HMA-III: A SELF-CALIBRATING WIRELESS MICROPHONE ARRAY SYSTEM

Harvey Silverman, Rahul Shakya, Ken Silverman

LEMS
School of Engineering
Box D, Brown University, Providence, RI 02912
hfs@lems.brown.edu

ABSTRACT

Our past research has focused on wired microphone arrays having a large number of elements and wide aperture. The wiring for these large arrays is tedious and only a tree structure with intermediate multiplexing of signals works well. A wireless array alleviates this immense wiring issue, albeit other difficulties are introduced including interferences in the wireless transmissions, synchronization, the need for automatic self-calibration, and module power, size and mounting. This paper introduces our first successful wireless array system, HMA-III. It discusses the desired attributes for such an array and the tradeoffs made to assemble a first working system. Results for the automatic self-calibration are presented.

I. INTRODUCTION

Research on arrays of microphones that could be supported by inexpensive real-time digital signal processing really began in earnest in the middle 1980's [1], [2], [3], [4]. About that time-frame, the large, essentially analog, array of some 400 microphones was installed in the Bell Laboratories auditorium [5]. Research continued for arrays of 16 microphones or fewer, but in the mid 1990's we at Brown University joined with the CAIP Center of Rutgers University to build the 512-microphone, 90 DSP Huge Microphone Array (HMA) and completed and installed it in 1998 at Brown University [6], [7], [8]. Its wiring was an intensive job, requiring a tree structure with concentrator node modules for groups of 16 microphones and a tree of 32 nodes using optical fiber. In 2004, we set out to build a lower-cost, improved, HMA-II, which still used the cumbersome tree wiring. However, HMA-II used 30 times faster signal processors, simpler node modules in the tree with ethernet-like 4 twisted-pair copper wiring. The hardware supported real-time DSP programming in 2006, but had no high-speed data outlet for recording to the PC. While a USB2 gateway was working intermittently in 2008 we saw that a standard four-core PC had the same processing power as the included DSP boards for a real-time 128-microphone system, so we eliminated the DSP's and used an off-the-shelf FPGA system and a home-designed board as an interface from the node modules to USB2. Since 2010 we have had a stable HMA-II system installed and running in a separate room from the HMA.

In 2010, we wanted to design and build another array, HMA-III, that did not have the tree wiring, could be easily reconfigured and used COTS parts. Some work on wireless systems has been reported; in [9] some theoretical ideas are presented and the wireless array used in [10] is a single unit used for ecology studies. HMA-III also had to be able to self-calibrate quickly, i.e., a three-dimensional coordinate system established and the coordinates of each of the microphones determined before the array can be effectively used. Based on all the accumulated experience with large arrays, our design criteria were:

1) No "tree" wiring of the array
2) Dynamic range and sampling rate of the previous arrays (24-bit A/D with 20kHz sampling).
3) A low-latency system that works in real time.
4) Accurate module synchrony guaranteeing accurate phase estimates for the DSP
5) Self-calibrating
6) Convenient, long-lasting module power
7) Use a single PC for all processing.
8) Small unobtrusive modules, *with* omnidirectional behavior.

Of course, design wishes and practicality often have to be traded-off. This has been the case in our first working system. We ran experiments and verified that current technology still uses too much power to be able to use energy harvesting of any practical sort in a normal room environment, no matter if done from heat, light or vibration. Thus we used a Li-ion rechargeable battery with about 8 hours life per charge and a built in charger as the practical satisfaction of #6 in the list above. We also abandoned item #8 above for the proof-of-concept system and traded-off size for ease of design and debugging. Also, unfortunately, the design of HMA-III began before off-the-shelf, moderately-priced, high-speed, wireless modules were generally available. Nevertheless we have met all the other design goals for HMA-III. In this paper we describe the new wireless microphone-array system and the design compromises required to actually get a working system built. Then we discuss our working self-calibration method and present results for its accuracy.

II. THE HMA-III SYSTEM

The HMA-III system design centered on making small microphone modules that wirelessly and synchronously sent uncompressed PCM-audio data from a single microphone in real-time to a standard PC. A block diagram of the module is shown in Figure 1 and one of the modules is shown with the cover off and labeled in Figure 2. For the proof-of-concept first design, we traded off size for ease of design and debugging so each module was packaged in a box that is about 3”x5”x1”.

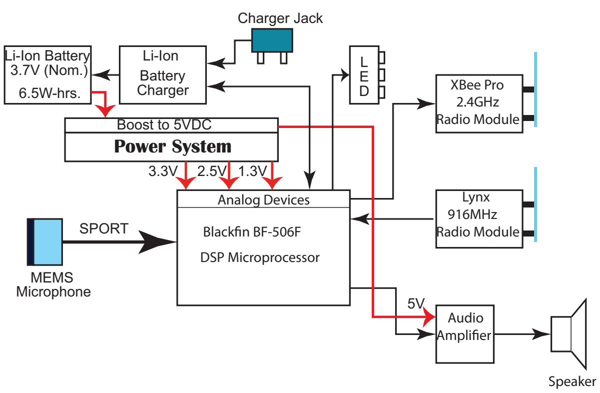

Fig. 1. Block Diagram of HMA-III Module

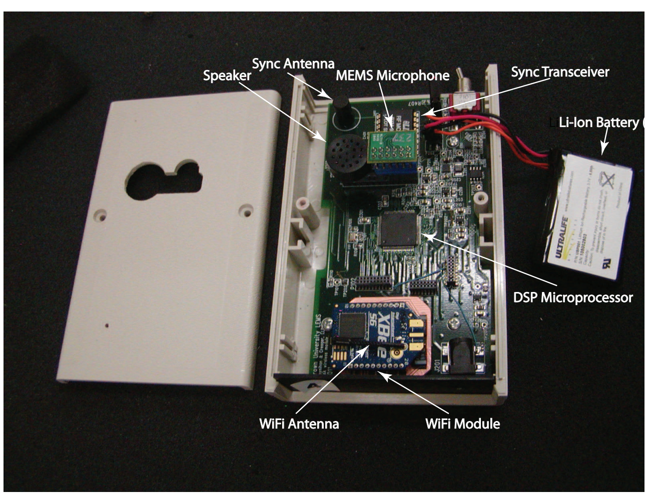

Fig. 2. Opened HMA-III Module

Microphone system: We decided to avoid the need to build a preamplifier and a traditional ADC by using an omnidirectional silicon MEMS microphone that produces a digital output, the Analog Devices ADMP421 [11]. The serial line digital output has data from a fourth-order sigma-delta modulator which is meant to be run at a very high rate. To connect this microphone directly to a microprocessor one needs to have sufficient DSP processing capability in the microprocessor to lowpass filter and decimate the oversampled serial signal. This implied the use of a low-end DSP microprocessor and we chose to use the Analog Devices ADSP504F Blackfin. This processor has more than enough DSP capability, a high-speed interface for the microphone and low, 0.5mw/MHz, power use. Maintaining the 20kHz sampling of previous systems, we used 100:1 oversampling or a rate of 2MHz. A cascaded three-stage decimation system was implemented in the processor with the first stage a fifth-order cascaded-integrated-comb(CIC) filter (decimate by 10), a half-band filter (decimate by 2), and then a final FIR filter to decimate by 5. The 20kHz output produces comparable output to a normal ADC of $\approx -3.23 + 9/2 log_2 M \approx 24 bits$ where M is the oversampling ratio (100 here). This gives sufficient dynamic range for sources both close and far away from the microphone.

Calibration System: The ideas of using camera or laser input were quickly discarded, if, for no other reason,due to the problems with directionality. In our earliest designs, we explored using ultrasonic, 40kHz transducers. These turned out to be highly directional as well and very band-limited, thus not really supporting useful chirp bandwidths. Using 40kHz ultrasonic signaling would also require either a mixing, analog front-end or a high ($> 80kHz$) sampling rate. For these reasons, we went back to normal audio, using a small speaker that could produce from $1kHz \rightarrow 8kHz$. Our design was to have the microphone and speaker as close to each other as possible, so that we could consider the error relative to a distant module, negligible. Figure 2 shows the module and the separation of the transducers.

Wireless System: A significant challenge was to correctly select a wireless module and scheme for simultaneous broadcasting from many array modules simultaneously in real-time and at 320kb/s (20kHz sampling 16 bits/sample). The only off-the-shelf transceiver modules of modest cost that were advertised for this rate were WiFi at the 2.4GHz band. It was unfortunate that our specification predated the availability of a reliable module by about two years. After trying about eight modules over that period, we found the Digi International XB24-WFWIT-001 which turned out to be quite robust, available, and easy to use. To achieve the needed low latency for real-time applications, we chose to implement the UDP protocol. Though there is a possibility of losing data packets with UDP losses were not observed in out room. Using a logarithmic detector [12], we measured the actual data rate being sent by the wireless module as just about 55Mb/s. The wireless system’s buffer dictated that the maximum packet size be just over 1kB(8kb). For 16-bit data and 20kHz sampling, a packet from each microphone would have data for 25.6ms, but at 55Mb/s, the packet is ”on the air” for just $145\mu s$. This puts the theoretical limit for the number of modules that could be used in the array to $N = 25,600/145 = 176$. However, the practical limit is probably closer about 100 since perfect time-multiplexing may be difficult to achieve.

Synchronization System: For ideal application of most array functions, it is essential that the data are aligned in time. To guarantee this, some form of synchronization among modules is essential. We investigated several mechanisms for an accurate sync system including visible light, infrared, and ultrasound. The "light" solutions suffered from the problem of being directional and ultrasound was too slow to have accuracy near a microsecond. Using the WiFi radio itself was also a problem in that the software protocols are used with WiFi and thus timing is not predictable accuracy. Thus we chose to use a very "simple" additional radio system whose timing and data structure we could design ourselves at the lowest level. We thus added a Lynx RXM-916-ES 916MHz transceiver and antenna to each module just for this purpose. In addition we had to build a base-station broadcaster. This base-station has been built into one of the module-container boxes. It is powered over USB by the base PC and contains an embedded Blackfin ADSP-BF506F processor with an accurate, crystal controlled clock, as well as a Lynx transceiver and antenna. The unit is shown in Figure 3. The processor is programmed to broadcast a current block number every 51.2ms. Testing has shown that these signals are received with a time accuracy of about one microsecond, which is more than sufficient for our system. This sync signal and the module number are used to compute an appropriate delay within the 25.6ms interval for each module to guarantee the uniqueness of its transmission time slot. We also do use some control bits and error correction, so we feel confident that this current scheme can easily work for at least 50 modules and might work with as many as 100.

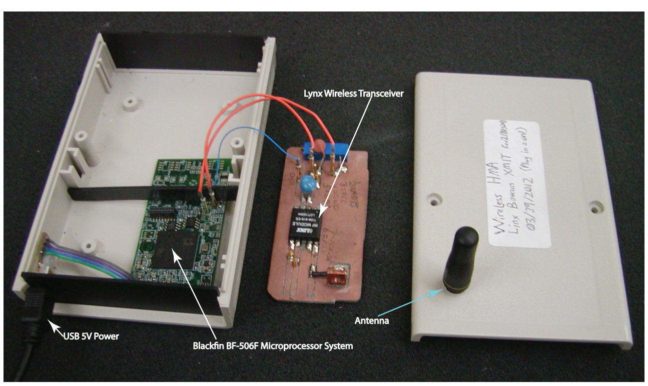

Fig. 3. Photo of the Synchronization Base Station

III. SELF-CALIBRATION

Our method uses a similar gradient-descent as used for calibrating a large-aperture wired microphone array described in [13]. However, in the wireless system, each microphone has a speaker next to it (1.8cm separation) and so no free movement of the source apparatus is needed. In [13] time differences of arrival(TDOA's) are calculated by computing the phase-transform-weighted generalized cross correlation [14] between the signals. We generate chirp signals and the computation is done through the frequency domain such that frequencies outside the chirp range are not considered. It would be natural to compare the N-1 distant signals against that from the microphone signal from the chirping module. However HMA-III is a real system and, as the chirp must be sufficiently strong at each distant receiving microphone, the local microphone, about 1.8cm away becomes overdriven. This engenders a very poor correlation output. To make both the local and distant microphone correlate effectively, we use a pre-recorded reference chirp signal that has been sent from one module's speaker to another module's microphone 1m away, call it r. Then the distance from speaker a to microphone b is $d(SPKR_a, MIC_b) = [t(MIC_b - r) - t(MIC_a - r)]C + 0.018$, where C is the nominal speed of sound in meters per second. This is repeated for all speakers ultimately, yielding M^2 speaker-to-microphone distances between all modules, given $d(SPKR_x, MIC_x) = 0.018$ for all modules. When using chirps, both the unweighted GCC and the phase-transform weighted GCC gave similar results for finding the distances.

Due to the directionality of the speaker and microphone and reflections, as shown in Figure 4, the correlation output for the chirp has a cluster of peaks for the direct wave and for the early significant reflections. It is necessary to find highest peak of the earliest cluster, rather than the strongest peak. Since, different configurations and environments of the microphone array will change the GCC output, we cannot find the first peak based on any fixed thresholds. Our method for detection of accurate TOA estimates is to (1), determine a noise threshold by obtaining a worst-case noise level from a non-chirp interval; (2), consider positive peak values of the normalized GCC over this threshold; (3) select the earliest bundle and consider its highest peak as the TOA. We have found this technique to be very robust for various configurations of the microphone array in our room.

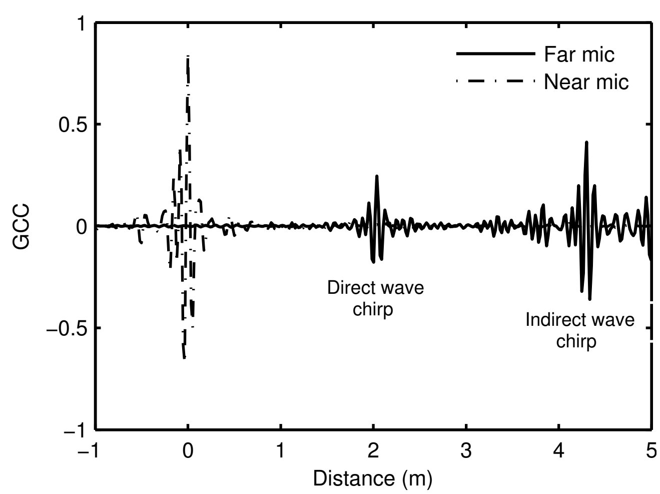

Fig. 4. Examples of normalized correlation output(times the speed of sound) for near and far wireless modules

Once we have the M^2 speaker-to-microphone distances, we use a $3(2M)$ dimensional gradient descent method, minimizing the mean-squared error between the measured and hypothesized speaker-to-microphone distances, similar to that used in [13], to determine the xyz coordinates for each microphone and speaker. After convergence, one module's microphone is selected as the origin and the results properly translated to the microphone-based, right-hand coordinate system. The result is that we determine the correct relative locations of the microphones (and speakers) with a mean error of less than 1.5cm. The entire calibration process for our 6-module array takes under a minute to complete using a c program on one core of a PC. Figure 5 shows the averaged error over ten calibrations for each configuration. The final errors are commensurate with the sum of all the potential sources of error – those from discrete sampling, temperature variations, non-point sources and sinks and background noise.

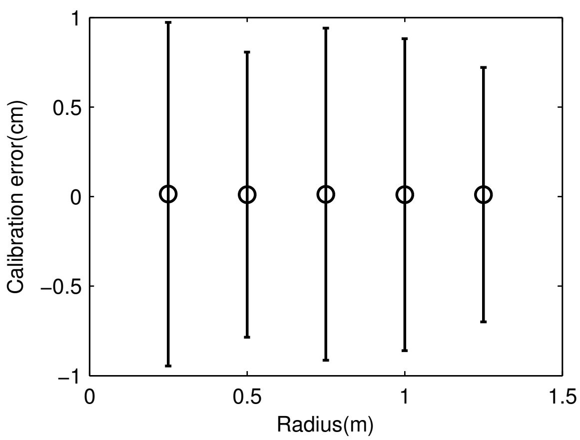

Fig. 5. Averaged mean error and standard deviation in microphone distance estimates for a six module system placed on the vertices of a regular hexagon of various radii

IV. CONCLUSION

We have discussed the major issues in putting together a self-configuring wireless microphone array, and have built a small proof-of-concept system using the ideas described herein. The hardware cost for each module built was about \$85, but we estimate that in modest quantity with today's technology the hardware cost of a module would be about \$15. It was shown that the system can perform the wireless transfers perfectly and configure itself properly and accurately, well within the tolerance needed for speech input. Our average measured error is only about 1.5cm which amounts to less than one sample of error, or a phase shift of 45^o at 4300Hz. Thus, for nearly all of the high-energy components of speech, the errors due to self-calibration will be small if not negligible.

Our proof-of-concept design, while having all the needed features, is much larger than we would like. Follow-on work will thus not only involve completing the algorithm set for the current system but also to designing a more numerous improved array system with smaller modules. We would have to verify that the calibration method extends to this higher dimensionality as well.

V. REFERENCES

[1] J. L. Flanagan, "Bandwidth design for speech-seeking microphone arrays," in *Proceedings of ICASSP85*, Tampa, FL, March 1985, pp. 732–735.

[2] ——, "Use of acoustic filtering to control the beamwidth of steered microphone arrays," *J. Acoust. Soc. Amer.*, vol. 78, no. 2, pp. 423–428, August 1985.

[3] P. M. Peterson, "Simulating the response of multiple microphones to a single acoustic source in a reverberant room," *J. Acoust. Soc. Amer.*, vol. 80, no. 5, pp. 1527–1529, November 1986.

[4] H. F. Silverman, "Some analysis of microphone arrays for speech data acquisition," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-35, no. 2, pp. 1699–1712, December 1987.

[5] J. L. Flanagan, J. D. Johnson, R. Zahn, and G. W. Elko, "Computer-steered microphone arrays for sound transduction in large rooms," *J. Acoust. Soc. Amer.*, vol. 78, no. 5, pp. 1508–1518, November 1985.

[6] H. F. Silverman, W. R. P. III, and J. L. Flanagan, "The huge microphone array (HMA)- Part I," *IEEE Transactions on Concurrency*, vol. 6, no. 4, pp. 36–46, October-December 1998.

[7] ——, "The huge microphone array (HMA) - Part II," *IEEE Transactions on Concurrency*, vol. 7, no. 1, pp. 32–47, January-March 1999.

[8] H. F. Silverman, W. R. P. III, and J. M. Sachar, "Factors affecting the performance of large-aperture microphone arrays," *(In submission) Journal of the Acoustical Society of America*, 2001.

[9] S. M. Golan, S. Gannot, and I. Cohen, "Performance analysis of a randomly spaced wireless microphone array," in *Proceedings of the 2011 IEEE International Conference on Acoustics Speech and Signal Processing ICASSP*, 2011, pp. 121–124.

[10] D. J. Menhill, M. Battiston, D. R. Wilson, J. R. Foote, and S. M. Doucet, *Field test of an affordable, portable, wireless microphone array for spatial monitoring of animal ecology and behaviour*. Wiley, 2012.

[11] *Omnidirectional Microphone with Bottom Port and Digital Output*, 01st ed., Analog Devices, Norwood, MA 02062, 2010-2011.

[12] "Analog devices: ADL5513 1mhz to 4ghz, 80db logarithmic detector-controller," www.analog.com, 2008.

[13] S. Khanal, H. Silverman, and R. Shakya, "A free-source method (frsm) for calibrating a large-aperture microphone array," *Accepted for publication in IEEE Transactions on Audio, Speech and Language Processing, also online at www.lems.brown.edu/array/mainpage*, 2013.

[14] C. H. Knapp and G. C. Carter, "The generalized correlation method for estimation of time delay," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-24, no. 4, pp. 320–327, August 1976.