A Low-Cost Acoustic Array for Detecting and Tracking Multiple Acoustic Targets

Ellen E. Case
Wright State University

Follow this and additional works at: https://corescholar.libraries.wright.edu/etd_all
Part of the Electrical and Computer Engineering Commons

Repository Citation
https://corescholar.libraries.wright.edu/etd_all/894

This Thesis is brought to you for free and open access by the Theses and Dissertations at CORE Scholar. It has been accepted for inclusion in Browse all Theses and Dissertations by an authorized administrator of CORE Scholar. For more information, please contact corescholar@www.libraries.wright.edu, library-corescholar@wright.edu.
A LOW-COST ACOUSTIC ARRAY FOR DETECTING AND TRACKING MULTIPLE ACOUSTIC TARGETS

A thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

by

ELLEN E. CASE
B.S. Electrical Engineering, 2007
Wright State University

2008
Wright State University
I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Ellen E. Case ENTITLED A Low-Cost Acoustic Array for Detecting and Tracking Multiple Acoustic Sources BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science in Engineering.

Brian D. Rigling, Ph.D.  
Thesis Director

Kefu Xue, Ph.D.  
Department Chair

Committee on Final Examination

Brian D. Rigling, Ph.D.

Fred Garber, Ph.D.

Michael Bryant, Ph.D.

Joseph F. Thomas, Jr., Ph.D.  
Dean, School of Graduate Studies
ABSTRACT


A wide variety of real-world targets emit distinct acoustic signatures that not only distinguish them from one another but also provide spectral separation from background clutter. While acoustic signatures are distinct, they can also be highly variable, even for an individual target, making detection and tracking a challenging problem. This thesis presents a low-cost acoustic array using commercial off-the shelf (COTS) hardware to detect and track multiple small, moving, acoustic targets. It implements a cross-correlation method for calibration, delay-and-sum beamforming, CA-CFAR detection, a discrete Kalman filter for tracking, and nearest neighbor data association. In addition, the array is designed to be adaptable in use, mobile, and reproducible.
CONTENTS

1 Introduction 1
  1.1 Background .................................................. 1
  1.2 Motivation ................................................... 2
  1.3 Outline ....................................................... 3

2 System Design 4
  2.1 Hardware ...................................................... 6
  2.2 Array Configuration ........................................... 7
  2.3 Calibration .................................................... 9

3 Algorithms 12
  3.1 Preconditioning ............................................... 12
  3.2 Beamforming .................................................. 12
  3.3 Detection ..................................................... 17
  3.4 Tracking ..................................................... 22
    3.4.1 Data Association ......................................... 24

4 Results 29

5 Conclusion 36
  References ..................................................... 37
LIST OF FIGURES

2.1 Acoustic array system design. ................................. 6
2.2 Microphone Design. ......................................... 7
2.3 Array configuration. ......................................... 8
2.4 Cross-correlation of calibration chirps. ....................... 10

3.1 Single microphone data before filtering. ....................... 13
3.2 Single microphone data after filtering. ....................... 13
3.3 Beamforming result: straight flight path. ..................... 14
3.4 Beamforming result: zigzag flight path in the field of view. 15
3.5 Beamforming result: zigzag flight path that crosses out of the field of view. ................................. 15
3.6 Beamforming result: circular flight path. ..................... 16
3.7 Beamforming anomalies. ...................................... 16
3.8 CA-CFAR results: waterfall plot. ............................ 19
3.9 CA-CFAR results: detections. ............................... 19
3.10 CA-CFAR local maxima results: CA-CFAR detections. ......... 20
3.11 CA-CFAR local maxima results: final detections. .......... 21
3.12 Tracking results: CA-CFAR local maxima. .................. 27
3.13 Tracking results: tracked target path. ..................... 28

4.1 Waterfall plots: circular and straight flight paths. .......... 29
4.2 Waterfall plots: circular and zigzag flight paths. .......... 30
4.3 CA-CFAR detections: circular and straight flight paths. .... 30
4.4 CA-CFAR detections: circular and zigzag flight paths. . . . . . . . . 31
4.5 Local maxima CA-CFAR detections: circular and straight flight paths. 31
4.6 Local maxima CA-CFAR detections: circular and zigzag flight paths. 32
4.7 Tracked target paths: circular and straight flight paths. . . . . . . . 33
4.8 Tracked target paths: circular and zigzag flight paths. . . . . . . . . 34
ACKNOWLEDGEMENT

I would like to thank:

* my advisor, Dr. Brian Rigling
  for his advice and guidance
  *
Wright Patterson Air Force
  for funding my graduate education and research
  *
my parents, Susan and Jeff Wehmeyer
  for their love and support
  *
1.1 Background

A sensor array consists of multiple transducers (seismometers, radar antennas, microphones, etc.), or sensors, that provide more enhanced location performance than a single sensor. The goal of the array is to detect, locate, identify, and track one or more objects by identifying points of concentrated energy in the field of view of the array [13]. This is accomplished using a spatial extension of the matched filter (a beamformer) to enhance signals impinging on the array from a desired direction, and to reduce signals impinging on the array from other directions. Many methods are used to accomplish this. For narrowband signals, maximum likelihood (ML), Capon, narrow-band MUSIC, and minimum norm are commonly used to estimate source location, while least-squares and wideband MUSIC methods are more commonly used for wideband signals [6].

Microphone arrays are a specific class of sensor arrays that are used to enhance audio signals [6]. They have been used for both military applications (such as perimeter security and intruder detection), and commercial applications (such as automatic video steering) [10], and can be variable in size, shape, and purpose. Acoustic arrays can be square, circular, linear, or randomly distributed (among other geometries), while some acoustic arrays are set up to have smaller, subarrays [7]. No matter the geometry of the array, acoustic arrays use beamforming, which exploits relative-time-delays-of-arrival (TDOAs) between sensors, to determine source location. Beam-
forming that accomplishes this without knowledge of the sensor locations is known as blind beamforming [5]. For all other beamformers, the locations of the sensors must be known prior to beamforming. This is accomplished with either fixed sensors or a calibration procedure [6].

In addition to source localization, an acoustic array may also be used for tracking, which is most commonly performed using a discrete Kalman filter [4]. For multiple targets with similar acoustic signatures, target-source-location association can be difficult. A method of source separation is presented in [7] which is based on a concept derived from the Szegö theorem. For the autocorrelation matrix of a signal, $R_N$, the eigenvalues of $R_N$ approach the power spectral density (PSD) of the signal as $N$ approaches infinity. For $M$ microphones, $M$ eigenvectors of $R_N$ associated with the $M$ largest eigenvalues are used and divided into pieces where each piece represents a sample. The eigenvectors in the case of multiple sources are not identical as they are each associated with a single source. To separate the sources, cross-correlation is performed across the eigenvector samples [7]. Data association may also be determined using a multiple hypothesis tracker, which poses a new hypothesis for every target-source-location pair. Incorrect hypotheses are culled by determining the likelihood of each possible track [11].

1.2 Motivation

A wide variety of real-world targets emit distinct acoustic signatures that not only distinguish them from one another but also provide spectral separation from background clutter. Examples include ground and air vehicles as well as a broad class of man made signatures that possess broad spectral content, particularly at high frequencies, that are not naturally occurring. While acoustic signatures are distinct, they can also be highly variable, even for an individual target, making detection and tracking a challenging problem [2].
The goal of the project outlined in this thesis is to design a low-cost acoustic array using commercial off-the-shelf (COTS) hardware to detect and track multiple small, moving, acoustic targets. This design should be easy to implement, both from a hardware and software perspective, and be easy to use. As a real-world application, the array should successfully detect and track small radio controlled aircraft using a cross-correlation method for calibration, delay-and-sum beamforming, CA-CFAR detection, a discrete Kalman filter for tracking, and nearest neighbor data association. In addition, the array should be adaptable in use, mobile, and reproducible.

1.3 Outline

The remainder of this thesis is outlined as follows. The system design of the array is described in Chapter 2 and is followed by a description of the algorithms used for processing the signals collected by the array in Chapter 3. Section 3.1 describes the pre-conditioning of the signals prior to beamforming, which is described in Section 3.2. Section 3.3 describes the method of detection used for this array. Section 3.4 describes the tracking algorithm including data association. This thesis concludes with results in Chapter 4 and a conclusion in Chapter 5.
CHAPTER 2
SYSTEM DESIGN

The physical parameters of an acoustic array and the algorithms it uses for calibration and beamforming are dependent on the parameters of the target source. The more variable the source, the more adaptive the array must be.

To begin with, an acoustic source is considered to be either wideband or narrowband. This is determined by the ratio of the highest frequency to the lowest frequency of the signal, where high ratios represent wideband, and ratios near unity represent narrowband. Narrowband signals, because they have a near-constant wavelength, require less complicated algorithms than wideband signals. Phase differences are used to locate narrowband sources, while time delays are used to locate wideband sources [6].

The distance from the source to the sensors also plays a role in the make-up of the array. Sound propagates outward from a source in a circular pattern, so that the wave front that is received by the array is curved. The amount of curvature depends on the distance the sound has traveled. When the distance from the source to the array is such that the wave front received by the array is approximately planar, the source is considered to be in the far-field. Otherwise, it is considered to be in the near-field. For a near-field source, array processing can be used to determine the location of the source. For a far-field source, however, only the direction of arrival (DOA) of the source relative to the array can be estimated.

To prevent spatial aliasing, the maximum element spacing, $d$ for an $m$-element uniformly spaced, linear array falls under the Nyquist constraint:
where $\lambda$ is the wavelength. This stems from the Nyquist theorem that states that the sampling frequency, $f_0$, must be at least twice the carrier frequency, $f_c$. In order for the spatial frequency,

$$f_s = f_c \frac{d \sin \theta}{c} = \frac{d \sin \theta}{\lambda},$$

(2.2)

to satisfy the Nyquist theorem,

$$|f_s| < \frac{1}{2} \iff d|\sin \theta| < \frac{\lambda}{2},$$

(2.3)

which lends to constraint on $d$ [13].

In addition to frequency range and source distance, the reverberation of the source environment also has an impact on the design of an acoustic array. For sources located in wide-open, outdoor fields, reverberation is minimal. For indoor rooms, or outdoor spaces with reflective surfaces, however, 10% to 90% of sound energy may be reflected, depending on the surfaces of the environment [6]. An acoustic array must then determine the strongest acoustic energy in the room and filter out any reverberations.

This thesis discusses the design of an acoustic array for detecting and tracking multiple small, moving acoustic targets. The acoustic targets are assumed to be wideband and to be located in the far-field of the array. Each source is assumed to emit unknown broadband signal that will vary from source to source. The environment is assumed to have little to no reflective surfaces.

The array discussed in this thesis is designed to be both low-cost and mobile while retaining the ability to detect and track multiple small targets. In order for the acoustic array to accommodate various signals without spatial aliasing, it is also designed
to have dynamic element positions, thus requiring a calibration procedure. It consists of 24 microphones, where the locations of the microphones are initially unknown, and uses a delay-and-sum beamforming algorithm. Post-beamforming signal processing consists of target detection, CFAR thresholding, and a Kalman filter with nearest neighbor association for tracking [2].

2.1 Hardware

![Acoustic array system design diagram](image)

Figure 2.1: Acoustic array system design.

The system architecture used for this project can be seen in Figure 2.1 and consists of twenty-four custom-made microphones, a COTS sound-recording board, a desktop Macintosh G4, and a video camera used as a visual reference.

As shown in Figure 2.2, the microphone design comprises a condenser microphone cartridge and a MOSFET for pre-amplification. Custom building the microphones allows a high-quality Panasonic™ microphone cartridge and amplifier circuit to be assembled at a fraction of the cost of a commercial microphone. The microphone cartridge and pre-amp are enclosed in a Fisherman’s Friend throat lozenge box (similar
to an Altoid case) and powered by a 9-volt battery. Ports are drilled in the metal casing for the microphone cartridge, a power switch, and a twenty-foot output cable with a quarter-inch jack. Each microphone cost $14.50 to construct.

To interface between the microphones and the computer, a sound recording board by Mark of the Unicorn (MOTU) is used. This sound recording board has 24 I/O channels with a sampling frequency of 44.1kHz and a maximum bandwidth of 96kHz per channel. A basic IR camera is used as a point of reference for calibration. All signal processing is performed using MATLAB. Acoustic data from each microphone is saved as a separate audio file resulting in up to 24 audio channels per trial. With costs being $1,420 for the MOTU board, $2,000 for a refurbished Macintosh G4, and $348 for 24 microphones; the total cost of an acoustic array of this design is $3,768 [3].

2.2 Array Configuration

The above described hardware supports an array of up to $M = 24$ microphones, though different numbers of microphones were used in subsequent experiments. For the RC aircraft data collection, the microphones are placed in a nominally straight line on the ground with a spacing of approximately 10 inches between elements. The
2-dimensional locations of the microphones \((x_m, y_m)\) for \(m = 1, 2, \ldots, M\) are assumed to be unknown.

Audio data collected by each microphone is represented as

\[
x_m(t) = s(t - t_m) + n_m(t) \quad m = 1, 2, \ldots, M, \tag{2.4}
\]

where \(s(t)\) is the target audio signal and \(n_m(t)\) is unknown noise and clutter experienced by microphone \(m\). The time delay from the target to the \(m\)th microphone is \(t_m\). A target in the far-field at bearing \(\phi\) has time delay \(t_m = \frac{1}{v}(x_m \cos \phi + y_m \sin \phi)\), where \(v\) is the speed of propagation and \(\phi = 0^\circ\) is broadside to the array [3]. The microphone locations relative to \(\phi\) can be seen in Figure 2.3.
2.3 Calibration

When the microphones are in place, broadband calibration signals, in this case chirps generated in MATLAB, are emitted periodically within the field of regard of the array. Any number of chirps can be used, but more chirps result in more accurate calibration. All chirps are generated within one data audio recording so that the calibration data consists of $M$ audio tracks with $C$ chirps per audio track.

Calibration is needed to determine the $x, y$-locations of the microphones relative to the target space. This is done by using the $M \times C$ time delay of arrival (TDOA) matrix associated with the microphones and corresponding chirps to interpolate the microphone locations. This matrix consists of time delays, $t_{m,c}$, representing the number of samples (at 44.1kHz) that occur between when chirp $c$ occurs and when the same chirp reaches microphone $m$ [3].

\[
TDOA = \begin{bmatrix}
  t_{1,1} & t_{1,2} & \ldots & t_{1,M} \\
  t_{2,1} & t_{2,2} & \ldots & t_{2,M} \\
  \vdots & \vdots & \ddots & \vdots \\
  t_{C,1} & t_{C,2} & \ldots & t_{C,M}
\end{bmatrix}
\] (2.5)

The TDOA matrix is found via the cross-correlation of each microphone channel, $x_m(t)$, with the MATLAB generated reference chirp, $r(t)$. Because the chirp is created in MATLAB, it can be exactly replicated digitally. The samples at which the recorded channels are most highly correlated with the reference signal indicate the relative time delays of arrival:

\[
t_{m,c} = \arg \max_\tau \left[ \sum_t h(t - t_c) x_m(t) r(t - \tau) \right]
\] (2.6)

The cross-correlation of calibration data collected for a single microphone with the reference chirp can be seen in Figure 2.4. The peaks shown are the relative time-
Figure 2.4: The cross-correlation of calibration data collected by a single microphone with the reference chirp. Seventeen chirps were used for this calibration.
delays associated with the microphone. The $c$th chirp is selected by the windowing function $h(t)$. The resultant matrix of time delays of arrival is normalized by identifying the center microphone as having zero time delay and relating the other TDOAs respectively, resulting in both positive and negative TDOAs.

The TDOA matrix is smoothed using a polynomial fit along each row to remove outlying time delay of arrival errors. The $x$ and $y$ coordinates of the microphones relative to the test area are then found using an unconstrained nonlinear optimization of the smoothed TDOA matrix.

Calibration need only be performed once for each data collection so long as the microphones remain in the same location. Microphone locations are saved, allowing a user to leave the array in one position for numerous data collections [3].
CHAPTER 3
ALGORITHMS

3.1 Preconditioning

After calibration, data collection begins, and the data is, once again, saved as \( M \) audio signals. Before beamforming, some preconditioning of the recorded signals is required in order to suppress interference and ambient noise clutter. For the experiment outlined in Section III, a band-pass filter from 450Hz to 3kHz was applied to each channel of data. A 450Hz cutoff is high enough to removed most of the ambient clutter noise. A 3kHz cutoff was selected to allow the recorded data to be downsampled from 44.1kHz to roughly 8kHz, thus greatly increasing the processing throughput without loss of significant signal energy. This preconditioning allows for beamforming to be performed more efficiently [3].

Figures 3.1 and 3.2 show the time-frequency plots for data collected by a single microphone before and after the filter described was applied.

3.2 Beamforming

Once the calibration is complete and each channel has been filtered and downsampled, delay-and-sum beamforming is straightforward. For a particular angle, \( \phi \), the time delay of arrival for each data channel is computed as

\[
t_m(\phi) = \frac{1}{v} (x_m \cos \phi + y_m \sin \phi).
\]
Figure 3.1: Single target data collected by a single microphone before filtering.

Figure 3.2: Single target data collected by a single microphone after filtering.
The beamformed signal at aspect $\phi$ is thus

$$X_B(t, \phi) = \sum_{m=1}^{M} x_m(t - t_m(\phi)). \quad (3.2)$$

Non-coherent integration is calculated over segments of $\delta = 1/30$ seconds for each beam to yield the waterfall image,

$$E_B(\tau, \phi) = \int_{\tau}^{\tau+\delta} |X_B(t, \phi)|^2 dt. \quad (3.3)$$

The result is plotted for the duration of the data collection versus DOA ($\phi$) and time ($\tau$) [3].

![Diagram](image)

Figure 3.3: Single target with straight flight paths.

Some results of the beamforming described in this chapter are shown in Figures 3.3 through 3.6. The array was placed linearly and was tested on remote controlled (RC) aircraft that were flown in an open, outdoor field. The RC aircraft flew with varying flight paths in the far-field (relative to the array). Beamforming was performed in $1^\circ$ increments over a field of view of $-45^\circ$ to $45^\circ$, where $0^\circ$ is broadside to the array. For Figures 3.3 through 3.6, the figures on the left represent the planned flight paths of the RC aircraft, while the figures on the right show the corresponding beamformed results.
Figure 3.4: Single target with a zigzag pattern in the field of view of the array.

Figure 3.5: Single target with a zigzag flight path that crosses out of the field of view of the array.
Figure 3.6: Single target with a circular flight path.

Figure 3.7: Waterfall plot for a straight flight path that details the anomalies present in the beamforming results.
In each plot, one can observe various anomalies which are detailed in Figure 3.7. A generator was used to power the acoustic array in the field, and the sound of the generator motor may have caused low-level interference, visible in A. A short-time event in the near-field is also visible in B, which may be the result of anything from a cough, to footsteps. Finally, one can observe the presence of grating lobes in C due to the aliasing of high frequency signal content. This is particularly noticeable for flights with the RC aircraft closer to the array and with higher SNR. With 10-inch element spacing, acoustic signals with wavelengths shorter than 20-inches or frequencies larger than roughly 700Hz will experience spatial aliasing. In general however, the beamformed results and waterfall plots are consistent with the actual flight paths.

3.3 Detection

Due to spatially varying clutter from interference sources and grating lobes, a constant threshold for detection would yield inconsistent detection and false alarm rates. To avoid this, a cell averaging constant false alarm rate (CA-CFAR) filter is used instead.

The CA-CFAR filter uses the minimum variance, unbiased estimate of the clutter and noise power to set a threshold for detection. An average of $\beta_1$ reference cells surrounding the cell under test (CUT) are used to estimate the mean, $\hat{\mu}$ of the unknown noise and clutter power. A target is considered present for a CUT at time $\tau$ and aspect $\phi$, if $E_B(\tau, \phi) \geq \alpha \hat{\mu}$ \cite{8}. Gandhi and Kassam found the CA-CFAR filter to be optimal (in that it maximized the probability of detection for a given probability of false alarm) for non-fluctuating targets in \cite{8}.

For the acoustic array described in this paper, the average ($\hat{\mu}(\tau, \phi)$) and standard deviation ($\hat{\sigma}(\tau, \phi)$) are estimated for each CUT over a sliding window:
\[ \hat{\mu}(t, \phi) = \frac{1}{\beta_1} \sum_{i=-\frac{\beta}{2}}^{\frac{\beta}{2}} X_B(t, i), \quad i \neq \text{i}_\text{CUT} \] (3.4)

\[ \hat{\sigma}(t, \phi) = \sqrt{\frac{1}{\beta_1} \sum_{i=-\frac{\beta}{2}}^{\frac{\beta}{2}} [X_B(t, i) - \hat{\mu}(t, i)]^2}, \quad i \neq \text{i}_\text{CUT} \] (3.5)

The weighting coefficient, \( \alpha \), is chosen to be \( 1 + \hat{\sigma}/\hat{\mu} \). The sliding window is one cell in the time axis by \( \beta_1 \) cells in the DOA axis and does not include the cell under test. Each cell is one time segment (1/30s) by one degree. A CUT is flagged as a detection \( (C(t, \phi_{\text{CUT}}) = 1) \) if it is at least one standard deviation over the mean of the cells in the window: [2]

\[
C(t, \phi) = \begin{cases} 
1, & E_B(t, \phi) \leq \hat{\mu}(t, \phi) + \hat{\sigma}(t, \phi) \\
0, & \text{otherwise}
\end{cases}
\] (3.6)

An example result of the CA-CFAR filter described is shown in Figure 3.9 for the flight path shown in Figure 3.8 where \( \beta_1 \) is set to the full DOA range, 90°. Note that there are detections at numerous DOAs for each time segment, and that there are false alarms around \( \phi = -40^\circ \).

To reduce the number of detections processed only reports that are also local maxima are used for tracking. A cell is identified as a local maximum by using a second sliding window of width \( \beta_2 \). If the CUT is greater than all other cells in the window, it is flagged as a maximum \( (M(t, \phi) = 1) \) [2].

\[
M(t, \phi) = \begin{cases} 
1, & E_B(t, \phi) \geq E_B(t, \hat{\phi}) \forall |\hat{\phi} - \phi| \leq \frac{\beta_2}{2} \\
0, & \text{otherwise}
\end{cases}
\] (3.7)

For each time segment \( t \), \( d_{n,t} \) is a vector of DOAs at which \( V(t, \phi) = C(t, \phi) \cdot M(t, \phi) = 1 \). The number of detections for any target \( n \), and time segment, \( t \), is \( D_t \)
Figure 3.8: Waterfall plot for a single target with a zigzag flight path.

Figure 3.9: CA-CFAR detections for a single target with a zigzag flight path.
so that:

\[ d_{n,t} = [\phi_1 \phi_2 \ldots \phi_D]. \tag{3.8} \]

Figure 3.10: CA-CFAR detections for a single target with a circular flight path.

An example result of applying a local maxima criteria to the CA-CFAR filter is shown in Figures 3.10 and 3.11 where \( \beta_2 \) is set to 10°. Note that the CA-CFAR local maxima approach produces an inconsistent number of detections at 0°. This may be a result of the microphones described in Section 2.1. The microphones are built in Fishermen’s Friend\textsuperscript{TM} lozenge boxes with the microphone cartridges in one face of the boxes. This flat surface may cause the acoustic signature received by the microphones to be strongest for acoustic sources that are broadside to the array, which would result in strong local maxima along 0°.
Figure 3.11: CA-CFAR local maxima detections for a single target with a circular flight path.
## 3.4 Tracking

Tracking is performed for an unknown number of possible targets using a discrete Kalman filter that updates both the direction of arrival of each target and the angular rate of each target. The algorithm tracks $N$ targets, each represented by $\hat{x}_{n,k}$ where $n = 1, 2, \ldots, N$ indexes tracks and $k = 1, 2, \ldots, K$ indexes time segments [2].

The discrete Kalman filter recursively estimates the state of a process, $x_k \in \mathbb{R}^n$, with measurements $z_k \in \mathbb{R}^n$, while minimizing the mean squared error. The state for updating the process is defined as

$$x_k = Ax_{k-1} + w_{k-1},\quad (3.9)$$

and the measurement is defined as

$$z_k = Hx_k + v_k,\quad (3.10)$$

where $w_k$ and $v_k$ are assumed to be independent white Gaussian noise with process noise covariance $Q$, and measurement noise covariance $R$ respectively. The matrix, $A$ in (3.9) relates the state at $k - 1$ to the state at $k$, and the matrix, $H$, in (3.10) relates the state to the measurement. The matrices $Q$, $R$, $A$, and $H$ are all assumed to be constant.

The goal of the Kalman filter is to predict the state of the process at $k$, given only information prior to $k$, and then to adjust the estimate of the state at $k$, with the measurement, $z_k$. The predicted state estimate, $\hat{x}_k^-$, is referred to as the a priori state estimate and is associated with the a priori error covariance matrix: $P_k^- = E[e_k^- e_k^-^T]$ where $e_k^- \equiv x_k - \hat{x}_k^-$. The updated estimate at $k$, $\hat{x}_k$, is referred to as the a posteriori state estimate and is associated with the a posteriori error covariance matrix: $P_k = E[e_k e_k^T]$ where $e_k \equiv x_k - \hat{x}_k$. The a posteriori estimate is updated using a weighted
difference between the measurement at \( k \) and the predicted, \( a \ priori \) state estimate:

\[
\hat{x}_k = \hat{x}_{k-1} + K_k(z_k - H\hat{x}_k).
\] (3.11)

The Kalman gain, \( K_k \), is chosen to minimize the \( a \ posteriori \) covariance matrix, \( P_k \):

\[
K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}.
\] (3.12)

The use of the Kalman gain allows the Kalman filter to adjust its state prediction to accommodate the \( a \ priori \) error covariance matrix. As \( P_k^- \) increases, the Kalman filter “trusts” the predicted measurement, \( H\hat{x}_k^- \), more and the actual measurement, \( z_k \), less.

The Kalman filter can be divided into two sets of equations: the time update equations and the measurement update equations. The time update equations, which project the state and \( a \ priori \) error covariance estimates from \( k - 1 \) to \( k \), for target \( n \) are:

\[
\hat{x}_{n,k}^- = A\hat{x}_{n,k-1}^-.
\] (3.13)

\[
P_{n,k}^- = AP_{n,k-1}^- A^T + Q.
\] (3.14)

The measurement update equations, which update the Kalman gain, and \( a \ posteriori \) state and error covariance matrix estimates, for target \( n \) are:

\[
K_{n,k} = P_{n,k}^- H^T (H P_{n,k}^- H^T + R)^{-1}
\] (3.15)

\[
\hat{x}_{n,k} = \hat{x}_{n,k}^- + K_{n,k}(z_{n,k} - H\hat{x}_{n,k}^-)
\] (3.16)
\[ P_{n,k} = (I - K_{n,k}H)P_{n,k}^{-} \]  \hspace{1cm} (3.17)

For the Kalman filter used in the acoustic array discussed in this paper, the estimate of the state vector, \( \hat{x}_{n,k} \) for each target, is initialized to the first detection associated with each target, and an angular rate of zero.

\[ \hat{x}_{n,0} = \begin{bmatrix} \phi_0 \\ \dot{\phi}_0 \end{bmatrix} = \begin{bmatrix} d_{n,0}(1) \\ 0 \end{bmatrix} \]  \hspace{1cm} (3.18)

The Kalman update assumes measurement noise covariance \( R = 0.1 \cdot I \) and process noise covariance \( Q = 0.001 \cdot I \) [14].

### 3.4.1 Data Association

Incorrectly associated track-measurement pairs could result in gaps in the track or a loss of the track altogether [11]. Data association is then necessary to determine the measurement, \( z_k \), to be used to update each target state estimate. A nearest neighbor standard filter (NNSF) is used to associate measurements with targets, and a track splitting method is used to determine the number of targets being updated.

There are two types of nearest neighbor standard filters: global nearest neighbor (GNN) and sub-optimal nearest neighbor (SNN). Both types use a validation gate to eliminate unlikely measurement-track pairs [11]. A validation gate is defined as the region around the previous state estimate in which the true measurement has a high probability of occurring [1]. A measurement-track pair is validated if it falls within the gate. A GNN approach uses a cost matrix to determine which validated measurement to assign to which track. A SNN approach, on the other hand, simply finds the minimum-distance measurement-track pair [11].

The tracking algorithm described in this paper uses the simpler, sub-optimal nearest neighbor standard filter. A threshold constant of \( G \) is used as a gate to determine
initial association. All detections that fall within $G$ degrees of arrival in either direction of the previous track update are initially associated with the target path. The difference in DOA between each track and each detection is computed as

$$g_{n,k}(i) = |\hat{x}_{n,k-1} - d_{n,k}(i)|$$

(3.19)

where $\hat{x}_{n,k-1}$ is the current DOA estimate for the $n$th track, and $d_{n,k}(i)$ is the DOA of the $i$th current detection. The $n$th track and the $i$th detection are only associated if $g_{n,k}(i) \leq G$ and if that pairing yields the minimum difference among all possible combinations. Tracks and detections are thus associated from the closest to the farthest pairing. The DOA of the $i$th detection paired with the $n$th track is stored as the state $x_{n,k} = d_{n,k}(i)$ for updating the measurement, $z_{n,k}$ in (3.10) [2]:

$$x_{n,k} = \min |g_{n,k}(i) - \hat{x}_{n,k-1}|, \quad i = 1, 2, \ldots, D_t,$$

(3.20)

If no detection is assigned to a track, the Kalman filter coasts to the next time segment. Conflicts arise, however, when multiple detections are validated for a single target $k$. One solution to this conflict is to split the track under conflict into multiple hypothesis tracks, one for each measurement in conflict. Each new track is initialized with a new update state and error covariance matrix. The new tracks are then propagated forward to new validation gates, where the process is restarted. This, however, could result in an infinite number of targets, and therefore “pruning” is needed to reduce the number of tracks. Typically, the number of tracks are reduced by determining the likelihood function of each target path and discarding unlikely target paths [1]. This method, known as a multiple hypothesis tracker, can become very complex, difficult to implement, and computationally exhausting [11]. To simplify the multiple hypothesis tracker, the likelihood of a track may be based only on the length of the track. Incomplete sequences (tracks that are missing measurements) are kept, but tracks that have less than $m$ out of $n$ sampling times are discarded [1].
Assume $u_n$ is the number of time segments that target $n$ has iterated through without a detection. That is, for track $n$, if no detections are validated, $g_{n,k} \neq \emptyset$, $u_n$ is incremented. If a target path iterates through $u_n \geq T_{\text{max}}$ time segments without a detection, it is truncated, and the number of target paths is reduced.

If a detection is not associated with any track at time-segment $k$, another target path, $n$, is added, and the detection, $d_{n,k}(i)$, is assigned to the new target path. The state, $x_{n,k}$ to update $z_{n,k}$ in (3.10), for the new target is the previously unclaimed detection:

$$x_{n,k} = \begin{cases} 
  d_{n,k}(i), & d_{n,k}(i) \cap g_{n,k} = \emptyset \forall n, \quad i = 1, 2, \ldots, D_k \\
  \emptyset, & \text{otherwise.}
\end{cases} \quad (3.21)$$

Conflicts also arise when more than one track claims a single detection, i.e., at an intersection of tracks. Data association becomes difficult for intersecting target paths, especially for targets with similar acoustic signatures. The tracking algorithm described in this paper tracks target paths without regard for specific target association which eliminates the need for a priori knowledge of the targets. As with target splitting, the likelihood of a measurement-track pair is based on the length of the track. The target intersection conflict is resolved by assigning the detection at intersection to the target path that is longest at $k$, and terminating the shorter target path(s) at $k^-$. New target paths may then be started at $k^+$. The tracking algorithm described in this section was used for a single-target case with favorable results. The target path shown in Figure 3.13 for the detections shown in Figure 3.12 does not contain gaps, which suggests that the gaps in multiple target tracks are the result of multiple target intersections.
Figure 3.12: CA-CFAR local maxima detections for a single target with a circular flight path.
Figure 3.13: Tracked target path for a single target with a circular flight path.
Due to the fragility of small RC aircraft, data collections using these aircraft are limited. As a result, no multiple target data was collected and was instead simulated by summing the signals of two single-target flight paths. An example of the beamforming results for two-target data can be seen in Figures 4.1 and 4.2.

A CA-CFAR filter was used with a window-size of $\beta_1 = 90^\circ$, the entire field of view of the array. The results of this filter can be seen in Figures 4.3 and 4.4.
Figure 4.2: Simulated two-target waterfall plot with circular and zigzag flight paths.

Figure 4.3: CA-CFAR detections corresponding to Figure 4.1.
Figure 4.4: CA-CFAR detections corresponding to Figure 4.2.

Figure 4.5: Local maxima CA-CFAR detections corresponding to Figures 4.2 and 4.3.
Figure 4.6: Local maxima CA-CFAR detections corresponding to Figures 4.2 and 4.4.

The local maxima criteria was applied to the CA-CFAR detections using a window of $\beta_2 = 10^\circ$. The local-maxima CA-CFAR detections can be seen in Figures 4.5 and 4.6. Note that there is an inconsistent number of detections at $0^\circ$ as discussed in Section 3.3. There are also gaps in the local maxima CA-CFAR detections, as can be seen on the left in Figure 4.5 at approximately $\tau = 17\, s$ and $\phi = -20^\circ$. These gaps are the result of the target intersection and local maxima criteria. For this two-target scenario, the target with the circular flight path had a higher energy signature and therefore achieved the local maxima criteria when compared to the target with the straight flight path.

A Kalman filter using data association as described in Section 3.4 was used with favorable results which can be seen in Figures 4.7 and 4.8. Allowing detections to be claimed by only one target path results in gaps in the target track such as the gap seen in the right figure at approximately $\tau = 29\, s$ and $\phi = -20^\circ$. In addition, false
Figure 4.7: Tracked target paths for the signal shown in Figure 4.1 using the detections in Figure 4.5.
Figure 4.8: Tracked target paths for the signal shown in Figure 4.2 using the detections in Figure 4.6.
alarm detections, such as those seen clearly in Figure 4.3 at $-40^\circ$ result in false target paths such as those seen in Figure 4.7 at $\phi = -40^\circ$. 
CHAPTER 5
CONCLUSION

The acoustic array described in this paper shows promising results while still fulfilling the initial goals of the project, which were for the array to be both low-cost and mobile. The hardware consists of custom-made and commercial off-the-shelf equipment. The signal processing is relatively simple and easy to implement, consisting of a calibration procedure, beamforming, and multi-target tracking. These algorithms showed favorable results in the single and two-target cases.

Future work may be performed to improve detection for intersecting target paths or to create new likelihood tests to determine detection-track association. In addition, the array described in this thesis may be duplicated to create a two-array system for better source localization. Future work may also include actual multiple-target data collections for further testing.
REFERENCES


