BFGUI: AN INTERACTIVE TOOL FOR THE SYNTHESIS AND ANALYSIS OF MICROPHONE ARRAY BEAMFORMERS

M. R. P. Thomas, H. Gamper, I. J. Tashev

Microsoft Research
Redmond, WA 98052, USA
{markth, hagamper, ivantash}@microsoft.com

ABSTRACT

Microphone arrays are beneficial for distant speech capture because the signals they capture can be exploited with beamforming to suppress noise and reverberation. The theory for the design and analysis of microphone arrays is well established, however the performance of a microphone array beamformer is often subject to conflicting criteria that need to be assessed manually. This paper describes BFGUI, a interactive graphical tool for MATLAB, for simulating microphone arrays and synthesizing beamformers, and whose parameters can be modified and performance metrics monitored in real-time. Primarily aimed at teaching and research, this tool provides the user with an intuitive insight into the effects of microphone types, number and geometry, and the influence of design constraints such as regularization and white noise gain on derived metrics. The resulting directivity pattern, directivity index and front-back ratio are examples of such metrics. Multiple analytic microphone models are supported and external measured microphone directivity patterns can also be loaded. The designs can be then exported in a variety of formats for processing of real-world data.

Index Terms— Microphone array, Beamformer, MVDR

1. INTRODUCTION

Microphone array beamformers have become increasingly prevalent in consumer electronic devices as they provide a robust and computationally straightforward method for spatial selectivity at the front end of a speech pipeline [1]. The design of microphone arrays is unfortunately subject to many conflicting requirements. For example, inter-microphone spacing affects the resolution of spatial sampling, so from the perspective of spatial aliasing it is desirable to have a small inter-microphone spacing. However, with small arrays the influence of increased noise coherence and reduced spatial diversity limits low frequency performance [2]. This is especially true in the design of small devices such as cellphones, in which omnidirectional Microelectromechanical Systems (MEMS) microphones may be spaced a few millimeters apart in either an endfire, broadside or planar configuration depending upon the number of microphones and whether the device is used in a hand-held or hands-free mode. In larger devices such as televisions and table-top computers, directional electret and condenser microphones have been known to be used in linear broadside and circular planar configurations. For research purposes, cylindrical and spherical microphone arrays have also become popular during the last decade [3]. In many cases, the influence of scattering in the microphone array enclosure is exploited to increase spatial sensitivity in a target direction, for which measured microphone directivity patterns are required [4].

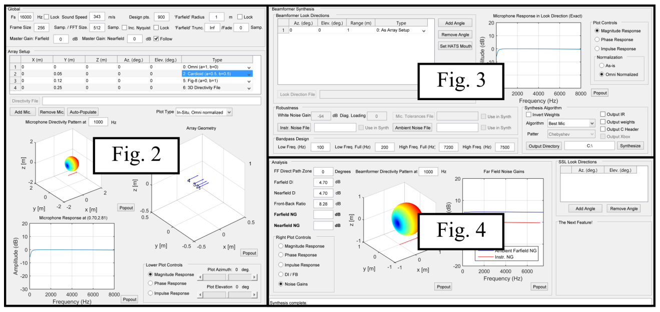


Fig. 1. Interface Overview

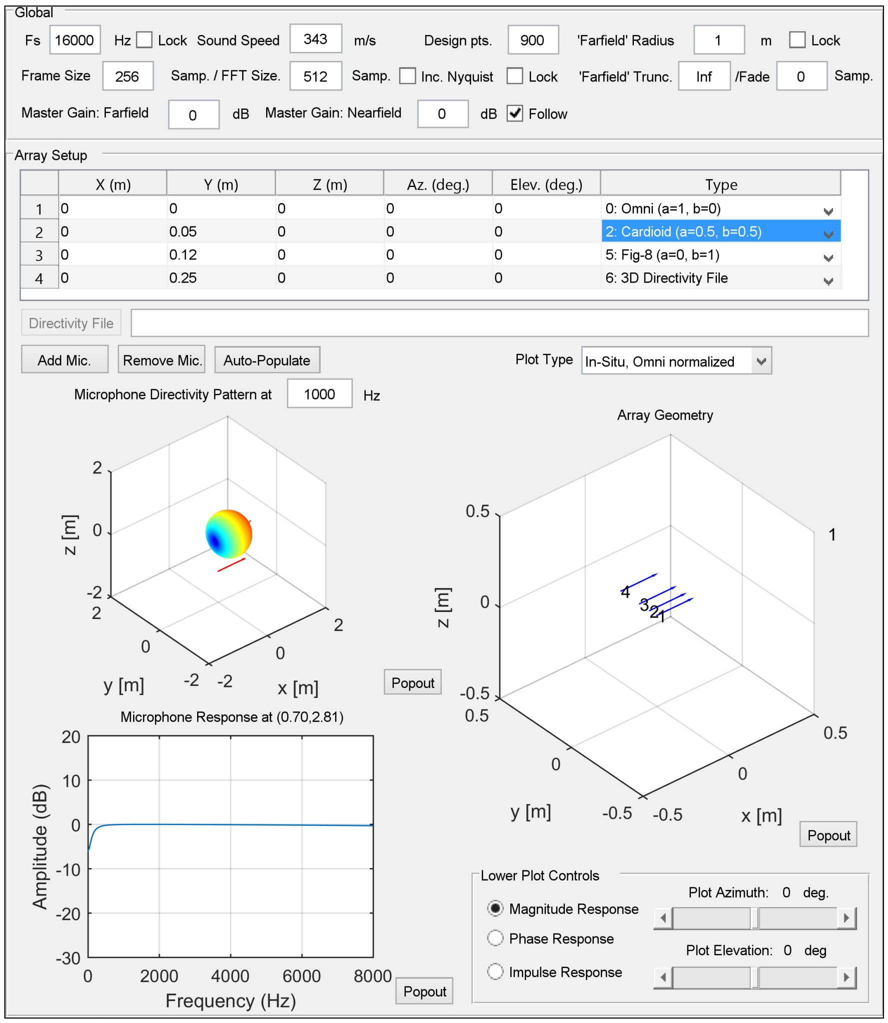
The theory of the design and analysis of time-invariant beamformers is well documented. Several standard algorithms such as delay-and-sum, Minimum Variance Distortionless Response (MVDR) [5] and beam pattern synthesis are commonly used and are relatively straightforward to implement. Performance metrics such as directivity index, front-back ratio and white noise gain are also well understood, and many standard textbooks e.g. [1, 2, 6, 7, 8] provide figures depicting the influence of design parameters on performance metrics. However there are relatively few readily available tools for interactively designing microphone array beamformers and discovering their behaviour when parameters are varied in real-time. This paper describes BFGUI [9], an interactive graphical tool programmed in MATLAB for simulating microphone arrays, synthesizing beamformers, and investigating performance metrics as design parameters are varied. Primarily it is designed as a teaching tool, but has also been used to synthesize beamformer weights based upon both microphone models and measured directivity patterns for deployment with real arrays for research purposes. All sessions can be saved in a human-readable format and reloaded at a later date, and multiple instances can run at one time. The remainder of this paper is organized in the order of a typical workflow.

2. DESCRIPTION

An overview of BFGUI is shown in Fig. 1 which is divided into four panels: Global, Array Setup, Beamformer Setup, and Analysis.

2.1. Terminology

The ‘farfield’ response is defined as the microphone directivity pattern as simulated or measured for a source at a fixed radius r and variable angle of incidence. For MVDR designs, the noise correla-


Global

Fs	16000 Hz	☐ Lock	Sound Speed 343 m/s	Design pts. 900	'Farfield' Radius 1 m	☐ Lock	
Frame Size	256	Samp. / FFT Size. 512 Samp.	☐ Inc. Nyquist	☐ Lock	'Farfield' Trunc. Inf	/Fade 0 Samp.	
Master Gain: Farfield	0 dB	Master Gain: Nearfield 0 dB	✓ Follow				

Array Setup

	X (m)	Y (m)	Z (m)	Az. (deg.)	Elev. (deg.)	Type
1	0	0	0	0	0	0: Omni (a=1, b=0)
2	0	0.05	0	0	0	2: Cardioid (a=0.5, b=0.5)
3	0	0.12	0	0	0	5: Fig-8 (a=0, b=1)
4	0	0.25	0	0	0	6: 3D Directivity File

Plot Type: In-Situ, Omni normalized

Microphone Directivity Pattern at 1000 Hz

Lower Plot Controls: [✓ Magnitude Response] Phase Response Impulse Response

Plot Azimuth: 0 deg.

Plot Elevation: 0 deg

Fig. 2. Global Variable and Array Setup Panels

tion matrices as described in Sec. 2.3 are derived exclusively from these responses. The 'nearfield' response is the set of impulse responses in the beamformer's look direction; while in many cases this is also a farfield measurement, it may be in the nearfield at radius r_0 in the case of close-talking microphones.

2.2. Global Variables and Microphone Array Setup Panels

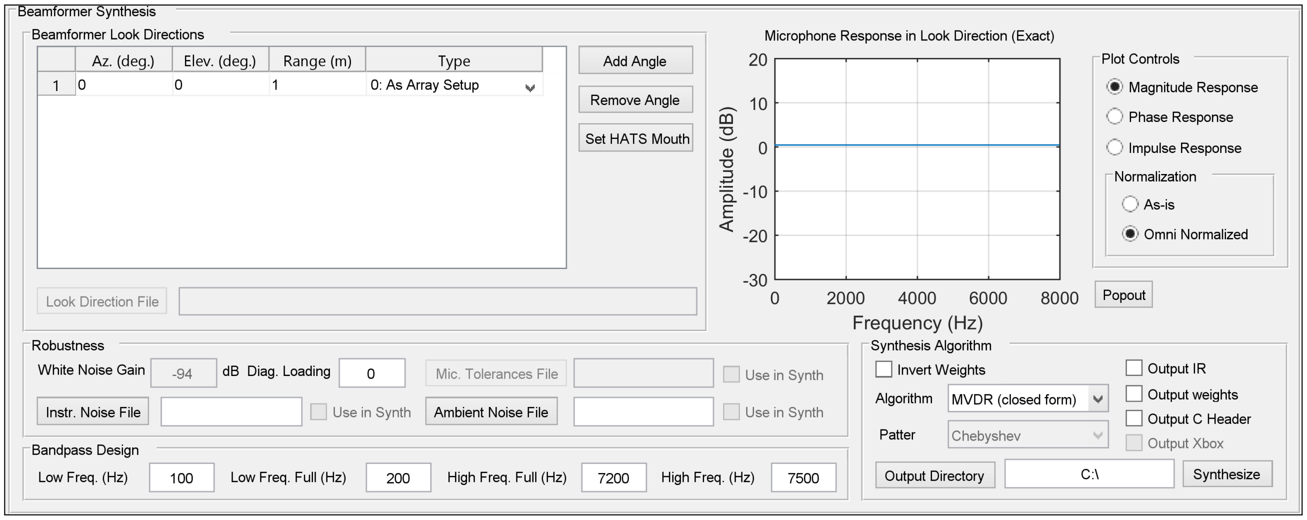
A typical session begins by setting global variables that are unlikely to change during design, although on-the-fly changes to the global variables are supported as all internal variables can be recalculated automatically. The Global panel is shown at the top of Fig. 2 and the corresponding descriptions are in Table 1. Optional lock boxes can be used to prevent accidental changes.

The Array Setup panel is shown in the lower half of Fig. 2. Individual microphones can be added or removed, or linear and circular arrays of microphones can be quickly designed with the 'Auto-Populate' button that opens a dialog box. Each microphone $m \in [1, M]$ has six parameters: position $\mathbf{x}_m = [x_m\ y_m\ z_m]^T$ [m], azimuth angle $\phi_m \in [0, 2\pi)$, elevation angle $\theta_m \in [-\pi/2, \pi/2]$, $\Omega_m \equiv (\theta_m, \phi_m)$, and type, which can either be a standard microphone model (omnidirectional, subcardioid, cardioid, supercardioid, hypercardioid, figure-8) [1] or a measured directivity pattern from an external file. A right-handed coordinate system is employed, with x pointing forward, y pointing left, and z pointing up with ϕ measured anticlockwise on the xy plane from the positive x-axis and elevation measured upward from the xy plane.

Let $\Omega_i \equiv (\theta_i, \phi_i)$ be an arbitrary angle of arrival with index $i \in [1, P]$. All complex directivity patterns are simulated by considering the transfer function between the microphone and a virtual source at (r, Ω_i). The directivity pattern of the microphone after rotation and translation to its intended position at frequency ω rad/s is denoted $\bar{\mathbf{G}}(\omega) \in \mathbb{C}^{P\times M}$, with entries $\bar{G}_m(r, \Omega_i, \omega)$. In many scenarios it is useful to remove the time of arrival and proximity gain or loss by dividing the complex response by that of an omnidirectional microphone in the center of the coordinate system, yielding $\mathbf{G}(\omega) \in \mathbb{C}^{P\times M}$, with entries

$$G_m(r, \Omega_i, \omega) = \frac{1}{r}\bar{G}_m(r, \Omega_i, \omega)e^{j\omega\tau_r}, \tag{1}$$

where τ_r is the time of arrival from any point on the bounding sphere of virtual sources to the center of the coordinate system. $\mathbf{G}(\omega)$ is used for all subsequent synthesis and analysis. Frequency ω is quan-



	Az. (deg.)	Elev. (deg.)	Range (m)	Type
1	0	0	1	0: As Array Setup

[● Magnitude Response] ○ Phase Response ○ Impulse Response

Normalization: ○ As-is [● Omni Normalized]

White Noise Gain -94 dB Diag. Loading 0

Low Freq. (Hz) 100 Low Freq. Full (Hz) 200 High Freq. Full (Hz) 7200 High Freq. (Hz) 7500

Algorithm MVDR (closed form)

Patter Chebyshev

Output Directory C:\

Fig. 3. Beamformer Setup Panel

Table 1. Global variables.

Variable	**Description**
Fs (Hz)	Sampling frequency, default $f_s = 16000$.
SoundSpeed (m/s)	Default $c = 343$.
Design pts	Total virtual sources, distributed uniformly on the surface of a bounding sphere [10]. Default $P = 900$, ranging $4 - 5000$.
'Farfield' Radius (m)	Virtual source radius to simulate real-world measurement rigs [11, 4]. Default $r = 1$ m; may take any value provided propagation time is less than the model length L.
Frame Size (samp)	Number of complex frequency bins used to model the system. Default $L/2 = 256$. If 'Inc. Nyquist' enabled, $L/2 + 1$.
FFT Size (samp)	Number of complex frequency bins used to calculate FFTs/IFFTs. Default $L = 512$. Changes to Frame Size alter FFT Size and vice versa automatically.
Truncation (samp)/ Fade (samp)	Truncate the impulse response to a fixed length with a raised-cosine fadeout. Default ∞ and 0 respectively.
Master Gain Farfield/Nearfield (dB)	Measured directivity patterns may require scaling if gains were not calibrated. Default: 0 dB. The follow box sets both nearfield and farfield to equal values.

tized with an FFT of length L for implementation, but is kept as a continuous variable for subsequent descriptions.

The microphone responses can be viewed within the tool as a directivity pattern at a particular frequency, or as magnitude response, phase response or impulse response in a particular direction as shown in the lower part of Fig. 2. Each plot can be manipulated using MATLAB's standard tools (pan, zoom, 3D rotate). Beside each figure is a 'Popout' button that creates a larger figure in a separate window that is useful when exporting for documentation. The 'Plot Type' drop-down list takes three values: 'Mic Model' (the prototype response prior to translation and rotation), 'In-Situ' (mic response after translation and rotation, yielding $\bar{\mathbf{G}}(\omega)$), and 'In-Situ, Omni Normalized' ($\mathbf{G}(\omega)$).

2.3. Beamformer Synthesis Panel

The beamformer's parameters are shown in Fig. 3. Each look direction can be specified in the nearfield at (r_0, Ω_0), producing response

$$d_m(\omega) = \bar{G}_m(r_0, \Omega_0, \omega). \tag{2}$$

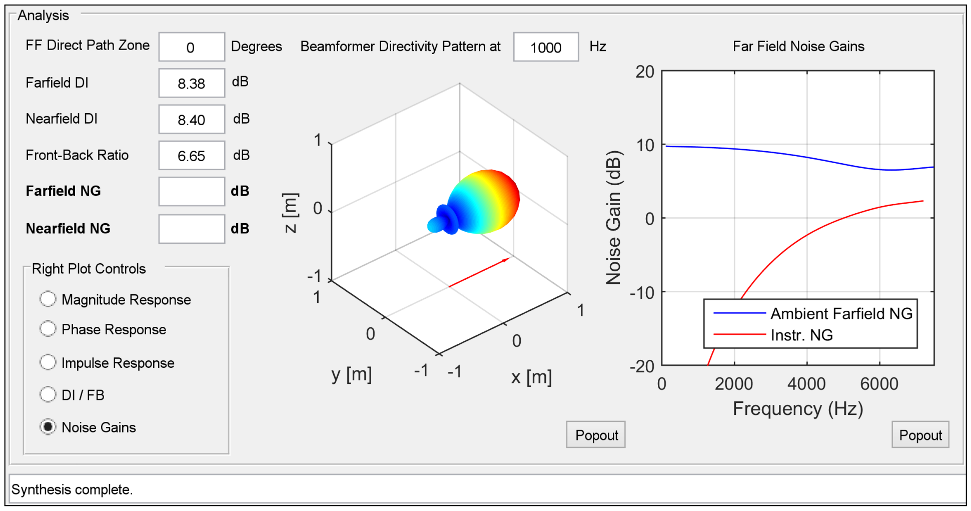
In vector form, $\mathbf{d}(\omega) = [d_1(\omega)\, d_2(\omega) \ldots d_M(\omega)]^T$. The type can be be set as either 'As Array Setup', in which case the same microphone models are used throughout, or 'From File', that allows external impulse responses or transfer functions to be loaded. The magnitude, phase and impulse responses can be viewed for each look direction in much the same way as the farfield directivity pattern.

The directivity pattern at the beamformer output is given by $\mathbf{B}(\omega) = \mathbf{G}(\omega)\mathbf{w}(\omega)$, where $\mathbf{w}(\omega) \in \mathbb{C}^{M\times 1}$ are the synthesized weights. Currently four synthesis algorithms are implemented as shown in Table 2.

Table 2. Beamformer Synthesis Algorithms

Algorithm	**Description**
Best Mic	Picks mic yielding highest directivity index.
Delay & Sum	$\mathbf{w}(\omega) = \frac{1}{M\mathbf{d}(\omega)^*}$
MVDR (closed)	$\mathbf{w}(\omega) = \frac{\left(\mathbf{\Phi}_{N'N'}^{-1}(\omega)\mathbf{d}(\omega)\right)^*}{\mathbf{d}^H(\omega)\mathbf{\Phi}_{N'N'}^{-1}(\omega)\mathbf{d}(\omega)}$
MVDR (adapt.)	$\mathbf{w}(\omega) = \arg\min_{\mathbf{w}(\omega)} \mathbf{G}(\omega)\mathbf{w}(\omega)$, subject to $\mathbf{w}^T(\omega)\mathbf{d}(\omega) = 1,\ \frac{\lvert\mathbf{w}(\omega)^T\mathbf{d}(\omega)\rvert^2}{\mathbf{w}(\omega)^H\mathbf{w}(\omega)} \geq \gamma$

The variable $\mathbf{\Phi}_{NN}(\omega) = \mathbf{G}(\omega)^H\mathbf{G}(\omega)/P$ is a noise correlation matrix under the assumption of a spatially homogeneous and isotropic noise field [12] and $\mathbf{\Phi}_{N'N'} = \mathbf{\Phi}_{NN} + \kappa\mathbf{I}$ is a regularized form. In the MVDR (closed form) case, robustness to sensor noise and mismatch is controlled by the κ in the regularization term, yielding the delay and sum result when $\kappa \to \infty$, implicitly increasing the white noise gain (WNG) and therefore robustness at the expense of


Analysis

FF Direct Path Zone: 0 Degrees

Farfield DI: 8.38 dB

Nearfield DI: 8.40 dB

Front-Back Ratio: 6.65 dB

Farfield NG: dB

Nearfield NG: dB

Right Plot Controls: Magnitude Response, Phase Response, Impulse Response, DI / FB, [✓ Noise Gains]

Beamformer Directivity Pattern at 1000 Hz

Synthesis complete.

Fig. 4. Beamformer Analysis Panel

directivity index. Alternatively, MVDR (adaptive) explicitly controls white noise gain by formulating the design as a constrained optimization problem that is convex but has no closed-form solution [13], and is solved using the CVX toolbox [14, 15]. Pattern synthesis for standard patterns such as Chebyshev and high-order cardioids will be implemented in a future release. The design can be bandlimited by four parameters, specifying the frequencies at which the response should be zero, and at unity (ω_1, ω_2), with frequencies in between faded in/out using a raised cosine window.

Optional noise files contain ambient noise spectra $N_{\text{A}}(\omega)$ and instrumental noise spectra $N_{\text{I}}(\omega)$ as captured for a specific microphone in a specific noise environment. Some beamformer synthesis algorithms e.g. [1] rely on these measurements to synthesize beamformers optimized to maximize noise gain, although they are used solely to calculate noise performance in BFGUI.

Any number of beams can be entered. Following synthesis (which typically takes less than a second per beam), the result can be exported as (a) a tab-separated list of complex frequency domain coefficients, (b) an M-channel .WAV file of impulse responses, calculated by IFFT$\rightarrow$circular shift$\rightarrow$raised cosine window, or (c) a C header file containing the same values as (a).

2.4. Performance Metrics

The beamformer's response can be analyzed in several different ways, as shown in Fig. 4. As the user steps through the beams with the arrow keys, metrics are updated instantaneously for quick comparison. The effect of a change to the array geometry or beamformer synthesis can be determined quickly as the performance metrics are updated automatically after resynthesis.

The directivity pattern is shown at a user-specified frequency in a similar way to the microphone models. Several derived metrics are provided as plots and single numbers, calculated according to Table 3. The integrals are evaluated as discrete sums over P near-uniform points calculated in [10]. Performance factors are converted to indices by taking $10\log_{10}$ to convert to dB. The magnitude, phase and impulse responses in the look direction can be viewed as a sanity check, and plots can be made that represent these metrics graphically. The noise gain is only reported if measured ambient and instrumental noise spectra $N_0(\omega)$, and $N_I(\omega)$ are provided by the user.

Table 3. Performance Metrics

Metric	**Description**
Directivity Factor	$\text{DF}(\omega) = \frac{\lvert B(\theta_0,\phi_0,\omega)\rvert^2}{\frac{1}{4\pi}\int_0^{2\pi}\int_0^{\pi}\lvert B(\theta,\phi,\omega)\rvert^2 \sin\theta d\theta d\phi}$
Directivity Factor (av.)	$\overline{\text{DF}} = \int_{\omega_1}^{\omega_2} \text{DF}(\omega)$
Front-Back Factor	$\text{FBF}(\omega) = \frac{\int_{\theta_0-\pi/2}^{\theta_0+\pi/2}\int_{\phi_0-\pi/2}^{\phi_0+\pi/2}\lvert B(\theta,\phi,\omega)\rvert^2 \sin\theta d\theta d\phi}{\int_{\theta_0+\pi/2}^{\theta_0+3\pi/2}\int_{\phi_0+\pi/2}^{\phi_0+3\pi/2}\lvert B(\theta,\phi,\omega)\rvert^2 \sin\theta d\theta d\phi}$
Front-Back Factor (av.)	$\overline{\text{FBF}} = \int_{\omega_1}^{\omega_2} \text{FBF}(\omega)$
WNG	$\text{WNG}(\omega) = \frac{\lvert \mathbf{w}(\omega)^T \mathbf{d}(\omega)\rvert^2}{\mathbf{w}(\omega)^H \mathbf{w}(\omega)}$
WNG (av.)	$\overline{\text{WNG}} = \int_{\omega_1}^{\omega_2} \text{WNG}(\omega)$
Noise Gain	$\text{NG}(\omega) = \frac{\lvert N_{\text{A}}(\omega)\rvert^2 + \lvert N_{\text{I}}(\omega)\rvert^2}{\frac{1}{\text{DF}(\omega)}\lvert N_{\text{A}}(\omega)\rvert^2 + \frac{1}{\text{WNG}(\omega)}\lvert N_{\text{I}}(\omega)\rvert^2}$

Two modifications to the standard performance metrics in table 3 are provided. Firstly, the numerator of the Directivity Factor can be set to integrate over a solid angle as set by 'FF Direct Path Zone'. This is useful for getting more stable results with noisy measured directivity patterns. A differentiation is also made between nearfield and farfield directivity index; the farfield definitions are stated in Table 3. In the nearfield case, the values are scaled to compensate for the proximity gain in $\mathbf{d}(\omega)$ when $r_0 < r$.

3. CONCLUSION

BFGUI is a MATLAB tool for interactive experimentation with microphone arrays, supporting 3D array geometry, analytic and measured microphone directivity patterns, and four standard beamformer synthesis algorithms. As the user modifies the array geometry and synthesis parameters, numerical and graphical metrics provide insight into the influence of each parameter on performance. All plots can be exported as MATLAB figures for easier manipulation and sessions can be saved in a human-readable format that can be reloaded at a later date. BFGUI is useful both as a teaching/research tool and for the synthesis of beamformer filters that can be exported for real-world use.

4. REFERENCES

[1] I. Tashev, *Sound Capture and Processing: Practical Approaches*, Wiley, 2009.

[2] M. S. Brandstein and D. B. Ward, Eds., *Microphone Arrays: Signal Processing Techniques and Applications*, Springer-Verlag, Berlin, Germany, 2001.

[3] B. Rafaely, "Analysis and design of spherical microphone arrays," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 1, pp. 135–143, Jan. 2005.

[4] M. R. P. Thomas, J. Ahrens, and I. Tashev, "Optimal 3D beamforming using measured microphone directivity patterns," in *Proc. Intl. Workshop Acoustic Signal Enhancement (IWAENC)*, Aachen, Germany, Sept. 2012.

[5] O. L. Frost, III, "An algorithm for linearly constrained adaptive array processing," *Proc. IEEE*, vol. 60, no. 8, pp. 926–935, Aug. 1972.

[6] H. L. van Trees, *Optimum Array Processing*, Detection, Estimation and Modulation Theory. Wiley, 2002.

[7] J. Benesty, J. Chen, and Y. Huang, *Microphone Array Signal Processing*, Springer-Verlag, Berlin, Germany, 2008.

[8] Y. Huang, J. Benesty, and J. Chen, *Acoustic MIMO Signal Processing*, Springer-Verlag Berlin Heidelberg, 2006.

[9] Microsoft Research, "Downloads," http://research.microsoft.com/apps/catalog/default.aspx?t=downloads, 2015.

[10] J. Fliege and U. Maier, "A two-stage approach for computing cubature formulae for the sphere," in *Mathematik 139T, Universitat Dortmund, Fachbereich Mathematik, Universitat Dortmund, 44221*, 1996.

[11] V. R. Algazi, R. O. Duda, D. M. Thompson, and C. Avendano, "The CIPIC HRTF database," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, New York, USA, Oct. 2001, pp. 99–102.

[12] G. W. Elko, "Superdirectional microphone arrays," in *Acoustic Signal Processing for Telecommunication*, S. L. Gay and J. Benesty, Eds., chapter 10, pp. 181–237. Kluwer Academic Publishers, Hingham, MA, USA, 2000.

[13] E. Mabande, A. Schad, and W. Kellermann, "Design of robust superdirective beamformers as a convex optimization problem," in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Taipei, Taiwan, Apr. 2009.

[14] CVX Research, Inc., "CVX: Matlab software for disciplined convex programming, version 2.0," http://cvxr.com/cvx, Aug. 2012.

[15] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, V. Blondel, S. Boyd, and H. Kimura, Eds., Lecture Notes in Control and Information Sciences, pp. 95–110. Springer-Verlag, 2008.