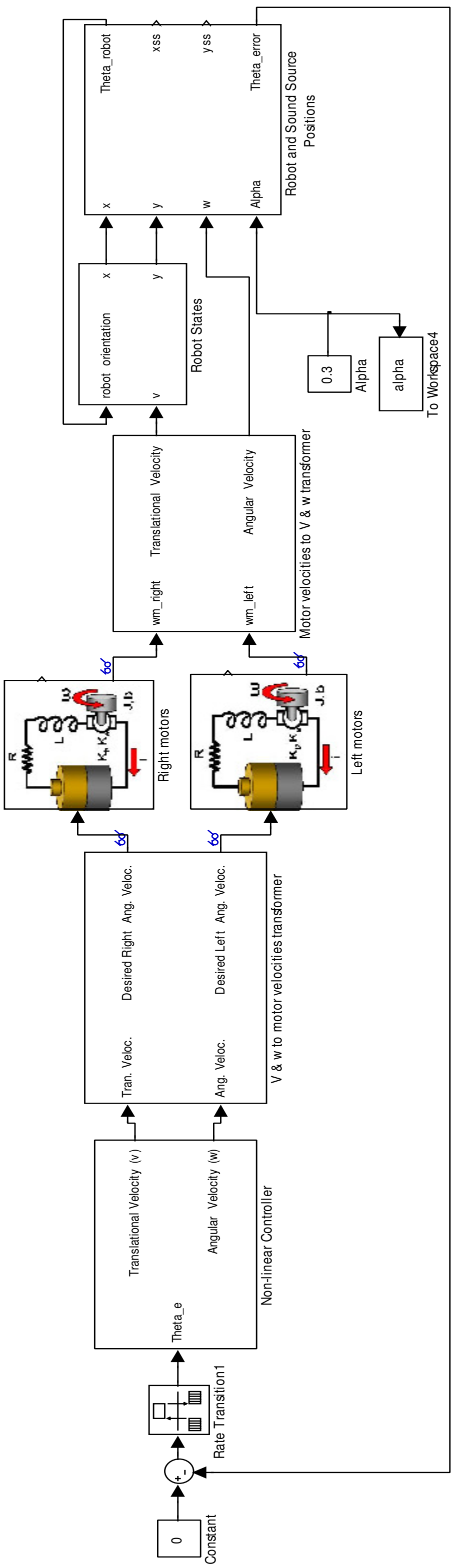


Figure G.5: Moving sound tracking using the Nonlinear Controller and wheels PID Controller.

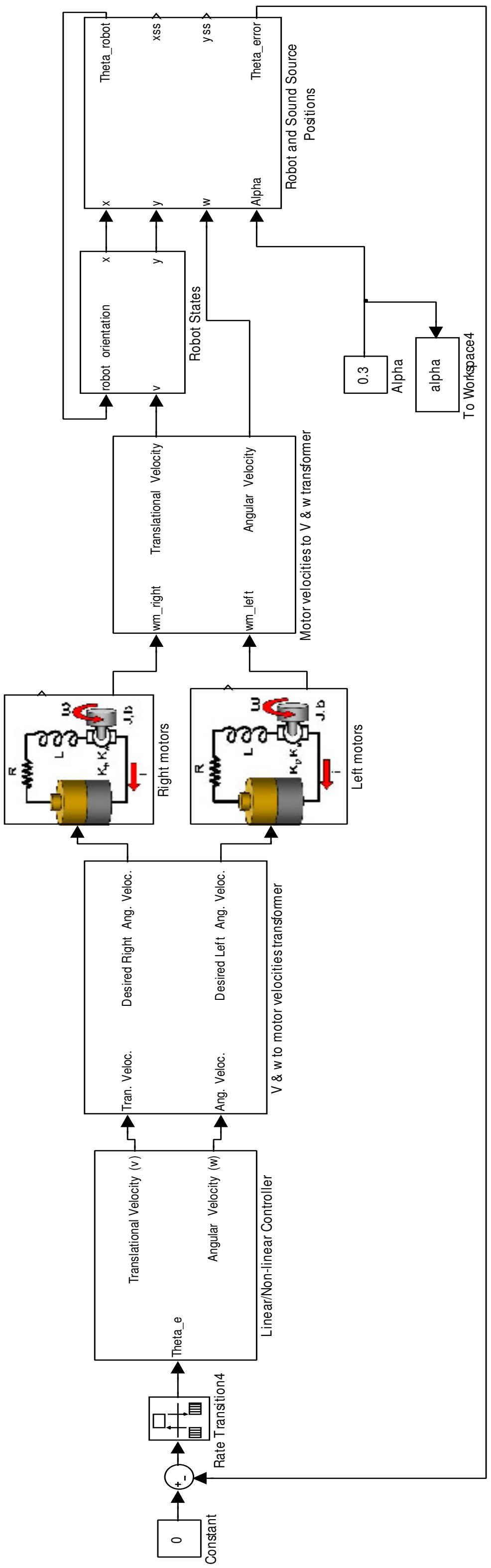


Figure G.6: Moving sound tracking using the Quadratic-linear Controller and wheels PID Controller.

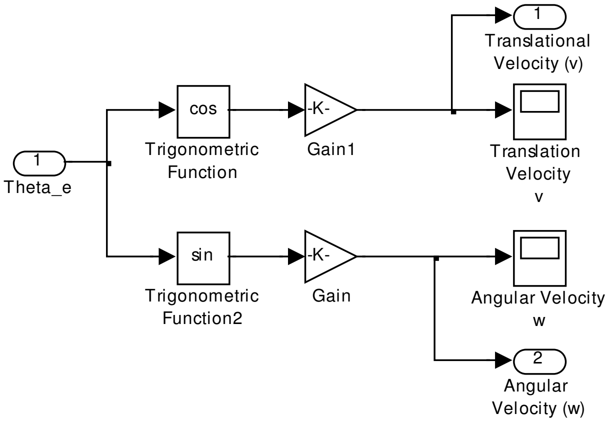


Figure G.7: Nonlinear Controller model.

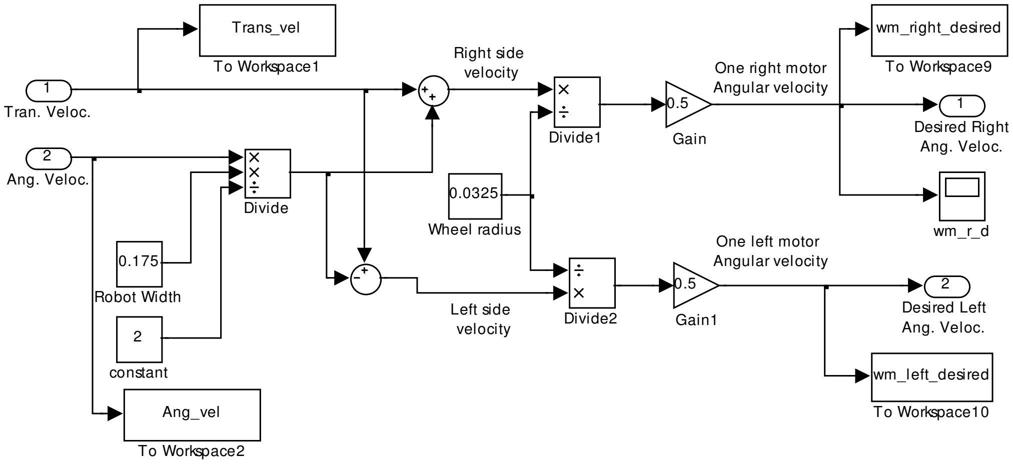


Figure G.8: Translational and Angular velocities to Left and Right motors Angular velocities map.

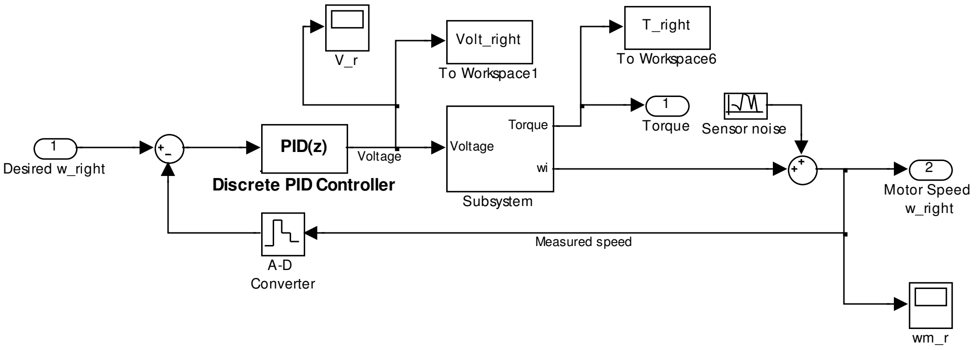


Figure G.9: Wheels PID Controller model.

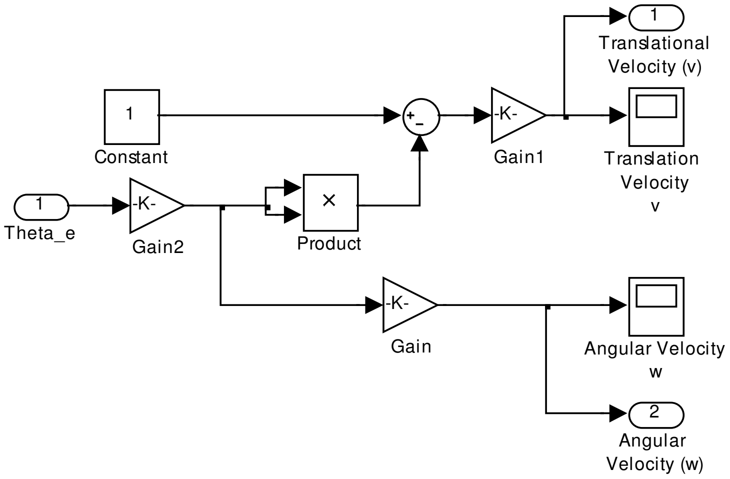


Figure G.10: Nonlinear Controller model.

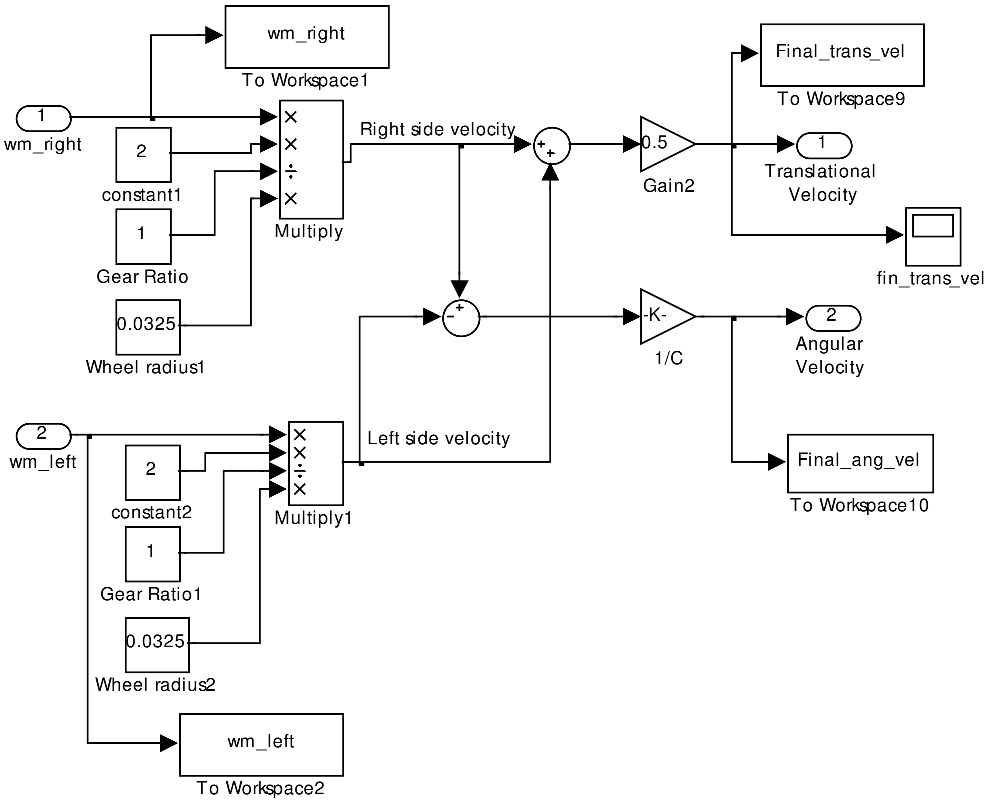


Figure G.11: Left Rights wheels speed to final translational and angular velocities map (after PID control).

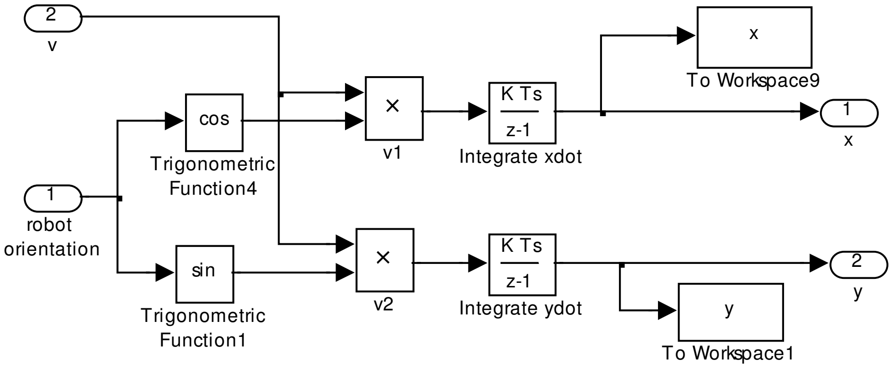


Figure G.12: Two-wheeled robot kinematics model.

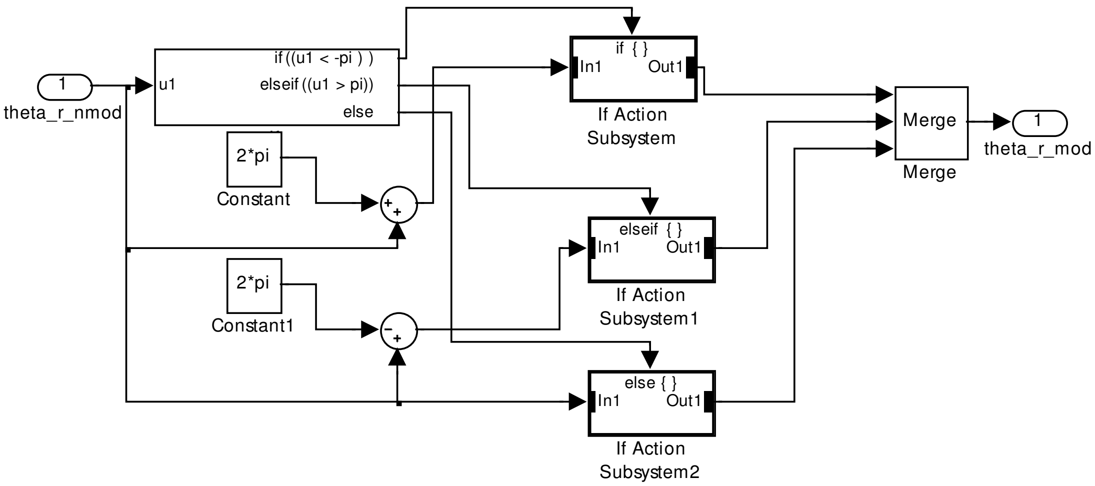


Figure G.13: Angle wrapping.

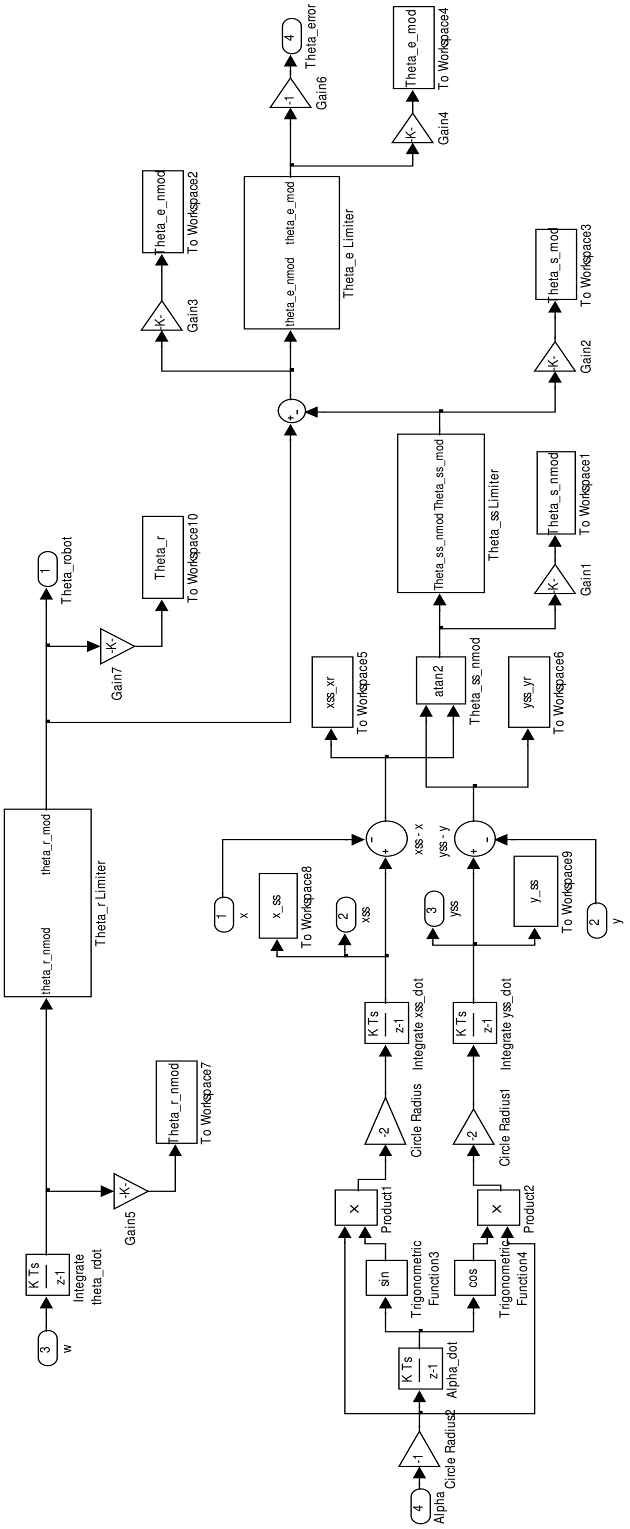
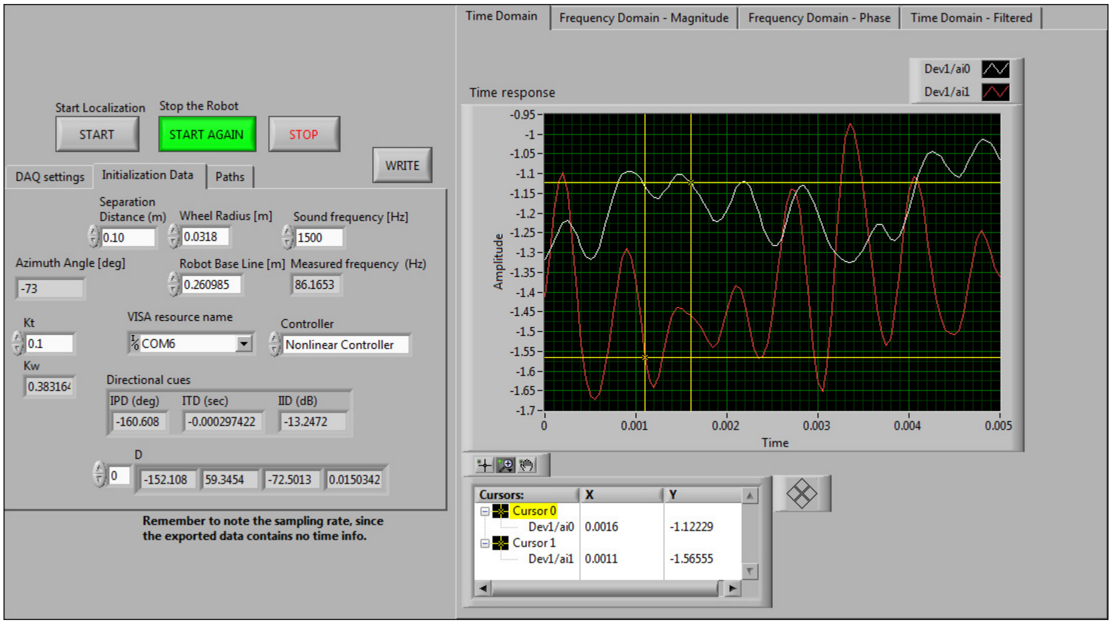


Figure G.14: Moving sound source model and heading error angle calculation.

Appendix H LabView Program for Signal Processing and Decision Making



Separation Distance (m): 0.10
Wheel Radius [m]: 0.0318
Sound frequency [Hz]: 1500
Azimuth Angle [deg]: -73
Robot Base Line [m]: 0.260985
Measured frequency (Hz): 86.1653
Kt: 0.1
Kw: 0.383164
VISA resource name: COM6
Controller: Nonlinear Controller

Directional cues

IPD (deg)	ITD (sec)	IID (dB)
-160.608	-0.000297422	-13.2472

D: 0 | -152.108 | 59.3454 | -72.5013 | 0.0150342

Remember to note the sampling rate, since the exported data contains no time info.

Cursors:	X	Y
Cursor 0		
Dev1/ai0	0.0016	-1.12229
Cursor 1		
Dev1/ai1	0.0011	-1.56555

Figure H.1: Front panel of data acquisition and decision making VI.

H.1 Description and Instructions

This VI program acquires the signals from the two microphones then process the signals (filter) to extract useful information about the phase difference between the two signals that can be used to estimate the AOA of the sound source. Original signals can be saved as required but the history of the decision making is saved and can be plotted later to show the behavior of the robot while localizing and tracking the sound source.

Instructions:

1) Enter the DAQ setting to acquire the signals from the channels the microphones are connected to. The values showed are compatible with the DAQ parameters values, i.e. sampling rate, maximum and minimum measured voltages…etc.

2) Initialize the data needed to estimate the sound AOA, the wireless communication port, the controllers gains, robot wheel radius, robot base line length, sound frequency (can be estimated as well), and the controller type (nonlinear/quadratic-linear controller).

3) Choose the path to which the data need to be saved.

4) Tracking can be start after running the program by the start localization button, the tracking platform can be stopped by the stop the robot button, and the program can be completely stopped by the STOP button.

5) The data can be acquired without moving the tracking robot by pressing the start localization button (become green) and the stop the robot button (become green). To move the robot again, press the stop the robot button again (become red).

6) To save the original microphone signals press the write button. The data will be saved in the directory specified in (3).

7) Four representations of the signals are displayed simultaneously. The time domain signal before and after filtering, the frequency domain signal magnitude and phase.

H.2 iRobot Commands

Start	Starts the OI. Should be always sent before sending any other commands to the OI.
Baud	Sets the baud rate in bits per second (bps). The default baud rate at power up is 57600 bps.
Safe	Puts the OI into Safe mode, enabling user control of Create.
Full	Gives the user complete control over Create by putting the OI into Full mode, and turning off the cliff, wheel-drop and internal charger safety features.
Demo	Starts the requested built-in demo.
Demo - Cover	Starts the Cover demo.
Demo - Cover and Dock	Starts the Cover and Dock demo.
Demo - Spot	Starts the Spot Cover demo.
Drive	Controls Create's drive wheels. It takes four data bytes, interpreted as two 16-bit signed values using two's complement. The first two bytes specify the average velocity of the drive wheels (mm/s), with the high byte being sent first. The next two bytes specify the radius (mm) at which Create will turn.
Drive Direct	Lets the user control the forward and backward motion of Create's drive wheels independently. It takes four data bytes, which are interpreted as two 16-bit signed values using two's complement. The first two bytes specify the velocity of the right wheel (mm/s), with the high byte sent first. The next two bytes specify the velocity of the left wheel, in the same format.
LEDs	Controls the LEDs on Create.
Digital Outputs	Controls the state of the 3 digital output pins on the 25 pin Cargo Bay Connector.
PWM Low Side Drivers	Lets the user control the three low side drivers with variable power.
Low Side Drivers	Lets the user control the three low side drivers.
Send IR	Sends the requested byte out of low side driver 1 (pin 23 on the Cargo Bay Connector), using the format expected by iRobot Create's IR receiver.
Song	Lets the user specify up to sixteen songs to the OI that can be played at a later time.
Play Song	Lets the user select a song to play from the songs added to iRobot Create using the Song command.
Sensors	Requests the OI to send a packet of sensor data bytes.
Quesy List	Lets the user ask for a list of sensor packets.
Stream	Starts a continuous stream of data packets.

Pause/Resume Stream

Lets the user stop and restart the steam without clearing the list of requested packets.

Script

Specifies a script to be played later.

Play Script

Loads a previously defined OI script into the serial input queue for playback.

Show Script

Returns the values of a previously stored script, starting with the number of bytes in the script and followed by the script's commands and data bytes.

Wait Time

Causes Create to wait for the specified time.

Wait Distance

Causes iRobot Create to wait until it has traveled the specified distance in mm.

Wait Angle

Causes Create to wait until it has rotated through specified angle in degrees.

Wait Event

Causes Create to wait until it detects the specified event.

Table H-1: iRobot commands opcodes

Item #	Command	Opcode	Item #	Command	Opcode
1	Start	128	16	Send IR	151
2	Baud	129	17	Song	140
3	Control	130	18	Play Song	141
4	Safe	131	19	Sensors	142
5	Full	132	20	Quesy List	149
6	Demo	136	21	Stream	148
7	Demo - Cover	135	22	Pause/Resume Stream	150
8	Demo - Cover and Dock	143	23	Script	152
9	Demo - Spot	134	24	Play Script	153
10	Drive	137	25	Show Script	154
11	Drive Direct	145	26	Wait Time	155
12	LEDs	139	27	Wait Distance	156
13	Digital Outputs	147	28	Wait Angle	157
14	PWM Low Side Drivers	144	29	Wait Event	158
15	Low Side Drivers	138	30	---	---

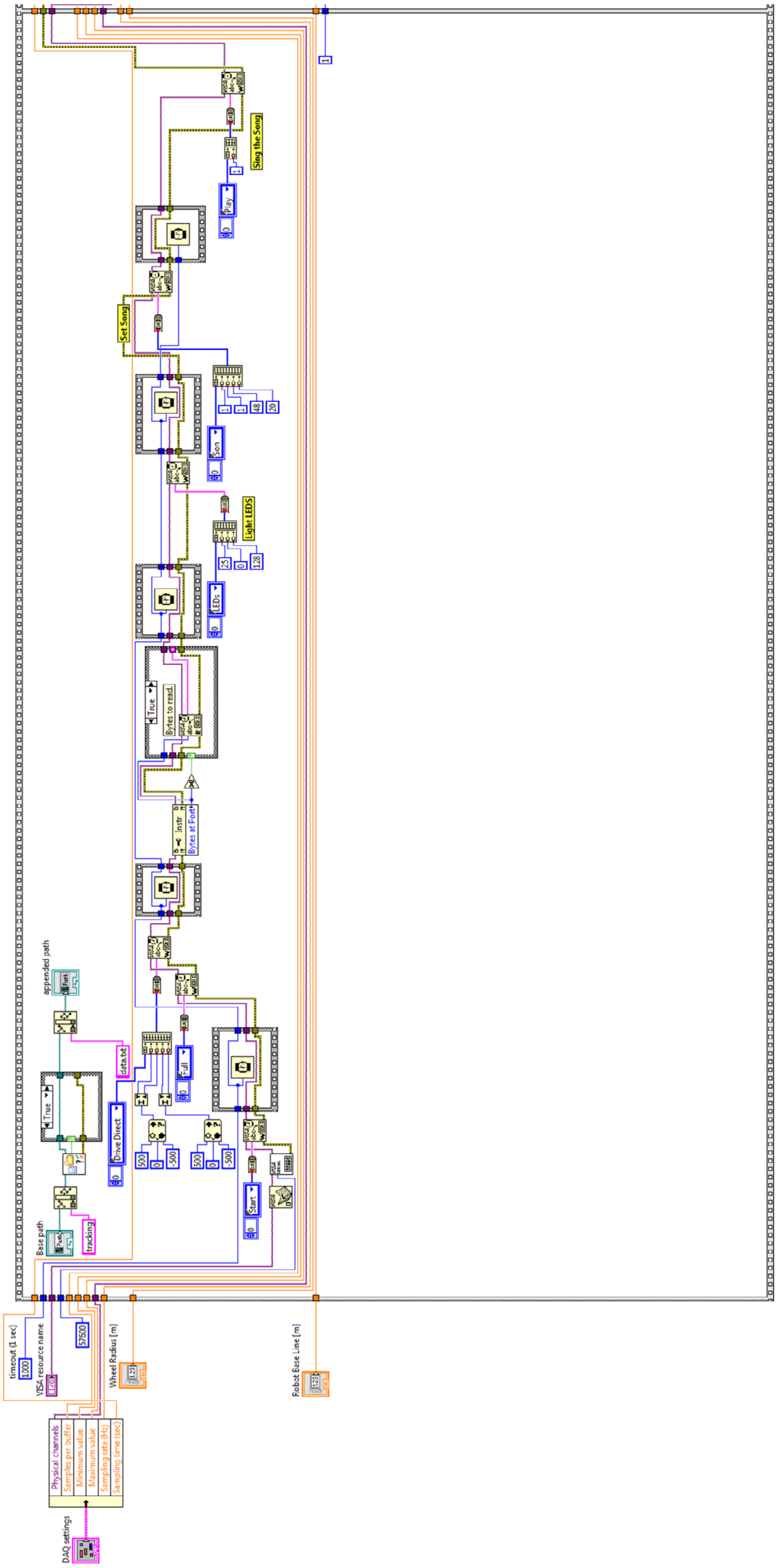
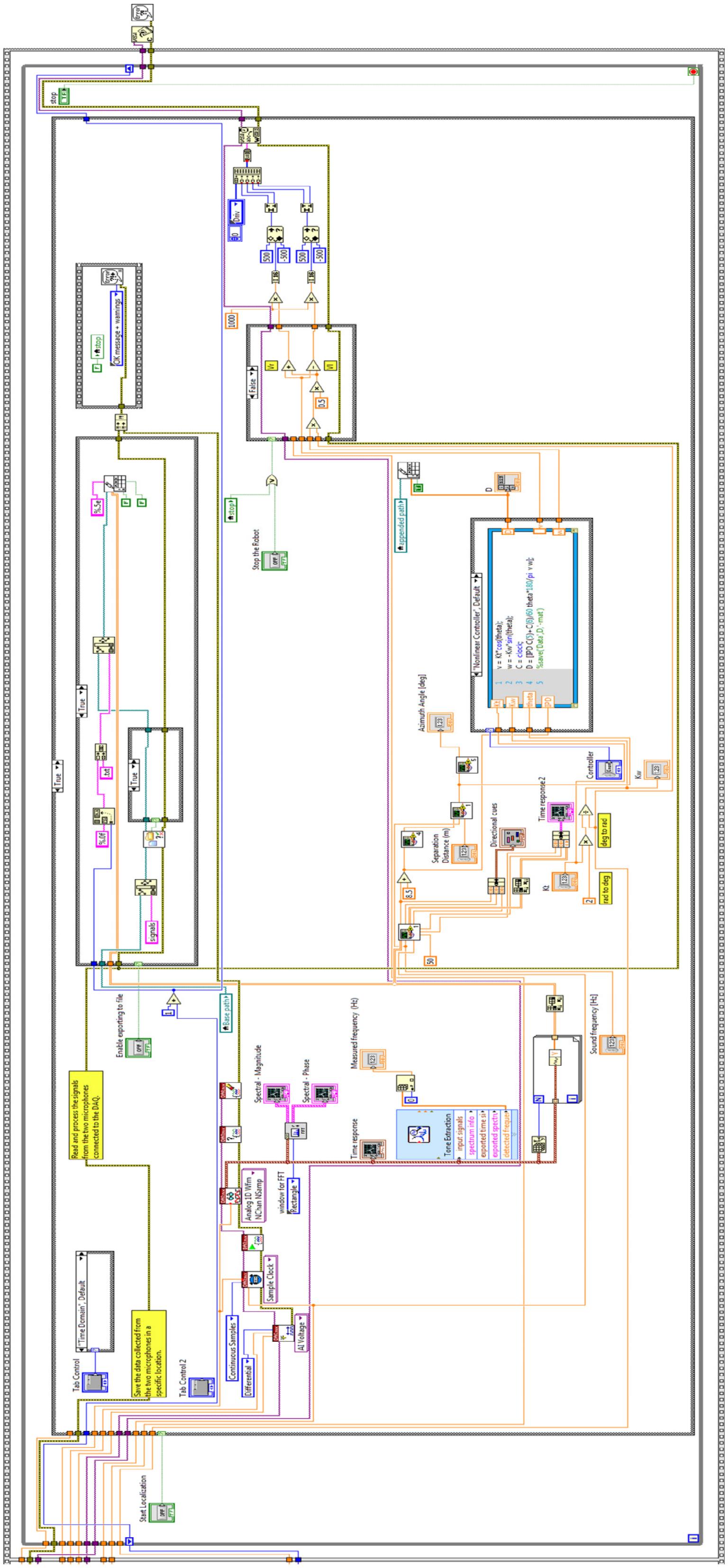


Figure H.2: Block diagram of the VI (left hand side).

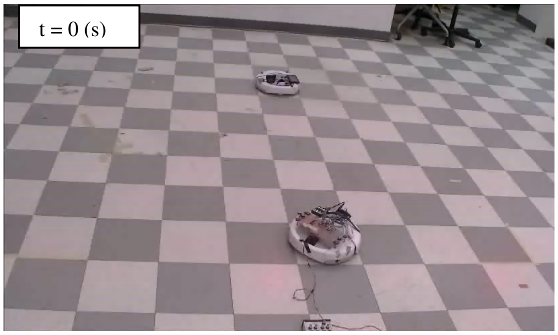


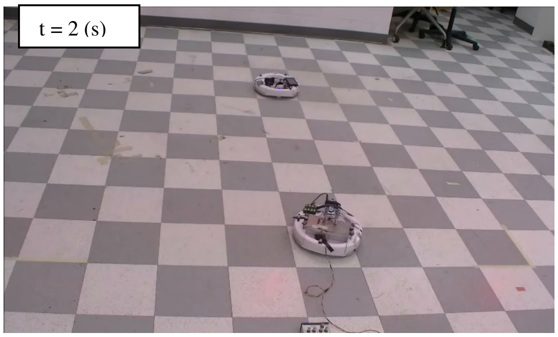
```
v = Kt*cos(theta);
w = -Kw*sin(theta);
C = clock;
D = [IPD C(5)+C(6)/60 theta*180/pi v w];
%save('Data',D,'-mat')
```

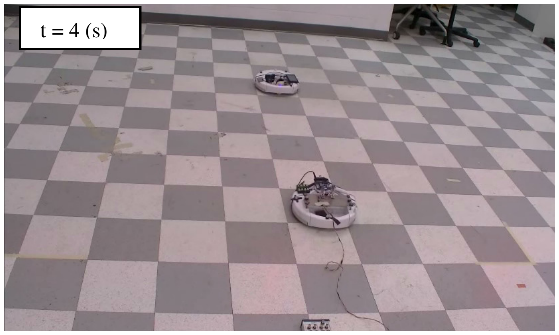
Figure H.3: Block diagram of the VI (right hand side).

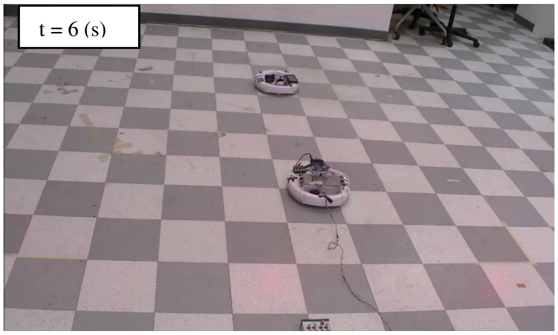
Appendix I Stationary Sound Source

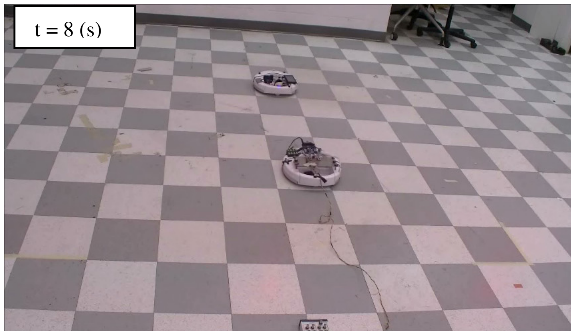
I.1 Nonlinear Controller (initial heading 45^{o})

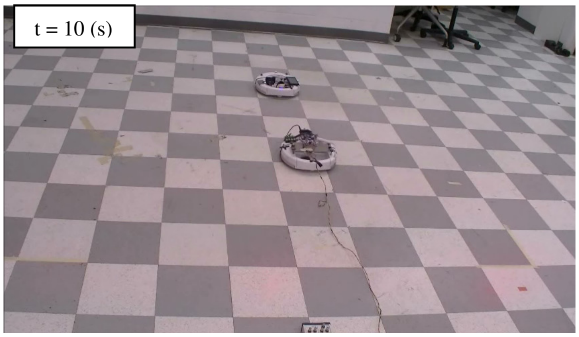


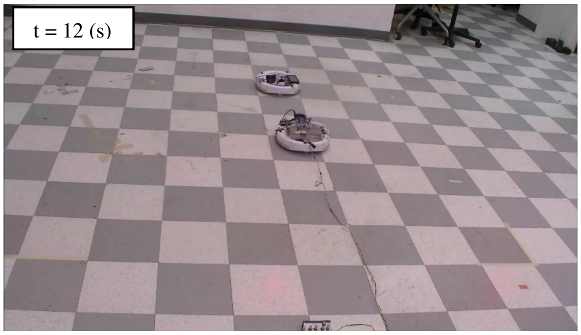


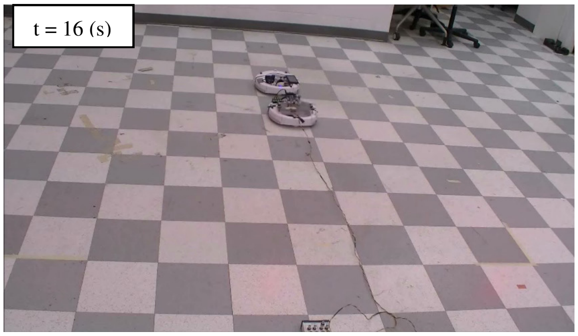




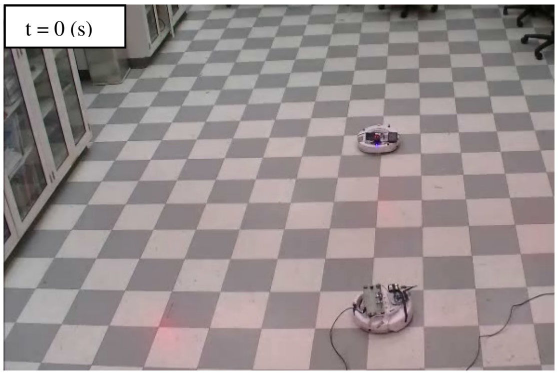


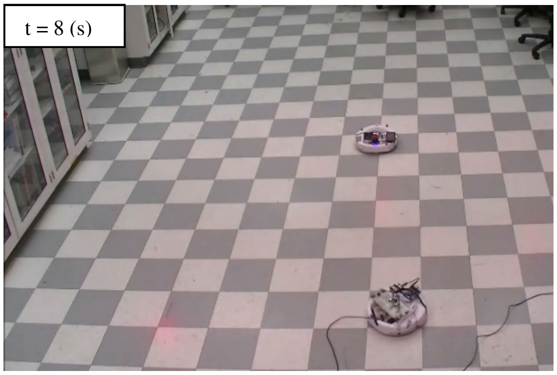


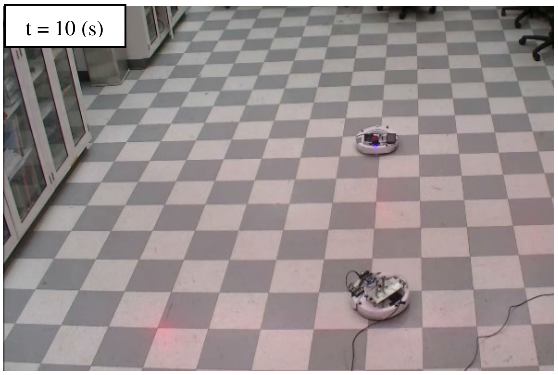


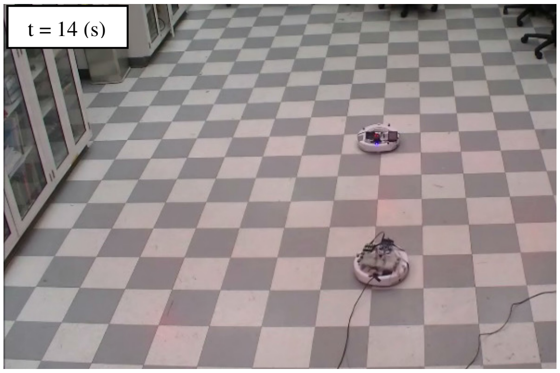


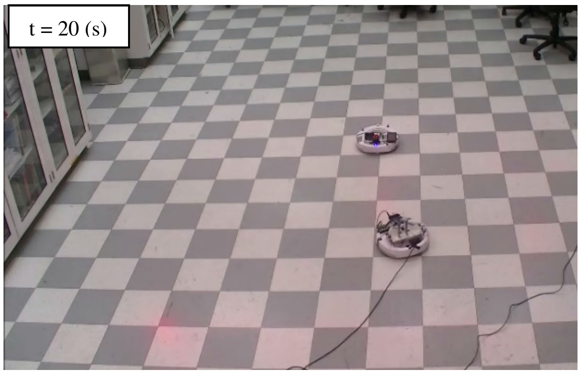
I.2 Nonlinear Controller (initial heading 90^o)

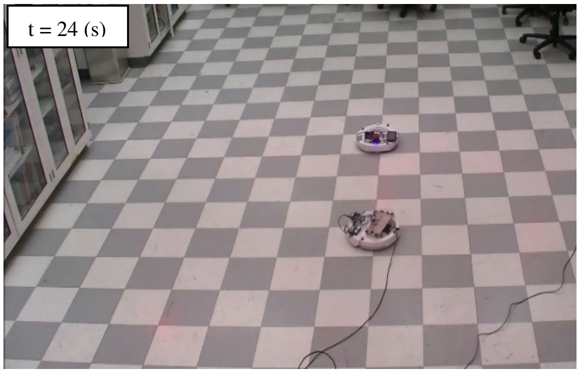

t = 0 (s)

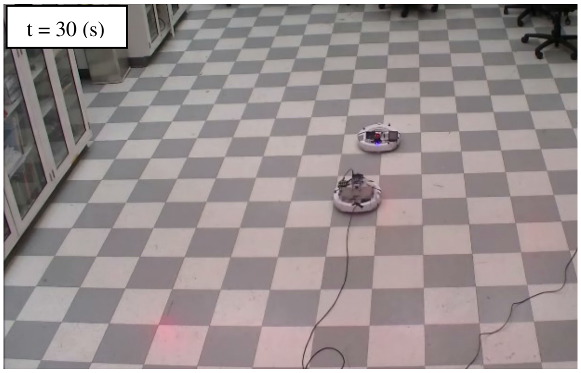

t = 8 (s)

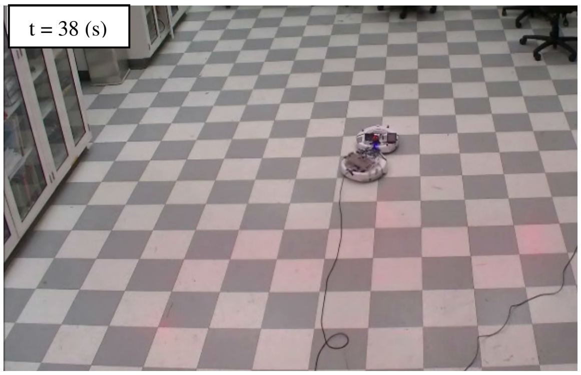

t = 10 (s)


t = 14 (s)

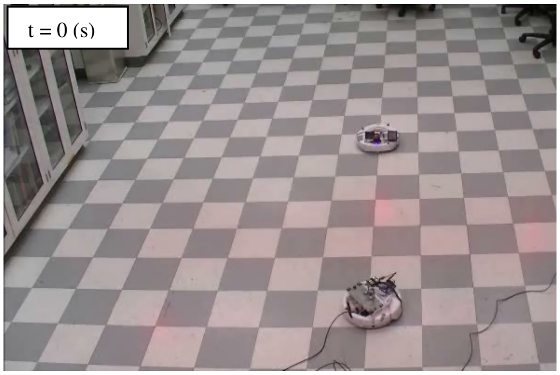

t = 20 (s)

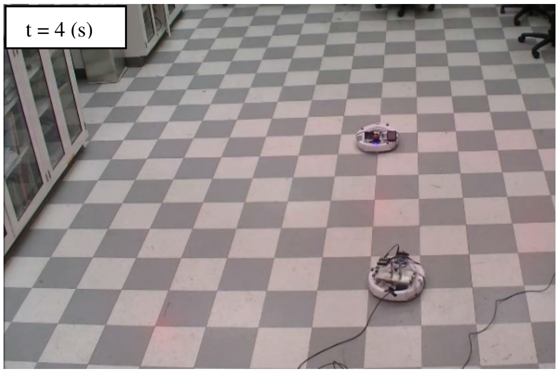

t = 24 (s)

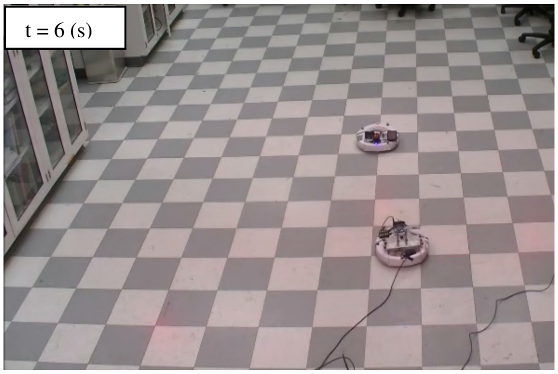

t = 30 (s)

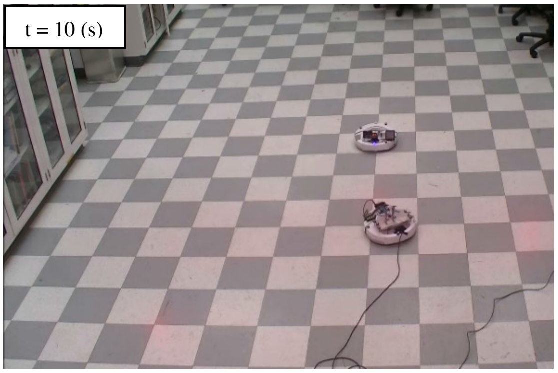

t = 38 (s)

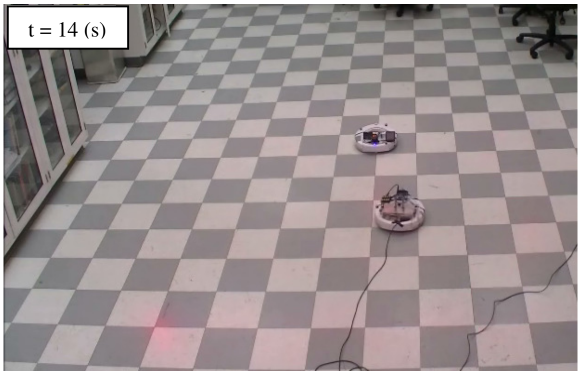
I.3 Quadratic-linear Controller (initial heading 45^o)

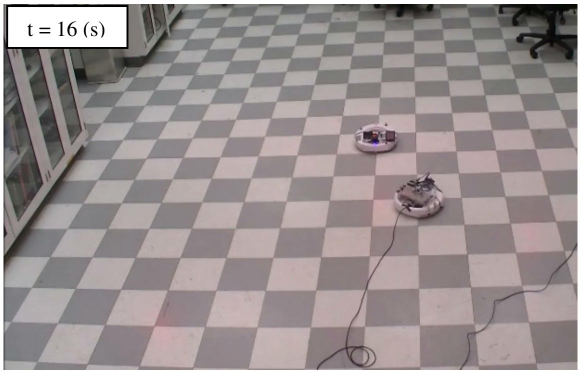


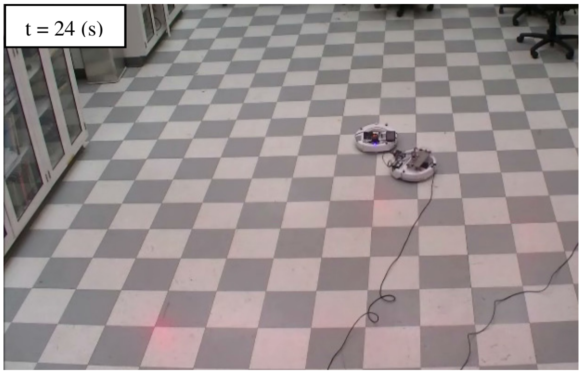


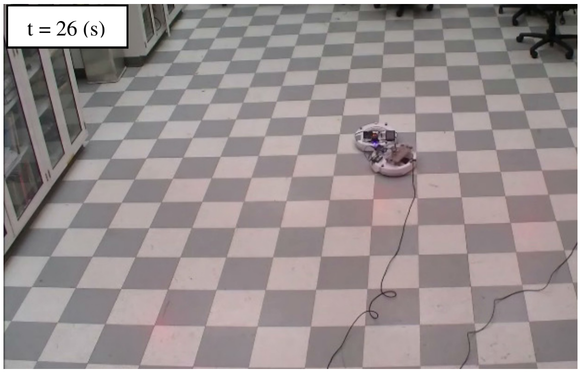




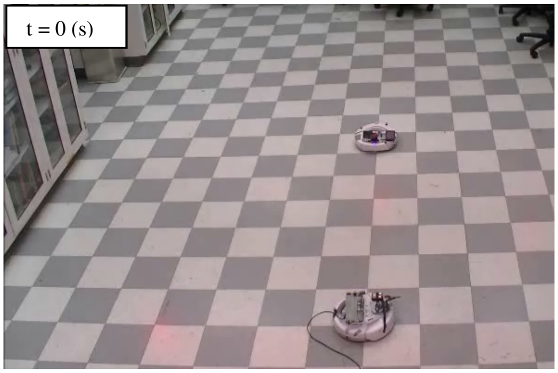


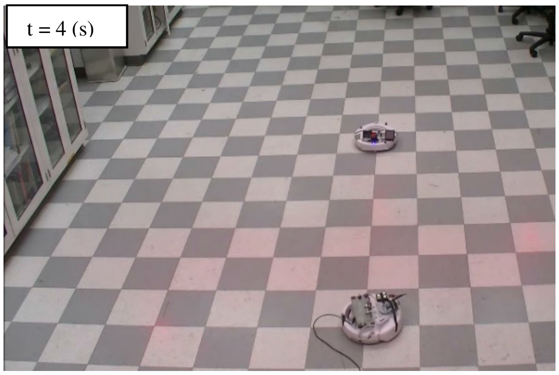


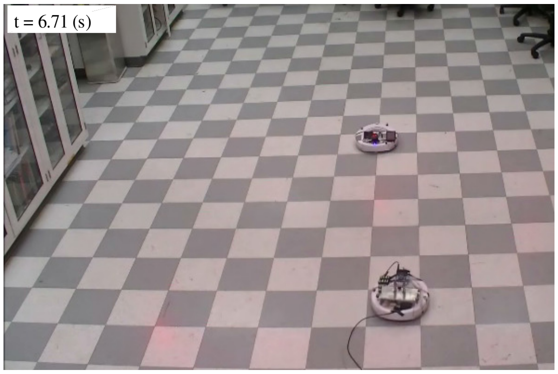


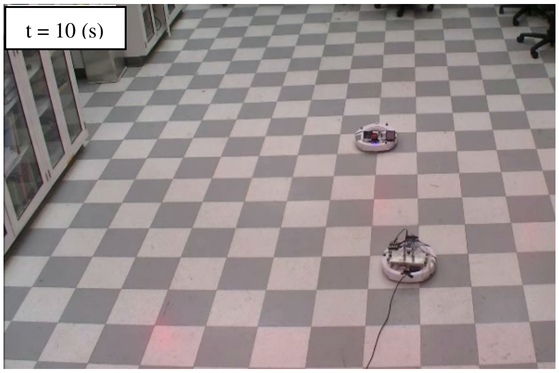


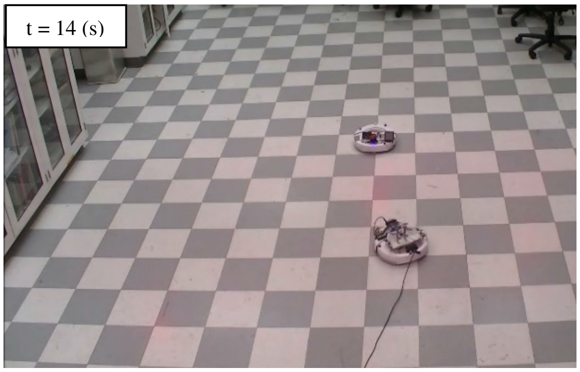
I.4 Quadratic-linear Controller (initial heading 90^o)

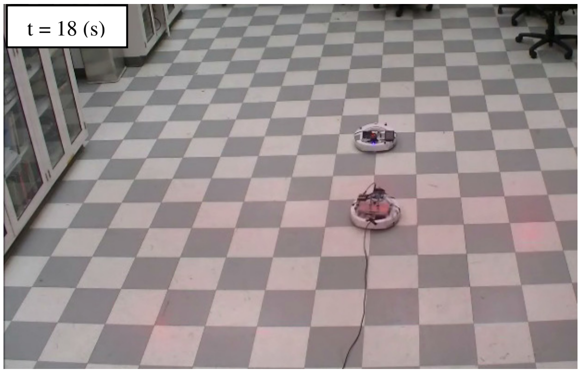


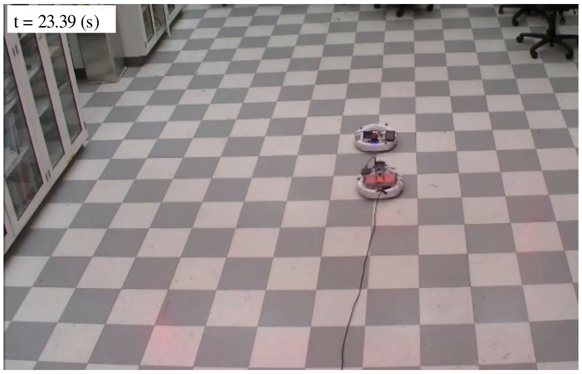


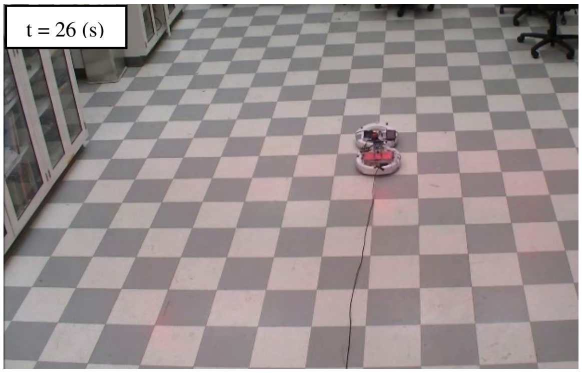






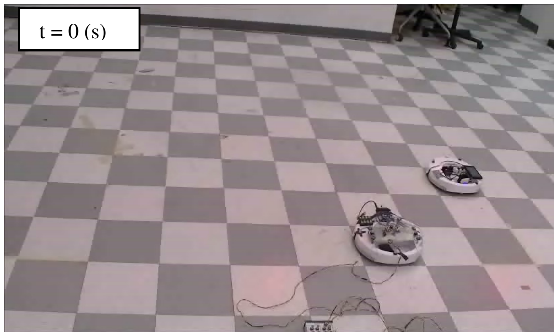


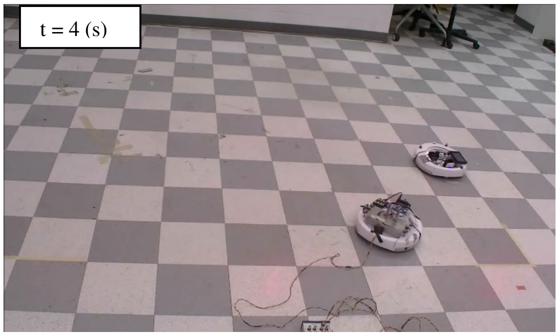


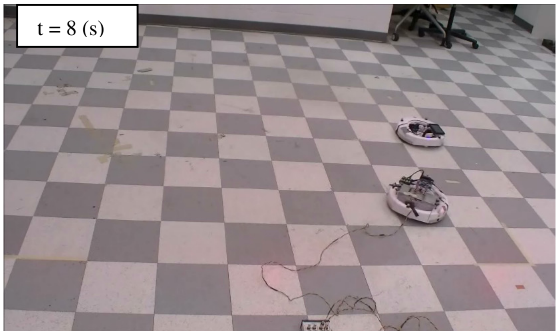


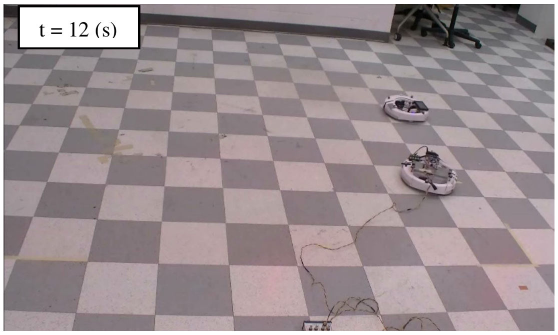
Appendix J   Moving Sound Source

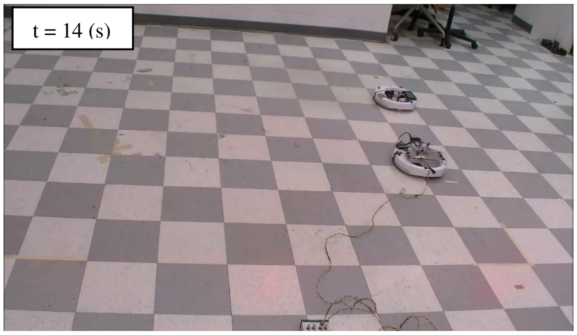
J.1 Nonlinear Controller (straight path)

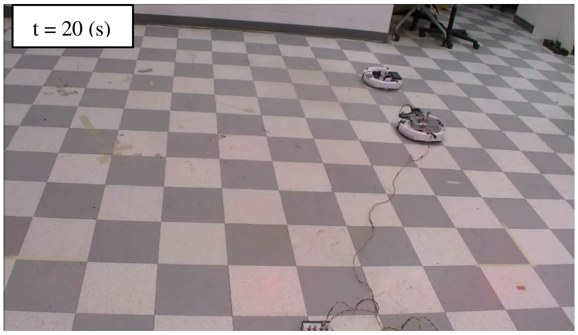


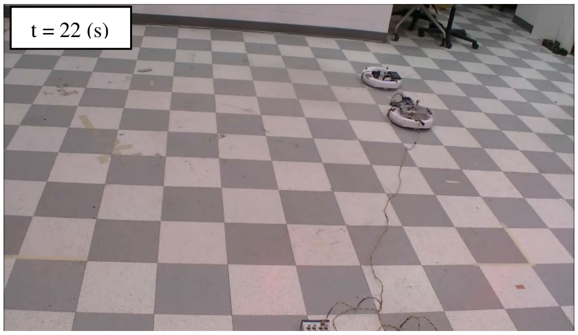


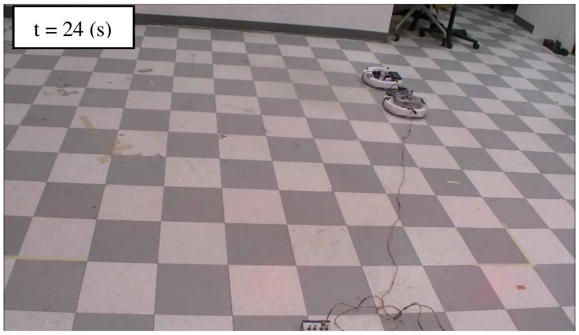




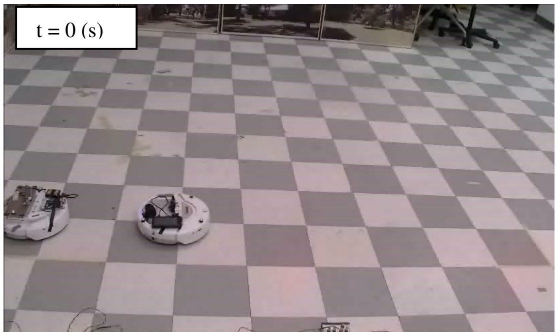


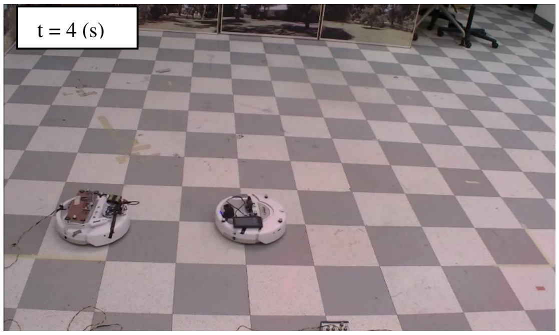


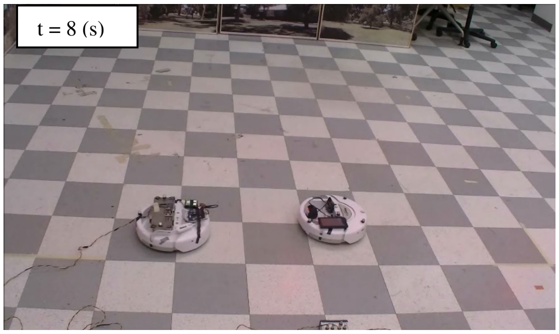


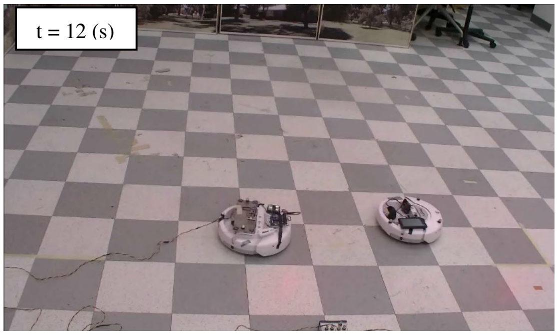


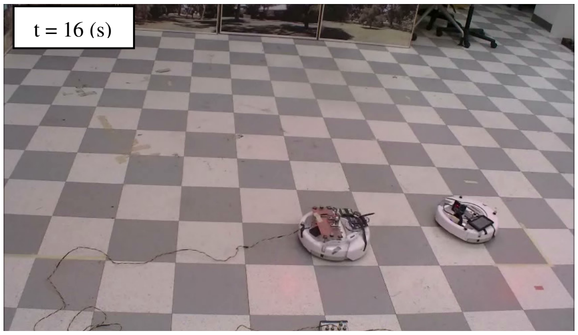
J.2 Nonlinear Controller (square path)

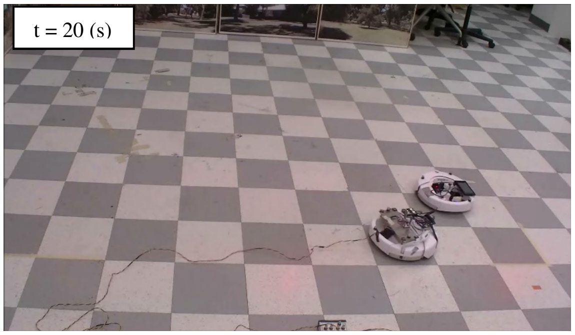


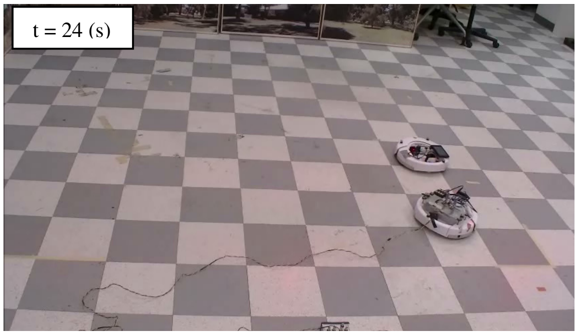


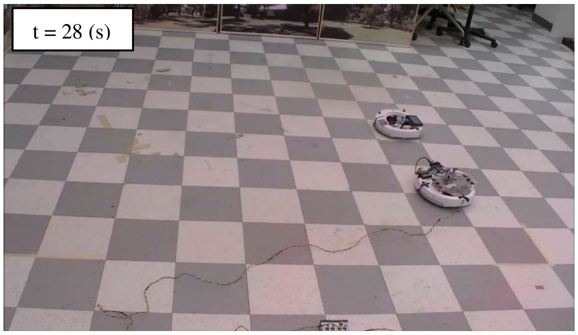


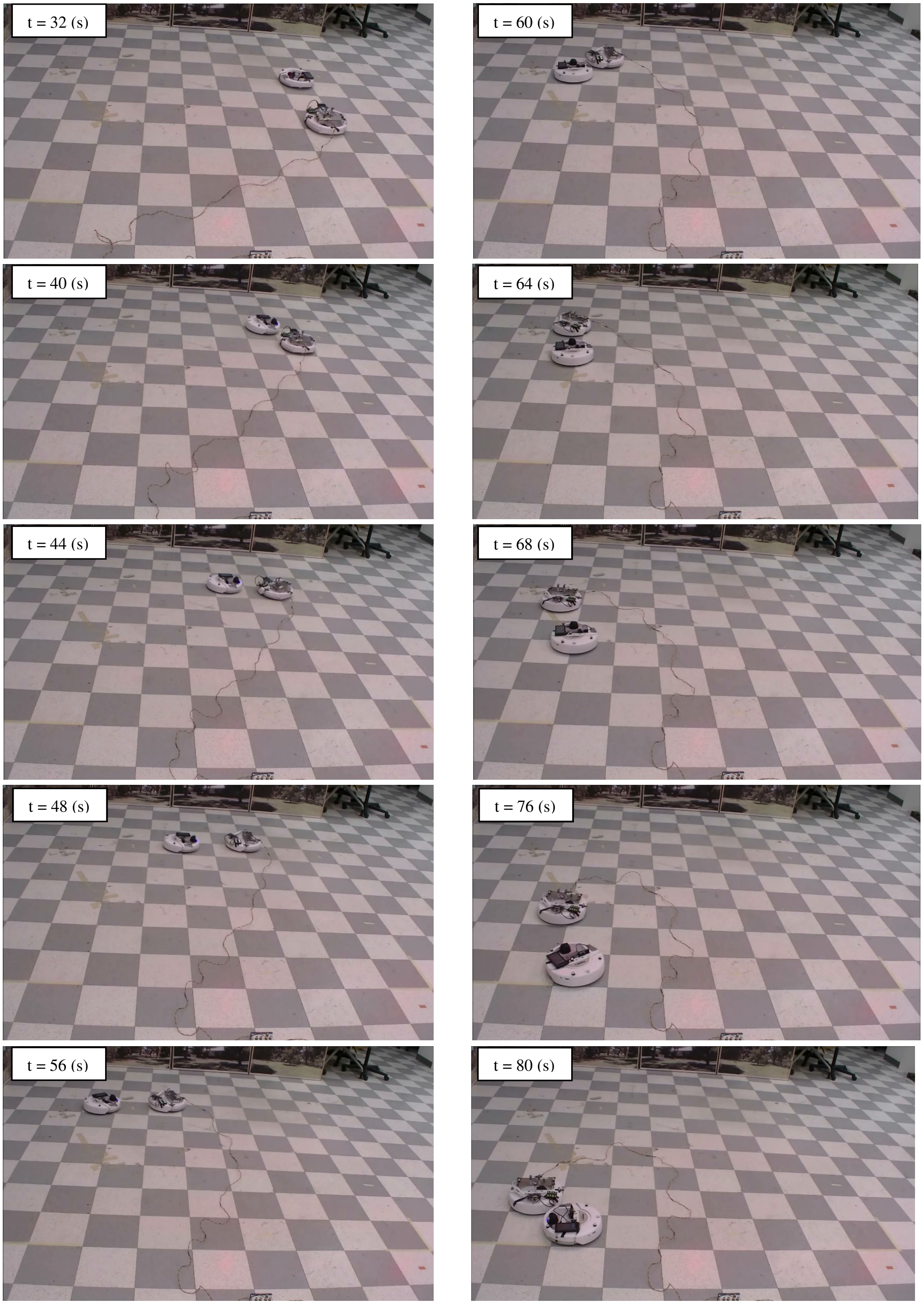




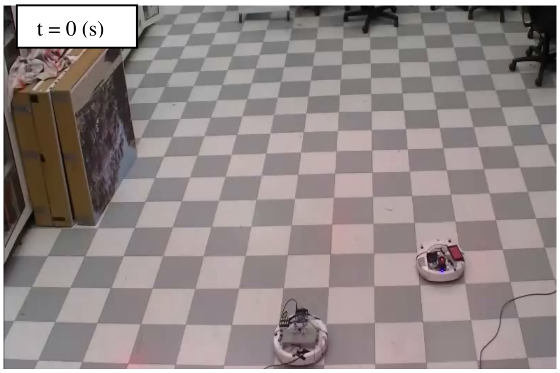


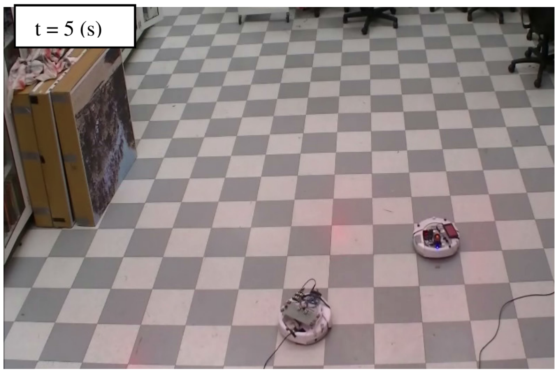


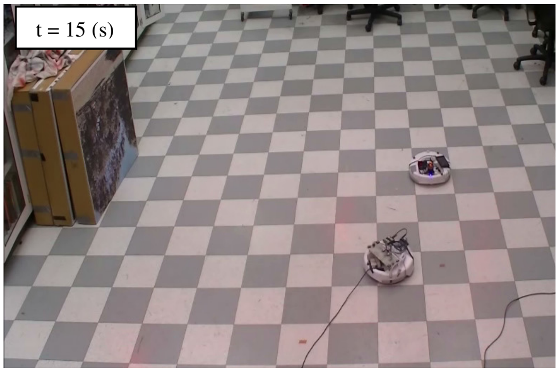


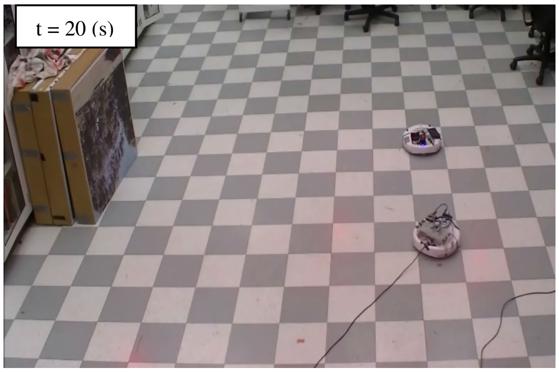

t = 32 (s)
t = 40 (s)
t = 44 (s)
t = 48 (s)
t = 56 (s)
t = 60 (s)
t = 64 (s)
t = 68 (s)
t = 76 (s)
t = 80 (s)

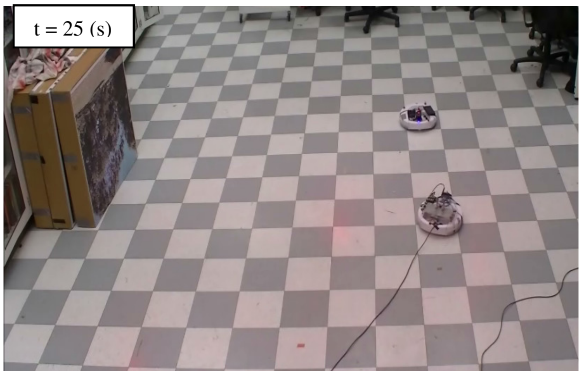
J.3 Quadratic-linear Controller (straight path)

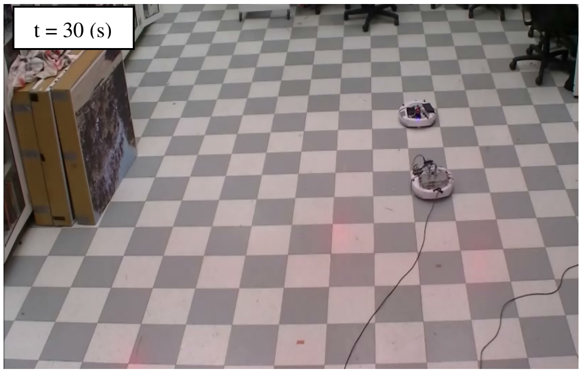


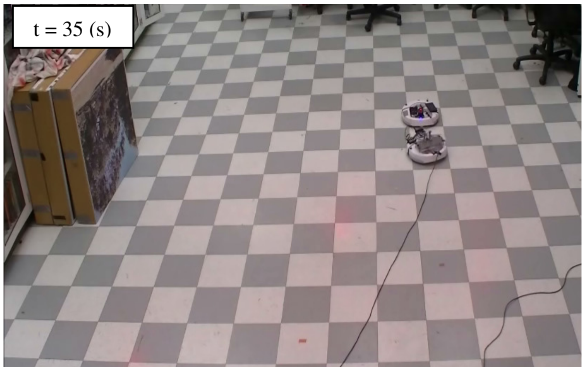


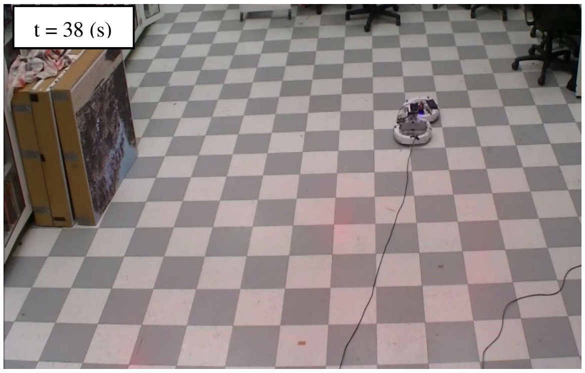




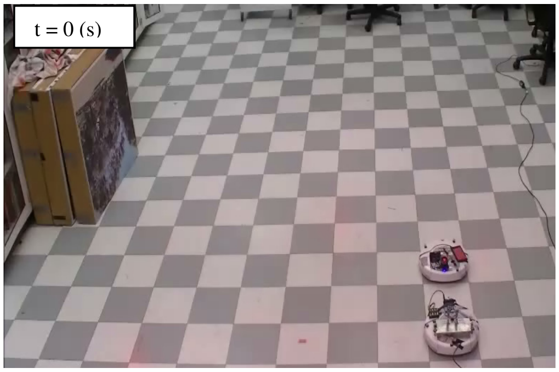


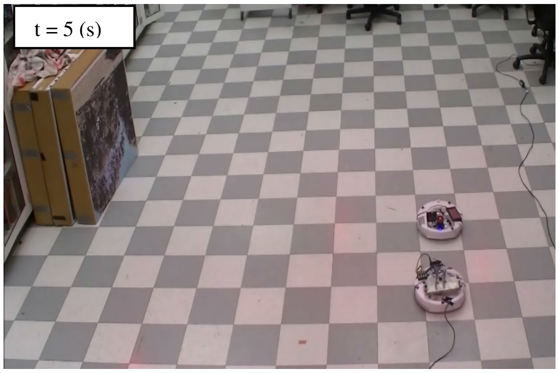


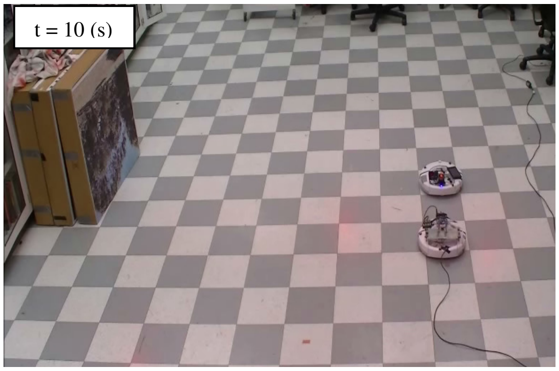


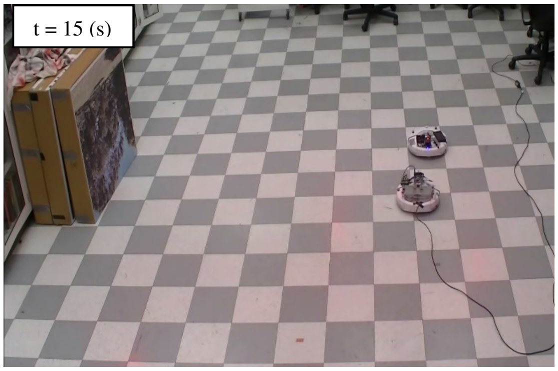


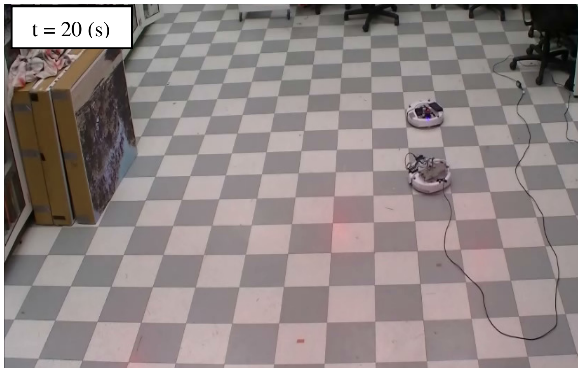
J.4 Quadratic-linear Controller (square path)

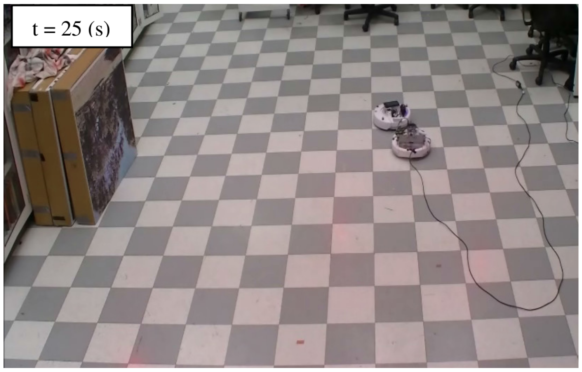


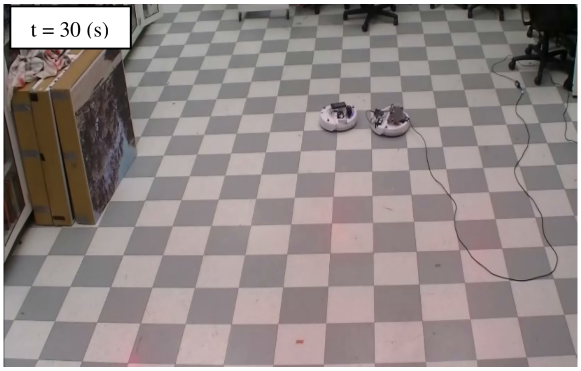


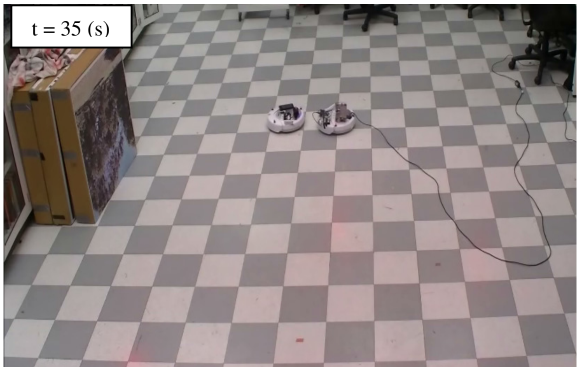


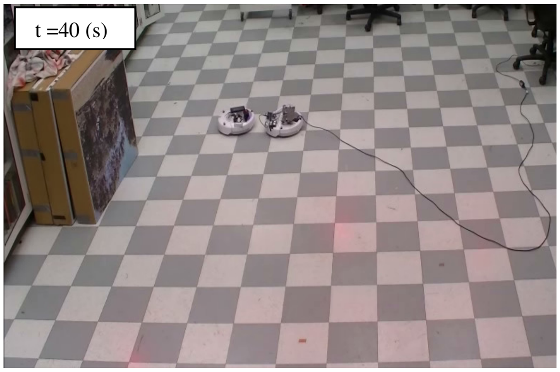


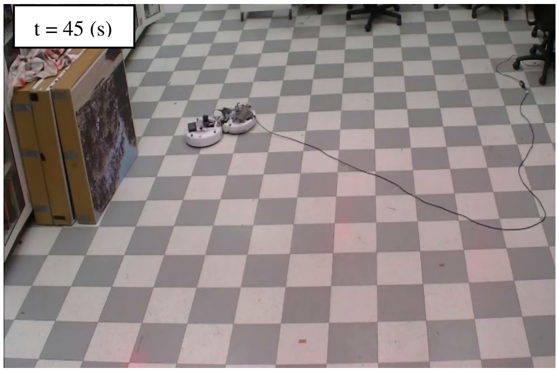


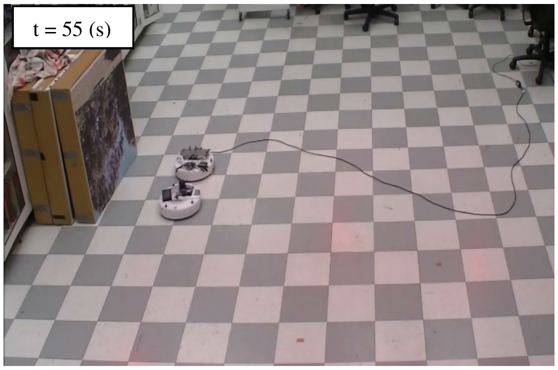


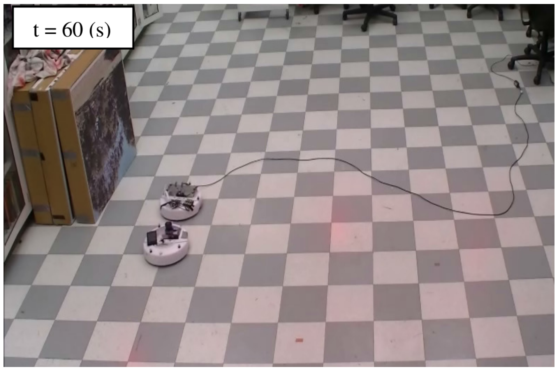


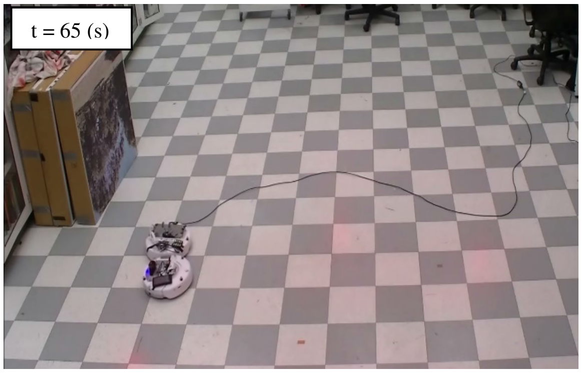


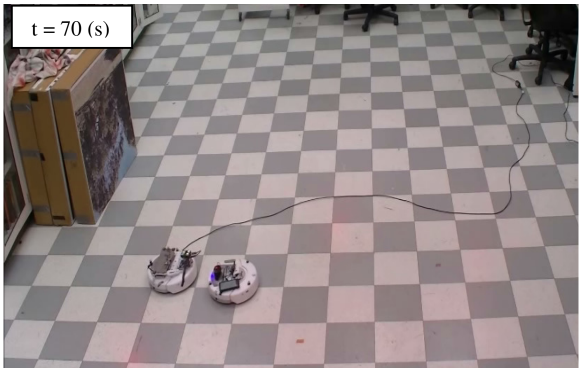

t =40 (s)

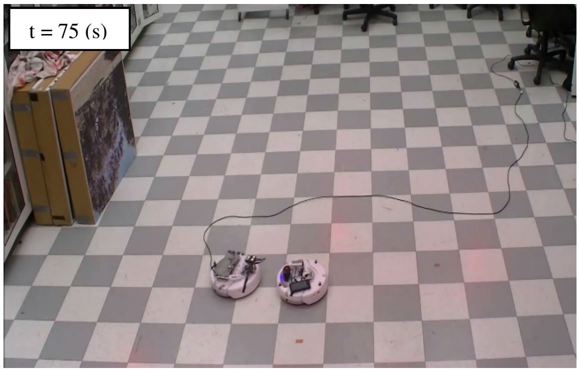

t = 45 (s)

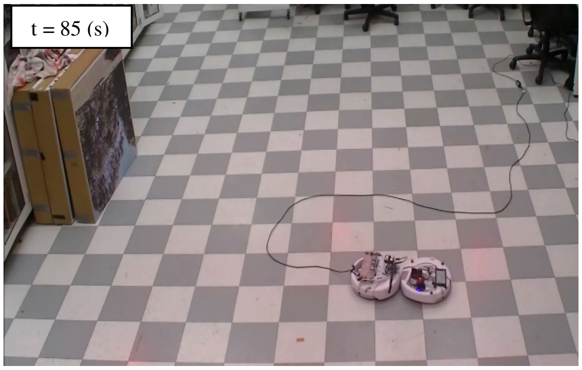

t = 55 (s)


t = 60 (s)


t = 65 (s)


t = 70 (s)


t = 75 (s)


t = 85 (s)

Bibliography

[1] W.-C. Wu, N. C. Cheng, C.-H. Hsieh, H.-C. Huang, and O. T.-C. Chen, “Hearing aid system with 3D sound localization,” in *TENCON 2007 - 2007 IEEE Region 10 Conference*, 2007, pp. 1–4.

[2] J.-M. Valin, F. Michaud, B. Hadjou, and J. Rouat, “Localization of simultaneous moving sound sources for mobile robot using a frequency-domain steered beamformer approach,” in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA ’04. 2004*, 2004, pp. 1033–1038 Vol.1.

[3] I. Marković and I. Petrović, “Speaker localization and tracking with a microphone array on a mobile robot using von Mises distribution and particle filtering,” *Robotics and Autonomous Systems*, vol. 58, no. 11, pp. 1185–1196, Nov. 2010.

[4] M. KUMON and S. UOZUMI, “Binaural Localization for a Mobile Sound Source,” *Journal of Biomechanical Science and Engineering*, vol. 6, no. 1, pp. 26–39, 2011.

[5] S. B. Andersson, A. a. Handzel, V. Shah, and P. S. Krishnaprasad, “Robot phonotaxis with dynamic sound-source localization,” in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA ’04. 2004*, 2004, pp. 4833–4838 Vol.5.

[6] H. Sun, P. Yang, L. Zu, and Q. Xu, “A Far Field Sound Source Localization System for Rescue Robot,” in *International Conference on Control, Automation and Systems Engineering (CASE)*, 2011, pp. 1–4.

[7] H. Sun, P. Yang, Z. Liu, L. Zu, and Q. Xu, “Microphone array based auditory localization for rescue robot,” in *Chinese Control and Decision Conference (CCDC)*, 2011, pp. 606–609.

[8] J. Huang and T. Supaongprapa, “A model-based sound localization system and its application to robot navigation,” *Robotics and Autonomous Systems*, vol. 27, no. 4, pp. 199–209, 1999.

[9] J. C. Murray, H. R. Erwin, and S. Wermter, “Robotic sound-source localisation architecture using cross-correlation and recurrent neural networks.,” *Neural networks : the official journal of the International Neural Network Society*, vol. 22, no. 2, pp. 173–89, Mar. 2009.

[10] N. Uchiyama, S. Sano, and A. Yamamoto, “Sound source tracking considering obstacle avoidance for a mobile robot,” *Robotica*, vol. 28, no. 07, pp. 1057–1064, Jan. 2010.

[11] F. Arvin, S. Doraisamy, K. Samsudin, and A. R. Ramli, “Self-localization of swarm robots based on voice signal acquisition,” in *International Conference on Computer and Communication Engineering (ICCCE’10)*, 2010, no. May, pp. 1–5.

[12] L. M. Kaplan and N. Molnar, “Maximum likelihood methods for bearings-only target localization,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, 2001, vol. 5, pp. 3001–3004.

[13] J. Valin, F. Michaud, J. Rouat, and D. Letourneau, “Robust sound source localization using a microphone array on a mobile robot,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, 2003, vol. 2, pp. 1228–1233.

[14] Y. Sasaki, S. Kagami, Y. Tamai, and H. Mizoguchi, “2D Sound Source Localization on a Mobile Robot with a Concentric Microphone Array,” in *IEEE International Conference on Systems, Man and Cybernetics*, 2005, vol. 4, pp. 3528–3533.

[15] Y. Tamai, Y. Sasaki, S. Kagami, and H. Mizoguchi, “Three ring microphone array for 3D sound localization and separation for mobile robot audition,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 4172–4177.

[16] S. Kagami, S. Thompson, Y. Sasaki, H. Mizoguchi, and T. Enomoto, “2D sound source mapping from mobile robot using beamforming and particle filtering,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 3689–3692.

[17] H. Yi and W. Chu-na, “A new moving sound source localization method based on the time difference of arrival,” in *International Conference on Image Analysis and Signal Processing*, 2010, pp. 118–122.

[18] B. Webb, “Robots, crickets and ants: models of neural control of chemotaxis and phonotaxis.,” *Neural networks : the official journal of the International Neural Network Society*, vol. 11, no. 7–8, pp. 1479–1496, Oct. 1998.

[19] K. Song, Q. Liu, and Q. Wang, “Olfaction and hearing based mobile robot navigation for odor/sound source search.,” *Sensors (Basel, Switzerland)*, vol. 11, no. 2, pp. 2129–54, Jan. 2011.

[20] S. Young and M. Scanlon, “Detection and localization with an acoustic array on a small robotic platform in urban environments,” 2003.

[21] H. Do and H. F. Silverman, “Stochastic particle filtering: A fast SRP-PHAT single source localization algorithm,” in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2009, no. 2, pp. 213–216.

[22] J.-F. Wang, J. Wang, B.-W. Chen, and Z.-W. Sun, “A Long-Distance Time Domain Sound Localization,” in *Ubiquitous Intelligence and Computing*, no. 1, F. Sandnes, Y. Zhang, C. Rong, L. Yang, and J. Ma, Eds. Springer Berlin / Heidelberg, 2008, pp. 616–625.

[23] B. Yang and J. Scheuing, “Cramer-Rao Bound and Optimum Sensor Array For Source Localization From Time Differences of Arrival,” in *Proceedings. (ICASSP ’05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, 2005, vol. 4, no. C, pp. 961–964.

[24] B. Yang, “Different Sensor Placement Strategies for TDOA Based Localization,” in *IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP ’07*, 2007, no. 3, pp. II–1093–II–1096.

[25] A. K. Tellakula, “Acoustic Source Localization Using Time Delay Estimation,” Indian Institute of Science, 2007.

[26] L. Tan, R. N. Miles, M. G. Weinstein, R. A. Miller, Q. Su, W. Cui, and J. Gao, “Response of a biologically inspired MEMS differential microphone diaphragm,” in *Proceedings of SPIE*, 2002, vol. 4743, pp. 91–98.

[27] A. Saito, N. Ono, and S. Ando, “Micro gimbal diaphragm for sound source localization with mimicking Ormia Ochracea,” in *Proceedings of the 41st SICE Annual Conference. SICE 2002.*, 2002, vol. 4, pp. 2159–2162.

[28] L. J. Currano, H. Liu, D. Gee, B. Yang, and M. Yu, “Microscale implementation of a bio-inspired acoustic localization device,” in *Proc. SPIE*, 2009, no. 301, p. 73210B–73210B–8.

[29] A. P. Lisiewski, H. J. Liu, M. Yu, L. Currano, and D. Gee, “Fly-ear inspired micro-sensor for sound source localization in two dimensions.,” *The Journal of the Acoustical Society of America*, vol. 129, no. 5, pp. EL166–71, May 2011.

[30] Q. Wang, “Bionic structure of mechanically coupled diaphragms for sound source localization,” *WSEAS TRANSACTIONS on SYSTEMS*, vol. 8, no. 7, pp. 855–865, 2009.

[31] C. C. Chen, Y. T. Cheng, and S. Member, “Physical Analysis of a Biomimetic Microphone with a Central-Supported (CS) Circular Diaphragm for Sound Source Localization,” *Sensors Journal, IEEE*, no. c, pp. 1–9, 2011.

[32] D. B. Ward, E. A. Lehmann, and R. C. Williamson, “Particle filtering algorithms for tracking an acoustic source in a reverberant environment,” *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 826–836, Nov. 2003.

[33] J. DiBiase, “A high-accuracy, low-latency technique for talker localization in reverberant environments using microphone arrays,” 2000.

[34] C. Knapp and G. Carter, “The generalized correlation method for estimation of time delay,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 4, pp. 320–327, Aug. 1976.

[35] H. Liu and M. Shen, “Continuous sound source localization based on microphone array for mobile robots,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, no. 4, pp. 4332–4339.

[36] J. Han, S. Han, and J. Lee, “The Tracking of a Moving Object by a Mobile Robot Following the Object’s Sound,” *Journal of Intelligent & Robotic Systems*, vol. 71, no. 1, pp. 31–42, Sep. 2012.

[37] R. N. Miles, D. Robert, and R. R. Hoy, “Mechanically coupled ears for directional hearing in the parasitoid fly Ormia ochracea.,” *The Journal of the Acoustical Society of America*, vol. 98, no. 6, pp. 3059–70, Dec. 1995.

[38] D. Robert, R. N. Miles, and R. R. Hoy, “Directional hearing by mechanical coupling in the parasitoid fly Ormia ochracea.,” *Journal of comparative physiology. A, Sensory, neural, and behavioral physiology*, vol. 179, no. 1, pp. 29–44, Jan. 1996.

[39] H. Liu, “Fly ear inspired optical directional microphones: from Bio-physics to Sensor development,” University of Maryland, College Park, 2009.

[40] C. Chang and A. Sahai, “Estimation bounds for localization,” *Sensor and Ad Hoc Communications and …*, vol. 00, no. C, pp. 415–424, 2004.

[41] N. Patwari, A. O. Hero, M. Perkins, N. S. Correal, and R. J. O’Dea, “Relative location estimation in wireless sensor networks,” *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2137–2148, Aug. 2003.

[42] S. Yazdanfar, C. Yang, M. Sarunic, and J. Izatt, “Frequency estimation precision in Doppler optical coherence tomography using the Cramer-Rao lower bound.,” *Optics express*, vol. 13, no. 2, pp. 410–6, Jan. 2005.

[43] S. Kay and S. Saha, “Mean likelihood frequency estimation,” *IEEE Transactions on Signal Processing*, vol. 48, no. 7, pp. 1937–1946, Jul. 2000.

[44] F. Rice, B. Cowley, B. Moran, and M. Rice, "Cramer-Rao lower bounds for QAM phase and frequency estimation," *IEEE Transactions on Communications*, vol. 49, no. 9, pp. 1582–1591, 2001.

[45] A. Weiss and B. Friedlander, "Range and bearing estimation using polynomial rooting," *IEEE Journal of Oceanic Engineering*, vol. 18, no. 2, pp. 130–137, Apr. 1993.

[46] B.-G. Song, "Angle of arrival estimation of plane waves propagating in random media," *The Journal of the Acoustical Society of America*, vol. 99, no. 3, p. 1370, 1996.

[47] B. Friedlander, "On the computation of the Cramer-Rao bound for ARMA parameter estimation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 4, pp. 721–727, Aug. 1984.

[48] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory (v. 1)*. Prentice Hall, 1993, pp. 27–81.

[49] M. Akcakaya and A. Nehorai, "Performance analysis of the Ormia ochracea's coupled ears.," *The Journal of the Acoustical Society of America*, vol. 124, no. 4, pp. 2100–5, Oct. 2008.

[50] H. J. Liu, M. Yu, and X. M. Zhang, "Biomimetic optical directional microphone with structurally coupled diaphragms," *Applied Physics Letters*, vol. 93, no. 24, p. 243902, 2008.

[51] E. K. P. Chong and S. H. Zak, *An Introduction to Optimization (Google eBook)*. John Wiley & Sons, 2004, pp. 113–134.

[52] "What Is Fuzzy Logic? :: Getting Started (Fuzzy Logic Toolbox^{TM})." [Online]. Available: http://www.mathworks.com/help/toolbox/fuzzy/fp72.html. [Accessed: 08-Aug-2012].

[53] L.-X. Wang, *A Course In Fuzzy Systems and Control*, 1st ed. Printice Hall, 1996, pp. 1–116.

[54] "Fuzzy Inference Systems :: Tutorial (Fuzzy Logic Toolbox^{TM})." [Online]. Available: http://www.mathworks.com/help/toolbox/fuzzy/fp351dup8.html. [Accessed: 17-Aug-2012].

[55] "What Is Sugeno-Type Fuzzy Inference?" [Online]. Available: http://www.mathworks.com/help/fuzzy/what-is-sugeno-type-fuzzy-inference.html.

[56] "anfis and the ANFIS Editor GUI :: Tutorial (Fuzzy Logic Toolbox^{TM})." [Online]. Available: http://www.mathworks.com/help/toolbox/fuzzy/fp715dup12.html. [Accessed: 08-Aug-2012].

[57] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, 1st ed. Prentice Hall, 1997, p. 614.

[58] "Fuzzy Clustering :: Tutorial (Fuzzy Logic Toolbox^{TM})." [Online]. Available: http://www.mathworks.com/help/toolbox/fuzzy/fp310.html. [Accessed: 08-Aug-2012].

[59] "Find cluster centers with subtractive clustering - MATLAB." [Online]. Available: http://www.mathworks.com/help/toolbox/fuzzy/subclust.html. [Accessed: 08-Aug-2012].

[60] J. Benesty, J. Chen, and Y. Huang, "Time-Delay Estimation via Linear Interpolation and Cross Correlation," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 5, pp. 509–519, Sep. 2004.

[61] E. Maalouf, M. Saad, and H. Saliah, "A higher level path tracking controller for a four-wheel differentially steered mobile robot," *Robotics and Autonomous Systems*, vol. 54, no. 1, pp. 23–33, Jan. 2006.

[62] J. R. Zhang, S. J. Xu, and A. Rachid, "Path tracking control of vehicles based on Lyapunov approach," *American Control Conference, …*, vol. 3, pp. 2132–2137, 2002.

[63] R. M. DeSantis, R. Hurteau, O. Alboui, and B. Lesot, "Experimental stabilization of tractor and tractor-trailer like vehicles," in *Proceedings of the IEEE Internatinal Symposium on Intelligent Control*, 2002, pp. 188–193.

[64] J. Zhang, S. J. Xu, and A. Rachid, "Sliding mode controller for automatic steering of vehicles," in *Annual Conference of the IEEE Industrial Electronics Society (Cat. No.37243)*, 2001, vol. 3, no. 2, pp. 2149–2153.

[65] J. R. Zhang, S. J. Xu, and A. Rachid, "Sliding mode controller for automatic path tracking of vehicles," in *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, 2002, vol. 5, no. 2, pp. 3974–3979.

[66] A. Shaout, M. A. Jarrah, H. Al-Araji, and K. Al-Tell, "A nonlinear optimal four wheels steering controller," in *Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems (Cat.No.CH37144)*, 2000, vol. 3, pp. 1426–1429.

[67] A. Gholipour and M. Yazdanpanah, “Dynamic tracking control of nonholonomic mobile robot with model reference adaptation for uncertain parameters,” in *Proc. of the 2003 European Control Conference*, 2003.

[68] T. H. Bui, T. T. Nguyen, T. L. Chung, and S. B. Kim, “A simple nonlinear control of a two-wheeled welding mobile robot,” *International Journal of Control, Automation, and Systems (IJCAS)*, vol. 1, pp. 35–42, 2003.

[69] J. E. Normey-Rico, I. Alcalá, J. Gómez-Ortega, and E. F. Camacho, “Mobile robot path tracking using a robust PID controller,” *Control Engineering Practice*, vol. 9, no. 11, pp. 1209–1214, Nov. 2001.

[70] A. Dixon, W.E., Dawson, D.M., Zergeroglu, E., Behal, *Lecture Notes in Control and Information Sciences*. Springer, 2001, p. 195.

[71] “Create Open Interface_v2.pdf.” [Online]. Available: http://www.irobot.com/filelibrary/pdfs/hrd/create/Create Open Interface_v2.pdf. [Accessed: 10-Jun-2013].

[72] “Electret condenser microphone,” 2008. [Online]. Available: http://www.adafruit.com/datasheets/CMA-4544PF-W.pdf.

[73] “Electret Microphone Amplifier - MAX4466 with Adjustable Gain -.” [Online]. Available: http://www.adafruit.com/products/1063#Description.

[74] “Microphone Preamplifiers with Complete Shutdown.” [Online]. Available: http://www.adafruit.com/datasheets/MAX4465-MAX4469.pdf.

[75] “Motion Capture Systems from Vicon.” [Online]. Available: http://www.vicon.com/index.html. [Accessed: 24-Jun-2013].

[76] C. Pang, M. Yu, X. M. Zhang, a. K. Gupta, and K. M. Bryden, “Multifunctional optical MEMS sensor platform with heterogeneous fiber optic Fabry–Pérot sensors for wireless sensor networks,” *Sensors and Actuators A: Physical*, vol. 188, pp. 471–480, Dec. 2012.

[77] L. Meirovitch, *Fundamentals of Vibrations*, 1st ed. Mcgraw-Hill, 2001, p. 816.