A SIMPLE ELECTRONICALLY-PHASED ACOUSTIC ARRAY

Martin L. Smith
Blindgoat Geophysics
2022 Quimby Mountain Road
Sharon, Vermont 05065

Michael R. Roddewig
Department of Electrical Engineering, Montana State University
Bozeman, Montana 59717

Kurt M. Strovink
Department of Physics, Colorado School of Mines
Golden, Colorado 80401

John A. Scales
Department of Physics, Colorado School of Mines
Golden, Colorado 80401

Introduction to phased arrays

Phased source arrays are physically-distributed arrays of radiators (source transducers) driven by time-delayed copies of a common signal. The time delays are chosen so that the net antenna pattern of the array is focused on either a particular target or lies in a particular direction. A simple example is an array of broadcast antennae with an individual element's time delay determined by the length of its feed cable. By switching among sets of feed cables the composite antenna may be made to have major lobes in different directions, in terms of both elevation and azimuth, as needed to meet the varying requirements of daytime and nighttime operation. These arrays have either fixed delays or a few, selectable sets of fixed delays.

"A field-programmable gate array (FPGA) is a complex, miraculously versatile piece of field-configurable hardware comprising hundreds of thousands of logical units of various types…"

Phased receiver arrays are based on exactly the same principles as their source counterparts—a correspondence that is spelled out in a little more detail in the implementation section of this article. In all other respects, the term phased array will mean phased source array.

We are interested here in electronically-steerable phased arrays, that is, phased arrays with electronically-controlled time delays. These systems are capable of rapid shifts over many antenna patterns which make it practical to quickly scan the major lobe of the outgoing signal over a range of azimuths and elevations. One application of this technology is military target-tracking radar systems, which may have to (nearly) simultaneously track many potential threats much more rapidly than would be feasible with a physically steered antenna. Perhaps the most active area of the application of phased-array technology is in bio-medical devices which image the body with ultrasound. This field is driven by the needs of medical diagnosis and treatment; it is fortunate that the human body represents a relatively homogeneous environment of acoustic speeds which rewards the development of sophisticated imaging methods. For discussion of the basic technologies and a number of examples, see, for example, Cobbold (2007) and Azhari (2010).

Phase-shift *versus* time-shift

There are two basic modes of phased-array implementation. In the first, the driving signal is sinusoidal and the signal routed to each element is given a prescribed phase shift. In the second more general, case the driving signal is an arbitrary function of time and the signal routed to each element is time-delayed by a prescribed amount. Note that the time-shift case includes the phase-shift case as a particular instance. We will focus on time-shifted systems which accept an arbitrary function of time as input.

Planar wavefield versus time-on-target

There are also two philosophically distinct ways of computing the pattern of element time delays. In the planar wavefield method, time delays are chosen so that the radiated field approximates a plane wave moving in a prescribed direction. Figure 1 schematizes the quasi-planar wavefront formed by the time-shifted circular wavefronts from an array of sources. Note that the approximation to a plane wave degrades to the right of the figure where the desired plane wave is closer to the array. At distances that are large compared to the array dimension the array appears as a point source and the wavefront is circular (in two dimensions, spherical in three). Of course spreading waveforms at large distances look locally like plane wavefronts.

In the time-on-target method, delays are chosen so that signals from all of the individual elements arrive at a prescribed location in space at the same time (the name derives from a practice in artillery management). Figure 2 shows the circular wavefronts from the five-element array arriving

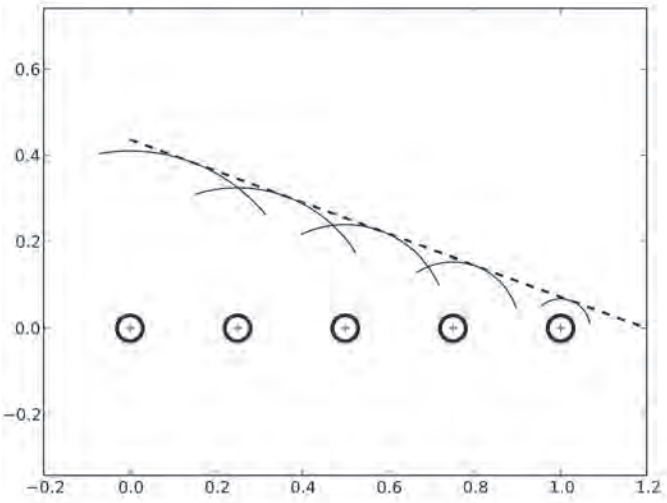

Fig. 1. Schematic of phased wavefronts approximating a plane wave.

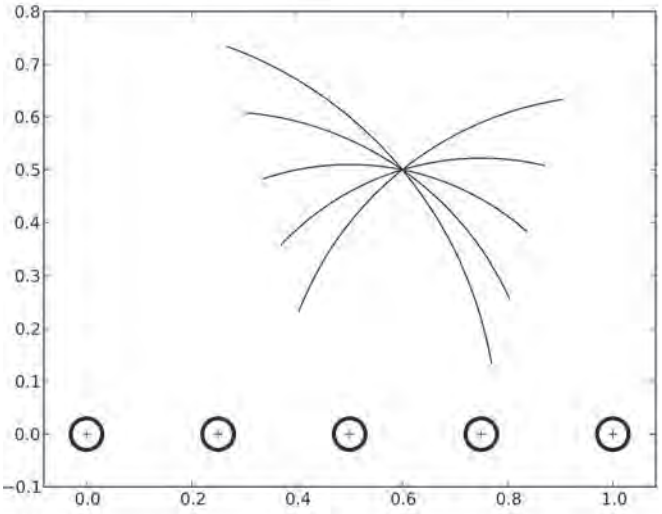

Fig. 2. Simultaneous arrival of circular wavefronts at the target site.

simultaneously at the target site. With a little thought, it is easy to identify the circular wavefronts with their respective sources.

As the time-on-target site moves away from the array, the two types of time delays become increasingly indistinguishable. In the implementation described later we use a time-on-target algorithm; the same code can be used to approximate a planar wavefield simply by making the distance to the target large compared to the array dimensions.

Antenna patterns and design criteria

The antenna pattern of a phased array depends upon the radiation pattern of the individual transducer elements, the distribution of transducers in the array, the size of the array, and the frequency band of operation. There are extensive engineering criteria for selecting both the size of the individual transducer elements as well as the number and distribution of transducers to achieve various desired types of behavior. There are numerous useful criteria: some applications exploit tight collimation of the main lobe of the array's response; some are dependent on minimizing the largest sidelobes while allowing greater spreading of the main lobe; some require other considerations.

The system that we built is very modestly engineered. It is a functional prototype constructed from off-the-shelf components with an eye to versatility and future experimentation.

Simulating the array

We decided to invest the time to write a simple modeling program for the phased array. The model is useful because it lets us study array performance in free space, a condition that is very hard to achieve experimentally, and because it is often faster to run a suite of modeling runs than it is to make observations on several experimental configurations. In the end, of course, it is the behavior of the physical system that matters.

A diagram of the transducer geometry we used is shown in Fig. 3. Each transducer is a loudspeaker (tweeter) 0.03 m in radius. The transducers are mounted in a piece of rigid Delrin 0.42 m by 0.22 m. The array is symmetric about the x and y axes but is not axisymmetric about z.

The system was modeled as an array of baffled circular pistons—each tweeter is a piston source with prescribed sinusoidal normal velocity mounted in an infinite rigid sheet. This model ignores interaction between transducers and also ignores acoustic coupling between the front and rear sides of the array: we assume the plastic mounting to be perfectly rigid and infinite in extent.

The field due to a single baffled circular piston is exactly described by a Green's function integral over the face of the piston (Dowling, 1998) We implemented this integral numerically so that the results would be correct in both the near-field and the far-field. The net array response was computed by summing the signal from each transducer at each desired point in space. We specify a time-on-target location and the signal from any particular transducer is advanced or delayed in time by an appropriate amount.

Our calculations occur in a coordinate system the origin of which is the geometric center of the array. We take the orthogonal coordinate x as horizontal, y as vertical, and z increasing away from the array. To specify a calculation we

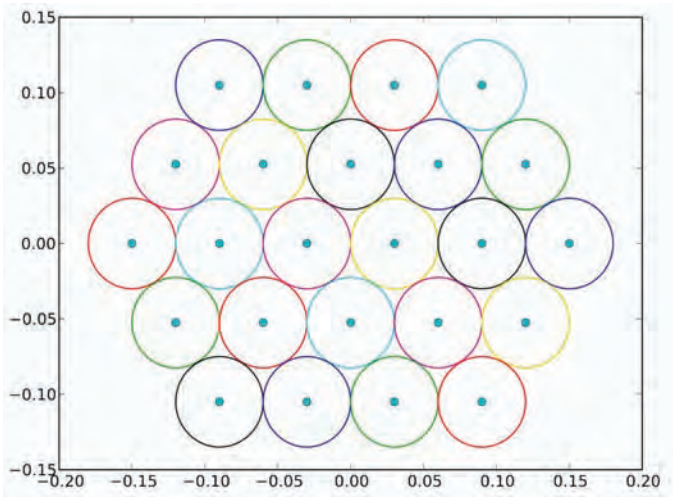

Fig. 3. Geometry of our transducer array. Axes are in meters.

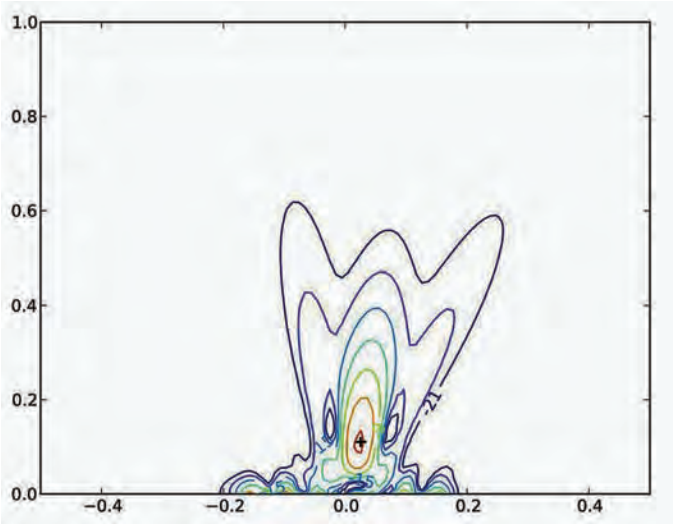

Fig. 4. Beam pattern at range of 0.12 m. Units in all the plots are meters. The target location is indicated with a "+" symbol. In all examples the data plane is perpendicular to and in the mid-line of the array.

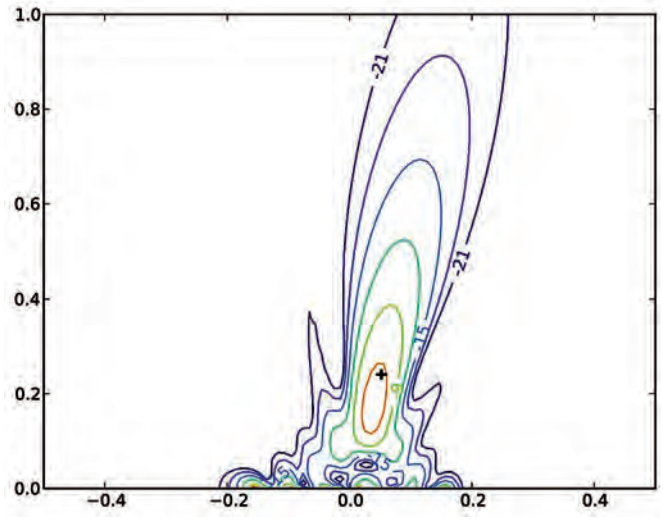

Fig. 5. 5 kHz beam pattern at range of 0.25 m.

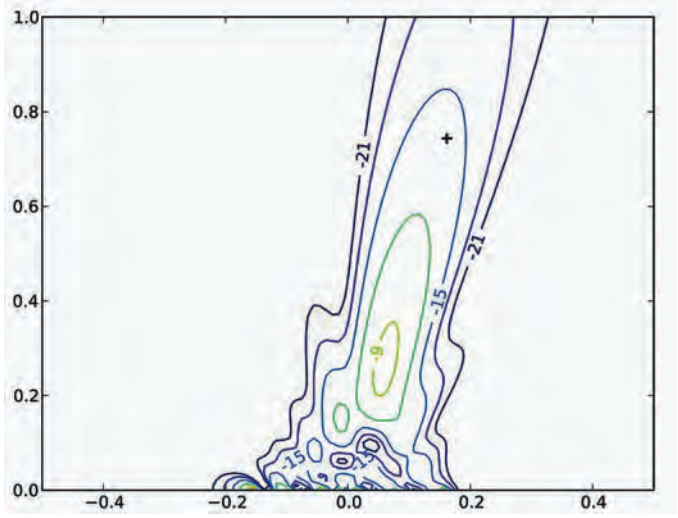

Fig. 6. 5 kHz beam pattern at range of 0.75 m.

must provide

- the transducer size,
- the array geometry,
- the frequency radiated by the pistons,
- a target position somewhere in space in front of the array, and,
- the sampling volume over which the response is to be computed.

In our computation, the sampling volume is always a 2D slice aligned with any two of the coordinate axes.

The simulation code was written in the Python language. It makes extensive use of the impressive numerical extension, Numpy, and the large collection of numerical utilities in Scipy. In particular, Numpy made it straightforward to vectorize the innermost computational loop at a substantial time savings. In the code's final form it took about five minutes to compute the response patterns shown below, each of which represents a 100 by 100 grid of values. This code along with all other elements of the system's design is available on our wiki:

http://mesoscopic.mines.edu/mediawiki/index.php/Acoustic_Phased_Array

Figures 4–6 show three simulations for a frequency of 5 kHz in which the target was moved from (x, y, z) of (0.05, 0.0, 0.25) to (0.15, 0, 0.75) meters along a linear path. Each plot shows the **xz** plane and uses logarithmic contours that are 3 dB apart, like all the contour plots that follow.

Notice that as the target moves away from the origin, the antenna pattern smoothly changes from a group of crossing beams to a single, nearly collimated beam. The case of crossing beams is an example of a near-field effect, which will be discussed further. The relatively minor change between the last two figures suggests that at distances of a half meter or more there is little difference between time-on-target scheduling, which we use here, and planar wavefield scheduling.

Figure 7 shows the substantial effect of frequency on the array's antenna pattern. In both plots we see an **xy** cross-section of the beam at the target range. The left subplot depicts the pattern for 5 kHz; the right subplot, for 10 kHz. The main lobe of the 10 kHz pattern is tighter than its counterpart on the left, but there are more spurious sidelobes at the higher frequency. In general as the frequency content of the time function increases, the array must be broken into more and smaller elements to minimize side lobes,

Implementation of an electronically-phased acoustic array

The one-sentence summary of a phased-array is simple enough—accept a common input signal, create a set of copies each time-shifted by an appropriate amount, and send the results to an array of transducers. The devil is in the details. We need some way to tell the system the target location, compute the necessary delays, distribute those to the time-shift elements, and make sure the shifted signals go to the appropriate transducer elements. This section is about the details and how we dealt with them. Our wiki provides complete

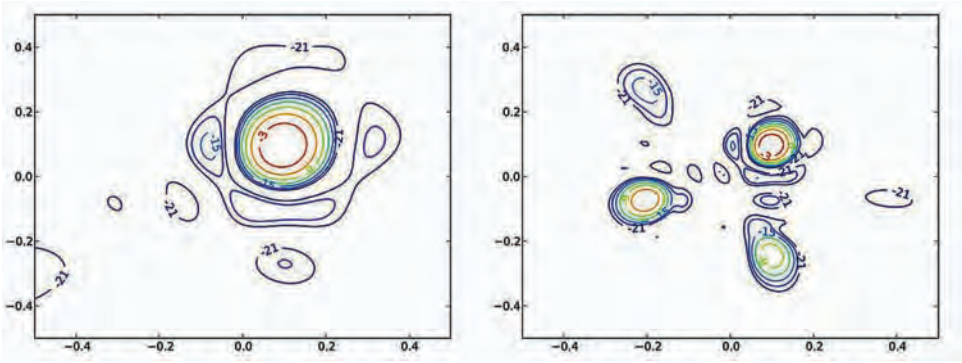

Fig. 7. Beam cross-section at the target for 5 kHz (left) and 10 kHz.

details on the system's electronics and software. It provides everything needed to reconstruct the system (except the hardware).

System components and interconnection

At the highest level, our implementation of a phased array consists of the major functional blocks shown in Fig. 8. Given a target location, time-shifting specifications are computed in a program—a Python script—running on a PC. This script has detailed information about the locations of the array's 24 transducers and their interconnection with the time-shifting electronics. The time-shifting electronics, discussed in more detail below, accepts and digitizes analog audio and sends the samples to each of 24 memory buffers. The electronics associated with each of these buffers time-shifts the buffer's contents and writes the shifted samples to a a digital-to-analog converter (DAC). Each of the 24 DACs is routed to a transducer (a tweeter in our case) by way of a 60W power amplifier; i.e., one amplifier for each channel.

Most of the design effort went into the large central box in Figure 8. The heart of the system is a set of 6 field-programmable gate arrays (FPGAs), each of which is responsible for managing 4 channels of time-shifted signal. Figure 9 shows an expanded view of the components of the central box and it's connected computer. The controller script on the computer actually communicates with a microcontroller, an Arduino, which in turn communicates with the phase-shifting FPGAs. The Arduino uses a Serial Peripheral Interface (SPI) bus as well as a dedicated control line for each FPGA to pass binary channel and shift information to the appropriate FPGA. It functions as a translator between the high-level ASCII commands from the controller script and the low-level binary required by the FPGAs. The SPI bus is buffered by a resistive network to enable the Arduino to simultaneously drive the lines to all six FPGAs.

Some design details

On FPGAs

The heart of the system is a bank of six field-programmable gate arrays (FGPAs) that do the work of signal-shifting. These devices are increasingly common in electronic systems of any appreciable complexity. Since they are not yet universally familiar to scientists, included is a short discussion of their nature and use.

An FPGA is a complex, miraculously versatile piece of field-configurable hardware comprising hundreds of thousands of logical units of various types, including simple logic devices, blocks of memory, adders, etc. The unit also includes

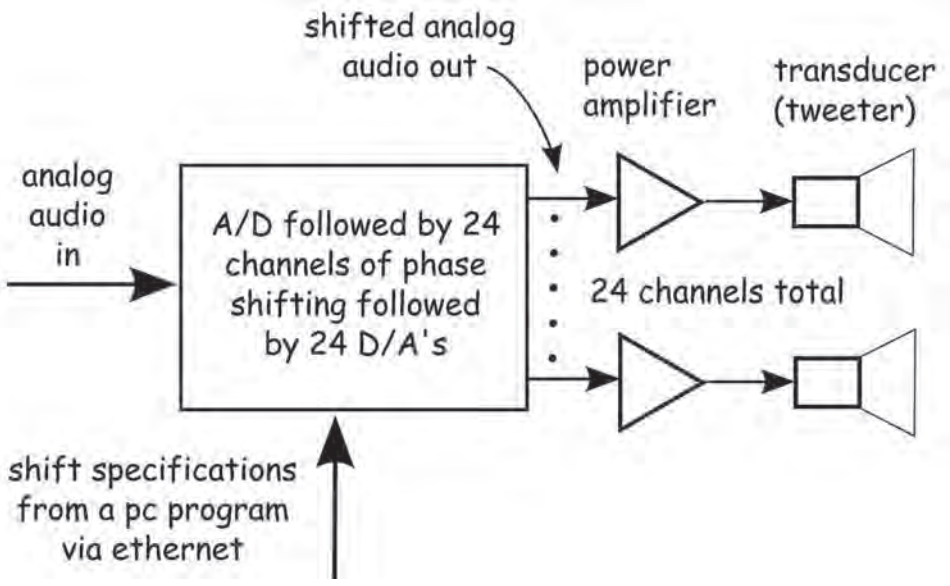

Fig. 8. High-level overview of the phase-shift array system.

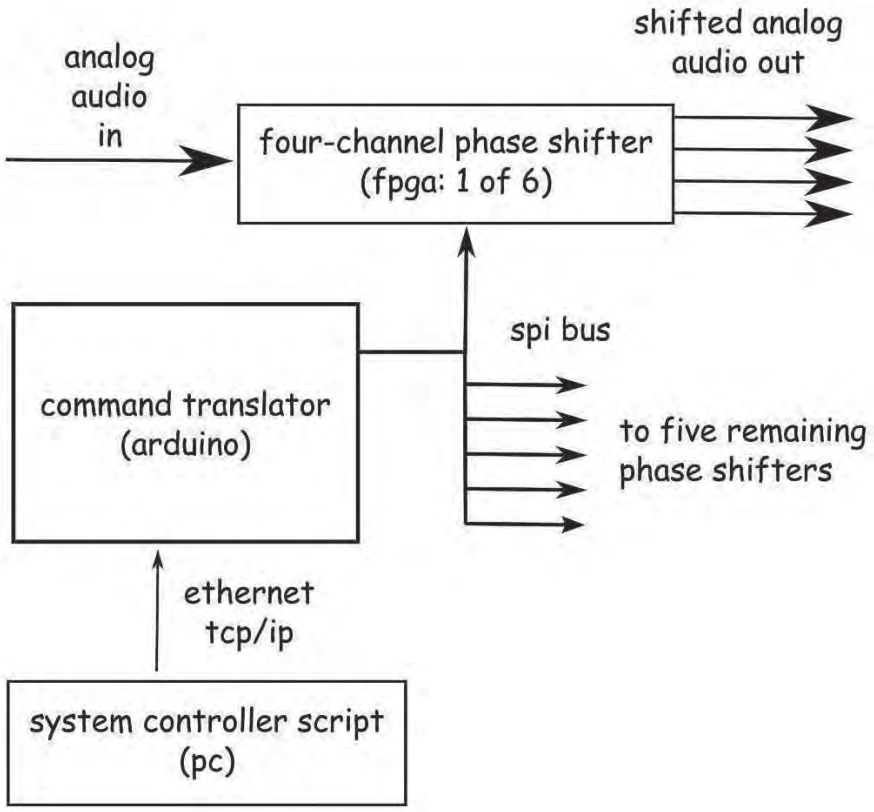

Fig. 9. Major internal components of the phase-shift electronics.

a programmable interconnection mesh which is used to configure the device. The designer turns this uncommitted functionality into a particular device by specifying the system's behavior in a high-level language; the two most popular hardware specification languages are Very High Speed Integrated Circuits Hardware Description Language (VHDL) and Verilog. The high-level specification is fed to a vendor's proprietary compiler which generates a detailed interconnection map. The map is applied by the FPGA at power-up and produces a device with the desired behavior.

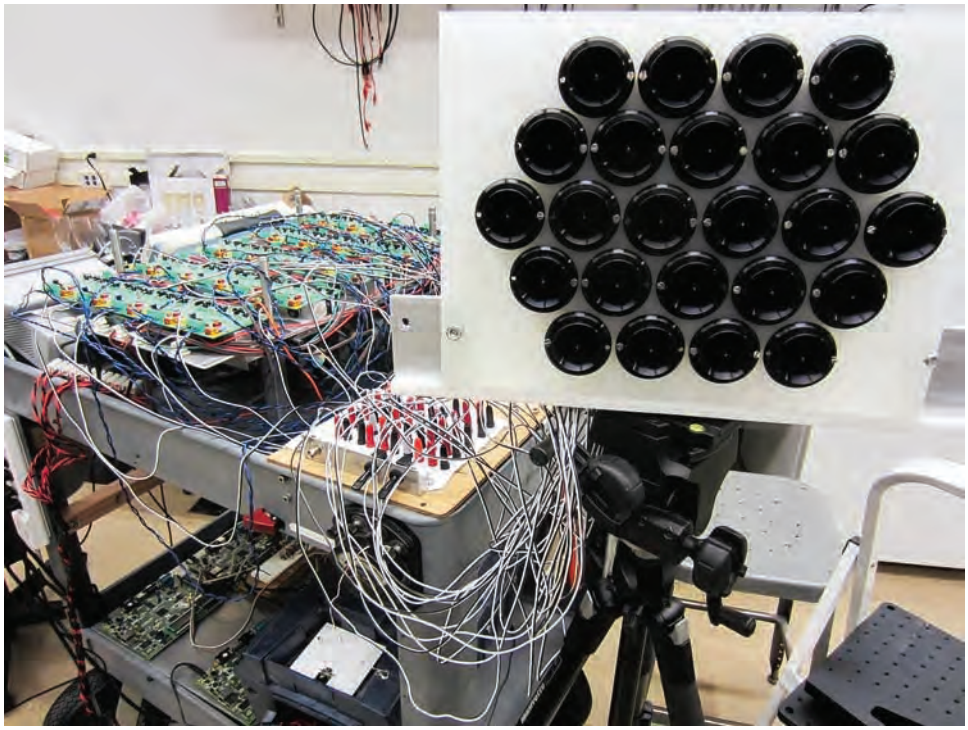

Fig. 10. Twenty-four commercial tweeters mounted in a Delrin sheet and fixed to a tripod. To the left is a cart carrying the amplifiers (top, one for each channel) and the FPG's (below). We used a patch-panel to connect the amplifiers and tweeters to make it easy to swap.

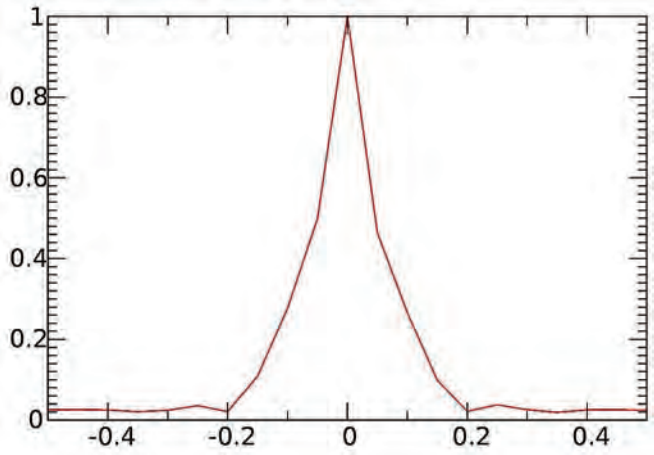

Fig. 11. A one-dimensional slice through the data 0.2 m in front of the array at y=0. The FWHM of this curve is about 1 wavelength. The data were normalized to 1 for this plot.

The internal interconnection of an FPGA that produces the desired ultimate functionality is not unique. Vendors' compilers are complex and highly-optimized applications which are fortunately available at little or no cost, especially to educational institutions.

An important point to bear in mind when using VHDL or Verilog is that although the languages look syntactically like conventional, procedural software languages, they are semantically vastly different. This difference is easy to see—a procedural programming language specifies a series of computations to be undertaken by a computer's central processing unit (CPU). An FPGA design language specifies the very wiring that interconnects the components (memory, adders, etc) of a device. VHDL can be used to specify the wiring of a CPU; Fortran can be used to specify a computation to execute on that CPU.

There are numerous excellent texts on VHDL. We particularly like Chu (2008) because the author emphasises the way simple VHDL statements are mapped onto particular circuits. We also recommend Pedroni (2004).

Sampling and buffering

The FPGAs digitize the incoming analog audio stream with 12-bit resolution at 347.2 ksps (thousand samples per second). Each buffer is 1024 samples long, allowing for a maximum shift of 2.96 ms (milliseconds). That shift corresponds to a spatial delay of about 1 meter. Thus with current design parameters the array has a Nyquist frequency of 174 kHz and can in principle accommodate spatial delay differences of about 1 meter across the array; in practice, radiation pattern effects come into play at much smaller delays.

Array dimensions and radiation properties

The array consists of 24 transducers (tweeters) arranged as shown in Fig. 3. The array is 0.3 m in its long dimension and 0.2 m in its short dimension. (These are the center-to-center distances between the edge transducers.) A wavelength of 0.34 m corresponds to a frequency of about 1 kHz. The array dimensions relative to the frequency are crucial in determining the radiation pattern of the array. Clearly, as the observer moves off to infinite range, any spatially finite source will look like a point source. In the other limit, as we approach the array we see the interference effects of individual radiating elements. These are, respectively, the near-field and far-field regions.

The principle of reciprocity, which is similar to time-reversal invariance, tells us that the radiation pattern of the array is the same as the receiver pattern, if all the sources are swapped for receivers. In our case, if we were to use the array as a receiver it would have a resolving angle of about 10 degrees. To improve this we would have to use a bigger array or perhaps two arrays that could be separated by some distance. A photograph of the complete system is shown in Fig. 10. A relatively easy extension of our system would be to split the array into parts that could be separated, or simply replace the current mount, with one having greater transducer separation.

Measurements of array performance

To measure the spatial wavefield, a simple x-y platform using extruded aluminum beams and two stepper motors was built. A microphone was mounted on this and the resulting sound pressure level was measured in the mid-plane of the array. For three-dimensional measurements the microphone was moved up and down.

The measurements shown next were made at 5 kHz, where the wavelength is 0.07 m. The Fraunhoffer distance (which separates the near-field and far-field radiation zones) is about 3 m at 5 kHz. Thus our laboratory measurements are all technically in the near-field or transition zone. The width of the central radiation lobe in the measurement plane closest to the array (0.2 m) is on the order of one wavelength. Even so, we have a well-defined and directional beam; as can be seen in Figs. 11 and 12.

Figure 12 shows a 2D section of the data, in a plane perpendicular to the array. A simulation (right) and measurements (left) are shown for a target location at (0.3,0.5). The beam is elongated in the direction of propagation, whereas transverse to the beam, a Gaussian fit to the main lobe gives a full-width at half-max of almost exactly one wavelength (0.07 m) (See Fig. 11).

Extremely high sound pressure levels (SPL) are achievable

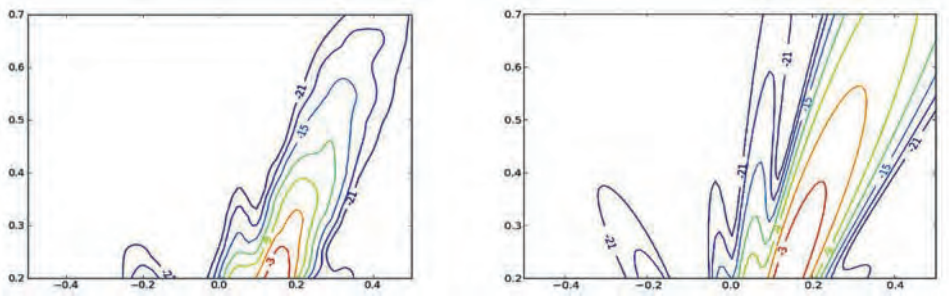

Fig. 12. Measurement (left) and simulation (right) for a target (focus) at (0.3,0.5). NB the measurement plane is at .2 m in front of the array.

with relatively inexpensive audio tweeters. In fact, since our SPL meter saturates at 130 dB, we can only estimate the total SPL. But it is worth noting that a phased array allows for the coherent superposition of the individual radiating elements and so we can achieve higher SPL than would otherwise be possible.

Summary

Using low-cost, off-the-shelf audio amplifiers and tweeters, we have designed and built an active acoustic phased array. The phasing, or time-shifting, of a single waveform is done by a bank of field programmable gate arrays which are connected to a personal computer via an Arduino microcontroller. Although we have shown application to focusing or steering the array, it is possible to imagine much more sophisticated applications where the time delays are chosen to produce more complex space-time patterns. The modeling code which is available at

http://mesoscopic.mines.edu/mediawiki/index.php/Acoustic_Phased_Array

does a full field calculation; but it could easily be simplified for far-field application, which would greatly speed it up. In addition to the simulation code (which is written in Python), the design and materials used in the array are also available on the web site.

This material is based upon work supported in part by the U.S. Office of Naval Research as a Multi-disciplinary University Research Initiative on Sound and Electromagnetic Interacting Waves under grant number N00014-10-1-0958. AT

References

Azhari, H. (**2010**). *Basics of Biomedical Ultrasound for Engineers* (John Wiley & Sons, Inc., Hoboken, 2010). ISBN 978-0-470-46547-9.

Chu, P. P. (**2008**). *FPGA Prototyping by VHDL Examples: Xilinx Spartan-3 Version* (John Wiley & Sons, Inc., Hoboken, 2008) ISBN 978-0-470-18531-5.

Cobbold, R. S. C. (**2007**). *Foundations of Biomedical Ultrasound* (Oxford University Press, New York, 2007). ISBN13: 978-0-19-516831-0.

Dowling, A. P. (**1998**). "Steady-state Radiation from Sources," in *Handbook of Acoustics* edited by Malcolm J. Crocker (John Wiley & Sons, Inc., New York, 1998).

Pedroni, V. A. (**2004**). *Circuit Design with VHDL* (MIT Press, Cambridge, 2004). ISBN 0-262-16224-5.

Martin Smith always wanted to be a scientist and through good fortune he was able to enjoy a career in geophysics. Since retiring from a small company, he has spent his time taking courses, volunteering at the local science museum, and consulting for a few clients. All of this has turned out to be much more fun than he expected. He and his wife, Terri, live in Sharon, Vermont, with four dogs and a bunch of great neighbors.

Kurt Strovink is a graduate student in physics at the Colorado School of Mines. His research interests include optics, computational physics and imaging. He enjoys applied experimental work and hopes to remain in that field.

Mike Roddewig received a bachelor's degree in electrical engineering from Michigan Technological University in 2009, a masters degree from Colorado School of Mines in 2012, and is currently a doctoral student at Montana State University in the Optical Remote Sensing Laboratory. His area of research is on optical remote sensing of the environment.

John Scales is a Professor of Physics at the Colorado School of Mines. He is on his third or fourth career, having started in computational and mathematical physics, then delved into inverse problems in seismic wave propagation. This led to an abiding interest in random dynamical systems, ergodic theory, waves in random media and quantum chaos. For the last 15 years he has built up a unique experimental center focused on sub-millimeter wave electromagnetic and acoustic wave propagation in random media—perusing both fundamental problems such as Anderson localization and the practicalities of material characterization and near-field imaging in strongly heterogeneous media. Scales has supervised a score of Ph.D. and MS students in Physics, Geophysics, Applied Mathematics and Electrical Engineering. In his spare time Scales races bicycles on the road and in cyclocross. He and Pamela have been married for 34 years and have two children currently at university.