Preliminary Ideas for a Self-Calibrating Speaker/Microphone Array

(CONFIDENTIAL)

Julius O. Smith III
Jonathan Abel
Marc Lebrun
The CCRMA LLAMA Team Who else? Co-authors invited!

Center for Computer Research in Music and Acoustics (CCRMA)
Department of Music, Stanford University
Stanford, California 94305

January 3, 2006

Abstract

The problem of a self-calibrating speaker and microphone array is considered. Each speaker in the array is assumed to have local processing power and the ability to communicate acoustically with all other speakers and microphones. This paper describes a preliminary high-level system design, including a proposed algorithm for computing array geometry from measured inter-speaker and speaker-to-microphone distances.

Contents

1 Introduction

An audio monitor (loudspeaker) with a digital audio input must have an embedded processor for converting the digital audio to analog form. This processor can also perform other functions, such as audio equalization. If it can also communicate with other monitors acoustically, it is possible for a group of monitors to spontaneously form a phased array. Such arrays could be useful for concert sound reinforcement, surround-sound systems, underwater acoustic arrays, and other applications.

This paper is concerned with the problem of *self-calibration* for an arbitrarily arranged array of audio monitors and one or more microphones. An example configuration is shown in Fig. 1. In this imagined implementation, each monitor is powered and accepts a digital audio channel via wireless antenna (e.g., 802.11a). The console device contains a built-in microphone which is normally placed in the center of the desired listening area (the "sweet spot"). It also has additional microphone inputs, enabling array calibration over a wider spatial range. When the reset button is pressed, the system self-calibrates to form an optimal phased array. This paper describes an initial system design and algorithm for computing array geometry from interspeaker and speaker-to-microphone distance measurements.

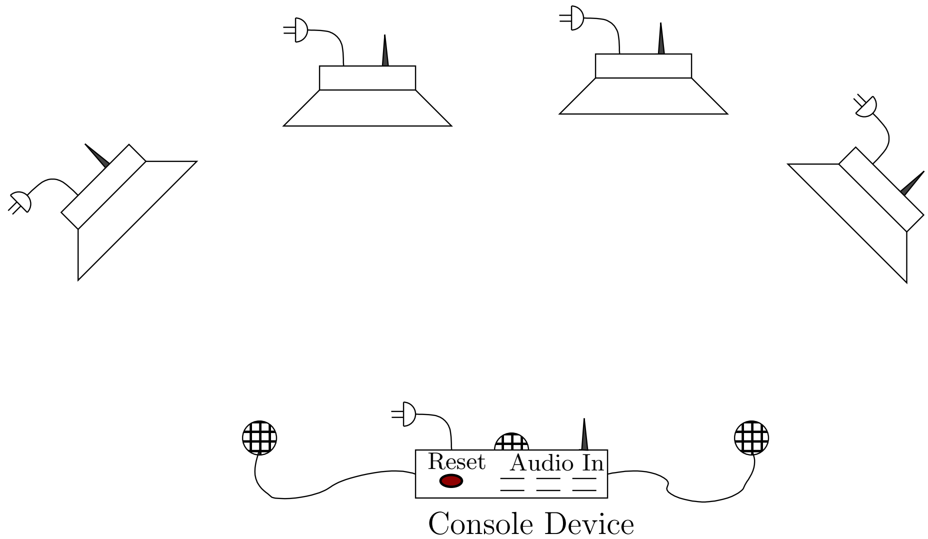

Figure 1: Schematic diagram of four wireless powered monitors, two optional external microphones, and a console device with built-in microphone.

2 Interspeaker Distance Measurement

We measure interspeaker distance $\hat{r}_{ij}$ using the same basic principle as the satellite-based global positioning system (GPS) [1]. GPS consists of 24 satellites orbiting the earth having precisely known positions and calibrated atomic clocks. A GPS receiver can use broadcasts from three or more GPS satellites to compute is lattitude and longitude accurate to within a few feet [8].

GPS radio satellites broadcast pseudo-random noise (PRN) packets on a common carrier which can be decoded by GPS receivers to determine their time of origin and satellite of origin. The receiver also measures precisely the time-of-arrival of each received packet and computes the satellite-receiver distance from the time difference of arrival. This distance then determines a sphere of "feasible locations" about the satellite broadcast position, which is known. Two satellites determine two intersecting spheres of feasible locations, resulting in a *circle* of feasible locations. Information from a third satellite reduces the set of feasible locations to two points, one of which is typically in space. Optionally, additional satellite signals may be used to further reduce the estimated location error.

For the acoustic speaker array, we perform a similar sequence of operations on a much smaller scale:

1. A *reset* protocol is sent *electronically* to all speakers in the array in order to establish "time 0" in all speakers and assign speaker numbers. The reset sequence is a special digital audio signal recognized and intercepted by the microprocessor in each speaker, resulting in no sound from the speaker during reset. A fresh reset is necessary whenever a speaker (or microphone) is moved, removed from, or added to the array. Alternatively, periodic resets can occur automatically while the audio program is silent.

2. The speakers take turns broadcasting pseudo-random noise (PRN) *probe packets* in their respective time slices. These are low-level signals which are not objectionable to an audience. The time-slice durations may be lengthened to obtain quieter probe packets.

3. Each speaker records the probe packets of its neighbors by means of a CODEC attached to its speaker terminals. By reciprocity, every speaker can also act as a microphone.[1]

4. Microphones in the array are routed to a console device which records and processes the probe packets from each speaker for each microphone. In the simplest configuration, only one microphone is present, and it defines the "sweet spot center" for the array. Since at least one microphone is necessary to determine the direction of the audience from the speaker array, an inexpensive mic is built into the console device.[2]

5. Speaker j measures the time-of-arrival (TOA) for the probe-packet broadcast by speaker i, for all $i \neq j$, and uses the TOA delay relative to the known time-slice origin to form an estimate of the distance from speaker i to speaker j, which we will call $\hat{r}_{ij}$, $i, j = 1, \ldots, N_s$.

6. The interspeaker distances $\hat{r}_{ij}$, $j = 1, \ldots, N$ are transmitted[3] to the console unit from speaker i during its time slice window, and the console processor assembles the matrix $\hat{\mathbf{R}}_{\mathbf{s}}[i, j] = \hat{r}_{ij}$. This matrix is close to symmetric with a zero diagonal.

7. For each microphone, the console measures the TOA for the PRN packet leaving speaker i, $i = 1, \ldots, N_s$, and arriving at microphone j, $j = 1, \ldots, N_m$. From these measured time delays, the console computes $\hat{r}_{ij}$, the distance from speaker i to microphone j, which we also

[1] Side research topic: monitor the "back emf" at the speaker terminals as a means of setting up a closed loop feedback control on the speaker. Presumably, this can be used to compensate various speaker distortions.

[2] Note that we can actually estimate orientation as well as distance. Once interspeaker distances have been estimated, we know what amplitude level to expect at each speaker (at each frequency), assuming no obstructions. The difference between observed and expected spectra can be used to estimate relative speaker rotation angle. Therefore, an array can potentially self-configure even with no microphones at all in the audience direction.

[3] Transmissions from the speakers to the consoles may be acoustic, like the probe packets, or a bidirectional digital audio interface may be used (Firewire, USB, ethernet, etc.).

denote $\hat{r}_{ij}$, $i = 1, \ldots, N_s$, $j = 1, \ldots, N_m$. In this case, the console sets $\hat{r}_{ji} = \hat{r}_{ij}$ in the distance matrix $\hat{\mathbf{R}}_{\mathbf{s}}$ to account for the fact that the microphones have no associated speaker.[4]

8. The complete $N_s \times N_m$ multichannel transfer function (N_m microphone inputs, N_s speaker outputs) is estimated by the console device for equalization purposes. (Use of this data is beyond the scope of the present paper.)

3 Geometry Estimation

This section describes an *equation-error* algorithm [4, 7] for estimating the array geometry from noisy measurements $\hat{r}_{ij}$ of the interspeaker and speaker-to-microphone distances.

3.1 Problem Formulation

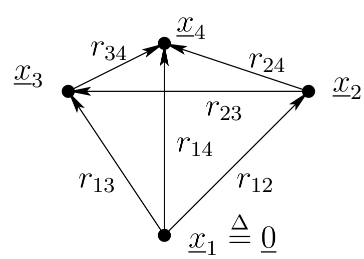

Figure 2: Example problem geometry for $N = 4$.

With reference to Fig. 2, we define the following notation:

$$
\begin{aligned}
\mathbf{R}^3 &\stackrel{\Delta}{=} \text{the set of all three-dimensional vectors } \underline{x} = [x_1, x_2, x_3]^T \text{ with real coordinates } x_i \in \mathbf{R} \\
N_s &\stackrel{\Delta}{=} \text{number of speakers} \\
N_m &\stackrel{\Delta}{=} \text{number of microphones} \\
N &\stackrel{\Delta}{=} N_s + N_m = \text{total number of nodes in the array geometry} \\
\underline{x}_i &\stackrel{\Delta}{=} [x_{i1}, x_{i2}, x_{i3}]^T = \text{vector in } \mathbf{R}^3 \text{ locating the } i\text{th speaker in 3D space}, i = 1, \ldots, N_s \\
r_i &\stackrel{\Delta}{=} \|\,\underline{x}_i\,\| = \text{length (norm) of } \underline{x}_i = \text{distance of } \underline{x}_i \text{ from the origin } \underline{0} \\
r_{ij} &\stackrel{\Delta}{=} \|\,\underline{x}_i - \underline{x}_j\,\| = \sqrt{(\underline{x}_i - \underline{x}_j)^T(\underline{x}_i - \underline{x}_j)} = \text{distance between } \underline{x}_i \text{ and } \underline{x}_j
\end{aligned}
$$

The problem is to find $\underline{x}_i$ for $i = 1, \ldots, N$ given measurements $\hat{r}_{ij}$ of r_{ij}, to within an arbitrary translation and rotation in $\mathbf{R}^3$.

[4]We could have a cheap little speaker in the console for complete symmetry in the single-microphone case. By reciprocity, we could turn all the microphones into little speakers, but I'm not trying this with my microphones (◡̈).

3.2 Problem Solution

The squared distance from speaker i to speaker j can be written as

$$\begin{aligned} r_{ij}^2 &\triangleq \left\| \underline{x}_i - \underline{x}_j \right\|^2 \\ &= \left\langle \underline{x}_i - \underline{x}_j, \underline{x}_i - \underline{x}_j \right\rangle \\ &= \left\langle \underline{x}_i, \underline{x}_i \right\rangle - \left\langle \underline{x}_i, \underline{x}_j \right\rangle - \left\langle \underline{x}_j, \underline{x}_i \right\rangle + \left\langle \underline{x}_j, \underline{x}_j \right\rangle \\ &= \left\| \underline{x}_i \right\|^2 - 2 \left\langle \underline{x}_i, \underline{x}_j \right\rangle + \left\| \underline{x}_j \right\|^2 \\ &= r_i^2 + r_j^2 - 2\underline{x}_i^T \underline{x}_j \end{aligned}$$

Thus,

$$r_{ij}^2 = r_i^2 + r_j^2 - 2\underline{x}_i^T \underline{x}_j, \quad i, j = 1, \ldots, N. \tag{1}$$

We have $r_{ii} = 0$ for all i, and $r_{ij} = r_{ji}$ for all i and j from 1 to N. Since the origin is arbitrary, we may assign $\underline{x}_1 = \underline{0}$, so that $r_{i1} = r_{1i} = r_i$. Since there is no information specifying orientation in 3D space, we expect a solution that is invariant with respect to rotation of the $\underline{x}_i$ about $\underline{x}_1 \triangleq \underline{0}$. In practice, it is convenient to have $\underline{x}_1$ be the listening position, or "sweet-spot center," which requires a microphone at that location in order for it to participate in the array calibration.

Define the matrix of squared speaker and microphone positions as

$$\mathbf{X} \triangleq \begin{bmatrix} \underline{x}_1^T \\ \vdots \\ \underline{x}_N^T \end{bmatrix}, \tag{2}$$

and the vector of squared distances from the origin as

$$\underline{r}_s \triangleq \begin{bmatrix} r_1^2 \\ \vdots \\ r_N^2 \end{bmatrix}.$$

Finally, define the matrix of squared inter-node[5] distances as

$$\mathbf{R_s} \triangleq \begin{bmatrix} r_{11}^2 & \cdots & r_{1N}^2 \\ \vdots & \cdots & \vdots \\ r_{N1}^2 & \cdots & r_{NN}^2 \end{bmatrix}.$$

For microphones, we measure the distance in only direction, from the speaker to the microphone, so given r_{ij} as this distance, we simply set $r_{ji} = r_{ij}$.

Now Eq. (1) can be written as

$$\mathbf{X_s} \triangleq \mathbf{X}\mathbf{X}^T = \frac{1}{2} \left[\underline{1} \cdot \underline{r}_s^T + \underline{r}_s \cdot \underline{1}^T - \mathbf{R_s} \right], \tag{3}$$

where $\underline{1} \triangleq [1, \ldots, 1]^T$. Note that all terms on the right-hand side are specified by the inter-node distances. On the left-hand side, we have a symmetric matrix which can be factored to obtain the

[5] A node is defined as either a speaker or a microphone.

relative node positions, as discussed in §3.4. The answer we seek, $\mathbf{X}$, is the *matrix square root* of the right-hand side, which is not unique.

From this expression, it is easy to see that the node positions $\mathbf{X}$ may be freely rotated about $\underline{0}$. To see this, let $\mathbf{Q}$ denote an arbitrary rotation matrix,[6] and let

$$\mathbf{X_Q} \stackrel{\Delta}{=} \mathbf{X}\mathbf{Q}^T = \begin{bmatrix} \underline{x}_1^T \mathbf{Q}^T \\ \vdots \\ \underline{x}_N^T \mathbf{Q}^T \end{bmatrix},$$

denote the set of node coordinates rotated about $\underline{0}$ by $\mathbf{Q}$. Then we have

$$\mathbf{X_Q}\mathbf{X_Q}^T = \mathbf{X}\mathbf{Q}^T(\mathbf{X}\mathbf{Q}^T)^T = \mathbf{X}\mathbf{Q}^T\mathbf{Q}\mathbf{X}^T = \mathbf{X}\mathbf{X}^T,$$

and thus Eq. (3) is shown to be invariant with respect to rotations, as it must be.

3.3 Effect of Measurement Errors

In practice, the inter-node distances r_{ij} are *measured* with some amount of error. Let $\hat{r}_{ij}$ denote the measured value of the distance from node i to node j, end let $\epsilon_{ij} \stackrel{\Delta}{=} r_{ij} - \hat{r}_{ij}$ denote the corresponding measurement error. I.e.,

$$r_{ij} \stackrel{\Delta}{=} \hat{r}_{ij} + \epsilon_{ij}.$$

Then from Eq. (1), we have

$$\begin{aligned} 2\underline{x}_i^T \underline{x}_j &= r_i^2 + r_j^2 - r_{ij}^2 \\ &= (\hat{r}_i + \epsilon_i)^2 + (\hat{r}_j + \epsilon_j)^2 - (\hat{r}_{ij} + \epsilon_{ij})^2 \\ &= \hat{r}_i^2 + 2\hat{r}_i\epsilon_i + \epsilon_i^2 + \hat{r}_j^2 + 2\hat{r}_j\epsilon_j + \epsilon_j^2 - \hat{r}_{ij}^2 - 2\hat{r}_{ij}\epsilon_{ij} - \epsilon_{ij}^2 \\ &\approx \hat{r}_i^2 + 2\hat{r}_i\epsilon_i + \hat{r}_j^2 + 2\hat{r}_j\epsilon_j - \hat{r}_{ij}^2 - 2\hat{r}_{ij}\epsilon_{ij} \\ &= 2\underline{\hat{x}}_i^T \underline{\hat{x}}_j + 2\hat{r}_i\epsilon_i + 2\hat{r}_j\epsilon_j - 2\hat{r}_{ij}\epsilon_{ij} \\ &\stackrel{\Delta}{=} 2\underline{\hat{x}}_i^T \underline{\hat{x}}_j + 2\eta_{ij}, \quad i,j = 1,\ldots,N. \end{aligned} \tag{4}$$

where η_{ij} is called the *equation error* for this problem:

$$\eta_{ij} \stackrel{\Delta}{=} \underline{x}_i^T \underline{x}_j - \underline{\hat{x}}_i^T \underline{\hat{x}}_j = \hat{r}_i\epsilon_i + \hat{r}_j\epsilon_j - \hat{r}_{ij}\epsilon_{ij}. \tag{5}$$

For the matrix formulation Eq. (3), we may define the *equation error matrix* by

$$\begin{aligned} \mathbf{E} &\stackrel{\Delta}{=} \mathbf{X}\mathbf{X}^T - \frac{1}{2}\left[\underline{1}\cdot\underline{\hat{r}}_s^T + \underline{\hat{r}}_s \cdot \underline{1}^T - \mathbf{\hat{R}_s}\right], \\ &\stackrel{\Delta}{=} \mathbf{X_s} - \mathbf{\hat{X}_s} \end{aligned} \tag{6}$$

In a typical geometry, such as shown in Fig. 2, the listening position is far away compared with the interspeaker distances, and distance from the listening position to each speaker is approximately the same. If the measurement errors are identically distributed, then in this case we have

$$\eta_{ij} \approx \hat{r}_{i \text{ or } j}(\epsilon_i + \epsilon_j) \stackrel{\Delta}{=} 2\hat{r}_i\epsilon_i'. \approx r_i\sqrt{2}\epsilon_i$$

[6]Any norm-preserving rotation about $\underline{0}$ in 3D space can be represented by a 3×3 orthogonal matrix $\mathbf{Q}$, i.e., $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$ [5].

and the measurement error ϵ_{ij} is approximately a scale factor times the equation error η_{ij}. In particular, for speakers arranged along a sphere about the listening position, the equation error and inter-node measurement error are essentially equivalent.

3.4 Equation-Error Minimization

From Eq. (6), we have

$$\mathbf{X_s} \stackrel{\Delta}{=} \mathbf{X}\mathbf{X}^T = \hat{\mathbf{X}}_\mathbf{s} + \mathbf{E} \stackrel{\Delta}{=} \frac{1}{2}\left[\underline{1}\cdot\underline{\hat{r}}_s^T + \underline{\hat{r}}_s\cdot\underline{1}^T - \hat{\mathbf{R}}_\mathbf{s}\right] + \mathbf{E}$$

where $\mathbf{E} = [\eta_{ij}]$ is the equation error matrix. It is straightforward to find $\underline{\hat{x}}_i \in \mathbf{R}^3$, $i = 1,\ldots,N$, such that the *Frobenius norm* of the equation error

$$\eta_{ij} = \underline{x}_i^T\underline{x}_j - \underline{\hat{x}}_i^T\underline{\hat{x}}_j$$

is minimized.

The *Frobenius norm* (or *F-norm*) of a matrix $\mathbf{A}$ is defined as

$$\|\,\mathbf{A}\,\|_F \stackrel{\Delta}{=} \sqrt{\sum_{i=1}^N\sum_{j=1}^N |\mathbf{A}[i,j]|^2}$$

It is well known [2] that the square of the Frobenius norm of $\mathbf{A}$ equals the sum of *singular values* of $\mathbf{A}$:

$$\|\,\mathbf{A}\,\|_F^2 = \sum_{i=1}^N \sigma_i^2$$

where the singular values are obtained from the *singular value decomposition* (SVD) [2]:

$$\mathbf{U}^T\mathbf{A}\mathbf{V} = \text{diag}\{\sigma_1,\ldots,\sigma_N\} \stackrel{\Delta}{=} \Sigma, \quad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_N, \tag{7}$$

where $\mathbf{U}$ and $\mathbf{V}$ are $N \times N$ orthogonal matrices. If $\mathbf{A}$ is symmetric, then $\mathbf{U} = \mathbf{V}$. The norm result follows from the orthogonality of $\mathbf{U}$ and $\mathbf{V}$ and from rewriting Eq. (7) as

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T = \sigma_1\underline{u}_1\underline{v}_1^T + \cdots + \sigma_N\underline{u}_N\underline{v}_N^T. \tag{8}$$

where $\underline{u}_i$ denotes the ith column of $\mathbf{U}$, and $\underline{v}_i$ denotes the ith column of $\mathbf{V}$.

From Eq. (2), we see that the matrix $\mathbf{X}$ is $N \times 3$. Therefore, the $N \times N$ matrix $\mathbf{X_s} = \mathbf{X}\mathbf{X}^T$ appearing in Eq. (3) is only rank 3, and its squared Frobenius norm is

$$\|\,\mathbf{X_s}\,\|_F^2 = \sigma_1^2 + \sigma_2^2 + \sigma_3^2.$$

However, due to measurement errors ϵ_{ij}, the matrix

$$\hat{\mathbf{X}}_\mathbf{s} \stackrel{\Delta}{=} \frac{1}{2}\left[\underline{1}\cdot\underline{\hat{r}}_s^T + \underline{\hat{r}}_s\cdot\underline{1}^T - \hat{\mathbf{R}}_\mathbf{s}\right]$$

will be full rank, in general. However, it suggests the following algorithm:

1. Perform the *symmetric* singular value decomposition of
$$\hat{\mathbf{X}}_{\mathbf{s}} \triangleq \frac{1}{2}\left[\underline{1}\cdot\underline{\hat{r}}_s^T + \underline{\hat{r}}_s\cdot\underline{1}^T - \frac{1}{2}(\hat{\mathbf{R}}_{\mathbf{s}} + \hat{\mathbf{R}}_{\mathbf{s}}^T)\right]$$
(we use $\frac{1}{2}(\mathbf{R}_{\mathbf{s}} + {\mathbf{R}_{\mathbf{s}}}^T)$ in place of $\mathbf{R}_{\mathbf{s}}$ to make $\hat{\mathbf{X}}_{\mathbf{s}}$ symmetric, thus averaging $\hat{r}_{ij}$ and $\hat{r}_{ji}$) to obtain
$$\hat{\mathbf{X}}_{\mathbf{s}} = \hat{\mathbf{U}}\mathbf{D}\hat{\mathbf{U}}^T.$$
where $\hat{\mathbf{U}} = [\underline{\hat{u}}_1, \ldots, \underline{\hat{u}}_N]$ is an orthogonal $N \times N$ matrix, and $\mathbf{D} = \text{diag}\{\sigma_1, \ldots, \sigma_N\}$ is the diagonal matrix of singular values of $\hat{\mathbf{X}}_{\mathbf{s}}$.

2. Take the solution $\hat{\mathbf{X}}$ to be the matrix square root of the rank three reduction of $\hat{\mathbf{X}}_{\mathbf{s}}$, i.e.,
$$\hat{\mathbf{X}} \triangleq [\underline{\hat{u}}_1, \underline{\hat{u}}_2, \underline{\hat{u}}_3]\cdot\text{diag}\{\sqrt{\sigma_1}, \sqrt{\sigma_2}, \sqrt{\sigma_3}\} = \sqrt{\sigma_1}\underline{\hat{u}}_1 + \sqrt{\sigma_2}\underline{\hat{u}}_2 + \sqrt{\sigma_3}\underline{\hat{u}}_3$$

The solution $\hat{\mathbf{X}}$ has the property that its "square," $\hat{\mathbf{X}}_{\mathbf{s}} = \hat{\mathbf{X}}\hat{\mathbf{X}}^T$ minimizes the Frobenius norm of the equation error $\mathbf{E}$ over the set of all rank-three approximations. Since only rank-three approximations are *feasible solutions* for this problem, we may say that $\hat{\mathbf{X}}$ is the optimum feasible solution which minimizes the Frobenius norm of the equation error $\mathbf{E}$.

More generally, $\hat{\mathbf{X}}\mathbf{Q}^T$ is an equivalent solution for any 3×3 orthogonal matrix $\mathbf{Q}$, In particular, $\mathbf{Q}$ can be chosen to place the speaker locations in more "natural" positions for visualization. For example, a plane can be fit to the set of node locations by means of 2D linear regression, and the rotation $\mathbf{Q}$ can be used to zero the 3rd ("z") coordinate of this plane. Moreover, a line can be found in this plane which passes through $\underline{0}$ (the listening position) and bisects the set of estimated node locations projected onto the plane, and the rotation can place this line along the "y" direction ($\underline{x} = [0, x_2, 0]^T$). Such a "standardized rotation" for the node locations facilitates visual display.

3.5 Fisher Information and the Cramèr-Rao Lower Bound

A valuable tool for evaluating estimation algorithms is the Cramèr-Rao Bound (CRB) [3]. Under general conditions, the CRB gives a lower bound for the variance of any unbiased estimator.

(Jonathan has some results on this which have yet to be written up.)

3.6 Weighted Estimation

From §3.4, the most straightforward error to minimize is the equation error $\mathbf{E} = [\eta_{ij}]$ defined in Eq. (5), which is close to minimizing measurement error for typical array configurations (e.g., spherical or planar). However, in practice, we are more likely to want to minimize ...

(Anyone want to work on this?)

4 Simulation Results

(Scott Wilson is working on this.)

5 Conclusions

A Appendix: But Can It Blow Out a Candle?

This appendix presents an approximate analysis of the sound pressure level available from a uniform planar array of identical small drivers which have limited linear excursion.

Let us first consider a plane-wave source. A pressure wave of amplitude p_0 emanates from the plane-source when its velocity is

$$v_0 = \frac{p_0}{R} \tag{9}$$

where R is the wave impedance of air, given at standard conditions by

$$R = \rho c = 42.7\ \frac{\text{g}}{\text{cm}^2 \cdot \text{sec}} = 427\ \frac{\text{kg}}{\text{m}^2 \cdot \text{sec}}$$

where ρ is air density in mass per unit volume, and c is sound speed in distance per second.

In the most sensitive frequency range for human hearing, the threshold of hearing is approximately 0 dB SPL ($p_{\text{ref}} \stackrel{\Delta}{=} 20$ uPa) [6]. Thus, to create a plane wave at 120 dB SPL (the "threshold of feeling"), for example, the peak plane-wave pressure must be 20 Pascals (Nt/m^2). The corresponding velocity is then, from Eq. (9),

$$v_0 = \frac{p_0}{R} = \frac{20}{427} = 0.047 \text{ m/sec}$$

Let the velocity of the plane-source be sinusoidal:

$$v(t) = v_0 \cos(\omega t)$$

Then the displacement is given by

$$x(t) = \int_0^t v(\tau) d\tau = \frac{v_0}{\omega} \sin(\omega t).$$

Therefore, the maximum displacement of the sinusoidal plane-source at frequency ω radians per second is

$$x_{\text{max}} = \frac{v_0}{\omega}$$

Thus, at 120 dB SPL, the maximum displacement of the plane source at 20 Hz is

$$x_{\text{max}} = \frac{p_0}{\omega R} = \frac{20}{2\pi 20 \cdot 427} = 0.00037 \text{ m} = 0.37 \text{ mm.} \tag{10}$$

A 0.4 mm source displacement sounds doable, but remember that this is for a true plane source, not an array of small drivers.

Working it the other way, given the peak displacement of the driver, we can calculate the maximum dB SPL generated:

$$p_{\text{max}} = x_{\text{max}} \omega R$$

Example: if a plane-source can move plus or minus 0.1 mm, the dB SPL level generated at 20 Hz is

$$20 \log_{10} (x_{\text{max}} \omega R / p_{\text{ref}}) = 20 \log_{10}(0.0001 \cdot 2\pi 20 \cdot 427/20 \times 10^{-6}) = 109 \text{ dB}$$

Since R and p_{ref} are constants, and $\omega = 2\pi f$, we can simplify the general formula to

$$\text{dB SPL} = 20\log_{10}(x_{\mathrm{max}} \cdot f) + 102.55$$

where x_{max} is in *millimeters* and f is in Hz.

Consider now a planar array of small drivers, e.g., 1" drivers spaced every 10" along x and y. To approximate a plane source, the surface integral of the normal velocity must be the same. (This is optimistic since it ignores diffraction/mode-conversion losses, but it should be pretty good at wavelengths much larger than the array spacing.) If the drivers are approximated as square, they must inject $(10/1)^2 = 100$ times the velocity of the equivalent plane source. For a 1"-diameter circular driver, the figure is $100/(\pi * (1/2)^2) = 127$ times. Since peak displacement is proportional to velocity, it must also be 127 times greater. Equivalently, the peak displacement of the driver can be divided by 127 and used as the equivalent displacement for a plane source. For example, if the 1" driver can move ± 1 mm, the maximum SPL we expect from the array is approximately

$$20\log_{10}((1/127) \cdot 20) + 103 = 87 \text{ dB}$$

at 20Hz. At 40 Hz we hope to get

$$20\log_{10}((1/127) \cdot 40) + 103 = 93 \text{ dB}$$

etc. (6dB extra for each doubling of the bass cut-off).

In summary, the SPL (in dB) generated by an array of drivers is approximately

$$20\log_{10}\left(x_{\mathrm{max}}\frac{A_d}{A_c}f_{\mathrm{min}}\right) + 102$$

where

$$\begin{aligned}
x_{\mathrm{max}} &= \text{maximum positive driver displacement in mm (half of peak-to-peak)} \\
A_d &= \text{area of driver} \\
A_c &= \text{array cell area} \\
f_{\mathrm{min}} &= \text{lowest frequency supported by the array}
\end{aligned}$$

Example: An array of 1" diameter speakers that can vibrate ± 1 mm using one-foot spacing delivers

$$\begin{aligned}
20\log_{10}\left(1 \cdot \frac{\pi(1/2)^2}{12^2} \cdot 20\right) + 102 &= \text{83 dB at 20 Hz and above} \\
20\log_{10}\left(1 \cdot \frac{\pi(1/2)^2}{12^2} \cdot 40\right) + 102 &= \text{89 dB at 40 Hz and above}
\end{aligned}$$

etc. Note that doubling the speaker excursion is equivalent to doubling the minimum frequency required of from the array.

B Appendix: Simplified System Design Using Off-the-Shelf Components

For prototyping purposes, a system can be constructed using multiple sound cards (e.g., two channels audio input and output per card), existing audio monitors (e.g., "multimedia speakers"), with a microphone placed near each monitor (e.g., on top of it). The host PC can perform the functions of the console device and monitor processors.

One issue with such a system is the precise *timing* of all audio channels, both input and output:

1. Precise timestamps are needed on microphone input data during system reset.
2. Precise timestamps are needed on each speaker output during system reset. If necessary, these can be measured using each speaker's own microphone.
3. In normal audio playback mode, all output channels must be precisely synchronized.

Under Linux, it is straightforward to modify the sound-card drivers to approach these goals. However, a nicer (and more realistic) prototyping environment would consist of sound cards having a programmable DSP chip and a high-resolution on-board clock.

Fernando says it is fairly easy nowadays to set up a Linux based system with up to 24 synchronized input/output channels (an RME Hammerfall card with three external banks of A/D and D/A converters would do it). 48 channels would be possible with two externally synchronized cards, etc.

Fernando also says the Delta 1010 cards we have on several workstations at CCRMA can do the equivalent with 8 A/D and D/A channels.

References

[1] E. D. K. (Editor), *Understanding GPS: Principles and Applications*, Artech House, 1996.

[2] G. H. Golub and C. F. Van Loan, *Matrix Computations, 2nd Edition*, Baltimore: The Johns Hopkins University Press, 1989.

[3] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*, Englewood Cliffs, NJ: Prentice-Hall, Inc., 1993.

[4] L. Ljung and T. L. Soderstrom, *Theory and Practice of Recursive Identification*, Cambridge, MA: MIT Press, 1983.

[5] B. Noble, *Applied Linear Algebra*, Englewood Cliffs, NJ: Prentice-Hall, Inc., 1969.

[6] A. D. Pierce, *Acoustics*, American Institute of Physics, for the Acoustical Society of America, 1989, `http: //asa.aip.org/publications.html`.

[7] J. O. Smith and J. S. Abel, "Closed-form least-squares location estimation from range-difference measurements," *IEEE Transactions on Acoustics, Speech, Signal Processing*, vol. 35, pp. 1661–1669, Dec. 1987.

[8] Trimble, *All About GPS*, `http: //www.trimble.com/gps/`, 2002.