16th European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland, August 25-29, 2008, copyright by EURASIP

OPTIMIZED DESIGN OF INTERPOLATED ARRAY AND SPARSE ARRAY WIDEBAND BEAMFORMERS

Gerhard Doblinger

Institute of Communications and Radio-Frequency Engineering
Vienna University of Technology
Gusshausstr. 25/389, A-1040 Vienna, Austria
phone: + (43) 1 58801 38927, fax: + (43) 1 58801 38999, email: gerhard.doblinger@tuwien.ac.at,
web: www.nt.tuwien.ac.at/staff/gerhard-doblinger/

ABSTRACT

We investigate two different design methods for wideband beamformers with applications to microphone arrays. Our major design goal is to reduce the number of sensors while maintaining a good beamforming performance. The different design methods considered are interpolated arrays with virtual sensors, and optimized sparse arrays. We present new optimization procedures for wideband beamforming with both types of arrays. Typical experimental results include array performance in case of noisy sensors, and in the presence of simulated room reverberations.

1. INTRODUCTION

The primary goal of an array beamformer design is to achieve a beam pattern with high directivity, i.e. a sharp mainlobe in conjunction with small sidelobe levels. The challenge thereby is to reach this goal with a minimum amount of sensors, with a small array size, and with a robustness against sensor position errors and sensor noise.

A conventional design of beamformers with a reduced sensor count is based on superdirective arrays (e.g. [1]). However, with high superdirectivity a large amplification of sensor noise and sensor position errors, especially in the low frequency range, must be taken into account. Sensor savings are also obtained with non-uniform arrays [2], and with well-known sparse arrays like nonredundant (NR) arrays [3], and minimum redundancy (MR) arrays [4]. These sparse arrays exhibit a narrow mainlobe width but suffer from significantly higher sidelobe levels. In addition, the large array size may be prohibitive in applications like microphone arrays. Sparse arrays with a smaller size can be designed using heuristic discrete optimization methods [5, 6, 7, 8].

A completely different design technique is based on array interpolation and extrapolation. The basic idea is to map an array of N sensors to an array of $M \geq N$ sensors by introducing virtual sensors. Originally, array interpolation has been used to cope with sensor failure by applying spatial linear prediction to create the missing sensor signals [9]. Interpolated arrays are also used to improve the performance of direction-of-arrival (DOA) estimation [10, 12]. Nonuniform arrays can be mapped to uniform arrays to enable the use of the fast root-MUSIC estimation algorithm [11]. In DOA estimation, the range of arrival directions is divided into sectors to get small interpolation errors. In beamforming applications, however, we want to process the whole range at once. Therefore, we must accept increased errors in a certain range of directions.

In this paper, we show that array interpolation can also be used in the design of beamformers. The main advantage of interpolated array beamformers is the possibility to improve a given array by software and not by changing the array layout or increasing the number of input sensors. In Section 2, we present a new interpolation/extrapolation algorithm for wideband beamformers like microphone arrays. For comparison reasons, we discuss a sparse array design by simulated annealing in Section 3. Representative simulation results of one-dimensional arrays operating under far-field and under simulated room conditions are given in Section 4, followed by concluding remarks in Section 5.

2. ARRAY INTERPOLATION

Array interpolation (and extrapolation) is carried out by preprocessing the N sensor signals to create $M \geq N$ sensor signals.[1] As shown in Fig. 1, virtual sensors are introduced by a multi-channel mapping system $T\{\cdot\}$. Array processing like beamforming is then applied to the M virtual sensor signals.

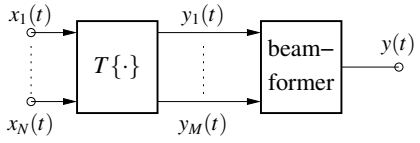

Figure 1: Mapping of sensor signals $x_i(t)$ to virtual sensor signals $y_i(t) = T\{x_i(t)\}$ linked with beamforming.

Sensor mapping in Fig. 1 is a very general technique that can be fitted to a large variety of array configurations. As an example, we can convert a nonuniform array to a uniform array with $M = N$, typically. Another example is extrapolation where we augment a uniform array of N sensors by $M - N$ sensors to create an M sensor uniform array. We intend to improve the beamforming behavior with the M sensor array.

Obviously, mapping $T\{\cdot\}$ depends on the geometries of the input array and of the virtual array. In addition, $T\{\cdot\}$ also depends on the signal model describing the wavefield under consideration. In this paper, we limit our discussion to one dimensional beamformers. It is a straightforward task to find $T\{\cdot\}$ in case of two or three dimensional beamformers. In the following, we use a far-field signal model where $x_i(t)$ are time-shifted versions $s(t-\tau_i)$ of a single source signal $s(t)$. The broadband source signal with spectrum $S(j\omega)$

[1] In the sequel, we generally speak of interpolation even if we append virtual sensors to the array.

is emitted from a direction given by azimuth $\phi \in [0,\pi]$. Therefore, the Fourier transform of the input signal vector $\mathbf{x}(t) = [x_1(t),\ldots,x_N(t)]^T$ is

$$\mathbf{X}(j\omega,\phi) = S(j\omega)\mathbf{d}(j\omega,\phi), \tag{1}$$

with steering vector $\mathbf{d}(j\omega,\phi) = [e^{-j\omega\tau_1(\phi)},\ldots,e^{-j\omega\tau_N(\phi)}]^T$. We implement a linear, time-invariant sensor mapping in the frequency domain using an $N \times M$ matrix $\mathbf{T}(j\omega,\phi)$:

$$\mathbf{Y}(j\omega,\phi) = S(j\omega)\mathbf{T}^H(j\omega,\phi)\mathbf{d}(j\omega,\phi) = S(j\omega)\mathbf{d}_V(j\omega,\phi), \tag{2}$$

where $\mathbf{d}_V(j\omega,\phi) = [e^{-j\omega\tau'_1(\phi)},\ldots,e^{-j\omega\tau'_M(\phi)}]^T$ is the steering vector of the desired virtual array (^{H} denotes conjugate transpose). This steering vector is then used to design the beamformer. In general, it is impossible to find $\mathbf{T}$ which fulfils (2) for desired frequency and azimuth ranges. We want to obtain a single matrix $\mathbf{T}(j\omega)$ for each frequency point needed by a wideband beamformer implementation using the FFT. Consequently, matrix $\mathbf{T}(j\omega)$ is usually obtained by minimizing a least-squares (LS) error criterion for a set of discrete $\phi = \phi_l \in [\phi_{\min},\phi_{\max}]$. In case of DOA estimation other more robust error criteria may be used [13, 14, 15]. When applying sensor mapping to a wideband beamformer design, we found that the standard least-squares method yields $\mathbf{T}$-matrices with large norms giving rise to a large amplification of sensor noise. This problem can be alleviated by the norm-constrained LS algorithm presented in the sequel.

Suppose we would like to find $\mathbf{T}$ of the desired mapping of steering vector $\mathbf{d}(j\omega,\phi)$ to $\mathbf{d}_V(j\omega,\phi)$ at L azimuths $\phi_l \in [\phi_{\min},\phi_{\max}]$. We collect the steering vectors in the $N \times L$ matrix $\mathbf{D}(j\omega) = [\mathbf{d}(j\omega,\phi_1),\ldots,\mathbf{d}(j\omega,\phi_L)]$, and $M \times L$ matrix $\mathbf{D}_V(j\omega) = [\mathbf{d}_V(j\omega,\phi_1),\ldots,\mathbf{d}_V(j\omega,\phi_L)]$, respectively. Our norm-constrained LS-approach can then be formulated as

$$\begin{aligned} \mathbf{t}_i(j\omega) &= \arg\min_{\mathbf{t}_i} \left\|\mathbf{D}^H(j\omega)\mathbf{t}_i(j\omega) - \bar{\mathbf{d}}_i(j\omega)\right\|_2^2 \\ &\text{with } \left\|\mathbf{t}_i(j\omega)\right\|_2 \le \alpha, \quad i = 1,\ldots,M, \end{aligned} \tag{3}$$

where $\mathbf{t}_i$ denotes the i-th column vector of $\mathbf{T}$, and $\bar{\mathbf{d}}_i$ is the i-th column vector of $\mathbf{D}_V^H$, respectively. The desired norm constraint in (3) is given by parameter α. This constrained LS-problem can be solved by the following algorithm [16]:

1. compute singular value decomposition $\mathbf{D}^H = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$ with singular values $\mathbf{\Sigma} = \mathrm{diag}([\sigma_1,\ldots,\sigma_N])$
2. determine rank r of $\mathbf{D}^H$ by finding smallest $\sigma_r \ge \varepsilon$
3. if $\sum_{k=1}^{r} \frac{|b_k|^2}{\sigma_k^2} \le \alpha^2$, with $\mathbf{b} = [b_1,\ldots,b_N]^T = \mathbf{U}^H\bar{\mathbf{d}}_i$

3a. then $\mathbf{t}_i = \sum_{k=1}^{r} \frac{b_k}{\sigma_k}\mathbf{v}_k$, with $\mathbf{V} = [\mathbf{v}_1,\ldots,\mathbf{v}_N]$

3b. else find $\lambda > 0$ that satisfies $\sum_{k=1}^{r} \frac{\sigma_k^2}{(\sigma_k^2+\lambda)^2}|b_k|^2 = \alpha^2$ set $\mathbf{t}_i = \sum_{k=1}^{r} \frac{\sigma_k b_k}{\sigma_k^2+\lambda}\mathbf{v}_k$

All steps of this algorithm are repeated for all column vectors of $\mathbf{D}_V$ to obtain $\mathbf{t}_i$, $i = 1,\ldots,M$. In addition, the whole procedure must be carried out for all desired frequency points $\omega_k \in [\omega_{\min},\omega_{\max}]$ (ω_k is omitted for clarity in the outline of the algorithm). Parameter λ is computed by numerical zero-finding of the function in step 3b. We can interpret the role of λ as automatic diagonal loading (regularization) of matrix $\mathbf{\Sigma}$. It should be noted that these steps are computed only once for given array layouts.

Storage demand and computational effort can be significantly reduced if sensor positions are symmetric around the center of the array and azimuth limits $\phi_{\min},\phi_{\max}$ are symmetric to $\pi/2$. Furthermore, a special case of extrapolation is given by extending a uniform array of N sensors at both ends by K virtual sensors. For such an array only $N \times K$ matrices $\mathbf{T}_l, \mathbf{T}_r$ are needed since (2) yields

$$\mathbf{d}_V = \mathbf{T}^H\mathbf{d} = \begin{pmatrix} \mathbf{T}_l^T \\ \mathbf{I} \\ \mathbf{T}_r^T \end{pmatrix}\mathbf{d} \tag{4}$$

($\mathbf{I}$ is the $N \times N$ identity matrix). With the above mentioned symmetries the elements of $\mathbf{T}_l$ and $\mathbf{T}_r$ are real-valued and related by

$$T_r(i,k) = T_l(N-i+1,K-k+1), \quad i = 1,\ldots,N,\ k = 1,\ldots,K. \tag{5}$$

Note that the N input sensor signals are not modified. Only the additional $2K$ virtual sensor signals are computed by the mapping system. A MATLAB program of the proposed algorithm is provided at the author's homepage.

3. ARRAY DESIGN BY SIMULATED ANNEALING

As discussed in Section 1, simulated annealing (SA) is a standard discrete optimization technique that has also been applied to the design of sparse arrays. In [6, 7, 8] both sensor positions and beamformer weights are determined by SA to obtain an array with a minimum number of sensors. In our different approach, we use SA only to find the optimum positions of N sensors located at $M > N$ possible locations. The weights are not computed by SA. We want to achieve a comparable wideband beampattern as a full array with all M positions occupied. For each selection of sensor positions, we compute the beamformer weights by a minimum-variance-distortionless-response (MVDR) design. An MVDR beamformer minimizes the output signal power but maintains signals from the desired direction [1]. The advantages of an MVDR design are its computational efficiency and an easy inclusion of desired spatial nulls in the beam pattern.

Our main objective is to reduce the degrees of freedom in the SA optimization algorithm. In case of wideband beamformers, weights must be optimized at e.g. up to 257 frequency points in the desired frequency range, assuming an FFT length of 512. This results in a large set of variables. Consequently, convergence of the SA algorithm may be poor and a high computational load is required. Furthermore, our approach has the advantage that desired spatial nulls are not influenced by the SA algorithm.

The SA cost function we use to select the sensor positions is the error norm between the desired wideband beam patterns of the full array and of the sparse array. Using (1) and $S(j\omega) = 1$, $\omega \in [\omega_{\min},\omega_{\max}]$, the wideband beam pattern of the array is given by

$$P(\phi) = \frac{1}{\pi}\int_{\omega_{\min}}^{\omega_{\max}} \left|\mathbf{w}(j\omega)^H\mathbf{d}(j\omega,\phi)\right|^2 d\omega, \tag{6}$$

with beamformer weight vector $\mathbf{w}(j\omega)$, and steering vector $\mathbf{d}(j\omega,\phi)$. The cost function to be minimized by the optimum sensor positions is actually computed with $P(\phi)$ evaluated at L discrete $\phi = \phi_l \in [0,\pi]$. Therefore, the wideband beam pattern is represented by $\mathbf{p} = [P(\phi_1),\ldots,P(\phi_L)]^T$. In addition, the integral in (6) is replaced by a sum since only discrete frequencies are used. Note that both the weight vector and the steering vector depend on the N sensor positions $\vec{\mathbf{R}} = [\vec{\mathbf{r}}_1,\ldots,\vec{\mathbf{r}}_N]$. Thus, the beam pattern also depends on $\vec{\mathbf{R}}$. The minimization problem solved by the SA algorithm is then expressed by

$$\vec{\mathbf{R}}_s = \arg\min_{\vec{\mathbf{R}}_s} \left\| \mathbf{p}_s(\vec{\mathbf{R}}_s) - \mathbf{p}_f \right\|_2, \qquad (7)$$

with sparse array sensor positions $\vec{\mathbf{R}}_s$, beam pattern $\mathbf{p}_s$, and full array beam pattern $\mathbf{p}_f$, respectively. The proposed SA design procedure can be summarized as follows:

1. compute MVDR weight vector $\mathbf{w}_f$ and $\mathbf{p}_f$ of the desired full array with M sensors
2. randomly select N sensor positions $\vec{\mathbf{R}}_s$ of the full array
3. select a start value T_0 of the annealing temperature T
4. set up a loop of N_i iterations, reducing T at each iteration index k
5. set up a loop of N_p iterations to permutate sparse array sensor positions at the M positions of the full array
6. compute MVDR weight vector $\mathbf{w}_s$ and wideband beam pattern $\mathbf{p}_s$ of the sparse array
7. compute cost function $C_{\text{new}} = \left\| \mathbf{p}_s - \mathbf{p}_f \right\|_2$ and $\delta C = C_{\text{new}} - C_{\text{old}}$, select a uniformly distributed random number $x \in [0,1]$
8. if $\delta C < 0$ or $x < e^{-\delta C/T}$, then $C_{\text{old}} = C_{\text{new}}$, save $\vec{\mathbf{R}}_s$
9. choose a new random permutation (new $\vec{\mathbf{R}}_s$) of the sparse array, and repeat at 6. until finished
10. modify T by $T = T_0/k$, and repeat at 5. until finished.

By selecting a proper starting temperature T_0 the cool down phase of the SA algorithms is slow enough to hopefully achieve a global minimum. The sensitivity to local minima is reduced by allowing the selection of solutions with higher cost functions. The probability for such a selection depends on T (see step 8.) and is high at the beginning of the iteration loop. Permutations of the sparse array are obtained by a random selection of two positions and exchanging their states (state = 1, if there is a sensor, 0 if not). The actual algorithm includes some refinements not given here for brevity. We refer to a MATLAB implementation of the proposed algorithm at the author's home page instead.

4. SIMULATION RESULTS AND COMPARISON

We implement the wideband beamformers with an overlap-add FFT filterbank for each sensor signal as shown in Fig. 2. The advantage as compared with beamformers using FIR filters is the direct application of interpolation matrices $\mathbf{T}[k]$, and frequency-domain weight vectors $\mathbf{w}[k]$. Weight vectors $\mathbf{w}[k]$ and matrices $\mathbf{T}[k]$ are computed at frequency points between 0.3 kHz and 6 kHz with a sampling frequency of 16 kHz and an FFT length of 512. Sensor spacing of uniform arrays is $d = 2.5$ cm to avoid spatial aliasing up to a frequency of $f_{\max} = \frac{c}{2d} = 6.8$ kHz (sound velocity $c = 340$ m/s).

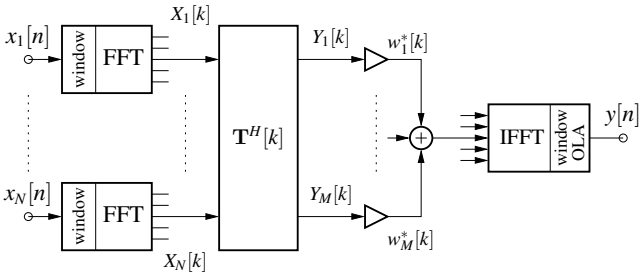

Figure 2: Interpolated wideband beamformer as overlap-add FFT filterbank (with processing at FFT index k shown).

In case of an interpolated array with a fixed desired beam pattern, we can avoid additional computations and storage demand, if we combine $\mathbf{T}[k]$ and $\mathbf{w}[k]$ in Fig. 2 to obtain new weight vectors $\widetilde{\mathbf{w}}[k] = \mathbf{T}[k]\mathbf{w}[k]$. In this case, the design of interpolated arrays is just another method to compute the beamformer weight vectors. The advantage of the system shown in Fig. 2 is its flexibility in regard to the combination of different beamformer designs with different mappings of array geometries.

In the following examples, weight vectors are computed with a standard MVDR design (signal model (1), diffuse noise field, diagonal loading of the spatio-spectral correlation matrix) [1]. As pointed out by an anonymous reviewer, the output of an MVDR beamformer is unaffected by a linear transform as long as we use ideal spatio-spectral correlation matrices. Thus, an MVDR design might not be the best example to be used with array interpolation. In a wideband beamformer design, however, ideal spatio-spectral correlation matrices are ill-conditioned and must be regularized. Therefore, in all MVDR design examples, we add the same constant $\mu = 0.1$ to the main diagonal of the respective spatio-spectral correlation matrix. The norm constraint parameter α of (3) is set to 4. We use $L = 100$ discrete azimuth values $\phi_l \in [0,\pi]$ needed by the optimization algorithms in Section 2 and 3.

In the example shown in Fig. 3, we map a uniform array with $N = 8$ input sensors to a nonuniform array with $M = 9$ virtual sensors. Therefore, both the number of sen-

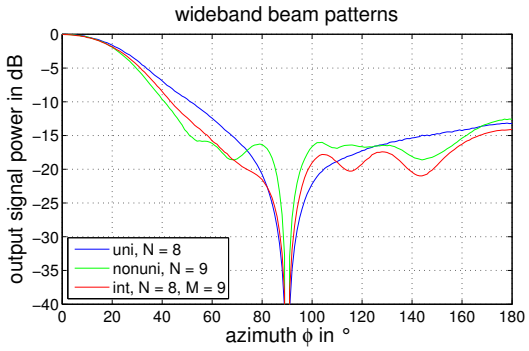

Figure 3: Measured far-field wideband beam patterns of $N = 8$ uniform array, $N = 9$ nonuniform array, and interpolation of uniform $N = 8$ array ($M = 9$ virtual sensors).

sors and the array geometry are changed by the mapping. The beamformer has a look direction $\phi_d = 0°$ (endfire), and a spatial null at $\phi_s = 90°$ (broadside). Beam pattern measurements are performed under far-field conditions using a directional pseudo-random source signal with white spectrum band-limited to 0.3 kHz ... 5.5 kHz. The sensors are assumed to be noise-free. If we mark a sensor presence by "1", then the nonuniform array has a layout given by 10001011111010001, with a sensor grid spacing of $d = 2.5$ cm. As indicated in Fig. 3, the larger size of this array reduces the mainlobe width when compared to the uniform array. Furthermore, the beam pattern of the interpolated array with 9 virtual sensors is close to that of the nonuniform array with 9 real sensors. In this example, array interpolation offers nearly the same performance with a much smaller array size (17.5 cm instead of 40 cm).

The influence of uncorrelated input sensor noise on the beam patterns is illustrated in Fig. 4 in case of an input signal-to-noise-ratio of 25 dB. The sidelobe levels of the interpolated array are slightly more increased as compared to the non-interpolated arrays. The noise sensitivity of the interpolated array would be much larger without the norm constraint imposed to the T-matrices.

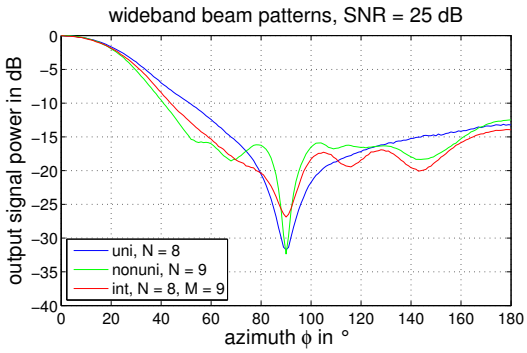

Figure 4: Measured far-field wideband beam patterns of $N = 8$ uniform array, $N = 9$ nonuniform array, and interpolation of uniform $N = 8$ array (noisy input sensors, SNR = 25 dB).

Next, we demonstrate the influence of simulated room acoustics on the beam patterns in Fig. 5.

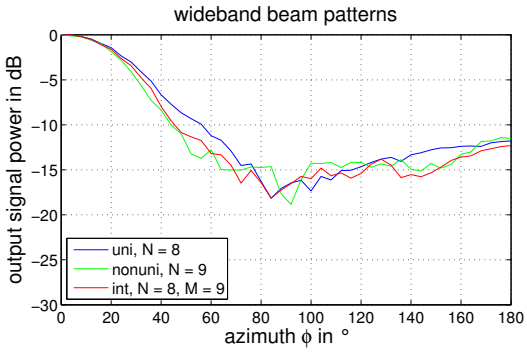

Figure 5: Simulated room wideband beam patterns of $N = 8$ uniform array, $N = 9$ nonuniform array, and interpolation of uniform $N = 8$ array (reverberation time $T_{60} = 0.13$ sec.).

Rooms are simulated using the image method [17] in combination with fractional delay interpolation to obtain an accurate signal delay to each sensor [18]. Room size is $L \times W \times H = 6\text{ m} \times 5\text{ m} \times 3\text{ m}$ with 0.13 sec. reverberation time T_{60}. Moderate room reverberations increase sidelobe levels by $2-4$ dB, and nearly remove spatial nulls. However, we observe only a minor influence on the mainlobe width.

The selected example of array interpolation illustrates just some aspects and is by no means exhaustive. Interested readers are invited to visit the author's homepage to see more examples.

We can obtain a similar beam pattern as in Fig. 3 with a sparse array design. Using the algorithm presented in Section 3, we get a sparse array layout 1110011110001 with an array length of 30 cm. This size is larger than that of the interpolated array but smaller than the size of the nonuniform array with $N = 9$ real sensors. The layout of a sparse array is optimized for the desired beam pattern. If we change e.g. look direction and spatial nulls, then the optimum layout will change too. As an example, we consider the beam patterns in Fig. 6. The optimized sparse arrays are designed with 8 sensors at 21, and at 31 possible locations.

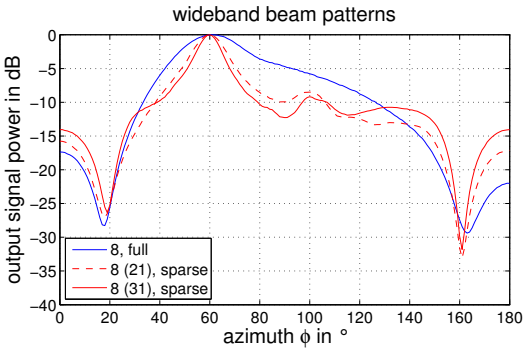

Figure 6: Measured far-field wideband beam patterns of $N = 8$ uniform array, sparse arrays with 8 of 21, and 8 of 31 sensor positions occupied.

The $8(21)$ array layout is 10001100110011001, and 10000010011001100100000001 for $8(31)$, respectively (leading and trailing zeros are not shown in the layout). Compared with interpolated arrays, the mainlobes are usually sharper whereas sidelobe levels are increased. In addition, the sparse arrays of this example show significantly larger sizes ($8(31)$: size $= 62.5$ cm, $8(21)$: size $= 42$ cm).

The result of a sensor noise experiment indicates a better performance of sparse arrays regarding sensitivity to noise (see Fig. 7). In general, MVDR sparse arrays have a larger aperture size and thus offer a lower noise gain, especially at low frequencies. In addition, room reflections also cause a lower degradation of the beam pattern (see Fig. 8).

We can summarize the comparison of sparse arrays and interpolated arrays as follows:

- Sparse arrays show a larger performance improvement for look directions between 45° and 90° than for look direction around 0°,
- sparse arrays have sharper mainlobes,
- sparse arrays are less prone to sensor noise, and room acoustics,
- sparse arrays show higher sidelobe levels,
- sparse arrays have significantly larger sizes,

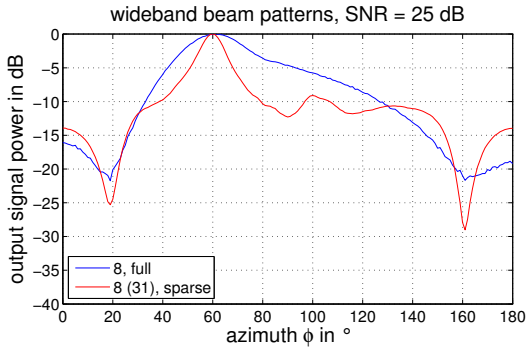

Figure 7: Measured far-field wideband beam patterns of $N = 8$ uniform array, and $8(31)$ sparse array in case of noisy sensors.

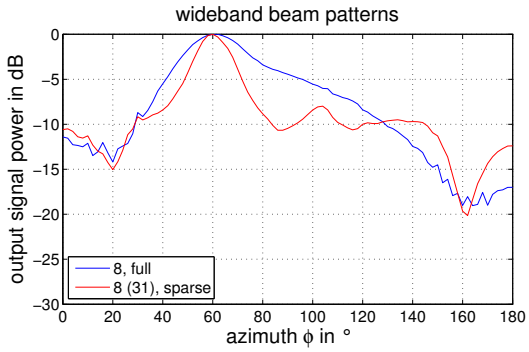

Figure 8: Simulated room wideband beam patterns of $N = 8$ uniform array, and $8(31)$ sparse array (reverberation time $T_{60} = 0.13$ sec.).

- sparse array layouts change, if design parameters (look direction, spatial nulls) are altered.

5. CONCLUSIONS

We have presented two different wideband beamformer design methods based on array interpolation, and on sparse arrays. Both methods exhibit a smaller number of sensors, a smaller array size, and offer a sharper mainlobe than conventional beamformer designs. An FFT filterbank beamformer implementation operating under far-field conditions and in simulated rooms support the performance improvements achieved by the proposed design methods.

Acknowledgement

The author would like to thank the anonymous reviewers for their valuable comments.

REFERENCES

[1] J. Bitzer, K. U. Simmer, "Superdirective microphone arrays," in *Microphone arrays*, M. Brandstein, and D. Ward (Eds.), Springer-Verlag, Berlin Heidelberg New York, 2001, ch. 2, pp. 19–38.

[2] R. W. Redlich, "Iterative least-squares synthesis of nonuniformly spaced linear arrays,", *Trans. Antennas and Propagation*, vol. AP-21, pp. 106–108, Jan. 1973.

[3] E. Vertatschitsch, S. Haykin, "Nonredundant arrays," *Proc. IEEE*, vol. 74, p. 217, Jan. 1986,

[4] A. T. Moffet, "Minimum-redundancy linear arrays," *IEEE Trans. Antennas and Propagation*, vol. AP-16, pp. 172–175, Mar. 1968.

[5] P. Jarske, T. Saramäki, S. K. Mitra, "On properties and design of nonuniformly spaced linear arrays," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 372–380, Mar. 1988.

[6] V. Murino, A. Trucco, C. S. Regazzoni, "Synthesis of unequally spaced arrays by simulated annealing," *IEEE Trans. Signal Processing*, vol. 44, pp. 119-123, Jan. 1996.

[7] A. Trucco, "Thinning and weighting of large planar arrays by simulated annealing," *IEEE Trans. Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 46, pp. 347–355, Mar. 1999.

[8] A. Trucco, "Synthesizing wide-band sparse arrays by simulated annealing," *Proc. Oceans 2001, MTS/IEEE Conference and Exhibitions*, vol. 2, pp. 989–994, Nov. 2001.

[9] D. N. Swingler, R. S. Walker, "Line-array beamforming using linear prediction for aperture interpolation and extrapolation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 16–30, Jan. 1989.

[10] T. P. Bronez, "Sector interpolation of nonuniform arrays for efficient high resolution bearing estimation," *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 5, pp. 2285–2888, Apr. 1988.

[11] B. Friedlander, "The root-MUSIC algorithm for direction finding with interpolated arrays," *Signal Processing*, vol. 30, pp. 15-29, Jan. 1993.

[12] A. B. Gershman, J. F. Böhme, "A note on most favorable array geometries for DOA estimation and array interpolation," *IEEE Signal Processing Letters*, vol. 4, pp. 232–235, Aug. 1997.

[13] M. Pesavento, A. B. Gershman, Z. Luo, "Robust array interpolation using second order cone programming," *IEEE Signal Processing Letters*, vol. 9, pp. 8–11, Jan. 2002.

[14] P. Hyberg, M. Jansson, B. Ottersten, "Array interpolation and bias reduction," *IEEE Trans. Signal Processing*, vol. 52, pp. 2711–2720, Oct. 2004.

[15] P. Hyberg, M. Jansson, B. Ottersten, "Array interpolation and DOA MSE reduction," *IEEE Trans. Signal Processing*, vol. 53, pp. 4464–4471, Dec. 2005.

[16] G. H. Golub, C. F. van Loan, *Matrix computations*, The Johns Hopkins University Press, Baltimore, 1985, ch. 12, pp. 405–408.

[17] J. B. Allen, D. A. Berkley, "Image method for efficiently simulating small-room acoustics," J. Acoust. Soc. Am., vol. 64, pp. 943–950, Apr. 1979.

[18] P. M. Peterson, "Simulating the response of multiple microphones to a single acoustic source in a reverberant room," J. Acoust. Soc. Am., vol. 80, pp. 1527–1529, Nov. 1986.