Theoretical evaluation of continuous-wave time reversal acoustics in a half-space environment

by

Blaine Harker

Submitted to Brigham Young University in partial fulfillment

of graduation requirements for University Honors

Department of Physics and Astronomy

Brigham Young University

May 2012

Advisor: Brian E. Anderson

Advisor: Kent L. Gee

Honors Representative: Roy W. Silcox

Signature: ____________________

Signature: ____________________

Signature: ____________________

ABSTRACT

Theoretical evaluation of continuous-wave time reversal acoustics in a half-space environment

Blaine Harker

Department of Physics and Astronomy

Bachelor of Science

Time reversal (TR) acoustics is a technique used to locate sources, using a set of transducers called a time reversal mirror (TRM) and is especially useful in reverberant environments. TR is commonly used to find acoustically small sources using a pulsed waveform. Here TR is applied to simple sources using steady-state waveforms using a straightforward, computational point source propagation theoretical model in a half-space environment. It is found that TR can effectively localize a simple source broadcasting a continuous wave, depending on the angular spacing. Furthermore, the aperture (angular coverage around the source) of the TRM is the most important parameter when creating a setup of receivers for imaging a source. This work quantifies how a TRM may be optimized when the source's location is known to be within a certain region of certainty.

ACKNOWLEDGMENTS

First of all, I owe a big thanks to one of my advisors, Dr. Anderson, for his willingness to assist as a long-distance advisor and see this project to its completion. He has been very helpful in every stage of the research over the past couple of years. Also, I'd like to thank Dr. Gee and Alan Wall for their help regarding my questions with the F-22 jet experiment. Thank you Acoustical Society of America and Office of Research and Creative Activities (ORCA) at Brigham Young University for funding this project. I am indebted to the BYU Acoustic Research Group for their facilities as well as the Fulton Supercomputing Lab for facilities and as well as help with generating my batch scripts. Finally, I'd like to thank my parents and grandparents who inspired me to work hard, dream big and to enjoy the journey.

Contents

List of Figures

Chapter 1

Introduction

Acoustic source localization methods are the subject on an ongoing field of study with a broad range of applications. One such method is time reversal (TR), a simple yet powerful technique for source localization in a complex environment. Fink, a well-regarded researcher in TR studies, provided general procedures and guidelines in this field, many which have set the standard.[1] One guideline considered in this thesis is the qualitative suggestion for array spacing for the transducers used. This introduction presents a brief overview of TR as well as the array spacing guideline suggested by Fink and a discussion regarding that guideline.

1.1 Time Reversal: Basic Theory

TR is a type of acoustic localization[2–4] with many applications, such as lithotripsy to destroy kidney stones,[5] earthquake localization and characterization,[6,7] crack localization and characterization,[8,9] target scatterer detection,[10,11] land mine detection,[12] and secure underwater sound communication,[13] though in some ways TR is still in the development stage. One of its fundamental uses is to localize a sound source of unknown location in a complex environment.

The essence of TR may be described in terms of a simple analogy. Imagine a movie of a pebble dropped into a pond. Circular ripples spread out away from the drop location. If one were to play the movie backwards in time, the ripples would converge at the drop location, recreating the initial disturbance caused by the pebble. This is the essence of the TR technique. Imagine there are sensors that record these ripples on the water surface from an unknown source at various sensor locations, we then reverse the detected signals, and broadcast the reversed signals from the sensor locations. As a result, part of the waveform broadcast from each of the sensors will arrive simultaneously at the initial disturbance location such that the waves will interfere constructively and reproduce the pebble's initial agitation.[4]

This analogy extends easily to the idea of sound source localization. Fink explains that since the linear wave equation contains a second-order time derivative, if a source with a waveform $p(\mathbf{r},t)$ solves this equation, then $p(\mathbf{r},-t)$ will solve it as well.[2] Let us assume that an unknown source creates a signal that propagates to a receiver [see Fig. 1.1(a)]. Sound propagates spherically from the source, reflecting from the boundary so that two paths arrive at the receiver. If the receiver is omnidirectional, it records the waveform of each arriving signal, regardless of whether the signal comes directly from the source or via a reflective surface. Note that reflective surfaces create multiple propagation paths from source to receiver. The received signal is reversed and broadcast from the receiver location. In Fig. 1.1(b), each path is now traversed by both the direct and reflected signals, resulting in four arrivals at the original source location. The solid lines correspond to the original paths traveled and the dotted lines are the alternate paths taken by the direct or reflected signal. Without directional information, each waveform arrival will not only retrace its original path to result in constructive interference, but they will also trace other paths back

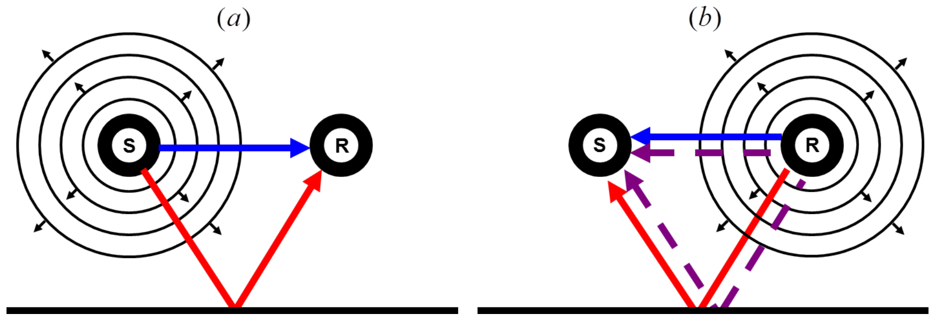

FIG. 1.1. Schematic drawing of the two steps of the time reversal process, (a) forward propagation and (b) backward propagation. This drawing helps illustrate the propagation paths in each stage.

to the source resulting in destructive interference. These additional paths are shown by the dotted lines. Fortunately, the constructive interference statistically creates a stronger response than does the destructive interference.

In the forward propagation step, the waveform recorded by a standard receiver receives no information as to the direction of incoming paths. Therefore, in the backward propagation the received waveform is simply reversed in time then broadcast from the receiver location. This results in a constructive focusing since the propagation paths are encoded in the timing of the forward propagation arrivals.

With the addition of multiple receivers, more information can be recorded from the source, and hence the backward propagation will yield a more precise and identifiable localization of the original source. If the sound source is a pulse, this localization results in localized spatial focusing and pulse-like temporal reconstruction. If the source is a continuous wave (CW), the localization may still be spatially focused but it may also require many transducers.[14] The use of multiple transducers in the TR process constitutes an array known as the time reversal mirror (TRM).[15]

As long as the environmental conditions (e.g. sound speed profile and boundary conditions) are either known or the environment is stable, an accurate model can be created for the backward propagation step and the transducers will broadcast their respective signals to retrace initial paths from the source to TRM.[15] To summarize, when the TRM records an arriving signal, the signal is reversed, and subsequently broadcast, the result at the original source location (in addition to the background side lobe byproducts of TR) is to reproduce a reversed-in-time reproduction of the original sound.

TR provides advantages to alternative sound localization techniques but it also has its limitations. TR is simple in that it makes no assumptions about the sound source and it does not require complex algorithms or inputs (such as computed phase delays as in the case of beamforming techniques). Furthermore, while other sound localization methods break down with an increasingly complex environment, such as with a number of reflective surfaces, the efficiency of TR actually increases.[2] However, the medium of interest cannot change between the forward and backward steps of the TR process, or the medium must be accurately known if the backward propagation step is done computationally. In addition, TR requires simultaneous recordings for each transducer used in the setup, and the source localization is restricted to the diffraction limit[15] (as are many localization methods). Finally, the spacing of transducers in the setup is an important factor, as grating lobes may be present in the field of interest (as is the case with most ray tracing techniques). The question of the appropriate transducer spacing in a TR experiment will be discussed in depth here.

1.2 Hypothesis

According to Fink *et al.*, spatial separation of TRM transducers by $\lambda/2$ (where λ represents the wavelength of the broadcast signal) is necessary to avoid grating lobes (also a general guideline in array design[16]), however they clarify that such spacing is not necessary if the

TRM is pre-focused on the source of interest.[1,15,17] The present study will add further discussion to this guideline. In a realistic experiment it can quickly become impractical to use half wavelength spacing and a more economical and simple experimental design is desired. We optimize the number of transducers needed for a TR experiment given a relative knowledge of the source location and frequency content. In optimizing the TRM layout we find that TR can effectively localize a simple source broadcasting a CW, depending on the angular spacing. The results of this analysis can be used in further research to generalize TR reconstruction of more complicated sources.

In addition, there are many artifacts of TR that can affect the extent and quality to which the original source can be identified in the backward propagation step. Many of these parameters depend principally on the positioning and design of the TRM. The purpose of this thesis is to determine the dependency of these parameters on the quality of the TR focusing. We show that the aperture of the TRM is a primary parameter to consider when designing a TRM layout for imaging a source. Other parameters are influential as well and will be discussed in the following chapters. This feasibility study has been developed to apply specifically to jet noise (See Appendix A); however it has been generalized so that a similar experiment using steady state waves in a half-space environment may benefit from these results.

This analysis is directed towards developing useful guidelines in order to set up an effective recording of the forward propagation step in a TR experiment. Qualitative details regarding how many sensors to set up, where they should be placed, and frequency limitations for this setup are also addressed. To this end, we explore the efficiency and limitations of this technique due to the number of TRM sensors used, the angular coverage of the TRM, frequency, and an assumed *region of certainty* of the source location.

Chapter 2

Methods

2.1 Model Design

The present study uses a point source broadcasting a single frequency in a half-space environment. From a theoretical study using a single frequency, further complexity can be added using superposition of single sine wave sources that could in principle synthesize a more complex source. Although in many experiments a source (such a jet source) is complex and extended, the point source is used in these analyses because the focus is to optimize the sensor positions for a TRM layout. This provides a foundation for future work.

The source broadcasts a simple CW sine-wave omnidirectionally. The source is placed at a position $z_0 = 1.5\lambda$ above a hard surface, thus representing a source in a half-space environment. In order to represent the ground reflection in the model, the source at height z_0 was reflected about the ground plane to create a virtual source at height $-z_0$. The TRM elements would thus record the superposition of two sine waves, the direct signal from the source to each coplanar TRM element and the reflected path arrival depicted in Fig. 2.1(a).

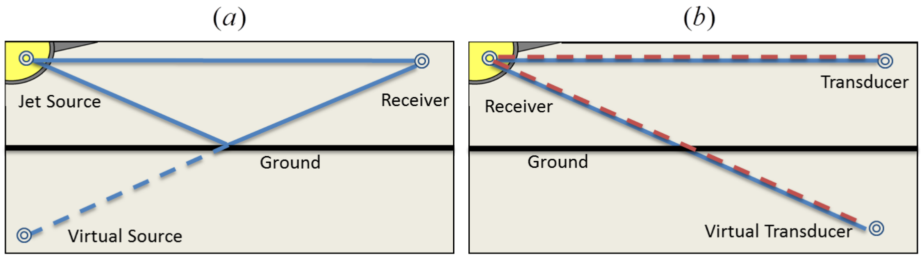

FIG. 2.1. Illustration of the arrival paths of (a) the forward propagation and (b) the backward propagation step. In (b), the direct arrival path from the forward propagation step is indicated by a solid line and the reflected arrival is indicated by a dashed line.

Then the equation for the field in the forward propagation, for a given wavenumber, k, and in the xy plane of the source, is

$$F_{fwd}(x,y,t) = \sum_{n=1}^{N} \frac{A}{r_d} e^{j(kr_d - \omega t)} + \frac{A}{r_r} e^{j(kr_r - \omega t)}, \tag{2.1}$$

where A is the amplitude, j is the unit imaginary number, ω is the angular frequency, and the vector r_d represents the distance from the source to the point (x,y) on the planar field and r_r is the path from the virtual source to (x,y). N represents the number of elements in the TRM. A pressure field in the plane of the source and the TRM elements is created for every element within a given TRM and summed linearly.

In the backward propagation step, only the information received by the sensor elements can be utilized, so the individual fields from the source and virtual source are not separated. Therefore, only the combination of the amplitude and phase information from each signal is used in the backward propagation. We define θ_d and θ_r as the phase information from the direct signal and reflected signal respectively. Similarly we define A_d and A_r as the amplitude from each respective signal.

The backward propagation step requires that the medium have similar conditions to those of the forward propagation step.[15] When each element transmits the reversed signal in the half-space, there will again be two paths of travel from the element to the original source location from each arrival in the forward step. Thus, there will be four arrivals at the source location, two that interfere constructively and two that interfere destructively (See Fig. 2.1(b)). The backward propagation equation of the field in the plane of the source is

$$\begin{aligned} F_{back}(x,y,t) = \sum_{n=1}^{N} \frac{A_d}{r_{d,d}} e^{j(kr_{d,d}-\omega t+\theta_d)} + \frac{A_d}{r_{d,r}} e^{j(kr_{d,r}-\omega t+\theta_d)} \\ + \frac{A_r}{r_{r,d}} e^{j(kr_{r,d}-\omega t+\theta_r)} + \frac{A_r}{r_{r,r}} e^{j(kr_{r,r}-\omega t+\theta_r)}, \end{aligned} \tag{2.2}$$

where $r_{d,d}$ represents a direct arrival from the TRM element to point (x,y) on the plane and it was also a direct arrival in the forward propagation step. Similarly, $r_{d,r}$ is a reflection arrival to (x,y) and it was a direct arrival in the forward propagation step. This type of notation for $r_{d,d}$ and $r_{d,r}$ holds as well for $r_{r,d}$ and $r_{r,r}$. Note that the signals include no additional background noise and they were recorded assuming a steady state. The component fields for each of the four terms in the equation are shown in Fig. 2.2. The fields of Fig. 2.2(a) and Fig. 2.2(d) show a positive amplitude at the original source location (marked by arrows) and represent the signal arrivals that will add constructively in this region. The fields in Fig. 2.2(b) and Fig. 2.2(c) are the undesirable signal arrivals that interfere destructively at the source location. Once the individual fields for each element-source combination are calculated in the backward propagation step, they are summed together to obtain a total pressure field, seen in Fig. 2.3.

2.2 Parameters of Study

Several parameters are varied in order to gain further insight applicable in the design of a TRM layout (whose geometry is illustrated in Fig. 2.4). The parameters include the TRM element spacing, the TRM aperture, the region of certainty of the source location, frequency, shape and size of the TRM layout and the position of the source relative to the TRM geometric center. These parameters are described below.

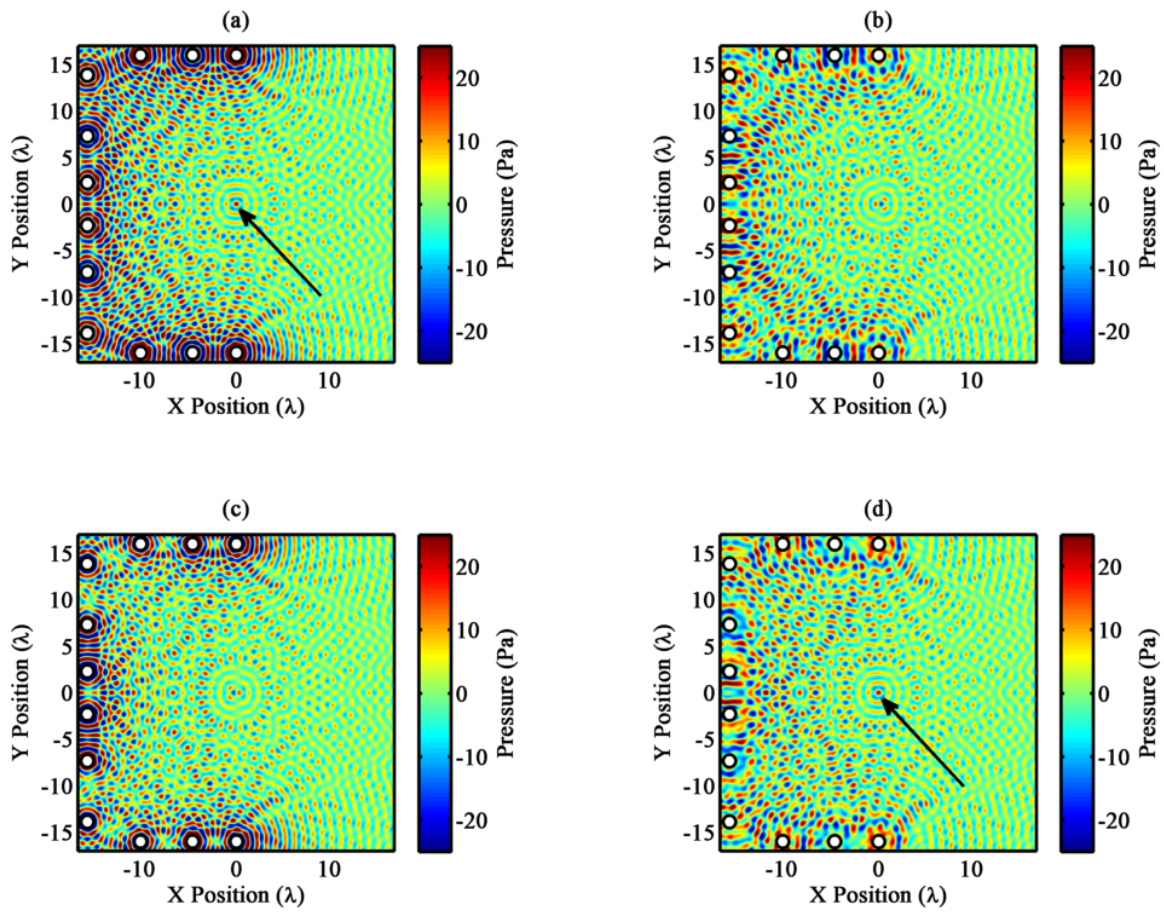

FIG. 2.2. The real part of the pressure wave fields created by each propagation path in the backward propagation step. The original paths from the forward propagation step and the arrivals paths are (a) the direct, direct path, (b) direct, reflected path, (c) reflected, direct path, and (d) reflected, reflected path [see Eq. (2.2)].

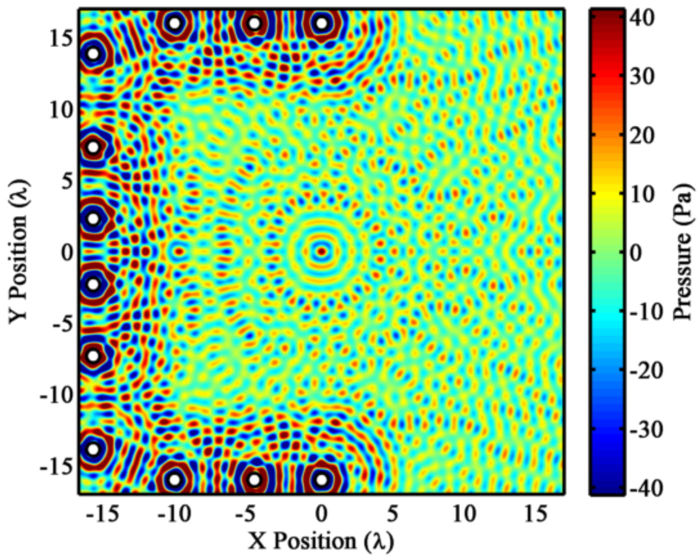

FIG. 2.3. The summed real part of the pressure field (from the fields in Fig. 2.2), which simulates the total backward propagation wave field.

The TRM *element spacing* is the distance between adjacent TRM elements. Half-wavelength spacing between elements is used to eliminate grating lobes globally. However, as will be shown later, for cases where the source is known to a certain extent this spacing may be relaxed considerably since grating lobes will not interfere with the region of interest. In this study, the element spacing was found to show unique properties associated with an angular separation rather than separation distance. Hence, the TRM elements were spaced at an equal angle about the source, even when the shape of the TRM was not circular. When the TRM layout was not circular, the distance between adjacent elements was varied but the angular spacing between elements was kept the same as for a circular TRM by projecting the element location along that angle. Thus the elements would not be equally spaced but they would be separated by an equal angle relative to the source location.

The *angular aperture* is the total angular span that the TRM spans around the source. The angular aperture may be varied anywhere between 0 to 2π radians about the source.

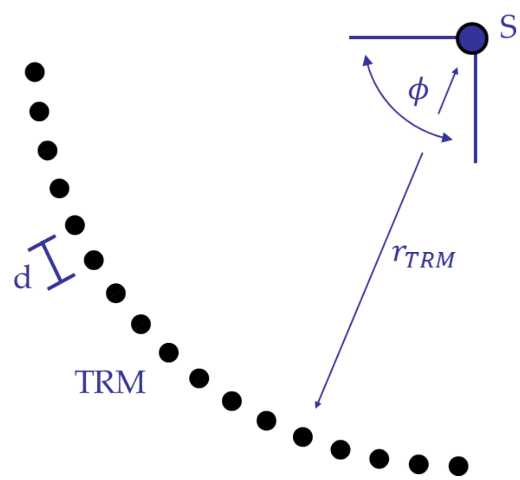

FIG. 2.4. Illustration of the geometry of the time reversal mirror layout with angular spacing d, time reversal mirror radius r_{TRM}, and aperture angle ϕ around source S.

Six different aperture sizes were used, varying from a small $\pi/12$ radian coverage to a full 2π radian coverage.

The *region of certainty* refers to the relative knowledge of the source position. This is defined by a square of a determined size which is centered upon the source location. In this study, a square with sides of length 5λ by 5λ was used. Notice that while the source location is identifiable in the center of Fig. 2.3, there are other artifacts in the field which make source localization more difficult and these artifacts would not be present if the elements were more tightly spaced. However, if the source location is known a priori to within 2.5λ, then the source location is easily recognizable in the backward propagation wave field. Altering the region of certainty affects the localization quality of TR as the field of interest about the source is changed. In practice, it is reasonable to assume some certainty of the location of the sound source.

The shape of the TRM used in the simulations traced out a circular shape or a square shape. As mentioned previously in the case of the square array, the elements were spaced at an equal angle from the source and not at a constant separation distance.

Finally, it is noteworthy to mention that when the position of the source was moved off of the geometric center of the TRM, that the angular spacing of the TRM elements and the aperture were not altered. However, the *effective* angular spacing and aperture would change as a consequence of the source being moved at a position off-center of the TRM layout. The results of each parameterization study are given in Chapter 3.

2.3 Computer Simulations

Using the simplified model described here, parameters are quickly varied, iterated and retested, producing results for hundreds of different scenarios. In this manner, the model allowed for a pressure field simulation over a large area around the source location. It also allowed for a large number of parameters to be studied. The script developed for this parameterization study allowed for multiple tests in a very small amount of time. Additionally, many of the tests were combined to utilize available RAM and run efficiently over multiple processors simultaneously. One such consideration was in utilizing MATLAB's matrix multiply capability and other functions which are designed for multiple processor use. This decreased overall computation time significantly. Tests were conducted on the BYU Acoustic Research Group's Kirchhoff Server, a Dell T Series Tower Server with an Intel® Xeon® E5530 with 2 processors for a total of 16 threads, and 48GB of installed RAM. Furthermore, BYU's Mary Lou 6 (M6) was also used. The Mary Lou 6 is a supercomputer with over 500 nodes and each node consists of 12 Hex-core Intel Westmere processors and 24GB of utilizable RAM. Because of restrictions associated with using MATLAB, only one node on the supercomputer could be utilized per job. However, multiple jobs could be submitted on the M6. Thus, scripts were converted to batch jobs which could be submitted simultaneously to the supercomputer. A general sampling of the scripts

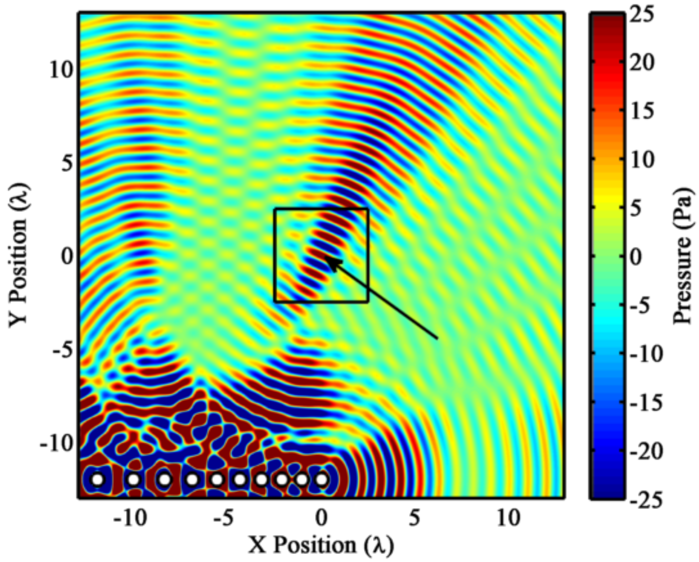

FIG. 2.5. Backward propagation wave field of a time reversal mirror using a partial square layout of aperture $\pi/4$ radians and 10 TRM elements. The arrow points to the original source location and the black outline a particular region of certainty ($5\lambda \times 5\lambda$).

is included in Appendix B.

2.4 Localization Quality Metric

While the source location of Figure 2.3 was easily identifiable, other situations such as that shown in Figure 2.5 are not as simple. In this case, the beaming from the small aperture results in a poor localization of the source. Optimizing the time reversal mirror requires that there be some type of methodology in determining the position of the source and the certainty that it is indeed the original source location. A metric was developed in order to quantify the quality of localization, the source to field ratio (SFR). This metric is useful not only in its ability to compare quality of localization as a single parameter is varied, but it also allows for a comparison across many parameters and their effects on the quality.

In the reversal and backward propagation of the waveforms from the received locations, a high amplitude pressure is created at or near the point where the original source was located. The SFR measures the pressure amplitude at the point that the source was originally located and compares this to the average of the amplitude surrounding this location as

$$\text{SFR} = \frac{|F_{0,0}|}{\frac{1}{K}\sum_{i,j}|F_{i,j}|}. \tag{2.3}$$

In this equation, $F_{i,j}$ is the pressure at a given point of the field, the source is located at the origin $(0,0)$, and K is the number of field points within the field to be averaged. The field points are taken from $\lambda/4 \leq |x(i), y(j)| \leq 5\lambda/2$ for the case of a local field to be averaged with 5λ by 5λ sides. The field points are not taken for $|x(i), y(j)| \leq \lambda/4$ so as to not mix the source pressure location with the surrounding field and detract from the SFR. A high SFR suggests a strong focusing of energy, thus allowing localization. An example of this source and field is shown in Fig. 2.6. The TRM includes ten elements with total aperture π radians and an inter-element spacing (measured by angle) of $\pi/9$ radians. The TRM of Fig. 2.5 is only $\pi/4$ radians with an inter-element spacing of $\pi/36$ radians. In Fig. 2.6, the pressure at the source location is compared with the average value of the nearby field, or region of certainty. Applying equation 2.3 in this example, the SFR is 3.91, whereas the SFR for Fig. 2.5 is only 2.64.

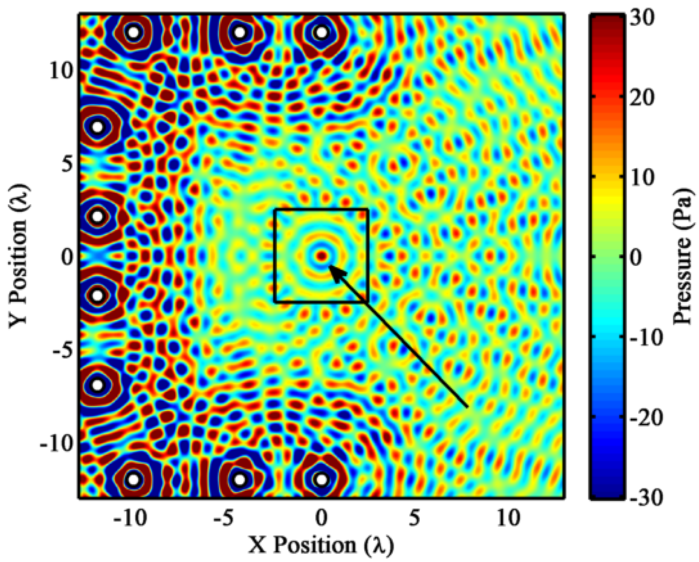

FIG. 2.6. Backward propagation wave field of time reversal mirror showing the source location (marked by the arrow) and region of certainty ($5\lambda \times 5\lambda$ black outline).

Chapter 3

Results and Analysis

This chapter is divided into sections for each parameter which was optimized in the study. It is organized in order of importance to the TRM setup for optimization and qualitative results. A few of the more important parameters to be discussed are the dependence of the SFR on the angular spacing and the angular aperture of the TRM. The *region of certainty*, frequency, shape and size of the TRM, and position of the source are also discussed.

According to prevailing theory and practice, half-wavelength spacing of TRM elements is necessary to avoid grating lobes, although this restriction can be relaxed if the source is pre-focused on the source.[1,15,17] There has been little or no discussion to quantify the extent to which the element separation distance may be increased. We assume that the source location is known to within a certain region. Results will be given for the case when the TRM layout is centered around the source, then in Section 3.3 results will be discussed which generalize our findings to a source which is moved off-center of the TRM layout.

Shown in Fig. 3.1(a) is a simple TRM centered about a simple source in a homogeneous half-space environment which broadcasts a continuous sine-wave in the forward propagation step. This is a general example of the setup used in the study. The TRM and source are coplanar, that is above, and parallel to the ground which is located in the *xy* plane,

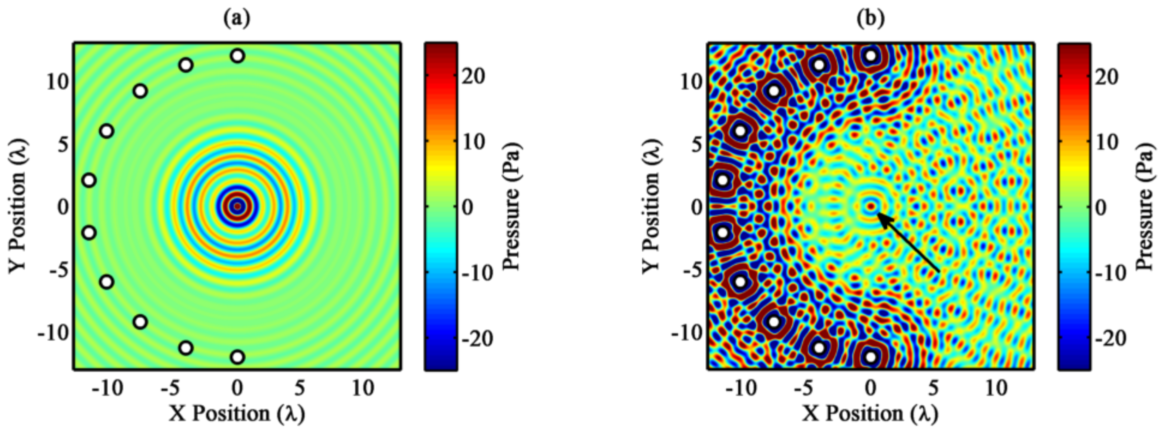

FIG. 3.1. Representative pressure wave fields from a time reversal computational model using 10 transducers for (a) the forward propagation and (b) the backward propagation step.

thus the wave field includes interference due to the ground reflection. The mirror elements (shown as white dots in the pressure field) record information received directly from the source as well as the reflection paths off the ground boundary. Note the interference null in the field at 6.9λ radius around the source where the direct and reflected wave fields cancel one another. In Fig. 3.1(b), the waveforms received are reversed in time and broadcast. In addition to other uncorrelated artifacts, this creates a localized pressure maxima at the location of the original source as shown by the red dot and the arrow. Ignoring the region near the TRM where the wave field is at higher amplitudes, the source location in this plot is the point of maximum pressure amplitude.

The pressure magnitude of the back propagation field is utilized for analysis instead of the real part of the pressure. This has advantage in pointing out features of the field without regard to the $e^{j\omega t}$ time dependence. An example of this advantage is shown in Fig. 3.2. Figure 3.2(a) shows the real part of the pressure wave field (see Eq. (2.2)) for the backward propagation step and Fig. 3.2(b) shows the pressure magnitude. In Fig. 3.2(b), the original source location is visibly more identifiable. Furthermore, the magnitude of the pressure at a particular field location is typically more desirable from experimental data.

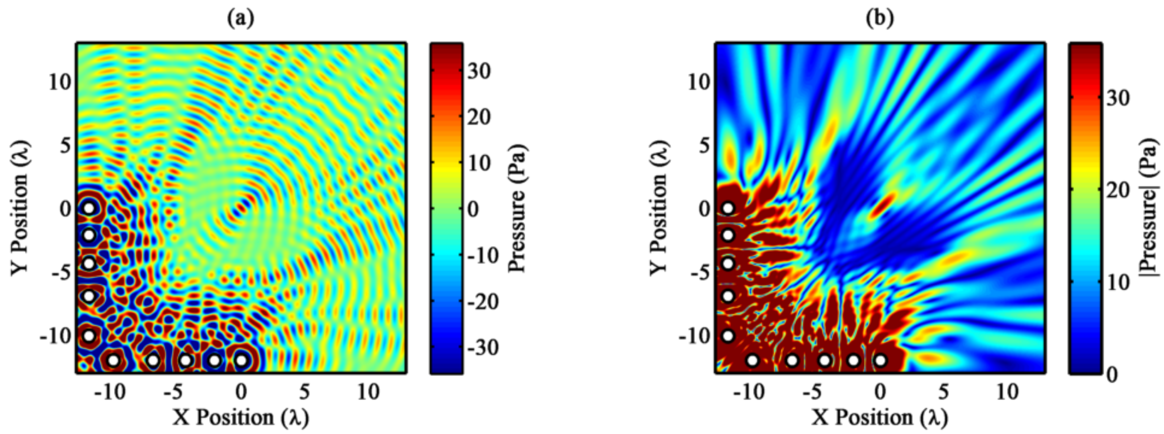

FIG. 3.2. Spatial maps of (a) the real part and (b) the magnitude of the pressure wave field from a sample time reversal model.

3.1 Dependence on Angular Spacing

Fink described the spacing of TRM elements as being a primary contributor to the quality of source localization.[1] Thus the main focus of this computational study was the qualification and further study of the dependency of the element spacing in the TRM to the SFR. As seen in Fig. 3.3, the backward propagation fields are plotted when three different TRM layout densities are used in similar conditions while the total angular aperture is kept constant. In this case, a circular TRM layout of radius 12λ centered about the source is used. The TRM elements are evenly distributed about a 2π radian circle surrounding the source. The region of certainty is a square with $5\lambda \times 5\lambda$ sides that is centered about the original source location. As the number of elements in the TRM increases, the field surrounding the immediate vicinity of the source location decreases in amplitude, increasing the localization quality. This effect is seen in Fig. 3.3. As additional TRM elements are used, the area of decreased amplitude surrounding the source increases. Once this area encompasses the region of certainty, the SFR arrives at a limiting value. The SFR for the cases shown in Fig. 3.3 are (a) 3.88, (b) 6.12, (c) 6.34, and (d) 6.34.

It was found that, in general, as the angular density of elements was increased, the

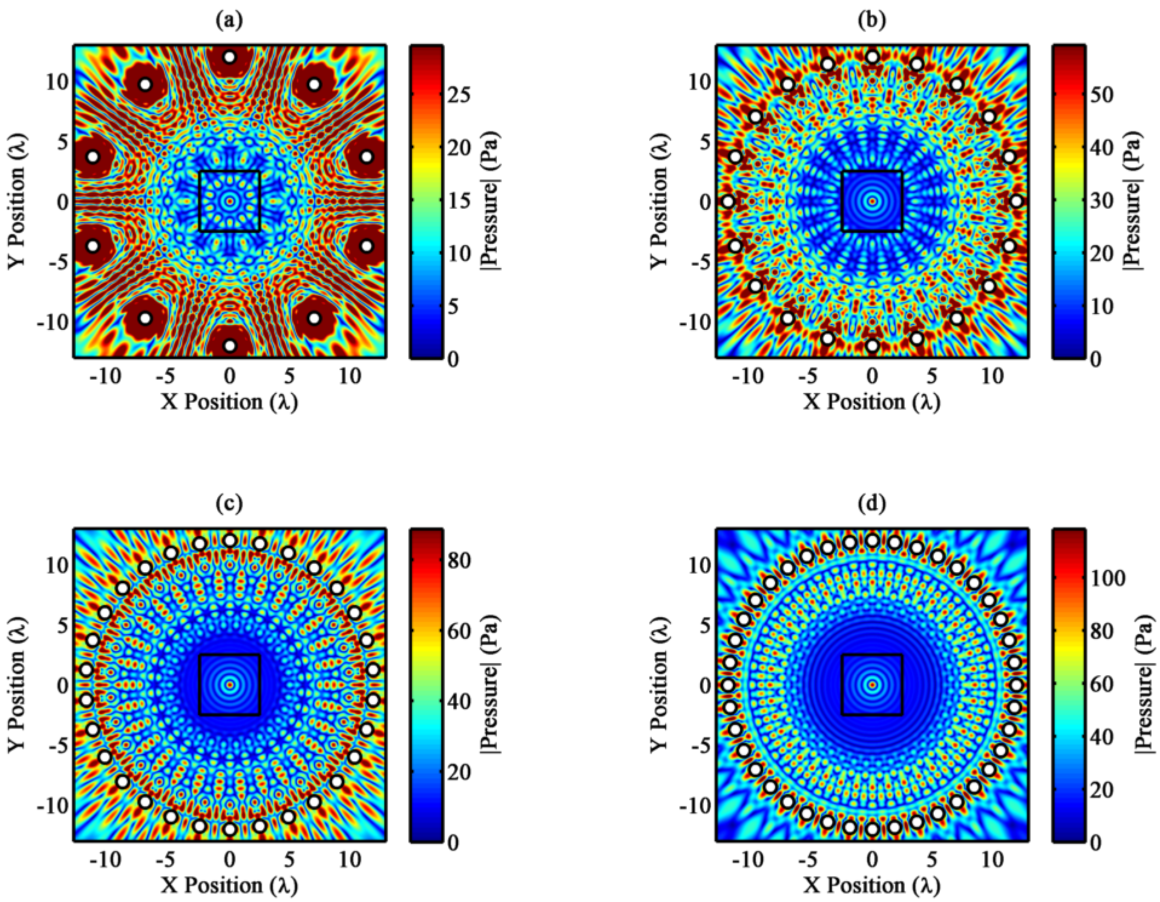

FIG. 3.3. The backward propagation wave fields for a time reversal mirror of (a) 10 elements, (b) 20 elements, (c) 30 elements, and (d) 40 elements.

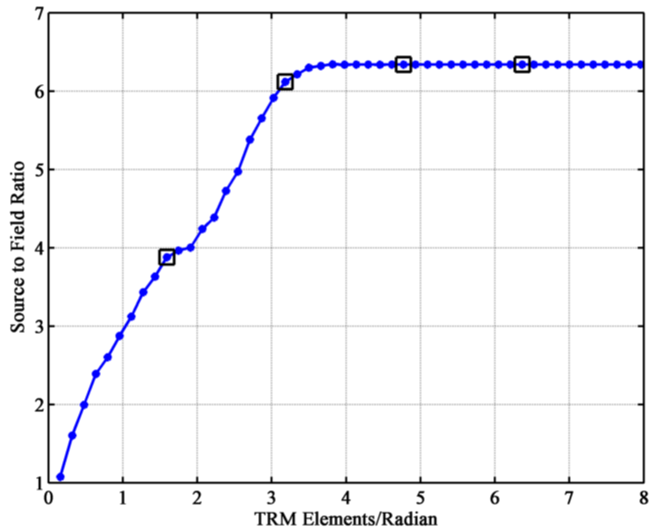

FIG. 3.4. Source to field ratio of for a time reversal mirror with a variable number of time reversal mirror elements given a set angular aperture vs. the number of mirror elements per radian. The TRM layout for the data shown is circular in shape and has a total angular aperture of 2π radians. The source to field ratio of the plots in Fig. 3.3 correspond to the box marker values.

quality of the localization increased up until it would reach a limiting value, irrespective of the shape of the TRM or the total angular coverage. Shown in Fig. 3.4 is the source to field ratio as a function of the TRM element spacing. Notice that for this particular set of models, a sufficient linear density of elements to optimize the SFR are approximately 3.5 elements per radian. Beyond this value, additional TRM elements do not increase the SFR for the given region of interest. This means that the optimal element spacing is about 3.4λ. An equivalent mirror with $\lambda/2$ spacing between each element would require 7 times as many TRM elements. In general, for a source centered on the TRM in the aforementioned conditions, the peak value of the SFR would require significantly fewer TRM elements than if a half-wavelength spacing criteria were used. This peak value, and the number of TRM elements sufficient to attain it, is affected by other parameters as discussed in the following sections.

We now show an additional example of a TRM that also does not require an angular

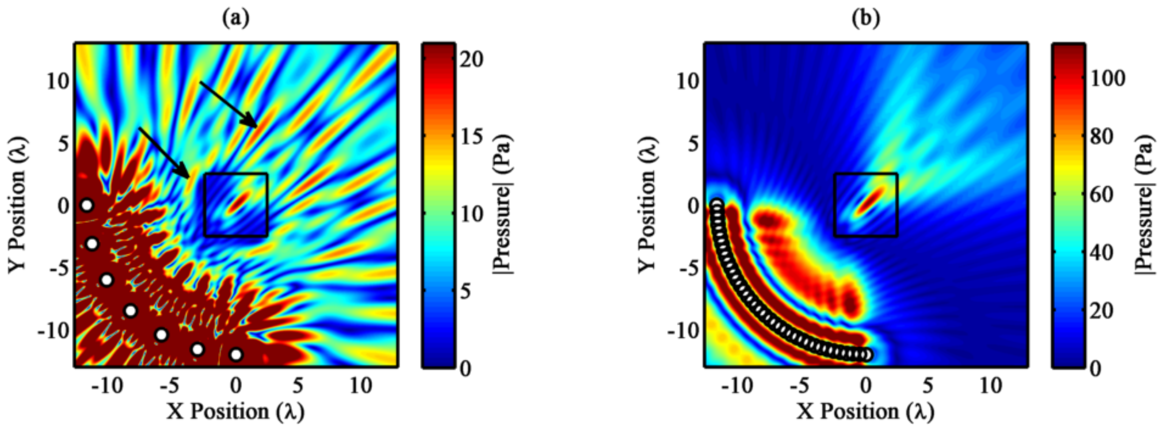

FIG. 3.5. Pressure magnitude spatial maps for a time reversal mirror with (a) 7 elements and (b) 37 elements. The time reversal mirrors have an angular aperture of $\pi/2$ radians.

spacing of $\lambda/2$. Figure 3.5 shows the backward propagation fields of a circular TRM layout with an angular aperture of $\pi/2$ radians. In Fig. 3.5(a), the element spacing is approximately 3λ and there are visible grating lobes type artifacts, some of which are identified by the arrows in the figure. Figure 3.5(b) shows the case where elements are spaced $\lambda/2$, there are no grating lobes, and the source location is easily discernible. However, note that both figures have similar fields within the region of certainty (denoted by the black box). Thus if the source location is known to within this region of certainty, a significantly fewer number of TRM elements is necessary in reconstructing the source location. A defined region of certainty of the source location will usually result in a significantly fewer number of TRM elements necessary to localize the source.

However, if the general source location is unknown, i.e. there is no region of certainty defined, then grating lobes result in features sufficiently similar to the true source localization as to prohibit proper source localization. This can be solved without the need of $\lambda/2$ element spacing through a larger angular aperture ($\geq 180°$ may be needed) as discussed in the following section. A large angular aperture refines the localization from a beam-like localization to a more point-like region. The original source location and other artifacts will be more point like in appearance. From this, an increased number of TRM elements will refine the wave field in the backward propagation step until the original source location is clearly identifiable. Geometric patterns also may aid in identifying the original source location as seen in Fig. 3.3, since features due to the grating lobes are symmetric about the original source location.

3.2 Dependence on Angular Aperture

The total aperture, or the angle which the TRM elements sweep out about the source, is often limited for practical purposes and we investigate here the effects that aperture would have on localization. In Fig. 3.6, an example is shown of different TRM layouts where the angular spacing is held constant and the aperture of the TRM is changed from $\pi/4$ radians to 2π radians about the source. As Fink *et al.* explains, the point spread function of the source localization is related to the angular aperture of the TRM.[15] As the angular aperture is increased, the localization becomes better resolved until the point spread function reaches the classical $\lambda/2$ diffraction limit.[2] In Fig. 3.6(a) the source location is more difficult to resolve due to the other artifacts in the wave field. In the low angular aperture regime, grating lobes are visible and symmetric about an axis from the TRM to the original source location. These make identification of the source difficult, however if these lobes are outside of the region of certainty the lobes do not impede the source localization. In the high aperture regime, as seen in Figs. 3.6(c) and 3.6(d), the source is visibly distinguishable and the classical diffraction limit of the source localization is the limiting factor. For the cases in Fig. 3.6(a-d) the SFR values are 2.13, 2.74, 3.61, and 4.97 respectively.

This limiting resolvability of the point spread function is also apparent in the SFR for

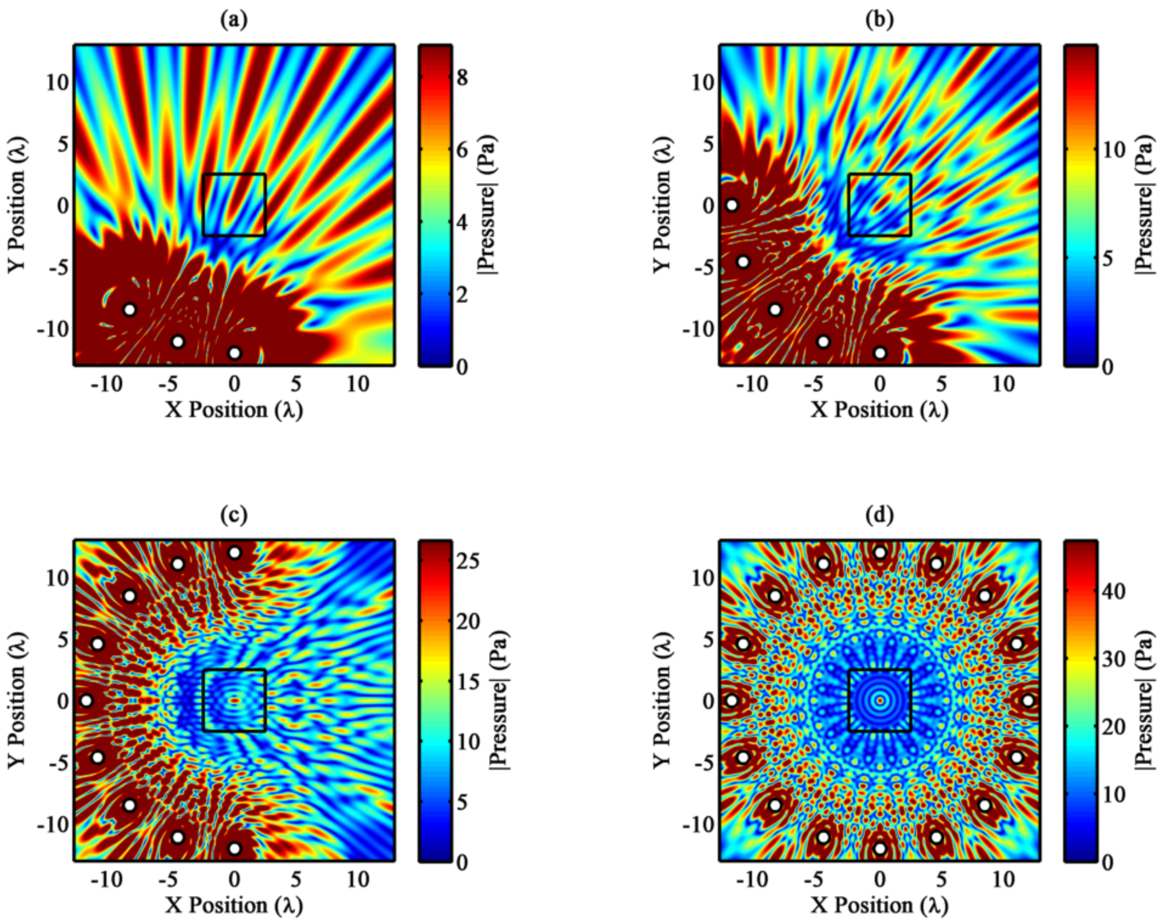

FIG. 3.6. Holding the angular spacing of TRM elements constant (2.55 time reversal mirror elements per radian), the aperture of the TRM is varied by (a) $\pi/4$ radians, (b) $\pi/2$ radians, (c) π radians and (d) 2π radians.

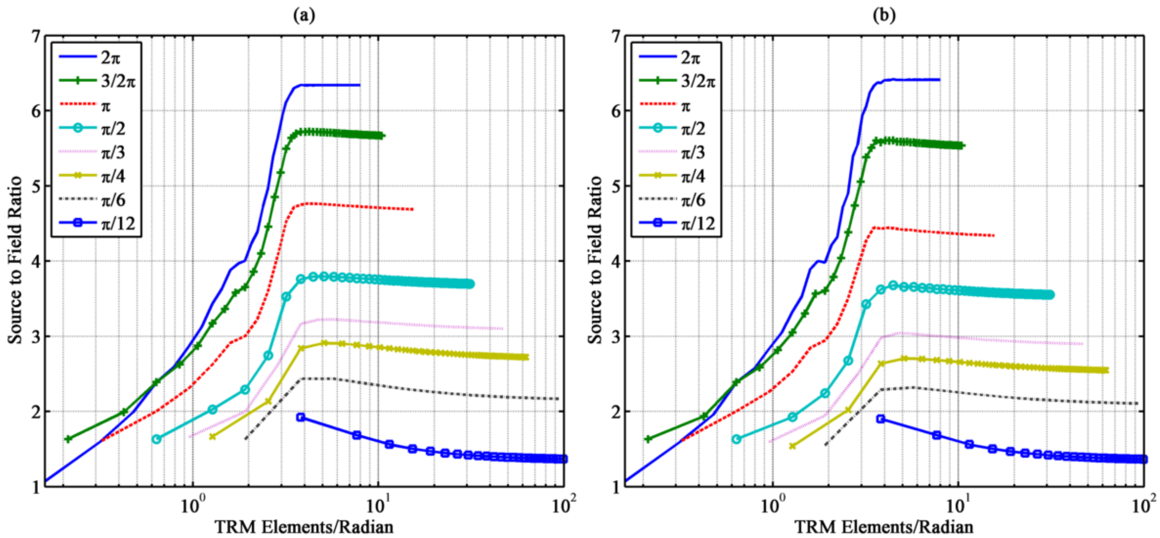

FIG. 3.7. Source to field ratio versus the number of mirror elements per radian for several different total angular coverages using (a) a circular time reversal mirror layout and (b) a square time reversal mirror layout.

these cases. Shown in Fig. 3.7 is a comparison of different TRM layouts with different total angular apertures while also varying the TRM element spacing for each fixed total aperture. As an example, the plots in Fig. 3.3, which each have a total aperture of 2π radians, correspond to the box marker values in Fig. 3.4, and the data from Fig. 3.4 may be seen in the blue solid line in Fig. 3.7(a). Figure 3.7(a) shows the results for circular TRM apertures while Fig. 3.7(b) shows the results for TRM layouts where the elements are arrayed in a square shape (the TRM shapes will be discussed in Section 3.3.3). If the total angular aperture is increased, there is a reduction in the point spread function near the source which yields a higher SFR. This general increase in resolvability of the source location due to an increased aperture was noted by O'Brien *et al.*[7] and Larmat *et al.*[18]

Of particular interest is the limiting value of the SFR in each trial. If the curves of Fig. 3.7 are normalized to their respective peaks as shown in Fig. 3.8, the optimal TRM spacing for each aperture is only minimally affected. The similar trend of each aperture suggests only a minimal dependence between the angular spacing of the elements and the

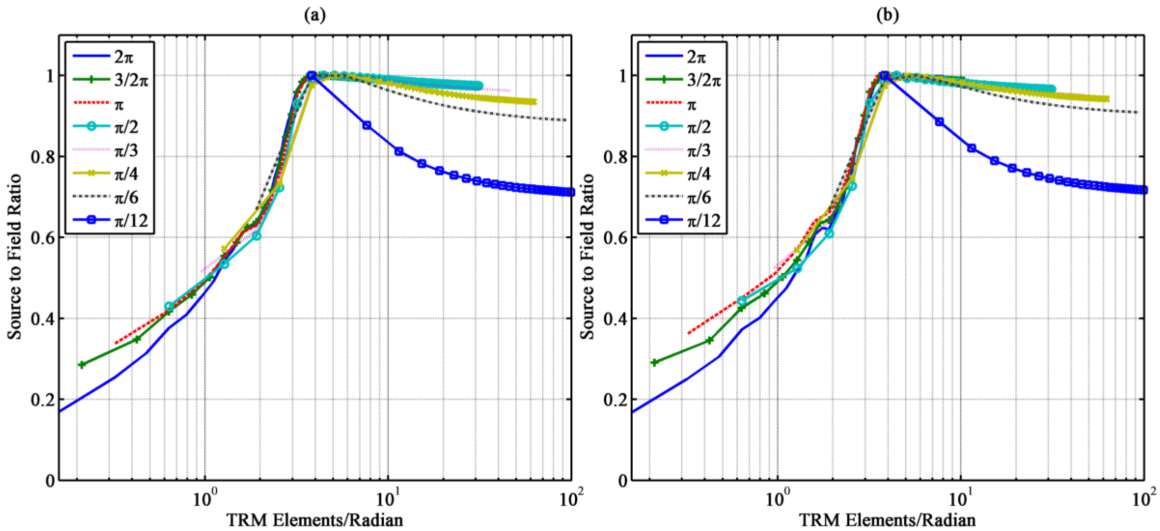

FIG. 3.8. Data from Fig. 3.7 which is normalized by the maximum source to field ratio of each set of data for (a) a circular time reversal mirror layout and (b) a square time reversal mirror layout.

angular aperture. As seen in Fig. 3.8, the value at which the SFR approaches a peak value is independent of the total aperture angle. For smaller aperture sizes, the effect of beaming in the direction of the source for TRMs of smaller total aperture does not necessarily result in a maximum pressure in the region of certainty, let alone at the actual source location (See Fig. 3.6(a)). Thus, in the lower limit of the aperture, the quality of localization does not increase with an increase in the angular spacing for small angular aperture TRMs (See Fig. 3.8) since the source region is not centered on the original source location. Therefore, a lower bound to the TRM aperture exists, dependent on the size of the region of certainty. We conclude that an increased total aperture angle, with fixed angular spacing between elements, increases the SFR resulting from a more optimal point spread function for the source reconstruction. Furthermore, given a region of certainty, in optimizing the peak value of each aperture we find that there is a fixed optimal spacing irrespective of the total aperture size for the TRM.

3.3 Other Dependencies

Further studies were done using other parameters that are taken into consideration when creating a TRM setup. These include the relative knowledge of the source location, the frequency of interest, the shape and size of the TRM layout, and the degree to which the TRM is centered on the source location. The following sections describe the effects each has in optimizing the TRM setup.

3.3.1 Region of Certainty

It has been found that when the area in which source is known to exist is increased (less certainty of the source location), the optimal number of TRM elements at which the peak SFR is reached increases proportionally. This shows a direct relationship between the knowledge of the source location and the necessary TRM element spacing. Stated another way, a better a priori knowledge of the location of the source directly corresponds to a reduced number of elements needed in order to optimize a given TRM layout.

3.3.2 Frequency

Given a TRM layout and a determined optimal angular spacing of TRM elements for a respective region of certainty, if the frequency of the source is decreased, there will also be a proportionate decrease in the optimal number of TRM elements required at which the quality of localization reaches a peak value. If the frequency of interest is increased, the TRM is optimized by either using additional elements or by gaining a better certainty of where the source is located. Thus for a certain TRM layout, there exists an upper cutoff frequency, above which the TRM element spacing is not optimal.

3.3.3 TRM Layout Shape

Different TRM layouts were tested to determine what effect each would have on the aforementioned results. Shown in Fig. 3.7(a) are the results of a parameterization study for circular layouts where each element is equally spaced and placed equidistantly from the source. In Fig. 3.7(b), each data set represents a square shaped mirror of a particular aperture size for the case that the angular aperture is 2π radians, or if it is a smaller aperture, the shape is a partial aperture of a square. The elements were equally spaced in terms of arc angle coverage, so that each part of the square shape had the same density of elements per unit angle. This allowed better comparison to the circular shaped layouts. In both cases, the optimal angular spacing of the elements is similar to one another, the circular shapes outperforming the square shapes by only an average of 3.5% for their respective peak values. As can be seen, there is some variation between the shapes, however the value where the quality factor is optimized remains the same. Thus the shape of the TRM layout does not appear to be very important.

3.3.4 TRM Radius

The TRM radius is the distance from a circular arc TRM to its geometric center. In the case of a square TRM layout, the TRM radius is defined as the distance from the square's center to its closest edge. As the radius of the TRM is varied, there was very little change in localization quality. For this reason, it is impractical to specify an optimal element spacing when optimizing the TRM layout for an experiment, since an element spacing when the mirror is one meter from the source will be ten times smaller than if the mirror is ten meters from the source, and this results in no change in the localization quality. Instead the number of elements per radian or angular density is found to be more useful for an optimization specification.

3.3.5 Moving the Source Off-Center

All previous results are determined for a source which is geometrically centered within the TRM. Cases where the source is moved off-center of the mirror are also considered. If the source is moved off-center from the TRM, the SFR varies with the apparent change in the angular aperture relative to the source position. If moving the source increases the angular aperture, the SFR increases. However, small deviations in the source position result in little effect on the optimal angular spacing of the elements. Shown in Fig. 3.9 are circular and square TRM layouts with an angular aperture of $\pi/2$ radians in 3.9(a) and 3.9(b) respectively, and π radians in 3.9(c) and 3.9(d) respectively. The figure depicts the resulting SFR when the source is moved from off center to various locations while the positions of TRM elements are fixed. Each value at different locations in the field represents the SFR for the field near the source at that location. The SFR increases as the source is moved closer to the mirror, which corresponds to an increased angular aperture. However, as the source continues to approach the mirror, the quality decreases as the high amplitude in the nearfield of each element relative to the desired focusing distorts the SFR.

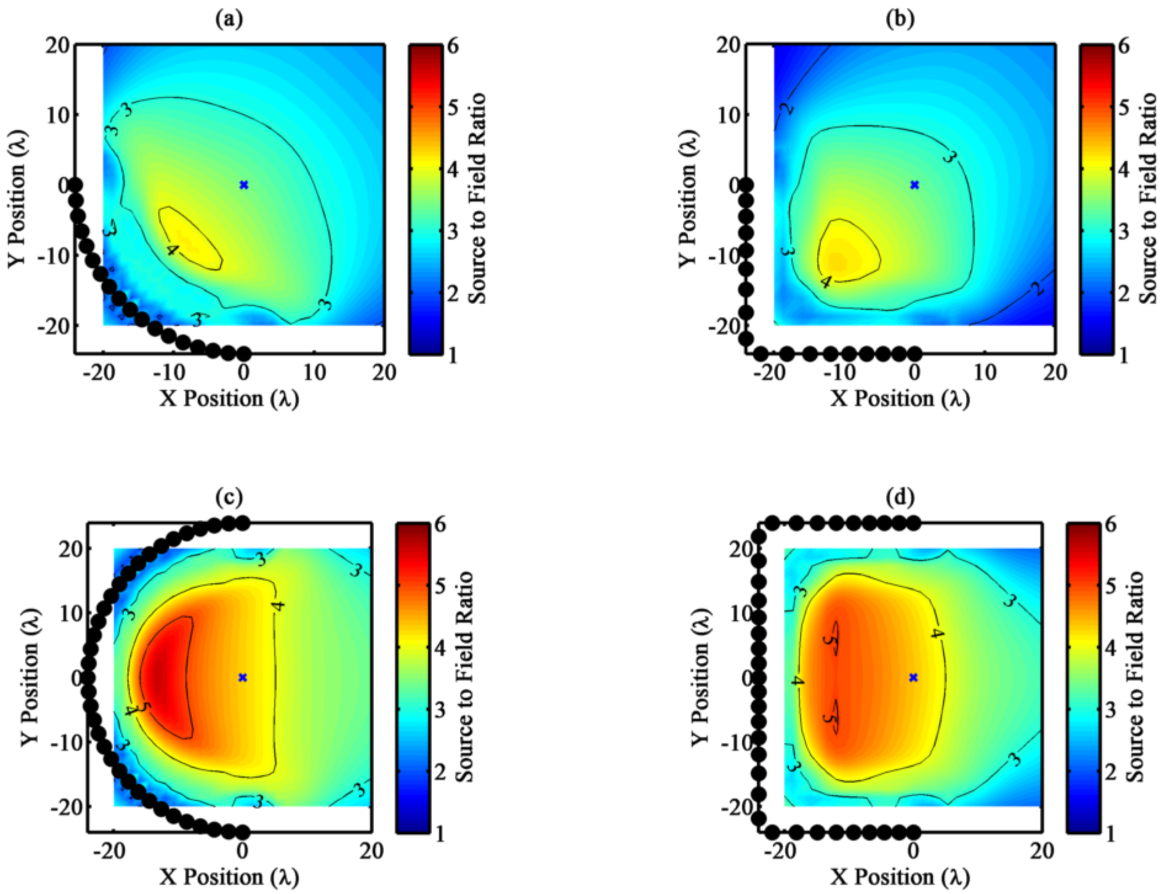

FIG. 3.9. Spatial maps of the source to field ratio when the source location is geometrically off-center of the time reversal mirror for (a) a circular time reversal mirror with $\pi/2$ radian aperture, (b) a square time reversal mirror with $\pi/2$ radian aperture, (c) a circular time reversal mirror with π radian aperture and (d) a square time reversal mirror with π radian aperture.

Chapter 4

Conclusions

In optimizing the layout of the TRM for a time reversal experiment it is found empirically that for a simple source emitting a CW signal in a half-space environment, the localization quality depends on the angular density of the TRM and the region of certainty. There is a peak localization quality value (SFR) that, depending on the relative knowledge of the source location (region of certainty), generally allows for relaxed conditions on the half-wavelength element spacing criteria suggested by Fink *et al*[15] and others. Furthermore, optimization of the TRM layout is dependent on an angular density of TRM elements with respect to the source location rather than an absolute distance. Grating lobes, which cause other artifacts in the wave-field and hinder proper source localization can be ignored to an extent, provided the source is within a region of certainty. We have shown that the angular spacing of elements is the most important parameter given a fixed angular aperture on the TRM.

By increasing the angular aperture for a given TRM element spacing, one will increase the SFR thus allowing more accurate localization, however this requires that the total number of elements be increased. In this manner, the angular density of the element spacing is preserved. An increased aperture has little effect of the optimal angular spacing of elements for an assumed region of certainty, except that an increased aperture reduces beaming effects that make the source location more difficult to identify since there is no temporally compressed reconstruction for a CW source as observed for TR experiments with pulsed sources.

Other parameters also contribute to the optimization of a TRM layout. As the region of certainty (i.e. relative knowledge of the source location) decreases in area, the angular density of TRM elements necessary to reach a peak SFR decreases proportionally. The frequency of the simple source is proportional to the number of TRM elements needed in a TRM layout to reach a peak SFR value. The shape of the TRM layout is only important inasmuch as it affects the angular density of the TRM elements. The TRM radius, or distance from the TRM layout to its geometric center, can be disregarded for idealized conditions (ignoring atmospheric absorption and signal to noise considerations for example) so long as the source does not lie within the nearfield of the TRM elements or at a null location in the field from direct and reflected interference of the source radiation. Finally, if the source is moved off of the geometric center of the TRM, the SFR is dependent on the TRM aperture relative to the new source location and whether the source is in a location where source reconstruction is more difficult. This may be a concern if the source is in the nearfield of the TRM elements or if the source is in a region of destructive interference caused by the ground reflection.

Further studies may be performed to increase understanding in applying TR to jet noise sources and other similar studies. This includes studying TR with an extended source,[19] understanding the effects on the TR process with a temperature gradient in the source region, and understanding the effects that non-linear propagation has on TR for jet noise sources, though some work has been done in this area for a different application.[20,21] With a better understanding regarding an optimized TRM layout, future experiments can be planned for increased efficiency. This allows for more effective use of the elements in any given experiment, be it jet noise or any similar half-space application.

Appendix A

Applying Time Reversal to Military Jet Noise

Military jet noise research is active in developing the understanding of how the jet engine and turbulence create and propagate sound. The near-field acoustic radiation and the source characteristics of military jet noise are not well understood. Because of this, various research approaches utilize different methods to gain new insight with each new approach. Theories on the nature of the sources of jet noise continue to be developed,[22] thus fundamental imaging studies increase knowledge which allows for a better engineering decisions in propulsion systems as well as safety guidelines for flight deck workers.

Jet noise is also an ever increasing concern for aircraft workers, airport communities and military personnel. During the 2010 fiscal year, over 1.4 million veterans received compensation due to hearing related damages while acting in service duties. While data are not limited to jet noise, it remains a primary culprit for hearing related damages, including tinnitus and hearing loss, which are the most prevalent service-connected disability for veterans.[23] This is an ongoing problem as the US military continues to anticipate for

FIG. A.1. Photograph of an F-22 Raptor held on the ground for acoustics testing.

permanent hearing damage of flight deck workers on aircraft carriers. Moreover, communities adjacent to airfields are constantly bombarded with the high intensity sounds from takeoffs. Increased sound levels from these aircraft can lead to declining real estate values.[24]

One of the challenges in this study is in dealing with a complex sound source which creates non-linear sound waves and has varying degrees of coherence. Because of this, many researchers have used different methods to analyze the data and better understand the sound source.[25–27]

Until now, there has been little research into the applicability of TR to jet noise. However, TR may prove to be effective in problems where beamforming and other ray-tracing techniques have limitations. A simple example and benchmark upon which other beamforming methods are compared is the delay-and-sum (DAS) method.[28,29] Here, the arrival times or phase information of array transducers are compared and from that the arrival signals are phased accordingly to propagate a beamed wave field in the direction of the source.

Even the simplest beamforming method however requires phasing calculations that TR can reproduce intrinsically. Beamforming techniques are also inhibited by reflective surfaces, while, as long as the environment is known, TR benefits from these reflections.

There are many limitations to the TR method however when working with jet noise. TR has historically found its use when dealing with transient signals since the TRM has the potential of spatially and temporally locating a source. Jet noise, although turbulent in nature, is more akin to a long duration source, whereby the source's time of emission is lost when using TR. Furthermore, a jet noise source is complicated in that it is best described as an extended source, and noise radiated has important finite amplitude propagation characteristics.[22,30] Temperature gradients in the propagation field are also an important consideration. Traditional methods of collecting jet noise data create further problems. Jet noise is usually recorded as the jet is on a hard surface in a half space, using an array which only partially surrounds the source. The temperature and size of the jet plume require that transducers recording the waveform be placed at a suitable distance from the plume and there are a limited number of microphones which record the waveform simultaneously.

With these considerations in mind, the purpose of this study is to gain a better understanding of the limitations and capabilities of TR in a perfectly rigid, half space environment using single frequency continuous waveforms with application to jet noise. As the potential of the method is realized, the development of useful procedures and techniques for utilizing TR with a jet noise source is possible.

Appendix B

MATLAB Code

B.1 Forward / Backward Propagation Step with Time Reversal

This code was used in the generation of most of the figures in the thesis. It calculates and plots the pressure fields of the forward or backward propagation steps of TR in a half-space environment using a simple source. The TRM can be a circular or square layout, aperture can be varied in terms of radians, the source to field ratio (SFR) is calculated in each iteration and stored in 'data'. Each iteration plots the pressure field. The pressure fields are vectorized which speeds up compuation time significantly.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%DESCRIPTION: FORWARD/BACKWARD TIME REVERSAL PROPAGATION
%AUTHOR: BLAINE HARKER (blaineharker at gmail dot com)
%INPUT:
%OUTPUT:
%SUBROUTINES:
%PROJECT: Simple time reversal in a half-space environment
%DATE: 05/09/2012
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Calculates and plots the pressure fields of the forward or backward
% propagation steps of TR in a half-space environment using a simple
```

```
% source. The TRM can be a circular or square layout, aperture can be
% varied in terms of radians, the source to field ratio (SFR) is calculated
% in each iteration and stored in 'data'. Each iteration plots the pressure
% field. The pressure fields are vectorized which speeds up compuation time
% significantly.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% INITIALIZE PARAMETERS
clear; close all;
tic
maxpoints=50;
param=[2,1.5,1,1/2,1/3,1/4,1/6,1/12]; %Different aperture sizes (radians/pi)

f=9000;                          %Frequency
A=1;                             %Amplitude
c=343;                           %Speed of Sound
T=1/f;                           %Period
w=2*pi*f;                        %Angular Frequency
k=w/c;                           %Wavenumber
lambda=c/f;                      %Wavelength
h=lambda/50;                     %Field Spacing
lam=lambda/h;                    %Points in one wavelength
t=0;                             %time (held constant for these plots)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Set default plot values for matlab session
set(0,'DefaultAxesFontName','Times New Roman');
set(0,'DefaultTextFontName','Times New Roman');
set(0,'DefaultAxesFontSize',14);  % for a paper, this should be 18
set(0,'DefaultTextFontSize',14);
set(0,'DefaultAxesFontWeight','demi')
set(0,'DefaultTextFontWeight','demi')
set(0,'DefaultAxesLineWidth',2); % for paper consider 2
set(0,'DefaultLineLineWidth',2);
set(0,'DefaultLineMarkersize',8); % for paper consider 8

xt=-13*lambda:h:13*lambda;       %Field of interest
yt=-13*lambda:h:13*lambda;       %Field Size adjusted for field being averaged
fsizex=length(xt);
fsizey=length(yt);

% INITIALIZE VARIABLES
Maxqual = zeros(maxpoints,2);     %Collects SFR for specific aperture

%'data' stores 'Maxqual' info before changing aperture size. It contains
%     data ( aperture run , SFR(TRM element #), 2 ) or
%     data ( aperture run , TRM Elements, 1 )
data=zeros(length(param),maxpoints,2);

trm.shape = 'square';            %TRM Shape 'circle' or 'square'
direction = 'backward';          %'forward' or 'backward' propagation step
for ww=1:length(param)           %loop through aperture sizes
    ang=param(ww);               %current aperture angle (radians)
    for points=1:maxpoints       %loop through number TRM elements in TRM

        [x y ~]=meshgrid(xt,yt,1:points); % Initialize pressure field

        z=1.5*lambda;
        xso=0;                   %Original Point Source
        yso=-12*lambda;          %(0,0) corresponds to xso=0,yso=-12*lambda
        zs=1.5*lambda;           %Source / TRM height above ground

```

```
        % TRM Element Location
        lxc=xso;                %X-Position center of Line Array
        ly=yso;                 %Y-Position center of Line Array
        lz=zs;                  %Z-Position center of Line Array

        % Arc Array Field
        r=12*lambda;            %Radius of TRM from its geometric center
        theta=ang*pi;           %aperture angle (radians)
        cent=r+ly;              %Y Distance from (lxc,ly) to circle center

        % Initialize arrays to calculate TRM element positions
        pa=zeros(length(points),3);
        pa2=zeros(length(points),3);
        pasquare=zeros(length(points),3);
        for i=1:points
            if points == 1
                pa(i,:)=[r,(i-1)*theta/(points)-(theta+pi/2) lz];
            elseif ang == 2
                pa(i,:)=[r,(i-1)*theta/(points)-(theta+pi/2) lz];
            else
                pa(i,:)=[r,(i-1)*theta/(points-1)-(theta+pi/2) lz];
            end

            if strcmp(trm.shape,'circle')
                % Calculate TRM element positions for a circular TRM
                pa2(i,:)=[lxc+pa(i,1)*cos(pa(i,2)),...
                    cent+pa(i,1)*sin(pa(i,2)),pa(i,3)];
            elseif strcmp(trm.shape,'square')
                % Calculate TRM element positions for a circular TRM
                %Divide up angles into eight groups
                %    (zero is south and + angles move ccwise)
                if pa(i,2)>= -5*pi/2 && pa(i,2)<-9*pi/4 % 0-45
                    pasquare(i,:)=[lxc+r*sqrt(1/(sin(pa(i,2))^2)-1),...
                        cent-r, pa(i,3)];
                elseif pa(i,2)>= -9*pi/4 && pa(i,2)<-2*pi % 45-90
                    pasquare(i,:)=[lxc+r,...
                        cent-r*sqrt(1/(cos(pa(i,2))^2)-1), pa(i,3)];
                elseif pa(i,2)>=-2*pi && pa(i,2) < -7*pi/4    % 90-135
                    pasquare(i,:)=[lxc+r,...
                        cent+r*sqrt(1/(cos(pa(i,2))^2)-1), pa(i,3)];
                elseif pa(i,2)>=-7*pi/4 && pa(i,2) < -3*pi/2    % 135-180
                    pasquare(i,:)=[lxc+r*sqrt(1/(sin(pa(i,2))^2)-1),...
                        cent+r, pa(i,3)];
                elseif pa(i,2)>= -3*pi/2 && pa(i,2)< -5*pi/4 % 180-225
                    pasquare(i,:)=[lxc-r*sqrt(1/(sin(pa(i,2))^2)-1),...
                        cent+r, pa(i,3)];
                elseif pa(i,2)>=-5*pi/4 && pa(i,2) < -pi  %225-270
                    pasquare(i,:)=[lxc-r,...
                        cent+r*sqrt(1/(cos(pa(i,2))^2)-1), pa(i,3)];
                elseif pa(i,2)>=-pi && pa(i,2) < -3*pi/4  %270-315
                    pasquare(i,:)=[lxc-r,...
                        cent-r*sqrt(1/(cos(pa(i,2))^2)-1), pa(i,3)];
                else    %315 - 360
                    pasquare(i,:)=[lxc-r*sqrt(1/(cos(pa(i,2)-pi/2)^2)-1),...
                        cent-r, pa(i,3)];
                end
            else
                error('Abort: Wrong TRM Shape')
            end
        end
        if strcmp(trm.shape,'square')
            %Individual X,Y,Z components of each Element
            x0=pasquare(:,1);y0=pasquare(:,2);z0=pasquare(:,3);
```

```
        elseif strcmp(trm.shape,'circle')
            %Individual X,Y,Z components of each Element
            x0=pa2(:,1);y0=pa2(:,2);z0=pa2(:,3);
        end

        % Vectorized TRM elements
        [~,~,x0grid]=meshgrid(1:fsizex,1:fsizey,x0);
        [~,~,y0grid]=meshgrid(1:fsizex,1:fsizey,y0);
        [~,~,z0grid]=meshgrid(1:fsizex,1:fsizey,z0);
        % data2=zeros(201,201,1,4);

        %There is code to calculate the SFR as the source is moved at
        %    different locations in the field. Default position (center)
        %    is x = 1, y = 51;
        %for this project the source is centered relative to the TRM
        for xx=1%26:201  % 15 wavelengths at 1/10 lambda incriments
            for yy=51%51:201
                xs=(xx-1)/10*lambda;          %Current source location (x)
                ys=cent;%(yy-1)/10*lambda;    %Current source location (y)

                %find source indices on the field
                xfieldp = find(xt>=xs,1,'first');
                xfieldn = find(xt<xs,1,'last');
                yfieldp = find(yt>=ys,1,'first');
                yfieldn = find(yt<ys,1,'last');
                if abs(xt(xfieldp)-xs)>abs(xt(xfieldn)-xs)
                    cpx = xfieldn;
                else cpx=xfieldp;
                end
                if abs(yt(yfieldp)-ys)>abs(yt(yfieldn)-ys)
                    cpy = yfieldn;
                else cpy=yfieldp;
                end

                %Define indices of the bounds where the field near the
                %    source will be averaged.
                boxLength = 5.0*lam;
                fl=cpx - boxLength/2;
                fr=cpx + boxLength/2;
                ft=cpy + boxLength/2;
                fb=cpy - boxLength/2;
                cpos=[ft-cpy,fr-cpx];

                %Define indices of the bounds that will be discluded from the
                %    field average (source location).
                sboxlength = .5*lam;
                sourcel=cpx - round(sboxlength/2);
                sourcer=cpx + round(sboxlength/2);
                sourceb=cpy - round(sboxlength/2);
                sourcet=cpy + round(sboxlength/2);

                %Forward propagation step amplitude (B) and phase(theta)
                %    for direct and reflected paths
                B1=A./sqrt((x0grid-xs).^2+(y0grid-ys).^2+(zs-z0grid).^2);
                B2=A./sqrt((x0grid-xs).^2+(y0grid-ys).^2+(zs+z0grid).^2);
                theta1=k*sqrt((x0grid-xs).^2+(y0grid-ys).^2+(zs-z0grid).^2);
                theta2=k*sqrt((x0grid-xs).^2+(y0grid-ys).^2+(zs+z0grid).^2);

                %Calculate the field of the forward propagation step
                [xfor yfor]=meshgrid(xt,yt);
                zfor = z0grid(1,1,1);
                ps1 = A./sqrt((xfor-xs).^2+(yfor-ys).^2+(zs-zfor).^2).*...
                    exp(-1i*(k*sqrt((xfor-xs).^2+(yfor-ys).^2+(zs-zfor).^2)));
```

```
            ps1 = ps1 + A./sqrt((xfor-xs).^2+(yfor-ys).^2+(zs+zfor).^2).*...
                exp(-1i*(k*sqrt((xfor-xs).^2+(yfor-ys).^2+(zs+zfor).^2)));

            %Remove infinite field points
            for i=1:length(yt)
                for ii=1:length(xt)
                    if isinf(ps1(i,ii))
                        ps1(i,ii)=0;
                    end
                end
            end

            %Backward propagation step partial pressure fields for:
            %    direct,direct / reflected,direct / direct,reflected /
            %    and reflected / reflected arrival paths
            ps1prime=B1./sqrt((x-x0grid).^2+(y-y0grid).^2+...
                (z-z0grid).^2).*exp(-1i*(w*t-k*sqrt((x-x0grid).^2+...
                (y-y0grid).^2+(z-z0grid).^2)+theta1));
            ps1prime=ps1prime+B2./sqrt((x-x0grid).^2+(y-y0grid).^2+...
                (z-z0grid).^2).*exp(-1i*(w*t-k*sqrt((x-x0grid).^2+...
                (y-y0grid).^2+(z-z0grid).^2)+theta2));
            ps1prime=ps1prime+B1./sqrt((x-x0grid).^2+(y-y0grid).^2+...
                (z+z0grid).^2).*exp(-1i*(w*t-k*sqrt((x-x0grid).^2+...
                (y-y0grid).^2+(z+z0grid).^2)+theta1));
            ps1prime=ps1prime+B2./sqrt((x-x0grid).^2+(y-y0grid).^2+...
                (z+z0grid).^2).*exp(-1i*(w*t-k*sqrt((x-x0grid).^2+...
                (y-y0grid).^2+(z+z0grid).^2)+theta2));

            %Plot either forward or backward propagation step
            if strcmp(direction,'forward')
                field = ps1;
            elseif strcmp(direction,'backward')
                field=ps1prime;
            end
            clear ps1prime B1 B2 theta1 theta2;

            %Sum vectorized portions of the field
            field=sum(field,3);

            %Remove infinite field points
            for i=1:length(yt)
                for ii=1:length(xt)
                    if isinf(field(i,ii))
                        field(i,ii)=0;
                    end
                end
            end

            % Calculate amplitude at source location
            centermax=abs(field(cpy,cpx));


            % Initialize temporary field
            fieldA=zeros(length(field(:,1)),length(field(1,:)));
            % Field of only source location area
            fieldA(sourceb:sourcet,sourcel:sourcer)=...
                field(sourceb:sourcet,sourcel:sourcer);
            % Initialize temporary field
            fieldB = field - fieldA;
            %Take field near source (box) minus the pressure at source
            ftot=fieldB(fb:ft,fl:fr);
            %Average field value
            fieldavg=mean(mean(abs(abs(ftot))));
```

```
                % Source to field ratio (SFR)
                ratio=centermax/fieldavg;

                clc; display(ww); disp(points);%display(xx); display(yy);
            end
        end

        %Store TRM element number and SFR for iteration
        Maxqual(points,:)=[points ratio];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % Plotting

        % %Plot Source and Field
        convNewPlot = h/h;
        ppoints=(max(max(real(field)))+20).*ones(length(x0));

        % Calulate Region of certainty outline
        xp=[xs/lambda-2.5*convNewPlot,xs/lambda-2.5*convNewPlot,...
            xs/lambda+2.5*convNewPlot,xs/lambda+2.5*convNewPlot];
        yp=[ys/lambda-2.5*convNewPlot,ys/lambda+2.5*convNewPlot,...
            ys/lambda+2.5*convNewPlot,ys/lambda-2.5*convNewPlot];
        zp=[max(max(real(field))) max(max(real(field))),...
            max(max(real(field))) max(max(real(field)))];

        % Create the Absolute Field Using the same metrics
        figure('Visible','on')
        hold on
        % surf(xt/lambda,yt/lambda,abs(abs(field))); %Plot pressure mag.
        surf(xt/lambda,yt/lambda,real(field)); %Plot real pressure mag.
        shading interp

        % TRM POSITION
        plot3(x0./lambda,y0./lambda,ppoints,...
            'ko','markersize',8,'markerfacecolor','w')

        % Region of certainty outline
        line(xp(1:2),yp(1:2),zp(1:2),'LineWidth',2,'Color','k');
        line(xp(2:3),yp(2:3),zp(2:3),'LineWidth',2,'Color','k');
        line(xp(3:4),yp(3:4),zp(3:4),'LineWidth',2,'Color','k');
        line([xp(4),xp(1)],[yp(4) yp(1)],[zp(4) zp(1)],...
            'LineWidth',2,'Color','k');

        axis image;
        colorbar
        xlabel('X Position (\lambda)'); %Set to 18 if main, 24 if subfigure
        ylabel('Y Position (\lambda)');

        caxis([-max(max(abs(ftot))) max(max(abs(ftot)))])
        box on
        hold off

    end
    % Save Maxqual data for aperture iteration
    data(ww,:,:)=Maxqual;
end
toc

% [EOF]
```

B.2 Parameter Iteration Testing

This code tests the various parameters that were mentioned in Chapter 3. This includes the angular spacing of elements, TRM shape, radius of TRM, aperture, frequency, and region of certainty. This code is optimized in order to iterate over the selected parameter for multiple conditions. It was designed for use on a workstation computer with 50GB of memory. As such, many of the variables were vectorized which significantly decreased computation time but required more memory.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%DESCRIPTION: QUALITY FACTOR MODELING
%AUTHOR: BLAINE HARKER (blaineharker at gmail dot com)
%INPUT:
%OUTPUT: ( OPTIONAL - LOGFILE, FIGURES, DIARY FILE )
%SUBROUTINES:
%PROJECT: Time Reversal applied to Jet Noise
%DATE: 07/28/11
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% This file tests the different quality factors for a variety of situations
% and determines how the quality factor is affected as a function of a
% distance from the number of TRM Elements, TRM center, TRM type, TRM
% angular aperture, frequency and field-box size.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% INITIAL PARAMETERS
 clear; close all;clc
 fprintf('\t\t-\t-\t-\t-\n\t\tBegin Program\n\t\t-\t-\t-\t-\n');

% NUMBER OF TRM ELEMENTS
Trm.Num = 12;

% TRM TYPE (STRING)
% (eg 'circle','line','modified_square')
Trm.Type = 'modified_square';

% TRM CENTER
% (IF A CIRCLE, IT WILL BE THE CIRCLE CENTER)
Trm.Center.x = 0;        % [m]
Trm.Center.y = 0;        % [m]
Trm.Center.z = 1.5*(343/9000);

% TRM ANGULAR APERTURE
% (ABOUT TRM CENTER)
Trm.AngAperture = pi/2;    % [Radians]

% MIRROR TO CENTER DISTANCE
% (IF A CIRCLE, IT WILL BE THE RADIUS)
Trm.Dist = 1.0;   % [m]
```

```

% LINE ARRAY CONSTANT ANGLE / CONSTANT INTERELEMENT SPACING (BOOLEAN)
% (FOR CONSTANT ANGLE - CHOOSE 1 , FOR CONSTANT SPACING - CHOOSE 2)
% (NOTE - IF Trm.AngAperture >= pi IT WILL BE MODIFIED TO A QUASI-INF LINE)
Trm.FixLineAngle = 1;

% LINE ARRAY ELEMENT SPACING
% (FOR LINE ARRAY USE ONLY)
Trm.LineArraySpacing = 5.00;    % [Wavelengths]

% FREQUENCY
f = 9000;  % [Hz]

% SOURCE SHIFT
%   (define the location where the source will be placed (relative to the
%   TRM Center. If a vector is given, seperate results are given for each
%   position)
Source.x = 0;%-20:4/3:20;                % [WAVELENGTHS]
Source.y = 0;%-20:4/3:20;                % ["]
Source.z = 0;%[0 -1.5 -3.0 -4.5 -(1/(343/9000))]; % ["] (SINGLE VALUE)

% Z-POSITION OF FIELD OF INTEREST
% (if length(Source.z) > 1 this value is replaced with z = zs )
z = Trm.Center.z;

% ITERATE THROUGH ONE OF INITIAL PARAMETERS (OTHER THAN SOURCE SHIFT)
% (STRING) (EG 'Trm.AngAperture')
Trm.iterPar = '';     % TYPE VARIABLE NAME OR LEAVE BLANK

% QUALITY FACTOR MEASUREMENT BOOLEANS
FieldAverage.GetQ = 1;      % Collect Quality Factor Information
FieldAverage.GetPeak = 1;   % Collect Source to Peak Ratio Information

% FIELD BOX SHAPE
% (SHAPE OF THE AREA WHERE FIELD WILL BE AVERAGED)(STRING)
FieldAverage.Shape = 'square';          % 'circle' or 'square'

% FIELD BOX SIDE TO CENTER DISTANCE
% (L/2 FOR FIELD SQUARE, R FOR FIELD CIRCLE)
FieldAverage.SetBoxSize = 1;    % Sets the box size relative to 9 kHz
FieldAverage.BoxSide = 5.0;   % [Wavelengths]

% DISCLUDE 1/4 WAVELENGTH BOX AROUND SOURCE IN FIELD AVERAGE? (BOOLEAN)
FieldAverage.MinusSource = 1;

% FIGURE BOOLEANS
% (Choose 1 for True, 0 for False)
fig.draw = 1;   % Plot the field
fig.visual = 1; % Display plot on screen (0 creates an invisible plot)
fig.iter = 0;   % Plot in iteration mode (closes figures at the end of loop)
fig.iterX = 0;  % Plot in Xtreme iteration mode (creates only field needed)

% SAVE BOOLEANS
% (Choose 1 to save data, 0 to supress saving data)
saveIt.data = 0;
saveIt.fig = 0;
saveIt.logfile = 0;

% LOGFILE COMMENTS
logme.my_comments = '[Enter Comments Here]';

% SAVE FILE LOCATION
saveIt.file = 'Location_to_save_data';
```

```
saveIt.date = 'Current_Data';
saveIt.ID = 'Desired_ID';

% COMPLETION NOTIFICATION BOOLEANS
% (Choose 1 to notify, 0 to supress notification)
done.email = 0;
done.text = 0;
done.sound = 0;

%___SET FIGURE PROPERTIES FOR REMAINDER OF MATLAB SESSION___%
set(0,'DefaultAxesFontName','Arial');
set(0,'DefaultAxesFontSize',14);
set(0,'DefaultAxesFontWeight','demi')
set(0,'DefaultAxesLineWidth',1.5);
set(0,'DefaultLineLineWidth',2);
set(0,'DefaultLineMarkersize',8);
set(0,'DefaultFigurePosition',get(0,'ScreenSize'));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% EXECUTE INITIAL SEQUENCES

% MAKE DIRECTORY
if saveIt.data || saveIt.logfile || saveIt.fig

    % CREATE DATE FOLDER
    if  exist([saveIt.file,'/',saveIt.date],'file') == 0
        mkdir([saveIt.file,'/',saveIt.date]);
    end

    % CREATE ID FOLDER
    if exist([saveIt.file,'/',saveIt.date,'/',saveIt.ID],'file') == 0
        mkdir([saveIt.file,'/',saveIt.date,'/',saveIt.ID]);
    end
end

if saveIt.logfile

    % TURN DIARY ON
    diary([saveIt.file,'/',saveIt.date,'/',saveIt.ID,'/',saveIt.ID,...
        '_diary.out'])
    disp('Diary is Recording');
end

% CHECK IF ID ALREADY EXISTS
if exist([saveIt.file,'/',saveIt.date,'/',saveIt.ID,'/',saveIt.ID,'.mat'],...
        'file') == 2 && (saveIt.data || saveIt.fig || saveIt.logfile)
    cont = input('The ID Being Used Already Exists. Continue ? y/n: ','s');
    if strcmp(cont,'n') || strcmp(cont,'N')
        return
    end
    clear cont
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MANY ITERATION INITIALIZATIONS

% PREALLOCATE ARRAYS AND INITIALIZE DUMMY VARIABLES
if isempty(Trm.iterPar)
    Trm.iterPar = 'f';
end
eval(['iterPar = ',Trm.iterPar,';']);
figNote = 0;
```

```
ttime = zeros(length(iterPar),...
    length(Source.x)*length(Source.y)*length(Source.z));
loc_quality = zeros(length(iterPar),length(Source.x),length(Source.y));
peak_quality = zeros(length(iterPar),length(Source.x),length(Source.y));
progress = 0;

% ITERATE THROUGH AN INITIAL PARAMETER IF DESIGNATED
for nn = 1:length(iterPar)
    % RUN THROUGH EACH OF THE PARAMETERS INDIVIDUALLY
    if nn == 1
        eval(['iterAll = ',Trm.iterPar,';']);
    end
    eval([Trm.iterPar,' = ','iterAll(',num2str(nn),');']);

    % NOTE OF MULTIPLE ITERATIONS
    if length(iterPar) > 1
        disp(['Iterating through parameter ''',Trm.iterPar,''' ',...
            num2str(length(iterPar)),' Times:']);
        disp(['Current Iteration : ',num2str(nn),' / ',...
            num2str(length(iterPar))]);
    end

    % FIGURE SAVE WARNING
    if length(iterPar) > 1 && figNote < 1 && saveIt.fig
        disp('Note: Figures will only be saved for the first iteration');
        cont = input('Continue ? y/n : ','s');
        if strcmp(cont,'n') || strcmp(cont,'N')
            return
        end
        figNote = 1;
    end

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % DEFINE SOURCE AND FIELD OF INTEREST

    % SIMPLE SOURCE
    A=1;                    %Amplitude
    c=343;                  %Speed of Sound
    T=1/f;                  %Period
    w=2*pi*f;               %Angular Frequency
    k=w/c;                  %Wavenumber
    lambda=c/f;             %Wavelength
    t = 0;                  %Time

    % FIELD PARAMETERS
    fieldvar.side = 2.0;    %Length of Field Edges [m]
    fieldvar.h = lambda/50; %Field Spacing [m]

    % XTREME ITERATION MODE (SMALLER FIELDS)
    if fig.iterX

        % CHECK IF USING PC (SUPERCOMPUTER SKIPS THIS)
        if ispc

            % MEMORY ERROR CHECK
            [mem.userv mem.sysv] = memory;
            mem.smallfield = (2*FieldAverage.BoxSide*lambda/fieldvar.h)^2;
            mem.req = mem.smallfield*Trm.Num*8 * (5) + mem.smallfield*16 + ...
                768e6 + 2e9;
            if mem.req > mem.sysv.PhysicalMemory.Available
                ME = MException('VerifyInput:Limit',...
                    'Not Enough Available Physical Memory');
                throw(ME);
```

```
            end
            clear mem
        end

    else

        % FIELD RANGES
        xt = Trm.Center.x-fieldvar.side/2:fieldvar.h:Trm.Center.x +...
            fieldvar.side/2;
        yt = Trm.Center.y-fieldvar.side/2:fieldvar.h:Trm.Center.y +...
            fieldvar.side/2;
        zt = Trm.Center.z;

        % CHECK IF USING PC (SUPERCOMPUTER SKIPS THIS)
        if ispc

            % MEMORY ERROR CHECK (For the Dell Workstation Used, about
            %      40GB Memory limit)
            [mem.userv mem.sysv] = memory;
            mem.req = length(xt)*length(yt)*Trm.Num*8 * (5) +...
                length(xt)*length(xt)*16 + 768e6 + 2e9;
            if mem.req > mem.sysv.PhysicalMemory.Available
                ME = MException('VerifyInput:Limit',...
                    'Not Enough Available Physical Memory');
                throw(ME);
            end
            clear mem
        end

        % DEFINE VECTORIZED FIELD
        fieldvar.size.x = length(xt);
        fieldvar.size.y = length(yt);
        [x y ~] = meshgrid(xt, yt, 1:Trm.Num);

    end

    % WAVELENGTH -> FIELD SPACING CONVERSION
    fieldvar.conv = lambda / fieldvar.h;

    % CONVERT BOX SIDE: [WAVELENGTHS] -> [M]
    if FieldAverage.SetBoxSize
        FieldAverage.BoxSideM = FieldAverage.BoxSide*(343/9000);
    else
        FieldAverage.BoxSideM = FieldAverage.BoxSide*lambda;
    end

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % DEFINE TIME REVERSAL MIRROR

    % CIRCLE OR MODIFIED SQUARE TRM CONSTRUCTION
    if strcmp(Trm.Type,'circle') || strcmp(Trm.Type,'modified_square')

        setup.pa = zeros(Trm.Num,3);
        setup.pa2 = zeros(Trm.Num,3);

        % LOOP THROUGH TRM ELEMENTS
        for i = 1:Trm.Num
            if Trm.Type == 1
                setup.pa(i,:)=[Trm.Dist,(i-1)*Trm.AngAperture/Trm.Num-...
                    (Trm.AngAperture+pi/2),...
                    Trm.Center.z];
            elseif Trm.AngAperture == 2*pi
                setup.pa(i,:)=[Trm.Dist,(i-1)*Trm.AngAperture/Trm.Num-...
```

```
                (Trm.AngAperture+pi/2),Trm.Center.z];
        else
            setup.pa(i,:)=[Trm.Dist,...
                (i-1)*Trm.AngAperture/(Trm.Num-1)-...
                (Trm.AngAperture+pi/2),Trm.Center.z];
        end

        % SQUARE TRM
        if  strcmp(Trm.Type,'modified_square')

            setup.papasquare=zeros(Trm.Num,3);

            %DIVIDE ANGLES INTO 8 GROUPS
            % (zero is south and + angles move ccwise)
            if setup.pa(i,2)>= -5*pi/2 && setup.pa(i,2)<-9*pi/4 % 0-45
                setup.pasquare(i,:)=[...
                    Trm.Center.x +...
                    Trm.Dist*sqrt(1/(sin(setup.pa(i,2))^2) - 1),...
                    Trm.Center.y-Trm.Dist,...
                    setup.pa(i,3)];
            elseif setup.pa(i,2)>= -9*pi/4 && setup.pa(i,2)<-2*pi % 45-90
                setup.pasquare(i,:)=[...
                    Trm.Center.x+Trm.Dist, Trm.Center.y-...
                    Trm.Dist*sqrt(1/(cos(setup.pa(i,2))^2) - 1),...
                    setup.pa(i,3)];
            elseif setup.pa(i,2)>=-2*pi && setup.pa(i,2) < -7*pi/4 % 90-135
                setup.pasquare(i,:)=[...
                    Trm.Center.x+Trm.Dist, Trm.Center.y+...
                    Trm.Dist*sqrt(1/(cos(setup.pa(i,2))^2) - 1),...
                    setup.pa(i,3)];
            elseif setup.pa(i,2)>=-7*pi/4 && setup.pa(i,2) < -3*pi/2 % 135-180
                setup.pasquare(i,:)=[Trm.Center.x+...
                    Trm.Dist*sqrt(1/(sin(setup.pa(i,2))^2) - 1),...
                    Trm.Center.y+Trm.Dist,...
                    setup.pa(i,3)];
            elseif setup.pa(i,2)>= -3*pi/2 && setup.pa(i,2)< -5*pi/4 % 180-225
                setup.pasquare(i,:)=[...
                    Trm.Center.x-Trm.Dist*...
                    sqrt(1/(sin(setup.pa(i,2))^2)-1),Trm.Center.y+...
                    Trm.Dist,setup.pa(i,3)];
            elseif setup.pa(i,2)>=-5*pi/4 && setup.pa(i,2) < -pi  %225-270
                setup.pasquare(i,:)=[Trm.Center.x-Trm.Dist,...
                    Trm.Center.y+Trm.Dist*...
                    sqrt(1/(cos(setup.pa(i,2))^2)-1),setup.pa(i,3)];
            elseif setup.pa(i,2)>=-pi && setup.pa(i,2) < -3*pi/4  %270-315
                setup.pasquare(i,:)=[Trm.Center.x-Trm.Dist,...
                    Trm.Center.y-Trm.Dist*...
                    sqrt(1/(cos(setup.pa(i,2))^2)-1),setup.pa(i,3)];
            else    %315 - 360
                setup.pasquare(i,:)=[Trm.Center.x-Trm.Dist*...
                    sqrt(1/(cos(setup.pa(i,2)-pi/2)^2) - 1),...
                    Trm.Center.y-Trm.Dist,setup.pa(i,3)];
            end
        else
            % CIRCLE TRM
            setup.pa2(i,:)=[...
                Trm.Center.x + setup.pa(i,1)*cos(setup.pa(i,2)),...
                Trm.Center.y + setup.pa(i,1)*sin(setup.pa(i,2)),...
                setup.pa(i,3)];
        end

        %Individual X,Y,Z components of each Element
        if  strcmp(Trm.Type,'modified_square')
```

```
                x0=setup.pasquare(:,1);
                y0=setup.pasquare(:,2);
                z0=setup.pasquare(:,3);
            else
                x0=setup.pa2(:,1);
                y0=setup.pa2(:,2);
                z0=setup.pa2(:,3);
            end
        end

        % LINE ARRAY CONSTRUCTION
    elseif strcmp(Trm.Type,'line')

        if Trm.FixLineAngle
            % CONSTANT ANGLE LINE TRM

            % CHECK THAT ANGULAR APERTURE IS LESS THAN 180 DEGREES
            if Trm.AngAperture >= pi

                % LINE ARRAY WILL BE QUASI-INFINITE
                setup.lineAngle = pi - .01;
            else
                setup.lineAngle = Trm.AngAperture;
            end

            % DEFINE ARRAY LENGTH [m]
            setup.ArrayLength = 2 * Trm.Dist * tan(setup.lineAngle/2);

            % REDEFINE ELEMENT SPACING [WAVELENGTHS]
            Trm.LineArraySpacing = setup.ArrayLength / (Trm.Num - 1)...
                / lambda;

        else
            % CONSTANT INTERELEMENT SPACING

            % DEFINE ARRAY LENGTH [m]
            setup.ArrayLength = (Trm.Num - 1) * Trm.LineArraySpacing * lambda;

        end

        for i = 1:Trm.Num
            % DEFINE LINE ARRAY ALONG X DIRECTION
            setup.pa(i,:) = [...
                (Trm.Center.x - setup.ArrayLength/2) +...
                (i-1)*Trm.LineArraySpacing*lambda,Trm.Center.y - Trm.Dist,...
                Trm.Center.z];
        end

        x0 = setup.pa(:,1);
        y0 = setup.pa(:,2);
        z0 = setup.pa(:,3);
    else
        ME = MException('VerifyInput:Undefined','TRM Type is not recognized');
        throw(ME);
    end
    clear setup

    if ~fig.iterX

        % VECTORIZE TRM ELEMENTS FOR FASTER COMPUTATION
        [~,~,x0grid]=meshgrid(1:fieldvar.size.x,1:fieldvar.size.y,x0);
        [~,~,y0grid]=meshgrid(1:fieldvar.size.x,1:fieldvar.size.y,y0);
        [~,~,z0grid]=meshgrid(1:fieldvar.size.x,1:fieldvar.size.y,z0);
```

```
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % RUN THROUGH TEST

    % LOOP THROUGH SOURCE POSITIONS: X-POSITION
    for xx=1:length(Source.x)

        %LOOP THROUGH SOURCE POSITIONS: Y-POSITIONS
        for yy= 1:length(Source.y)

            % TIMER AND COUNTER
            tic
            progress = progress + 1;

            % CURRENT SOURCE POSITION
            xs = Source.x(xx) * lambda + Trm.Center.x;
            ys = Source.y(yy) * lambda + Trm.Center.y;

            % CHECK IF ITERATING THROUGH Z-POSITION
            if length(Source.z) > 1
                zs = Source.z(nn) * lambda + Trm.Center.z;
                z = zs;
            else
                zs = Source.z * lambda + Trm.Center.z;
            end

            % XTREME ITERATION MODE
            if fig.iterX

                xt = xs - FieldAverage.BoxSideM:fieldvar.h:xs + ...
                    FieldAverage.BoxSideM;
                yt = ys - FieldAverage.BoxSideM:fieldvar.h:ys + ...
                    FieldAverage.BoxSideM;
                zt = Trm.Center.z;

                % DEFINE VECTORIZED FIELD
                fieldvar.size.x = length(xt);
                fieldvar.size.y = length(yt);
                [x y ~] = meshgrid(xt, yt, 1:Trm.Num);

                % VECTORIZE TRM ELEMENTS FOR FASTER COMPUTATION
                [~,~,x0grid]=meshgrid(1:fieldvar.size.x,...
                    1:fieldvar.size.y,x0);
                [~,~,y0grid]=meshgrid(1:fieldvar.size.x,...
                    1:fieldvar.size.y,y0);
                [~,~,z0grid]=meshgrid(1:fieldvar.size.x,...
                    1:fieldvar.size.y,z0);

            end

            % AMPLITUDES AND PHASING FROM FORWARD PROPAGATION
            % (1 = DIRECT, 2 = REFLECTED)
            B1=A./sqrt((x0grid-xs).^2+(y0grid-ys).^2+(zs-z0grid).^2);
            B2=A./sqrt((x0grid-xs).^2+(y0grid-ys).^2+(zs+z0grid).^2);
            theta1=k*sqrt((x0grid-xs).^2+(y0grid-ys).^2+(zs-z0grid).^2);
            theta2=k*sqrt((x0grid-xs).^2+(y0grid-ys).^2+(zs+z0grid).^2);

            % BACKWARD PROPAGATION
            % (DIR/REF TAKE DIRECT PATH, DIR/REF TAKE REFLECTED PATH)
            ps1prime=B1./sqrt((x-x0grid).^2+(y-y0grid).^2+(z-z0grid).^2).*...
                exp(-1i*(w*t-k*sqrt((x-x0grid).^2+(y-y0grid).^2+...
                (z-z0grid).^2)+theta1));
```

```
            ps1prime=ps1prime+B2./sqrt((x-x0grid).^2+(y-y0grid).^2+...
                (z-z0grid).^2).*...
                exp(-1i*(w*t-k*sqrt((x-x0grid).^2+(y-y0grid).^2+...
                (z-z0grid).^2)+theta2));
            ps1prime=ps1prime+B1./sqrt((x-x0grid).^2+(y-y0grid).^2+...
                (z+z0grid).^2).*...
                exp(-1i*(w*t-k*sqrt((x-x0grid).^2+(y-y0grid).^2+...
                (z+z0grid).^2)+theta1));
            ps1prime=ps1prime+B2./sqrt((x-x0grid).^2+(y-y0grid).^2+...
                (z+z0grid).^2).*...
                exp(-1i*(w*t-k*sqrt((x-x0grid).^2+(y-y0grid).^2+...
                (z+z0grid).^2)+theta2));

            % COLLAPSE FIELD TOGETHER AND CLEAR TEMPORARY VARIABLES
            field=sum(ps1prime,3);
            clear ps1prime B1 B2 theta1 theta2;

            % REMOVE UNBOUNDED VALUES
            for i=1:length(yt)
                for ii=1:length(xt)
                    if isinf(field(i,ii))
                        field(i,ii)=0;
                    end
                end
            end
            clear i ii

            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            % QUALITY FACTOR MEASUREMENTS

            % LOCATE SOURCE POSITION ON FIELD
            xfp = find(xt >= xs,1,'first');
            xfn = find(xt < xs,1,'last');
            yfp = find(yt >= ys,1,'first');
            yfn = find(yt < ys,1,'last');
            if abs(xt(xfp)-xs) > abs(xt(xfn)-xs)
                cpx = xfn;
            else cpx=xfp;
            end
            if abs(yt(yfp)-ys)>abs(yt(yfn)-ys)
                cpy = yfn;
            else cpy=yfp;
            end
            clear xfp xfn yfp yfn

            % AMPLITUDE AT SOURCE POSITION
            centermax = abs(field(cpy,cpx));

            if FieldAverage.GetQ

                % DETERMINE FIELD TYPE TO CREATE
                if strcmp(FieldAverage.Shape,'square')
                    % SQUARE FIELD AVERAGE

                    if ~fig.iterX

                        %ERROR CHECKING
                        if abs(xs - FieldAverage.BoxSideM) > abs(xt(1))...
                                || abs(xs + FieldAverage.BoxSideM) > abs(xt(end))...
                                || abs(ys - FieldAverage.BoxSideM) > abs(yt(1)) ...
                                || abs(ys + FieldAverage.BoxSideM) > abs(yt(end))
                            ME = MException('VerifyInput:Limit',...
                                'Field is not large enough for averaging');
```

```
                    throw(ME);
                end
            end

            % FIND FIELD SIDES [INDEXED POINTS]
            fl = find(xt >= xs - FieldAverage.BoxSideM,1,'first');
            fr = find(xt <= xs + FieldAverage.BoxSideM,1,'last');
            fb = find(yt >= ys - FieldAverage.BoxSideM,1,'first');
            ft = find(yt <= ys + FieldAverage.BoxSideM,1,'last');

            if FieldAverage.MinusSource

                % FIND SOURCE SIDES [INDEXED POINTS]
                sbox = .25*lambda;
                sl=cpx - round(sbox * fieldvar.conv);
                sr=cpx + round(sbox * fieldvar.conv);
                sb=cpy - round(sbox * fieldvar.conv);
                st=cpy + round(sbox * fieldvar.conv);

                % ZERO POINTS ABOUT SOURCE
                Fieldtemp = field;
                Fieldtemp(sb:st,sl:sr) = 0;

                % AVERAGE FIELD WITHOUT SOURCE
                FieldAverage.Avg = ...
                    mean(mean(abs((Fieldtemp(fb:ft,fl:fr)))));
                clear Fieldtemp sl sr sb st

            else

                % AVERAGE FIELD WITH SOURCE
                FieldAverage.Avg = ...
                    mean(mean(abs((field(fb:ft,fl:fr)))));

            end

            clear fl fr fb ft

        elseif strcmp(FieldAverage.Shape,'circle')
            % CIRCLE FIELD AVERAGE

            if ~fig.iterX

                %ERROR CHECKING
                if abs(xs - FieldAverage.BoxSideM) > abs(xt(1))...
                        || abs(xs + FieldAverage.BoxSideM) > abs(xt(end))...
                        || abs(ys - FieldAverage.BoxSideM) > abs(yt(1)) ...
                        || abs(ys + FieldAverage.BoxSideM) > abs(yt(end))
                    ME = MException('VerifyInput:Limit',...
                        'Field is not large enough for averaging');
                    throw(ME);
                end
            end

            % INITIALIZE TEMPORARY ITERATORS
            count = 0;
            fieldavgtot = 0;

            % LOOP THROUGH EACH POINT ON THE FIELD
            for i = 1:length(field(:,1))
                for ii = 1:length(field(1,:))

                    % DISTANCE FROM SOURCE TO FIELD POINT
```

```
                        %     [INDEXED POINTS]
                        rDist = sqrt((cpx-ii)^2+(cpy-i)^2);

                        % CIRCLE RADIUS [INDEXED POINTS]
                        circleR = FieldAverage.BoxSideM / lambda *...
                            fieldvar.conv;

                        % SOURCE IS WITHIN CIRCLE AND NOT WITHIN
                        %     1/4 WAVELENGTH OF SOURCE
                        if rDist <= circleR && rDist > 0.25*fieldvar.conv
                            fieldavgtot = fieldavgtot + abs(field(i,ii));
                            count = count + 1;
                        end
                    end
                end
                if count == 0
                    FieldAverage.Avg = 0;
                else
                    FieldAverage.Avg = real(fieldavgtot/count);
                end
                clear i ii rDist count fieldavgtot circleR;

            else
                ME = MException('VerifyInput:Undefined',...
                    'Field Average Shape is not recognized');
                throw(ME)
            end

            % GIVE LOCALIZATION QUALITY RATIO
            loc_quality(nn,xx,yy) = centermax / FieldAverage.Avg;

        end

        % SOURCE TO PEAK RATIO
        if FieldAverage.GetPeak

            % DEFINE TEMPORARY FIELD
            Fieldtemp = field;

            % FIND FIELD SIDES [INDEXED POINTS]
            fl = find(xt >= xs - FieldAverage.BoxSideM,1,'first');
            fr = find(xt <= xs + FieldAverage.BoxSideM,1,'last');
            fb = find(yt >= ys - FieldAverage.BoxSideM,1,'first');
            ft = find(yt <= ys + FieldAverage.BoxSideM,1,'last');

            % DISCLUDE AMPLITUDES FROM SOURCE POSITION
            % FIND SOURCE SIDES [INDEXED POINTS]
            sbox = .25; % [WAVELENGTHS]
            sl=cpx - round(sbox * fieldvar.conv);
            sr=cpx + round(sbox * fieldvar.conv);
            sb=cpy - round(sbox * fieldvar.conv);
            st=cpy + round(sbox * fieldvar.conv);

            % ZERO POINTS ABOUT SOURCE
            Fieldtemp(sb:st,sl:sr) = 0;

            % DEFINE FIELD WHERE PEAK WILL BE LOCATED
            fieldPeak = Fieldtemp(fb:ft,fl:fr);

            % LOCATE PEAK AMPLITUDE ( MAX( |AVERAGE BOX| ) )
            [px py] = find(real(fieldPeak) == ...
                max(max(real(fieldPeak))),1,'first');

```

```
            % GIVE SOURCE TO NEXT PEAK RATIO
            peak_quality (nn,xx,yy) = centermax / real(fieldPeak(px,py));

            % CLEAR TEMPORARY VARIABLES
            clear Fieldtemp fieldPeak sl sr sb st fl fr fb ft sbox ...
                px py

        end

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        % PLOTTING

        % WHETHER TO PLOT DATA
        if fig.draw

            % PLOT IS VISIBLE OR NOT
            if fig.visual
                figure
            else
                figure('Visible','off')
            end

            % PLOT USING PCOLOR
            pcolor(xt,yt,real(field));
            hold on

            % FIGURE PROPERTIES
            shading interp

            % ADD CONTOUR LINES
%             [C,hc] = contour(xt,yt,abs(field),10,'k',...
%                 'LevelList',centermax/2);

            % ADD POINTS FOR THE TRM ELEMENTS
            if ~(Trm.AngAperture >= pi && strcmp(Trm.Type,'line'))
                ppoints=(max(max(real(field)))+20).*ones(length(x0));
                plot3(x0,y0,ppoints,'ko','markersize',6,'markerfacecolor','k')
            end

            % ADD A POINT AT THE SOURCE
            % plot3(xs,ys,max(max(max(real(field))))+20,'ko',...
            % 'markersize',6,'markerfacecolor','k');

            % ADD A POINT AT THE TRM CENTER
%             plot3(Trm.Center.x,Trm.Center.y,...
%                 max(max(max(real(field))))+20,...
%                 'bx','markersize',6,'markerfacecolor','k')

            if strcmp(FieldAverage.Shape,'square')

                % ADD A SQUARE OVER WHICH THE FIELD IS AVERAGED
                xp=[xs-FieldAverage.BoxSideM,xs-FieldAverage.BoxSideM,...
                    xs+FieldAverage.BoxSideM,xs+FieldAverage.BoxSideM];
                yp=[ys-FieldAverage.BoxSideM,ys+FieldAverage.BoxSideM,...
                    ys+FieldAverage.BoxSideM,ys-FieldAverage.BoxSideM];
                zp=[max(max(real(field))), max(max(real(field))),...
                    max(max(real(field))) max(max(real(field)))];
                line(xp(1:2),yp(1:2),zp(1:2),'LineWidth',2,'Color','k');
                line(xp(2:3),yp(2:3),zp(2:3),'LineWidth',2,'Color','k');
                line(xp(3:4),yp(3:4),zp(3:4),'LineWidth',2,'Color','k');
                line([xp(4),xp(1)],[yp(4) yp(1)],[zp(4) zp(1)],...
                    'LineWidth',2,'Color','k');

```

```
                elseif strcmp(FieldAverage.Shape,'circle')

                    % ADD A CIRCLE OVER WHICH THE FIELD IS AVERAGED
                    J=circle([Trm.Center.x,Trm.Center.y,...
                        max(max(real(field)))+200],...
                        FieldAverage.BoxSideM,1000,'k-');
                    set(J,'LineWidth',2);

                end

                % OTHER FIGURE PROPERTIES
                axis image;
                colorbar
                xlabel('X Position (m)','Fontsize',18);
                ylabel('Y Position (m)','Fontsize',18);
                if FieldAverage.GetQ || FieldAverage.GetPeak
                    if FieldAverage.GetPeak
                        title([...
                            %'Source / Field = ',num2str(loc_quality(nn,xx,yy)),...
                            'Source / Peak = ',num2str(peak_quality(nn,xx,yy))]...
                            ,'Fontsize',16);
                    else
                        title(['Source vs Field = ',...
                            num2str(loc_quality(nn,xx,yy))],'Fontsize',16);
                    end
                end
                caxis([-centermax centermax])
                set(gca,'fontsize',16);
                hold off


                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                % SAVE DATA AND NOTIFY OF COMPLETION

                % SAVE FIGURE
                if saveIt.fig
                    if nn == 1
                        print('-dpng',[saveIt.file,'/',saveIt.date,'/',...
                            saveIt.ID,'/',saveIt.ID,'_Fig_',...
                            num2str(xx),'_',num2str(yy)]);
                    end
                end

                % ITERATION MODE
                if fig.iter
                    close all;
                end

            end

            % DISPLAY PROGRESS
            ttime(nn,progress) = toc;
            disp(['Source Position:(',num2str(xs),',',num2str(ys),',',...
                num2str(zs),') Finished in ',num2str(toc),...
                ' sec - Progress: ',...
                num2str(progress), ' / ',num2str(length(Source.x)*...
                length(Source.y)*length(Source.z))]);
        end
    end

    % HOLD ON TO TRM VALUES IF IT CHANGES
    if length(iterPar) > 1
```

```
        if length(Trm.iterPar) > 3
            if strcmp(Trm.iterPar(1:4),'Trm.')
                eval(['x0tot.N',num2str(nn),' = x0;']);
                eval(['y0tot.N',num2str(nn),' = y0;']);
                eval(['z0tot.N',num2str(nn),' = z0;']);
            end
        end
    end

end

% SAVE VARIABLES
if saveIt.data && (FieldAverage.GetQ || FieldAverage.GetPeak)
    if FieldAverage.GetQ
        loc_quality = squeeze(loc_quality);
        save([saveIt.file,'/',saveIt.date,'/',saveIt.ID,'/',saveIt.ID],...
        'loc_quality');
    end
    if FieldAverage.GetPeak
        peak_quality = squeeze(peak_quality);
        save([saveIt.file,'/',saveIt.date,'/',saveIt.ID,'/',saveIt.ID,'P'],...
        'peak_quality');
    end
end

% CREATE LOG FILE
if saveIt.logfile

    % RETURN PARAMETER ITERATOR TO INITIAL STATE
    if length(iterPar) > 1
        eval([Trm.iterPar,' = ','iterAll;']);
    end

    logme.date = ['Processing Date: ',datestr(now)];
    logme.mfile = mfilename;

    % SAVE TRM BASED ON ITERATION PARAMETER
    if length(Trm.iterPar) > 3 && strcmp(Trm.iterPar(1:4),'Trm.')
        logme.Parameters.x0 = x0tot;
        logme.Parameters.y0 = y0tot;
        logme.Parameters.z0 = z0tot;
    else
        logme.Parameters.x0 = x0;
        logme.Parameters.y0 = y0;
        logme.Parameters.z0 = z0;
    end

    logme.Parameters.f = f;
    logme.Parameters.lambda = lambda;
    logme.Source = Source;
    logme.Trm = Trm;
    if FieldAverage.GetQ
        logme.FieldAverage = FieldAverage;
    end
    save([saveIt.file,'/',saveIt.date,'/',saveIt.ID,'/',saveIt.ID,...
        '_logfile'],'logme');

    % TURN DIARY OFF
    diary off
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CLEAN UP AND FINISH
```

```
tottime = sum(sum(ttime));
fprintf('\n\t Total Time Elapsed: %f seconds\n',tottime);

clearvars
fprintf('\t\t-\t-\t-\t-\n\t\tProgram Complete\n\t\t-\t-\t-\t-\n');

% [EOF]
```

Bibliography

[1] M. Fink, "Time-reversal of ultrasonic fields. 1. basic principles", Ieee Trans. Ultra. Ferro. Freq. Ctrl. **39**, 555–566 (1992).

[2] M. Fink, "Time reversed acoustics", Phys. Today **50**, 34–40 (1997).

[3] A. Parvulescu and C. Clay, "Reproducibility of signal transmissions in the ocean", Radio Elec. Eng. **29**, 223 –228 (1965).

[4] B. E. Anderson, M. Griffa, C. Larmat, T. J. Ulrich, and P. A. Johnson, "Time reversal", Acoust. Today **4**, 5–16 (2008).

[5] J.-L. Thomas, F. Wu, and M. Fink, "Time reversal focusing applied to lithotripsy", Ultra. Imag. **18**, 106–121 (1996).

[6] C. S. Larmat, R. A. Guyer, and P. A. Johnson, "Time-reversal methods in geophysics", Phys. Today **63**, 31–35 (2010).

[7] G. S. O'Brien, I. Lokmer, L. De Barros, C. J. Bean, G. Saccorotti, J. P. Metaxian, and D. Patane, "Time reverse location of seismic long-period events recorded on mt etna", Geophys. J. Int. **184**, 452–462 (2011).

[8] T. J. Ulrich, P. A. Johnson, and R. A. Guyer, "Interaction dynamics of elastic waves with a complex nonlinear scatterer through the use of a time reversal mirror", Phys. Rev. Lett. **98**, 104301 (2007).

[9] T. J. Ulrich, A. M. Sutin, T. Claytor, P. Papin, P. Y. Le Bas, and J. A. TenCate, "The time reversed elastic nonlinearity diagnostic applied to evaluation of diffusion bonds", Appl. Phys. Lett. **93**, 151914 (2008).

[10] C. Prada, S. Manneville, D. Spoliansky, and M. Fink, "Decomposition of the time reversal operator: Detection and selective focusing on two scatterers", J. Acoust. Soc. Am. **99**, 2067–2076 (1996).

[11] B. E. Anderson, M. Griffa, P.-Y. L. Bas, T. J. Ulrich, and P. A. Johnson, "Experimental implementation of reverse time migration for nondestructive evaluation applications", J. Acoust. Soc. Am. **129**, EL8–EL14 (2011).

[12] A. Sutin, B. Libbey, L. Fillinger, and A. Sarvazyan, “Wideband nonlinear time reversal seismo-acoustic method for landmine detection”, J. Acoust. Soc. Am. **125**, 1906–1910 (2009).

[13] A. Song and M. Badiey, “Time reversal acoustic communication for multiband transmission”, J. Acoust. Soc. Am. **131**, EL283–EL288 (2012).

[14] B. E. Anderson, R. A. Guyer, T. J. Ulrich, and P. A. Johnson, “Time reversal of continuous-wave, steady-state signals in elastic media”, AIP **94**, 111908 (2009).

[15] M. Fink, D. Cassereau, A. Derode, C. Prada, P. Roux, and M. Tanter, “Time-reversed acoustics”, Rep. on Prog. Phys. **63**, 1933–1995 (2000).

[16] J. Kraus and R. Marhefka, *Antennas for all applications*, 3rd edition (McGraw-Hill, Woodbury, NY) (2002).

[17] M. Fink, “Time-reversal mirrors”, J. Phys. D: Appl. Phys. **26**, 1333 (1993).

[18] C. S. Larmat, R. A. Guyer, and P. A. Johnson, “Tremor source location using time reversal: Selecting the appropriate imaging field”, Geophys. Res. Lett. **36**, L22304 (2009).

[19] B. E. Anderson, M. Griffa, T. J. Ulrich, and P. A. Johnson, “Time reversal reconstruction of finite sized sources in elastic media”, J. Acoust. Soc. Am. **130**, EL219–EL225 (2011).

[20] M. Tanter, J. L. Thomas, F. Coulouvrat, and M. Fink, “Breaking of time reversal invariance in nonlinear acoustics”, Phys. Rev. E **64**, 016602 (2001).

[21] K. B. Cunningham, M. F. Hamilton, A. P. Brysev, and L. M. Krutyansky, “Time-reversed sound beams of finite amplitude”, J. Acoust. Soc. Am. **109**, 2668–2674 (2001).

[22] K. L. Gee, A. A. Atchley, L. E. Falco, M. R. Shepherd, L. S. Ukeiley, B. J. Jansen, and J. M. Seiner, “Bicoherence analysis of model-scale jet noise”, J. Acoust. Soc. Am. **128**, EL211–EL216 (2010).

[23] D. of Veterans Affairs, “Veterans benefits administration annual benefits report fiscal year 2010”, 5.

[24] T. R. Board, “Effects of aircraft noise: Research update on selected topics”, **2008**.

[25] A. T. Wall, K. L. Gee, M. D. Gardner, T. B. Neilsen, and M. M. James, “Near-field acoustical holography applied to high-performance jet aircraft noise”, Proc. Meet. Acoust. **9**, 040009 (2011).

[26] S. H. Swift and K. L. Gee, “Examining the use of a time-varying loudness algorithm for quantifying characteristics of nonlinearly propagated noise (l)”, J. Acoust. Soc. Am. **129**, 2753–2756 (2011).

[27] J. Morgan, T. B. Neilsen, K. L. Gee, A. T. Wall, and M. M. James, “Simple-source model of high-power jet aircraft noise”, submitted to Noise Control Eng. J. (2011).

[28] S. F. Wu and N. Zhu, “Locating arbitrarily time-dependent sound sources in three dimensional space in real time”, J. Acoust. Soc. Am. **128**, 728–739 (2010).

[29] Q. Zhang, H. Abeida, M. Xue, W. Rowe, and J. Li, “Fast implementation of sparse iterative covariance-based estimation for source localization”, J. Acoust. Soc. Am. **131**, 1249–1259 (2012).

[30] K. L. Gee, V. W. Sparrow, M. M. James, J. M. Downing, C. M. Hobbs, T. B. Gabrielson, and A. A. Atchley, “The role of nonlinear effects in the propagation of noise from high-power jet aircraft”, J. Acoust. Soc. Am. **123**, 4082–4093 (2008).