

ALGORITHMS FOR IDENTIFYING CLUSTERS OF NEAR-COMMON ZEROS IN MULTICHANNEL BLIND SYSTEM IDENTIFICATION AND EQUALIZATION

Andy W.H. Khong, Xiang (Shawn) Lin, Patrick A. Naylor

Department of Electrical and Electronic Engineering, Imperial College London, UK

ABSTRACT

Blind system identification (BSI) and equalization algorithms have been applied to multichannel systems with high order such as found in acoustic impulse responses. Studies on the performance of such algorithms in the presence of near-common zeros have been limited to low order systems. In this work, we propose two high order clustering algorithms which efficiently extract clusters of near-common zeros within a specified pairwise distance in the z -plane. Using these algorithms, we then quantify the number of common zeros that exist in acoustic systems. In addition, we show how these algorithms can be applied to study of BSI and equalization algorithms in the presence of near-common zeros for such acoustic systems.

Index Terms— blind system identification, channel equalization, near-common zeros, acoustic array

1. INTRODUCTION

Blind system identification (BSI) and equalization are of importance for acoustic dereverberation where the main aim is to recover the source signal using channel equalizers derived from the inverse of the estimated impulse responses. Several BSI [1, 2] and equalization [3] algorithms based on second-order statistics have been shown to identify and equalize high order acoustic impulses of length in the order of hundreds of coefficients for realistic applications. One of the *identifiability conditions* for these algorithms is that the channels are coprime, i.e., they do not share common zeros [4]. If common roots exist across all channels, BSI algorithms, for example, fail to identify the channels correctly as they cannot distinguish whether the common terms are due to the input signal or the acoustic channels.

The notion of *near-common zeros* was introduced recently in [5, 6]. It has been shown that, for a simple two-channel system each having three coefficients, the rate of convergence of an adaptive BSI algorithm [4] reduces with the reduction of the Euclidean distances between zeros of different channels. Studies pertaining to the effect of near-common zeros on the performance of BSI algorithms have generally been limited to two channels with small number of zeros. Although an algorithm for estimating zeros that are exactly common has been proposed for two-channel systems [5], no algorithm exists however for the clustering of near-common zeros in high order multichannel systems. Identifying clusters of near-common zeros in such systems is therefore important and, in this work, we develop two clustering algorithms which will provide a quantification of how close to being common these zeros are for an acoustic system so as to study how these near-common zeros affect BSI and channel equalization algorithms. The propose algorithms can also aid the design of algorithms robust to near-common zeros for multichannel high order system identification and equalization.

We first describe conditions whereby zeros can be considered near-common. We explain why, under these conditions, classical

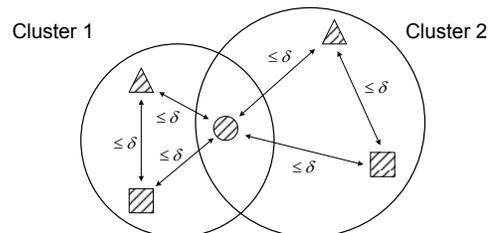


Fig. 1. An example of two clusters for a three-channel system in the z -plane.

clustering algorithms will not operate effectively in this context. Using a novel approach, we compute a dissimilarity matrix containing the Euclidean distances between zeros across any two channels as presented in Section 3. Utilizing this matrix, two algorithms namely (i) the Divide-and-Conquer (DC) and (ii) the Search-and-Trim (ST) are proposed in Section 4 for the identification of clusters of near-common zeros. We present, in Section 5, simulation results depicting the performance of these algorithms in terms of their computational times. In addition, we illustrate how the proposed algorithms can be applied to study the performance of BSI and channel equalization algorithms for acoustic channels in the presence of near-common zeros.

2. CONDITIONS FOR CLUSTERING

For a multichannel system, clusters of near-common zeros must satisfy two conditions: (i) the number of zeros within each cluster must correspond to the number of channels for that system with each channel contributing exactly one zero and (ii) all possible pairs of zeros in a cluster must lie within a vicinity δ in terms of their Euclidean distances where $\delta \geq 0$ is defined as the tolerance. The first condition results from the definition of zeros being near common across *all* channels. Condition (ii) defines the closeness between *pairs* of near-common zeros. It is worthwhile noting that the channel disparity depends on these pairwise distances [7] and any zero can be a member of more than one cluster. This implies that classical clustering algorithms such as the k - and c -means algorithms [8] cannot be employed for our application since the k -means algorithm requires a *priori* knowledge of the number of clusters and it assumes that each zero belongs to only one cluster. The c -means algorithm on the other hand violates condition (i) since it does not appropriately constrain the number of zeros within a cluster. Figure 1 shows an illustrative example of two clusters of near-common zeros in the z -plane for a three-channel system satisfying the above conditions. Symbols \triangle , \square and \circ represent zeros for each channel and they lie within pairwise δ vicinity from each other.

3. EUCLIDEAN DISTANCES BETWEEN ZEROS FOR TWO CHANNELS

Extraction of clusters of near-common zeros in a multichannel system involves the computation of the Euclidean distances between any pair of zeros from different channels. We describe a novel approach to compute these distances between any two channels. We assume that each of these high order polynomials has been factorized using efficient factorization algorithms presented in [9]. Defining $\mathbf{h}_m = [h_m(0) \ h_m(1) \ \dots \ h_m(L)]^T$ as the m^{th} channel impulse response of length $L + 1$, with $[\cdot]^T$ being the transposition operator, we can then express for $z = e^{i2\pi f}$ and $i = \sqrt{-1}$,

$$H_m(z) = h_m(0) + \dots + h_m(L)z^{-L} = K \prod_{p=1}^L (z - z_m(p)),$$

where f is the normalized frequency and K is the gain constant. The term $z_m(p) = x_m(p) + iy_m(p)$ is the p^{th} zero and its location in the z -plane is defined by $x_m(p)$ and $y_m(p)$ along the real and imaginary axis of the unit circle respectively. It has been shown [10] that as L increases, the radii of these zeros tend toward unity while their angles tend toward a uniform distribution.

The rate of convergence for adaptive BSI algorithms reduces with the reduction of Euclidean distances between the zeros [6]. In addition, the relationship between channel disparity and pairwise distances of the zeros has been established in [7]. To quantify these pairwise distances, we introduce an $L \times L$ dissimilarity matrix $\mathbf{D}_{\{m,n\}}$ defined between channels m and n where the p^{th} row and q^{th} column element is given by

$$\begin{aligned} D_{\{m,n\}}(p, q) &= |z_m(p) - z_n(q)| \\ &= \sqrt{[x_m(p) - x_n(q)]^2 + [y_m(p) - y_n(q)]^2} \end{aligned} \quad (1)$$

for $p, q = 1, \dots, L$ and $m \neq n$. Unless the zeros are exactly common, (i) the diagonal elements of $\mathbf{D}_{\{m,n\}}$ are non-zero and (ii) $\mathbf{D}_{\{m,n\}}$ is not symmetric. We next define two $L \times 1$ vectors

$$\mathbf{z}_m = [z_m(1) \ z_m(2) \ \dots \ z_m(L)]^T, \quad (2)$$

$$\mathbf{z}_n = [z_n(1) \ z_n(2) \ \dots \ z_n(L)]^T \quad (3)$$

containing L zeros of the m^{th} and n^{th} channel respectively. Defining $\tilde{\mathbf{Z}}_m = \mathbf{Z}_m \odot \mathbf{Z}_m$ with \odot being the Hadamard product, $\mathbf{Z}_m = \mathbf{z}_m \mathbf{1}^T$, $\mathbf{Z}_n = \mathbf{1} \mathbf{z}_n^T$ and $\mathbf{1}_{L \times 1} = [1 \ 1 \ \dots \ 1]^T$, computation of (1) for all p and q can be efficient using

$$\mathbf{D}_{\{m,n\}} = \left[\left[\tilde{\mathbf{Z}}_m - 2\mathbf{z}_m \mathbf{z}_n^T + \tilde{\mathbf{Z}}_n \right]^{\circ \frac{1}{2}} \right], \quad (4)$$

with $[\cdot]^{\circ \frac{1}{2}}$ and $|\cdot|$ being the Hadamard square root and elemental absolute respectively. To illustrate the validity of (4), we write

$$\begin{aligned} &\tilde{\mathbf{Z}}_m - 2\mathbf{z}_m \mathbf{z}_n^T + \tilde{\mathbf{Z}}_n \\ &= \begin{bmatrix} (z_m(1) - z_n(1))^2 & \dots & (z_m(1) - z_n(L))^2 \\ \vdots & \ddots & \vdots \\ (z_m(L) - z_n(1))^2 & \dots & (z_m(L) - z_n(L))^2 \end{bmatrix}. \end{aligned} \quad (5)$$

Let $a = x_m(p) - x_n(q)$ and $b = y_m(p) - y_n(q)$. Invoking Euler's identity $a + ib = r e^{i\theta}$ where $r = \sqrt{a^2 + b^2}$ and $\theta = \tan^{-1}(b/a)$, we have $|(a + ib)^2| = |r^2 e^{i2\theta}| = r^2$, from which we obtain the

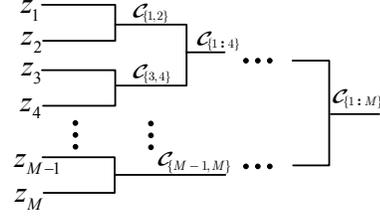


Fig. 2. The GMC-DC algorithm.

important result

$$\begin{aligned} \mathbf{D}_{\{m,n\}} &= \left[\left[\tilde{\mathbf{Z}}_m - 2\mathbf{z}_m \mathbf{z}_n^T + \tilde{\mathbf{Z}}_n \right]^{\circ \frac{1}{2}} \right] \\ &= \begin{bmatrix} \sqrt{|(z_m(1) - z_n(1))^2|} & \dots & \sqrt{|(z_m(1) - z_n(L))^2|} \\ \vdots & \ddots & \vdots \\ \sqrt{|(z_m(L) - z_n(1))^2|} & \dots & \sqrt{|(z_m(L) - z_n(L))^2|} \end{bmatrix} \end{aligned}$$

hence verifying the validity of (4). For $m \neq n$, since all pairwise distances are computed only once, no computational redundancy occurs making this computation efficient.

4. GENERALIZED MULTICHANNEL CLUSTERING

We now propose two algorithms for identifying clusters of near-common zeros in multichannel systems. For the two-channel case, clusters can be obtained from $\mathbf{D}_{\{1,2\}}$ by selecting indices p and q for which $D_{\{1,2\}}(p, q) \leq \delta$. We generalize this approach by employing $\mathbf{D}_{\{m,n\}}$ with the aim of extracting clusters for M -channel systems from which elements in $\{\mathbf{z}_1 \ \mathbf{z}_2 \ \dots \ \mathbf{z}_M\}$ satisfy conditions as described in Section 2.

Employing $\mathbf{D}_{\{m,n\}}$ between any two out of M channels, consider the case where there are c_{mn} clusters of near-common zeros between channels m and n . We next define a sub-cluster group matrix $\mathbf{C}_{\{m,n\}}$ containing these c_{mn} clusters. This $c_{mn} \times 2$ matrix can be obtained by searching within elements in $\mathbf{D}_{\{m,n\}}$ for indices p and q such that

$$\mathbf{C}_{\{m,n\}} = \arg D_{\{m,n\}}(p, q) \leq \delta \quad (6)$$

is satisfied. We next denote c_t as the total number of clusters for the M -channel system. We define cluster matrix $\mathbf{C}_{\{1:M\}}$ of dimension $c_t \times M$ for the whole system where each row contains a cluster with M elements each containing indices corresponding to elements in $\{\mathbf{z}_1 \ \mathbf{z}_2 \ \dots \ \mathbf{z}_M\}$ for the respective channels 1 to M . We employ (6) across each pair of channels selected from M . The aim of the proposed Generalized Multichannel Clustering (GMC) algorithm is to obtain $\mathbf{C}_{\{1:M\}}$ using sub-cluster groups $\mathbf{C}_{\{m,n\}}$. This can be achieved using two approaches as described below.

4.1. The Divide-and-Conquer Algorithm

The GMC Divide-and-Conquer (GMC-DC) algorithm compares two channels in a binary tree fashion as depicted in Fig. 2. Stage 1 computes sub-cluster groups $\mathbf{C}_{\{1,2\}}, \mathbf{C}_{\{3,4\}}, \dots$ using (4) and (6). The second stage then extracts sub-cluster groups $\mathbf{C}_{\{1:4\}}, \mathbf{C}_{\{5:8\}}, \dots$ from Stage 1 and this process is repeated until comparisons between all branches of the binary tree have been completed. Condition (ii) requires that any pairwise distances greater than tolerance δ are excluded from the cluster during each stage. The number of stages

Table 1. The GMC-ST algorithm

```

10  Compute  $\mathcal{C}_{\{m,n\}}$ , Get  $\mathcal{C}_{\{m_s,n_s\}}$  using (7)
20  for each row of  $\mathcal{C}_{\{m_s,n_s\}}$ , initialize  $\mathbf{R} = \mathbf{0}_{1 \times M}$ 
30      set  $r(m_s) = p, r(n_s) = q$ 
40      while  $\mathbf{R}$  contains no empty elements
50          for each row  $\mathbf{r}$  in  $\mathbf{R}$ 
60              find all elements of  $\mathbf{r}$  within  $\mathcal{C}_{\{m,n\}}$ 
70              if found  $k$  values invoke (8)
80              else delete row
90              if  $\mathbf{R}$  has < 1 row GOTO 20
100             else
110             for each row of  $\mathbf{R}$ 
120                 Check if all pairwise elements  $\in \mathcal{C}_{\{m,n\}}$ 
130                 if yes GOTO 110, else delete row
140             end for
150         end for
160     end while
170      $\mathbf{R} \in \mathcal{C}_{\{1:M\}}$ 
180 end for

```

required for this approach corresponds to $\lfloor \log_2(M) + 1 \rfloor$ while the computation of $\mathbf{D}_{\{m,n\}}$ using (4) is required a total of $M - 1$ times.

4.2. The Search-and-Trim Algorithm

The GMC Search-and-Trim (GMC-ST) algorithm, given in Table 1, first computes $\mathbf{D}_{\{m,n\}}$ for all possible pairs of channels with $m = 1, \dots, M - 1$ and $n = m + 1$. This employs (4) a total of $0.5M(M - 1)$ times. Invoking (6), this algorithm then aims to extract $\mathcal{C}_{\{1:M\}}$ from these $0.5M(M - 1)$ sub-cluster groups $\mathcal{C}_{\{m,n\}}$ using an efficient search technique. This is achieved by first selecting the sub-cluster group

$$\mathcal{C}_{\{m_s,n_s\}} = \min_{c_{mn}} \left\{ \mathcal{C}_{\{1,2\}} \mathcal{C}_{\{1,3\}} \dots \mathcal{C}_{\{M-1,M\}} \right\}, \quad (7)$$

having the smallest c_{mn} as a reference group. This is equivalent to finding the two channels m_s and n_s having the smallest number of sub-clusters. Since near-common zeros must satisfy condition (i), GMC-ST begins its search from $\mathcal{C}_{\{m_s,n_s\}}$. For each row of $\mathcal{C}_{\{m_s,n_s\}}$, GMC-ST initializes a row vector $\mathbf{R} = [r(1) \dots r(M)]$ with only two non-empty elements $r(m_s) = p$ and $r(n_s) = q$ where p and q are two elements obtained from each row in $\mathcal{C}_{\{m_s,n_s\}}$. The next stage is to search, for row vector \mathbf{r} of \mathbf{R} , the remaining $M - 2$ empty elements. This search space is confined within $\{\mathcal{C}_{\{1,2\}} \mathcal{C}_{\{1,3\}} \dots \mathcal{C}_{\{M-1,M\}}\}$ excluding $\mathcal{C}_{\{m_s,n_s\}}$ since, from (6), only zeros within these groups are all within tolerance δ . If k elements are found, then \mathbf{R} is updated as

$$\tilde{\mathbf{R}} = \mathbf{1}_{k \times 1} \mathbf{r}, \quad \mathbf{R} = [\tilde{\mathbf{R}}^T \mathbf{R}^T]^T, \quad (8)$$

where $\mathbf{1}_{k \times 1} = [1 \dots 1]^T$. This implies that the zeros belong to k different clusters. The trimming process then ensures that all pairwise elements for each row in \mathbf{R} can be found within the search space $\{\mathcal{C}_{\{1,2\}} \mathcal{C}_{\{1,3\}} \dots \mathcal{C}_{\{M-1,M\}}\}$ in order to satisfy condition (ii). If this condition is violated, the entire row is deleted and the search-and-trim process is repeated until every element in each row of \mathbf{R} is found or all rows have been deleted.

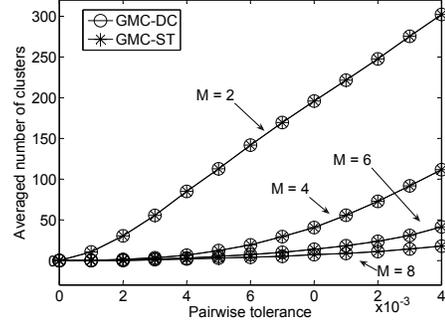


Fig. 3. Number of clusters found using the proposed algorithms against tolerance δ with different number of channels M .

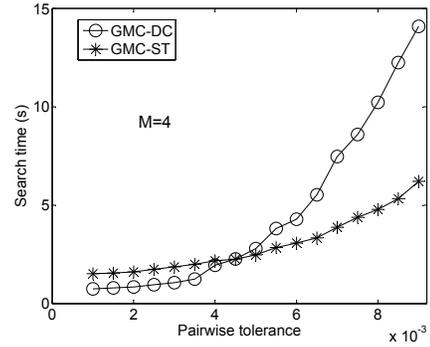


Fig. 4. Search time against tolerance δ using GMC-ST and GMC-DC for an example case of $M = 4$.

5. SIMULATIONS

We demonstrate the performance and illustrate the use of GMC-ST and GMC-DC using simulated acoustic impulse responses generated by the method of images [11] with a linear array of $M = 8$ microphones in a room of dimension $10 \times 10 \times 3$ m. The source is located 1 m in front of the microphone array with uniform microphone spacings of 8 cm. The sampling rate was 16 kHz with each channel impulse response having 512 coefficients. A set of 160 impulse responses are then generated by placing this source-sensor configuration in different positions of the room. Results are then obtained by spatial averaging over all positions in the room using the generated set of impulse responses.

We first analyze how the total number of clusters c_t vary with tolerance δ for different number of channels M . Clusters of near-common zeros are found using the proposed GMC-ST and GMC-DC algorithms. Figure 3 shows the number of clusters found in each M -channel system for various δ . It can be seen that c_t increases with δ as expected. More importantly, for each δ , the number of clusters reduces with increasing M . The number of clusters found using GMC-ST is the same as GMC-DC. For all cases of M , there are no clusters of zeros which are exactly common ($\delta = 0$).

We now compare, for an example case of $M = 4$, the efficiency of GMC-ST and GMC-DC in terms of their simulation times in MATLAB on a 2 GHz processor with 2 GB of memory. Figure 4 shows the time required to extract clusters $\mathcal{C}_{\{1:M\}}$ across various δ . The simulation times for both algorithms increase with δ since c_t increases with δ . The search time of GMC-ST increases less significantly compared to GMC-DC. This is because when c_t

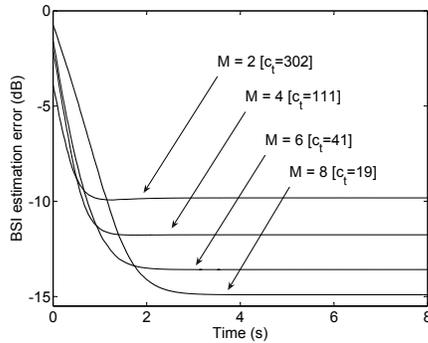


Fig. 5. Performance of NMCFLMS [4] against number of channels with different c_t for BSI application using WGN input and simulated impulse responses.

is large, the dimensions of $\mathcal{C}_{\{m,n\}}$ are large. Consequently, dissimilarity matrices $\mathbf{D}_{\{m,n\}}$ used for the next stage of processing in GMC-DC are large. These matrices propagate along the different stages of GMC-DC resulting in a significant increase in computational times. Although GMC-ST computes $\mathbf{D}_{\{m,n\}}$ a total of $M(M-1)/2$ times initially, it confines its search within the search space $\{\mathcal{C}_{\{1,2\}}, \mathcal{C}_{\{1,3\}}, \dots, \mathcal{C}_{\{M-1,M\}}\}$ and it does not compute $\mathbf{D}_{\{m,n\}}$ again. This reduces the computational times significantly compared to GMC-DC.

We illustrate application examples of how the proposed algorithms can be used to study the effect of near-common zeros for both BSI and channel equalization. We used the normalized multichannel frequency-domain least-mean-square (NMCFLMS) algorithm [4] with a step-size of 0.5 and a signal-to-noise ratio of 60 dB for BSI. A high SNR is used to avoid the misconvergence problem [2]. We employ the normalized projection misalignment (NPM) [12] to quantify the BSI estimation error. The number of clusters c_t found for $\delta = 6 \times 10^{-3}$ in each case of M is as shown in Fig. 5. It can be seen that the performance of NMCFLMS increases with M since c_t reduces with increasing M . The performance of the MINT algorithm [3], quantified using the signal-to-distortion ratio (SDR) [13] measure, for channel equalization across different number of channels is shown in Fig. 6. Its performance can be seen to increase with M since c_t reduces with increasing M . These results indicate that the performance of BSI and equalization algorithms improve with reducing number of near-common zeros brought about by introducing more channels to the system. In particular, the dependence of the performance of BSI and equalization algorithms can now be quantified using our near-common zero clustering algorithms.

6. CONCLUSION

We proposed two efficient algorithms for the clustering of near-common zeros in multichannel systems. These algorithms employ an efficient approach to compute the dissimilarity matrix, based on which the GMC-DC algorithm extracts clusters using a binary tree approach whereas the GMC-ST algorithm concentrates on searching solutions within sub-cluster groups. The GMC-ST algorithm has computational times of the order 5 s on a typical MATLAB implementation for a four-channel system with 512 coefficients per channel. We showed how the proposed algorithms can be applied for the study of BSI and channel equalization algorithms in the presence of near-common zeros.

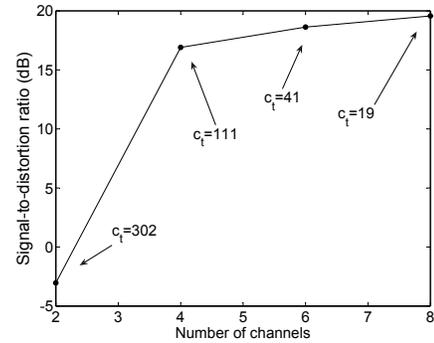


Fig. 6. Performance of the MINT algorithm [3] for channel equalization against number of channels with different c_t using simulated impulse responses.

7. REFERENCES

- [1] Y. Huang, J. Benesty, and J. Chen, "A blind channel identification-based two-stage approach to separation and dereverberation of speech signals in a reverberant environment," *IEEE Trans. Speech and Audio Process.*, vol. 13, no. 5, pp. 882–895, Sept. 2005.
- [2] R. Ahmad, A. Khong, and P. Naylor, "Proportionate frequency domain adaptive algorithms for blind channel identification," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, May 2006.
- [3] M. Miyoshi and Y. Kaneda, "Inverse filtering of room acoustics," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, no. 2, pp. 145–152, Feb. 1988.
- [4] Y. Huang and J. Benesty, "A class of frequency-domain adaptive approaches to blind multichannel identification," *IEEE Trans. Signal Process.*, vol. 51, no. 1, pp. 11–24, Jan. 2003.
- [5] N. D. Gaubitch, J. Benesty, and P. A. Naylor, "Adaptive common root estimation and the common zeros problem in blind channel identification," in *Proc. European Signal Process. Conf. (EUSIPCO)*, Sept. 2005.
- [6] X. S. Lin, N. D. Gaubitch, and P. A. Naylor, "Two-stage blind identification of SIMO systems with common zeros," in *Proc. European Signal Process. Conf. (EUSIPCO)*, Sept. 2006.
- [7] I. Fijalkow, "Multichannel equalization lower bound: a function of channel noise and disparity," in *proc. IEEE Signal Process. Workshop on Statistical Signal and Array Process. (SSAP)*, June 1996, pp. 344–347.
- [8] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.
- [9] G. Sitton, C. Burrus, J. Fox, and S. Treitel, "Factoring very-high-degree polynomials," *IEEE Signal Processing Magazine*, vol. 20, no. 6, pp. 27–42, Nov. 2003.
- [10] C. P. Hughes and A. Nikeghbali, "The zeros of random polynomials cluster uniformly near the unit circle," *ArXiv Mathematics e-prints*, June 2004.
- [11] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small room acoustics," *J. Acoust. Soc. Am.*, vol. 65, no. 4, pp. 943–950, April 1979.
- [12] D. Morgan, J. Benesty, and M. M. Sondhi, "On the evaluation of estimated impulse responses," *IEEE Signal Process. Letters*, vol. 5, no. 7, pp. 174–176, July 1998.
- [13] T. Hikichi, M. Delcroix, and M. Miyoshi, "Inverse filtering for speech dereverberation less sensitive to noise and room transfer function fluctuations," *EURASIP J. Applied Signal Process.*, vol. 2007, pp. Article ID 34 013, 12 pages, 2007.