Parametrization of Linear Systems Using Diffusion Kernels

Ronen Talmon, *Member, IEEE*, Dan Kushnir, Ronald R. Coifman, Israel Cohen, *Senior Member, IEEE*, and Sharon Gannot, *Senior Member, IEEE*

***Abstract*—Modeling natural and artificial systems has played a key role in various applications and has long been a task that has drawn enormous efforts. In this work, instead of exploring predefined models, we aim to identify implicitly the system degrees of freedom. This approach circumvents the dependency of a specific predefined model for a specific task or system and enables a generic data-driven method to characterize a system based solely on its output observations. We claim that each system can be viewed as a *black box* controlled by several independent parameters. Moreover, we assume that the perceptual characterization of the system output is determined by these independent parameters. Consequently, by recovering the independent controlling parameters, we find in fact a generic model for the system. In this work, we propose a supervised algorithm to recover the controlling parameters of natural and artificial linear systems. The proposed algorithm relies on nonlinear independent component analysis using diffusion kernels and spectral analysis. Employment of the proposed algorithm on both synthetic and practical examples has shown accurate recovery of parameters.**

***Index Terms*—Kernel, linear systems, modeling, multidimensional signal processing, non-parametric estimation, nonlinear dynamical systems, system identification.**

I. Introduction

MODELING natural and artificial systems has played a key role in various applications and has long been a task that has drawn enormous efforts. Usually a predefined model is developed for every type of task or system. Then the parameters of that model are estimated based on observations of the system output. In this work, we take a different approach. Instead of exploring predefined models, we aim to identify implicitly the system degrees of freedom or modes of variability. This approach enables to capture the intrinsic geometric structure of the system. Moreover, it circumvents the dependency of a specific predefined model for a specific task or system and provides a generic data-driven method to characterize a system based solely on its output observations. We claim that each system can be viewed as a *black box* controlled by several independent parameters. Furthermore, we assume that the perceptual characterization of the system output is determined by these independent parameters. Consequently, by recovering the independent controlling parameters, we find in fact a generic modeling for the system. Thus, in this work, we aim to recover the controlling parameters of natural and artificial systems.

Musical instruments are examples of such systems, as each musical instrument is controlled by several independent parameters. For example, a flute is controlled by covering its holes. Formally, the parameter space can be written as a d-dimensional binary space $\{0,1\}^d$, assuming the flute has d holes and each hole can be either open or covered. An important observation is that the output signal of the flute depends on the blow of air (the input signal) and the covering of the holes. However, the audible music, or the music tones, depends only on the covering of the holes. In other words, the played music depends solely on a finite set of the instrument's controlling parameters. Another example worth mentioning is a violin. Violin music is determined by the length of the strings. We note that unlike the controlling parameters of the flute, the parameter space of the violin is continuous, and can be written as $[0,\ell]^d$, assuming d strings of length ℓ. In both examples, by recovering the independent controlling parameters of the musical instrument, we may naturally characterize the music and identify the played tones.

Recently, Singer and Coifman [1] have proposed a nonlinear independent component analysis (ICA) method based on diffusion kernels [2]–[7]. They assume that the observable data is a nonlinear mapping of few independent parameters. Moreover, the parameters are assumed to realize a specific variability scheme, described by an Itô process [8], [9]. Based on estimation of the local distortions of the observations, an intrinsic metric is computed. This metric is invariant to the nonlinear mapping and conveys the distance between the parameters. Using this intrinsic metric, a kernel between the observations is computed, and a spectral ICA [10] is employed. The obtained spectral decomposition is used to build an inverse mapping of the observable data into the parametric space.

The spectral embedding proposed in [1] is computed for a given set of observations. However, in practice, and specifically in supervised learning tasks, not all the data is available. Therefore, various extension methods for the spectral decomposition have been explored [4], [11]–[14]. Unfortunately, none of these methods can be naturally employed in [1] since the metric used in the kernel relies on estimates of the local distortions of the parameters in the observable space, which are unavailable. Kushnir *et al.* [15] extend [1] and propose an efficient extendable spectral ICA algorithm. The authors propose a different intrinsic metric between the observations, which depends only on estimates of the local distortions of just a few reference points.

Manuscript received May 20, 2011; revised September 21, 2011; accepted November 10, 2011. Date of publication December 02, 2011; date of current version February 10, 2012. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Z. Jane Wang. This research was supported by the Israel Science Foundation (Grant 1130/11).

R. Talmon, D. Kushnir, and R. R. Coifman are with the Department of Mathematics, Yale University, New Haven 06520 CT USA (e-mail: ronen.talmon@yale.edu; dan.kushnir@yale.edu; ronald.coifman@yale.edu).

I. Cohen is with the Department of Electrical Engineering, Technion—Israel Institute of Technology, Haifa 32000, Israel (e-mail: icohen@ee.technion.ac.il).

S. Gannot is with the Faculty of Engineering, Bar-Ilan University, Ramat-Gan, 52900, Israel (e-mail: Sharon.Gannot@biu.ac.il).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2011.2177973

The ability to extend the embedding provides an efficient recovering of the independent parameters of observations, which are not available in advance.

In this work, we exploit the nonlinear ICA method to recover the independent parameters of systems. The main difference and challenge is that the output of a system heavily depends on the excitation signal. Thus, it is not solely determined by the controlling parameters, whereas the nonlinear maps discussed in [1] and [15] are functions of the parameters. For that reason, we restrict the scope of this work and consider only *linear* systems and propose a spectral algorithm based on [1] and [15] to recover the independent parameters of a system. Based on Fourier theory, we show that the problem of recovering the parameters of a linear system can be formulated as a problem of recovering the parameters of a nonlinear map, which is independent of the excitation signal. Consequently, we are able to present the recovering algorithm as an application of the algorithm presented in [15]. This algorithm is data-driven and not specifically tailored for a certain task. These attractive features can make it useful in the design, control, and calibration of a variety of systems. We employ the algorithm on both synthetic and practical examples. First, we show that the proposed method can accurately recover the poles of an autoregressive (AR) process. Next, we utilize the algorithm to retrieve the controlling parameters of acoustic channels in practical setups. It is worthwhile noting that acoustic channels are known to be highly difficult to model and acquire and play a key role in developing audio processing applications, e.g., [16]–[23]. In [24], we applied the proposed algorithm to the problem of source localization in a reverberant room using measurements from a single microphone. Experimental results in a real reverberant environment demonstrated accurate recovery of the source location.

This paper is organized as follows. In Section II, we formulate the problem. In Section III, we present the computation of a diffusion kernel. In Section IV, the proposed algorithm for recovering the independent parameters is presented, including a synthetic example. Finally, the application to acoustic channels and experimental results are shown in Section V.

II. Problem Formulation

Throughout this paper, vectors are denoted by bold small letters and matrices are denoted by bold capital letters. In addition, elements in vectors and matrices are written with a superscript index in parentheses, e.g., the ith element of a vector $\mathbf{a}$ is expressed as $\mathbf{a}^{(i)}$.

Let $\boldsymbol{\theta} \in \mathbb{R}^d$ denote a vector of parameters that control a particular natural or artificial system of interest. We follow Singer and Coifman [1], and Singer *et al.* [25], and assume that the controlling parameters evolve in time according to two evolution regimes: 1) a small fluctuations regime, representing fast natural changes of the system and 2) perceptual slow system variations. In the violin example mentioned in the Introduction, a perceptual system variation corresponds to setting different lengths of violin strings. On the other hand, small fluctuations may correspond to different finger placements on the strings aiming to produce the same tone. As in [1], we model these propagation regimes by stochastic Itô processes in order to obtain a traceable derivation.

We restrict the scope of the work and consider only linear systems. Let x be a real-valued input signal, $h_{\boldsymbol{\theta}}$ a real-valued impulse response of a linear system, which varies in time as a result of the time varying controlling parameter $\boldsymbol{\theta}$, and y the corresponding observable output signal. We assume that the input signal x is a zero-mean, wide-sense stationary (WSS) process. In practice, we only require that the input signal is a quasi-stationary process (i.e., WSS process in short time intervals), which is a much weaker assumption that holds for many natural signals such as speech and music.

We observe the output signal in short-term intervals of length N. We assume the interval length is sufficiently short, such that in each interval, the linear system can be considered time-invariant. Consequently, the controlling parameter of the system in each interval is assumed to be constant. In the flute example, each time interval may correspond to a different flute tone, which is configured by a different cover of the flute's holes. According to our assumption, the finger placement on the flute does not vary during the entire short-term interval. We note that we discard intervals with varying parameters as it exceeds the scope of this paper and intend to address it in future work.

Let M be the number of time intervals, and let $\Theta = \{\boldsymbol{\theta_i}\}_{i=1}^{M} \subset \mathbb{R}^d$ be the set of the controlling parameters, such that $\boldsymbol{\theta}_i$ is the parameter vector controlling the system in the ith interval. Let x_i and y_i denote the corresponding input and output signals in time interval i. Thus, the relation between x_i and y_i is expressed using linear convolution, denoted by $*$, as

$$y_i(t) = h_{\boldsymbol{\theta}_i}(t) * x_i(t) = \sum_{s=-\infty}^{\infty} h_{\boldsymbol{\theta}_i}(s) x_i(t-s)$$

where t and s are discrete time indices.

We assume m short-term intervals are available beforehand. Let $\bar{\Theta}$ denote a subset of m training parameters corresponding to the available intervals. Each training parameter $\bar{\boldsymbol{\theta}}_i \in \bar{\Theta}$ is recovered L times from different measurements. Unfortunately, in practice we cannot repeat the measurement with exactly the same parameter. Thus, for each training parameter $\bar{\boldsymbol{\theta}}_i$, we have a set of additional L intervals of the measured signal with corresponding parameters $\{\boldsymbol{\theta}_{i_j}\}_{j=1}^{L}$.

With respect to the proposed temporal evolution model of the controlling parameters, a pair of parameters $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$, in time intervals i and j, convey perceptually different system configurations, e.g., a different cover of the flute holes producing different tones. On the other hand, for each training interval i, the parameters $\{\boldsymbol{\theta}_{i_j}\}$ in the additional intervals are seen as small perturbations of $\bar{\boldsymbol{\theta}}_i$, e.g., different finger placements producing the same tone. See Appendix I for mathematical details of the temporal evolution of the parameters where we represent the slow and rapid variation regimes as drift and noise coefficients of a stochastic Itô process. The Itô process modeling provides a unique solution to the generally ill-posed nonlinear ICA problem. In addition, it enables to compute the local Jacobian-based distortion metric induced by the nonlinear transformation that maps the parameter space into the observable space, as described in Section III.

Our goal in this work is to recover the inaccessible set of controlling parameters Θ given the output signal observations. In addition, we assume m training observations and their corresponding parameters are given beforehand. These training samples are utilized for the calibration and training of the proposed recovering algorithm.

III. Diffusion Kernel

In this section we construct an anisotropic diffusion kernel. We utilize an approximation of the Euclidean distance in the parametric space (i.e., the controlling parameters domain) [1], [15] and build a kernel between the given observations of the output signal of the linear system of interest.

A. The Observations and Covariance Matrices

Let $c_y(\tau)$ denote the covariance function of the output signal $y(t)$, which is defined as [26]

$$c_y(\tau) = \mathbb{E}\left[y(t)y(t+\tau)\right] = h_{\boldsymbol{\theta}}(\tau) * h_{\boldsymbol{\theta}}(-\tau) * c_x(\tau) \tag{1}$$

where $\mathbb{E}[\cdot]$ denotes mathematical expectation, and $c_x(\tau)$ is the covariance of the input signal $x(t)$. Since x is a WSS process, $c_x(\tau)$ is time invariant, and therefore (1) implies that the time variations of $c_y(\tau)$ depend solely on the evolution of the controlling parameters $\boldsymbol{\theta}$ of the linear system. Thus, we obtain a representation of the observable output signal y as a function of the (dynamics of the) controlling parameters $\boldsymbol{\theta}$. It is worthwhile noting, that the same result could be obtained by observing the short-term power spectral density (PSD) of the output signal. For simplicity, we prefer to observe the second order statistics of the signal directly, conveyed by the covariance of the observable signal in the time domain, rather than to use the ("predefined") Fourier transform.

Let c denote the (nonlinear) mapping of the parameter vectors $\boldsymbol{\theta} \in \mathbb{R}^d$ to the first D covariance function elements of the output signal, given by

$$\mathbf{c} = c\left(\boldsymbol{\theta}\right) \tag{2}$$

where $\mathbf{c} \in \mathbb{R}^D$ is a vector of length D consisting of the covariance function elements, i.e.,

$$\mathbf{c}^{(j)} = c_y(j) = \mathbb{E}\left[y(t)y(t+j)\right]$$

for $j = 1, \ldots, D$.

In the remainder of this paper, we view the covariance function elements of the linear system output as *observations*. Moreover, these observations are interpreted as (nonlinear) mappings of the controlling parameters via the function c. In practice, the covariance elements are not available. However, they can be estimated given the output signal as an empiric average of the cross-multiplication of the output signal in each time interval separately. In each interval i, according to (2), we calculate D elements of the corresponding covariance function $\mathbf{c}_i = c(\boldsymbol{\theta}_i)$ based on the output signal y_i. Let $\Gamma = \{\mathbf{c}_i\}_{i=1}^M$ denote the set of observations, and let $\bar{\Gamma} = \{\bar{\mathbf{c}}_i\}_{i=1}^m$ denote the subset of observations corresponding to the training parameters in $\bar{\Theta}$. Using this notation we can define $c : \Theta \rightarrow \Gamma$ to be the nonlinear map between an unknown parametric manifold $\mathcal{M} \subseteq \Theta \subset \mathbb{R}^d$ and a corresponding observed data set $\Gamma \subset \mathbb{R}^D$. This presentation enables to relax our problem setting to the proposed setting in [15].

We illustrate the settings by observing the following autoregressive (AR) process of order 1

$$y(t) = x(t) - \theta y(t-1)$$

where $x(t)$ is a zero-mean white noise with σ_x^2 variance, and $0 < |\theta| < 1$ is the AR coefficient. Clearly, in this example, the system is controlled by a single parameter $\theta \in \mathbb{R}$. However, observing the output signal $y(t)$ in the time domain heavily depends on the random white noise input $x(n)$. Consequently, the evolution of the controlling parameter θ may be weakly emerged in $y(t)$, and hence hard to recover.

Fortunately, we are able to represent the AR process using convolution with the following infinite AR impulse response

$$h_{\boldsymbol{\theta}}(t) = \theta^t,\ t = 0, 1, \ldots.$$

This impulse response demonstrates a nonlinear dependency between the system and the controlling parameter θ. The corresponding covariance function of the AR process is given by

$$c_y(\tau) = \sigma_x^2 \frac{\theta^{-|\tau|}}{1-\theta^2} \tag{3}$$

where $|\cdot|$ denotes absolute value. In (3) we represent the covariance function of the observable signal as a (nonlinear) function of the controlling parameter θ. We note that θ should satisfy $0 < |\theta| < 1$ in order to get a stable impulse response.

In this work, given the output signal measurements, we estimate their covariance function elements. These elements are viewed as observations of the nonlinear mapping (3). Thus, the goal in this work is to recover the controlling parameter θ from such observations.

Let $\boldsymbol{\Sigma}(\mathbf{c})$ denote the covariance matrix of size $D \times D$ of the observation $\mathbf{c}$, defined by

$$\boldsymbol{\Sigma}^{(jk)}(\mathbf{c}) = \mathrm{Cov}\left(\mathbf{c}^{(j)}, \mathbf{c}^{(k)}\right). \tag{4}$$

It can be shown (see Appendix I for the derivation and mathematical details) that the covariance matrix can be expressed as

$$\boldsymbol{\Sigma}(\mathbf{c}) = \mathbf{J}(\mathbf{c})\mathbf{J}^T(\mathbf{c}) \tag{5}$$

where $\mathbf{J}$ is the Jacobian matrix of the function c, whose elements are given by $\mathbf{J}^{(ji)} = c_i^j$, where c_i^j are first-order partial derivatives of the jth coordinate of the mapping c with respect to the parameter $\boldsymbol{\theta}^{(i)}$, and $(\cdot)^T$ denotes matrix transpose. The Jacobian of the function c is of key importance to the proposed algorithm as described in Section IV but unfortunately is unavailable. However, (5) allows to represent $\mathbf{J}(\mathbf{c})\mathbf{J}^T(\mathbf{c})$ via the *accessible* covariance matrix of the observations.

We are able to compute the covariance matrices of only the training observations. Given measurements of the output signal corresponding to the perturbations of the training samples $\{\boldsymbol{\theta}_{i_j}\}$, we compute their corresponding observations $\{\mathbf{c}_{i_j}\}$. Now, based on the "cloud" of L observations $\{\mathbf{c}_{i_j}\}$, we estimate the local covariance matrix $\boldsymbol{\Sigma}(\bar{\mathbf{c}}_i)$ for each training observation empirically via

$$\hat{\boldsymbol{\Sigma}}(\bar{\mathbf{c}}_i) = \frac{1}{L}\sum_{j=1}^{L}(\mathbf{c}_{i_j} - \boldsymbol{\mu}_i)(\mathbf{c}_{i_j} - \boldsymbol{\mu}_i)^T \tag{6}$$

where $\boldsymbol{\mu}_i = \frac{1}{L}\sum_{j=1}^{L}\mathbf{c}_{i_j}$.

B. Computation of the Anisotropic Kernel

The proposed parametrization method is based on the computation of an anisotropic diffusion kernel. In order to obtain recovery of the independent parameters, the computed kernel is based on the Euclidean distance between the parameters. Unfortunately, the parameters are available only via the nonlinear

observations. In [1], Singer and Coifman showed that the Euclidean distance between two samples in the parametric space can be approximated by the observations. In this work, we adopt a similar strategy to approximate the Euclidean distance between the parameters, as proposed by Kushnir *et al.* in [15], which also enables a natural extension.

Let $\boldsymbol{\theta}_j$, $\boldsymbol{\theta}_k$ be two parameter samples, i.e., two system configurations in the parametric space. According to previous notation, we observe their nonlinear mapping via $c : \mathbb{R}^d \rightarrow \mathbb{R}^D$. Let $\mathbf{c}_j = c(\boldsymbol{\theta}_j)$ and $\mathbf{c}_k = c(\boldsymbol{\theta}_k)$ be the mapping of $\boldsymbol{\theta}_j$ and $\boldsymbol{\theta}_k$ into the observable space. It is shown in [15] that a second-order approximation of the squared Euclidean distance in the parametric space is given by (see Appendix II)

$$\|\boldsymbol{\theta}_j - \boldsymbol{\theta}_k\|^2 = 2(\mathbf{c}_j - \mathbf{c}_k)^T \left[\left(\mathbf{J}\mathbf{J}^T\right)(\mathbf{c}_j) + \left(\mathbf{J}\mathbf{J}^T\right)(\mathbf{c}_k)\right]^{-1} \times (\mathbf{c}_j - \mathbf{c}_k) + \mathcal{O}\left(\|\mathbf{c}_j - \mathbf{c}_k\|^4\right). \quad (7)$$

We note that the rank of $\left(\mathbf{J}\mathbf{J}^T\right)$ is d and therefore a pseudo-inverse should be computed.

We compute an $M \times m$ affinity matrix $\mathbf{A}$ between the observations in $\bar{\Gamma}$ and the observations in Γ. The affinity is based on a Gaussian kernel with a scale parameter ε and given by

$$\mathbf{A}^{(kj)} = \exp\left\{-\frac{\|\mathbf{J}^{-1}(\bar{\mathbf{c}}_j)(\bar{\mathbf{c}}_j - \mathbf{c}_k)\|^2}{\varepsilon}\right\}. \quad (8)$$

We observe the following $m \times m$ matrix

$$\mathbf{W} = \mathbf{S}^{\frac{-1}{2}}\mathbf{A}^T\mathbf{A}\mathbf{S}^{\frac{-1}{2}} \quad (9)$$

where $\mathbf{S}$ is a diagonal matrix containing the sum of $\mathbf{A}^T\mathbf{A}$ along columns, i.e., $\mathbf{S} = \mathrm{diag}\{\mathbf{A}^T\mathbf{A}\mathbf{1}\}$. It is shown [15] that $\mathbf{W}^{(kj)}$ corresponds to

$$\mathbf{W}^{(kj)} = \frac{\pi}{\sqrt{\det((\mathbf{J}^T\mathbf{J})(\tilde{\mathbf{c}}))}} \times \exp\left\{-\frac{(\bar{\mathbf{c}}_j - \bar{\mathbf{c}}_k)^T\left[\mathbf{J}\mathbf{J}^T(\bar{\mathbf{c}}_k) + \mathbf{J}\mathbf{J}^T(\bar{\mathbf{c}}_j)\right]^{-1}(\bar{\mathbf{c}}_j - \bar{\mathbf{c}}_k)}{\varepsilon}\right\} \quad (10)$$

for $\bar{\mathbf{c}}_j, \bar{\mathbf{c}}_k \in \bar{\Gamma}$ and $\tilde{\mathbf{c}} = \frac{(\bar{\mathbf{c}}_j + \bar{\mathbf{c}}_k)}{2}$, and $\det(\mathbf{X})$ denotes the determinant of the matrix $\mathbf{X}$. Based on [15, Lemma 3.3], $\mathbf{W}$ is a Gaussian kernel based on approximation of the Euclidean distance between the training samples in the parametric space (7). Moreover, (9) implies that the affinity between the training samples in $\mathbf{W}$ is conveyed via the affinity between just the training samples and all the other samples in $\mathbf{A}$. In other words, two training samples are similar if they are "seen" the same way by the rest of the samples. This property provides the possibility to naturally extend the kernel to new samples as shown in Section IV.

It is worthwhile to emphasize two points for further insight on the equivalency between the direct computation of the kernel between the training samples in (10) and the computation via new samples in (9). First, [15, Theorem 3.2] claims that the equivalence between the two alternative computations hold only for sufficient number of new (randomly distributed) samples. Second, the role of the Gaussian (or exponentially decaying) kernel is to discard implicitly from the distance approximation in (9) new samples that do not lie between the training samples. See Appendix III for a simulation of the kernel computation in a simple 2-D example.

Substituting (5) and (6) into (10) yields

$$\mathbf{W}^{(kj)} = \frac{\pi}{\sqrt{\det\left(\hat{\boldsymbol{\Sigma}}(\tilde{\mathbf{c}})\right)}} \times \exp\left\{-\frac{(\bar{\mathbf{c}}_j - \bar{\mathbf{c}}_k)^T\left[\hat{\boldsymbol{\Sigma}}(\bar{\mathbf{c}}_k) + \hat{\boldsymbol{\Sigma}}(\bar{\mathbf{c}}_j)\right]^{-1}(\bar{\mathbf{c}}_j - \bar{\mathbf{c}}_k)}{\varepsilon}\right\}.$$

This kernel constitutes the key point of the algorithm. The affinity between the observations is based on an approximation of the Euclidean distance between the corresponding underlying parameters.

IV. From the Observable Data to the Linear System Parameters

In this section, we propose a supervised algorithm for the reparametrization of linear systems based on the algorithm presented in [15]. The recovery of the controlling parameters of the system relies on the kernel computed in Section III. Based on the kernel eigendecomposition, the observations are mapped into a new domain, which corresponds to the parametric domain up to a monotonic distortion.

A. Inverse Mapping

Let $\mathbf{L}$ be the normalized graph-Laplacian [27] defined as

$$\mathbf{L} = \mathbf{D}^{-1}\mathbf{W} - \mathbf{I}$$

where $\mathbf{D}$ is a diagonal matrix with $\mathbf{D}^{(ii)} = \sum_{j=1}^{d}\mathbf{W}^{(ij)}$. It can be shown that $\mathbf{L}$ converges to the backward Fokker–Planck operator $\mathcal{L}$ on the *parametric manifold* [6], [28]

$$\mathcal{L} = \Delta - \nabla U \cdot \nabla \quad (11)$$

where U is the density potential $U = -2\log p_{\bar{\Theta}}$, and Δ denotes the Laplacian operator. Assuming the set $\bar{\Theta}$ is a uniform sampling of the underlying parameter θ yields constant potential and $\nabla U = 0$. By (11), we obtain a convergence of the graph-Laplacian to the Laplace–Beltrami operator $\mathbf{L} \rightarrow \Delta$.

There exist eigenfunctions $\{\varphi_i\}$ of $\mathcal{L}$ that are monotonic functions of the parameters $\boldsymbol{\theta}$ as guaranteed by the Sturm–Liouville theory. These eigenfunctions can be chosen as suggested in [10]. Thus, they can be used to represent the data in terms of its independent controlling parameters. For simplicity, we assume that these eigenvectors correspond to the largest eigenvalues. Let $\Phi_d : \bar{\Gamma} \rightarrow \mathbb{R}^d$ be a map from the observations to the space spanned by d eigenfunctions of $\mathcal{L}$, given by

$$\Phi_d : \bar{\mathbf{c}}_i \rightarrow [\varphi_1(\bar{\mathbf{c}}_i), \ldots, \varphi_d(\bar{\mathbf{c}}_i)]^T. \quad (12)$$

Ideally, the map Φ_d can be seen as the inverse map of the nonlinear function c up to a monotonic distortion. Unfortunately, in practice we have the eigenvectors of $\mathbf{L}$, which only approximate the eigenfunctions $\{\varphi_i\}$ of $\mathcal{L}$.

B. Restriction and Extension Operators

Let $\tilde{\mathbf{A}}$ be a normalized affinity matrix $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{S}^{\frac{-1}{2}}$, and let $\{\lambda_j\}_{j=1}^m$, $\{\boldsymbol{\varphi}_j\}_{j=1}^m$ and $\{\boldsymbol{\psi}_j\}_{j=1}^M$ be the singular values and the left and right singular vectors of the $M \times m$ matrix $\tilde{\mathbf{A}}$, where the singular values are denoted in descending order. The vectors $\{\boldsymbol{\varphi}_j\}_{j=1}^m \in \mathbb{R}^m$ and $\{\boldsymbol{\psi}_j\}_{j=1}^M \in \mathbb{R}^M$ form an orthonormal basis

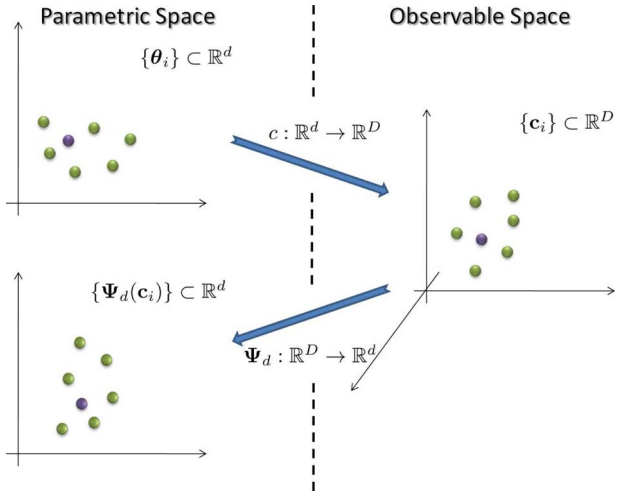

Fig. 1. A diagram of the parametric and observable space. We illustrate the set of samples and the mappings.

of $\mathbb{R}^m$ and $\mathbb{R}^M$ respectively. In addition, we have that $\{\lambda_j^2\}$ and $\{\boldsymbol{\varphi}_j\}$ are the eigenvalues and eigenvectors of $\mathbf{W} = \tilde{\mathbf{A}}^T\tilde{\mathbf{A}}$. It implies that $\{\boldsymbol{\varphi}_j\}$ establish the reparametrization of the training observations in $\bar{\Gamma}$. Accordingly, let $\boldsymbol{\Phi}_d$ be an embedding of the training observations into the space spanned by the d eigenvectors, given by

$$\boldsymbol{\Phi}_d : \bar{\mathbf{c}}_i \longrightarrow [\boldsymbol{\varphi}_1(\bar{\mathbf{c}}_i), \ldots, \boldsymbol{\varphi}_d(\bar{\mathbf{c}}_i)]^T. \tag{13}$$

Clearly, (13) can be considered an approximation of the map (11). Consequently, it may represent the data in terms of its independent controlling parameters.

On the other hand, $\{\boldsymbol{\psi}_j\}$ are the eigenvectors of $\tilde{\mathbf{A}}\tilde{\mathbf{A}}^T$, which is an $M \times M$ affinity matrix between observations in Γ. To circumvent additional spectral decomposition, the extended eigenvectors can be computed via

$$\boldsymbol{\psi}_j = \frac{1}{\lambda_j}\tilde{\mathbf{A}}\boldsymbol{\varphi}_j. \tag{14}$$

Reference [15, Lemma 4.1] implies that $\{\boldsymbol{\varphi}_j\}$ and $\{\boldsymbol{\psi}_j\}$ coincide on the training observations. It further implies that $\{\boldsymbol{\psi}_j\}$ are the extension of $\{\boldsymbol{\varphi}_j\}$ to new observations according to a regularized mean square error criterion. In addition, since $\{\boldsymbol{\psi}_j\}$ are obtained by the SVD, the extension preserves orthonormality. As a consequence, we conclude that $\{\boldsymbol{\psi}_j\}$ provide reparametrization of the observations. It is worthwhile noting that in (14) the extended eigenvector $\boldsymbol{\psi}_j$ is given by linear combinations of $\{\boldsymbol{\varphi}_j\}$. Thus, in the case of a large training set and a small kernel scale, the extended eigenvectors may accurately approximate the eigenfunctions of $\mathcal{L}$ as well.

Let $\boldsymbol{\Psi}_d$ be an embedding of the observations onto the eigenvectors of $\mathbf{W}$, given by

$$\boldsymbol{\Psi}_d : \mathbf{c}_i \longrightarrow [\boldsymbol{\psi}_1(\mathbf{c}_i), \ldots, \boldsymbol{\psi}_d(\mathbf{c}_i)]^T. \tag{15}$$

Consequently, the map $\boldsymbol{\Psi}_d$ approximates the independent parameters of the linear system corresponding to the observations up to monotonic distortion. See the illustration of the mapping in Fig. 1.

In order to obtain an estimate of the parameters, we interpolate the training samples according to distances in the embedded space. Let $\mathcal{N}_i$ consist of the k-nearest training embedded samples $\{\boldsymbol{\Psi}_d(\bar{\mathbf{c}}_j)\}$ of $\boldsymbol{\Psi}_d(\mathbf{c}_i)$ with the Euclidean metric, and let $\{\gamma_j\}$ be interpolation coefficients between $\{\boldsymbol{\Psi}_d(\bar{\mathbf{c}}_j)\} \in \mathcal{N}_i$ and $\boldsymbol{\Psi}_d(\mathbf{c}_i)$, given by

$$\gamma_j(\mathbf{c}_i) = \frac{\exp\left(\frac{-\|\boldsymbol{\Psi}_d(\mathbf{c}_i)-\boldsymbol{\Psi}_d(\bar{\mathbf{c}}_j)\|^2}{\sigma_{\gamma_j}}\right)}{\sum_{\boldsymbol{\Psi}_d(\bar{\mathbf{c}}_k)\in\mathcal{N}_i}\exp\left(\frac{-\|\boldsymbol{\Psi}_d(\mathbf{c}_i)-\boldsymbol{\Psi}_d(\bar{\mathbf{c}}_k)\|^2}{\sigma_{\gamma_j}}\right)}$$

where σ_{γ_j} is set to the minimal distance between $\boldsymbol{\Psi}_d(\mathbf{c}_i)$ and its nearest neighbor. Thus, an estimate of the parameters is given by the following weighted sum of the training parameters

$$\hat{\boldsymbol{\theta}}_i = \sum_{j:\boldsymbol{\Psi}_d(\bar{\mathbf{c}}_j)\in\mathcal{N}_i}\gamma_j(\mathbf{c}_i)\bar{\boldsymbol{\theta}}_j. \tag{16}$$

Accordingly, let $\mathrm{err}_{\mathrm{prm}}$ denote the reparametrization error, defined by

$$\mathrm{err}_{\mathrm{prm}}(\mathbf{c}_i) = \|\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_i\|^2. \tag{17}$$

We note that in the case that the parameters of merely few training samples are available, we can use them to pinpoint the embedded samples $\{\boldsymbol{\Psi}_d(\bar{\mathbf{c}}_j)\}$ into the proper scale (from the scale of the eigenvectors to the scale of the parameters). For that matter, we replace $\bar{\boldsymbol{\theta}}_j$ with the rescaled $\boldsymbol{\Psi}_d(\bar{\mathbf{c}}_j)$ in the estimation (16).

C. Setting the Algorithm Parameters

We define an inverse mapping $\boldsymbol{\Phi}_d^{-1}$ from the parameter space to the observable space, which approximates the mapping c, as follows:

$$\boldsymbol{\Phi}_d^{-1}(\bar{\boldsymbol{\theta}}) = \sum_{i:\bar{\boldsymbol{\theta}}_i\in\mathcal{B}_{\bar{\boldsymbol{\theta}}}}\beta_i(\bar{\boldsymbol{\theta}})\bar{\mathbf{c}}_i \tag{18}$$

where $\mathcal{B}_{\bar{\boldsymbol{\theta}}}$ is a set of the neighbors of $\bar{\boldsymbol{\theta}}$, and β_i are interpolation coefficients which are given as

$$\beta_i(\bar{\boldsymbol{\theta}}) = \frac{\exp\left(\frac{-\|\bar{\boldsymbol{\theta}}-\bar{\boldsymbol{\theta}}_i\|^2}{\sigma_{\beta_i}}\right)}{\sum_{j:\bar{\boldsymbol{\theta}}_j\in\mathcal{B}_{\bar{\boldsymbol{\theta}}}}\exp\left(\frac{-\|\bar{\boldsymbol{\theta}}-\bar{\boldsymbol{\theta}}_j\|^2}{\sigma_{\beta_i}}\right)} \tag{19}$$

where σ_{β_i} is set to the minimal distance between $\bar{\boldsymbol{\theta}}_i$ and its nearest neighbor in the parametric space. In case the parameters of the training samples are unavailable, we can use the mapping $\boldsymbol{\Phi}_d(\bar{\mathbf{c}}_i)$ as approximation of $\bar{\boldsymbol{\theta}}_i$ in (19). Let $\mathrm{err}_{\mathrm{val}}(\bar{\mathbf{c}}_i)$ denote the following *validation error*:

$$\mathrm{err}_{\mathrm{val}}(\bar{\mathbf{c}}_i) = \|\bar{\mathbf{c}}_i - \boldsymbol{\Phi}_d^{-1}(\bar{\boldsymbol{\theta}}_i)\|^2 = \|c(\bar{\boldsymbol{\theta}}_i) - \boldsymbol{\Phi}_d^{-1}(\bar{\boldsymbol{\theta}}_i)\|^2 \tag{20}$$

which conveys the accuracy of $\boldsymbol{\Phi}_d^{-1}$ in estimating c.

The mean error of (20) is computed for all training samples. Then the algorithm parameters are set to minimize this error. A particular parameter of interest is the kernel scale ε. As discussed in [29] and [30], setting the scale conveys a tradeoff between integration of large number of samples (large scale), and locality (small scale). We note that this tradeoff emerged in our empirical testing. In [15], the authors define a more general map for every sample. However, in practice, we use this mapping for setting the parameters in a training stage, where only the training observations are available. Therefore, for this particular use, (18) is sufficient.

The complete description of the proposed method is summarized in Algorithm 1.

Algorithm 1: Reparametrization Algorithm

Training stage:

1) Obtain m intervals of the system output corresponding to (known) training samples of the controlling parameters $\{\bar{\boldsymbol{\theta}}_i\}$.
2) Calculate D elements of the covariance function of the measurements, which constitute m training observations $\{\bar{\mathbf{c}}_i\}$.
3) Given clouds of additional observations corresponding to perturbations of the training parameters, estimate local covariance matrices $\{\boldsymbol{\Sigma}(\bar{\mathbf{c}}_i)\}$ of the m training observations.
4) Compute the affinity matrix $\mathbf{W}$ according to (10), for an arbitrary kernel scale ε.
5) Employ eigenvalue decomposition of $\mathbf{W}$ and obtain the eigenvalues $\{\lambda_j^2\}$ and the eigenvectors $\{\boldsymbol{\varphi}_j\}$.
6) Construct the map $\boldsymbol{\Phi}_d$ according to (13) to obtain reparametrization of the independent controlling parameters of the training observations.
7) Construct the inverse map $\boldsymbol{\Phi}_d^{-1}$ according to (18).
8) Find the optimal kernel scale ε that minimizes (20), by repeating 4–7 for different scales.

Testing stage:

1) Given a set of new observations $\{\mathbf{c}_i\}$ corresponding to new controlling parameters, compute the normalized affinity matrix $\tilde{\mathbf{A}}$ according to (8).
2) Calculate $\boldsymbol{\psi}_j$ as a weighted combination of $\boldsymbol{\varphi}_j$ via (14).
3) Construct the map $\boldsymbol{\Psi}_d$ according to (15) to obtain reparametrization of the independent controlling parameters of the new observations.
4) Recover the independent parameters according to (16) and compute the mean reparametrization error (17).

D. Example: Autoregressive Model

In this section, we recover the parameters of an AR system. Consider the following AR process of order d

$$y(t) = x(t) - \sum_{l=1}^{d} \boldsymbol{\alpha}^{(l)} y(t-l) \tag{21}$$

where $x(t)$ is a zero-mean white noise with variance σ_x^2, and $\boldsymbol{\alpha}$ are the AR coefficients. Such an AR process is commonly used in many signal processing applications. In particular, it is widely used in modeling the human vocal tract in speech recognition tasks [31], [32].

An AR process can be viewed as a white noise going through a linear system, where the corresponding transfer function $H_{\boldsymbol{\theta}}(\omega)$ is given by

$$H_{\boldsymbol{\theta}}(\omega) = \frac{1}{1 - \sum_{l=1}^{d} \boldsymbol{\alpha}^{(l)} e^{-jl\omega}}. \tag{22}$$

Alternatively, we can express $H_{\boldsymbol{\theta}}(\omega)$ in a canonical form as

$$H_{\boldsymbol{\theta}}(\omega) = \frac{1}{\prod_{l=1}^{d} \left(1 - \boldsymbol{\theta}^{(l)} e^{-j\omega}\right)} \tag{23}$$

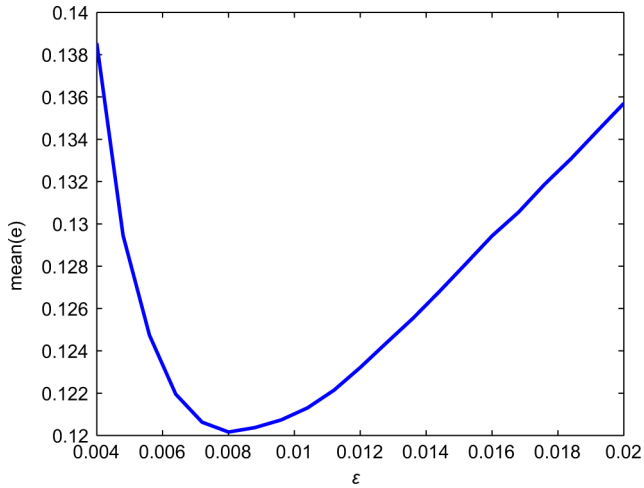

Fig. 2. The mean validation error obtained for all training samples as function of the kernel scale ε.

where $\boldsymbol{\theta}^{(l)}$ are the system poles. Consequently, according to (23), the system is controlled by d independent parameters $\boldsymbol{\theta} \in \mathbb{R}^d$.

Let $P_y(\omega)$ be the PSD of the AR process, which is given by

$$P_y(\omega) = \sigma_x^2 \prod_{l=1}^{d} \left|1 - \boldsymbol{\theta}^{(l)} e^{-j\omega}\right|^{-2}. \tag{24}$$

We observe in (24), that the PSD depends only on the controlling parameter $\boldsymbol{\theta}$. Consequently, the variations of the controlling parameters are conveyed by the PSD. Now, from (24), we can express the covariance function of the output signal as

$$c_y(\tau) = \mathcal{F}^{-1}\{P_y(\omega)\} \tag{25}$$

where $\mathcal{F}^{-1}$ denotes the inverse Fourier transform. For simplicity, we omit the explicit expression of the covariance function. In (25) we represent the covariance of the observable signal as a (nonlinear) function of the controlling parameter $\boldsymbol{\theta}$. We assume that the poles satisfy $0 < |\boldsymbol{\theta}^{(l)}| < 1$ to maintain system stability.

Next, we examine the ability of the proposed algorithm to recover the parameters of an AR system of order $d = 2$. For training, we randomly generate $m = 1000$ uniformly distributed training samples in a rectangular $\boldsymbol{\theta}_i \sim U[-0.8, -0.2] \times U[0.2, 0.8]$. Each realization represents a pair of AR poles, i.e., the controlling parameters of an AR system of order $d = 2$. Let $\bar{\Theta}$ denote the set of training parameters. For each realization $\bar{\boldsymbol{\theta}}_i$ of the 2-poles in $\bar{\Theta}$, we create 200 low variance Gaussian perturbations $\boldsymbol{\theta}_{i_j}$ to create a local "cloud" in the vicinity of $\bar{\boldsymbol{\theta}}_i$, such that

$$\boldsymbol{\theta}_{i_j} = \bar{\boldsymbol{\theta}}_i + \sqrt{dt} d\mathbf{w}$$

with $dt = 0.001$, and $d\mathbf{w}$ is a 2-D zero-mean unit-variance Gaussian noise.

The training parameters and their clouds are mapped to an observable space as follows. For each system $h_{\bar{\boldsymbol{\theta}}_i}$, we generate a white Gaussian excitation signal $\mathbf{x}_i$ of length $N = 16384$ and measure the corresponding output signal $\mathbf{y}_i$. It is worthwhile noting that this experiment was repeated with a uniformly distributed excitation signal, and similar results were obtained. Based on the measured output, estimates of $D = 8$ elements

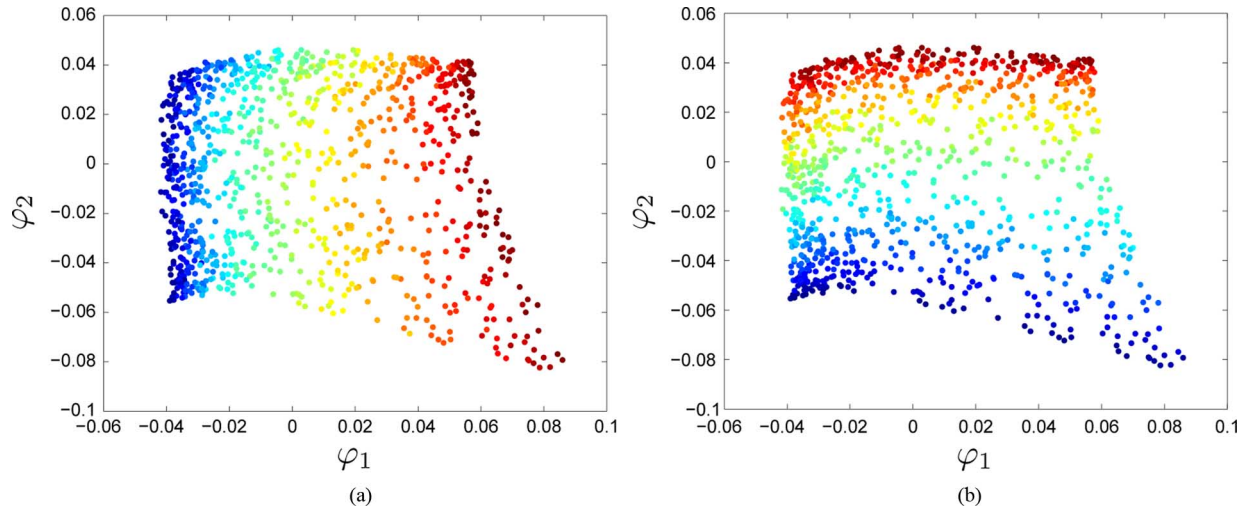

Fig. 3. Scatter plot of the embedded $m = 1000$ training samples: (a) Color coding according to the values of the first pole $\boldsymbol{\theta}^{(1)}$ and (b) color coding according to the values of the second pole $\boldsymbol{\theta}^{(2)}$.

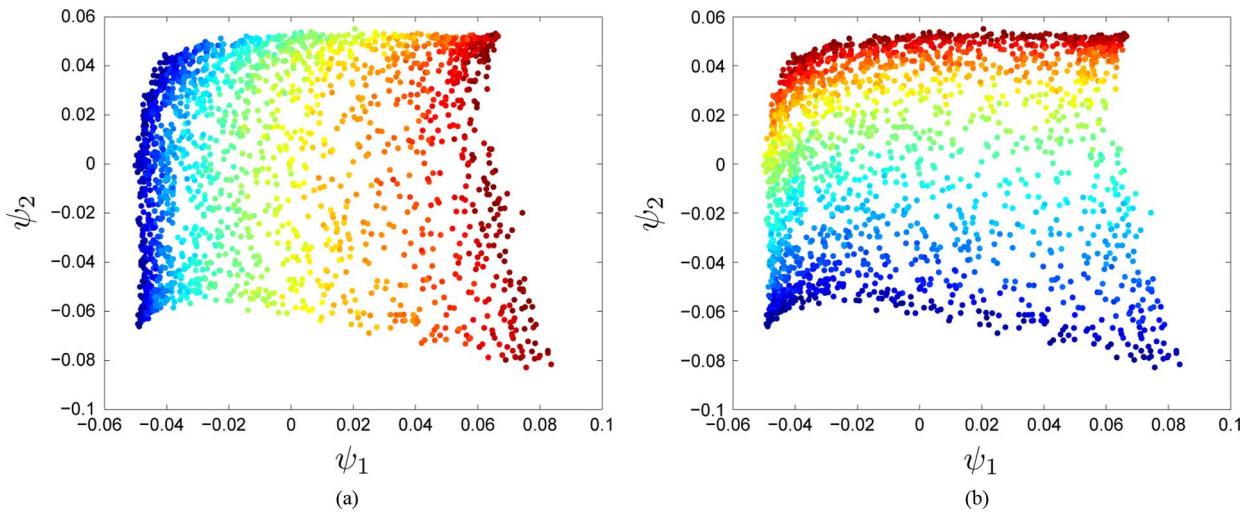

Fig. 4. Scatter plot of the embedded samples: (a) Color coding according to the values of the first pole $\boldsymbol{\theta}^{(1)}$ and (b) color coding according to the values of the second pole $\boldsymbol{\theta}^{(2)}$.

of the covariance function $\mathbf{c}_i$ are computed. Let $\bar{\Gamma} = \{\bar{\mathbf{c}}_i\}_{i=1}^m$ denote the set of observations corresponding to the training parameters. In addition, based on the observations $\{\mathbf{c}_{i_j}\}$ corresponding to the cloud of points around $\bar{\boldsymbol{\theta}}_i$, the covariance matrix of each training observation is computed as $\boldsymbol{\Sigma}(\bar{\mathbf{c}}_i)$ via (6). In summary, we have a set of m training parameters in a 2-D parametric space, and a corresponding set of m observations in an 8-D observable space. The observations are obtained via a nonlinear mapping $c : \mathbb{R}^2 \rightarrow \mathbb{R}^8$ of the controlling parameters, i.e., $\hat{\mathbf{c}}_i = c(\hat{\boldsymbol{\theta}}_i)$.

We follow steps 1–8 in Algorithm 1. Accordingly, we construct a 2-D embedding of $\bar{\Gamma}$ via

$$\boldsymbol{\Phi}_2 : \bar{\mathbf{c}}_i \rightarrow [\boldsymbol{\varphi}_1(\bar{\mathbf{c}}_i), \boldsymbol{\varphi}_2(\bar{\mathbf{c}}_i)]^T .$$

Next, we determine the proper kernel scaling ε. Fig. 2 shows the mean validation error (20) obtained by averaging over all the training samples as a function of the kernel scale ε. Accordingly, we choose the scale $\varepsilon = 0.008$, which minimizes the mean error. In addition, the existence of an optimal scale is derived from the tradeoff between a small scale for better locality and a large scale for better sample integration, which is evident from the curve.

Fig. 3 shows a scatter plot of the embedded training samples in $\mathbb{R}^2$ via $\boldsymbol{\Phi}_2$, where the color coding corresponds to the values of the parameters. We observe that an approximate rectangular shape is retrieved and that the coloring of the points is parallel to the axes. Hence, it implies that the embedding comprises the independent controlling parameters of the AR system. Moreover, $\boldsymbol{\varphi}_1(\bar{\mathbf{c}}_i)$ and $\boldsymbol{\varphi}_2(\bar{\mathbf{c}}_i)$ can be interpreted as a reparametrization of the pair of poles $\boldsymbol{\theta}_i^{(1)}$ and $\boldsymbol{\theta}_i^{(2)}$.

An additional 1000 samples are generated from the same distribution and mapped to the observable space as described above. Let Θ and Γ denote the sets of all $M = 2000$ samples in the parameter and observable spaces, respectively. We construct the matrix $\tilde{\mathbf{A}}$, which measures the affinity between the training samples and the additional samples. The extended eigenvectors $\{\boldsymbol{\psi}_j\}$ are calculated, which correspond to the right singular vectors of $\tilde{\mathbf{A}}$. We construct a map using the extended eigenvectors as

$$\boldsymbol{\Psi}_2 : \mathbf{c}_i \rightarrow [\boldsymbol{\psi}_1(\mathbf{c}_i), \boldsymbol{\psi}_2(\mathbf{c}_i)]^T .$$

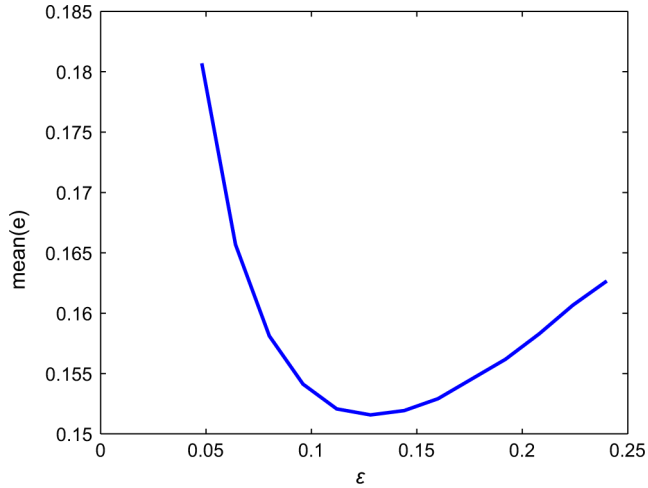

Fig. 5. The mean validation error obtained for all training samples as a function of the kernel scale ε, now with the fourth-order process.

In Fig. 4, we illustrate the embedding of the extended samples. Although the shape of the scatter plot is slightly deformed, the general rectangular shape is maintained. Moreover, the coloring of the samples implies that the embedding of the extended samples comprises the independent controlling parameters of the AR system as well. Based on the embedding, the interpolation coefficients γ_j are computed. Since the embedding organizes the points according to the values of the underlying parameters, the interpolation coefficients are comprised of a notion of affinity between the parameters. Using the interpolation coefficients, we estimate the parameters via (16). To demonstrate the ability to recover the parameters from the obtained reparametrization, we compute the parametrization error (17). The obtained mean error of the extended samples is $\mathrm{err}_{\mathrm{prm}} = 0.0376$.

We further illustrate the ability to recover the *independent* controlling parameters of the system. We extend the AR model by adding a pair of poles $\boldsymbol{\theta}^{(1)}\boldsymbol{\theta}^{(2)}$ and $\boldsymbol{\theta}^{(1)}\boldsymbol{\theta}^{(2)}$. Thus, we find that the fourth-order AR system is still controlled by just two independent parameters (the additional two poles are determined by $\boldsymbol{\theta}^{(1)}$ and $\boldsymbol{\theta}^{(2)}$). We note that model-based algorithms, such as the widely used Levinson–Durbin algorithm [26], provide estimates of the four AR coefficients but cannot detect the actual degrees of freedom.

In Fig. 5, we show the mean validation error (20) as a function of the kernel scale ε. Accordingly, the kernel scale for this experiment is set to $\varepsilon = 0.13$. A much larger scale is used in this experiment compared to the previous one, which results in integration of more samples. In addition, the mean error values in this experiment are higher than the mean error values obtained in Fig. 2.

Figs. 6 and 7 show the same trends as Figs. 3 and 4. We observe a rectangular shape and color lines parallel to the axes. Consequently, we obtain that the map $\boldsymbol{\Psi}_2$ captures the actual degrees of freedom, i.e., the two independent poles of the system. In this case, recovering the parameters yields mean error (17) $\mathrm{err}_{\mathrm{prm}} = 0.0392$. We note that the recovering error value is slightly higher than the mean error achieved in the previous experiment, where the dependency of the observations on the parameters was less complicated.

V. Acoustic Channels

We demonstrate the recovering of independent parameters of acoustic channels by first describing the simulation model and laying out theoretical background. Then, we present some experimental results.

A. The Image Model

The propagation of a sound wave within an enclosure can be considered linear if the medium is homogeneous. In this case, the propagation is governed by the wave equation. Accordingly, the acoustic channel from a source to a microphone is obtained by solving the wave equation. However, this solution can hardly ever be expressed analytically, and therefore, must be approximated. The most common method for approximating the solution is the *Image Method*, presented by Allen and Berkley [33]. This method efficiently computes a finite impulse response (FIR) that approximates the acoustic channel between a source and a sensor in a rectangular room. To model an ideal impulse response from a source to a sensor, all possible sound reflection paths should be resolved. These paths propagate through the room and are reflected after every collision with the room walls. The energy of the sound in each such propagation path decreases as a consequence of the sound absorption of the air and of the walls. To circumvent the calculations of all the reflections and collisions, the image method is based on simulating virtual sources, called images. These virtual sources are located beyond the room boundaries, such that the direct propagation path between the virtual source and the microphone, approximates the reflected path.

Consider a rectangular room with length, width, and height denoted by L_x, L_y, and L_z. Let the sound source be at a location $\mathbf{r}_s = [x_s, y_s, z_s]$, and let the microphone be at a location $\mathbf{r} = [x, y, z]$. Both vectors are with respect to the origin, which is located at one of the corners of the room. The relative positions of the images are computed with respect to the walls at $x = 0$, $y = 0$, and $z = 0$ can be written as

$$\mathbf{r}_p = [(1-2p_x)x_s - x; (1-2p_y)y_s - y; (1-2p_z)z_s - z]$$

where $p = (p_x, p_y, p_z)$ is a triplet consisting of binary elements $p_x, p_y, p_z \in \{0, 1\}$ representing the eight different reflection directions. In order to consider all images, let $\mathbf{r}_m = [2m_xL_x, 2m_yL_y, 2m_zL_z]$, where m_x, m_y, and m_z are integer values between $-P$ and P, where P represents the maximal order of reflection taken into account. Accordingly, let $\mathbf{r}_i$ denote the position of an image

$$\mathbf{r}_i = \mathbf{r} + \mathbf{r}_p + \mathbf{r}_m.$$

The corresponding distance between each image and the microphone is given by $d = \|\mathbf{r}_i - \mathbf{r}\| = \|\mathbf{r}_p + \mathbf{r}_m\|$, and time delay of arrival of the reflected sound is expressed by $\tau = \frac{d}{v} = \frac{\|\mathbf{r}_p + \mathbf{r}_m\|}{v}$.

The finite impulse response can now be written as a superposition of all attenuated and delayed reflections, given by

$$h_{\mathbf{r},\mathbf{r_s},\boldsymbol{\beta}}(t) = \sum_{p\in\mathcal{P}}\sum_{m\in\mathcal{M}} \beta_{x_1}^{|m_x-q|}\beta_{x_2}^{|m_x|} \times \beta_{y_1}^{|m_y-j|}\beta_{y_2}^{|m_y|}\beta_{z_1}^{|m_z-k|}\beta_{z_2}^{|m_z|}\frac{\delta(t-\tau)}{4\pi d} \quad (26)$$

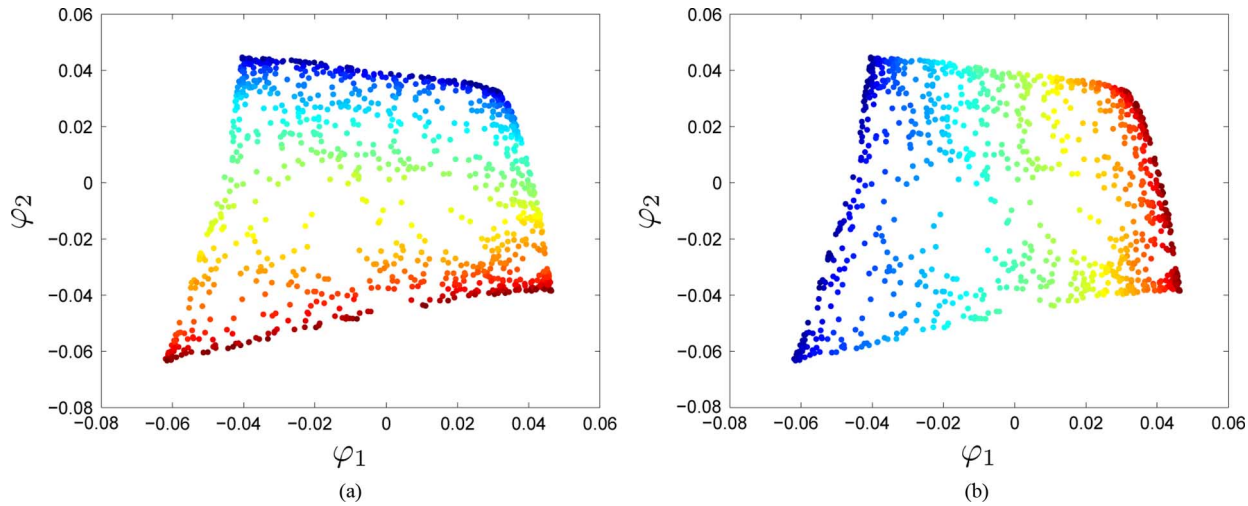

Fig. 6. Scatter plot of the embedded training samples with the fourth-order case: (a) Color coding according to the values of the first pole $\boldsymbol{\theta}^{(1)}$ and (b) color coding according to the values of the second pole $\boldsymbol{\theta}^{(2)}$.

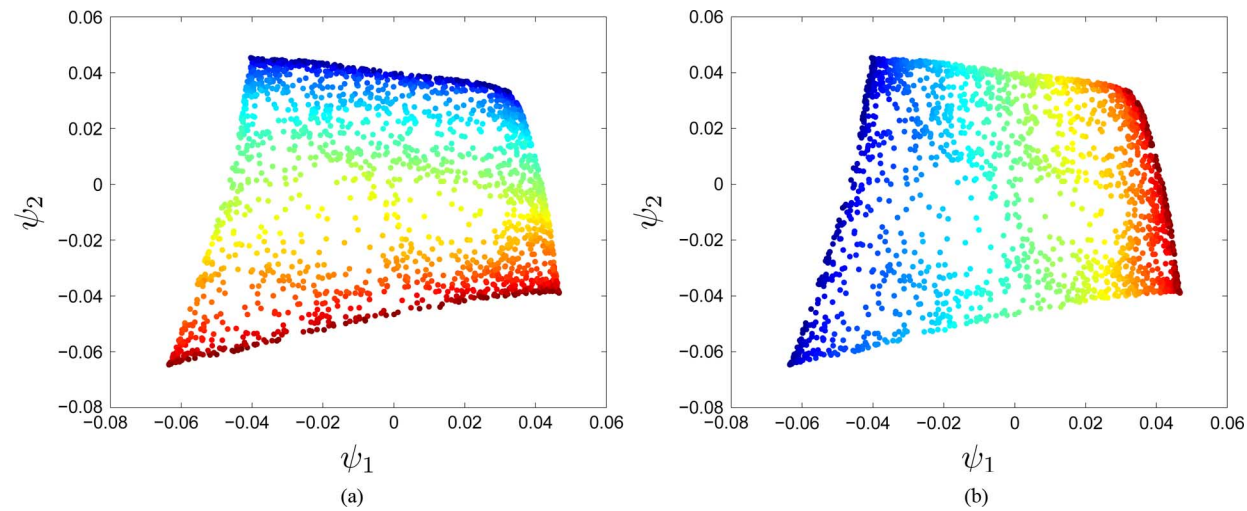

Fig. 7. Scatter plot of the embedded samples with the fourth-order case: (a) Color coding according to the values of the first pole $\boldsymbol{\theta}^{(1)}$ and (b) color coding according to the values of the second pole $\boldsymbol{\theta}^{(2)}$.

where $\mathcal{P} = \{(p_x, p_y, p_z) | p_x, p_y, p_z \in \{0, 1\}\}$, $\mathcal{M} = \{(m_x, m_y, m_z) | -P \leq m_x, m_y, m_z \leq P\}$, and $\boldsymbol{\beta} = (\beta_{x_1}, \beta_{x_2}, \beta_{y_1}, \beta_{y_2}, \beta_{z_1}, \beta_{z_2})$ are the reflection coefficients of the six walls. In discrete time simulations, the delays do not always fall at the sampling instants. However, for simplicity, we assume band-limited excitation and that the sampling frequency f_s is sufficiently high, such that $f_s \tau$ is approximately an integer for each delay τ. See [34] for alternative discrete simulation. Finally, in order to simulate the signal picked up by the microphone, the source signal can be convolved with the corresponding impulse response. For more details regarding acoustic channel modeling and simulating, we refer the readers to [35] and the references therein.

In order to approximate channels in typical rooms, we usually need to take into account delayed reflections ranging between 0.1 and 2 seconds. For example, sampling frequency $f_s = 16$ kHz corresponds to impulse responses of length ranging between 1600 and 32 000. Consequently, typical impulse responses consist of thousands of taps. In other words, each impulse response can be expressed as a vector $h_{\mathbf{r},\mathbf{r_s},\boldsymbol{\beta}}$ in a high-dimensional space. However, the presentation of the finite impulse response in (26) implies that the acoustic channel between a source and a microphone inside a rectangular room is controlled by a set of $d = 12$ parameters: 1) the six reflection coefficients of the walls $\boldsymbol{\beta}$; 2) the location of the source $\mathbf{r}_s$; and 3) the location of the microphone $\mathbf{r}$. It is worthwhile noting, that the dependency between the impulse response of an acoustic channel and the controlling parameters, as conveyed in (26), is highly nonlinear. Therefore, the task of recovering the controlling parameters from measurements of the signal picked up in the microphone, is challenging.

Particular parameters of interest are the source coordinates. Locating the source is a problem that has drawn enormous research efforts in the last few decades [36]–[38]. Usually, a beamformer based on microphone array measurements is implemented [39]–[43]. In this work, we show how to recover the source location based on measurements from a single microphone relying on training.

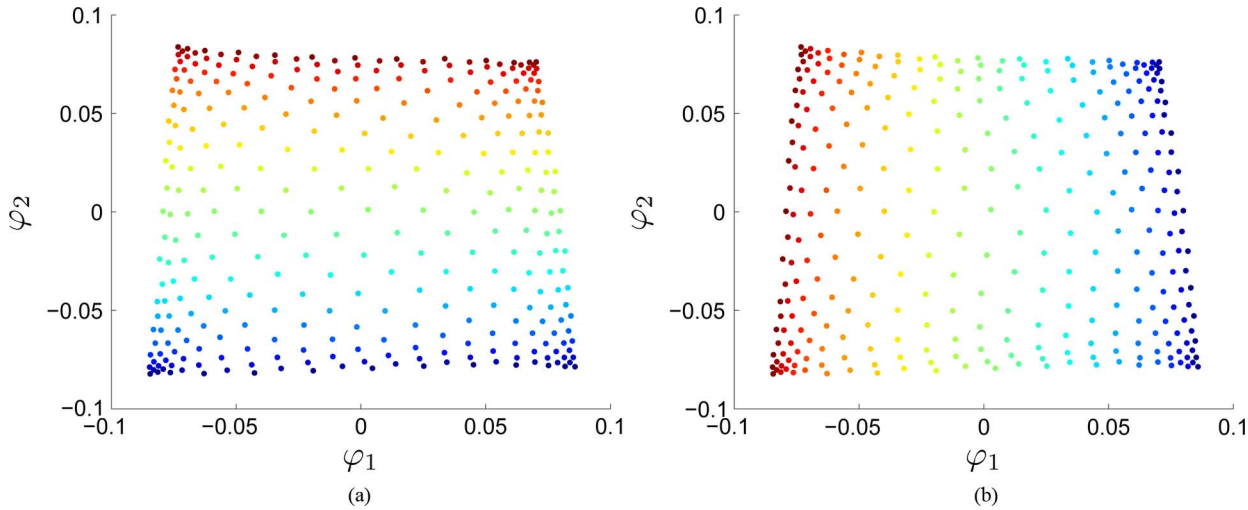

Fig. 8. Scatter plot of the embedding $\mathbf{\Phi_2}$: (a) With color coding according to the values of the reflection coefficient β_{x_1} and (b) with color coding according to the values of the reflection coefficient β_{x_2}.

B. Experimental Results

In this section, we examine the ability of the proposed method to recover the controlling parameters of acoustic channels, simulated using the image method, as described in Section V-A.

In the first experiment, we recover the *reflection coefficients* of two walls. We start by generating $m = 300$ training channels. We equally distribute 300 reflection coefficients of two walls β_{x_1}, β_{x_2} in the range $[0.15, 0.55]$, creating a 2-D grid. The other four coefficients are set to 0.15. Then we simulate a room of size $[L_x, L_y, L_z] = [6, 6, 3]$. We place a microphone at $\mathbf{r} = [3, 1, 1]$, and a source at $[1.9, 3.8, 1]$, distant 3 m from the microphone and at the same altitude. The azimuth angle of the direction of arrival of the direct sound propagating from source to the microphone is $\frac{5\pi}{8}$, and the elevation angle is 0. Let $\boldsymbol{\theta} = [\beta_{x_1}, \beta_{x_2}]$ denote the 2 controlling parameters of the acoustic channel of order $d = 2$, and let $\bar{\Theta}$ denote the set of parameters on the grid. For each parameter $\bar{\boldsymbol{\theta}}_i$, we create $L = 20$ low variance Gaussian perturbations $\boldsymbol{\theta}_{i_j}$ to create a local "cloud" in the vicinity of $\bar{\boldsymbol{\theta}}_i$. Now, using the image method, we simulate $m \cdot L = 300 \cdot 20$ acoustic channels, where each channel $\mathbf{h}_{\bar{\boldsymbol{\theta}}_i}$ corresponds to $\bar{\boldsymbol{\theta}}_i$.

The training channels and their clouds are mapped to an observable space as follows. For each parameter vector $\bar{\boldsymbol{\theta}}_i$, we generate a white Gaussian excitation signal $\mathbf{x}_i$ of length $N = 24\,000$ and measure the output signal $\mathbf{y}_i$, of $\mathbf{x}_i$ going through the corresponding acoustic channel $\mathbf{h}_{\bar{\boldsymbol{\theta}}_i}$. Based on the measured output, the first $D = 24$ elements $\mathbf{c}_i$ of the covariance function are calculated. Let $\bar{\Gamma} = \{\bar{\mathbf{c}}_i\}_{i=1}^m$ denote the set of observations corresponding to the training parameters. In addition, based on the observations $\{\mathbf{c}_{i_j}\}$ corresponding to the cloud of parameters around $\bar{\boldsymbol{\theta}}_i$, the covariance matrix $\mathbf{\Sigma}(\bar{\mathbf{c}}_i)$ of each training sample is computed.

Now, an additional 325 pairs of reflection coefficients are generated in the same range and mapped to the observable space as described above. Let Θ and Γ denote the sets of all $M = 625$ samples in both the parameter and observable spaces.

In summary, we have a set of $M = 625$ parameter vectors of a 2-D reflection coefficients space, and a corresponding set of M observations in 24-D observable space. The observations are obtained via a nonlinear mapping $c : \mathbb{R}^2 \longrightarrow \mathbb{R}^{24}$ of the reflection coefficients, i.e., $\mathbf{c}_i = c(\boldsymbol{\theta}_i)$. In this case, the nonlinearity conveys the relation between the reflection coefficients and the acoustic channel, combined with the relation between the acoustic channel and the observation.

According to Algorithm 1, and similarly to the construction in Section IV-D, we obtain a 2-D embedding of $\bar{\Gamma}$ via

$$\mathbf{\Phi}_2 : \bar{\mathbf{c}}_i \longrightarrow [\boldsymbol{\varphi}_1(\bar{\mathbf{c}}_i), \boldsymbol{\varphi}_2(\bar{\mathbf{c}}_i)]^T$$

using $\varepsilon = 0.007$. This kernel scale was chosen such that it brings the validation error to a minimum. Fig. 8 shows a scatter plot of the embedded training samples, where the color coding corresponds to the values of the parameters β_{x_1} and β_{x_2}. We observe that the samples are organized on a rectangular grid. In addition, the coloring of the samples is parallel to the axes. Hence, the embedding represents the two reflection coefficients.

By constructing the matrix $\tilde{\mathbf{A}}$, the extended eigenvectors $\boldsymbol{\psi}_j$ are calculated. Thus, we obtain embedding of the entire observation set via

$$\mathbf{\Psi}_2 : \mathbf{c}_i \longrightarrow [\boldsymbol{\psi}_1(\mathbf{c}_i), \boldsymbol{\psi}_2(\mathbf{c}_i)]^T .$$

In Fig. 9, we scatter plot the embedded samples. From the coloring of the samples, we conclude that the extended embedding captures the independent controlling parameters. The recovering of the reflection coefficients based on interpolating the training samples according to the distance in the embedded space (15) yields a mean error of $\mathrm{err}_{\mathrm{prm}} = 0.0067$.

In the second experiment, we test the ability of the proposed method to recover the *location* of the source. We simulate the same room dimensions and location of the microphone. We uniformly distribute $m = 300$ source locations on a sector of a sphere around the microphone. The sphere radius is 3 m, the sector azimuth and elevation angles range between $[0, \frac{\pi}{16}]$. Therefore, we have approximately one source per 1° in both look directions. In this experiment, the independent controlling parameter $\boldsymbol{\theta}$ is a pair of azimuth and elevation angles. The rest of the experiment is performed similarly to the first experiment.

Fig. 10 shows a scatter plot of the embedded training samples, where the color coding corresponds to the parameters $\{\bar{\boldsymbol{\theta}}_i\}$. It implies that both the azimuth and elevation angles are accurately recovered. The scatter plot takes the shape of a rectangular grid

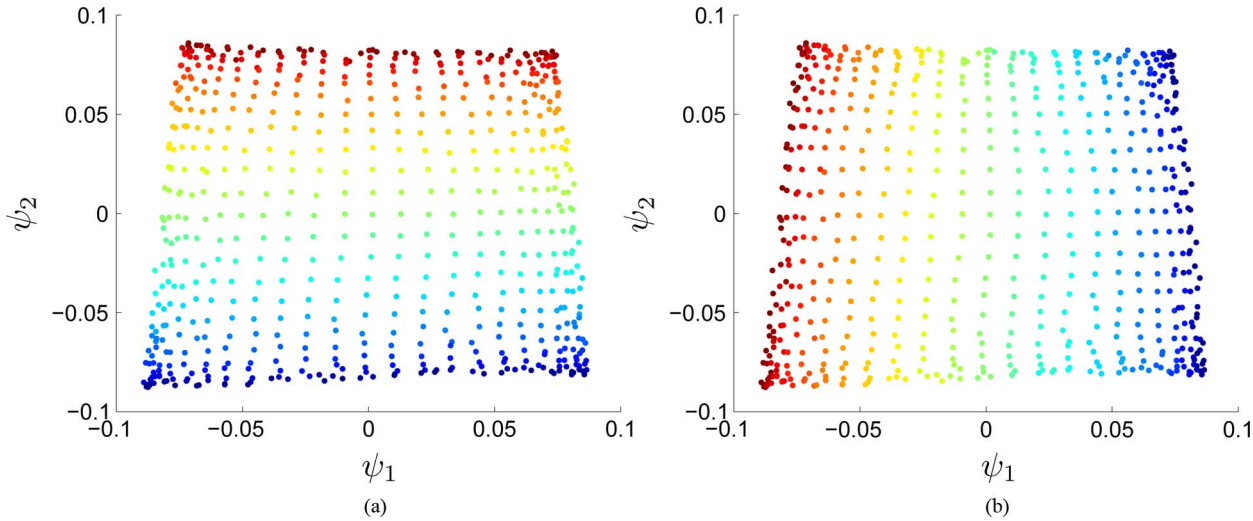

Fig. 9. Scatter plot of the extended embedding $\mathbf{\Psi}_2$: (a) With color coding according to the values of the reflection coefficient β_{x_1} and (b) with color coding according to the values of the reflection coefficient β_{x_2}.

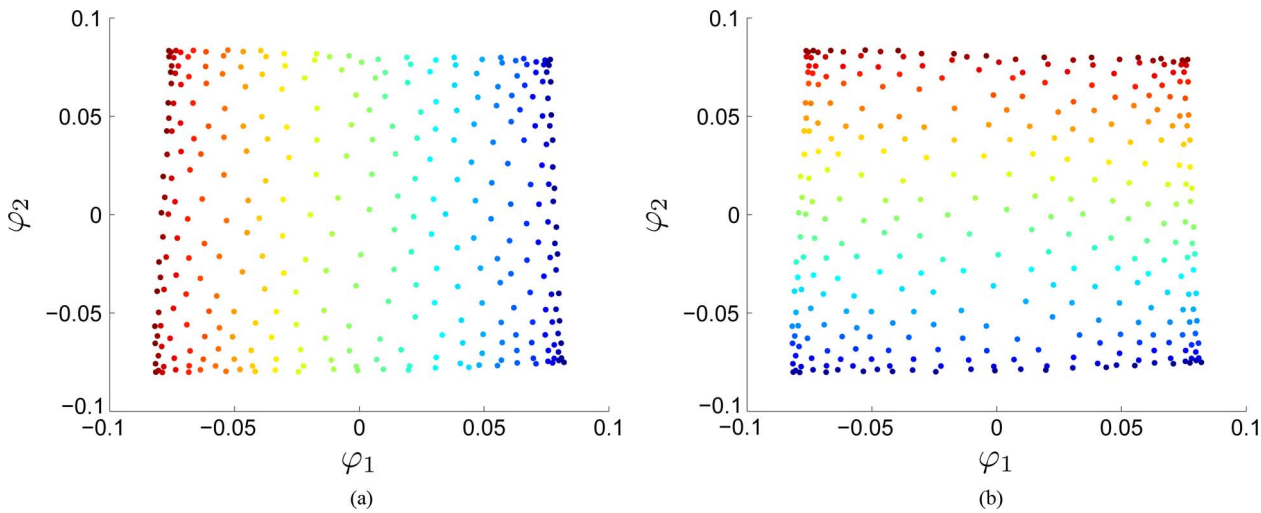

Fig. 10. Scatter plot of $\mathbf{\Phi}_2$: (a) With color coding according to the azimuth angle and (b) with color coding according to the elevation angle.

where, according to the coloring, each axis represents either the azimuth or the elevation angle.

In Fig. 11, we present the embedding of the additional points. From the coloring of the points, we conclude that the extended embedding captures the independent controlling parameters, as the coloring scheme is maintained. In this case, recovering the original parameters yields a mean error $\mathrm{err}_{\mathrm{prm}} = 0.0155$. This result is of particular interest since we accurately recover the direction of arrival of a random source in a room based on observations from a *single* microphone and training.

VI. Conclusion

We proposed a general algorithm for reparametrization of linear systems using diffusion kernels. The proposed algorithm is based on recent developments of spectral and nonlinear independent component analysis techniques, anisotropic kernels, and classical results from statistical signal processing and Fourier analysis. We claim that each system can be viewed as a black box controlled by several independent parameters. By recovering these parameters, we reveal the actual degrees of freedom of the system and obtain its intrinsic modeling. These attractive features are extremely useful for system design, control, and calibration. We employed the proposed algorithm on both synthetic and practical examples. We showed that the proposed method can accurately recover the poles of an autoregressive process and retrieve the controlling parameters of acoustic channels. Acoustic channels are a fundamental component in speech processing applications, such as speech dereverberation, source localization, and echo cancellation. Therefore, the parametrization of acoustic channels is highly important, especially since acoustic channels are known to be challenging to model and acquire.

The characterization of processes (e.g., an AR process) is of particular interest since it opens the door for intrinsic modeling of audio signals. As described in the paper, we can view any audio signal as a product of artificial or natural (e.g., human vocal tract) musical instruments. Thus, by capturing the instrument's intrinsic geometric structure, we are able to provide perceptual analysis. For future work, we aim to explore this new lead in order to obtain characterization of, for example, different music tones, various instruments, speech phonemes, or different

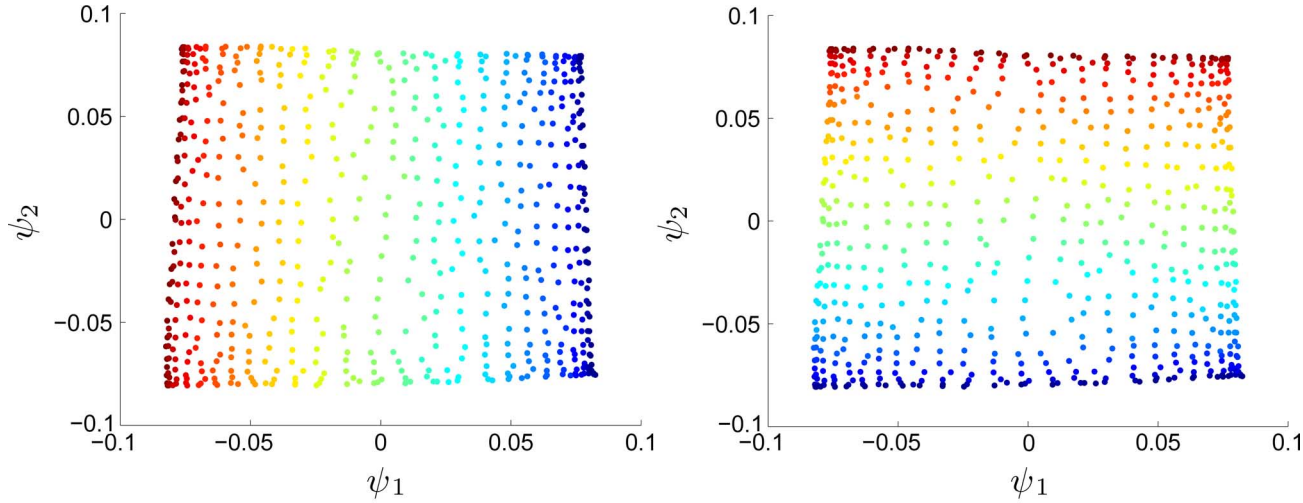

Fig. 11. Scatter plot of Ψ_2: (a) With color coding according to the azimuth angle and (b) with color coding according to the elevation angle.

speakers. Such features may enable us to naturally cluster, classify, or even filter music genres, speakers, phonemes, and other similar tasks which are challenging to perform using existing tools.

Appendix I
Temporal Evolution Model of the Controlling Parameters

Following Singer and Coifman [1], we assume that the controlling parameters evolve according to a stochastic differential equation. Specifically, the parameters are described as independent Itó processes [8], [9], given by

$$d\boldsymbol{\theta}^{(i)} = \mathbf{a}^{(i)}(\boldsymbol{\theta}^{(i)})dt + \mathbf{b}^{(i)}(\boldsymbol{\theta}^{(i)})d\mathbf{w}^{(i)},\ i = 1, \ldots, d \quad (27)$$

where $\mathbf{a} \in \mathbb{R}^d$ and $\mathbf{b} \in \mathbb{R}^d$ are unknown drift and noise coefficients, and $d\mathbf{w}$ are independent white noises ($\mathbf{w}$ are Brownian motions). As described in Section II, this implies that the controlling parameters evolve according to two regimes: 1) small fluctuation regime conveyed by Brownian motion and the noise coefficients $\mathbf{b}$ and 2) slow system variations dependent on the drift coefficients $\mathbf{a}$.

From 2), we have that the controlling parameters are observed via the nonlinear map $c : \mathbb{R}^d \longrightarrow \mathbb{R}^D$. The observed elements $\mathbf{c}^{(j)}$ satisfy the stochastic dynamics given by the Itô lemma [8], [9]

$$d\mathbf{c}^{(j)} = \sum_{i=1}^{d} \left(\frac{1}{2} \left(\mathbf{b}^{(i)} \right)^2 c_{ii}^j + \mathbf{a}^{(i)} c_i^j \right) dt + \sum_{i=1}^{d} \mathbf{b}^{(i)} c_i^j d\mathbf{w}^{(i)},\ j = 1, \ldots, D \quad (28)$$

where c_i^j and c_{ii}^j are first- and second-order partial derivatives of the jth coordinate of the mapping c with respect to $\boldsymbol{\theta}^{(i)}$.

From (4), using (28), we obtain

$$\boldsymbol{\Sigma}^{(jk)}(\mathbf{c}) = \sum_{i=1}^{d} (\mathbf{b}^{(i)})^2 c_i^j c_i^k. \quad (29)$$

In matrix form, we can express (29) by the Jacobian matrix $\mathbf{J}$ of the function c as

$$\boldsymbol{\Sigma}(\mathbf{c}) = \mathbf{J}(\mathbf{c})\mathbf{B}^2\mathbf{J}^T(\mathbf{c})$$

where $\mathbf{B}$ is a diagonal matrix with $\mathbf{B}^{(ii)} = \mathbf{b}^{(i)}$. The matrix $\mathbf{B}$ can be assumed to be the identity matrix $\mathbf{B} = \mathbf{I}$, by applying a change of variables on (27) such that

$$d\tilde{\boldsymbol{\theta}}^{(i)} = \tilde{\mathbf{a}}^{(i)}(\tilde{\boldsymbol{\theta}}^{(i)})dt + \mathbf{1}d\mathbf{w}^{(i)}, i = 1, \ldots, d$$

where $\mathbf{1}$ is a vector of ones of length d. In this case, using the Itô lemma, we obtain

$$\boldsymbol{\Sigma}(\mathbf{c}) = \mathbf{J}(\mathbf{c})\mathbf{J}^T(\mathbf{c}).$$

Appendix II
Euclidean Distances on the Parametric Manifold

We briefly review the derivation of the approximation of the Euclidean distance in the parametric space from [15]. Let $\boldsymbol{\theta}, \boldsymbol{\vartheta} \in \mathbb{R}^d$ be two parameter vectors (i.e., two system configurations in the parametric space). According to previous notation, we observe the nonlinear mapping $c : \Theta \longrightarrow \Gamma$. Let $\mathbf{c} = c(\boldsymbol{\theta})$ and $\boldsymbol{\gamma} = c(\boldsymbol{\vartheta})$ be the mapping of $\boldsymbol{\theta}$ and $\boldsymbol{\vartheta}$ into the observable space. Define $g : \Gamma \longrightarrow \Theta$ to be the inverse map of c. Each coordinate of g at $\boldsymbol{\vartheta} = g(\boldsymbol{\gamma})$ can be approximated by a Taylor series at the middle point $\frac{\boldsymbol{\gamma}+\mathbf{c}}{2}$, denoting $\boldsymbol{\delta} = g\left(\frac{\boldsymbol{\gamma}+\mathbf{c}}{2}\right)$:

$$\boldsymbol{\vartheta}^{(i)} = \boldsymbol{\delta}^{(i)} + \frac{1}{2}\sum_j g_j^i \left(\frac{\boldsymbol{\gamma}+\mathbf{c}}{2}\right)\left(\boldsymbol{\gamma}^{(j)} - \mathbf{c}^{(j)}\right) + \frac{1}{8}\sum_{k,l} g_{kl}^i \left(\frac{\boldsymbol{\gamma}+\mathbf{c}}{2}\right)\left(\boldsymbol{\gamma}^{(k)} - \mathbf{c}^{(k)}\right)\left(\boldsymbol{\gamma}^{(l)} - \mathbf{c}^{(l)}\right) + \mathcal{O}\left(\|\boldsymbol{\gamma} - \mathbf{c}\|^3\right) \quad (30)$$

where g_j^i is the first order derivatives of $g^{(i)}$ with respect to $\mathbf{c}^{(j)}$, and g_{kl}^i are the second order derivatives of $g^{(i)}$ with respect to $\mathbf{c}^{(k)}$ and $\mathbf{c}^{(l)}$. Similarly to (30), we approximate g at $\boldsymbol{\theta} = g(\mathbf{c})$

$$\begin{aligned}\boldsymbol{\theta}^{(i)} = &\boldsymbol{\delta}^{(i)} + \frac{1}{2}\sum_j g_j^i \left(\frac{\boldsymbol{\gamma}+\mathbf{c}}{2}\right)\left(\mathbf{c}^{(j)} - \boldsymbol{\gamma}^{(j)}\right) \\ &+ \frac{1}{8}\sum_{k,l} g_{kl}^i \left(\frac{\boldsymbol{\gamma}+\mathbf{c}}{2}\right)\left(\mathbf{c}^{(k)} - \boldsymbol{\gamma}^{(k)}\right)\left(\mathbf{c}^{(l)} - \boldsymbol{\gamma}^{(l)}\right) \\ &+ \mathcal{O}\left(\|\mathbf{c}-\boldsymbol{\gamma}\|^3\right). \end{aligned} \tag{31}$$

By the definition of the norm, we have

$$\|\boldsymbol{\theta} - \boldsymbol{\vartheta}\|^2 = \sum_{i=1}^{d}\left(\boldsymbol{\theta}^{(i)} - \boldsymbol{\vartheta}^{(i)}\right)^2. \tag{32}$$

Substituting (30) and (31) into (32) yields

$$\begin{aligned}\|\boldsymbol{\theta} - \boldsymbol{\vartheta}\|^2 = &\sum_{i,j,k=1}^{d} g_j^i \left(\frac{\boldsymbol{\gamma}+\mathbf{c}}{2}\right) g_k^i \left(\frac{\boldsymbol{\gamma}+\mathbf{c}}{2}\right) \\ &\times \left(\mathbf{c}^{(j)} - \boldsymbol{\gamma}^{(j)}\right)\left(\mathbf{c}^{(k)} - \boldsymbol{\gamma}^{(k)}\right) \\ &+ \mathcal{O}\left(\|\mathbf{c}-\boldsymbol{\gamma}\|^4\right) \end{aligned} \tag{33}$$

and in matrix form

$$\begin{aligned}\|\boldsymbol{\theta} - \boldsymbol{\vartheta}\|^2 = (\mathbf{c}-\boldsymbol{\gamma})^T &\left[\left(\mathbf{J}\mathbf{J}^T\right)^{-1}\left(\frac{\boldsymbol{\gamma}+\mathbf{c}}{2}\right)\right](\mathbf{c}-\boldsymbol{\gamma}) \\ &+\mathcal{O}\left(\|\mathbf{c}-\boldsymbol{\gamma}\|^4\right). \end{aligned} \tag{34}$$

It is shown in [15] that

$$\begin{aligned}\left(\mathbf{J}\mathbf{J}^T\right)^{-1}\left(\frac{\boldsymbol{\gamma}+\mathbf{c}}{2}\right) = 2\left[\left(\mathbf{J}\mathbf{J}^T\right)(\mathbf{c}) + \left(\mathbf{J}\mathbf{J}^T\right)(\boldsymbol{\gamma})\right]^{-1} \\ +\mathcal{O}\left(\|\mathbf{c}-\boldsymbol{\gamma}\|^4\right) \end{aligned} \tag{35}$$

where $\mathbf{J}$ is the Jacobian of c. Substituting (35) into (34) yields a second-order approximation of the squared Euclidean distance in the parametric space (7)

$$\begin{aligned}\|\boldsymbol{\theta} - \boldsymbol{\vartheta}\|^2 = 2(\mathbf{c}-\boldsymbol{\gamma})^T &\left[\left(\mathbf{J}\mathbf{J}^T\right)(\mathbf{c}) + \left(\mathbf{J}\mathbf{J}^T\right)(\boldsymbol{\gamma})\right]^{-1} \\ &\times(\mathbf{c}-\boldsymbol{\gamma}) + \mathcal{O}\left(\|\mathbf{c}-\boldsymbol{\gamma}\|^4\right). \end{aligned}$$

Appendix III
An Example of the Anisotropic Kernel Computation

We present a simple 2-D simulation to illustrate the equivalence of computing the kernel between two reference samples directly (10) and via new samples (9). In this example, we assume no mapping between the parameters and the observations, i.e., the nonlinear map c is the identity map. As a result, the Jacobian is a unit matrix. We fix two reference samples at $p_1 = [0, 1]$ and $p_2 = [1, 0]$ and generate M random new samples $\{q_m\}_{m=1}^M$, uniformly distributed in the unit square. We continue by computing the kernel W between p_1 and p_2 in two ways. Direct computation according to (10) yields

$$w_1(p_1, p_2) = \pi \exp\left\{-\frac{\|p_1 - p_2\|^2}{\varepsilon}\right\} \tag{36}$$

and computation based on the new samples according to (9) implies

$$w_2(p_1, p_2) = \frac{1}{\sqrt{s_1 s_2}}\sum_{m=1}^{M} a(p_1, q_m)a(p_2, q_m) \tag{37}$$

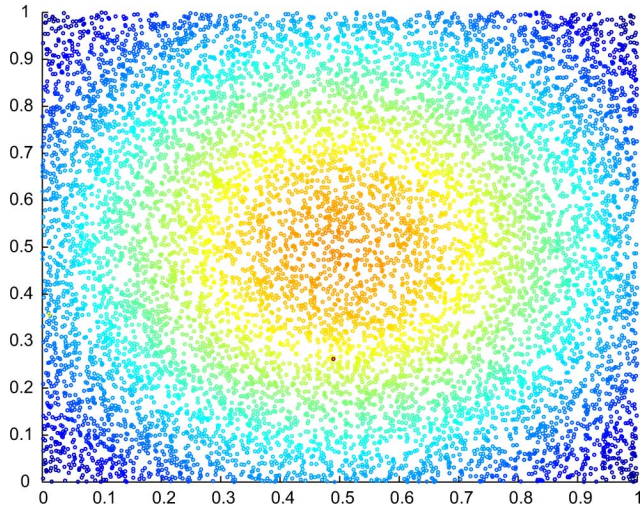

Fig. 12. Scatter plot of the new samples $\{q_m\}$.

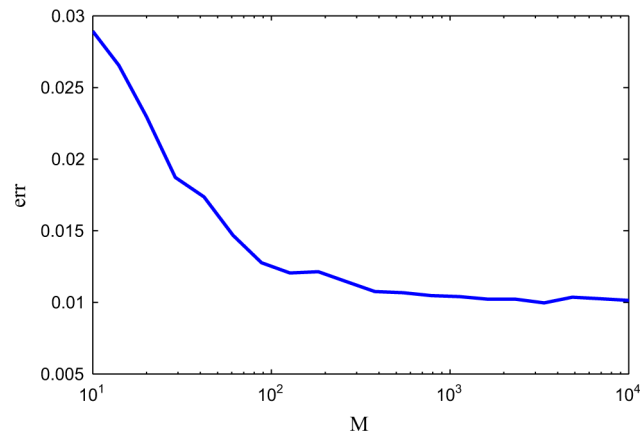

Fig. 13. The mean normalized square error between the distance as function of the number of new samples.

with $a(p_i, q_m) = \exp\left\{\frac{-\|p_i - q_m\|^2}{\varepsilon}\right\}$, and

$$s_i = \sum_{m=1}^{M} a(p_i, q_m)a(p_i, q_m)$$

for $i = \{1, 2\}$. We fix the kernel scale to $\varepsilon = 1$.

Fig. 12 depicts a scatter plot of the new samples. The color coding of each sample q_m is set according to $a(p_1, q_m)a(p_2, q_m)$, which can be seen as the weight of the sample in the summation (37). As seen, only new samples which lie approximately in the middle between p_1 and p_2 obtain high values and therefore are taken into account in (37). For these points $\|p_1 - p_2\| \approx \|p_1 - q_m\| + \|p_2 - q_m\|$. We note that this result is obtained due to the Gaussian kernel, which significantly attenuates the value of the summation term in (37) in case the new sample is located remotely from either of the reference samples.

The described experiment is tested with a different number of new samples M and is repeated several times to yield consistent results. Fig. 13 shows the following mean normalized square error

$$\text{err} = \frac{(w_1(p_1, p_2) - w_2(p_1, p_2))^2}{w_1(p_1, p_2)^2}$$

between the computed distances as a function of the number of new samples. We observe that the error is relatively small and decreases as more new samples are available. In addition, we observe a convergence to a fixed error of small value. This result demonstrates [15, Theorem 3.2] as it implies accurate approximation of the distance between two reference samples computed based on the new samples as proposed in (9), given a sufficient number of new samples.

REFERENCES

[1] A. Singer and R. Coifman, "Non-linear independent component analysis with diffusion maps," *Appl. Comput. Harmon. Anal.*, vol. 25, pp. 226–239, 2008.

[2] R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker, "Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps," *Proc. Nat. Acad. Sci.*, vol. 102, no. 21, pp. 7426–7431, May 2005.

[3] R. Coifman and S. Lafon, "Diffusion maps," *Appl. Comput. Harmon. Anal.*, vol. 21, pp. 5–30, Jul. 2006.

[4] S. Lafon, Y. Keller, and R. R. Coifman, "Data fusion and multicue data matching by diffusion maps," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1784–1797, Nov. 2006.

[5] S. Lafon and A. B. Lee, "Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 9, pp. 1393–1403, Sep. 2006.

[6] B. Nadler, S. Lafon, R. Coifman, and I. G. Kevrekidis, "Diffusion maps, spectral clustering and reaction coordinates of dynamical systems," *Appl. Comput. Harmon. Anal.*, pp. 113–127, 2006.

[7] Y. Keller, R. R. Coifman, S. Lafon, and S. W. Zucker, "Audio-visual group recognition using diffusion maps," *IEEE Trans. Signal Process.*, vol. 58, no. 1, pp. 403–413, Jan. 2010.

[8] K. Ito, "On stochastic differential equations," in *Memoirs Series, Books on Demand*. Providence, RI: American Mathematical Society, 1951.

[9] K. Ito and H. P. McKean, *Diffusion Processes and Their Simple Paths*. New York: Academic Press, 1965.

[10] A. Singer, "Spectral independent component analysis," *Appl. Comput. Harmon. Anal.*, vol. 21, pp. 128–134, 2006.

[11] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using Nystrom method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 214–225, Feb. 2004.

[12] R. Coifman and S. Lafon, "Geometic harmonics: A novel tool for multiscale out-of-sample extension of empirical functions," *Appl. Comput. Harmon. Anal.*, vol. 21, pp. 31–52, Jul. 2006.

[13] A. D. Szlam, M. Maggioni, and R. R. Coifman, "Regularization on graphs with function-adapted diffusion processes," *J. Mach. Learn. Res.*, vol. 9, pp. 1711–1739, 2008.

[14] L. Rosasco, M. Belkin, and E. De Vito, "On learning with intergral operators," *J. Mach. Learn. Res.*, vol. 11, pp. 905–934, 2010.

[15] D. Kushnir, A. Haddad, and R. Coifman, "Anisotropic diffusion on sub-manifolds with application to earth structure classification," *Appl. Comput. Harmon. Anal.*, 2011.

[16] J. Benesty, T. Gansler, D. R. Morgan, M. M. Sondhi, and S. L. Gay, *Advances in Network and Acoustic Echo Cancellation*. New York: Springer, 2001.

[17] E. Hänsler and G. Schmidt, *Acoustic Echo and Noise Control: A Practical Approach*. New York: Wiley, 2004.

[18] I. Cohen, "Relative transfer function identification using speech signals," *IEEE Trans. Speech Audio Process.*, vol. 12, no. 5, pp. 451–459, 2004.

[19] M. Wu and D. Wang, "A two-stage algorithm for one-microphone reverberant speech enhancement," *IEEE Tran. Audio, Speech, Lang. Process.*, vol. 14, no. 3, pp. 774–784, May 2006.

[20] S. Gannot, D. Burshtein, and E. Weinstein, "Signal enhancement using beamforming and nonstationarity with applications to speech," *IEEE Trans. Signal Process.*, vol. 49, no. 8, pp. 1614–1626, Aug. 2001.

[21] S. Gannot and I. Cohen, "Speech enhancement based on the general transfer function GSC and postfiltering," *IEEE Trans. Speech Audio Process.*, vol. 12, no. 6, pp. 561–571, Nov. 2004.

[22] R. Talmon, I. Cohen, and S. Gannot, "Relative transfer function identification using convolutive transfer function approximation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 4, pp. 546–555, May 2009.

[23] R. Talmon, I. Cohen, and S. Gannot, "Convolutive transfer function generalized sidelobe canceler," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 7, pp. 1420–1434, Sep. 2009.

[24] R. Talmon, I. Cohen, and S. Gannot, "Supervised source localization using diffusion kernels," in *Proc. IEEE Workshop Applications Signal Process. Audio Acoust. (WASPAA)*, 2011, pp. 245–248.

[25] A. Singer, R. Erban, I. G. Kevrekidis, and R. Coifman, "Detecting intrinsic slow variables in stochastic dynamical systems by anisotropic diffusion maps," *PNAS*, vol. 106, no. 38, pp. 16090–1605, 2009.

[26] J. G. Proakis and D. K. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, 4th ed. Englewood Cliffs, NJ: Prentice Hall, 2006.

[27] F. R. K. Chung, *Spectral Graph Theory*. Providence, RI: CBMS-AMS, 1997.

[28] B. Nadler, S. Lafon, R. R. Coifman, and I. G. Kevrekidis, "Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators," in *Advances in Neural Information Processing Systems (NIPS)*, Y. Weiss, B. Schölkopf, and J. Platt, Eds. Cambridge: MIT Press, 2006, vol. 18, pp. 955–962.

[29] M. Hein and J. Y. Audibert, L. De Raedt and S. Wrobel, Eds., "Intrinsic dimensionality estimation of submanifold in r^d," in *Proc. 22nd Int. Conf. Mach. Learn., ACM*, 2005, pp. 289–296.

[30] A. Singer, "From graph to manifold Laplacian: The convergence rate," *Appl. Comput. Harmon. Anal.*, vol. 21, pp. 135–144, 2006.

[31] R. Schafer and L. Rabiner, "Digital representations of speech signals," *Proc. IEEE*, vol. 63, no. 4, pp. 662–679, Apr. 1975.

[32] T. F. Quatieri, *Discrete-Time Speech Signal Processing: Principles and Practice*, ser. Prentice-Hall Signal Processing Series. Englewood Cliffs, NJ: Prentice-Hall, 2005.

[33] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small room acoustics," *J. Acoust. Soc. Amer.*, vol. 65, no. 4, pp. 943–950, 1979.

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

[35] E. A. P. Habets, S. Gannot, and I. Cohen, "Dual-microphone speech dereverberation in a noisy environment," in *Proc. IEEE Int. Symp. Signal Process. Inf. Technol.*, Aug. 2006, pp. 651–655.

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

[37] J. Benesty, "Adaptive eigenvalue decomposition algorithm for passive acoustic source localization," *J. Acoust. Soc. Amer.*, vol. 107, no. 1, pp. 384–391, 2000.

[38] J. Chen, J. Benesty, and Y. Huang, "Time delay estimation in room acoustic environments: An overview," *EURASIP J. Appl. Signal Process.*, vol. 2006, p. 170, Jan. 2006.

[39] T. G. Dvorkind and S. Gannot, "Time difference of arrival estimation of speech source in a noisy and reverberant environment," *Signal Process.*, vol. 85, no. 1, pp. 177–204, 2005.

[40] S. Gannot and T. G. Dvorkind, "Microphone array speaker localizers using spatial-temporal information," *EURASIP J. Appl. Signal Process.*, p. 174, Jan. 2006.

[41] J. Dmochowski, J. Benesty, and S. Affes, "Linearly constrained minimum variance source localization and spectral estimation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 8, pp. 1490–1502, Nov. 2008.

[42] J. P. Dmochowski and J. Benesty, "Steered beamforming approaches for acoustic source localization," in *Speech Processing in Modern Communication*, I. Cohen, J. Benesty, and S. Gannot, Eds. New York: Springer, 2010, pp. 307–337.

[43] A. Levy, S. Gannot, and E. A. P. Habets, "Multiple hypothesis extended particle filter for acoustic source localization in reverberant environments," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 6, pp. 1540–1555, 2011.

Ronen Talmon (S'09–M'11) received the B.A. degree (*cum laude*) in mathematics and computer science from the Open University, Ra'anana, Israel, in 2005 and the Ph.D. degree in electrical engineering from the Technion—Israel Institute of Technology, Haifa, in 2011.

From 2000 to 2005, he was a software developer and researcher at a technological unit of the Israeli Defense Forces. From 2005 to 2011, he was a Teaching Assistant and a Project Supervisor with the Signal and Image Processing Lab (SIPL), Electrical Engineering Department, Technion. In 2011, he joined the Mathematics Department at Yale University, where he is currently a Gibbs Assistant Professor. His research interests are statistical signal processing, analysis and modeling of signals, speech enhancement, applied harmonic analysis, and diffusion geometry.

Dr. Talmon is the recipient of the Irwin and Joan Jacobs Fellowship for 2011, the Viterbi Fellowship for 2011–2012, the Excellent Project Supervisor Award for 2010, and the Excellence in Teaching Award for outstanding teaching assistants for 2008 and 2011.

Dan Kushnir received the B.Sc. degree in computer science from the Hebrew University, Jerusalem, in 2001 and the M.Sc. and Ph.D. degrees both from the Department of Computer Science and Applied Mathematics at the Weizmann Institute of Science, Rehovot, Israel, in 2005 and 2008, respectively.

From 2008 to 2011, he held the J. W. Gibbs Assistant Professor position at the Mathematics Department in Yale University, New Haven, CT. In January 2012, he joined Bell Laboratories, Murray Hill, NJ, as a researcher. His current research interests include applied harmonic analysis, numerical analysis, signal processing, and machine learning. He developed efficient spectral algorithms for data analysis and focuses on the development of fast methods for massive numerical computations in machine learning and data analysis.

Ronald R. Coifman received the Ph.D. degree from the University of Geneva, Geneva, Switzerland, in 1965.

He was formerly a Professor at Washington University in St. Louis. In 1980, he joined Yale University, New Haven, CT, where he was Chairman of the Yale Mathematics Department from 1986 to 1989 and where he is currently Phillips Professor of mathematics. His recent publications have been in the areas of nonlinear Fourier analysis, wavelet theory, numerical analysis, and scattering theory. He is currently leading a research program to develop new mathematical tools for efficient transcription of data, with applications to feature extraction recognition, denoising, and information organization.

Prof. Coifman is a member of the National Academy of Sciences, American Academy of Arts and Sciences, and the Connecticut Academy of Sciences and Engineering. He received the DARPA Sustained Excellence Award in 1996, and the 1996 Connecticut Science Medal. the 1999 Pioneer award from the International Society for Industrial and Applied Mathematics, the National Science Medal 1999, and the Wavelet Pioneer Award in 2007.

Israel Cohen (M'01–SM'03) received the B.Sc. (*summa cum laude*), M.Sc., and Ph.D. degrees in electrical engineering from the Technion—Israel Institute of Technology, Haifa, Israel, in 1990, 1993 and 1998, respectively.

From 1990 to 1998, he was a Research Scientist with RAFAEL Research Laboratories, Haifa, Israel Ministry of Defense. From 1998 to 2001, he was a Postdoctoral Research Associate with the Computer Science Department, Yale University, New Haven, CT. In 2001, he joined the Electrical Engineering Department of the Technion, where he is currently an Associate Professor. His research interests are statistical signal processing, analysis and modeling of acoustic signals, speech enhancement, noise estimation, microphone arrays, source localization, blind source separation, system identification, and adaptive filtering. He is a coeditor of the Multichannel Speech Processing section of the *Springer Handbook of Speech Processing* (Springer, 2008), a coauthor of *Noise Reduction in Speech Processing* (Springer, 2009), and a coeditor of *Speech Processing in Modern Communication: Challenges and Perspectives* (Springer, 20010).

Dr. Cohen is a recipient of the Alexander Goldberg Prize for Excellence in Research, and the Muriel and David Jacknow award for Excellence in Teaching. He served as Associate Editor of the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING and the IEEE SIGNAL PROCESSING LETTERS, and as Guest Editor of the *EURASIP Journal on Advances in Signal Processing* Special Issue on Advances in Multimicrophone Speech Processing and the *EURASIP Speech Communication Journal* Special Issue on Speech Enhancement. He is General Co-Chair of the 2010 International Workshop on Acoustic Echo and Noise Control (IWAENC).

Sharon Gannot (S'92–M'01–SM'06) received the B.Sc. degree (*summa cum laude*) from the Technion—Israel Institute of Technology, Haifa, Israel, in 1986 and the M.Sc. (*cum laude*) and Ph.D. degrees from Tel-Aviv University, Israel, in 1995 and 2000 respectively, all in electrical engineering.

In 2001, he held a postdoctoral position at the Department of Electrical Engineering (ESAT-SISTA) at K.U.Leuven, Belgium. From 2002 to 2003, he held a research and teaching position at the Faculty of Electrical Engineering, Technion—Israel Institute of Technology. Currently, he is an Associate Professor at the School of Engineering, Bar-Ilan University, Israel, where he is heading the Speech and Signal Processing Laboratory. His research interests include parameter estimation, statistical signal processing, especially speech processing using either single- or multi-microphone arrays.

Prof. Gannot is the recipient of the Bar-Ilan University Outstanding Lecturer Award for 2010. He is currently an Associate Editor of the IEEE TRANSACTIONS ON SPEECH, AUDIO AND LANGUAGE PROCESSING. He served as an Associate Editor of the *EURASIP Journal of Advances in signal Processing* between 2001 and 2003, and as an Editor of two special issues on multi-microphone speech processing of the same journal. He also served as a Guest Editor of the ELSEVIER *Speech Communication* journal and a reviewer of many IEEE journals and conferences. He has been a member of the Audio and Acoustic Signal Processing (AASSP) technical committee of the IEEE since January 2010. He has also been a member of the Technical and Steering Committee of the International Workshop on Acoustic Echo and Noise Control (IWAENC) since 2005 and was the General Co-Chair of IWAENC held at Tel-Aviv, Israel, in August 2010. He will serve as the General Co-Chair of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA) in 2013.