Sound Source Localization and Tracking using Kalman Filtering

By
Rabeea Shahid

Department of Electronics
Faculty of Natural Sciences
Quaid-i-Azam University, Islamabad
2021

In The Name of Allah, The Most Merciful, The Most Beneficent

Sound Source Localization and Tracking using Kalman Filtering

By

Rabeea Shahid

A THESIS SUBMITTED IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE IN ELECTRONICS

Supervised By

Dr. Syed Aqeel Abbas Bukhari

Department of Electronics
Faculty of Natural Sciences
Quaid-i-Azam University, Islamabad
2021

Declaration

I "Rabeea Shahid" hereby solemnly declare that this thesis entitled "Sound Source Localization and Tracking using Kalman Filtering", submitted by me for the partial fulfillment of Bachelor of Science in Electronics.

Dated: ________________ Signature: ________________

Dedication

I am feeling great honor and pleasure to dedicate this research work to

My Brother,
for being the
wind beneath my wings.

Acknowledgment

First and foremost I praise and acknowledge Allah Almighty, the Lord and creator of the heavens and earth. All respect and gratitude goes to the Holy Prophet Hazrat Muhammad (SAW) who enlightens our hearts with the light of Islam and whose way of life has been a continuous guidance for us.

I would like to express my heartiest gratitude to my venerated supervisor Dr. Syed Aqeel Abbas Bukhari for his consistent support and supervision in every step of research. I am indebted to his guidance specifically throughout my thesis. I am very privileged and blessed to his guidance in my corner. I owe him many thanks.

I offer my deepest sense of gratitude, profound respect and tribute to all those teachers specially Prof. Dr. Qaisar Abbas Naqvi, Prof. Dr. M. Farhan Saif, Dr. Muhammad Aqeel Ashraf, and Dr. Arshad Hussain who have enlightened my knowledge through out my academic career and guidance throughout research work.

My sincere thanks go to my parents for their love and support throughout my life. I owe my loving thanks to my sister, and my whole family for their great love and support. Without their support and encouragement it would have been impossible for me to complete this work. I would like to say thanks all my friends and classmates for their cooperation and moral support especially Muntaha Bajwa, Laraib Fatima, Bushra Noor and Rafia Amjad. No acknowledge would ever adequately express obligations to my teammate Urwa Ejaz. Finally, I would like say thanks to Sana Ejaz who have been supportive and cooperative to me during my research work.

Rabeea Shahid

Abstract

Beamforming and Kalman filtering have vast applications in signal processing. It has been used widely in source location estimation algorithms. The first part of this thesis deals with the estimation of the location of sound source using microphone array processing and beamforming. The algorithm based on power maximization gives raw position estimates of the source. An array is composed of a set of microphones that are placed in certain locations. Microphone array is to give high gain to signal from one desired direction and attenuate the other. This is implemented by maximizing the power of beamformer's output in one specific direction.

Following that, a source tracking system based on the Kalman filter algorithm is proposed. The microphone array approach is employed in the first step to finding the sound source location. The Kalman filtering method is then used to refine those estimations. For the motion of the target source, three system models are proposed: motion along with a single coordinate, constant velocity, and constant acceleration model. The outcome of combining these two techniques is a robust system to sound source localization and tracking.

Contents

List of Figures

Chapter 1

Introduction

The modern study of sound source localization started in the late 19th century, however, researchers and philosophers were interested in the matter since the time of ancient Greeks. The question that arose at the time was how humans were able to detect a source based on the sound it produced. A "Garden Experiment" was conducted by Lord Rayleigh, he concluded the experiment with the statement that the ability to determine the position of people who spoke in the garden could be explained by a binaural ratio of sound level at each ear. The current study of the acoustic cues employed for sound source localisation began with this kind of examination. Initially, during the first world war, the tracking and destruction of armaments were accomplished utilizing the sound they produced.

Hearing aids, headsets, and meeting rooms are just a few examples of how localization technology has advanced and become a part of our daily lives. This technique can be employed to handle a variety of challenges, including tracking, voice improvement, and human-robot interface, among others. Using microphone arrays, numerous algorithms can be used to pinpoint the location of the sound source.

Billingsley in 1974, pioneered the microphone antenna, which has since experienced significant advances due to the availability of better data gathering and computer technology. A microphone array is made up of a couple of microphone antennas arranged in such a way that spatial information is recorded well. The geometry of the microphone array may play a crucial role in the development of the processing algorithms, based on the nature of the applications. In source localization, for example, the array geometry must be determined to effectively localize a source. Furthermore, a standard geometry can sometimes make estimation easier, which is why uniform linear and circular arrays are commonly utilized. These two geometries currently dominate the market. The array's advantage is that it can suppress background noise, allowing researchers to explore sources in reverberant or loud situations. When sound sources are explored in difficult environments, beamforming through microphone arrays has become a regular procedure.

The signals from an array of microphones can be utilized to examine acoustic sources in a variety of ways. One of the approaches offered is beamforming. It is the process of the desired signal coming from a single direction and impinging on several sensors at the receiver. With the advancement of the beamforming approach, the number of microphones, sampling frequency, and dynamic range of the analysis could all be expanded. For the localization of sound sources on moving objects, such as flying airplanes, high-speed trains, moving automobiles, and open rotors such as helicopter and wind turbine rotors, beamforming is essential.

The next stage is to track a mobile source using the Kalman filtering approach after the source has been localized using the above procedures. Video compression, video surveillance, vision-based control, human-computer interfaces, medical imaging, augmented reality, and robotics are all implementations of object detection and tracking in computer vision. It is also useful in video databases for features like content-based indexing and retrieval.

Kalman filtering has been popular for the past few years for tracking a moving object. Since Sutherland's effort at the beginning of 1960, the systems for tracking heads are dealt with. Due to the efficiency of Kalman filtering to track an object, the majority of researchers make use of or compare it with other techniques. Kalman filter is composed of a set of mathematical equations. Those equations are used to effectively minimize the mean squared error covariance by iteratively evaluating the state and error covariance. Kalman filters are easy to understand and work with also they demand less computation capability.

1.1 Significance of the Project

In an effort to assess the acoustic information, the sense of hearing plays a crucial role as the other senses are not able to do so. The sense of hearing is a miraculous gift from nature. The ability of humans to be able to accompany a conversation in the congregation by the virtue of this bestowal makes it more remarkable.

With the rise in the aptness of sound source localization, it plays a vital role in certain areas. Sound source localization has been used in several fields for locating the sound sources including videogames, audio surveillance, hands-free acquisition in a car, musical control interfaces, system monitoring, voice recognition, virtual reality, human-machine interaction, and teleconferencing systems.

Sound source localization and tracking attracted considerable attention in the past few years due to its significance in a variety of future applications. Several applications including automatic speech recognition, motion planning, and beamforming took the benefit of the acquaintance of localization and tracking. Further applications in acoustic scene analysis such as robotics, autonomous systems, smart environments, and hearing aids had a powerful impact on them.

1.2 Objectives

The objectives of the project are,

- To accurately locate a sound source applying the TDOA technique.
- To model a tracking system for a nonstationary sound source.

1.3 Organization Of the Thesis

The second chapter begins with a discussion of the fundamentals of beamforming using microphone arrays. Following this, the project's system model for sound source localization is described. This chapter also presents and explains the algorithm used to locate the source. Finally, the simulation results for the algorithm are discussed, and a comparison of results is made to determine which approximations provide the best localization results. Various noise values are tested in the simulation for comparison and the creation of a robust localization model.

The tracking model for a mobile sound source is discussed in Chapter 3. The chapter introduces a traditional tracking algorithm, Kalman filtering. For a better understanding of the concept, the fundamentals of the Kalman Filter are discussed. Following that, the models proposed for this project are presented. The models include all of the necessary elements for simulating the trajectory of the moving source. The simulation results of the algorithm are presented at the end of this chapter to demonstrate the effective result of this tracking technique.

The thesis is concluded in Chapter 4 with a discussion of what parameters should be optimised in this model to make it more effective and to produce better results.

Chapter 2

Microphone Beamforming for Source Localization

2.1 Delay and Sum Beamforming

Delay and sum beamformer is easy to understand and work with. It is the process in which a source signal arrives from far-field and is received at an array of sensors. The signal is received at each sensor with some delay with respect to the reference microphone. After some weight compensation, the linear combination of the outputs gives a high gain as a result of synchronization of the received signals as indicated in fig 2.1

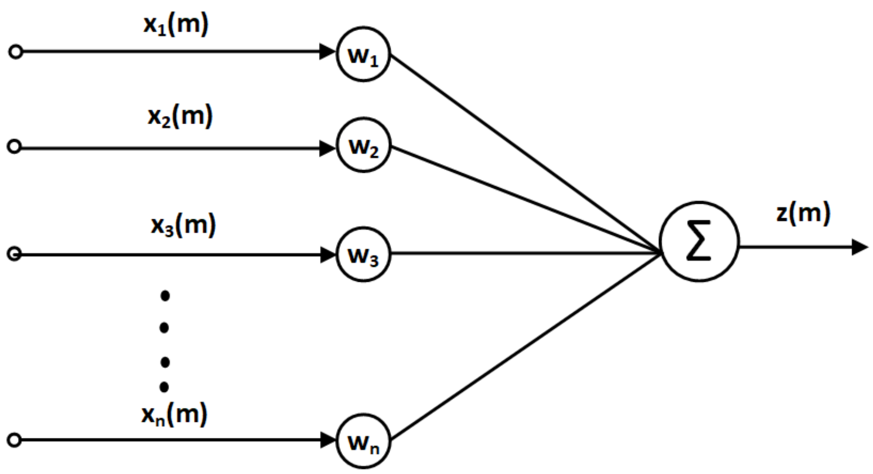


Figure 2.1: Beamformer

One of its most useful advantage is that it can enhance a signal that has been corrupted by noise, reverberation or other sound sources. It is basically comprised of two parts: Synchronization and weight and sum. Synchronization is the process of imparting proper phases on the outputs of the microphones in such a way that they all constructively interfere and the signals coming from desired direction synchronize. On the other hand, weight and sum is to make a linear combination of the output signals by weighting them appropriately. Beamforming can also be used to achieve information about the direction of arrival(DOA) and number of incoming source signals. For a given input signal, beamformer maximizes the output power.

2.2 Array Model

The array model for direction of arrival(DOA) estimation used here is Uniform Linear Array Model.

2.2.1 Uniform Linear Array(ULA)

Uniform Linear array structure (ULA) is used in this thesis for the geometry of the microphone array. The uniform linear array consists of n identical sensors as shown in figure 2.2

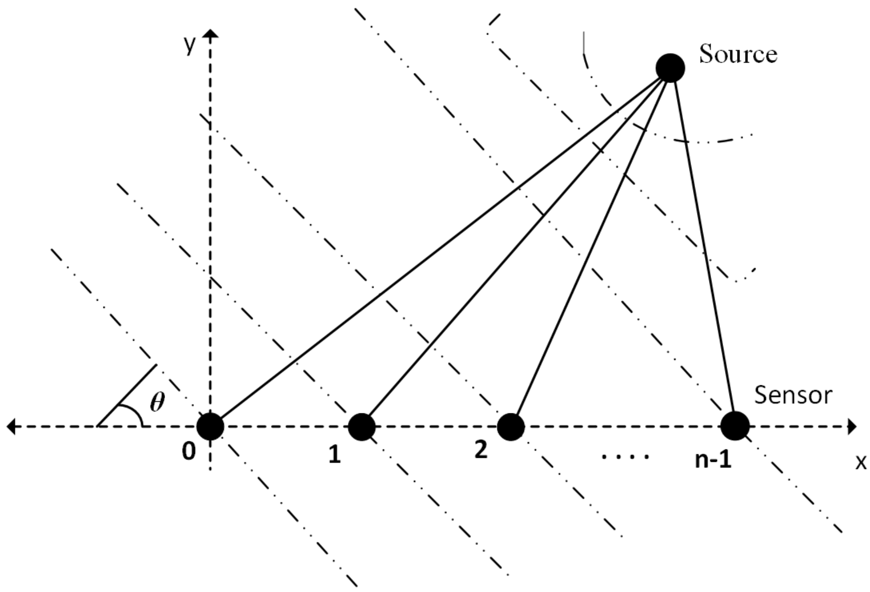


Figure 2.2: Uniform Linear Array(ULA)

An array element is represented as a point receiver on the spatial coordinates. As shown in figure 2.2 the 2D coordinates of the sensors are written as $l_n = (x_n y_n)^T$. The signal arriving from the source to the array are considered plane wavefronts. This is because the source is assumed to be present in the far-field region.

The field estimated at the microphone array can be expressed as,

$$E(l,t) = s(t)e^{jwt-k(x_n cos\theta + y_n sin\theta)} \tag{2.1}$$

Let r be the separation between two adjacent microphones and θ be the angle of arrival of the incoming source signal estimated anticlockwise from line normal to the sensors. Using the first microphone as the reference, the delay is calculated using the following equation,

$$\tau = \frac{(n-1)rcos\theta}{c} \tag{2.2}$$

where c is the speed of the incoming sound signal and the condition on the separation between two adjacent microphones can be written as,

$$r < \frac{\lambda}{2} \tag{2.3}$$

and λ being the wavelength of the source signal.

For a source signal received at the array,there results an array propagation vector which can be written as,

$$v(\theta) = \begin{bmatrix} v_1(\theta) & , & v_2(\theta), & . & . & . & . & , v_n(\theta) \end{bmatrix}$$

Considering that all the array elements have same directivity,

$$d_1(\theta) = d_2(\theta) = ... = d_n(\theta)$$

The array steering vector for uniform linear array(ULA) can be expressed as,

$$v(\theta) = d(\theta) \begin{bmatrix} 1 & e^{-jrcos(\theta)w/c}, & . & . & . & , e^{-j(n-1)rcos(\theta)w/c} \end{bmatrix} \tag{2.4}$$

Assuming that all the microphones are identical therefore their directivity is equal to unity. So the array steering vector becomes,

$$v(\theta) = \begin{bmatrix} 1 & e^{-jrcos(\theta)w/c}, & . & . & . & , e^{-j(n-1)rcos(\theta)w/c} \end{bmatrix} \tag{2.5}$$

2.3 Source Localization Algorithm

The algorithm for the sound source localization is based on conventional beamforming. An array of sensors is placed that receives the source signal incident on the array. For the acquisition of the source,a scan range is set in which the source is to be located. The scan range is set depending upon the number of array elements. Delay is estimated from each point in the scan range to all possible sensor locations. Those delays are then imparted on the output of the microphones when they are processed. In this way,signals from the microphone array constructively interfere and their amplitude is enlarged. Subsequently, power is estimated at each scan point. The source signal will be the one having maximum power amplitude.

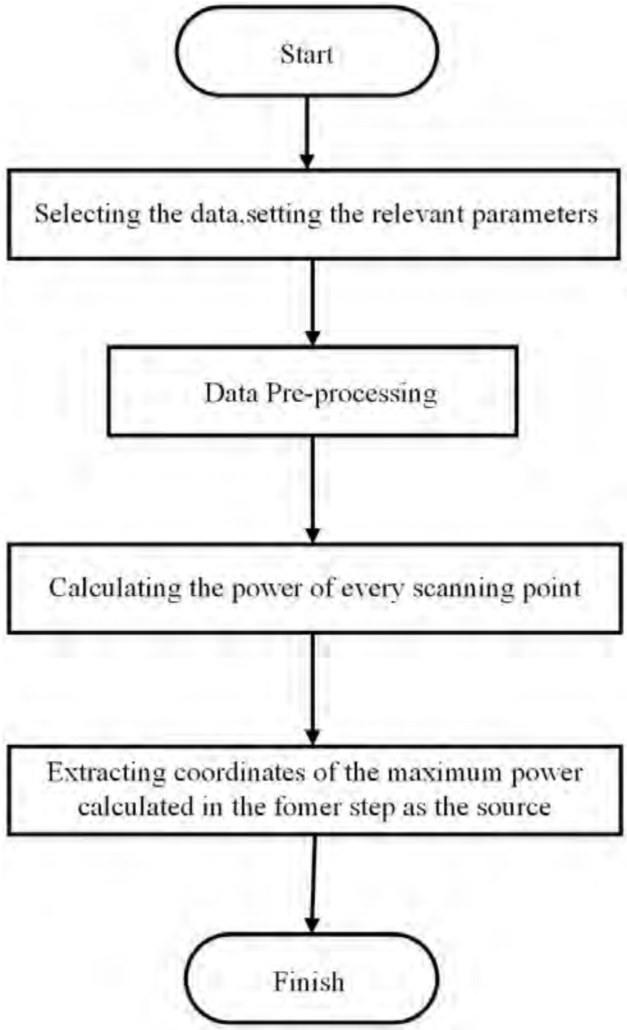


Figure 2.3: Overall Source Localization Flowchart

2.3.1 Mathematical model for source localization

Suppose that a source signal $s(m)$ is arriving from far-field and is incident on an n element microphone array. The signal received at the microphone array are pre-processed and they are discretized at arbitrary time instances for simplicity.The received signal can be expressed as,

$$x(m) = \Big[x_0(m) \quad , \quad x_1(m), \quad . \quad . \quad . \quad . \quad , x_{n-1}(m)\Big]^T \tag{2.6}$$

As the signal at each microphone is received with some delay so the vector can be written as,

$$x(m) = \Big[s(m) \quad , \quad s(m-\tau_{10}), \quad . \quad . \quad . \quad . \quad , s(m-\tau_{(n-1)0})\Big]^T \tag{2.7}$$

where τ_{i0} is the delay experienced at each microphone. From Equation 2.5 we can write x as,

$$x(m) = v(\theta)s(m)$$

and the noise added from each microphone is expressed as,

$$u(m) = \Big[u_0(m) \quad , \quad u_1(m), \quad . \quad . \quad . \quad . \quad , u_{n-1}(m)\Big]^T \tag{2.8}$$

Adding the noise in the received vector at the microphone array, so the x vector becomes,

$$x(m) = v(\theta)s(m) + u(m) \tag{2.9}$$

It is important to measure the similarity of the signals between pair of sensors as the parameters considered in here are spatial in nature. To do so cross covariance information is required which is taken from the cross-covariance matrix. As only sample estimates are available and easier to work with therefore sample covariance is estimated. The sample covariance matrix is written as,

$$\hat{R} = \frac{1}{M}\sum_{m=1}^{M} x(m)x^H(m) \tag{2.10}$$

$$\hat{R} = E[x(m)x^H(m)] \tag{2.11}$$

$$\hat{R} = E[(v(\theta)s(m) + u(m))(v(\theta)s(m) + u(m))^H]$$

$$\hat{R} = E[(v(\theta)s(m) + u(m))(v^H(\theta)s^H(m) + u^H(m))]$$

After simplifying the sample covariance matrix becomes,

$$\hat{R} = E[(v(\theta)(v^H(\theta))(s(m)s^H(m))] + E[u(m)u^H(m)] \tag{2.12}$$

As elaborated before, the power is estimated at each scan point by "steering" the array in all the directions. The scan point having maximum power amplitude is extracted as the source location. First, the output from each microphone is summed up to make a linear combination of signals. Then the array feedback is steered by multiplying it with a weight vector.

$$z(m) = \sum_{m=1}^{M} w_n^*(m)x(m) \tag{2.13}$$

The power is measured at each scan point through the following equation,

$$P(\theta) = \frac{1}{M}\sum_{m=1}^{M} |z(m)|^2 \tag{2.14}$$

$$P(\theta) = \frac{1}{M}\sum_{m=1}^{M} w^H x(m)$$

$$P(\theta) = \frac{1}{M}\sum_{m=1}^{M} w^H x(m) x^H(m) w$$

From Equation 2.10 we can write,

$$P(\theta) = \frac{1}{M}\sum_{m=1}^{M} w^H \hat{R} w \tag{2.15}$$

where the location having maximum power is estimated from,

$$(x_n, y_n) = max[\frac{1}{M}\sum_{m=1}^{M} w^H \hat{R} w] \tag{2.16}$$

The power estimated at each location is normalized to 1, so that the maximum power amplitude is equal to unity.

2.4 Simulation Results

2.4.1 In the Presense of Noise

Simulation results for different noise amplitudes are shown below. For different noise amplitudes source location is estimated using different cell sizes so that the localization is done more precisely. The variable r used in cell size represents the separation of the microphones.

- **Noise Amplitude–1**

  It can be seen that in this case $SNR = 1$ i.e., the amplitude of the noise is equal to the amplitude of the source signal. Therefore, decreasing the cell size has no impact on improving the source localization results. As the source cannot be distinguished from the noise.

- **Noise Amplitude–0.5**

  When the noise amplitude was 1, the estimated and real locations had no effect on reducing cell size, as shown in Figure 2.4. However, when the noise is reduced to 0.5, the results are slightly improved when the grid size is $r/10$ x $r/10$, as indicated in Figure 2.5(d). However, they are not as precise as they should be.

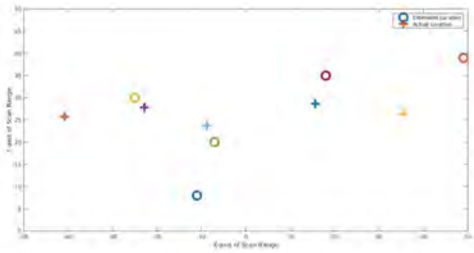
- **Noise Amplitude–0.25**

  Figure 2.6 indicates that the noise amplitude is reduced. The localization results have improved and the estimated and actual localizations become closer.

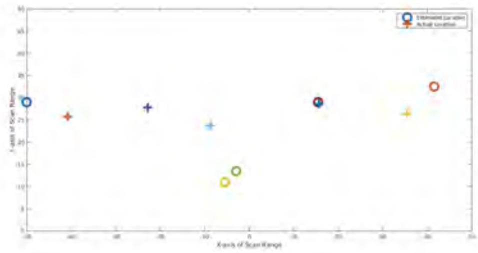
- **Noise Amplitude–0.1** As indicated in the Figure 2.7, when the noise amplitude is set to 0.1 the localization results are much better as compared to higher noise amplitudes. Since the SNR is high in this case and the noise has little effect on the accuracy of the results.

2.4.2 In the Absence of Noise

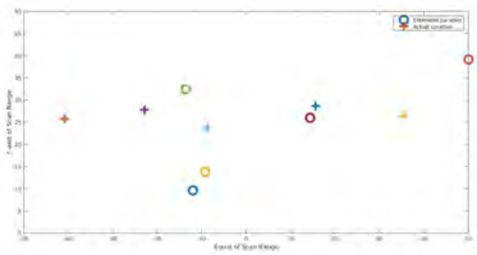
In this case, the source localization is performed in the absence of noise. Since noise has no effect on the algorithm, almost perfect localization results are obtained as the cell size is reduced even further. Figure 2.8 illustrates that when the cell size is large, the estimated and actual locations are far apart. However, as the cell size decreases, the estimated and actual location points coincide.



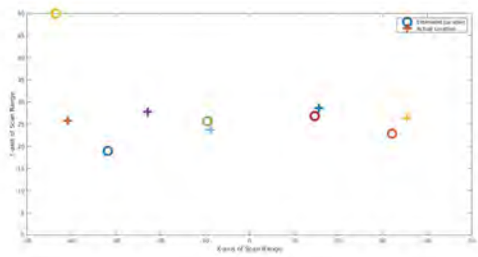
(a) Cell Size: rxr



(b) Cell Size: $\frac{r}{2}$x$\frac{r}{2}$

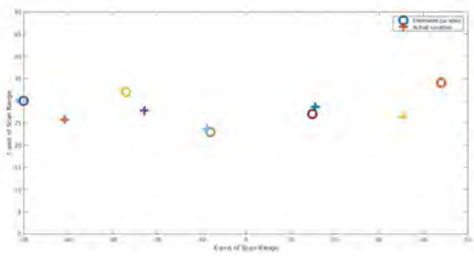


(c) Cell Size: $\frac{r}{5}$x$\frac{r}{5}$

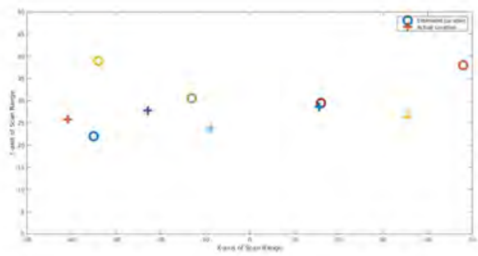


(d) Cell Size: $\frac{r}{10}$x$\frac{r}{10}$

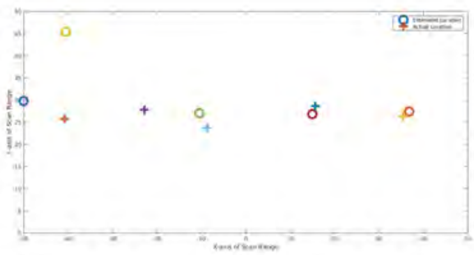
Figure 2.4: Localization results for noise amplitude = 1



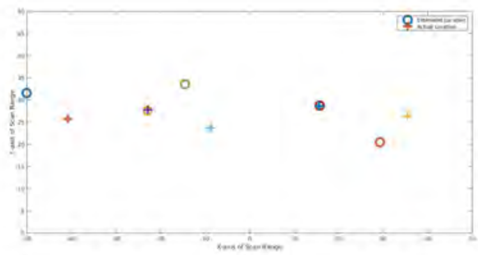
(a) Cell Size: rxr



(b) Cell Size: $\frac{r}{2}$x$\frac{r}{2}$

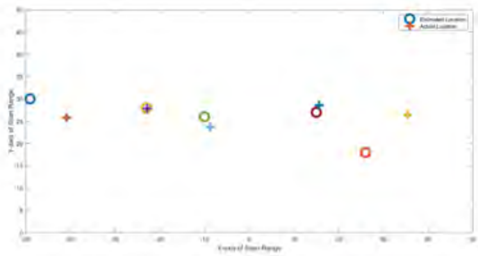


(c) Cell Size: $\frac{r}{5}$x$\frac{r}{5}$

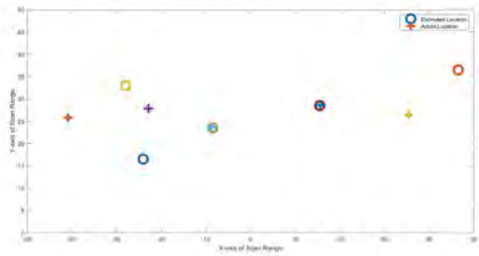


(d) Cell Size: $\frac{r}{10}$x$\frac{r}{10}$

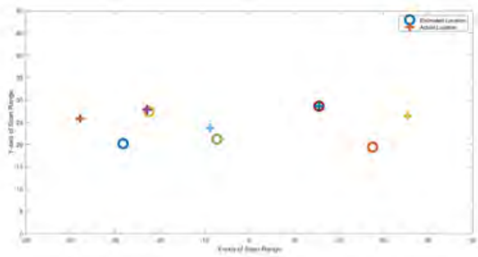
Figure 2.5: Localization results for noise amplitude = 0.5



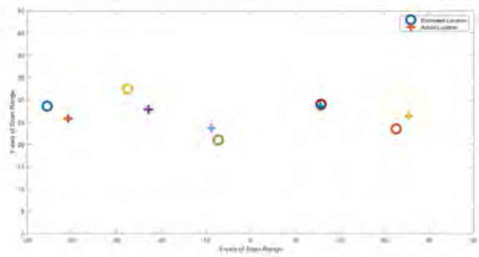
(a) Cell Size: rxr



(b) Cell Size: $\frac{r}{2}$x$\frac{r}{2}$

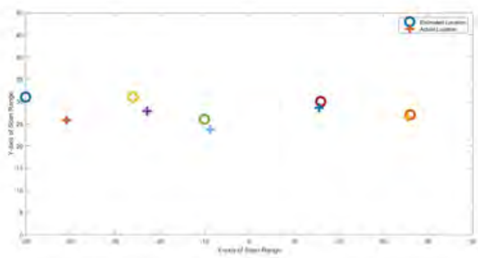


(c) Cell Size: $\frac{r}{5}$x$\frac{r}{5}$

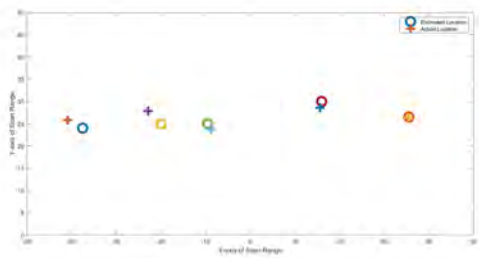


(d) Cell Size: $\frac{r}{10}$x$\frac{r}{10}$

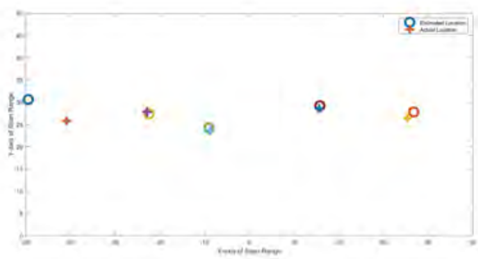
Figure 2.6: Localization results for noise amplitude = 0.25



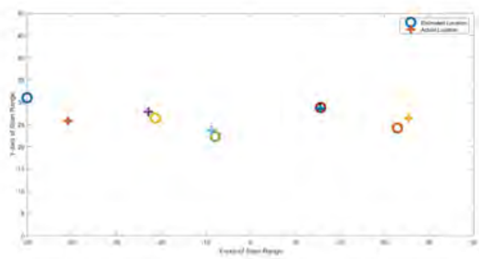
(a) Cell Size: rxr



(b) Cell Size: $\frac{r}{2}$x$\frac{r}{2}$

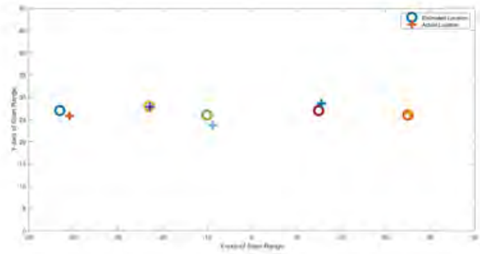


(c) Cell Size: $\frac{r}{5}$x$\frac{r}{5}$

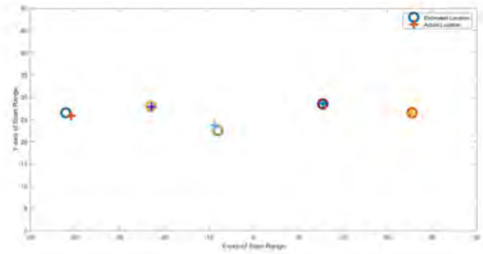


(d) Cell Size: $\frac{r}{10}$x$\frac{r}{10}$

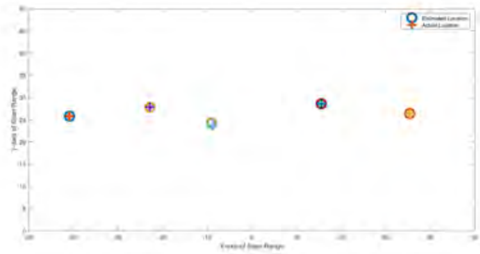
Figure 2.7: Localization results for noise amplitude = 0.1



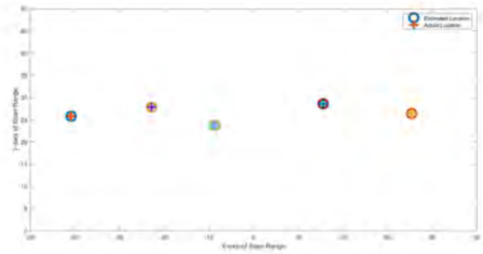
(a) Cell Size: rxr



(b) Cell Size: $\frac{r}{2}$x$\frac{r}{2}$



(c) Cell Size: $\frac{r}{5}$x$\frac{r}{5}$



(d) Cell Size: $\frac{r}{10}$x$\frac{r}{10}$

Figure 2.8: Localization results without noise

Chapter 3

Tracking of a Variable Sound Source using Kalman Filtering

The challenge of evaluating the unknown parameters from one or more observations is known as an estimation. When we use a 'sensor' to collect measurements for a quantity of interest, we run into an estimated difficulty. The estimation phenomenon is critical for integrating the real world as seen by a sensor with human decisions about how to manipulate or impact our surroundings.

The requirement to represent uncertainty lies at the foundation of all estimating challenges. It would be simple to figure out what was going on in reality if we could always take precise measurements of a completely understood process. Unfortunately, this is impossible. Our observations are either imprecise or unreliable. They deal with a procedure that is not fully understood. Furthermore, we rarely view the actual quantity of interest; rather, the variables we observe infer the value of this quantity indirectly.

The estimation algorithm applied in this model is The Kalman filter. In current systems theory, the Kalman filter algorithm is the most commonly used estimating technique, with applications in practically every field of engineering.

R.E. Kalman's renowned paper outlining a recursive solution to the discrete-data linear filtering problem was presented in 1960. The Kalman filter has been the source of much development and practice since that time, owing in substantial part to advancements in digital computing, especially in the field of autonomous or assisted navigation.

The Kalman filter is a set of mathematical equations that provides an efficient computational and recursive solution to the least-squares method. The filter is extremely versatile. It is capable of evaluating the previous, current, as well as future states even if the true nature of the modeled system is unclear at the time.

The figure below illustrates the overall procedure of the Kalman filter. A system model is operated by established controls on the other hand a measurement device is used to supply relevant quantities. The physical system data available for estimation is simply information about these system inputs and outputs.

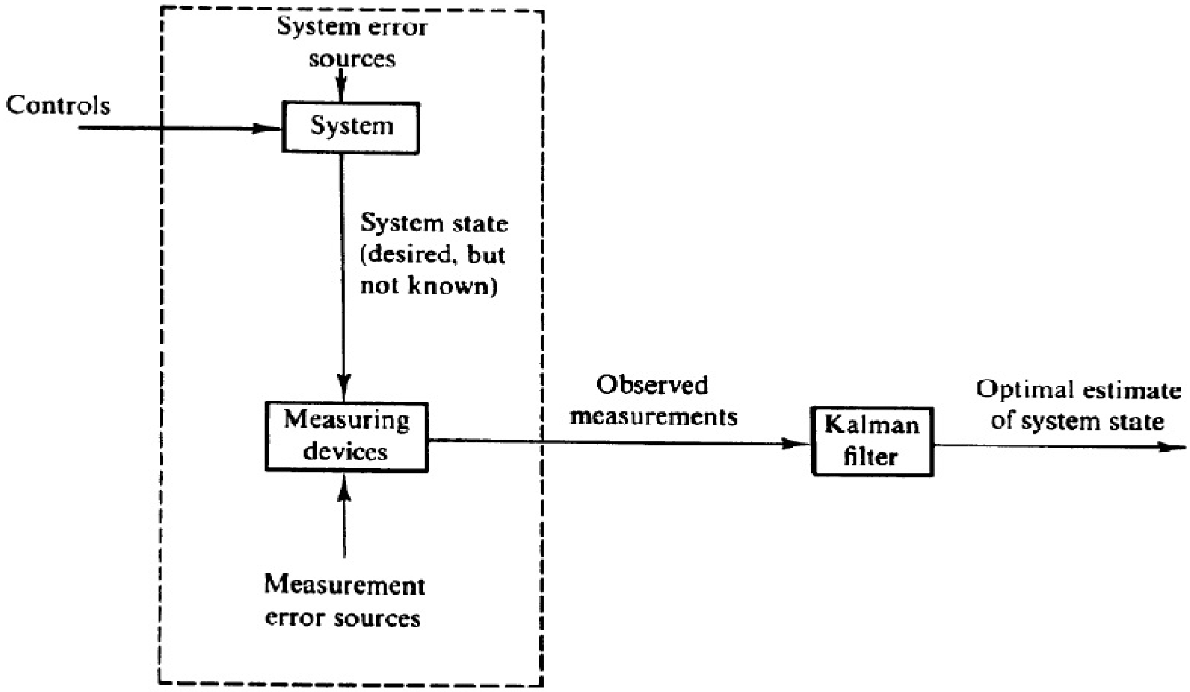


Figure 3.1: Complete Overview of Kalman Filter

3.1 Static Model

A static model is defined as a model which deals with the constant structural analysis and does not change over time. It represents the system's structure, which is less likely to vary with time. The purpose of this model is to describe the fundamentals of the Kalman filter in a straightforward and intuitive manner.

3.1.1 Kalman Filter in One-Dimension

The model of the system is fully described by five equations. These equations cover the past, present, and future aspects of the system state.

- **State Update Equation**

  The first equation is called the state update equation. This equation takes into account the system's current condition and is, given by

  $$\hat{x}_{k,k} = \hat{x}_{k,k-1} + K_k(z_k - \hat{x}_{k,k-1})$$

  where $x_{k,k}$ and $x_{k,k-1}$ is the estimate of the current state and the predicted value of the current state respectively. The Kalman gain and observed value are represented by the factors K_k and z_k, respectively. As demonstrated by the subscript k, both of these variables change with each cycle.

- **State Extrapolation Equation**

The state extrapolation equation, often known as a prediction equation, given by

$$\hat{x}_{k,k+1} = \hat{x}_{k,k}$$

It extrapolates from the current state to the next. The system's dynamic model is deemed constant in this scenario, its state does not vary over time.

- **Kalman Gain**

  The variations between the measurements and the true value are referred to as measurement errors. Because measurement errors are arbitrary, we can describe them using variance. The measurement uncertainty is the variance of the measurement errors and is represented by r.

  The estimated error is the variation between the estimate and the true system state. Although the estimated error is unknown, the degree of uncertainty in the estimate can be calculated. The estimated uncertainty will be represented by the letter p.

  The Kalman gain is the third Kalman filter equation, represented as

$$K_k = \frac{p_{k,k-1}}{p_{k,k-1} + r_k}$$

  It ranges from 0 to 1. In the preceding equation, $p_{k,k-1}$ represents the predicted uncertainty in estimate and r_k represents the uncertainty in measurement.

- **Covariance Update Equation**

  As the name suggests this equation calculates the current uncertainty in the estimate, give by

$$p_{k,k} = (1 - k_k)p_{k,k-1}$$

  where k_k and $p_{k,k-1}$ are respectively the Kalman gain and the predicted uncertainty in estimate.

- **Covariance Extrapolation Equation**

  The covariance extrapolation equation, given by

$$p_{k+1,k} = p_{k,k} + q_k$$

  where q represents the process noise variance. The process noise is a factor added in the covariance extrapolation equation due to the uncertainty of the dynamic model. The dynamic model at this stage is constant so the later state equals the current state.

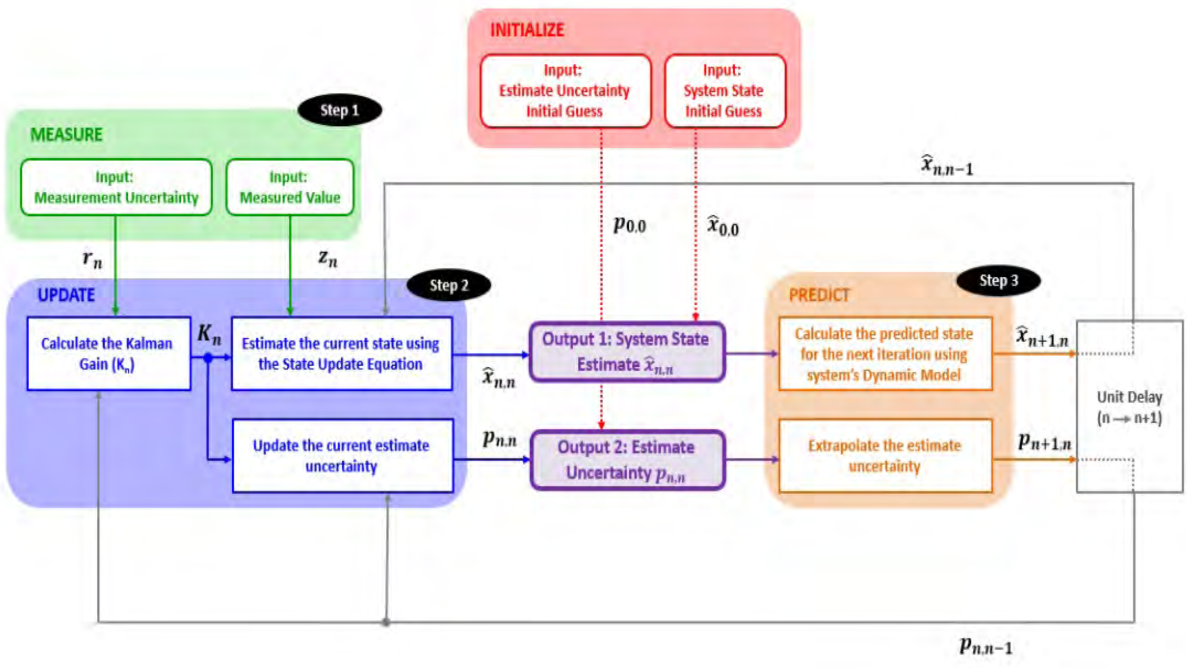


Figure 3.2: Block Diagram of Kalman Filter

3.2 Dynamic Model

Dynamic models can be used to describe many real-world scenarios using mathematical equations or computer algorithms. As the name implies, these models explain system states that change over time.

3.2.1 Multidimensional Kalman Filter

The multidimensional Kalman filter is discussed in the following topic. Most dynamic models have system states with two, three, or even additional dimensions. This section explains the Kalman filter's matrix notation.

- **State Extrapolation Equation**

  The state extrapolation equation, often known as the prediction equation connects the system's current and future states. In a multidimensional setting, the equation is defined as,

$$\hat{\mathbf{x}}_{\mathbf{k+1,k}} = \phi\hat{\mathbf{x}}_{\mathbf{k,k}} + \mathbf{G}\mathbf{u}_{\mathbf{k}} + \mathbf{w}_{\mathbf{k}}$$

  in this equation $\hat{\mathbf{x}}_{\mathbf{k+1,k}}$ and $\hat{\mathbf{x}}_{\mathbf{k,k}}$ represent the extrapolated system state and estimated system state vectors respectively. $\mathbf{u}_{\mathbf{k}}$ is the input or control variable while the matrix $\mathbf{w}_{\mathbf{k}}$ is the process noise. ϕ and $\mathbf{G}$ refers to the state transition matrix and input transition matrix respectively.

  The input variable $\mathbf{u}_{\mathbf{k}}$ in this work is considered zero as there is no external force acting on the system.

- **Covariance Extrapolation Equation**

  In a multidimensional setting, the general description of this equation is as follows

$$\mathbf{\hat{P}_{k+1,k}} = \phi \mathbf{\hat{P}_{k,k}} \phi^T + \mathbf{Q}$$

  where $\mathbf{P_{k,k}}$ and $\mathbf{P_{k+1,k}}$ is covariance matrix and extrapolated covariance matrix respectively. ϕ is state transition matrix and $\mathbf{Q}$ is process noise matrix.As previously stated, process noise has a significant impact on the Kalman filter's functioning. This noise is represented in the multidimensional setting by a covariance matrix $\mathbf{Q}$. The process noise included in this working configuration is dependent and correlated between the state variables.

- **Measurement Equation**

  The measurement value was symbolized by $\mathbf{z_k}$ in the preceding section; it relates to the hidden true system state that is damaged owing to measurement noise $\mathbf{v_k}$. The measurement noise variance, denoted by $\mathbf{r_k}$, might be either constant or variable.

  In matrix form, the measurement equation for the multidimensional scenario is

$$\mathbf{z_k} = \mathbf{H}\mathbf{x_k} + \mathbf{v_k}$$

  where $\mathbf{z_k}$, $\mathbf{x_k}$ and $\mathbf{v_k}$ is the measurement or output, hidden system state and random noise vector respectively. $\mathbf{H}$ corresponds to the observation matrix.The measurement does not correspond to the expected output in certain system models. In that instance, a transformation is required to translate the system state to the measured state. The matrix $\mathbf{H}$ serves as a bridge between these two states.

- **State Update Equation**

  In matrix form, the state update equation for the multidimensional scenario is

$$\mathbf{\hat{x}_{k,k}} = \mathbf{\hat{x}_{k,k\text{-}1}} + \mathbf{K_k}(\mathbf{z_k} - \mathbf{H}\mathbf{\hat{x}_{k,k\text{-}1}})$$

  where $\mathbf{\hat{x}_{k,k}}$, $\mathbf{\hat{x}_{k,k\text{-}1}}$ and $\mathbf{z_k}$ is estimated, extrapolated and measured system state respectively. $\mathbf{K_k}$ is the Kalman gain and $\mathbf{H}$ is the observation matrix.

- **Covariance Update Equation**

  In a multidimensional setting, the general description of this equation is as follows

$$\mathbf{P_{k,k}} = (\mathbf{I} - \mathbf{K_k}\mathbf{H})\mathbf{P_{k,k\text{-}1}}(\mathbf{I} - \mathbf{K_k}\mathbf{H})^T + \mathbf{K_k}\mathbf{R_k}\mathbf{K_k}^T$$

  where $\mathbf{P_{k,k}}$ and $\mathbf{P_{k,k\text{-}1}}$ is the updated covariance matrix and the extrapolated covariance matrix at previous state receptively. $\mathbf{K_k}$ is the kalman gain also $\mathbf{H}$

and $\mathbf{R_k}$ is observation and measurement uncertainty matrix respectively.

- **Kalman Gain**

  The Kalman gain equation is the final of the five equations. It is defined as follows in a multidimensional setting

$$\mathbf{K_k} = \frac{\mathbf{P_{k,k\text{-}1}}\mathbf{H}^T}{\mathbf{H}\mathbf{P_{k,k\text{-}1}}\mathbf{H}^T + \mathbf{R_k}}$$

  where $\mathbf{K_k}$ denotes the Kalman gain. $\mathbf{H}$ and $\mathbf{R_k}$ is observation and measurement uncertainty matrix respectively. $\mathbf{P_{k,k\text{-}1}}$ is the extrapolated covariance matrix at previous state.

3.3 Proposed Models

The purpose of this project is to track a source using the Kalman filter algorithm.The source is free to roam around in the surrounding.Three motion models are appropriate for modeling this scenario: motion along one axis, constant velocity, and constant acceleration. Following are the three proposed models for this thesis.

3.3.1 First Model: Motion Along One Axis

For this model the system state consists of the target position and velocity in y coordinate.

$$\mathbf{x_k} = \begin{bmatrix} \hat{y}_k \\ \hat{\dot{y}}_k \end{bmatrix}$$

The expected source states at n instants can be specified using two equations based on the motion along one axis model.

$$\begin{aligned} \hat{y}_{k+1,k} &= \hat{y}_{k,k} + \hat{\dot{y}}_{k,k}\delta t \\ \hat{\dot{y}}_{k+1,k} &= \hat{\dot{y}}_{k,k} \end{aligned}$$

The extrapolated state vector can be derived from the equation,

$$\mathbf{\hat{x}_{k+1,k}} = \phi \mathbf{\hat{x}_{k,k}} \tag{3.1}$$

In matrix representation,

$$\begin{bmatrix} \hat{y}_{k+1,k} \\ \hat{\dot{y}}_{k+1,k} \end{bmatrix} = \begin{bmatrix} 1 & \delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{y}_k \\ \hat{\dot{y}}_k \end{bmatrix}$$

The state transition matrix thus derived is

$$\phi = \begin{bmatrix} 1 & \delta t \\ 0 & 1 \end{bmatrix}$$

The estimate uncertainty derived from the covariance update equation can be written in the form,

$$\mathbf{P} = \begin{bmatrix} P_y & P_{y\dot{y}} \\ P_{\dot{y}y} & P_{\dot{y}} \end{bmatrix}$$

where P_y and $P_{\dot{y}}$ is the variance of y coordinate position and velocity estimate respectively. $P_{y\dot{y}}$ and $P_{\dot{y}y}$ represent the covariance.

$$\mathbf{P} = \begin{bmatrix} P_y & 0 \\ 0 & P_{\dot{y}} \end{bmatrix}$$

Because the estimation error in position and velocity are believed to be unrelated in this work, the covariance terms are set to zero.

$$\mathbf{P} = \begin{bmatrix} 600 & 0 \\ 0 & 600 \end{bmatrix}$$

The state vector begins with a random guess, which explains why the location and velocity variances are so large.

The process noise matrix is defined in the following manner.The discrete noise model is used to account for process noise. It states that the noise's value varies between time periods but remains constant within a time period. The variance and covariance of location and velocity can be described using random acceleration covariance σ_a^2.The random acceleration covariance in this model is set at 0.25.

$$\mathbf{Q} = \begin{bmatrix} \sigma_y^2 & \sigma_{y\dot{y}}^2 \\ \sigma_{\dot{y}y}^2 & \sigma_{\dot{y}}^2 \end{bmatrix}$$

Now constructing the process noise model for this case,lets consider a matrix

$$\mathbf{Q_b} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \sigma_a^2$$

the process noise matrix Q can be derived from the equation

$$\mathbf{Q} = \phi Q_b \phi^T$$

substituting the values results in the matrix below,

$$\mathbf{Q} = \begin{bmatrix} \delta t^2 & \delta t \\ \delta t & 1 \end{bmatrix} \sigma_a^2$$

The measurement equation is defined below, it provides us with the y coordinate location of the source.

$$\mathbf{z_k} = \mathbf{H}\mathbf{x_k} \tag{3.2}$$

$$\begin{bmatrix} y_{k,measured} \end{bmatrix} = \mathbf{H} \begin{bmatrix} y_k \\ \dot{y}_k \end{bmatrix}$$

So the observation matrix is defined as,

$$\mathbf{H} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

The measurement covariance matrix R_k models the error in y coordinate position measurement.

$$\mathbf{R_k} = \begin{bmatrix} \sigma_{ym}^2 \end{bmatrix}$$

For the sake of this model the measurement uncertainty between each measurement is kept constant.

$$\mathbf{R_k} = \begin{bmatrix} 10 \end{bmatrix}$$

3.3.2 Second Model: Constant Velocity Model

For this model the system state consists of the target position and velocity in x and y coordinate respectively.

$$\mathbf{x_k} = \begin{bmatrix} \hat{x}_k \\ \hat{\dot{x}}_k \\ \hat{y}_k \\ \hat{\dot{y}}_k \end{bmatrix}$$

The expected source states at n instants can be specified using these four equations based on the constant velocity motion model.

$$\begin{aligned} \hat{x}_{k+1,k} &= \hat{x}_{k,k} + \dot{x}_{k,k}\delta t \\ \hat{\dot{x}}_{k+1,k} &= \hat{\dot{x}}_{k,k}\delta t \\ \hat{y}_{k+1,k} &= \hat{y}_{k,k} + \dot{y}_{k,k}\delta t \\ \hat{\dot{y}}_{k+1,k} &= \hat{\dot{y}}_{k,k} \end{aligned}$$

The extrapolated state vector can be derived from the equation,

$$\mathbf{\hat{x}_{k+1}} = \phi \mathbf{\hat{x}_{k,k}} \tag{3.3}$$

In matrix representation,

$$\begin{bmatrix} \hat{x}_{k+1,k} \\ \hat{\dot{x}}_{k+1,k} \\ \hat{y}_{k+1,k} \\ \hat{\dot{y}}_{k+1,k} \end{bmatrix} = \phi \begin{bmatrix} \hat{x}_k \\ \hat{\dot{x}}_k \\ \hat{y}_k \\ \hat{\dot{y}}_k \end{bmatrix}$$

The state transition matrix thus derived is

$$\phi = \begin{bmatrix} 1 & \delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The estimate uncertainty derived from the covariance update equation can be written in the form,

$$\mathbf{P} = \begin{bmatrix} P_x & 0 & 0 & 0 \\ 0 & P_{\dot{x}} & 0 & 0 \\ 0 & 0 & P_y & 0 \\ 0 & 0 & 0 & P_{\dot{y}} \end{bmatrix}$$

where P_y and $P_{\dot{y}}$ is the variance of y coordinate position and velocity estimate respectively. P_x and $P_{\dot{x}}$ is the variance of x coordinate position and velocity estimate respectively. Th covariance are set to zero as the terms are not correlated.

$$\mathbf{P} = \begin{bmatrix} 600 & 0 & 0 & 0 \\ 0 & 600 & 0 & 0 \\ 0 & 0 & 600 & 0 \\ 0 & 0 & 0 & 600 \end{bmatrix}$$

The state vector begins with a random guess, which explains why the location and velocity variances are so large.

The process noise matrix is defined in the following manner.The discrete noise model is used to account for process noise. It states that the noise's value varies between time periods but remains constant within a time period. The variance and covariance of location and velocity can be described using random acceleration covariance σ_a^2. The random acceleration covariance in this model is set at 0.25.

Constructing the process noise model for this case in the same manner as previously stated using the equation,

$$\mathbf{Q} = \phi \mathbf{Q_b} \phi^T$$

where the matrix Q_a is

$$\mathbf{Q_b} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \sigma_a^2$$

so the final process noise matrix is,

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \delta t^2 & \delta t \\ 0 & 0 & \delta t & 1 \end{bmatrix} \sigma_a^2$$

The measurement equation is defined below, it provides us with the x and y coordinate location of the source.

$$\mathbf{z_k} = \mathbf{H}\mathbf{x_k} \tag{3.4}$$

from the above equation the observation matrix H derived is mentioned below, where $z_k = \begin{bmatrix} x_{k,measured} \\ y_{k,measured} \end{bmatrix}$

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

The measurement covariance matrix R_n models the error in y and x coordinate position measurements.

$$\mathbf{R} = \begin{bmatrix} \sigma_{xm}^2 & 0 \\ 0 & \sigma_{ym}^2 \end{bmatrix}$$

For the sake of this model the measurement uncertainty between each measurement is kept constant.

$$\mathbf{R} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$$

3.3.3 Third Model: Constant Acceleration Model

For this model the system state consists of the source position, velocity and acceleration in x and y coordinates respectively.

$$\mathbf{x_k} = \begin{bmatrix} \hat{x}_k \\ \hat{\dot{x}}_k \\ \hat{\ddot{x}}_k \\ \hat{y}_k \\ \hat{\dot{y}}_k \\ \hat{\ddot{y}}_k \end{bmatrix}$$

The expected source states at n instants can be specified using these nine equations based on the constant velocity motion model.

$$\hat{x}_{k+1,k} = \hat{x}_{k,k} + \dot{x}_{k,k}\delta t + \frac{1}{2}\ddot{x}_{k,k}\delta t^2$$
$$\hat{\dot{x}}_{k+1,k} = \hat{\dot{x}}_{k,k} + \ddot{x}_{k,k}\delta t$$
$$\hat{\ddot{x}}_{k+1,k} = \ddot{x}_{k,k}$$
$$\hat{y}_{k+1,k} = \hat{y}_{k,k} + \dot{y}_{k,k}\delta t + \frac{1}{2}\ddot{y}_{k,k}\delta t^2$$
$$\hat{\dot{y}}_{k+1,k} = \hat{\dot{y}}_{k,k} + \ddot{y}_{k,k}\delta t$$
$$\hat{\ddot{y}}_{k+1,k} = \ddot{y}_{k,k}$$

The extrapolated state vector can be derived from the equation,

$$\mathbf{\hat{x}_{k+1,k}} = \phi\mathbf{\hat{x}_{k,k}} \tag{3.5}$$

The state transition matrix thus derived is

$$\phi = \begin{bmatrix} 1 & \delta t & 0.5\delta t^2 & 0 & 0 & 0 \\ 0 & 1 & \delta t & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \delta t & 0.5\delta t^2 \\ 0 & 0 & 0 & 0 & 1 & \delta t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The estimate uncertainty derived from the covariance update equation can be written in the form,

$$\mathbf{P} = \begin{bmatrix} P_x & P_{x,\dot{x}} & P_{x,\ddot{x}} & 0 & 0 & 0 \\ P_{\dot{x},x} & P_{\dot{x}} & P_{\dot{x},\ddot{x}} & 0 & 0 & 0 \\ P_{\ddot{x},x} & P_{\ddot{x},\dot{x}} & P_{\ddot{x}} & 0 & 0 & 0 \\ 0 & 0 & 0 & P_y & P_{y,\dot{y}} & P_{y,\ddot{y}} \\ 0 & 0 & 0 & P_{\dot{y},y} & P_{\dot{y}} & P_{\dot{y},\ddot{y}} \\ 0 & 0 & 0 & P_{\ddot{y},y} & P_{\ddot{y},\dot{y}} & P_{\ddot{y}} \end{bmatrix}$$

where P_y and $P_{\dot{y}}$ is the variance of y coordinate position and velocity estimate respectively. P_x and $P_{\dot{x}}$ is the variance of x coordinate position and velocity estimate respectively.The off diagonal entries are covariance.They are set to zero as the terms are not correlated.

$$\mathbf{P} = \begin{bmatrix} 600 & 0 & 0 & 0 & 0 & 0 \\ 0 & 600 & 0 & 0 & 0 & 0 \\ 0 & 0 & 600 & 0 & 0 & 0 \\ 0 & 0 & 0 & 600 & 0 & 0 \\ 0 & 0 & 0 & 0 & 600 & 0 \\ 0 & 0 & 0 & 0 & 0 & 600 \end{bmatrix}$$

The state vector begins with a random guess, which explains why the location and velocity variances are so large.

The process noise matrix is defined in the following manner.The discrete noise model is used to account for process noise. It states that the noise's value varies between time periods but remains constant within a time period. The variance and covariance of location and velocity can be described using random acceleration covariance σ_a^2.The random acceleration covariance in this model is set at 0.25.

$$\mathbf{Q} = \begin{bmatrix} \frac{\delta t^4}{4} & \frac{\delta t^3}{2} & \frac{\delta t^2}{2} & 0 & 0 & 0 \\ \frac{\delta t^3}{2} & \delta t^2 & \delta t & 0 & 0 & 0 \\ \frac{\delta t^2}{2} & \delta t & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\delta t^4}{4} & \frac{\delta t^3}{2} & \frac{\delta t^2}{2} \\ 0 & 0 & 0 & \frac{\delta t^3}{2} & \delta t^2 & \delta t \\ 0 & 0 & 0 & \frac{\delta t^2}{2} & \delta t & 1 \end{bmatrix} \sigma_a^2$$

The measurement equation is defined below, it provides us with the x and y coordinate location of the source.

$$\mathbf{z_k} = \mathbf{H}\mathbf{x_k} \tag{3.6}$$

from the above equation the observation matrix H derived is mentioned below, where $z_n = \begin{bmatrix} x_{k,measured} \\ y_{k,measured} \end{bmatrix}$

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

The measurement covariance matrix R_k models the error in y and x coordinate position measurements.

$$\mathbf{R} = \begin{bmatrix} \sigma_{xm}^2 & 0 \\ 0 & \sigma_{ym}^2 \end{bmatrix}$$

For the sake of this model the measurement uncertainty between each measurement is kept constant.

$$\mathbf{R} = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$$

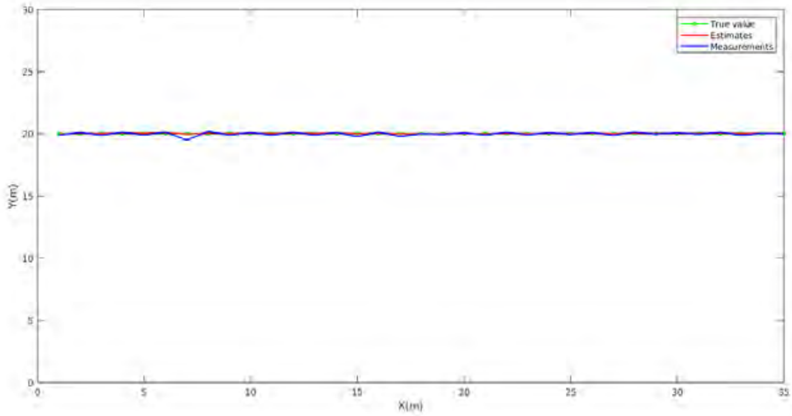
3.4 Simulation Results

The simulation was carried out using MATLAB to demonstrate the effectiveness of the speaker tracking algorithm. To achieve better results 35 iterations were performed to estimate the trajectory of sound source.

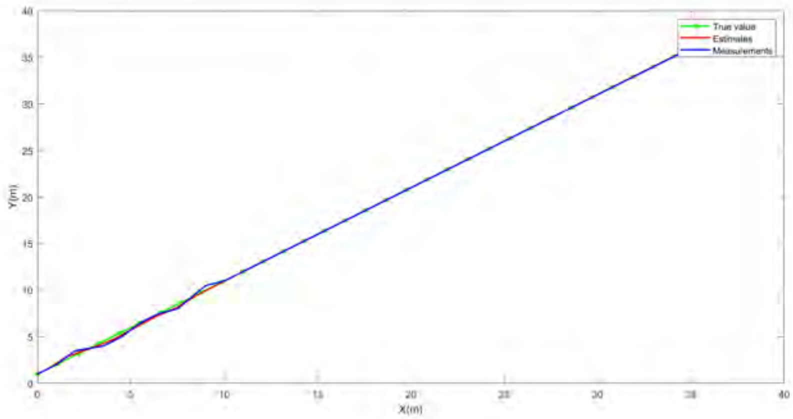
The results of tracking a sound source moving along a single Cartesian coordinate are shown in Figure 3.3(a). The true values are fed into the first algorithm, to model the source movement. The measurement values are obtained from the sound source localization algorithm and the Kalman filtering algorithm is used to smooth those location values. The results indicate that the Kalman algorithm estimates are smooth and perfectly track the trajectory of source.

Figure 3.3(b) depicts the Kalman filter estimates for modeling the constant velocity motion of a sound source along with the two Cartesian coordinates x and y. As previously stated, the true values are the input to the first algorithm, and the outputs are referred to as measurement values.In this model, the Kalman filtering algorithm is indeed effective. The estimates are perfectly consistent with the source's trajectory.

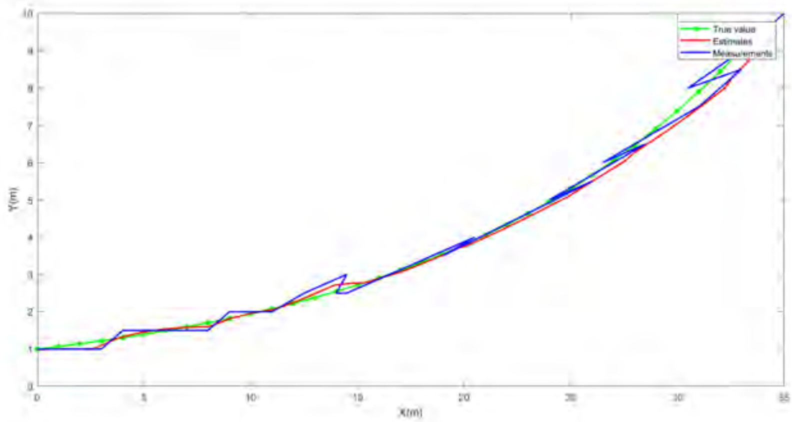
The final model proposed is the motion of the source with constant acceleration along with the Cartesian coordinates x and y. Simulation rsults for this model are depicted in Figure 3.3(c) where True values represent the model of the source trajectory, while measurement values represent the output of the localization algorithm. The measurement values in this model are rather irregular and noisy, however, the Kalman filter algorithm estimates are even and perfectly track the trajectory of the source.



(a) First Model: Motion along Single Axes



(b) Second Model: Motion with Constant Velocity



(c) Third Model: Motion with Constant Acceleration

Figure 3.3: Simulation Results for Tracking of Moving Sound Source

Chapter 4

Conclusion

The theory of microphone beamforming and Kalman filtering has been assessed in this thesis for sound source localization and tracking. The main goal is to distinguish the noise signal from the source signal so that localization is done expeditiously. Aiming that, a power maximization algorithm is used. It assists to extract the source from the noise based on the power amplitude. Conventional Beamforming has been reviewed for the location estimation. It has some restrictions because of its beam wideness and they can be handled by enlarging the number of array elements. But doing so requires a lot of cost and effort. So methods to work with or remove some of the limitations have been observed throughout.

The proposed system in this study is the multidimensional system for sound source localization and tracking. The performance of the localization algorithm was evaluated in both the absence and presence of noise. The technique uses noise values of 0.1, 0.25, 0.5 and 1. The search grid size was lowered to counteract the influence of noise and make the localization mechanism more robust. Localization estimations with a smaller search grid size have proven to be more efficient.

The Kalman filter algorithm is applied to improve the estimates and track a moving sound source. To completely characterize a sound source in a multidimensional environment, a model is required that takes into account all the aspects. To model the arbitrary mobility of the sound source, three approaches are utilized. Motion in a single coordinate, motion in two coordinates with constant velocity, and motion in two coordinates with constant acceleration are the proposed models. In the suggested model, the coordinate corresponding to height is kept constant.

Kalman filtering has been utilized to smooth out the trajectory of the speaker's movement through its estimation procedure. In conventional beamforming, the delay and sum technique has been debated over. The efficiency and restraints of these algorithms have been observed through MATLAB simulations. Using these techniques, the localization and tracking have been done efficiently and better SNR is achieved.

Appendix A

MATLAB Code for Sound Source Localization

```
clc
clearvars -except sourceXCoord sourceYCoord
close all;

%% Constants
global noOfMics d xRef yRef spSnd freqSrc fs T t L w wvNum micXCoord micYCoord

noOfMics = 10;
d = 1;                  %distance b/w sensors
xRef = 0;               %position of refrence microphone
yRef = 0;
spSnd = 343;            %speed of sound
freqSrc = 400;          %frequency of source signal
fs = 9000;              %sampling frequency
T = 0.1;                %sampling period
t = 0:1/fs:T;
L= length(t);
w = 2*pi*freqSrc;
wvNum = w/spSnd;        %wave number

%% Setting up the microphone array

micXCoord = 1:d:noOfMics;
micXCoord = micXCoord';
micYCoord = zeros(noOfMics,1);
%% Scan Range
stRangeX = -10;
stRangeY = 0;
stepSize = 0.5;
endRangeX = 50;
endRangeY = 50;
```

```
function [column,row] = beamForming(sourceXCoord,sourceYCoord)
defineConstants

sourceSig = sin(2*w*t);

% sourceXCoord = input('Please input source X Coordinate: ') ;
% sourceYCoord = input('Please input source Y Coordinate: ') ;

[distFromSource2RefMic,distFromSource2MicArray] = calcSource2MicDist(sourceXCoord,sourceYCoord);

receivedSignals = returnSignalsReceivedByMicArray(sourceSig,distFromSource2RefMic,distFromSource2MicArray);

receivedSignals = addNoise(receivedSignals);

scanRangeXCoord = stRangeX:stepSize:endRangeX;%linspace(1,noOfMics,noOfMics);
scanRangeYCoord = stRangeY:stepSize:endRangeY;%linspace(1,noOfMics,noOfMics);
outputPower = computePower(receivedSignals, scanRangeXCoord,scanRangeYCoord);

maxOpPower = max(max(outputPower));
[row, column] = find(outputPower == maxOpPower);
%outputPower'

row = (row.*stepSize) + stRangeY -stepSize;
column = (column.*stepSize) + stRangeX -stepSize;

sprintf("Source is located at (%0.2f,%0.2f)",column,row)

end

function [distFromSource2RefMic,distFromSource2MicArray] = calcSource2MicDist(sourceXCoord,sourceYCoord)
    global xRef yRef noOfMics micXCoord micYCoord

    distFromSource2RefMic = sqrt((sourceXCoord - xRef).^2 + (sourceYCoord - yRef).^2);
    for k= 1:noOfMics
        distFromSource2MicArray(k) = sqrt((sourceXCoord - micXCoord(k)).^2 + (sourceYCoord - micYCoord(k)).^2);
    end
end

function receivedSignals = returnSignalsReceivedByMicArray(sourceSig,distFromSource2RefMic,distFromSource2MicArray)
```

```
    global noOfMics wvNum

    A = distFromSource2RefMic./distFromSource2MicArray;          % amplitude of received signal
    dist_wr_refmic = distFromSource2MicArray - distFromSource2RefMic;
    for k = 1:noOfMics
        SteeringVect(k) = cos(wvNum*dist_wr_refmic(k))-1i*sin(wvNum*dist_wr_refmic(k));
        receivedSignals(k,:) = A(k)*sourceSig.*SteeringVect(k);
    end
end

function receivedSignals = addNoise(receivedSignals)
    global noOfMics L

    Am = 10^(-1);
    n1 = Am * (randn(noOfMics, L) + j*randn(noOfMics, L));
    receivedSignals = receivedSignals + n1;
    % %% Adding noise to the signal
    % Am = 10^(-1);
    % n1 = Am * (randn(noOfMics, L) + j*randn(noOfMics, L));
    % mic_in_sig = received_sig + n1;
end

function outputPower = computePower(receivedSignals, scanRangeXCoord, scanRangeYCoord)
    global noOfMics micXCoord micYCoord xRef yRef wvNum

    Source2AnyMicDist = nan(length(scanRangeXCoord),length(scanRangeYCoord));
    Source2RefMicDist = nan(length(scanRangeXCoord),length(scanRangeYCoord));
    covar_matrix = (receivedSignals*receivedSignals')/length(receivedSignals);

    for p = 1:length(scanRangeYCoord)
        for q = 1:length(scanRangeXCoord)
            Source2AnyMicDist = sqrt((scanRangeXCoord(q)-micXCoord).^2 + (scanRangeYCoord(p) - micYCoord).^2);
            Source2RefMicDist = sqrt((scanRangeXCoord(q)-xRef).^2 + (scanRangeYCoord(p) - yRef).^2);
            dist_wr_refmic_ = Source2AnyMicDist - Source2RefMicDist;
            weightVec = cos(wvNum*dist_wr_refmic_)-1i*sin(wvNum*dist_wr_refmic_);
            outputPower(p,q) = abs(weightVec'*covar_matrix*weightVec);
        end
    end

```

```
    for k1 = 1:length(scanRangeYCoord)
        maxP(k1) = max(outputPower(k1,:));
    end
    outputPower = outputPower/max(maxP);
end
```

Appendix B

MATLAB Code for Sound Source Tracking

```
clear all
clc
close all

x = 0:1:35-1;
linearLocations = x-min(x)+1;
constantLocations = 20*ones(1,length(x));
exponentialLocations = exp(x./15);

y = linearLocations; % Select Which Type of Input to Run

xArrOut = [];yArrOut = [];
for k = 1:length(x)
    [xOut,yOut] = beamForming(x(k),y(k));
    xArrOut = [xArrOut, xOut];
    yArrOut = [yArrOut, yOut];
end

trueValues = [x;y];
measurements = [xArrOut;yArrOut];

maximumError = max(max(abs(measurements - trueValues)))

plot(x,y,'.-')
hold on
plot(xArrOut,yArrOut,'.-')
legend('Input Values','Output Values')
```

References

Baig, N. A. and Malik, M. B. (2013). Comparison of direction of arrival (doa) estimation techniques for closely spaced targets. *International journal of future computer and communication*, 2(6):654.

Bechler, D., Grimm, M., and Kroschel, K. (2003). Speaker tracking with a microphone array using kalman filtering.

Benesty, J., Chen, J., and Huang, Y. (2008). *Microphone array signal processing*, volume 1. Springer Science & Business Media.

Billingsley, J. and Kinns, R. (1976). The acoustic telescope. *Journal of Sound and Vibration*, 48(4):485–510.

Brandstein, M. (2001). *Microphone arrays: signal processing techniques and applications*. Springer Science & Business Media.

Brookner, E. (1998). Tracking and kalman filtering made easy.

Durrant-Whyte, H. et al. (2001). Introduction to estimation and the kalman filter. *Australian Centre for Field Robotics*, 28(3):65–94.

Gruber, M. H. (1997). Statistical digital signal processing and modeling.

Jung, K. K., Shin, H. S., Kang, S. H., and Eom, K. H. (2007). Object tracking for security monitoring system using microphone array. In *2007 International Conference on Control, Automation and Systems*, pages 2351–2354. IEEE.

Krim, H. and Viberg, M. (1996). Two decades of array signal processing research: the parametric approach. *IEEE signal processing magazine*, 13(4):67–94.

Lazarus, K., Noordin, N. H., and Ibrahim, Z. (2018). An enhanced simulated kalman filter algorithm and its application in adaptive beamforming. In *2018 IEEE International RF and Microwave Conference (RFM)*, pages 321–324. IEEE.

Maybeck, P. S. (1990). The kalman filter: An introduction to concepts. In *Autonomous robot vehicles*, pages 194–204. Springer.

McDonough, J., Kumatani, K., Arakawa, T., Yamamoto, K., and Raj, B. (2013). Speaker tracking with spherical microphone arrays. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3981–3985. IEEE.

Woo-han, Y., Cheon-in, O., Kyu-Dae, B., and Su-young, J. (2006). The impulse sound source tracking using kalman filter and the cross-correlation. In *2006 SICE-ICASE International Joint Conference*, pages 317–320. IEEE.