

Ultra-Directional Microphones: Part 4

James A. Moorer
Sonic Solutions

Abstract

In a series of articles dating from the early 1970's, Michel Gerzon suggested using cancellation between two adjacent microphones to achieve high directionality in a limited frequency range. In this paper, we extend this analysis to linear arrays of microphones by borrowing certain aspects of phased-array radar. The unique issue that audio has is the requirement that the frequency response be flat over 5 octaves or more. We show that this requirement can be met by the use of multiple colinear arrays, followed by a significant amount of signal processing.

Background

Derivation of the Phased-Array Microphone

We start with an array of microphones placed at equal distances along a line. Let d be their separation. Let a plane wave impinge on the array at an angle of θ from the perpendicular to the array. Assume that the plane wave is a sinusoid with a wavelength of λ . If n is the number of microphones, then we can write the response to the plane wave in microphone k as follows:

$$(1) \quad \sin\left(2\pi \frac{c}{\lambda} \left(t + \frac{kd}{c} \sin \theta\right)\right)$$

For convenience, we let the number of microphones be odd, and we call the center microphone number zero. The variable t represents time in seconds. If we sum these signals over all the microphones and simplify, we obtain the following:

$$(2) \quad \sin\left(2\pi \frac{c}{\lambda} t\right) \left\{ 1 + 2 \sum_{k=1}^{(n-1)/2} \cos\left(2\pi \frac{kd}{\lambda} \sin \theta\right) \right\}$$

The second term of the above represents the amplitude of the resulting sum. This is plotted for various values of wavelength in Figure 2. Note that the maximum response is developed in a direction perpendicular to the microphone array. The varying width of the response maximum show that different wavelengths will have different pickup patterns.

We can "steer" the entire array by applying a simple delay to each microphone as follows:

$$(3) \quad \Delta t_k = \frac{d}{c} \sin \theta$$

where θ is the angle where the greatest sensitivity is desired.

This has the effect of moving the maximum of the response of the array, but it also changes the width of the center lobe. Figure 3 shows the effect of “steering” the array from -45° to 45° . Note that the main response widens a bit as the array is steered away from the center. This is because the “effective” microphone spacing is reduced by the cosine of the angle.

Since the amplitude term in equation (1) resembles a Fourier series, we might envision the use of window functions to change the tradeoff between center lobe width and side lobe suppression. Indeed, it works pretty much as one might anticipate. Figure 4 shows the effect of changing the strength of the window. We can see clearly the increase in lobe width with increasing window strength.

So far, this is all taken directly from phased-array radar technology. To make this useful for audio, we need to accomplish the following:

- Produce a uniform lobe width over all frequencies.
- Achieve 10-octave range with flat frequency response (roughly 20 Hz to 20 kHz)

The reason we want uniform lobe width is to reduce the coloration of the sound in the principal direction of the array. Since the array depends on cancellation and reinforcement of the wave fronts, it is necessarily a highly frequency-dependent process. We need to follow it with sufficient processing to minimize the frequency dependencies.

The basic array exhibits reasonable response over about 2 octaves covering wavelengths from about $1.5d$ and $6d$. Wavelengths longer than this produces very wide principal lobes, and wavelengths shorter than this produce multiple principal lobes. We can take the center octave of this (in a geometric -mean sense) as the main region of response, which is from about $2.12d$ to about $4.14d$. The remainder of the response range will be used to overlap with other arrays that cover other octaves.

We obtain wide response by having multiple arrays on the same line with the same microphone in the center. Figure 5 shows a simplified diagram with three colinear arrays with spacings at d , $2d$ and $4d$. To cover the full audio range with equal spatial resolution would require a total of ten arrays. Each array will contribute one octave of frequency response to the overall result. The upper and lower half-octave of each array will overlap with the adjacent arrays.

Controlling the Width of the Principal Lobe:

The next problem to be addressed is control of the width of the principal lobe. As noted above, a window function can be used to adjust the width of the center lobe. Since we need a different lobe width at each different frequency, we must filter the output of each array with individual filters that are designed to realize a certain window function at each frequency. The filters have a further requirement that they sum properly with the responses of adjacent arrays to produce flat frequency response and uniform lobe width when summed over all the arrays.

Since window functions always make the lobe wider and never more narrow, we must take the widest lobe width and match all the other widths to this. The widest lobe in the range of interest occurs at $6d$. By a simple optimization, we can derive values of the beta parameter of the Kaiser-Bessel window that give us the desired window width. Figure 6 shows the result of such an optimisation. As the wavelength moves from $6d$ down to $1.5d$, the beta parameter can be increased steadily to widen the principal lobe. Figure 7 shows the result of applying different window functions to the array at different wavelengths. Note that at the shortest wavelength, the sideband rejection starts to rise again, probably due to the effective “shortening” of the array.

There is nothing particularly special about the Kaiser-Bessel window. It is used here simply because it comes with a single parameter that controls the width of the window in a smooth, continuous, and monotonic fashion. One could equally derive an “optimum” window by a least-squares technique. This

would allow us to “fine tune” the response at any given frequency by adjusting the tradeoff between matching the center lobe to the prototype response (which is the response at the longest wavelength, $6d$) to the off-axis response. We note in Figure 6 that the off-axis peaks get greater as the wavelength gets longer. This is to be expected, since smaller values of Beta allow the sidelobes to increase in amplitude. We can define a window function, w_k , then define a weighting function at each angle as p_i . We may then describe an objective function as follows:

$$(4) \quad F = \sum_{i=1}^M p_i \left\{ D_i - 1 - 2 \sum_{k=1}^{(n-1)/2} w_k \cos(2p \frac{kd}{I} \sin \mathbf{q}_i) \right\}^2$$

where D_i represents the “desired” response. In our case, we might produce a desired response by windowing the response at the maximum wavelength of $6d$. Using this as the prototype response, we can match this as closely as we like by choosing the weighting function, p_i , and finding the window function coefficients, w_k , that minimize F in equation (4). Since the response of the array is linear with respect to any given window coefficient, equation (4) represents a linear least-squares problem. The normal equations can be formed and solved by any number of methods, such as singular-value decomposition [***ref]. One might choose, for instance, $p_i \equiv 1$ to match the desired response as well as possible over the entire function. One might choose $p_i = 10$ over the main lobe and $p_i = 1$ to force the response to match the desired response as well as possible at the main lobe and less well outside the main lobe.

Since the Kaiser-Bessel window is relatively simple, we will use this in the remainder of this discussion with the understanding that any suitable window that allows matching of the principal lobes can be used.

Implementing the Frequency-Dependent Window:

To implement a window function that varies with frequency, we must implement a filter for each microphone that has the desired gain at each wavelength. This gain is determined by the value of the Kaiser-Bessel window for that microphone at the value of beta indicated by the curve of Figure 6. The resulting window function is, in fact, a family of window functions, since the window function will be different for each different frequency. We might represent this as $w_k(\mathbf{I})$ for the weighting of microphone k at a wavelength of \mathbf{I} . Figure 7 shows a plot of four different microphone coefficients as functions of wavelength. These represent the filters that must be realized to produce equal main lobe widths over the frequency range of interest. There are many ways to calculate the filter coefficients [***refs McLellan, Dzecky, etc], so this aspect need not be discussed any further here. Since a filter will respond over the entire range, we do need to specify the curves outside of the range shown in Figure 7. It is sufficient to just extend the curves to zero frequency and the Nyquist rate by simply duplicating the values at the end points shown in Figure 7. That is, the response of the filter at wavelengths greater than $6d$ can have the same response at a wavelength of $6d$, and wavelengths shorter than $1.5d$ can have the same response as at a wavelength of $1.5d$. These values are somewhat arbitrary but are sufficient to produce a working design.

Note that window functions are symmetric. This means that for an array of n microphone, only $(n-1)/2$ windowing filters need be implemented. Microphones on each side of the center microphone may be summed before filtering, thus eliminating the need for a number of filters.

Overlapping the Arrays:

As noted above, each array covers about two octaves. We will separate this into the *main* region from about $2.12d$ to about $4.14d$, and the *overlap* regions which constitute the remainder of the full two octave range. At the extremes of the frequency range, there is no overlap, so the highest array will cover up to $1.5d_j$, and the lowest array will cover down to $6d_1$, where d_j represents the microphone spacing of array j . Using 24 kHz as the highest frequency for which coverage is desired, we can set the spacing of the microphones in the highest frequency array as about 1 cm. From this, we can derive the following:

Microphone Spacing	Low Frequency	High Frequency
1 cm	8000 Hz	22067 Hz
2 cm	4000 Hz	8000 Hz
4 cm	2000 Hz	4000 Hz
8 cm	1000 Hz	2000 Hz
16 cm	500 Hz	1000 Hz
32 cm	250 Hz	500 Hz
64 cm	125 Hz	250 Hz
1.28 m	62.5 Hz	125 Hz
2.56 m	22.11 Hz	62.5 Hz

These frequencies are not exact, they have been rounded to convenient boundaries for clarity. Note again that the highest frequency array extends from $1.5d$ to $4.14d$, and the lowest frequency band extends from $2.12d$ to $6d$. All the others extend from $2.12d$ to $4.14d$. This shows that the entire frequency range may be captured by 9 collinear arrays. If desired, the larger arrays at lower frequencies may be eliminated. The only effect of this is that the pickup will not be highly directional at low frequencies due to the widening of the principal lobe of the array response.

Note again that steering the array away from angle zero (straight ahead) does have the effect of widening the principal lobes, since it lowers the effective distance between the microphones. This table was computed at angle zero. We might choose the table based on a different angle. To be as consistent as possible, we should compute a different set of frequency-dependent window functions for each desired pickup angle so that the principal lobe width would be constant over the entire steering range of the array, which is from -45° to 45° . For many applications, however, it is acceptable to allow the width of the principal lobe to change, as long as other properties of the array are preserved, such as overall frequency response flatness, and matching of the principal lobes among the arrays to prevent coloration of the sound in the principal lobe.

In addition to the filtering described above to apply the frequency-dependent window function to each microphone in each array, there is a filter that must be applied to the total response from a given array so that each array contributes to the overall response mainly in its principal frequency region. We also require that the sum of the responses across all the arrays be flat over the audible range. We may express this by

considering the impulse response of each array, then stating conditions on these responses which represent the design goals. We may say for convenience that the impulse response of each array will be symmetric. This is not strictly necessary, but it guarantees that there will be no phase variance from one array to the next. If we represent the impulse response of filter i by h_{is} , then we may state the conditions for flatness of overall frequency response as follows:

$$(5) \quad \sum_i h_{is} = \begin{cases} 1, & s = 0 \\ 0, & s \neq 0 \end{cases}$$

This is necessary and sufficient to guarantee perfectly flat frequency response. In general, this condition will not be met exactly. All we require is that the deviation from identity be sufficiently small so it is not heard as a coloration of the sound.

To compute the overlap filters, we first create an “ideal” prototype filter that is constructed so that it overlaps perfectly. We then compute approximations to the prototype filter using standard approximation techniques [***refs Parks McLellan, etc]. Although we need to construct a separate prototype filter for each band, there are some similarities that make the process simpler. We can separate the filters into the two at the extremes of frequency, and all the rest. For the filters that are not at the extremes, we can require that they are identical, except that each band spans twice the frequency of the previous band. If we say that a particular frequency band goes from f to $2f$, then we may define a filter as follows:

$$(6) \quad f_c \equiv (4/3)f$$

$$(7) \quad f_1 \equiv (2/3)f$$

$$(8) \quad f_2 \equiv (8/3)f$$

$$(9) \quad H(\mathbf{J}) = \begin{cases} 0 & \mathbf{J} < f_1 \\ (f_1 - \mathbf{J}) / (f_1 - f_c) & f_1 \leq \mathbf{J} < f_c \\ (f_2 - \mathbf{J}) / (f_2 - f_c) & f_c \leq \mathbf{J} < f_2 \\ 0 & f_2 \leq \mathbf{J} \end{cases}$$

Figure 10 shows a plot of this function for the frequency band 2000-4000 Hz. As noted, the filter extends down to 1333 Hz and up to 5333 Hz. It will perfectly overlap the filters in the next higher and next lower frequency bands, and the sum of these overlapping filters is exactly one by construction. This is only one way that prototype filters may be chosen. There are any number of prototype filters that have this property.

At the extremes of frequency, we simply allow the filter to stay at unity gain on one side or the other. Using the definitions above, we may define the filters for the extremes as follows:

$$(10) \quad H(\mathbf{J}) = \begin{cases} 1 & \mathbf{J} < f_c \\ (f_2 - \mathbf{J}) / (f_2 - f_c) & f_c \leq \mathbf{J} < f_2 \\ 0 & f_2 < \mathbf{J} \end{cases}$$

$$(11) \quad H(\mathbf{J}) = \begin{cases} 0 & \mathbf{J} < f_1 \\ (\mathbf{J} - f_1)/(f_c - f_1) & f_1 \leq \mathbf{J} < f_c \\ 1 & f_c \leq \mathbf{J} \end{cases}$$

We are being somewhat careless with the notation, in that the above formulas all use the same symbols for the important frequencies (f_1 , f_2 , and f_c), but we intend them to apply just to the particular band of interest. As noted above, for the band from 2000 to 4000 Hz, f_1 would be 1333 Hz, and f_2 would be 5333 Hz. For other bands, these frequencies would be scaled appropriately to represent the frequency range of the particular band. As an example, in the lowest band as shown in the table above, f_c would be 41.667 Hz, and f_2 would be 83.333 Hz. Equation (10) represents the lowest filter, which extends down to zero frequency.

Having defined a suitable set of prototype filters for overlapping the microphone arrays, we may compute filter coefficients that approximate these filters to any degree of accuracy. If the filters are all of zero-phase, then they will sum to an approximation of an impulse, described by Equation (5). This is by construction. Since the sum of all the prototype filters is unity, the resulting impulse response must be a simple impulse. Consequently, the sum of a series of filters that approximate the prototype filters will naturally be an approximation to an impulse. Of course, if the filters are not of zero-phase design, they will not necessarily sum to an impulse.

We should point out that as we steer the array so that the principal lobe is at a non-zero angle, the effective shortening of the microphone spacing by the factor of $\cos(\mathbf{q})$ indicates that all the filters, both the windowing filters and the overlapping filters, should be recomputed using a microphone spacing of $d \cos(\mathbf{q})$. Additionally, we can adjust the Beta parameter of the Kaiser-Bessel window (or whatever window function is used) so that the width of the principal lobes remains constant over the usable steering range of -45° to 45° .

There has been an implicit decision in the above to implement the frequency-dependent window function and the overlapping filter using FIR, or finite impulse-response filters. This is not strictly necessary, but it allows us to use linear-phase filters. A linear-phase filter has an inherent delay in the signal path. If all the filters have the same number of multiplies, then they will all exhibit the same delay, and they may be summed. If the filters do not have the same number of multiplies, then we will have to equalize the delays before summing the results of the windowing filters. We can offset these delays by combining them with the delays necessary for “steering” the array (Equation (3)). If some microphones end up with negative delays, then all the microphones must be delayed to assure causality.

About Directional Microphones:

So far, we have not discussed the directional characteristics of the individual microphones in the array. This discussion is perfectly accurate if the microphones are omni-directional. Some modifications to the exposition will have to be made to show the effect of directional microphones, such as the pressure-gradient type. Figure 11 shows a schematic representation of a pressure-gradient microphone. There are two diaphragms that are used to generate a voltage. These may then be weighted and summed to produce a directional pickup. This kind of microphone has the following angular response:

$$(12) \quad C + (1 - C)\cos(\mathbf{q})$$

The response straight ahead (zero angle) is exactly one. The response to the rear is $(2C-1)$. For a cardioid pattern, C is set to one-half, so the response to the rear is exactly zero. Other values of C produce different patterns.

The effect of using a pressure-gradient microphone in this array is that the off-angle response will be multiplied by the directional pattern described by Equation (12). The effect would be that, for instance, the plot shown in Figure 3 would also show an amplitude difference as the principal lobe was steered from left to right. All the curves in Figure 3 would be multiplied by Equation (12). Note that we can easily normalize the peak amplitude of the principal lobes in Figure 3 by simply correcting for the expected attenuation due to the directional characteristics of the microphones.

As Gerzon noted in his seminal work in this domain [***ref], it is also possible to take the voltages from the anterior and posterior diaphragms separately, thus producing two separate feeds from each microphone. These can then be combined later to produce directional characteristics. For instance, we might weight the anterior diaphragm by one-half and the posterior diaphragm by minus one-half and sum them to produce a forward-facing cardioid pickup, with 100% rejection of sounds coming from directly behind. Alternately, we might weight the posterior diaphragm with one-half and the anterior diaphragm with minus one-half to produce a rear-facing cardioid pickup with 100% rejection of sounds coming from directly in front. In this manner, using a single array of pressure-gradient microphones, we can mix the feeds of the diaphragms differently so that the same microphone array may be used for sounds in front of the array and behind the array with equal angular resolution and identical fidelity (frequency-response). Of course, the filtering shown in Figure 9 would have to be duplicated for the rear-facing array.

Curvature of the Wavefront

With phased-array radar, there is always the explicit assumption that the incoming wave is a plane wave. With the phased-array microphone, the plane wave assumption may be used when the sound sources are sufficiently distant from the microphone itself. If this is not the case, the wavefront will be curved. We can correct for this curvature, but we need to know the location of the sound source to make this correction. If the plane-wave approximation can be made, then we need not know the distance between the sound source and the array.

To correct for the curvature of the wavefront, we need to apply a correction to the amplitude and to the arrival time. The amplitude correction is needed to offset the $1/r^2$ attenuation the wavefront experiences. The correction to the arrival time is necessary since the curvature will have the effect of delaying the off-center parts of the wavefront. We can quantize this as follows: Let \mathbf{q} and r_0 be the angle and distance from the sound source to the center microphone of the array. We can then describe the amplitude and time delay compensation as follows:

$$(13) \quad P_n = \frac{r_n^2}{r_0^2} = \cos^2 \mathbf{q} + \left(\sin \mathbf{q} - n \frac{d}{r_0} \right)^2$$

$$(14) \quad \Delta_n = \frac{r_n - r_0}{c} = \frac{1}{c} \left\{ \sqrt{r_0^2 \cos^2 \mathbf{q} + (r_0 \sin \mathbf{q} - nd)^2} - r_0 \right\}$$

where r_n represents the distance from the sound source to microphone n . The feed from microphone n should be multiplied by P_n and should be advanced by Δ_n seconds.

Since this correction is specific to the particular location of the sound source, we would expect that the rejection of the off-axis sound would be affected. Indeed, we will experience more "leakage" from off-axis sounds when this kind of correction is applied.

Further Sharpening of the Response

Note that when the sound source consists of a number of discrete sources at known angles and possibly known distances, then the response in a particular direction can be enhanced by subtracting off the signals from the known directions. Of course, the delays across the varying angles must be equalized before a signal from one angle can be subtracted from a signal from another angle. We might think of this as a kind of analog to the lateral inhibition found in optical receptors in the retina of the eye.

Microphone Mismatch

So far in this exposition, we have operated under the implicit assumption that the microphones were identical. This is, of course, not a valid assumption: there will be some mismatch. We should examine the effect of the mismatch and see what this requires of the microphones.

We can obtain a worst-case bound on the error in the array by taking the second term of Equation (2), applying a window function, assuming that the cosine term is always unity, and assuming that the microphone error is a uniform factor of ϵ . This gives us the following upper bound:

$$(15) \quad M = \epsilon \left\{ w_0 + 2 \sum_{k=1}^{(n-1)/2} w_k \right\}$$

The window function is normalized so that the above sum (across all the points of the window function) is unity, so the error is bounded by the individual microphone error. We can take ϵ to represent the expected value of the error. Some microphones will exhibit somewhat more error and some will exhibit somewhat less.

A mean deviation of 1 dB then will produce error in the resulting pickup pattern that is about 18 dB down. The error we are talking about is a distortion of the pickup pattern itself, as shown in Figures 2, 3, and 4. This is not so important for the principal lobe, but it will make a significant difference in the sideband suppression, since in some cases, the error will be of the same order of magnitude as the sideband amplitude itself. We can expect that the actual sideband rejection will be several dB less than the theoretical values with a 1 dB variation among the microphones. Of course, better matching will allow us to achieve more sideband rejection.

Effects of Room Reverberation on the Array

So far we have discussed sounds coming from point sources that are in front of (or behind) the array. What happens when we have room reverberation, which can come from any direction? We may (somewhat artificially) divide room reverberation into three epochs: the direct sound, the early reflections, and everything else. The direct sound and the early reflections can all be treated as point sources of sound. The array can be steered to pick up each one of these sources separately (or not, depending on the goals of the recording). The late reverberation can be considered to be omnidirectional, and will thus affect the array uniformly regardless of the steering direction. Of course, non-uniform reflections, such as slap echos, will appear as specular reflections and thus will appear as point sources to the array.

Extension to 3 Dimensions

To extend the phased-array microphone to three dimensions, we must first extend it to two dimensions. This can be done by extending the array as shown in Figure 12. This shows a regular 2-dimensional array of microphones that is capable of steering plus or minus 45° in the horizontal direction and plus or minus 45° in the vertical direction. Note that for some applications, it may not be necessary to have the same resolution in the vertical direction as in the horizontal direction. Figure 13 shows an array with higher resolution in the horizontal direction than in the vertical direction.

A single 2-dimensional array can only be steered across about a 90° range in the forward direction and a 90° range in the reverse direction. To allow steering through the full 360° range, we need to use two arrays at

right angles as shown in Figure 14. Note that for this to work, each array would have to be acoustically “transparent”, so that off-axis sounds will easily pass through it to reach the other array.

To extend the array to three dimensions, we take two 2-dimensional arrays shown in Figure 14 and place another array in the horizontal plane to cover the vertical direction. In this manner, we may achieve pickup in any direction.

Construction of the Array

There is a wide range of ways to implement the array, depending on the goals of the implementation.

The most straightforward way to construct the array would be to simply connect wires to each transducer in the array and run all the wires to the required processing hardware. Of course, there must be preprocessing for each transducer in the form of a microphone preamplifier and an A/D converter. Figure 15 shows the processing necessary for each microphone in the array. Of course, different technology can affect the elements in the figure. For instance, the use of electret or other microphone technology may render the pre-amplifier unnecessary. Similarly, it is possible to combine the microphone preamplifier (if any) with the first stage of the A/D converter. In any case, the result of the preprocessing is a sequence of digital audio samples. Since a large array may contain hundreds of microphones, running individual wires from each microphone to the required pre-processing and subsequent processing may be undesirable.

With modern technology, high-levels of integration are possible. Both analog and digital circuitry can be put into the same package, if not the same substrate. See, for instance, US patent 5,051,799 [Paul *et al*]. It is possible to produce a very compact realization of the preamplifier and D/A converter. It is even possible to combine the microphone preamplifier with the first stage of the D/A converter for even a more compact realization. Such circuitry can be on the order of the same size as the microphone capsule or even smaller. Figure 17 shows the idea of including the preprocessing and A/D conversion in the same physical location as the microphone capsule itself. In addition, some kind of data multiplexing circuit is included with each microphone so that the outputs of multiple microphones may be combined into a single wire. A wide range of multiplexing technology may be used, ranging from simple time-domain or frequency-domain multiplexing [US patent 4,922,536, Hoque] to computer-type network technology, such as Ethernet [Metcalf and Boggs, 1976]. The end result of this multiplexing is that the data from the entire array is available in a small number of cables, or even a single cable, in a manner such that the samples from each individual microphone may be separated for the required spatial processing as shown in Figure 9.

Figure 18 shows the logical extension of this train of logic to the microphone array “fabric”. Here, we show power being fed to each transducer/processor/multiplexor node via alternating vertical positive and negative supply wires. One of the wires can also be used as the network medium itself by AC-coupling the data back onto the wire. Similarly, clock distribution to the individual A/D converters may be accomplished by placing the clock itself on one of the supply wires. By use of frequency-domain multiplexing, the data can be placed on the wire in frequency bands that are well above the clock frequency.

Note that the entire array could just as easily be wireless (except for the supply rails). Each node could simply broadcast a low-power RF signal that could be received and demultiplexed for further processing. Each node would have to have some unique ID in the form of a network address, a dedicated frequency, a dedicated time slot, or any other way of identifying the node so that the samples may be recovered and related back to the original array position of the node.

It is clear that any medium of transmission could be used to convey the data from the array to the processing elements. For instance, each node could emit digital data as light on wavelengths that people can not see. The data could be multiplexed either by the wavelength of the individual lights, or by time so that only one node transmitted data at a time.

Hybrid schemes are also possible. That is, “clusters” of some number of nodes in a particular area could be multiplexed together with, say, fiber-optic cables used to relay the data from each cluster back to the spatial processing equipment.

Relation to Sound-Field Theory

In so-called “sound field” theory [***refs Gerzon], we expand the sound pressure wave about the listener in a series of spherical harmonics [***refs Hobson, MacRoberts]. This is not an artificial construct. It falls directly out of the solution to Laplace’s equation in spherical coordinates [***Morse & Feschbach]. To the extent that air is linear, sound waves will obey Laplace’s equation, and thus the sound field around a listener can always be represented as a sum of spherical harmonics. This sum is not necessarily finite. If the sound source is a true point source, then the sum will not be finite. It can be approximated by a finite sum. As is typical with this kind of expansion, applying a window function can help smooth out the overshoot (“Gibbs-type” phenomena) inherent in truncating an infinite series.

The point of making this expansion is that it gives a rational basis for trying to recreate the recording environment at the time of playback. The idea is that if we can recreate the spherical harmonic expansion of the sound field about the listener, then we have recreated the waveform at one point in space. This assertion is not controversial: it is a tautology. What can be argued is how many spherical harmonics are necessary to do a good job of reconstructing the sound field. I have no particular wisdom to offer on this point except that more is better.

The problem with actually doing this is two-fold: first is that we need at least one speaker for each harmonic that we wish to reproduce, and second is that modern microphones are only capable of first-order directional patterns, as noted in Equation (12). The point of the phased-array microphone is that it is possible to use this directionality to directly measure the higher-order harmonics of the sound field around the center microphone of the array. By using more and more microphones in the array, the directional pattern can be made arbitrarily narrow. Consequently we can recover any number of terms of the spherical harmonic expansion about the center microphone by increasing the number of microphones.

References:

[S] Paul, J.D.; Clayton, M.D.; Agnello, A.M. “Digital Output Transducer,” US Patent 5,051,799, issued September 24 1991.

[S] Hoque, T.I. “Digital Audio Transmission for Use in Studio, Stage, or Field Applications” US Patent 4,922,536, issued May 1 1990.

[S] Metcalfe, R.M., and Boggs, D.R. “Ethernet: Distributed Packet Switching for Local Computer Networks”, *Communications of the ACM*, Volume 19, Number 7, pp395-404, July 1976

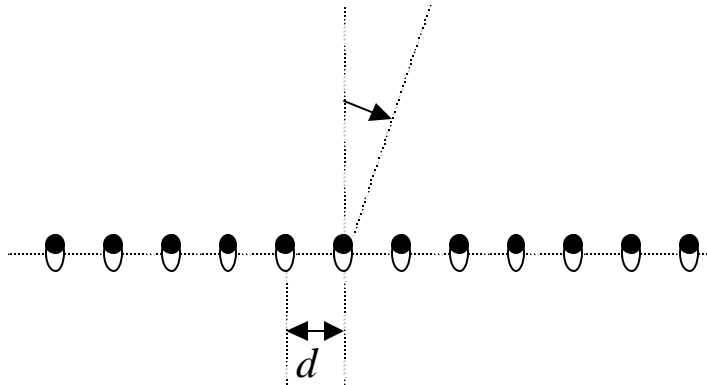


Figure 1 – A linear array of microphones with a spacing of d . We assume a plane wave impinges on the array at an angle from the perpendicular to the array.

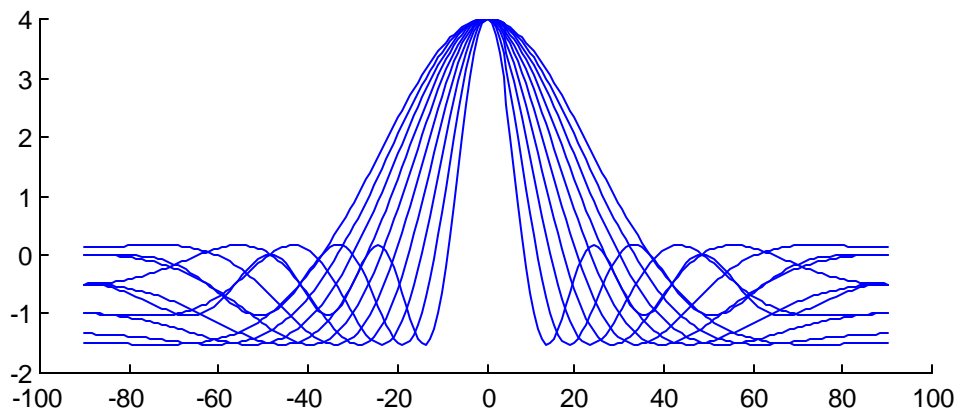


Figure 2 – Amplitude of the response of the sum of all the feeds from the microphone array with changing angle of incidence. Each curve represents a different wavelength from $1.5d$ (narrowest) to $6d$ (widest).

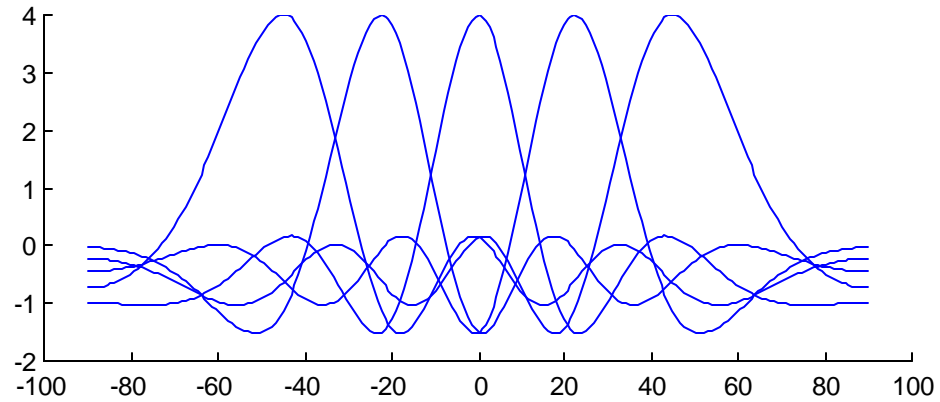


Figure 3 – This shows the effect of “steering” the array by adding a simple delay to each microphone. The wavelength of the test signal was set to a constant $2.5d$. Note the widening of the principal lobe as we steer the array away from directly in front. This is due to the effective narrowing of the microphone spacing by a factor of $\cos(\mathbf{q})$.

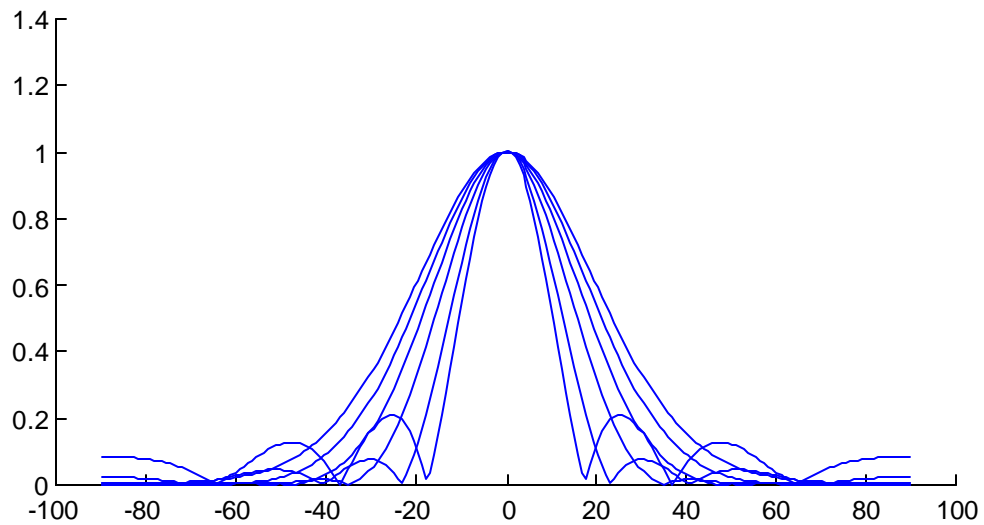


Figure 4 – this shows the effect of using a window function to change the tradeoff between center lobe width and side lobe suppression. The window was the Kaiser-Bessel window with the parameter varying between 0.5 and 5.5.

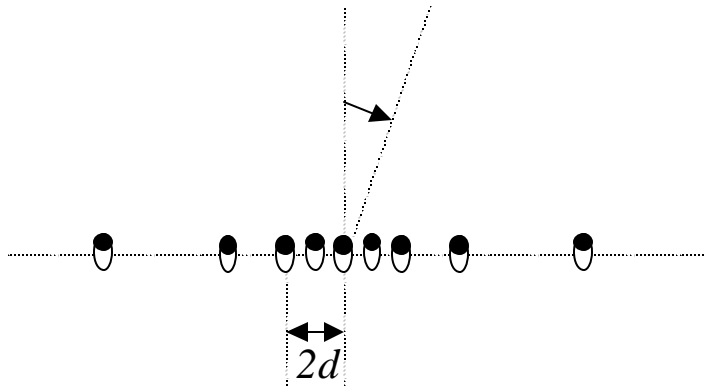


Figure 5 – Three overlapping arrays sharing center microphones. The arrays have spacings of d , $2d$, and $4d$. To attain full frequency response over the audio range with equal spatial resolution at all frequencies, a total of at least ten colinear arrays would be required.

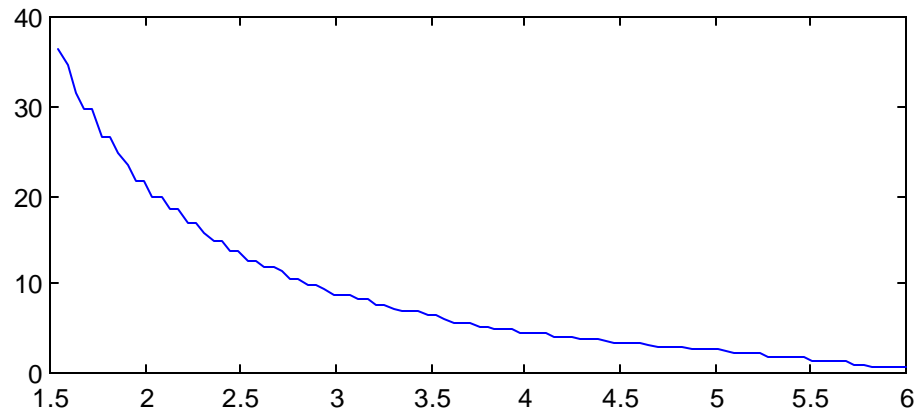


Figure 6: Plot of Beta parameter to Kaiser-Bessel window for values of wavelength in multiples of the microphone spacing. These values of Beta equalizes the main lobe widths for the given wavelength. This curve appears to be largely independent of the number of microphones in the array.

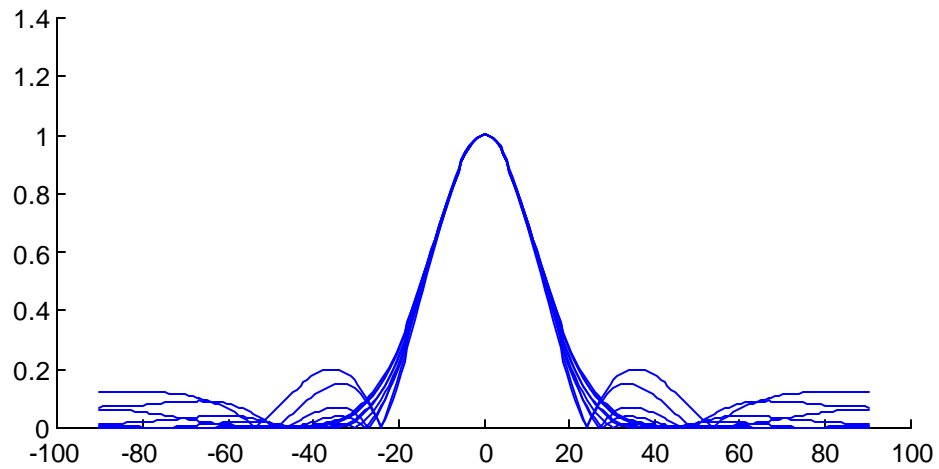


Figure 7: Lobe widths after normalization by adjusting the Beta parameter of the Kaiser-Bessel window. The wavelengths span the range from $1.5d$ to $6d$. Note that the sideband gain increases at the ends of the frequency range due to the windowing. This is using 15 microphones in a single array.

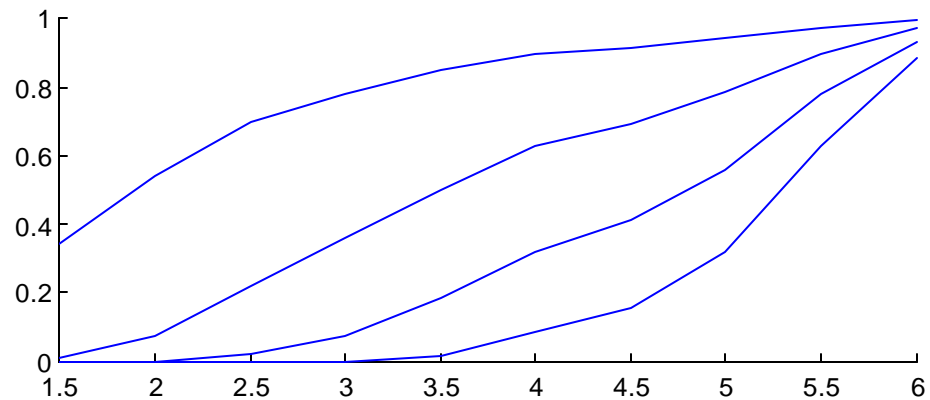


Figure 8: Typical windowing gain curves for four microphones in a 9-microphone array at various values of wavelength (in multiples of d). These represent particular points of the Kaiser-Bessel window as the Beta parameter is swept as shown in Figure 6. The upper curve represents the center microphone, and the center point of the window function.

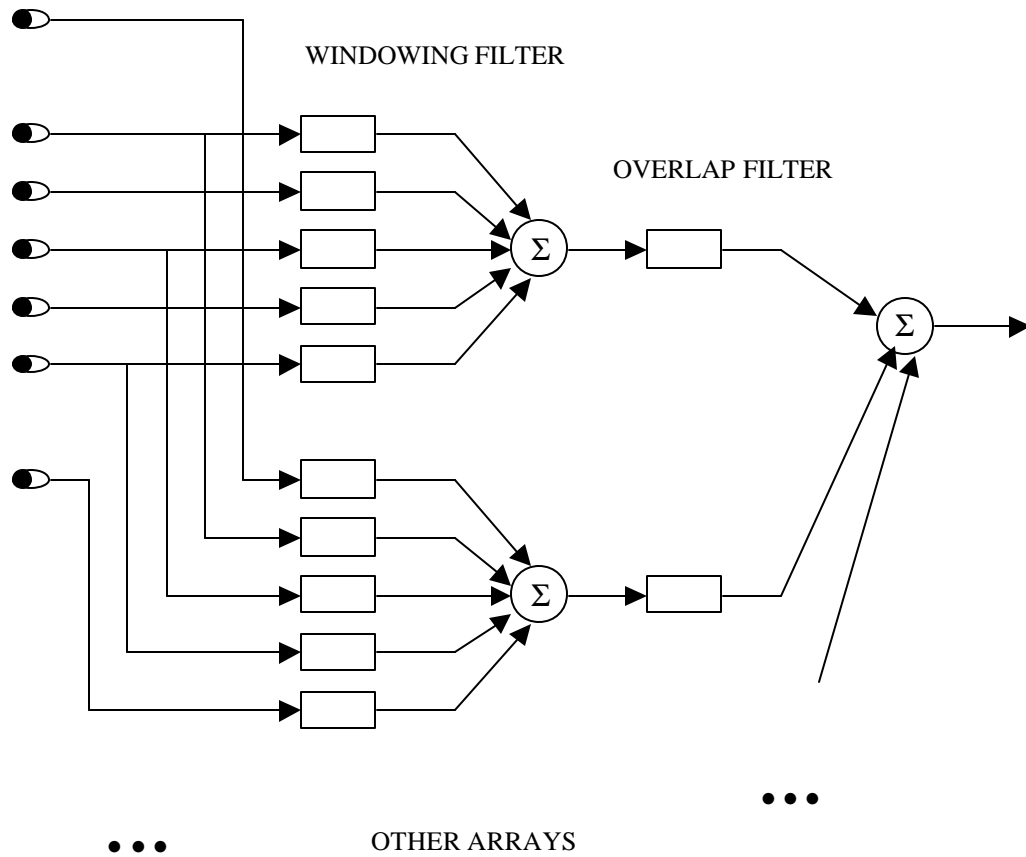


Figure 9: Complete diagram of processing for overlapped microphone arrays. Each microphone goes to a filter that implements the frequency-dependent window and the “steering” delay. Each windowed array is then filtered so that the arrays overlap properly to produce an overall flat response. One windowing filter is shown for each microphone for clarity. Since the window functions are symmetric, pairs of microphones equidistant from the center microphone would be summed, then filtered by a single frequency-dependent window filter. If it is desired to simultaneously receive signals from different directions (that is, with the array “steered” to different angles), then separate processing would have to be supplied for each desired angle. Of course, the direct microphone feeds could be stored and processed to extract signals at different angles at a later time.

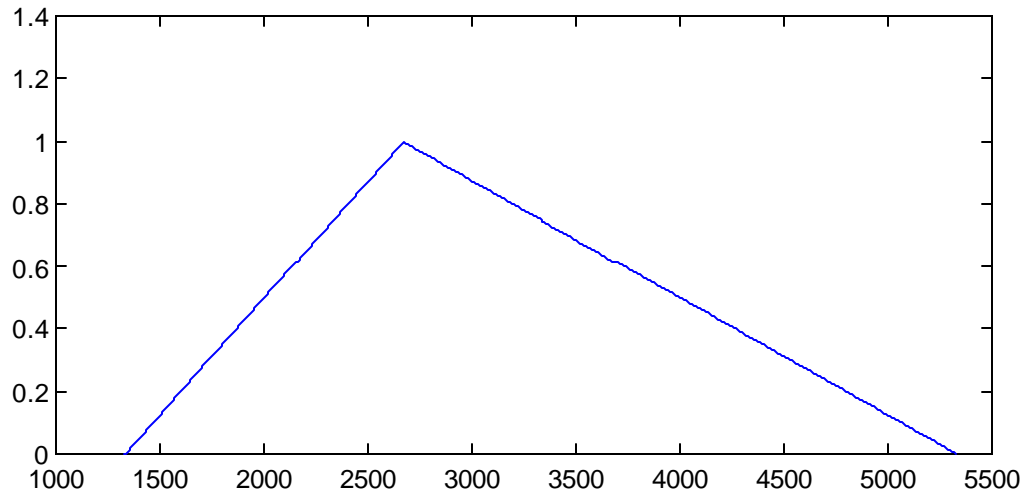


Figure 10: One kind of prototype filter covering the band from 2000 Hz to 4000 Hz. For proper overlap, the filter extends into the adjacent bands from 1333 Hz to 5333 Hz. The filter for the next higher or lower frequency band may be obtained simply by relabeling the frequency axis with either twice the frequencies or half the frequencies. Of course, this filter design is not unique. There are many suitable choices for the overlap filter.

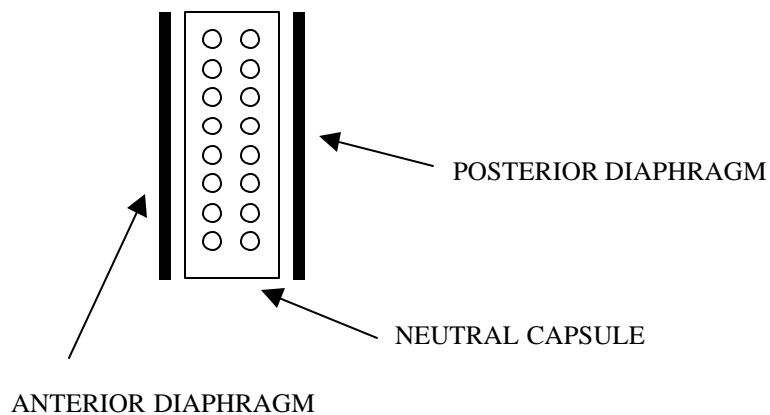


Figure 11: Diagram of a pressure-gradient condenser microphone. Typically, the interior capsule is held at ground, and the variations of capacitance between the diaphragms and the capsule generate a voltage. To obtain directional characteristics, the voltages of the anterior and posterior diaphragms may be weighted and subtracted. This produces the familiar directional patterns, such as cardioid, hypercardioid, and so on.

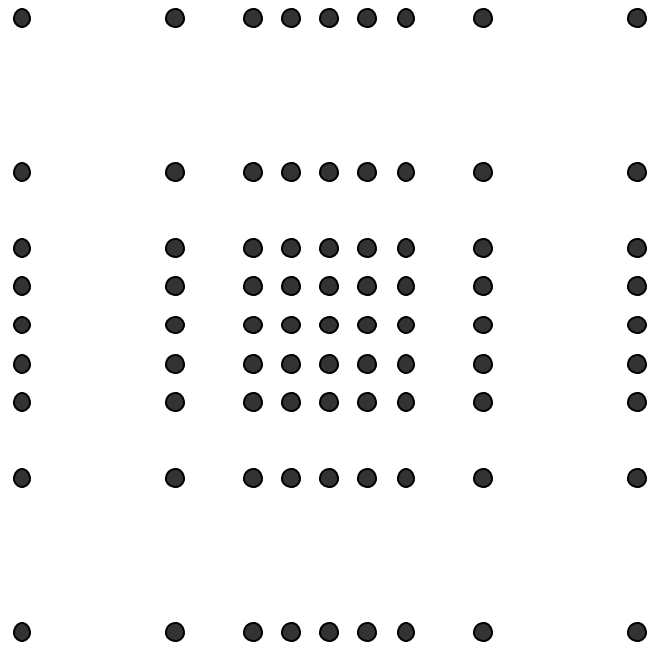


Figure 12: Regular 2-dimensional array with equal resolution in horizontal and vertical directions.

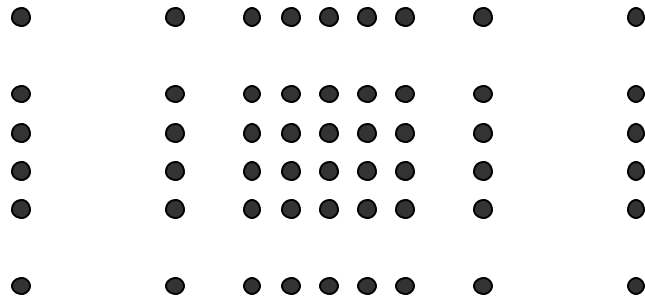


Figure 13: 2-dimensional microphone array showing unequal resolution in vertical and horizontal directions.

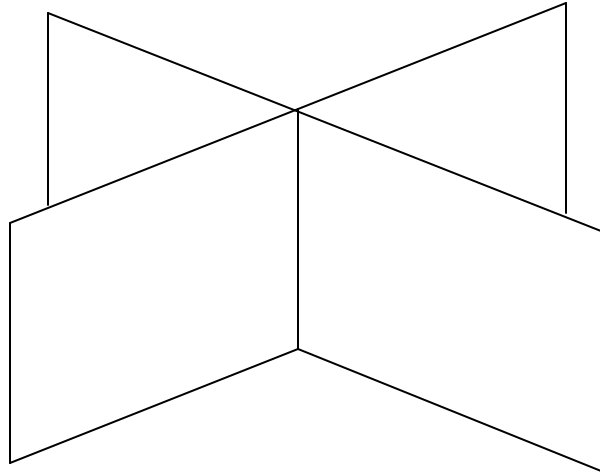


Figure 14: Two 2-dimensional arrays placed at right angles. Since each array is capable of steering across an angle of 90° in the forward direction and 90° in the backward direction, two arrays placed at right angles can cover all directions.

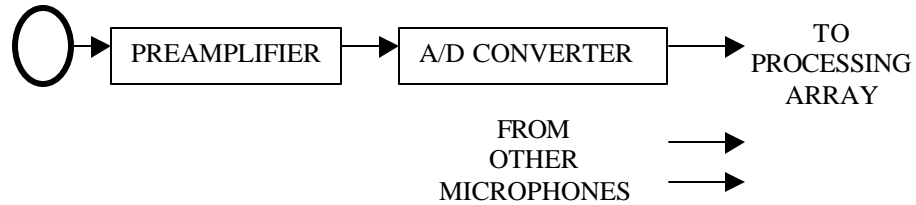


Figure 15: A direct implementation of the array has a wire from each microphone in the array to the required preprocessing (microphone preamplifier and A/D converter) and subsequent processing (Figure 9).

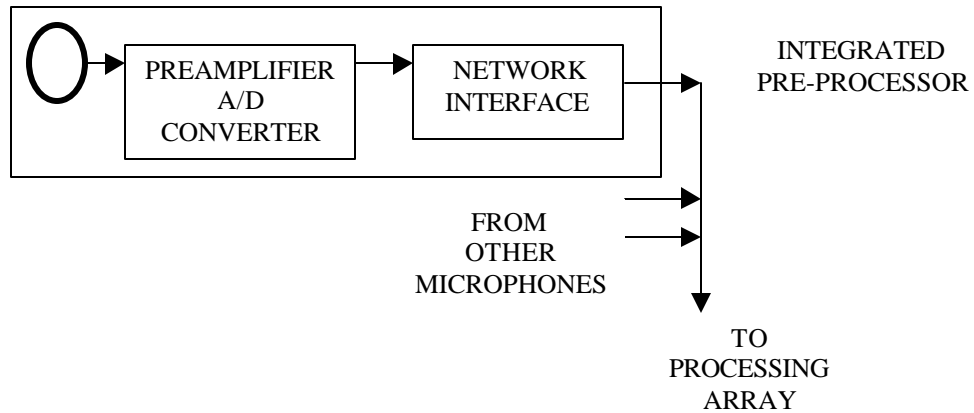


Figure 16: Microphone capsule with integrated pre-processing. In this configuration, miniaturized preamplifier and A/D stages are integrated with some kind of multiplexing (network) interface. This interface can use frequency or time-domain multiplexing, or may interface to a general-purpose network, such as Ethernet [**ref].

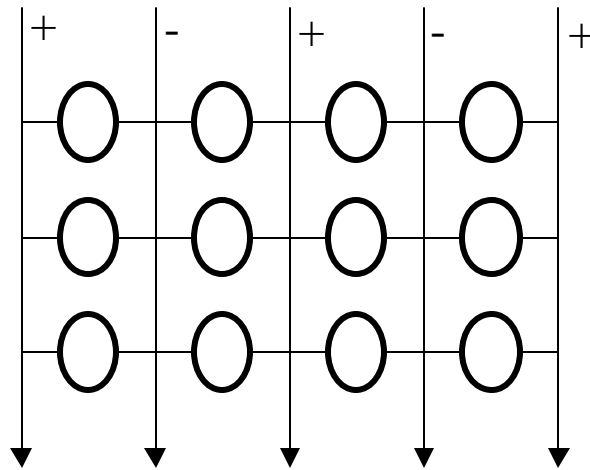


Figure 17: Each oval above represents a complete transducer, preprocessor, and network interface as shown in Figure 16. This figure shows how the array may be powered by a vertical array of alternating positive and negative supplies. One rail (*e.g.* the positive wires) may also serve as the medium for the network (or additional wires may be used for the network interface).