HIGH-RESOLUTION SOURCE IMAGING WITH BIO-INSPIRED SENSING SYSTEMS

BY

NAM NGUYEN

DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Electrical and Computer Engineering in the Graduate College of the University of Illinois at Urbana-Champaign, 2011

Urbana, Illinois

Doctoral Committee:

Professor Douglas L. Jones, Chair
Professor Yoram Bresler
Professor Andreas C. Cangellaris
Associate Professor Minh N. Do
Source localization is ubiquitous in nature. It is a survival skill in many species to help them find food, avoid predators, or navigate. For example, blind cave fish use their lateral lines to swim in dark water by sensing flows, weakly electric fish generate an electric field to detect electric distortions caused by nearby objects, and bats emit ultrasound and listen to echoes to capture insects. It is always the desire and challenge for engineers to build man-made systems that can deliver such capabilities.

In this thesis, two new bio-inspired, man-made sensing systems are developed. Using new hair-cell sensors built from the Micro-Electro-Mechanical-Systems (MEMS) technology, we develop an artificial lateral line system similar to the one of fish. An adaptive beamforming algorithm is used to provide high-resolution images of source locations. The other system is built based on the principle of weakly electric fish. As it is an active sensing system, signals from multiple sources are coherent, and the previous adaptive beamforming fails. We then introduce the concept of sparse beamforming by exploiting the fact that objects to be localized are sparse in space. It is shown that the sparse beamforming technique is capable of resolving coherent sources.

We not only devise those man-made sensing systems, but we also develop new algorithms to process the input sensor signals and enhance the output images.
First, we provide a new $\ell_1$-minimization algorithm using a backward basis elimination technique. The algorithm outperforms the well-known $\ell_1$ magic package for small-scale problems. This algorithm can be used in the sparse beamforming application. Second, we introduce the reassignment method into the source localization problem to sharpen output images. The algorithm is verified in both the artificial lateral line with a vibrating sphere and the weakly electric sensing system with an insulating plastic ball.

Overall, we have demonstrated the practical possibility of constructing novel man-made sensing systems that can imitate several source localization capabilities previously found only in nature.
## Contents

Chapter 1  Introduction ........................................... 1

Chapter 2  Artificial Lateral Line with Adaptive
Beamforming ......................................................... 4
  2.1 Artificial Lateral Line .................................. 6
  2.2 Beamforming Techniques .............................. 10
  2.3 Self-Calibration Algorithm .......................... 12
  2.4 Experiment and Results .............................. 17
    2.4.1 Experimental Setup ............................ 17
    2.4.2 Calibration ..................................... 18
    2.4.3 Dipole Imaging ................................. 19
    2.4.4 Crayfish Imaging ............................. 21
  2.5 Cramer-Rao Bound .................................. 22

Chapter 3  Underwater Electrolocation with Sparse Beam-
  forming ........................................................... 26
  3.1 Electrolocation System Model ....................... 27
  3.2 Sparse Beamforming ................................... 30
  3.3 Simulations .............................................. 32
  3.4 Experiments and Results ........................... 34
    3.4.1 Experimental Setup ............................ 34
    3.4.2 Calibration ..................................... 34
    3.4.3 Single Object Mapping ......................... 35
    3.4.4 Multiple Objects Mapping .................... 36
  3.5 Evaluation of Array Response Model for Multiple Objects . 36

Chapter 4  Projected Gradient Algorithm for Sparse Beam-
  forming ......................................................... 39
  4.1 Description of the Algorithm ....................... 39
  4.2 Optimality Test and Basis Readmission via the Simplex
    Method .................................................... 43
  4.3 Convergence Analysis ................................ 46
  4.4 Performance Analysis ................................ 48
  4.5 Projected Subgradient Algorithm ................... 49
Chapter 5  Source Localization via the Reassignment Method .......................... 53
  5.1  Reassignment Principle ................................................................. 55
      5.1.1  Reassignment for Spectrogram .............................................. 56
      5.1.2  Reassignment for Scalogram .................................................. 58
  5.2  Reassignment for Source Localization ............................................. 59
      5.2.1  Reassignment for Fish Lateral Line ....................................... 60
      5.2.2  Reassignment for the Weakly Electric Fish ............................. 66
  5.3  Reassignment for Sparse Beamforming Speedup ................................. 69
  5.4  Reassignment with Multiple Sources .............................................. 71

Chapter 6  Conclusion .......................................................... 73

Appendix A  Reassignment of Spectrogram and Scalogram ......................... 76
  A.1  Reassignment Derivation for Spectrogram and Scalogram ................. 76
      A.1.1  Spectrogram as 2D Convolution of WV Distributions ............... 76
      A.1.2  New Reassignment Formula for Spectrogram ......................... 78
      A.1.3  Scalogram as 2D Convolution of WV Distributions ............... 79
      A.1.4  New Reassignment Formula for Scalogram ............................ 80

References .................................................. 82
Chapter 1

Introduction

Through millions of years of evolution, species in nature have developed complex and sophisticated sensing capabilities, on which they rely for everyday survival activities such as navigation, predator avoidance, hunting, and food gathering. For example, the lateral lines along the body of many fish species are known to help them navigate even in murky water and to school [1, 2]. Bats use their ultrasound echolocation capability to catch flying insects or avoid obstacles in caves [3, 4]. Some spiders can use their hairs to sense the air movement caused by the subtle wing-flapping of a fly [5]. Not just passively sensing, some fish even actively generate an electric field around their bodies to detect the presence of objects in the surrounding water [6, 7].

Inspired by these intrinsic sensing capabilities in nature, our goal is to study and develop bio-inspired man-made systems that can mimic some of these capabilities. This approach could have some significant implications if successful. First, it can directly offer many new applications. Submarines and underwater autonomous vehicles can navigate in narrow spaces with a near-field sensing capability. Cars can have a smart skin to reduce the risk of collision. Second, we can better understand certain features and limitations of the biological systems by applying theoretical knowledge to the models of those systems. Last, we may devise new signal-processing algorithms by learning the mechanisms used by those species.
In this thesis, we focus more on the first implication, i.e., the application aspect. We explore the source localization capability in different bio-inspired sensing systems. First, we develop and demonstrate an artificial lateral line that mimics the capability of imaging flow sources in fish. Using an adaptive beamforming technique, we manage to construct 3D images of a vibrating sphere in water. Second, we also study weakly electric fish and design a system operating on a similar principle. We introduce a new sparse beamforming framework and algorithm to map the source locations in the near field. Finally, we provide a new method for enhancing source localization for both of the above systems. The images of source distribution are sharpened using the reassignment method adapted from time-frequency signal analysis.

Below is the outline of our work:

- **Chapter 2** presents a complete development of an artificial lateral line. Using hair-cell sensors designed and fabricated by our collaborators based on the Micro-Electro-Mechanical-Systems (MEMS) technology, we assemble and model the response of a sensor array. We then provide algorithms for both system self-calibration and signal processing. The performance of the system is demonstrated and tested through a range of experiments with both standard flow sources and unstructured but natural sources. The limitation of the system’s performance is also analyzed in this chapter.

- **Chapter 3** focuses on a new bio-inspired sense analogous to the weakly electric fish system. A system model is built upon the same principle as used by the electric fish. A new beamforming framework is intro-
duced to process the output signals of the system. The results of both simulation and experimental tests indicate a promising application of this technique.

- *Chapter 4* proposes a new optimization algorithm adapted for the new beamforming framework. The algorithm solves the overcomplete linear inverse problem by a new backward basis elimination technique. It joins the family of $\ell_1$-minimization algorithms for sparse representation recovery.

- *Chapter 5* provides a new technique that can be used for source localization with both the fish lateral line system and weakly electric fish. The technique is adapted from the reassignment method introduced by Kodera et al. in [8] to improve the readability and sharpness of spectrograms. We demonstrate that the reassignment method can sharpen the images of either a vibrating sphere via an artificial lateral line or an insulating plastic ball via a weakly electric sensing system.
Chapter 2

Artificial Lateral Line with Adaptive Beamforming

Most fish can sense dynamic flows using the lateral-line sensory organs along their bodies [9] (Figure 2.1A). A lateral line consists of numerous tiny hair-cell receptors, which invoke neural spiking when moving fluid causes the hairs to bend (Figure 2.1B and 2.1C). Fish use this sensing capability to perform a wide range of activities including navigating in dark and murky water, localizing prey, avoiding predators, and schooling [10, 11].

Figure 2.1: (A) Distribution of neuromasts, a cluster of hair-cell organs, on a fish’s body (modified from [12]). (B) Close-up image of a neuromast which is made of many hairs (modified from [13]). (C) Illustration of a neuromast’s function to generate excitation spikes when hairs are bent (modified from [14]). (D) Testing platform with 15 artificial hair-cell sensors. (E) A close-up image of a sensor (modified from [15]). (F) Operating principle of the artificial hair-cell sensor [16].
This fish flow-sensing capability is the obvious motivation to build a man-made system with equivalent functionality. Such a system, if realized, could lead to many applications in oceanic exploration and underwater navigation. Imagine a submarine with smart skin maneuvering in narrow water or an autonomous vehicle tracking a wake left behind a ship or a submarine. It could be a great application of both military and commercial interest.

Thanks to recent developments in Micro-Electro-Mechanical-Systems (MEMS) technology, it is now possible to realize this sensing capability with artificial hair-cell sensors [17, 18]. The MEMS sensor, developed by Prof. Chang Liu’s group at UIUC in collaboration with us, also has a micrometer-scale hair, which bends when intercepting a flow. Depending on the level of bending, the mechanical force applied to the hair is translated to an electrical signal with a certain magnitude. Those sensors can be mass-produced and cheaply assembled into a larger sensing system such as a man-made lateral line [19].

In this chapter, we will study the world’s first artificial MEMS hair-cell lateral line. We deliver a complete system from a physical structure, to a system model, and a signal-processing algorithm. We focus on a specific application to image a simple but typical flow-generating source, the dipole source. The system can produce a radar-like image for a near-field range. The rest of this chapter is organized as follows. Section 2.1 describes the structure and operating mechanism of single MEMS hair-cell sensors before assembling them into an array configuration. We also model the array responses to a dipole source. Section 2.2 explains in detail the beamforming algorithm and why we select Capon’s method. Before presenting the experimental results in Section 2.4, we introduce a new self-calibration algorithm to improve the output
images in Section 2.3. Last, we also derive in Section 2.5 the theoretical performance limit, which can help to understand fundamental limitations of lateral-line systems and to provide guidance in designing and using artificial lateral-line systems.

2.1 Artificial Lateral Line

Figure 2.1E illustrates a man-made hair-cell sensor consisting of a horizontal cantilever with a vertical hair attached at the distal end and an embedded piezoresistor at the fixed end. When a flow impinges upon the vertical hair, a bending force acts on the horizontal cantilever to induce a stress change at the piezoresistor, causing a change in resistance (Figure 2.1F). When an electric current runs through the piezoresistor, the change in the resistance causes a change in the voltage across the sensor. Within an operational range, the change in electrical signal strength is linearly proportional to the change in the force acting on the hair, which can be used to infer the local flow velocity [16, 20, 19].

Note that, due to its physical structure, the sensor is sensitive to flows along the direction of the horizontal cantilever and almost insensitive to flows in the perpendicular direction. By inspecting a number of sample sensors, we find that the sensor’s directivity pattern or response function is approximately a figure-eight shape with two peaks (top and bottom of the figure-eight shape) lying along the direction of the horizontal cantilever. Therefore, the direction of the horizontal cantilever defines the orientation of the sensor, an important parameter when assembling sensors into an artificial lateral line.
Using these MEMS sensors, an artificial lateral line can be built and installed on Autonomous Underwater Vehicles (AUV) for sensing surrounding flow activities. Figure 2.1D shows the working prototype of a lateral line consisting of 15 sensors on a half-cylinder surface which models the body of an AUV. The cylinder has an outer diameter of 89 mm and a length of 240 mm. There are 9 sensors along the central line of the cylinder surface with linear spacing of 30 mm, and 6 sensors on both sides of the middle cross curvature with angular spacing of 30°. Also note the alternating pattern of the sensors’ orientation. This configuration is selected to achieve good 3D localization performance with a limited number of sensors based on numerical evaluation of the Cramer-Rao lower bound.

One of the most common forms of water disturbance that is extensively used as a stimulus to study fish lateral-line organs is the acoustic dipole source, or a sinusoidally vibrating sphere [21]. Since dipole motions are also the main components in the model of a fish’s tail-beating movement, these responses can also be used to localize many real-world sources such as a tail-flicking crayfish.

**Model of Sensor Response to a Dipole Source**

Consider a dipole source of diameter $a$ placed at the origin and oscillating along the $z$ axis at frequency $f$ and amplitude $U_o$, i.e., $z(t) = U_o \sin(2\pi ft)$. The flow velocity at all points at distance $r$ and symmetrically positioned around the $z$ axis by angle $\theta$ is derived in [22] as

$$
\vec{v}_{flow}(r, \theta) = \left( a^3 U_o \cos(\theta) \right) \hat{r} + \left( a^3 U_o \sin(\theta) \right) \hat{\theta},
$$

(2.1)
where $\hat{r}$ and $\hat{\theta}$ are unit vectors of the dipole’s spherical coordinates at the sensor’s position as shown in Figure 2.2A.

![Diagram showing dipole's spherical coordinates and the sensor's position](image)

Figure 2.2: (A) Dipole’s spherical coordinates with $(r, \theta, \phi)$ defining the location with respect to the dipole and $(\hat{r}, \hat{\theta}, \hat{\phi})$ defining the unit vectors of the dipole’s spherical coordinates at the sensor’s position. (B) Lateral line’s Cartesian coordinates

Equation (2.1) indicates the flow velocity at any sensor location around the dipole source. Since we are more interested in knowing the sensor’s response relative to the dipole location with respect to the sensor, it is more convenient to translate Equation (2.1) from the dipole’s spherical coordinates to the lateral line’s Cartesian coordinates as shown in Figure 2.2B. So if in the lateral line’s Cartesian coordinates, $\vec{s} = (x_s, y_s, z_s)$ specifies the position of a sensor and $\vec{d} = (x_d, y_d, z_d)$ indicates the location of the dipole source, then the flow velocity seen at the sensor is

$$\vec{v}_{flow}(\vec{d}) = \frac{a^3 U_o}{2r^3} (3 \cos(\theta) \hat{r} - \hat{z}_d)$$  \hspace{1cm} (2.2)
where \( \hat{z}_d \) is the unit vector specifying the oscillating direction of the dipole source, and

\[
r = \|\vec{s} - \vec{d}\| \quad \text{and} \quad \hat{r} = \frac{\vec{s} - \vec{d}}{\|\vec{s} - \vec{d}\|}.
\]

From Equation (2.2), it is clear that the flow velocity impacting on the sensor’s hair is not necessarily in the horizontal plane of the cantilever. In that case, we can project the flow velocity \( \vec{v}_{\text{flow}} \) onto the horizontal plane and approximate that projected component \( \vec{v}_{\perp} \) as the final flow velocity sensed by the sensor’s hair. Therefore, the vector that defines the horizontal plane (which also defines the direction of the sensor’s hair) is also an important parameter of the sensor. So a sensor in a lateral line is determined by three vectors: the position vector \( \vec{s} \), the orientation vector \( \vec{u} \), and the hair direction vector \( \vec{h} \) as shown in Figure 2.1E.

Also recall that the directivity pattern or the response function of a sensor has a figure-8 shape. So if \( \alpha \) is the angle between the orientation vector \( \vec{u} \) and \( \vec{v}_{\perp} \) as the projection of the flow velocity onto the horizontal plane, we can model the sensor response of the dipole source at location \( \vec{d} \) as

\[
f(\vec{d}) = \|\vec{v}_{\text{flow}}\| (a \cos \alpha + b \sin \alpha + c)
\]

(2.3)

where \( a, b, c \) are the directivity parameters defining the figure-8-shaped response pattern. For an ideal sensor, we should have \( a = 1, b = c = 0 \). In practice, those parameters will be computed for each sensor by a self-calibration algorithm explained later.
2.2 Beamforming Techniques

With the artificial lateral line as the hardware platform to collect signals from the outside world, we need to develop a signal-processing algorithm to process those collected signals. The goal of the signal-processing algorithm is to produce a 3D image of the energy-level distribution of flow-creating moving objects in the surroundings. The approach is to scan all possible source locations in the surrounding region and to compute a likelihood function representing the level of agreement between the measured excitation array pattern and the expected pattern assuming a dipole source at that location. The expected patterns can be derived from the analytical model in Equations (2.2) and (2.3) or from training templates. Different algorithms may produce different likelihood functions. In [23], the authors used a continuous wavelet transform (CWT) technique to produce a 2D contour map. From such a map, we can infer the region of a single dipole source; but the CWT has very low resolution, so the source is difficult to locate precisely, and multiple sources probably could not be distinguished. Similarly, [24] used a mean squared-error function to generate 2D images for detection and localization of a single source. Although this method is a maximum-likelihood estimator under single-source assumptions, it also produces low-resolution images.

Our approach [25] is based on a minimum-variance beamforming framework, or Capon’s method. In particular, assume that there are \( N \) sensors on the array and we sample \( K \) possible source locations surrounding the array (i.e. \( \vec{d}_k \) with \( 1 \leq k \leq K \)). Let \( \mathbf{c} \) be a vector of the measured data from the array and \( \mathbf{e}_k \) be the expected array pattern stimulated by a dipole source at position \( \vec{d}_k \); that means \( \mathbf{e}_k = [f_1(\vec{d}_k), f_2(\vec{d}_k), \ldots, f_N(\vec{d}_k)]^T \) where \( f_n(\vec{d}_k) \)
is the response of sensor \( n \) according to the model in Equations (2.2) and (2.3). The basic concept of the minimum-variance beamforming framework is that for each scanned location \( k \), an optimal weight vector \( w_k \) is applied to the sensor data to recover the signal from the point \( k \) while maximally suppressing the noise and signals from other locations. Mathematically, \( w_k \) is the solution to the optimization problem

\[
\min_{w_k} E \left[ |w_k^H c|^2 \right] \quad \text{subject to } \Re \left[ w_k^H e_k \right] = 1
\]

which is given by [26] as

\[
w_k = \frac{R^{-1} e_k}{e_k^H R^{-1} e_k}
\]

and the final output power corresponding to location \( k \) is

\[
\frac{1}{e_k^H R^{-1} e_k}
\]

where \( R \) is the correlation matrix of the sensor array signals [26].

Applying Capon’s method, we can build the flow-imaging algorithm as follows.

- Step 1: Using the analytical model from Equations (2.2) and (2.3), compute the expected array pattern \( e_k \) for all locations of interest \( 1 \leq k \leq K \):

\[
e_k = [f_1(\vec{d}_k), f_2(\vec{d}_k), \ldots, f_N(\vec{d}_k)]^T.
\]

Note that the sensor response depends not only on the position vector \( \vec{d}_k \) but also on the oscillation orientation of the dipole source which is defined by the azimuth angle \( \theta_d \) and the zenith angle \( \phi_d \). Therefore,
there are actually two more angular dimensions to scan through for each point \( k \).

- **Step 2:** Compute the pseudo-correlation matrix or the mean outer-product of the signals measured from all sensors. If the signals from all sensors are sampled for a certain duration to collect \( L \) samples for each channel, i.e., \( c[i] \) for \( i = (1, 2, \ldots, L) \), then the outer-product-based empirical estimate of the correlation matrix is

  \[
  R = \frac{1}{L} \sum_{i=1}^{L} c^H[i] \ast c[i].
  \]

- **Step 3:** Use Capon’s method to compute the energetic magnitude of any flow source at each scanned location \( k \):

  \[
  E_k = \frac{1}{e_k^H R^{-1} e_k}.
  \]

- **Step 4:** Plot the energy-level 3D image with \( E_k \) representing the energy at point \( k \) in the space. The high-energy regions in the image correspond to the locations of dipole sources.

This algorithm implementing Capon’s method produces much higher resolution 3D images of flow activities than the other two algorithms mentioned above.

### 2.3 Self-Calibration Algorithm

Calibration is an indispensable step when working with actual physical systems. The performance of a system can be significantly enhanced if the system is calibrated well; in many cases, accurate calibration is essential
even to get intelligible results. In our case, each of these MEMS flow sensors in the array has different gain sensitivity and directivity pattern due to manufacturing and installation. This is especially true with sensors that are still in the laboratory stage of development. Even with commercial sensors operating long-term in an open environment such as an array deployed on an AUV, on-line calibration to compensate for environmental drift such as biofouling or damage is essential to maintain good performance. Calibration is therefore an essential pre-processing step to enhance performance and to support practical deployment of the system.

In Equation (2.3), the total response of a sensor is modeled as \((a \cos \alpha + b \sin \alpha + c)\) or can be re-written as \(g(\cos \alpha + \lambda \sin \alpha + \mu)\) where \(g = a\), \(\lambda = b/a\), \(\mu = c/a\). Hence, the two parameters \(\lambda\) and \(\mu\) are directivity coefficients defining the shape and orientation of a figure-8 directivity pattern, while the parameter \(g\) is the total gain which captures both the mechanical and electrical gain of the whole sensor channel from the mechanical bending of the hair to the output of the final electrical signal amplifier or analog-to-digital converter.

The calibration process can be separated into the sensor-directivity-pattern calibration and the sensor gain calibration. The sensor-directivity-pattern calibration determines the two directivity parameters \(\lambda\) and \(\mu\). It can be done easily before installation on the array platform by measuring the signal strength of the sensor under test when moving a dipole source circularly around the sensor at a fixed distance. If there are \(n\) testing points on the circle, for each point \(i\) we have sensor reading \(y_i = (a \cos \alpha_i + b \sin \alpha_i + c)\) with the known angle \(\alpha_i\). We can then formulate the problem as a system of
linear equations:

\[
\begin{bmatrix}
  y_1 \\
  y_2 \\
  \vdots \\
  y_n \\
\end{bmatrix}
\approx
\begin{bmatrix}
  \cos \alpha_1 & \sin \alpha_1 & 1 \\
  \cos \alpha_2 & \sin \alpha_2 & 1 \\
  \vdots & \vdots & \vdots \\
  \cos \alpha_n & \sin \alpha_n & 1 \\
\end{bmatrix}
\begin{bmatrix}
  a \\
  b \\
  c \\
\end{bmatrix}
\]

or

\[
y \approx A [a \ b \ c]^T.
\]  

(2.5)

So \(a, b, c\) in Equation (2.5) can be solved as \([a \ b \ c]^T = (A^T A)^{-1} A^T y\) to best fit the sensor readings \(y\) in a least-squared error sense. Therefore, we can estimate the directivity parameters \(\lambda = b/a\) and \(\mu = c/a\), which are independent of the size of the testing circle. This calibration process can also be applied after the installation of sensors on the platform.

The sensor gain calibration involves computing the relative gains of all sensors on the platform. We jointly fit the gains of all sensors so as to minimize the squared error with respect to the expected model. Therefore, this process is also called the lateral line calibration. Since each sensor may react differently to the change of surrounding conditions (e.g. temperature, density, etc.), the relative gains may deviate gradually by time and place. In this thesis, we provide an online self-calibration algorithm that can help the system update the gains itself during the operation.

In the rest of this section, we explain the general concept of the self-calibration algorithm and refer readers to the next section for a specific deployment of this algorithm in our system. Suppose a dipole source moves along a simple...
path such as a straight line during the calibration process (which would commonly occur with a moving vessel passing a stationary dipole or vice versa), measurements are taken for \( M \) positions spaced at equal distance along the path. Denoting \( \vec{d}_0 \) and \( d^* \) as the initial position and spacing vectors, then all \( M \) positions are \( \vec{d}_0, \vec{d}_0 + d^*, \ldots, \vec{d}_0 + (M - 1)d^* \). With \( N \) sensors attached on the platform, the expected response of sensor \( i \) from the dipole at position \( j \) can be computed using Equations (2.2) and (2.3) as

\[
g_i f_i (\vec{d}_0 + (j - 1)d^*).
\]

Scanning through \( M \) dipole positions produces the measurement matrix \( C \) for \( M \) expected array patterns:

\[
C \approx \begin{bmatrix}
g_1 f_1 (\vec{d}_0) & g_2 f_2 (\vec{d}_0) & \cdots & g_N f_N (\vec{d}_0) \\
g_1 f_1 (\vec{d}_0 + d^*) & g_2 f_2 (\vec{d}_0 + d^*) & \cdots & g_N f_N (\vec{d}_0 + d^*) \\
\vdots & \vdots & \ddots & \vdots \\
g_1 f_1 (\vec{d}_0 + (M - 1)d^*) & g_2 f_2 (\vec{d}_0 + (M - 1)d^*) & \cdots & g_N f_N (\vec{d}_0 + (M - 1)d^*)
\end{bmatrix}
\]

The aim of calibration is to find a set of sensor gains \([g_1, g_2, \ldots, g_N]\) and position parameters \( \vec{d}_0, d^* \) in order to approximate \( C \) as closely as possible. The position parameters \( \vec{d}_0, d^* \) can be used as indicators of the approximation accuracy.

In the model for the measurement matrix \( C \) above, the right-hand side can be separated into two matrices, one as a linear function of sensor gains
\[ \mathbf{C} \approx \begin{bmatrix}
  f_1(\vec{d}_0) & f_2(\vec{d}_0) & \cdots & f_N(\vec{d}_0) \\
  f_1(\vec{d}_0 + \vec{d}^*) & f_2(\vec{d}_0 + \vec{d}^*) & \cdots & f_N(\vec{d}_0 + \vec{d}^*) \\
  \vdots & \ddots & \ddots & \vdots \\
  f_1(\vec{d}_0 + (M-1)\vec{d}^*) & f_2(\vec{d}_0 + (M-1)\vec{d}^*) & \cdots & f_N(\vec{d}_0 + (M-1)\vec{d}^*)
\end{bmatrix} \]

\begin{bmatrix}
  g_1 & 0 & \cdots & 0 \\
  0 & g_2 & \cdots & 0 \\
  \vdots & \ddots & \ddots & \vdots \\
  0 & 0 & \cdots & g_N
\end{bmatrix}.

The problem becomes a mixed linear-nonlinear least-squares problem, which can be solved via an iterative approach by alternatively optimizing between nonlinear and linear coefficients while keeping the other fixed [27]. The algorithm is summarized as follows:

- **Step 1:** Initialize with uniform gains \( g_1 = g_2 = \ldots = g_N = 1 \) and initial values for \( \vec{d}_0, \vec{d}^* \).

- **Step 2:** Fix the position vectors \( \vec{d}_0, \vec{d}^* \), and compute the gains \( g_1, g_2, \ldots, g_N \) as a least-squares solution to best fit the measurement matrix \( \mathbf{C} \) in Equation (2.6).

- **Step 3:** Fix the gains, then use a gradient method to search for next values of the nonlinear parameters \( \vec{d}_0, \vec{d}^* \) that minimize the Frobenius norm between the measurement matrix \( \mathbf{C} \) and the optimized product on the right-hand side of Equation (2.6).

- **Step 4:** Iterate steps 2 and 3 until the Frobenius norm is small enough or \( \vec{d}_0, \vec{d}^* \) converge asymptotically to some values. Usually the number
of iterations can be set manually after some trials.

Note that the algorithm can be applied for any number of nonlinear parameters. To simplify the explanation, we just use two position parameters $\vec{d}_0, \vec{d}^s$ to specify $M$ locations of the dipole source (i.e., a straight-line path) for calibration. However, we actually use four position parameters (two along the axis and two along the arc of the cylinder) to define locations of the calibrated source in our experiment.

2.4 Experiment and Results

So far, we have presented all main components of our lateral-line system from sensor hardware, and analytical sensor response models, to self-calibration and signal-processing algorithms. To show how those components work together in practice, we present the results of a complete experimental procedure from setting up the system, calibrating sensors and the lateral line, to generating the final outputs.

2.4.1 Experimental Setup

Figure 2.3 displays the setup of the experiment. A dipole source powered by an electric motor and moving up and down at a frequency of 45 Hz with a peak-to-peak amplitude of 10 mm is fixed at the center of a large water tank. Fifteen sensors are attached to a cylinder in a cross formation as described in Section 2.1. The cylinder is attached to a metal frame to allow easy adjustment of its position and angle relative to the dipole source. Denoting the length of the cylinder (the largest distance between any two sensors) as one body length unit ($BL = 240$ mm), we mainly work with the dipole source within the near-field range of 0.5 BL to the surface of the cylinder. Beyond
this distance, the signal magnitude and localization resolution fall off rapidly.
The dipole orientation and offset position (position projected on the cylinder surface) can be changed by adjusting the cylinder frame position and sliding the cylinder on the frame.

### 2.4.2 Calibration

In order to achieve best performance, both the individual sensors and the overall lateral line need to be calibrated. Sensor calibration aims at computing the directivity parameters defining the figure-8-shaped directivity pattern. Ideally, this can be done after the manufacturing process by measuring the sensor response with a dipole source circling around it at a fixed range. In our experiment, we ran a number of tests with different dipole positions and used the Minimum Mean Square Estimator (MMSE) to best-fit our model.
to the collected data.

The calibration of the lateral line (the sensor gain estimation) applies the self-calibration algorithm explained in Section 2.3. Specifically, we collect calibration data by stepping the dipole source from one sensor to the next one at a 0.5 BL range. For each run, we compute the array pattern by extracting the signal strength at the dipole frequency (45 Hz). Note that the testing dipole source actually moves along the central line and then along the cross-curvature line at the middle. Therefore, we have two initial position vectors and two step-sizes (longitudinal and angular) as the nonlinear parameters to be estimated in the self-calibration algorithm.

### 2.4.3 Dipole Imaging

Before we can apply the beamforming algorithm to localize a dipole source at arbitrary positions and vibrating directions, we need to compute the expected array patterns for the dipole source at all the locations of interest in 3D space and for dipole orientations of all different directions. There will thus be five parameters to scan through, including three parameters defining the dipole source position in 3D space and two parameters defining the dipole source vibrating direction. This computation is heavy but can be done once offline.

Executing the beamforming algorithm is straightforward although potentially expensive depending on the size of the array and the scan region. However, this algorithm can be easily implemented as a multi-threaded program to enable parallel computing.

For each experimental run, we recorded signals for 10 seconds at a 1 kHz
sampling rate. The signals are then bandpass filtered at a central frequency of 45 Hz, which is the vibrating frequency of the dipole source. This step is not necessary although it increases the SNR; wideband and multiple sources at different frequencies can be imaged without this step.

Figure 2.4 shows a 3D energy-distribution image demonstrating the localization of a dipole source in front of the center point of the half-cylinder surface at a range of 0.5 BL. There is a well-defined high-energy spot corresponding to the location of the dipole source. Although we search for the best match of the dipole source in both location and vibrating direction, we only display the location on the 3D map. In this case, the estimated vibration orientation also matches well with that of the dipole source, which vibrates along the cylinder’s main axis.

\[ z(t) = A \sin(2\pi ft) \]

Figure 2.4: A 3D energy-like image of a dipole source at the central position and vibration along the cylinder

Figure 2.5 presents the localization results for various positions of the dipole source. The circles represent the actual locations of the dipole source while
Figure 2.5: More results of dipole imaging for different locations shown by cutting through the center of the 3D images along the cylinder, across the cylinder, and along the curved surface of the cylinder. The solid dots represent the estimated ones. In order to have a good view of a 3D image, we slice the 3D image through the peak in three directions, i.e. along the cylinder, across the cylinder, and along the curvature of the cylinder. The concentration of the high-energy spots differs for each location of the dipole source, but this is theoretically expected as will be shown in the Cramer-Rao bound analysis in the next section.

2.4.4 Crayfish Imaging

To further explore the performance of the artificial lateral line in a real-world environment and with real-world signals which are not exactly dipole sources, we selected a tail-flicking crayfish as a hydrodynamic stimulus. The moving
tail of the crayfish was brought near to the cylinder, close to the central sensor (Figure 2.6A). The signal recorded from the central sensor shows a pulsed pattern rather than the sinusoidal patterns generated by the dipole source (Figure 2.6B). However, still using the previous ideal template generated from the dipole source flow model, we applied the beamforming algorithm unaltered to the crayfish data and still achieved sharp localization results as shown in Figure 2.6C. This confirms that the proposed method can robustly handle a real-world signal source even just using a simple dipole signal model in the beamforming algorithm.

2.5 Cramer-Rao Bound

As mentioned in Section 2.1, the configuration of the lateral line was designed based on knowledge of the fundamental performance limits of our system. Cramer-Rao lower bound (CRLB) provides an upper bound on the smallest average estimation error the system could possibly achieve, so we derive the CRLB for a lateral-line array and use it both to optimize the array design and to understand the fundamental performance limits of any lateral-line system.

Consider the model for a signal seen by sensor $n$ in the lateral line as

$$c_n = f_n(\vec{d}) + w_n$$  \hspace{1cm} (2.7)

where $w_n$ is additive white Gaussian noise with distribution $\mathcal{N}(0, \sigma_n^2)$, and $f_n(\vec{d})$ is the expected reading at sensor $n$ produced by a dipole at location $\vec{d}$. In our case, $f_n(\vec{d})$ can be computed by Equation (2.3) and $\vec{d} = (x_d, y_d, z_d)$, the spatial coordinates of the dipole source. Then the CRLB theorem in [28]
states that the variance matrix in estimating position vector $\vec{d}$ is limited by

$$Var[\vec{d}] \geq [\mathbf{F}]^{-1} \quad (2.8)$$

where $\mathbf{F}$ is the Fisher information matrix, which can be derived as

$$\mathbf{F} = \frac{1}{\sigma_N^2} \begin{bmatrix} \sum_{n=1}^{N} \left( \frac{\partial f_n}{\partial x_d} \right)^2 & \sum_{n=1}^{N} \frac{\partial f_n}{\partial x_d} \frac{\partial f_n}{\partial y_d} & \sum_{n=1}^{N} \frac{\partial f_n}{\partial x_d} \frac{\partial f_n}{\partial z_d} \\ \sum_{n=1}^{N} \frac{\partial f_n}{\partial x_d} \frac{\partial f_n}{\partial y_d} & \sum_{n=1}^{N} \left( \frac{\partial f_n}{\partial y_d} \right)^2 & \sum_{n=1}^{N} \frac{\partial f_n}{\partial y_d} \frac{\partial f_n}{\partial z_d} \\ \sum_{n=1}^{N} \frac{\partial f_n}{\partial x_d} \frac{\partial f_n}{\partial z_d} & \sum_{n=1}^{N} \frac{\partial f_n}{\partial y_d} \frac{\partial f_n}{\partial z_d} & \sum_{n=1}^{N} \left( \frac{\partial f_n}{\partial z_d} \right)^2 \end{bmatrix} \quad (2.9)$$

where $f_n$ denotes for $f_n(x_d, y_d, z_d)$. Applying the results above, we compute the CRLB for our system with 14 sensors ($N = 14$), because one sensor was broken during installation. Figure 2.7 illustrates the CRLB ellipses after projecting onto three planes. The size of each ellipse indicates the performance limit at the corresponding point. The smaller the size of an ellipse, the better estimation performance the system could achieve. The slight asymmetry of these uncertainty ellipses is due to a broken sensor at the second position on the right side of the array. From the graphs, we can conclude that the system performs better at points in front of the center of the cylinder. As the dipole source moves to the end or the edge of the array, the performance is degraded. Also, the system seems to perform well within the range of one body length of the array. Beyond that one-body-length range, the performance decays very quickly. Thus a lateral-line sense is inherently a short-range, near-field sense.

The CRLB can help to evaluate the performance limit of the system before actually installing sensors on the platform. It can be used as a guide to design a complex sensor array with many sensors.
Figure 2.6: Crayfish imaging: (A) An image of a real crayfish on the top, near the center of the cylinder. (B) A sample of the tail-flipping signal waveform detected by one sensor in the array. (C) 2D image showing the relative level of dynamic flow activities caused by the crayfish.
Figure 2.7: Ellipses display Cramer-Rao bounds for different dipole locations by projecting on three planes. Three plots in the top row illustrate the projections of dipole locations on three planes. Plots in the bottom row depict the Cramer-Rao lower bound ellipses after projection. The slight asymmetry of these uncertainty ellipses is due to a broken sensor at the second position on the right side of the array.
Weakly electric fish have been a subject of intense study in neurobiology for their ability to emit and sense electric fields. This ability allows them to hunt in total darkness and muddy environments where vision becomes useless [6, 7]. For example, the black ghost knifefish (Apteronotus albifrons) found in the Amazon Basin uses an electric organ and receptors distributed over the length of its body in order to locate insect larvae [29]. Another well-known weakly electric fish is Peters’ elephantnose fish (Gnathonemus petersii). These fish can be found in rivers of West and Central Africa. It has poor eyesight and uses a weak electric field, which it generates by muscular contractions, to find food, to navigate in dark or turbid waters, and to find a mate [30].

Generally, a weakly electric fish creates an electric field around its body like the field of an electric dipole. When an object enters this field, the conductivity difference between the object and surrounding water causes a change in electric current density. Less conductive objects will decrease local current density causing an electrical dark spot. More conductive objects increase local current density causing an electrical bright spot. A fish with electoreceptor organs covering its body can detect the perturbation and learn to estimate the location of the objects [31, 32]. Based on this principle, we can design an artificial weakly electric sensing system. This system, if applicable, can
be used not just under the water but also in the air. In fact, recent work by MacIver and colleagues shows that they can build robotic systems with an electric sense to locate and track objects either underwater or in the air [33, 34, 35].

As with the lateral-line application, our goal is to image the surroundings using a weakly electric sense. Our approach is to use a beamforming technique. However, the challenge in this system is that all the sources are correlated so that the beamforming technique in Chapter 2 is not applicable. In this chapter, we use a new approach called sparse beamforming which has been introduced in [36]. The sparse beamforming framework translates a beamforming problem into an $\ell_1$-norm minimization problem and then uses any current $\ell_1$-norm minimization algorithm to solve it. We make a contribution by applying this concept to create an artificial weakly electric sense. Most of the work in this chapter has been published in [37].

The remainder of the chapter is organized as follows. We first present a model of an electrolocation system in Section 3.1. Next, Section 3.2 introduces the concept of sparse beamforming which can be used for source localization in this application. In order to demonstrate the success of both our man-made electrolocation system and the sparse beamforming concept, we provide both simulation and experimental results in Sections 3.3 and 3.4.

### 3.1 Electrolocation System Model

Placing an object with a different conductivity than the surroundings in an electric field alters the field. If the object is conductive, the electric field
moves the free electrons to one side of the object and creates an induced electric dipole. In general, when a sphere of radius $a$ with conductivity of $\sigma_{\text{object}}$ is placed in water with conductivity of $\sigma_{\text{water}}$ at the point $\vec{r}$ with electric field $\vec{E}_f$, the perturbation caused by the induced electric field was derived in [31]:

$$\Delta \phi(\vec{r}) = \frac{a^3 \vec{E}_f \cdot \vec{r}}{||\vec{r}||^3} \left( \frac{\sigma_{\text{object}} - \sigma_{\text{water}}}{\sigma_{\text{object}} + 2\sigma_{\text{water}}} \right),$$ \hspace{1cm} (3.1)

where $\Delta \phi(\vec{r})$ is the change in potential at position $\vec{r}$ relative to the sphere’s center and $||.||$ is the magnitude of a vector. Based on the model in (3.1), we design a weakly electric field sensing system consisting of two electrodes to form an electric dipole and an array of electric field sensors aligned with the electrodes as shown in Figure 3.1. This is crudely analogous to the biological system found in a weakly electric fish, which generates an oscillating electric field at its head and tail and which has several hundred electrosensors dis-

Figure 3.1: A weakly electric field sensing system
tributed across its body. In our system, the electric field sensors are simply made of pairs of electrodes placed symmetrically about the dipole axis. If the dipole has charges of $+Q$ and $-Q$, then the electrostatic field at a point is

$$\vec{E}_f = \frac{Q}{4\pi \epsilon} \left( \frac{\vec{r}_Q^+}{\|\vec{r}_Q^+\|^3} - \frac{\vec{r}_Q^-}{\|\vec{r}_Q^-\|^3} \right),$$

(3.2)

where $\vec{r}_Q^+$ and $\vec{r}_Q^-$ are vectors originating from the point of interest to the electrodes $Q^+$ and $Q^-$, respectively. Without an object in the field, each sensor should see zero voltage across its pair of electrodes. When an object is placed in the field, the perturbation caused by the object can be measured in terms of voltages across all sensors based on Equations (3.1) and (3.2). These perturbations on the sensors form an array pattern. The array pattern changes according to the position of the object. Figure 3.2 shows the simulated array patterns for different object positions; an object at different locations produces distinct array patterns. The array patterns, or more precisely the relative shape of the patterns, can be used to estimate the locations of the object without the knowledge of its size and conductivity.

One can approach this localization problem using a generalized beamforming technique, which means scanning all possible positions and identifying a position that maximizes a likelihood function. In our recent work [38], we use Capon’s beamforming technique for a similar problem in which we map the location of a vibrating object in water sensed by an array of underwater fluid-flow sensors. However, locating multiple objects requires that their signals be uncorrelated, which is not possible when all signals are induced by a common active source. In the next section, we propose a new beamforming technique that turns the localization problem into an overcomplete signal representation problem.

29
Figure 3.2: Array patterns in simulation of a weakly electric sensing system as shown in Figure 3.1 with 25 sensors, \( d = 50 \text{ mm} \), \( s = 8 \text{ mm} \). For each pattern, the object stays 70 mm away from the array and in front of one specified sensor.

3.2 Sparse Beamforming

In the adaptive beamforming approach in Chapter 2, we scan through the sampling grid of all source locations of interest and then compute the output power of the beamformer that focuses on a source location under inspection while minimizing the impact from sources in other locations. Since the impact from sources which are not in the focus is minimized, the adaptive beamforming technique provides a much sharper spectral-like image than the conventional beamforming approach. However, some level of power from other locations still leaks into and contributes to the output power of the beamformer (which is focusing on a particular location). Moreover, the adaptive beamformer cannot separate two strongly correlated sources.
The concept of sparse beamforming naturally emerges as the result of both the recent development of sparse signal representation [39, 40] and the observation that the signal received by a sensor array, if ignoring the additive noise, is actually the linear combination of component signals generated from each individual source. Therefore, if we scan all possible source locations and form an overcomplete dictionary of all expected array outputs of each location, then the problem of source location can be cast into the problem of decomposing a received signal into an optimal superposition of dictionary elements. There are some conditions for this approach to recover correct source locations. One of the conditions is the sparse representation of the signal, i.e., the number of components or the number of sources, is small in comparison to the number of dictionary elements.

Mathematically, consider a discrete 2D localization problem in which we sample the plane on a 2D grid with \( N \) points. For each position of the object on the grid, we compute the expected array pattern using the model in Equation (3.1). For the \( n \)th point in the grid, we denote the expected array pattern as \( \mathbf{a}(n) = [a_1(n), a_2(n), \ldots, a_K(n)]^T \), where \( K \) is the number of sensors in the array. Scanning all \( N \) points on the grid, we form a \( K \times N \) matrix

\[
\mathbf{A} = [\mathbf{a}(1), \mathbf{a}(2), \ldots, \mathbf{a}(N)].
\]

An object located at an unknown point on the grid induces a pattern \( \mathbf{b} = [b_1, b_2, \ldots, b_K]^T \). The vector \( \mathbf{b} \) must be equivalent to some column \( i \) in the matrix \( \mathbf{A} \). With multiple small objects that do not significantly distort the overall field, this problem reduces to solving \( \mathbf{A} \mathbf{x} = \mathbf{b} \), where \( \mathbf{x} \) is a length-
$N$ vector with all zero elements except the elements corresponding to the object locations. In most cases, $N > K$ and the equation $Ax = b$ is a highly underdetermined linear system of equations. This system has infinitely many solutions but under the condition to maximize sparsity of the solution, a unique solution can be found. Many algorithms have been proposed to solve this problem [39]. In this thesis, we cast our problem as an $\ell_1$-minimization problem

$$\min \|x\|_1, \quad \text{subject to} \quad Ax = b,$$

which can be effectively solved via a linear program [41]. In the next two sections, we demonstrate the performance of this beamforming method in simulations and especially in experiments where a real data vector $b$ works with a modelled matrix $A$.

### 3.3 Simulations

To test the idea of sparse beamforming for underwater electrolocation, we simulate a weakly electric sensing system as shown in Figure 3.1 with an array of $K = 25$ sensors. The spacing between sensors is $s = 8$ (mm) and the distances from the dipole’s electrode to the nearest sensor are $d = 50$ (mm). All other parameters such as the dipole’s charges, the conductivity of water and of the object, and the radius of the object in Equation (3.1) can be combined as a constant factor. This factor will be cancelled out during normalization of array patterns $a(n)$, so those parameters can take any values.

The first step is to form a 2D grid of probing points and then build the matrix $A$ as a dictionary of all array patterns for each point on the grid. Note that
Figure 3.3: (A) Image of mapping an ideal simulated object at $x = 125$ mm and $y = 50$ mm. (B) Image of mapping a simulated object with additive noise at $x = 125$ mm and $y = 50$ mm. (C) Image of mapping two ideal simulated objects at $x_1 = 100$ mm, $y_1 = 50$ mm and $x_2 = 150$ mm, $y_2 = 50$ mm.

Each array pattern is normalized before forming a column in the matrix $A$. Next, we select one array pattern $b$ corresponding to a point of interest on the grid. This pattern is used as the measurement from the sensor array to estimate the position of the object. To solve the $\ell_1$-minimization problem

$$\min \|x\|_1, \quad \text{subject to} \quad Ax = b,$$

we use the primal-dual interior point method from the $\ell1magic$ package [42]. Figure 3.3 shows the results of imaging a single simulated object located at position $x = 125$ mm and $y = 50$ mm. We actually analyze two cases of sensor measurement with and without noise. For the case of no noise in the sensor output, Figure 3.3(A) displays a single sharp peak exactly at the original position of the object. In the presence of noise (SNR $\approx 20$ dB) in sensor outputs, Figure 3.3(B) shows a lower peak at the original position and a few small peaks close to the sensor array. Furthermore, we also test the capability of the beamforming technique to localize multiple sources. Figure 3.3(C)
shows two separate peaks corresponding to exact locations of two simulated objects at $x_1 = 100$ mm, $y_1 = 50$ mm and $x_2 = 150$ mm, $y_2 = 50$ mm.

The simulation results demonstrate the potential of the sparse beamforming technique. In the next section, we demonstrate the effectiveness of this technique in an experimental setting.

### 3.4 Experiments and Results

#### 3.4.1 Experimental Setup

A weakly electric sensing system was built according to the design in Figure 3.1. Using LEGO components, we set up a small rack to attach 7 pairs of electrodes serving as 7 sensors. Another pair of electrodes forms a dipole to generate an electric field. The legs of the 16 electrodes are placed in a basin of salt water as shown in Figure 3.4A. The space between the sensors is $s = 32$ mm. The gap between 2 electrodes of a sensor is 25.4 mm. The distance from the dipole electrodes to the nearest sensors is $d = 50$ mm. A square waveform of 1 kHz with magnitude of 5 V drives the dipole to generate an electric field. Those configuration parameters are used to generate the dictionary matrix $A$ as in the simulation (Section 3.3). The only difference from the simulation is that we measure the real sensors’ outputs when an object is brought near the sensor array.

#### 3.4.2 Calibration

One challenge of working with a real sensor array is that we must first figure out the gain of each sensor to calibrate the array. For our sensor array, the
Figure 3.4: (A) Experimental setup. (B) Image of successfully mapping a plastic ball at about 45 mm in front of sensor 5. (C) Image of mapping simultaneously two plastic ball objects at about 45 mm in range and between Sensors 1 & 2 and Sensors 5 & 6.

calibration process is performed by moving a pair of test electrodes with a fixed spacing and voltage over each sensor. When the test electrodes move in front of a sensor, the output of that sensor is recorded and used to work out the gain of that sensor in the array.

In the model, we assume that the sensors’ legs are symmetrical on both sides of the dipole’s axis so that the voltages across each sensor are perfectly zero. It is not the case for the real experimental setting. In fact, we first carefully tune each sensor to get the lowest possible output voltage before putting an object in and recording the perturbations.

3.4.3 Single Object Mapping

In this experiment, we put a plastic ball into the space in front of the sensor array. We recorded the sensor outputs for several positions in front of some sensors or in the gap between them. Figure 3.4(B) shows the results of mapping the location of the ball when it lies about 45 to 50 mm in front of sensor 5. In those plots, we can clearly see one large peak very close to the
expected location of the ball.

### 3.4.4 Multiple Objects Mapping

We then tested our sparse beamforming technique for multiple sources. Note that signals from multiple sources in the weakly electric sensing system are highly correlated as they are induced from the same dipole source, so conventional beamforming techniques do not work for this case. In the experiment, we put two similar plastic balls in two positions roughly between sensors 1 and 2 and between sensors 5 and 6. Both balls were at a range of about 45 to 50 mm away from the sensor array. The results displayed in Figure 3.4 clearly show that we can separate two sources and map the locations accurately.

### 3.5 Evaluation of Array Response Model for Multiple Objects

In this chapter, the sparse beamforming technique is applicable for the electrolocation system with multiple sources because it is assumed that the total array response from each object is the summation of all individual responses from each object. There is a question whether the accuracy of this superposition assumption will be affected if any two objects are close to each other, because they will interact and alter the surrounding electric field. In this section, we investigate the question and quantify how much the superposition assumption is affected.

Recall that when a sphere of radius $a$ with conductivity of $\sigma_{\text{object}}$ is placed
in water with conductivity of $\sigma_{\text{water}}$ at the point $\vec{r}$ with electric field $\vec{E}_f$, the potential change is derived in Equation (3.1) as

$$\Delta \phi(\vec{r}) = \frac{q^3 \vec{E}_f \cdot \vec{r}}{||\vec{r}||^3} \left( \frac{\sigma_{\text{object}} - \sigma_{\text{water}}}{\sigma_{\text{object}} + 2\sigma_{\text{water}}} \right).$$

If we follow the relation between the electric field and electric potential of a point charge $q$ at a distance $\vec{r}$ where the electric field is

$$\vec{E} = \frac{1}{4\pi\varepsilon_0} \frac{q}{r^3} \vec{r},$$

and the electric potential is

$$\phi_E = \frac{1}{4\pi\varepsilon_0} \frac{q}{r},$$

then the change in electric field is

$$\Delta \vec{E}_f(\vec{r}) = \Delta \phi(\vec{r}) \frac{\vec{r}}{||\vec{r}||^2} = \frac{q^3 \vec{E}_f}{||\vec{r}||^4} \left( \frac{\sigma_{\text{object}} - \sigma_{\text{water}}}{\sigma_{\text{object}} + 2\sigma_{\text{water}}} \right).$$

(3.3)

In order to evaluate how much the superposition assumption will be affected when multiple objects interact with each other, we need to compute the induced local electric fields surrounding those objects. Figure 3.5 shows the discrepancy between two cases with and without multi-object interaction. For the case of multi-object interaction, we compute the induced local electric field of one object due to another object. The induced local electric field of each object is then used to compute the change in potential detected by the sensor. The discrepancy percentage is the ratio of the difference between the two cases (with and without object interaction) divided by the case of without object interaction. As can be seen, the discrepancy percentage in-
creases rapidly as the gap closes in. However, the discrepancy percentages are relatively small. So we can claim that the superposition assumption is generally valid.

Figure 3.5: The discrepancy percentage of sensor measurement between the cases of with and without computing multi-object interaction. The radius of each object is 15 mm and the two objects are parallel to the sensor array on the x axis at a distance from the array of 45 mm
Chapter 4

Projected Gradient Algorithm for Sparse Beamforming

In Section 3.2, the sparse beamforming framework casts the source localization problem into a sparse signal representation problem with an overcomplete dictionary. Both [43] and [44] show that this problem, in turn, can be solved via convex relaxation as an $\ell_1$-minimization problem. There are various optimization tools to solve the $\ell_1$-minimization problem. One may use the modern linear programming algorithms such as simplex methods [45], interior-point methods [46], or standard convex optimization algorithms. In this chapter, we present a new $\ell_1$-minimization algorithm adapted for a sparse beamforming problem with the constraint that the global optimal solution has no negative non-zero elements.

4.1 Description of the Algorithm

The algorithm emerges from three intuitive observations:

- The $\ell_1$-minimization algorithm actually tries to search for the intersection point when the $\ell_1$ ball is expanding and first touches the constraint set $Ax = b$. If $x_0$ is the $\ell_2$ optimal solution of $Ax = b$, which is also the projection of the origin onto the constraint set, then in many cases $x_0$ and the intersection point lie on the same face of the $\ell_1$ ball.

- Since the constraint set is linear and the $\ell_1$ norm is a piecewise linear function over the same-sign region of $x$, the projected gradient stays
unchanged along with the signs of $\mathbf{x}$. One can then drive $\|\mathbf{x}\|$ to a lower value until one element hits zero. Graphically, from an initial point in the constraint set, we can move along the projected gradient direction until we hit the edge of the polytope containing all points of same signs as the initial point.

- If we compress all non-zero elements of $\mathbf{x}$ into a new vector $\overline{\mathbf{x}}$ and also form a matrix $\overline{\mathbf{A}}$ of those columns in $\mathbf{A}$ corresponding to a non-zero element in $\overline{\mathbf{x}}$, then $\mathbf{b} = \mathbf{Ax} = \overline{\mathbf{Ax}}$ and $\|\mathbf{x}\|_1 = \|\overline{\mathbf{x}}\|_1$. Therefore the search for an $\ell_1$ optimal solution of $\mathbf{Ax} = \mathbf{b}$ can be repeated with $\overline{\mathbf{Ax}} = \mathbf{b}$.

Based on these observations, we come up with a new algorithm that starts with an initial constraint-satisfying solution (we use the $\ell_2$ solution), and uses the projected gradient method to drive one or more elements to zero; it then removes the columns in $\mathbf{A}$ corresponding to zero elements in the current solution to repeat the whole process. This iterative process will exit when the projected gradient equals zero. This algorithm has two obvious trends. First, it always reduces the $\ell_1$ norm of the solution. Second, it increases the sparsity of the solution (decreases the $\ell_0$ norm). However, it is not guaranteed to converge to the global optimal solution, as confirmed by numerical experimentation. Note that, during the whole process, the algorithm only drives more elements to zero but cannot switch the sign of the elements. That means the algorithm drives the current solution to an edge (with fewer non-zero elements) of the polytope containing all points of same signs as the initial points. Therefore, it will not produce the optimal solution if it already started with a wrong initial sign.
In order to guarantee the convergence to the optimal solution, we introduce an optimality test at the end of the Projected Gradient Algorithm. Exploiting the fact that the projected gradient becomes zero if the dictionary matrix $A$ is shrunk into a square matrix, the solution on exiting the Projected Gradient Algorithm, in most cases, is a basic solution of the simplex method. As a result, we can use the optimality test in the simplex method to check if an optimal solution has been reached. Furthermore, if the converged solution fails the optimality test, we can exploit the result from the simplex test to readmit new bases into the dictionary matrix and insert corresponding elements into the converged solution to form a new initial solution and re-run the Projected Gradient Algorithm. The bases are readmitted into the dictionary matrix in a way that will produce a new initial solution with lower or equal $\ell_1$ norm than the converged solution.

In summary, the overall algorithm loops around two processes, i.e., the Projected Gradient Algorithm, and the Optimality Test and Basis Readmission via the Simplex Method. The first process uses the Projected Gradient Algorithm to minimize the $\ell_1$ norm as well as reduce the number of active bases until a basic solution is reached. The second process checks if the basic solution as the output of the first process is a globally optimal solution by using the optimality test from the simplex method. If global convergence has not been achieved, it then includes back a group of bases and computes a new initial solution with a lower $\ell_1$ norm and feeds it back to the first process to minimize the $\ell_1$ norm further.

The mathematical summary of the two processes is given in Algorithms 1 and 2:
Algorithm 1: Projected Gradient Algorithm

1: **input:** initial solution $x_0$ where $x_0$ belongs to the constraint set i.e., $b = Ax_0$.
2: **repeat**
3: (At iteration $n$)
   - Compute the support index set: $\Lambda_n = \text{supp}(x_n) \equiv \{i \in \Omega : x_n(i) \neq 0\}$;
   - Remove zero elements in $x_n$: $x_n = (x_n)_{\Lambda_n}$;
   - Compute gradient: $d_n = \text{sign}(x_n)$
   - Compute projected gradient: $g_n = (I - A_{\Lambda_n}^+ A_{\Lambda_n})d_n$;
4: **if** $g_n \neq 0$ **then**
5: $x_{n+1} = x_n - \alpha z_n$ where $\alpha = \min \left\{ \frac{x_n(i)}{g_n(i)} : \frac{x_n(i)}{g_n(i)} > 0 \right\}$ and $z_n = g_n$;
6: **end if**
7: **until** $g_n = 0$
8: **exit:** Send the output to Algorithm 2 to check if a globally optimal solution has been reached.

Algorithm 2: Optimality Test and Basis Readmission via the Simplex Method

1: **Step 0:** Form a tableau as in Table 4.1 corresponding to the converged solution of the Projected Gradient Algorithm.
2: **Step 1:** If each relative cost $r_j \geq 0$, stop; the converged solution is globally optimal.
3: **Step 2:** Select all indices $q$ such that $r_q < 0$ to readmit them in the support index set $\Lambda$ of a new initial solution for another run of the Projected Gradient Algorithm.
4: **Step 3:** Find the new initial solution, which has lower $\ell_1$-norm than the converged solution, by computing $\epsilon > 0$ to set as the value of all newly added non-basic variables in the support index set $\Lambda$ so that the signs of all basic variables are preserved.
5: **Step 4:** Re-run the Projected Gradient Algorithm on the new initial solution.
4.2 Optimality Test and Basis Readmission via the Simplex Method

The goal of this part of the algorithm is to check whether the solution on exiting from the Projected Gradient Algorithm is the optimal solution. If it is not optimal (i.e., not the lowest $\ell_1$ norm), then a new initial solution with lower $\ell_1$ norm will be created to submit back in the Projected Gradient Algorithm to reduce the $\ell_1$ norm further.

Since the Projected Gradient Algorithm will surely terminate when the compressed dictionary matrix $\overline{A}_n$ becomes a full-rank square matrix, there are two cases for the final solution on exiting the Projected Gradient Algorithm. The first and typical case is when $\overline{A}_n$ is a square matrix. Then the supporting index set has a size of $K$. The final solution is actually a basic solution for the simplex method. The second case is when $\overline{A}_n$ is still a fat matrix; i.e., the number of columns is larger than the number of rows.

- **Case 1: Exit with a basic solution**

  With this basic solution, we can set up a simplex test to check if it is optimal. Note that the simplex method requires that all solutions $x$ are non-negative. So if we denote $x^*$ as the final solution on exiting the Projected Gradient Algorithm, then the simplex method will test optimality under the cost $c = \text{sign}(x^*)$ with a new matrix

  $$A^* = [A(1)c(1), A(2)c(2), \ldots, A(N)c(N)]$$
where the *sign* function is defined as below:

$$\text{sign}(a) = \begin{cases} 
1 & \text{if } a \geq 0 \\
-1 & \text{else.}
\end{cases}$$

Without loss of generality, we can assume the supporting index set for \(x^*\) contains the first \(K\) indices, i.e., \([1, 2, \ldots, K]\). Then we canonicalize the first \(K\) columns of the matrix \([A^*|b]\) using the method of Gaussian elimination. Table 4.1 shows the result of that process. According to the simplex method \([45, 47]\), we can perform the test for optimality of the basic solution by constructing the sum of products

$$z_j = y_{1j}c(1) + y_{2j}c(2) + \ldots + y_{Kj}c(K)$$

for all \(K + 1 \leq j \leq N\), then compare \(z_j\) against \(c(j)\). If \((c(j) - z_j) \geq 0\) for all \(K + 1 \leq j \leq N\), we can conclude the solution \(x^*\) is optimal under the cost-vector \(c\). That means

$$\forall x \text{ such that } Ax = b \text{ and } \text{sign}(x) = \text{sign}(x^*) \equiv c \text{ then } c^T x^* \leq c^T x.$$
negative conditions.

If the final solution $x^*$ fails the optimality test, we can construct a new initial solution $x^\dagger$ for the Projected Gradient Algorithm with lower $\ell_1$ norm than $x^*$. The approach to constructing $x^\dagger$ is to readmit into the supporting index set (alongside the existing $K$ indices from $x^*$) all indices corresponding to the failed test; i.e., $(c(j) - z_j) < 0$ for all $K + 1 \leq j \leq N$, and then increase the value of elements corresponding to newly readmitted indices from 0 in $x^*$ to a small value $\epsilon > 0$ so that the amount of re-adjustment in the value of elements originally in $x^*$ is small enough to not invert their signs. This can be done by noting that the column of $b$ in Table 4.1 is actually $|x^*|$. Therefore, $\alpha$ should be selected so that if subtracting from the column $b$ a sum of all columns $a_j^*$ times $\alpha$, then the result is still a positive vector. With this approach, we can easily show that $c^T x^\dagger \leq c^T x^*$ and equivalently $\|x^\dagger\|_1 \leq \|x^*\|_1$. The new solution $x^\dagger$ can then be used as an initial solution for the next run of the Projected Gradient Algorithm.

- **Case 2: Exit with a non-basic solution**

  In this case, we can conclude that the solution is not optimal. The next task is to construct a new solution with lower $\ell_1$-norm to feed back to the Projected Gradient Algorithm. This can be done by following the approach in Case 1 except that we canonicalize the matrix $A^*$ with $K$ indices as the subset of the supporting index set and ignore the other indices when checking the condition to readmit new bases. Also, the readjustment only applies to those elements corresponding to the $K$ selected indices.
4.3 Convergence Analysis

In this part, we will show that the whole process including the two algorithms presented above guarantees convergence to a globally optimal solution. In Section 3.2, we mentioned the conditions to recover the correct solution for the source localization problem using the sparse signal representation framework. Here, we will present the recovery conditions in mathematical terms and relate them to the convergence analysis.

- **Unique optimum point:** Fuchs [48] shows the condition for the unique optimum point to guarantee that this point will be recovered by any $\ell_1$ minimization algorithm. In this thesis, we assume the solution for the source localization meets the condition for a unique optimum point. That means if $x_0$ is the expected solution, then $x^o$ is not only sparse but $\|x^o\|_1 < \|x\|_1$ for all $Ax = b$.

- **Non-negative solution:** For a beamforming application, we can assume the optimal solution $x^o$ has all elements non-negative.

With the conditions for the optimal solution $x^o$, we now show that the whole process will guarantee recovery of $x^o$. It is clear that the projected gradient algorithm part reduces the $\ell_1$ norm as well as the $\ell_0$ norm at every step. The key remaining issue is that the optimality test can only verify that the converged solution is optimal in terms of a specific cost vector, which in this case is the sign of the converged solution, but not guarantee optimality in term of the $\ell_1$ norm. In other words, if we denote $x^*$ as the converged solution, then the optimality test can only verify if

$$\forall x \text{ such that } Ax = b \text{ and } \text{sign}(x) = \text{sign}(x^*) \equiv c \text{ then } c^T x^* \leq c^T x, \quad (4.1)$$
We need to show that if \( x^* \) satisfies Condition 4.1, then \( \|x^*\|_1 = \|x^o\|_1 \) and we can conclude that \( x^* \equiv x^o \) under the unique optimum point condition.

Note that the optimal solution \( x^o \) has all elements non-negative; i.e., \( x^o(i) \geq 0 \) \( \forall i \). So if \( x^* \) is also non-negative then we have

\[
c = \text{sign}(x^*) = \text{sign}(x^o) = [1, 1 \ldots 1]^T
\]

\[
\Rightarrow c^T x^* \leq c^T x^o
\]

\[
\Rightarrow \|x^*\|_1 \leq \|x^o\|_1
\]

but \( x^o \) is \( \ell_1 \)-norm-optimal \( \Rightarrow \|x^*\|_1 \geq \|x^o\|_1 \)

\[
\Rightarrow \|x^*\|_1 = \|x^o\|_1 \quad \square
\]

If \( x^* \) is not non-negative; i.e., \( \exists k : x^*(k) < 0 \), then consider three cases.

- **Case 1:** \( \Lambda(x^*) \cap \Lambda(x^o) = \emptyset \)

  We then still have \( c^T x^o = \|x^o\|_1 \) for \( c = \text{sign}(x^*) \). Following the same argument above we have \( \|x^*\|_1 = \|x^o\|_1 \).

- **Case 2:** \( \Lambda(x^*) \cap \Lambda(x^o) = S \neq \emptyset \) but \( x^* \) is non-negative in \( S \); i.e., \( x^*(i) \geq 0 \) \( \forall i \in S \).

  In this case, the argument in Case 1 still valid.

- **Case 3:** \( \Lambda(x^*) \cap \Lambda(x^o) = S \neq \emptyset \) but \( x^* \) is NOT non-negative in \( S \), i.e., \( \exists k \in S : x^*(k) < 0 \).

  For this case, we can find \( \lambda \in (0,1) \) such that
\[ x^+ = \lambda x^* + (1 - \lambda) x^o \] satisfies \( \text{sign}(x^+) = c \). Then

\[
\begin{align*}
    c^T x^+ &= \lambda c^T x^* + (1 - \lambda) c^T x^o \\
    &= \lambda c^T x^* + (1 - \lambda) \| x^o \|_1 \\
    &\leq \lambda c^T x^* + (1 - \lambda) \| x^* \|_1 \\
    &= c^T x^*
\end{align*}
\]

This is actually a contradiction to the Condition 4.1.

To sum up, the new algorithm starts from an initial solution; it then uses the Projected Gradient Algorithm to produce a new sequence of solutions with decreasing \( \ell_1 \) norm. After a finite number of iterations, the Projected Gradient Algorithm will exit with a final solution. This solution is then checked by the Optimality Test to see if it is actually the optimal solution. If not, a new initial solution with slightly lower \( \ell_1 \) norm than the final solution can be created and re-submitted into the projected gradient algorithm. So the whole process will always drive the solution to lower \( \ell_1 \) norm until the optimality test is successful. Therefore, the algorithm guarantees a convergence to the optimal solution.

### 4.4 Performance Analysis

In this part, we evaluate the performance of our algorithm against the frequently used \( \ell_1 \)-minimization package, the \textit{l1magic} [42]. The \textit{l1magic} formulates the \( \ell_1 \)-minimization problem into a linear programming problem and uses the primal-dual interior point method to solve it.

In terms of speed, the \textit{l1magic} is superior than our technique for large sys-
Figure 4.1: CPU time of the l1magic and the projected gradient algorithm with number of inner iterations for a system of $N = 100$ bases, $M = 20$ measurements, and sparsity $K = 2$.

4.5 Projected Subgradient Algorithm

In this section, we modify the Projected Gradient Algorithm to work with the subgradient and so-called projected subgradient algorithm. A subgradient $g$
of a function $f$ at $x$ is defined as

$$f(y) \geq f(x) + g^T(y - x) \text{ for all } y.$$  

When $f$ is convex and differentiable at $x$, then the only subgradient $g = \nabla f(x)$. In our case of an $\ell_1$-norm cost function, the gradient $d_n = \text{sign}(\mathbf{x}_n)$ if $\mathbf{x}_n$ has no zero entry. If $\mathbf{x}_n$ has one or more zero entry, e.g. $\mathbf{x}_n(i) = 0$, then the gradient is no longer $d_n = \text{sign}(\mathbf{x}_n)$. Instead, we have the subgradient $d_{n_{\text{sub}}}$ with all the same entries as $d_n$, except $d_{n_{\text{sub}}}(i)$ can take any value from -1 to 1.

The main concept of the projected subgradient is that when we reached Step 5 of Algorithm 1, instead of removing the zero entry $i$ from $\mathbf{x}_n$, we find the subgradient $d_{n_{\text{sub}}}$ of $\mathbf{x}_n$ so that the projected subgradient $g_{n_{\text{sub}}} = (I - A_{\Lambda_n}^+ A_{\Lambda_n})d_{n_{\text{sub}}}$ has $g_{n_{\text{sub}}}(i) = 0$. By doing so, we can try to drive more entries toward zero while still preserving existing zero entries of $\mathbf{x}_n$. However, projected subgradients are not guaranteed to lower the $\ell_1$ norm even though it happens most of the time in the case of the $\ell_1$-norm cost function. Therefore, we safeguard the property of decreasing the $\ell_1$ norm by stopping the projected subgradient loop and exiting to the main loop (i.e, the projected gradient loop) when we cannot (1) find more subgradients or (2) reduce the $\ell_1$ norm further. One advantage of doing subgradient projection is that the computation of the subgradient $d_{n_{\text{sub}}}$ to satisfy $g_{n_{\text{sub}}}(i) = 0$ involves inverting a small matrix (of the same size as the current number of zero entries in $\mathbf{x}_n$). The computation of the projected gradient involves inverting a larger matrix.

The mathematical summary of the Projected Subgradient Algorithm is given
in Algorithm 3.

Algorithm 3: Projected Subgradient Algorithm

1: input: initial solution \( x_0 \) where \( x_0 \) belongs to the constraint set, i.e., \( b = Ax_0 \).
2: repeat
3: (At iteration \( n \))
   - Compute the support index set: \( \Lambda_n = \text{supp}(x_n) \equiv \{ i \in \Omega : x_n(i) \neq 0 \} \);
   - Remove zero elements in \( x_n \): \( x_n = (x_n)_{\Lambda_n} \);
   - Compute gradient: \( d_n = \text{sign}(x_n) \);
   - Compute projected gradient: \( g_n = (I - A_{\Lambda_n}^+ A_{\Lambda_n})d_n \);
4: if \( g_n \neq 0 \) then
5: \( x_{n+1} = x_n - \alpha z_n \) where \( \alpha = \min \left\{ \frac{x_n(i)}{g_n(i)} : \frac{x_n(i)}{g_n(i)} > 0 \right\} \) and \( z_n = g_n \);
6: input: \( x_{\text{sub}0} = x_{n+1} \);
7: repeat
8: (At iteration \( k \) the subgradient inner loop)
   - Compute the index set of zero entries: \( \Phi_k \equiv \{ i \in \Omega : x_{\text{sub}k}(i) = 0 \} \);
   - Compute the subgradient \( d_{\text{sub}k} \) so that the projected gradient \( g_{\text{sub}k} = (I - A_{\Lambda_n}^+ A_{\Lambda_n})d_{\text{sub}k} \) has \( g_{\text{sub}k}(i) = 0 \ \forall i \in \Phi_k \);
9: if \( g_n \neq 0 \) then
10: \( x_{\text{sub}k+1} = x_{\text{sub}k} - \beta g_{\text{sub}k} \) where \( \beta = \min \left\{ \frac{x_{\text{sub}k}(i)}{g_{\text{sub}k}(i)} : \frac{x_{\text{sub}k}(i)}{g_{\text{sub}k}(i)} > 0 \right\} \);
11: end if
12: until \( d_{\text{sub}k} \) does not exist
13: exit: \( x_{n+1} = x_{\text{sub}k} \);
14: end if
15: until \( g_n = 0 \)
16: exit: Send the output to Algorithm 2 to check if a globally optimal solution has been reached.

Figure 4.2 shows the performance of three algorithms i.e., the \textit{l1magic}, the Projected Gradient Algorithm and the Projected Subgradient Algorithm in terms of CPU time. We realize that the Projected Subgradient Algorithm does not show better performance than the Projected Gradient Algorithm. The reason could be the overhead in switching between two loops; i.e., the outer projected gradient loop and the inner projected subgradient loop.
Figure 4.2: CPU time of the \textit{l1magic}, the Projected Gradient Algorithm and the Projected Subgradient Algorithm with number of inner iterations for a system of $N = 100$ bases, $M = 20$ measurements, and sparsity $K = 2$. 
Chapter 5

Source Localization via the Reassignment Method

In the two previous chapters, we use beamforming techniques to solve the source localization problem. Our objective is to produce a 2D or 3D map indicating the likelihood of a source’s presence in a region. This kind of map is similar to the energy distribution of a time-frequency representation in non-stationary signal analysis. Specifically, a spectrogram of a signal [49] shows the energy distribution of the signal at a certain frequency and time. This is done by moving a short time-window along the signal and then taking a Fourier transform of that windowed signal. This technique is called short-time Fourier transform (STFT).

More recently, the continuous wavelet transform (CWT) has been discovered and has become an alternative tool for analyzing non-stationary signals. Instead of creating a time-frequency representation by shifts in time and in frequency, wavelet transform replaces frequency shifts by dilations and generates a time-scale representation of signals such as the scalogram [50]. In particular, the analyzing waveforms in the STFT can be viewed as a time-shift (by $t$) of a baseband signal modulated at a certain frequency $w$ (i.e., $h(t - \tau)e^{-j\omega\tau}$) while the analyzing waveforms in the CWT are the scaled version of a basic wavelet after shifting in time, i.e., $h(\frac{\tau t}{a})$.

Given that the goal of source localization is to produce an energy distri-
bution map similar to the spectrogram or scalogram, the question is how we can actually apply the concept of the wavelet transform to achieve this goal. Note that in previous chapters our approach is to scan through the plane and form a dictionary of array patterns assuming a source presenting at that location \((b, d)\) in order to compute the value for the likelihood of a source’s presence at that location. Supposing a sensor on the array is located on the \(x\) axis at \((s, 0)\), then the signal model is usually the function of the relative distance from a source to the sensor, i.e., \(f(s - b, d)\). In some cases, we can simplify the basic model into a form of \(f(s-b, d)\). Therefore, we can view the array patterns as wavelets and apply wavelet transform techniques to analyse the sensor array measurements. In fact, work by Curcic-Blake and van Netten [23] already followed this approach by formulating their source localization with the fish’s lateral line canal as a continuous wavelet transform solution.

However, maps generated by the wavelet transform are not focused, nor are the spectrograms and scalograms. In order to increase the focus level, the reassignment method, which was discovered by Kodera et al. [8], can be used as a post-processing technique. By exploiting the phase information of Fourier coefficients, which is thrown away in the spectrogram, Kodera et al. have shown that the readability of a spectrogram can be improved significantly by reassigning an energy distribution value from the geometric center to the center of gravity of the distribution mass it represents. Later, a series of works [51, 52, 53] has extended the reassignment technique to more general classes of distributions including the scalogram. Equipped with results from the extension, we discover here that the application of the reassignment method is not only limited to non-stationary signal analysis but can also be extended to cover the source localization problem.
In this chapter, we apply the reassignment technique to solve the source localization problems with the fish lateral line from Chapter 2 and also the weakly electric fish sense from Chapter 3. First, we provide some background on the reassignment method in the context of improving spectrograms and scalograms. Next, we present the application of this technique for source localization with the fish lateral line in Section 5.2.1 and with the weakly electric fish sense in Section 5.2.2. Last, we open our discussion for further extension to a more general source localization problem.

5.1 Reassignment Principle

The reassignment method was first proposed by Kodera et al. in 1976 [8] to improve the readability and sharpness of spectrograms. In time-frequency (TF) analysis, there is always a trade-off between the localization and interference. Reassignment is a post-processing technique to overcome this trade-off. By taking advantage of phase information in the STFT, Kodera et al. argued that a value of the spectrogram at point \((t, w)\) should be reassigned to another point \((t', w')\) which is the center of gravity of the distribution mass it represents. Although this technique promises good improvement, the implementation is not computationally efficient. Only after the publication of Auger and Flandrin’s work in 1995 [51, 53, 54] (about 15 years after Kodera’s work), does the reassignment method resurface as an attractive post-processing tool for time-varying signal analysis. In their work, Auger and Flandrin reformulate the derivation of reassigned time and frequency so that the computation is straightforward. More importantly, they generalize the time-frequency reassignment to any Cohan’s class of distributions and
then extend further with a concept of time-scale reassignment which is based on continuous Wavelet transform. In the next two sections, we review some details of Auger and Flandrin’s results for the spectrogram and scalogram to provide a complete background of the reassignment method before discussing its application in source localization.

5.1.1 Reassignment for Spectrogram

In general, a spectrogram of a signal $x(t)$ represents the energy distribution of the signal over time and frequency. The spectrogram is defined as the squared magnitude of the short-time Fourier transform (STFT):

$$S_h(x; t, w) := |\text{STFT}_h(x; t, w)|^2$$

where the STFT is the Fourier transform of the signal $x(t)$ after going through a time-shift analysis window $h(t)$

$$\text{STFT}_h(x; t, w) := \int_{-\infty}^{+\infty} x(\tau)h^*(t - \tau)e^{-j\nu \tau}d\tau.$$  

However, the spectrogram can be also computed as a 2D convolution of the Wigner-Ville (WV) distribution of the signal $x(t)$ and the WV distribution of analyzing the window $h(t)$, i.e,

$$S_h(x; t, w) = \frac{1}{2\pi} \int \int_{-\infty}^{+\infty} \text{WV}(x; \tau, \nu)\text{WV}(h; t - \tau, w - \nu)d\tau d\nu \quad (5.1)$$

$$= \frac{1}{2\pi} \int \int_{-\infty}^{+\infty} \text{WV}(x; t - \tau, w - \nu)\text{WV}(h; \tau, \nu)d\tau d\nu \quad (5.2)$$

56
where the WV distribution of \(x(t)\) is defined as

\[
WV(x; t, w) := \int_{-\infty}^{+\infty} x(t + \tau/2)x^*(t - \tau/2)e^{-j\tau w}d\tau.
\]

Proof of this relation can be found in Appendix A.1.1.

Equation (5.1) reveals that the time-frequency value \(S_h(x; t, w)\) at any point \((t, w)\) is a sum of the weighted Wigner-Ville distribution values at the neighboring points \((t - \tau, w - \nu)\). Therefore, the average value \(S_h(x; t, w)\) should represent the signal energy at the center of gravity \((\hat{t}, \hat{w})\) instead of being at the center of the domain \((t, w)\). The coordinates of the center of gravity can be computed as below:

\[
\hat{t}(x; t, w) := \frac{1}{2\pi S_h(x; t, w)} \int \int_{-\infty}^{+\infty} (t - \tau)WV(h; \tau, \nu)WV(x; t - \tau, w - \nu)d\tau d\nu \quad (5.3)
\]

\[
= t - \frac{1}{2\pi S_h(x; t, w)} \int \int_{-\infty}^{+\infty} \tau WV(h; \tau, \nu)WV(x; t - \tau, w - \nu)d\tau d\nu (5.4)
\]

\[
\hat{w}(x; t, w) := \frac{1}{2\pi S_h(x; t, w)} \int \int_{-\infty}^{+\infty} (w - \nu)WV(h; \tau, \nu)WV(x; t - \tau, w - \nu)d\tau d\nu \quad (5.5)
\]

\[
= w - \frac{1}{2\pi S_h(x; t, w)} \int \int_{-\infty}^{+\infty} \nu WV(h; \tau, \nu)WV(x; t - \tau, w - \nu)d\tau d\nu (5.6)
\]

Although Equation (5.4) and (5.6) clearly capture the concept of the reassignment method, its computation is not efficient. It is the contribution of Auger and Flandrin to reformulate those equations as below:

\[
\hat{t}(x; t, w) := t - Re \left\{ \frac{\text{STFT}_h(x; t, w)\text{STFT}^*_h(x; t, w)}{|\text{STFT}^*_h(x; t, w)|^2} \right\} \quad (5.7)
\]
\[ \hat{w}(x; t, w) := w + \text{Im} \left\{ \frac{\text{STFT}_{Dh}(x; t, w) \text{STFT}^*_h(x; t, w)}{|\text{STFT}^*_h(x; t, w)|^2} \right\} \] (5.8)

with \( T h(t) := th(t) \) and \( D h(t) := (dh/dt)(t) \). The derivation of those results is given in A.1.2 of the appendix.

### 5.1.2 Reassignment for Scalogram

Recently, the wavelet transform (WT) has been introduced as an alternative analysis tool for nonstationary signals. Instead of using time-shift and modulated analyzing waveforms \( h(t - \tau)e^{-jw\tau} \) as in the STFT, the wavelet transform uses the time-shift and scaling analyzing waveforms \( \frac{1}{\sqrt{|a|}}h(\frac{\tau-t}{a}) \) as in Equation 5.9:

\[
\text{CWT}_h(x; t, a) = \frac{1}{\sqrt{|a|}} \int x(\tau)h^*(\frac{\tau-t}{a})d\tau. \] (5.9)

Then a scalogram is defined as a squared magnitude of the wavelet transform and represents the energy distribution of the signal over time \( t \) and scale \( a \)

\[ SC_h(x; t, a) = |\text{CWT}_h(x; t, a)|^2. \]

If we analyze a pure sine wave at frequency \( f \), the association between scale and frequency is

\[ a = \frac{f_0}{f} \]

where \( f_0 \) is the central frequency of the mother wavelet \( h(t) \) [55].

Similarly to the spectrogram, the scalogram can be decomposed as a 2D-
convolution of two Wigner-Ville distributions (A.1.3)

\[ SC_h(x; t, a) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \text{WV}(h; \frac{\tau - t}{a}, a\nu)\text{WV}(x; \tau, \nu) d\tau d\nu. \]  

(5.10)

Based on Equation (A.5), we can follow the same argument for reassignment of the spectrogram to derive the reassignment for time and scale of a scalogram [52]:

\[
\hat{t}(x; t, a) := \frac{1}{SC_h(x; t, w)} \int_{-\infty}^{+\infty} \tau \text{WV}(h; \frac{\tau - t}{a}, a\nu)\text{WV}(x; \tau, \nu) d\tau d\nu \frac{\nu}{2\pi} 
\]

(5.11)

\[
\hat{w}(x; t, a) := \frac{w_0}{\hat{a}(x; t, a)} = \frac{1}{SC_h(x; t, w)} \int_{-\infty}^{+\infty} \nu \text{WV}(h; \frac{\tau - t}{a}, a\nu)\text{WV}(x; \tau, \nu) d\tau d\nu \frac{\nu}{2\pi} 
\]

(5.12)

In [51], Auger and Flandrin again provided more straightforward formula of reassigned time and scale for a scalogram. The details of their derivation are given in A.1.4

\[
\hat{t}(x; t, w) = t + \text{Re} \left\{ \frac{a\text{CWT}_T h(x; t, a)\text{CWT}_T^* h(x; t, a)}{|\text{CWT}_h(x; t, a)|^2} \right\} 
\]

(5.13)

\[
\hat{w}(x; t, w) = \frac{w_0}{\hat{a}(x; t, a)} = -\text{Im} \left\{ \frac{\text{CWT}_D h(x; t, a)\text{CWT}_D^* h(x; t, a)}{a|\text{CWT}_h(x; t, a)|^2} \right\} 
\]

(5.14)

with \( T h(t) = th(t) \) and \( D h(t) = (dh/dt)(t) \).

5.2 Reassignment for Source Localization

The reassignment method has significantly improved the readability of spectrograms and scalograms. Its applications, however, are not limited only to
the domain of non-stationary signal analysis. In this section, we demonstrate that the reassignment method can enhance the focus of the source localization maps of systems in Chapters 2 and 3. There are three reasons why the concept of reassignment can be utilized for source localization:

- Maps of energy distribution like spectrograms and scalograms can be used for source localization.
- The array patterns in source localization can be formulated as analyzing wavelets in a scalogram.
- Reassignment can be applied on the energy distribution map in source localization in order to increase the focus of the map.

5.2.1 Reassignment for Fish Lateral Line

In Chapter 2, we have studied the use of artificial lateral lines for source localization. An array of sensors is constructed to capture the movement of water created by a vibrating sphere. Depending on the type of sensors, different models for sensor response are used to construct a map of distributions of the sources. The model we use in Chapter 2 is for hair-cell sensors. In this section, we will work with two other models, i.e. the pressure-gradient sensor model and the hotwire flow sensor. Moreover, we will demonstrate the benefit of the reassignment method to generate more focused maps of sources.

Pressure-gradient sensor model

The pressure-gradient sensor model is used in [23] when studying the fish lateral line canal. A lateral line canal consists of series of pores running along the fish body and neuromasts inside the canal. The pressure difference
between two adjacent pores induces a corresponding flow of fluid inside the canal. This flow moves the neuromasts, which have a hair-like structure with nerve cells to sense the motion. Therefore, what the fish senses is directly proportional to the pressure gradient along the canal. In [23], Curcic-Blake and van Netten have derived the model of the pressure gradient created along an array of sensors by a vibrating sphere. Assuming that the sensor array lies along the $x$ axis and a vibrating sphere located at range $d$ and offset $b$, i.e. $(b, d)$, then the signal captured by a sensor at location $(s, 0)$ is

$$f(s, b, d) = \frac{\rho w^2 a^3 X_0}{d^3} (\Psi_e \cos \varphi + \Psi_o \sin \varphi), \quad (5.15)$$

where $a$ is the radius of the sphere, $w$ and $X_0$ are the angular frequency and the amplitude of the sphere’s vibration, $\rho$ is the density of the fluid, and $\varphi$ denotes the angle between the direction of vibration of the sphere and the $x$-direction. In Equation (5.15), $\Psi_e$ and $\Psi_o$ are basis functions

$$\Psi_e(s, d, b) = \frac{1 - 2 \left(\frac{s-b}{d}\right)^2}{\left[1 + \left(\frac{s-b}{d}\right)^2\right]^{\frac{3}{2}}} \quad (5.16)$$

and

$$\Psi_o(s, d, b) = \frac{-3 \left(\frac{s-b}{d}\right)}{\left[1 + \left(\frac{s-b}{d}\right)^2\right]^{\frac{3}{2}}} \quad (5.17)$$

Note that the two basis functions $\Psi_e$ and $\Psi_o$ have the format of wavelet functions as defined in Section 5.1, where the offset $b$ is equivalent to the time shift and the range $d$ is equivalent to the dilation or scale. Therefore, a map equivalent to a scalogram can be generated using the wavelet transform. In fact, this approach has been used by Curcic-Blake and van Netten to map a vibrating source. However, we proceed further by applying the reassignment
method to increase the focus level of the map. Equipped with the reassignment formula derived by Auger and Flandrin (i.e., Equations (5.13) and (5.14)), the implementation is straightforward except that all the analyzing wavelets and array patterns need to be in a complex format. We use the Hilbert transform to generate analytic signals from those waveforms.

Figure 5.1: Images demonstrate improvement by the reassignment method for source localization maps. (A) and (B) are maps before and after reassignment for a vibrating sphere at (0, 20). (C) and (D) are maps before and after reassignment for a vibrating sphere at (0, 30). (E) and (F) are maps before and after reassignment for a vibrating sphere at (10, 20). The number of contour levels is 25.

Figure 5.1 demonstrates the reassignment focus effect on simulated signals.
The images on the top row show the source localization maps for three different sphere positions. The bottom images show how the reassignment technique produces more focused maps. In all three cases, we simulate an array of 51 sensors spaced 2 mm apart from -50 mm to 50 mm on the x axis. The direction of vibration of the simulated sphere is zero, i.e., $\varphi = 0$.

**Hotwire flow sensor model**

With promising simulation results from the pressure-gradient sensor model, we would like to test the technique with actual experiments. Unfortunately, we currently lack the sensors to build a canal lateral line from the hair-cell MEMS sensors in our artificial lateral line. We hence resort to using commercial hotwire sensors to test the concept of Capon beamforming for source localization [38]. This is actually a good test of robustness of our reassignment technique for different models and with real signals.

The commercial hotwire sensors operate on a heat dissipation principle different from the pressure-gradient sensors. Voltage applied across a sensor heats up the wire. Movement of water or air particles across the hot wire carries away heat, causing a change in the wire’s resistance and in turn the current. The change in current reflects the speed of water or air particles moving across the wire but does not record the direction of the flow. Figure 5.2 shows an artificial lateral line built from eight millimeter-scale commercial hotwire sensors. The space between two adjacent sensors is 12.5 mm. This setup is used for our source localization experiment together with the reassignment method.
A model for the hotwire flow sensor is derived in [18] as

$$f(s, b, d) = a^3 X_0 \frac{r^3}{2r^3} \sqrt{3\cos^2(\varphi)} + 1$$

(5.18)

where $r$ is the Euclidean distance from the sensor to the sphere

$$r = \sqrt{(s - b)^2 + d^2}$$

$a$ is the radius of the sphere, $X_0$ is amplitude of the sphere’s vibration, and $\varphi$ denotes the angle between the direction of vibration of the sphere and the $x$-direction. Then, wavelet functions we can use for
reassignment have the form:

\[
\Psi(s, b, d) = \frac{1}{\left[1 + \left(\frac{s-b}{d}\right)^2\right]^{\frac{3}{2}}}.
\]  

(5.19)

Figure 5.3: Simulation and experimental results of reassignment with an array of eight commercial hotwire sensors. (A) and (B) are maps before and after reassignment for a simulation with a vibrating sphere at (0, 15). (C) and (D) are maps before and after reassignment for an experiment with a vibrating sphere in front of sensor 4 at (0, 0). (E) and (F) are maps before and after reassignment for an experiment with a vibrating sphere in front of sensor 5 at (12.5, 0). The number of contour levels is 25.

Although there are only eight sensors in the array, images from Figure 5.3 show how the reassignment method can focus the maps. The first pair of images (A) and (B) are maps before and after reassignment for a simulated
vibrating sphere at (15, 0). The second pair (C, D) and third pair (E, F) are maps from real experimental data when a vibrating sphere is in front of sensor 4 (located at (0,0)) and sensor 5 (located at (12.5, 0). The experiment demonstrates that the assignment method works well with real data with noise.

5.2.2 Reassignment for the Weakly Electric Fish

It is now verified that the reassignment method can improve the focus of source localization maps for the fish lateral line system in Chapter 2. For the weak-electrosense system of artificial electric fish (Chapter 3), the application of the reassignment method is not straightforward. The reason lies in the model of array response. Unlike the wavelets of the lateral-line system as shown in Equation (5.16), (5.17) and (5.19), the wavelets of the electric-fish system cannot be formulated as a scaled wavelet function of $t = \frac{s-b}{d}$. However, with some modification, we will show that reassignment method can be extended to work with this electric fish model.

The model of weakly electric fish is shown in Equation (3.1):

$$\Delta \phi(\vec{r}) = \frac{a^3 \vec{E} \cdot \vec{r}}{\|\vec{r}\|^3} \left( \frac{\sigma_{\text{object}} - \sigma_{\text{water}}}{\sigma_{\text{object}} + 2\sigma_{\text{water}}} \right),$$

where $\Delta \phi(\vec{r})$ is the change in potential at position $\vec{r}$ relative to the sphere’s center and $\vec{E}$ is the electrostatic field at the source:

$$\vec{E} = \frac{Q}{4\pi\epsilon} \left( \frac{\vec{r}_{Q^+}}{\|\vec{r}_{Q^+}\|^3} - \frac{\vec{r}_{Q^-}}{\|\vec{r}_{Q^-}\|^3} \right),$$

66
Note that \( \vec{E} \) only depends on the source location \((b, d)\), so if we define \((E_x(b, d), E_y(b, d))\) as the coordinate of \(\vec{E}\), then Equation (3.1) becomes

\[
f(s, b, d) = \left( \frac{\sigma_{\text{object}} - \sigma_{\text{water}}}{\sigma_{\text{object}} + 2\sigma_{\text{water}}} \right) \frac{d^3[(s - b)E_x(b, d) + dE_y(b, d)]}{\|\vec{r}\|^3}
\]

\[
= C \frac{E_y(b, d) + E_x(b, d) \left( \frac{s-b}{d} \right)}{d^2 \left[ 1 + \left( \frac{s-b}{d} \right)^2 \right]^{\frac{3}{2}}}
\]

If all array patterns are normalized, we can ignore the constants \(C\) and \(d^2\). Then the wavelet function is

\[
\Psi(s, b, d) = \frac{E_y(b, d) + E_x(b, d) \left( \frac{s-b}{d} \right)}{\left[ 1 + \left( \frac{s-b}{d} \right)^2 \right]^{\frac{3}{2}}}
\] (5.20)

According to Section 5.1.2, the process of computing the reassignment coordinates requires the derivative of the wavelet function with respect to \(t = \frac{s-b}{d}\). Since we normalize the array patterns for each source location (i.e., fixed \((b, d)\)), we can take the derivative of \(\Psi(s, b, d)\) with respect to \(t\) while fixing \(E_x(b, d)\) and \(E_y(b, d)\). That means

\[
\Psi(t) = \frac{E_y(b, d) + E_x(b, d)t}{(1 + t^2)^{\frac{3}{2}}}
\]

and

\[
\frac{d\Psi(t)}{dt} = E_x(b, d) - 3E_y(b, d)t - 2E_x(b, d)t^2
\]

\[
(1 + t^2)^{\frac{3}{2}}
\]

With this modification, we now can apply the reassignment method described in Section 5.1.2 for an artificial weakly electric fish system. Figure 5.4 provides simulation results of an array of 28 sensors positioned 8 mm apart from 50 mm to 242 mm on the \(x\) axis. The electric field is generated from a dipole electric charge located at \((0,0)\) and \((292,0)\). The reassignment is
Figure 5.4: Simulation results of the reassignment with an array of 28 potential sensors. (A) and (B) are maps before and after reassignment in a simulation with an insulating sphere at (146, 20). (C) and (D) are maps before and after reassignment in a simulation with an insulating sphere at (146, 30). (E) and (F) are maps before and after reassignment in a simulation with an insulating sphere at (120, 20).

tested for three cases where an insulated source is located at three different positions (i.e., middle at (146, 20), greater range at (146, 30), and to the left at (120, 20)).

The simulation results indicate that the reassignment method can be extended to work with artificial weakly electric fish systems with small adaptation. We now test this method on real data with an array of seven voltage potential sensors spaced 32 mm apart. Although the number of sensors is quite limited, Figure 5.5 shows that the reassignment method works well
Figure 5.5: Simulation and experimental results of reassignment with an array of seven potential sensors. (A) and (B) are maps before and after reassignment in a simulation with an insulating sphere at (146, 20). (C) and (D) are maps before and after reassignment in an experiment with an insulating sphere in front of sensor 4 at (146, 0). (E) and (F) are maps before and after reassignment in an experiment with an insulating sphere in front of sensor 5 at (178, 0). The number of contour levels is 25.

with both simulated and real data.

5.3 Reassignment for Sparse Beamforming

Speedup

In Chapter 4, we introduce a new $\ell_1$-minimization algorithm to solve the sparse basis selection in the sparse beamforming technique. The algorithm is actually a backward basis elimination process which iteratively excludes
a basis at each step. The computational cost is high at the beginning of the process because of handling a large matrix. Therefore, it would yield a significant improvement of speed if a number of bases are correctly excluded together at the beginning.

Results from the reassignment method in previous sections indicate that it could be a good candidate as a pre-processing step before the backward elimination process. While reassigning the values on the map, the method produce a new map with some zero-value regions. Those bases corresponding to the regions can be removed altogether before executing the backward elimination process with our algorithm in Chapter 4. The pre-processing basis removal also further reduces the number of bases included back to the dictionary after failing the optimality check step.

However, the pre-processing step does not arrive at no cost. In fact, the cost of computing the reassignment method is significant as well. But fortunately, the reassignment vector at each location can be computed in parallel. This is an important feature since all the current $\ell_1$-minimization algorithms are sequential. Therefore, we can argue that we can speedup any backward basis-elimination algorithm by parallelizing the process of eliminating a group of bases at the beginning.

In order to illustrate the benefit of this approach, we run a numerical analysis in Matlab with the assumption that the reassignment is executed in a parallel manner, i.e., one iteration per thread. We simulated an array of 51 sensors and an analysis grid of 13 by 25. Without the pre-processing reassignment step, the total time to compute the source localization map is 1.3246 seconds.
With the pre-processing reassignment, the number of bases is reduced from 325 to 99. The average time for a reassignment iteration is 0.0395 seconds and the time to compute the map from 99 bases is 0.424 seconds. Thus, the total time with the reassignment is 0.4735 seconds, about 36% of the time without the pre-processing reassignment.

5.4 Reassignment with Multiple Sources

Although the reassignment method works well in the case of a single source, there is degradation in the case of multiple sources. When there are two sources, the original map has an incorrect lump of energy between the two sources; the reassignment method then mistakenly focuses energy into that lump location and sharpens it. In other words, the reassignment method still sharpens the source images but does not help to resolve or separate multiple overlapping sources. Figure 5.6 shows the simulation results of two sources as they are moving closer toward each other. Those results indicate that further investigation is needed to improve the reassignment method for multiple sources.
Figure 5.6: Simulation results of reassignment with an array of 120 potential sensors. (A) and (B) are maps before and after reassignment with two insulating spheres at \((-40,15)\) and \((40,15)\). (C) and (D) are maps before and after reassignment with two insulating spheres at \((-30,15)\) and \((30,15)\). (E) and (F) are maps before and after reassignment with two insulating spheres at \((-20,15)\) and \((20,15)\). (G) and (H) are maps before and after reassignment with two insulating spheres at \((-10,15)\) and \((10,15)\). The number of contour levels is 25.
Chapter 6

Conclusion

Source localization is one of the most vital skills needed for many species to survive. Inspired by this sensing capability in nature, we have studied and demonstrated the possibility of engineering man-made systems that can imitate the sensing mechanisms of some species. Our artificial lateral line is capable of 3D imaging of a vibrating sphere in water. Even based on a dipole model, the system seems to work for more natural sources such as tail-flickering crayfish. This could lead to an application of equipping autonomous underwater vehicles with a near-field sensing capability similar to that of fish. However, our current system does not work in the context of a flow phenomenon such as a wake. It is still an open research area to design a man-made system that can detect vortices in water and track wakes as to seals with their whiskers. This could be another exciting application for an automatic detect-and-track underwater system. By analyzing the theoretical performance limit via the Cramer-Rao bound, we discover the fundamental limitations of the lateral line system. That is, the system’s sensing capability degrades quickly when reaching beyond a range of one body length. In fact, studies in nature confirm that the range of a fish’s lateral line sensing stays within a body length [56].

Weakly electric fish have been studied extensively in nature, but we are one of the very few groups to build an engineering prototype system to test
the capability of localizing an object in water. Exploiting the fact that the number of sources for detection in many natural scenarios is small, we propose the new concept of sparse beamforming. Although sparse beamforming does not meet the restricted isometry property (RIP) condition to guarantee correct estimation of locations of more than one source, results from our simulations and experiments still demonstrate good estimation. Obviously, the RIP condition is not a sufficient condition, so a lot of improvements are needed before this weakly electric sensing can be practically realized. But this could generate new interesting applications in the future not only in water but in air following the capacitive-sensing principle of the Theremin instrument [57].

In terms of algorithm contribution, we have proposed two new methods (i.e, the $\ell_1$-minimization algorithm and the reassignment method) for enhancing the source localization maps. Our $\ell_1$-minimization algorithm follows the backward basis elimination approach by excluding bases at each iteration to reduce the $\ell_1$ cost function. The algorithm also has a mechanism to include back bases if it has not reached the global minimum. This mechanism could open a new research direction for algorithms with the capability of updating bases when the measurements change. It is similar to many species’ capability of moving and adaptively updating their sensing outputs based on previous ones.

The reassignment method is an old technique in the time-frequency analysis literature. However, we have revived this technique by applying it to sharpen the source images generated by both the artificial lateral line system and the weakly electric sensing system. Currently, the technique only
works for array pattern models of wavelet format or with slight modification. We are still looking for extensions of this technique for more general models. This then could be a good post-processing tool for source-localization applications.
Appendix A

Reassignment of Spectrogram and Scalogram

A.1 Reassignment Derivation for Spectrogram and Scalogram

In this section, we review results from the time-frequency and time-scale literature needed for the reassignment method discussed in Chapter 5. These results are taken variously from [51, 52, 53, 55].

A.1.1 Spectrogram as 2D Convolution of WV Distributions

We show that the spectrogram of $x(t)$ can be computed as a 2D convolution of Wigner-Ville (WV) distribution of the signal $x(t)$ and WV distribution of analyzing window $h(t)$, i.e.,

$$\text{STFT}_h(x; t, w)\text{STFT}_{h}^*(x; t, w) = \int_{-\infty}^{+\infty} \text{WV}(x; \tau, \nu)\text{WV}(h; t-\tau, w-\nu)d\tau d\nu$$

(A.1)

Before proving this, we assume some key properties:

$$\int_{-\infty}^{+\infty} wH(w)e^{jwt}\frac{w}{2\pi} = -j\frac{dh}{dt}(t)$$

$$\int_{-\infty}^{+\infty} e^{-j\nu(\tau_1-\tau_2)}d\frac{\nu}{2\pi} = \delta(\tau_1 - \tau_2)$$
Then the left-hand side (LHS) of Equation (A.1) is

\[
\text{LHS} = \int_{-\infty}^{+\infty} x(\tau_1) h^*(t - \tau_1) e^{-jw_1 d\tau_1} \int_{-\infty}^{+\infty} x^*(\tau_2) h(t - \tau_2) e^{jw_2 d\tau_2}
\]

\[
= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x(\tau_1) x^*(\tau_2) h^*(t - \tau_1) h(t - \tau_2) e^{-jw(t_1-t_2)} d\tau_2 d\tau_1
\]

Then, the right-hand side (RHS) equals

\[
\text{RHS} = \int \int \int_{-\infty}^{+\infty} x(\tau + \tau_1/2) x^*(\tau - \tau_1/2) e^{-j\tau_1\nu} d\tau_1
\]

\[
= \int \int \int_{-\infty}^{+\infty} x(\tau + \tau_1/2) x^*(\tau - \tau_1/2) h(t - \tau + \tau_2/2) h^*(t - \tau - \tau_2/2)
\]

\[
e^{-j\nu \tau_2} e^{-j\nu(\tau_1-\tau_2)} d\tau_1 d\tau_2 d\tau d\nu
\]

\[
= \int \int \int_{-\infty}^{+\infty} x(\tau + \tau_1/2) x^*(\tau - \tau_1/2) h(t - \tau + \tau_2/2) h^*(t - \tau - \tau_2/2)
\]

\[
e^{-j\nu \tau_2} \delta(\tau_1 - \tau_2) d\tau_1 d\tau_2 d\tau
\]

\[
= \int \int x(\tau + \tau_0/2) x^*(\tau - \tau_0/2) h(t - \tau + \tau_0/2) h^*(t - \tau - \tau_0/2) e^{-jw_0 d\tau_0} d\tau_0 d\tau
\]

If we substitute \( \tau_1 = \tau + \tau_0/2 \) and \( \tau_2 = \tau - \tau_0/2 \), then \( \tau_0 = \tau_1 - \tau_2 \), and we conclude LHS = RHS.

Similarly, we can show that

\[
\text{STFT}_g(x; t, w) \text{STFT}_h^*(x; t, w) = \int \int_{-\infty}^{+\infty} \text{WV}(h \cdot g; \tau, \nu) \text{WV}(x; t-\tau, w-\nu) d\tau d\nu
\]

\[
\frac{\nu}{2\pi}
\]

(A.2)
where \( WV(g \cdot f; t, w) \) is defined as

\[
WV(g \cdot f; t, w) := \int_{-\infty}^{+\infty} g(t + \tau/2)f^*(t - \tau/2)e^{-j\nu \tau}d\tau.
\]

### A.1.2 New Reassignment Formula for Spectrogram

We need to show that

\[
\int \int_{-\infty}^{+\infty} \tau WV(h; \tau, \nu)WV(x; t - \tau, w - \nu)d\tau d\nu = Re \left\{ \text{STFT}_{\tau \nu}(x; t, w) \right\}
\]

and

\[
\int \int_{-\infty}^{+\infty} \nu WV(h; \tau, \nu)WV(x; t - \tau, w - \nu)d\tau d\nu = -Im \left\{ \text{STFT}_{\tau \nu}(x; t, w) \right\}
\]

To prove Equation (A.3) and (A.4), we use the property A.2 from Section A.1.1. Then the RHS of Equation (A.3) is

\[
\text{RHS} = Re \left\{ \int \int_{-\infty}^{+\infty} WV(h \cdot T h; \tau, \nu)WV(x; t - \tau, w - \nu)d\tau d\nu \right\}
\]

but

\[
Re \left\{ WV(h \cdot T h; \tau, \nu) \right\} = Re \left\{ \tau WV(h; \tau, \nu) - \int_{-\infty}^{+\infty} \frac{\tau_1}{2} h(\tau + \tau_1/2) h^*(\tau - \tau_1/2)e^{-j\nu \tau_1} d\tau_1 \right\}
\]

\[
= \tau WV(h; \tau, \nu).
\]
Similarly, the RHS of Equation (A.4) is

\[
\operatorname{RHS} = -\Im \left\{ \int_{-\infty}^{+\infty} \mathcal{WV}(h \cdot \mathcal{D}h; \tau, \nu) \mathcal{WV}(x; t - \tau, w - \nu) d\tau d\nu \right\}
\]

\[
= \int_{-\infty}^{+\infty} -\Im \{ \mathcal{WV}(h \cdot \mathcal{D}h; \tau, \nu) \} \mathcal{WV}(x; t - \tau, w - \nu) d\tau d\nu
\]

but

\[
-\Im \{ \mathcal{WV}(h \cdot \mathcal{D}h; \tau, \nu) \} =
\]

\[
= -\Im \left\{ \int_{-\infty}^{+\infty} h(\tau + \tau_1/2) \mathcal{D}h^*(\tau - \tau_1/2) e^{-j\nu \tau_1} d\tau_1 \right\}
\]

\[
= -\Im \left\{ \int_{-\infty}^{+\infty} h(\tau + \tau_1/2) \int jw H(w) e^{jw(\tau - \tau_1/2)} \frac{dw}{2\pi} e^{-j\nu \tau_1} d\tau_1 \right\}
\]

\[
= -\Im \left\{ -j\nu \mathcal{WV}(h; \tau, \nu) + \int jw H(\nu + w/2) H^*(\nu - w/2) e^{jw\tau} \frac{dw}{2\pi} \right\}
\]

\[
= \nu \mathcal{WV}(h; \tau, \nu)
\]

A.1.3 Scalogram as 2D Convolution of WV Distributions

We need to show that

\[
\mathcal{CWT}_h(x; t, a) \mathcal{CWT}_h^*(x; t, a) = \int_{-\infty}^{+\infty} \mathcal{WV}(x; \tau, \nu) \mathcal{WV}(h; \frac{\tau - t}{a}, a\nu) d\tau d\nu
\]

(A.5)

\[
\operatorname{LHS} = \left( \frac{1}{|a|} \int_{-\infty}^{+\infty} x(\tau_1) h^*(\frac{\tau_1 - t}{a}) d\tau_1 \right) \left( \frac{1}{|a|} \int_{-\infty}^{+\infty} x(\tau_2) h^*(\frac{\tau_2 - t}{a}) d\tau_2 \right)^*
\]

\[
= \frac{1}{|a|} \int_{-\infty}^{+\infty} x(\tau_1) h^*(\frac{\tau_1 - t}{a}) x^*(\tau_2) h(\frac{\tau_2 - t}{a}) d\tau_1 d\tau_2
\]

\[
= \frac{1}{|a|} \int_{-\infty}^{+\infty} x(\tau_1) x^*(\tau_2) h(\frac{\tau_2 - t}{a}) h^*(\frac{\tau_1 - t}{a}) d\tau_1 d\tau_2
\]
\[ \text{RHS} = \int \int \left( \int_{-\infty}^{+\infty} x(\tau + \tau_1/2)x^*(\tau - \tau_1/2)e^{-j\tau_1 \nu}d\tau_1 \right. \]
\[ \times \int_{-\infty}^{+\infty} h\left(\frac{\tau - t}{a} + \tau_2/2\right)h^*\left(\frac{\tau - t}{a} - \tau_2/2\right)e^{-j\tau_2 \omega}d\tau d\nu \]
\[ = \int \int \int_{-\infty}^{+\infty} \left[ x(\tau + \tau_1/2)x^*(\tau - \tau_1/2)h\left(\frac{\tau - t}{a} + \tau_2/2\right)h^*\left(\frac{\tau - t}{a} - \tau_2/2\right) \right. \]
\[ \times \left( \int_{-\infty}^{+\infty} e^{-j(\tau_1 + a\tau_2)\nu}d\nu \right) \]
\[ \left. \right] d\tau_1 d\tau_2 d\tau \]
\[ = \int \int \int_{-\infty}^{+\infty} \left[ x(\tau + \tau_1/2)x^*(\tau - \tau_1/2)h\left(\frac{\tau - t}{a} + \tau_2/2\right)h^*\left(\frac{\tau - t}{a} - \tau_2/2\right) \right. \]
\[ \times \delta(\tau_1 + a\tau_2) \]
\[ \left. \right] d\tau_1 d\tau_2 d\tau \]
\[ = \frac{1}{|a|} \int \int_{-\infty}^{+\infty} x(\tau + \tau_0)x^*(\tau - \tau_0)h\left(\frac{\tau - t}{a} - \tau_0\right)h^*\left(\frac{\tau - t}{a} + \tau_0\right)d\tau_0 d\tau \]

where \( \tau_1 = 2\tau_0 \) and \( \tau_2 = -2\tau_0/a \). By resubstituting \( \tau_1 = \tau + \tau_0 \) and \( \tau_2 = \tau - \tau_0 \) we get the LHS.

Similarly, we can show that

\[ \text{CWT}_g(x; t, a)\text{CWT}^*_h(x; t, a) = \int \int_{-\infty}^{+\infty} \text{WV}(x; \tau, \nu)\text{WV}(h \cdot g; \frac{\tau - t}{a}, a\nu)d\tau d\nu \frac{\nu}{2\pi} \]

(A.6)

### A.1.4 New Reassignment Formula for Scalogram

We need to show that

\[ \int \int_{-\infty}^{+\infty} (\tau-t)\text{WV}(x; \tau, \nu)\text{WV}(h; \frac{\tau - t}{a}, a\nu)d\tau = \text{Re} \left\{ a\text{CWT}_{a\tau h}(x; t, a)\text{CWT}^*_h(x; t, a) \right\} \]

(A.7)

and

\[ \int \int_{-\infty}^{+\infty} \nu\text{WV}(x; \tau, \nu)\text{WV}(h; \frac{\tau - t}{a}, a\nu)d\tau = -\text{Im} \left\{ \frac{1}{a}\text{CWT}_{Dh}(x; t, a)\text{CWT}^*_h(x; t, a) \right\} \]

(A.8)
To prove Equations (A.7) and (A.8), we use the property (A.6) from Section A.1.3. Then the RHS of Equation (A.7) is

\[
\text{RHS} = a \text{Re} \left\{ \int_{-\infty}^{+\infty} \text{WV}(x; \tau, \nu) \text{WV}(h \cdot T h; \tau - t / a, a \nu) d\tau d\nu / 2\pi \right\}
\]

but

\[
\text{Re} \{ \text{WV}(h \cdot T h; \tau, \nu) \} = \text{Re} \{ \tau \text{WV}(h; \tau, \nu) - \\
\int_{-\infty}^{+\infty} \frac{\tau_1}{2} h(\tau + \tau_1/2) h \ast (\tau - \tau/2) e^{j\nu \tau_1} d\tau_1 \}
\]

\[= \tau \text{WV}(h; \tau, \nu),\]

then

\[
\text{Re} \left\{ \text{WV}(h \cdot T h; \frac{\tau - t}{a}, a \nu) \right\} = \frac{\tau - t}{a} \text{WV}(h; \frac{\tau - t}{a}, a \nu).
\]

So we have RHS = LHS.

Similarly, the RHS of Equation (A.8) is

\[
- \text{Im} \left\{ \text{CWT}_{D h}(x; t, a) \text{CWT}_{h}^*(x; t, a) \right\} = \\
- \text{Im} \left\{ \int_{-\infty}^{+\infty} \text{WV}(x; \tau, \nu) \text{WV}(h \cdot D h; \tau - t / a, a \nu) d\tau d\nu / 2\pi \right\}
\]

\[= \int_{-\infty}^{+\infty} a \nu \text{WV}(x; \tau, \nu) \text{WV}(h; \frac{\tau - t}{a}, a \nu) d\tau d\nu / 2\pi \]
References


