Modeling natural and artificial systems has a key role in various applications, and has long been a task that drew enormous efforts. In this work, instead of exploring predefined models, we aim at implicitly identifying 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 modeling 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 real examples has shown accurate recovery of parameters.

Parametrization of Linear Systems Using Diffusion Kernels

R. Talmon[†], D. Kushnir[‡], R. R. Coifman[†], I. Cohen[‡], S. Gannot[*]
Technical Report YALEU/DCS/TR-1445
April 27, 2011

[†] Dept. of Electrical Engineering, Technion, Haifa, 32000, Israel
[‡] Dept. of Mathematics, Yale University, New Haven CT 06511
[*] School of Engineering, Bar-Ilan University, Ramat-Gan, 52900, Israel

Keywords:

Parametrization of Linear Systems Using Diffusion Kernels

Ronen Talmon, *Student 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 a key role in various applications, and has long been a task that drew enormous efforts. In this work, instead of exploring predefined models, we aim at implicitly identifying 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 modeling 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 real examples has shown accurate recovery of parameters.

Index Terms

TODO: keywords

Ronen Talmon and Israel Cohen are with the Department of Electrical Engineering, Technion - Israel Institute of Technology, Haifa 32000, Israel (e-mail: ronenta2@techunix.technion.ac.il; icohen@ee.technion.ac.il).

Dan Kushnir and Ronald R. Coifman are with the Department of Mathematics, Yale University, New Haven 06520, CT (e-mail: dan.kushnir@math.yale.edu; coifman@math.yale.edu).

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

I. Introduction

Modeling natural and artificial systems has a key role in various applications, and has long been a task that drew 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 at implicitly identifying the system degrees of freedom or modes of variability. This approach enables capturing 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. 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 modeling for the system. Thus, in this work, we aim at recovering the controlling parameters of natural and artificial convolution 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. The 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], [3]. 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. 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 [4] 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 [5], [6], [7], [8], [9]. 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. [10] 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. The ability to extend the embedding enables 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 in this work compared to [1], [10] is that the output of a system is not solely determined by the controlling parameters. For that reason, we restrict the scope of this work and consider only *convolution* systems, and propose a spectral algorithm, based on [1], [10], to recover the independent parameters of a convolution system. The proposed algorithm is data-driven, nonlinear, 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 proposed algorithm on both synthetic and real examples. First, we show that the proposed method can accurately recover the poles of an auto-regressive (AR) process. Second, we utilize the proposed 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., [11], [12], [13], [14], [15], [16], [17], [18].

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 by bold capital letters. In addition, accessing elements in vectors and matrices is 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. [19], 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; (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.

We restrict the scope of the work and consider only linear systems. Let x be a real-valued input signal, $h_{\boldsymbol{\theta}}$ be 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 be 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 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 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 measured L times. 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 configuration, e.g. 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 differential equation.

Our goal in this work is to recover the inaccessible set of controlling parameters Θ given the output signal observations. In practice, we would only recover the parameters up to scaling. In addition, we assume m training observations and their corresponding parameters are given beforehand. These training samples are utilized for 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], [10], 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 [20]

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

where $c_x(\tau)$ 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 preferred to observe the second order statistics of the signal directly, conveyed by the covariance of the observable signal in the time domain, rather than using the predefined Fourier transform.

Let $c : \mathbb{R}^d \to \mathbb{R}^D$ 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.

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

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, we calculate according to (2) 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 $\bar{\Theta}$. See Appendix II for demonstration of the setting by observing an auto-regressive (AR) process of order 1.

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

$$\boldsymbol{\Sigma}^{(jk)}(\mathbf{c}) = \text{Cov}\left(\mathbf{c}^{(j)}, \mathbf{c}^{(k)}\right). \tag{3}$$

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

$$\boldsymbol{\Sigma}(\mathbf{c}) = \mathbf{J}(\mathbf{c})\mathbf{J}^T(\mathbf{c}) \tag{4}$$

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 $\boldsymbol{\theta}^{(i)}$. The Jacobian of the function c is of a key importance to the proposed algorithm as described in Section IV, but unfortunately is unavailable. However, (4) enables 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 $\Sigma(\bar{\mathbf{c}}_i)$ for each training observation empirically via

$$\hat{\boldsymbol{\Sigma}}(\bar{\mathbf{c}}_i) = \frac{1}{L}\sum_{j=1}^{L} \mathbf{c}_{i_j}\mathbf{c}_{i_j}^T \tag{5}$$

using the fact that the input signal x is zero mean.

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 [10], 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 [10] that a second-order approximation of the squared Euclidean distance in the parametric space is given by (see Appendix III)

$$\|\boldsymbol{\theta}_j - \boldsymbol{\theta}_k\|^2 = 2(\mathbf{c}_j - \mathbf{c}_k)^T \left[\left(\mathbf{J}\mathbf{J}^T\right)(\mathbf{c}_j) + \left(\mathbf{J}\mathbf{J}^T\right)(\mathbf{c}_k)\right]^{-1} (\mathbf{c}_j - \mathbf{c}_k) + \mathcal{O}\left(\|\mathbf{c}_j - \mathbf{c}_k\|^4\right). \tag{6}$$

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 scale parameter ε, and given by

$$\mathbf{A}^{(kj)} = \exp\left\{-\frac{\|\mathbf{J}^{-1}(\bar{\mathbf{c}}_j)\,(\bar{\mathbf{c}}_j - \mathbf{c}_k)\,\|^2}{\varepsilon}\right\}. \tag{7}$$

We observe the following $m \times m$ matrix

$$\mathbf{W} = \mathbf{S}^{-1/2}\mathbf{A}^T\mathbf{A}\mathbf{S}^{-1/2} \tag{8}$$

where $\mathbf{S}$ is a diagonal matrix containing the sum of A along columns, i.e. $\mathbf{S} = diag\{\mathbf{A}^T\mathbf{1}\}$. It is shown [10] that $\mathbf{W}^{(kj)}$ corresponds to

$$\mathbf{W}^{(kj)} = \frac{\pi}{\sqrt{\det\left((\mathbf{J}^T\mathbf{J})\,(\tilde{\mathbf{c}})\right)}} \exp\left\{-\frac{(\bar{\mathbf{c}}_j - \bar{\mathbf{c}}_k)^T \left[\mathbf{J}\mathbf{J}^T(\bar{\mathbf{c}}_k) + \mathbf{J}\mathbf{J}^T(\bar{\mathbf{c}}_j)\right]^{-1} (\bar{\mathbf{c}}_j - \bar{\mathbf{c}}_k)}{\varepsilon}\right\} \tag{9}$$

for $\bar{\mathbf{c}}_j, \bar{\mathbf{c}}_k \in \bar{\Gamma}$ and $\tilde{\mathbf{c}} = (\bar{\mathbf{c}}_j + \bar{\mathbf{c}}_k)/2$. Based on Lemma 3.3 in [10], $\mathbf{W}$ is a Gaussian kernel based on approximation of the Euclidean distance between the training samples in the parametric space. Moreover, (8) 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 enables the possibility of naturally extending the kernel to new samples as shown in Section IV.

Substituting (4) and (5) into (9) yields

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

Thus, the desired affinity kernel based on the Euclidean distance between the underlying parameters is given by the observations and their covariance matrices.

IV. From the Observable Data to the Linear System Parameters

In this section we propose a supervised algorithm for re-parametrization of linear systems. The recovery of the controlling parameters of the system relies on the kernel computed in Section III. Based on the kernel eigen-decomposition, the observation are mapped into a new domain, which correspond to the parametric domain up to scaling.

A. Inverse mapping

Let $\mathbf{L}$ be the normalized graph-Laplacian [21] 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* [22], [23]

$$\mathcal{L}q = \Delta q - \nabla U \cdot \nabla q,$$

where U is the density potential $U = -2\log p_{\bar{\Theta}}$. Assuming uniform sampling (i.e., constant potential) yields $\nabla U = 0$ and convergence of the graph-Laplacian to the Laplace-Beltrami operator $\mathbf{L} \to \Delta$.

There exist eigenfunctions $\{\varphi_i\}$ of $\mathcal{L}$ that are monotonic functions of the parameters $\boldsymbol{\theta}$ as guarantied by the Strum-Liouville theory. These eigenfunction can be chosen as suggested in [4]. Thus, they can be used to represent the data in terms of its independent controlling parameters. Let $\Phi_d : \bar{\Gamma} \to \mathbb{R}^d$ be a map from the observations to the space spanned by d eigenfunctions of $\mathcal{L}$, given by

$$\Phi_d : \bar{\mathbf{c}}_i \to \left[\varphi_1(\bar{\mathbf{c}}_i), \ldots, \varphi_d(\bar{\mathbf{c}}_i)\right]^T. \tag{10}$$

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

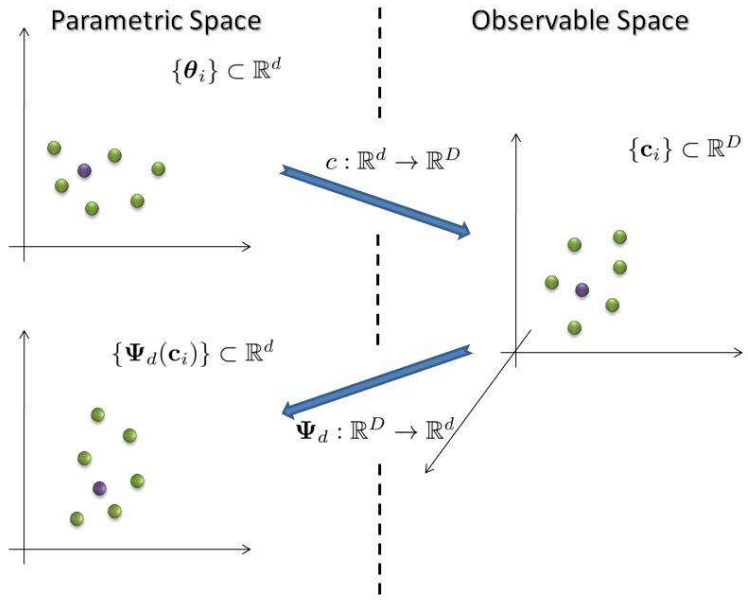

Fig. 1. A diagram of the parametric and observable space. We illustrate the set of samples and the mappings.

B. Restriction and extension operators

Let $\tilde{\mathbf{A}}$ be a normalized affinity matrix $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{S}^{-1/2}$, and let $\{\boldsymbol{\varphi}_j\}_{j=1}^m$ and $\{\boldsymbol{\psi}_j\}_{j=1}^M$ be the left and right singular vectors of the $M \times m$ matrix $\tilde{\mathbf{A}}$. In addition, 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 of $\mathbb{R}^m$ and $\mathbb{R}^M$ respectively. Accordingly, we have that $\{\boldsymbol{\varphi}_j\}$ are the eigenvectors of $\mathbf{W} = \tilde{\mathbf{A}}^T\tilde{\mathbf{A}}$. It implies that $\{\boldsymbol{\varphi}_j\}$ establish re-parametrization 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

$$\boldsymbol{\Phi}_d : \bar{\mathbf{c}}_i \to [\boldsymbol{\varphi}_1(\bar{\mathbf{c}}_i), \ldots, \boldsymbol{\varphi}_d(\bar{\mathbf{c}}_i)]^T . \tag{11}$$

Clearly, (11) can be considered as an approximation of the map (10).

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. As shown in [10], $\{\boldsymbol{\psi}_j\}$ provide re-parametrization of the observations. Moreover, $\{\boldsymbol{\psi}_j\}$ are extensions of $\{\boldsymbol{\varphi}_j\}$ outside the set $\bar{\Gamma}$. Let $\boldsymbol{\Psi}_d$ be an embedding of the observations onto the eigenvectors of $\mathbf{W}$, given by

$$\boldsymbol{\Psi}_d : \mathbf{c}_i \to [\boldsymbol{\psi}_1(\mathbf{c}_i), \ldots, \boldsymbol{\psi}_d(\mathbf{c}_i)]^T . \tag{12}$$

Consequently, the map $\boldsymbol{\Psi}_d$ recovers the independent parameters of the convolution system corresponding

to the observations, up to scaling. See 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 $\{\mathbf{\Psi}_d(\bar{\mathbf{c}}_j)\}$ of $\mathbf{\Psi}_d(\mathbf{c}_i)$ with the Euclidean metric, and let $\{\gamma_j\}$ be interpolation coefficients between $\{\mathbf{\Psi}_d(\bar{\mathbf{c}}_j)\} \in \mathcal{N}_i$ and $\mathbf{\Psi}_d(\mathbf{c}_i)$, given by

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

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

$$\hat{\boldsymbol{\theta}}_i = \sum_{j:\mathbf{\Psi}_d(\bar{\mathbf{c}}_j)\in\mathcal{N}_i} \gamma_j(\mathbf{c}_i)\bar{\boldsymbol{\theta}}_j. \tag{13}$$

Accordingly, let err_{prm} denote the re-parametrization error, defined by

$$\text{err}_{\text{prm}}(\mathbf{c}_i) = \|\boldsymbol{\theta}_i - \hat{\boldsymbol{\theta}}_i\|^2. \tag{14}$$

We note that in case that the parameters of merely few training samples are available, we can re-scale the embedded samples $\{\mathbf{\Psi}_d(\bar{\mathbf{c}}_j)\}$. For that matter, we replace $\bar{\boldsymbol{\theta}}_j$ with the re-scaled $\mathbf{\Psi}_d(\bar{\mathbf{c}}_j)$ in the estimation (13).

C. Setting the algorithm parameters

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

$$\mathbf{\Phi}_d^{-1}(\bar{\boldsymbol{\theta}}) = \sum_{i:\bar{\boldsymbol{\theta}}_i\in\mathcal{B}_{\boldsymbol{\theta}}} \beta_i(\bar{\boldsymbol{\theta}})\bar{\mathbf{c}}_i \tag{15}$$

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

$$\beta_i(\bar{\boldsymbol{\theta}}) = \frac{\exp\left(-\|\bar{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}_i\|^2/\sigma_{\beta_i}\right)}{\sum\limits_{j:\bar{\boldsymbol{\theta}}_j\in\mathcal{B}_{\boldsymbol{\theta}}} \exp\left(-\|\bar{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}_j\|^2/\sigma_{\beta_i}\right)} \tag{16}$$

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 $\mathbf{\Phi}_d(\bar{\mathbf{c}}_i)$ as approximation of $\bar{\boldsymbol{\theta}}_i$ in (16).

Let $\text{err}_{\text{val}}(\bar{\mathbf{c}}_i)$ denote the following *validation error*

$$\text{err}_{\text{val}}(\bar{\mathbf{c}}_i) = \|\bar{\mathbf{c}}_i - \mathbf{\Phi}_d^{-1}(\bar{\boldsymbol{\theta}}_i)\|^2 = \|c(\bar{\boldsymbol{\theta}}_i) - \mathbf{\Phi}_d^{-1}(\bar{\boldsymbol{\theta}}_i)\|^2, \tag{17}$$

Algorithm 1 Re-parametrization 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 the local covariance matrices $\{\boldsymbol{\Sigma}(\bar{\mathbf{c}}_i)\}$ of the m training observations.
4) Compute the affinity matrix $\mathbf{W}$ according to (9), for an arbitrary kernel scale ε.
5) Employ eigenvalue decomposition of $\mathbf{W}$ and obtain the eigenvalues $\{\lambda_j\}$ and the eigenvectors $\{\boldsymbol{\varphi}_j\}$.
6) Construct the map $\boldsymbol{\Phi}_d$ according to (11) to obtain re-parametrization of the independent controlling parameters of the training observations.
7) Construct the inverse map $\boldsymbol{\Phi}_d^{-1}$ according to (15).
8) Find the optimal kernel scale ε that minimizes (17), 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 (7).
2) Calculate $\boldsymbol{\psi}_j$ as a weighted combination of $\boldsymbol{\varphi}_j$ via
$$\boldsymbol{\psi}_j = \frac{1}{\sqrt{\lambda_j}}\tilde{\mathbf{A}}\boldsymbol{\varphi}_j$$
3) Construct the map $\boldsymbol{\Psi}_d$ according to (12) to obtain re-parametrization of the independent controlling parameters of the new observations.
4) Recover the independent parameters according to (13) and compute the mean re-parametrization error (14).

which conveys the accuracy of $\boldsymbol{\Phi}_d^{-1}$ in estimating c. We assume it provides a notion of the accuracy of $\boldsymbol{\Phi}_d$ estimating the inverse map c^{-1}.

The mean error of (17) 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 [24], [25], setting the scale conveys a tradeoff between integration of large number of samples (large scale),

and locality (small scale). We note that this tradeoff is indeed emerged in our empirical testing. In [10], 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, (15) is sufficient.

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

D. Example: autoregressive model

In this section, we recover the parameters of an auto-regressive (AR) system. Consider the following AR process of order d

$$y(t) = x(t) - \sum_{l=1}^{d} \boldsymbol{\alpha}^{(l)} y(t-l) \tag{18}$$

where $x(t)$ is a zero-mean white noise with variance σ_x^2, and $\boldsymbol{\alpha}$ are the AR coefficients. Such AR process is commonly used in many signal processing applications. In particular, it is widely spread in modeling the human vocal tract in speech recognition tasks [26].

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

$$H_{\boldsymbol{\theta}}(\omega) = \frac{1}{1 - \sum\limits_{l=1}^{d} \boldsymbol{\alpha}^{(l)} e^{-jl\omega}}. \tag{19}$$

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

$$H_{\boldsymbol{\theta}}(\omega) = \frac{1}{\prod\limits_{l=1}^{d} \left(1 - \boldsymbol{\theta}^{(l)} e^{-j\omega}\right)}. \tag{20}$$

where $\boldsymbol{\theta}^{(l)}$ are the system poles. Consequently, according to (20), 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

$$P_y(\omega) = \sigma_x^2 \prod_{l=1}^{d} \left|1 - \boldsymbol{\theta}^{(l)} e^{-j\omega}\right|^{-2}. \tag{21}$$

We observe in (21), 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 (21), we can express the covariance function of the output signal as

$$c_y(\tau) = \mathcal{F}^{-1}\left\{P_y(\omega)\right\}, \tag{22}$$

where $\mathcal{F}^{-1}$ denotes the inverse Fourier transform. For simplicity, we omit the explicit expression of the covariance function. In (22) we represent the covariance of the observable signal as a (nonlinearly) 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 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 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 (5). In summary, we have a set of m training parameters in a 2-D parametric space, and a corresponding set of m observations in a 8-D observable space. The observations are obtained via a nonlinear mapping $c : \mathbb{R}^2 \to \mathbb{R}^8$ 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

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

Next, we determine the proper kernel scaling ε. Figure 2 shows the mean validation error (17) 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 trade-off between a small scale for better locality and a large scale for better sample integration is clearly emerged in the curve.

Figure 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,

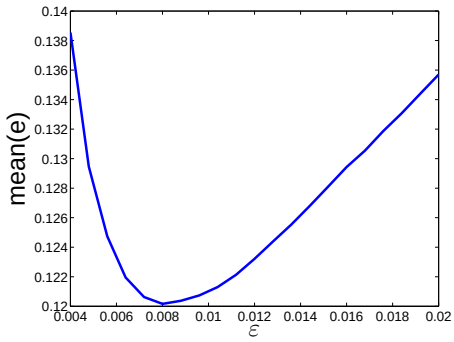

Fig. 2. The mean validation error obtained for all training samples as function of the kernel scale ε.

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

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 indeed be interpreted as a re-parametrization of the pair of poles $\boldsymbol{\theta}_i^{(1)}$ and $\boldsymbol{\theta}_i^{(2)}$.

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

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

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

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

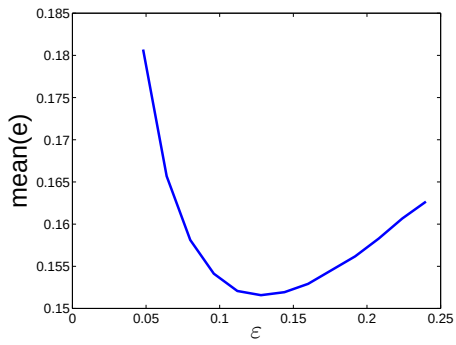

Fig. 5. The mean validation error obtained for all training samples as a function of the kernel scale ε.

implies that the embedding of the extended samples comprises the independent controlling parameters of the AR system as well. To demonstrate the ability to recover the parameters from the obtained re-parametrization, we compute the parametrization error (14). The obtained mean error of the extended samples is $\text{err}_{\text{prm}} = 0.0376$.

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 obtain that the 4th 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 [20], provide estimates of the 4 AR coefficients, but cannot detect the actual degrees of freedom.

In Fig. 5 we show the mean validation error (17) as a function of the kernel scale ε. Accordingly, the kernel scale for this experiment is set to $\varepsilon = 0.13$. We note that a much larger scale is used in this

Fig. 6. Scatter plot of the embedded training samples. (a) Color coding according to the values of the first pole $\boldsymbol{\theta}^{(1)}$. (b) Color coding according to the values of the second pole $\boldsymbol{\theta}^{(2)}$.

Fig. 7. Scatter plot of the embedded samples. (a) Color coding according to the values of the first pole $\boldsymbol{\theta}^{(1)}$. (b) Color coding according to the values of the second pole $\boldsymbol{\theta}^{(2)}$.

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.

Figures 6 and 7 show the same trends as Fig. 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 (14) $\mathrm{err}_{\mathrm{prm}} = 0.0392$. We note that the recovering error value is slightly higher that the mean error achieved in the previous experiment, where the dependency of the observations in the parameters was less complicated.

V. Acoustic Channels

In this section we demonstrate the recovering of independent parameters of acoustic channels. We first describe the simulation model and lay out theoretic background. Then we present some experimental results.

A. The image model

The propagation of a sound wave within an enclosure can be considered linear in case the medium is homogeneous and at rest. 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 [27]. 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, which 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 measured with respect to the microphone position using 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 8 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 $\tau = d/\upsilon = \|\mathbf{r}_p + \mathbf{r}_m\|/\upsilon$.

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

$$h_{\mathbf{r},\mathbf{r_s},\boldsymbol{\beta}}(t) = \sum_{p\in\mathcal{P}} \sum_{m\in\mathcal{M}} \beta_{x_1}^{|m_x - q|} \beta_{x_2}^{|m_x|} \beta_{y_1}^{|m_y - j|} \beta_{y_2}^{|m_y|} \beta_{z_1}^{|m_z - k|} \beta_{z_2}^{|m_z|} \frac{\delta(t-\tau)}{4\pi d} \tag{23}$$

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 that the sampling frequency f_s is sufficiently high, such that $f_s\tau$ is an integer for each delay τ. 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 channels modeling and simulating, we refer the readers to [28], 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, for sampling frequency $f_s = 16$ kHz it corresponds to impulse responses of length ranging between 1600 and 32000. 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 (23) 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 (23), 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 [29], [30], [31]. Usually, a beamformer based on microphone array measurements is implemented [32], [33], [34], [35]. In this work, we show how to recover the source location based on measurements from a single microphone relying on training.

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-dimensional grid. All other 4 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 3m from the microphone, in the same altitude, and with azimuth angle $5\pi/8$. 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, 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 $N = 24000$, 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 $\boldsymbol{\Sigma}(\bar{\mathbf{c}}_i)$ of each training sample is computed.

Now, 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 the parameter and observable spaces, respectively.

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 $c : \mathbb{R}^2 \to \mathbb{R}^{24}$ 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

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

using $\varepsilon = 0.007$. This kernel scale was chosen such that it brings the validation error to a minimum. Figure 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

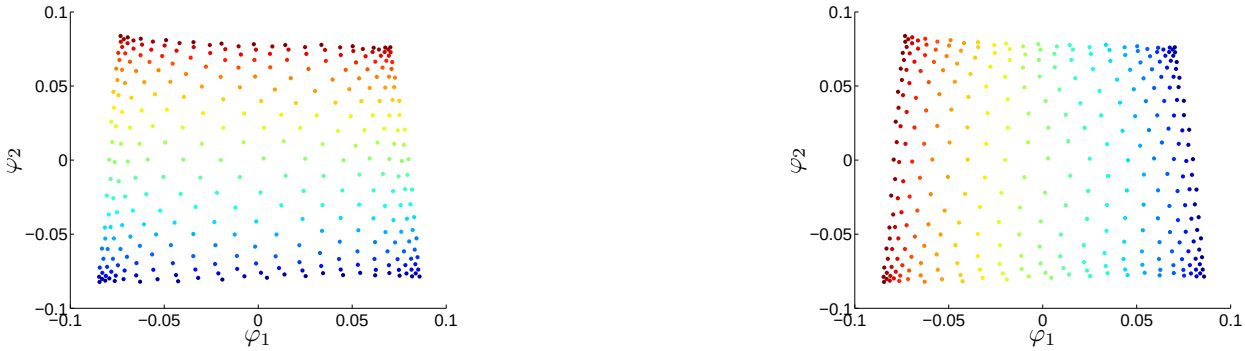

Fig. 8. Scatter plot of the embedding $\mathbf{\Phi_2}$. (a) Color coding according to the values of the azimuth angle. (b) Color coding according to the values of the elevation angle.

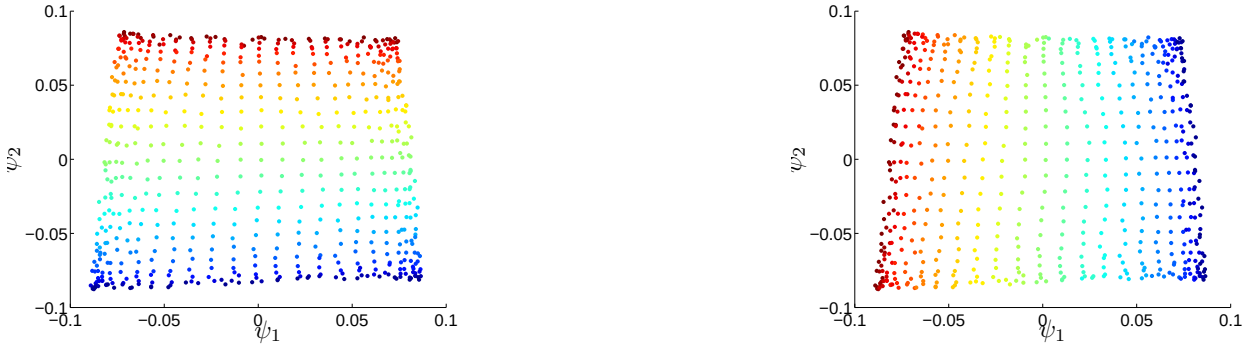

Fig. 9. Scatter plot of the extended embedding Ψ_2. (a) Color coding according to the values of the azimuth angle. (b) Color coding according to the values of the elevation angle.

two reflection coefficients up to scaling.

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

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

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 (13) yields a mean error of $\text{err}_{\text{prm}} = 0.0067$.

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

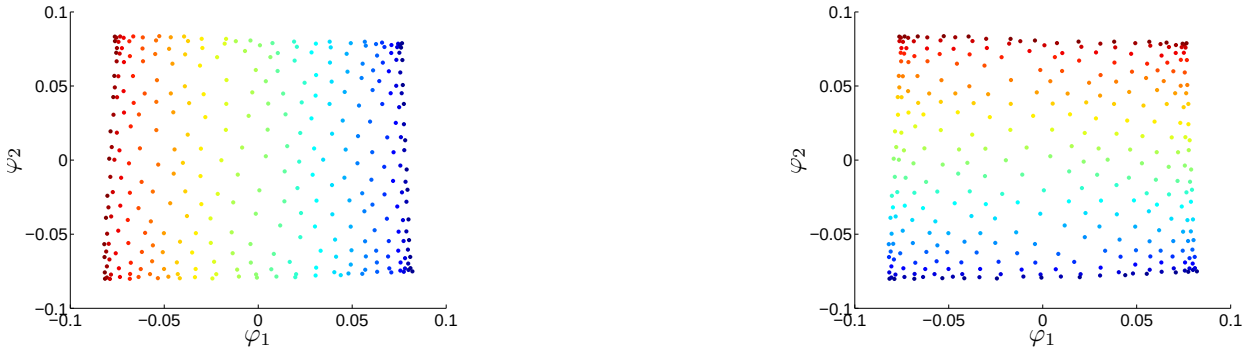

Fig. 10. Scatter plot of $\mathbf{\Phi}_2$. (a) The color coding according to the azimuth angle. (b) The color coding according to the elevation angle.

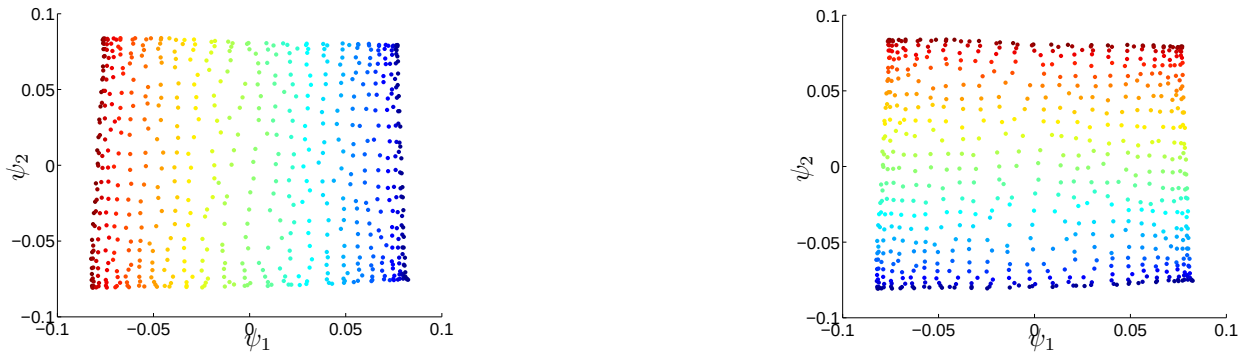

Fig. 11. Scatter plot of $\mathbf{\Phi}_2$. (a) The color coding according to the azimuth angle. (b) The color coding according to the elevation angle.

$m = 300$ source locations on a sector of a sphere around the microphone. The sphere radius is 3m, the sector azimuth and elevation angles range between $[0, \pi/16]$. Therefore, we have approximately a 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.

Figure 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, 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 $\text{err}_{\text{prm}} = 0.0155$. This result is of particular interest, as we accurately recover the direction of arrival of a random source in a room based on observations from a *single* microphone, and training.

VI. Conclusions

We proposed a general algorithm for re-parametrization 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 real examples. We showed that the proposed method can accurately recover the poles of an auto-regressive process and retrieve the controlling parameters of acoustic channels. Acoustic channels are a fundamental component in frontend 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 an auto-regressive 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 at exploring this new lead in order to obtain characterization of, for example, different music tones, various instruments, speech phonemes, or different 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 [36], [37], given by

$$d\boldsymbol{\theta}^{(i)} = \mathbf{a}^{(i)}(\boldsymbol{\theta}^{(i)})dt + \mathbf{b}^{(i)}(\boldsymbol{\theta}^{(i)})d\mathbf{w}^{(i)}, i = 1, \ldots, d \tag{24}$$

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 fluctuations regime conveyed by the Brownian motion and the noise coefficients $\mathbf{b}$; (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 $c : \mathbb{R}^d \to \mathbb{R}^D$. The observed elements $\mathbf{c}^{(j)}$ satisfy the stochastic dynamics given by the Itô lemma [36], [37]

$$d\mathbf{c}^{(j)} = \sum_{i=1}^{d} \left(\frac{1}{2} \left(\mathbf{b}^{(i)} \right)^2 c_{ii}^j + \mathbf{a}^{(i)} c_i^j \right) dt + \sum_{i=1}^{d} \mathbf{b}^{(i)} c_i^j d\mathbf{w}^{(i)}, j = 1, \ldots, D \tag{25}$$

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 (3), using (25), we obtain

$$\boldsymbol{\Sigma}^{(jk)}(\mathbf{c}) = \sum_{i=1}^{d} (\mathbf{b}^{(i)})^2 c_i^j c_i^k. \tag{26}$$

In matrix form, we can express (26) 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 change of variables on (24) 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
Auto-regressive Process Example

We illustrate the settings from Section II by observing the following auto-regressive (AR) process of order 1

$$y(t) = x(t) - \theta y(t-1) \tag{27}$$

where $x(t)$ is 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. By employing the Fourier transform, (27) can be written in the Fourier transform domain with frequency index ω as

$$Y(\omega) = X(\omega)H_{\boldsymbol{\theta}}(\omega)$$

where $X(\omega)$ and $Y(\omega)$ are the Fourier transforms of $x(t)$ and $y(t)$, and $H_{\boldsymbol{\theta}}(\omega)$ is the AR system transfer function, given by

$$H_{\boldsymbol{\theta}}(\omega) = \frac{1}{1-\theta e^{-j\omega}}. \tag{28}$$

Equation (28) can be seen as a sum of a geometric series, yielding the infinite AR system impulse response

$$h_{\boldsymbol{\theta}}(t) = \theta^t,\ t = 0, 1, \ldots. \tag{29}$$

This impulse response demonstrates a nonlinear dependency between the system and the controlling parameter θ. From (1), using (29) the corresponding covariance function of the AR process is given by

$$c_y(\tau) = \sigma_x^2 \frac{\theta^{-|\tau|}}{1-\theta^2}. \tag{30}$$

In (30) 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 (30). Thus, the goal in this work is to recover the controlling parameter θ from such observations.

Appendix III

Euclidean distances on the parametric manifold

We briefly review the derivation of the approximation of the Euclidean distance in the parametric space from [10]. 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 $c : \mathbb{R}^d \to \mathbb{R}^D$. 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 $g : \mathbb{R}^D \to \mathbb{R}^d$ to be the inverse map of $c : \mathbb{R}^d \to \mathbb{R}^D$. We have by the definition of the norm that

$$\left\| \frac{\boldsymbol{\vartheta} - \boldsymbol{\theta}}{2} \right\|^2 = \left\| \boldsymbol{\vartheta} - \frac{\boldsymbol{\theta} + \boldsymbol{\vartheta}}{2} \right\|^2 = \sum_i \left(\boldsymbol{\vartheta}^{(i)} - \frac{\boldsymbol{\theta}^{(i)} + \boldsymbol{\vartheta}^{(i)}}{2} \right)^2. \tag{31}$$

Then, each coordinate of g at $\boldsymbol{\vartheta} = g(\boldsymbol{\gamma})$ can be approximated by a Taylor series at the middle point $(\boldsymbol{\theta} + \boldsymbol{\vartheta})/2$:

$$\begin{aligned} \boldsymbol{\vartheta}^{(i)} &= \tfrac{\boldsymbol{\theta}^{(i)}+\boldsymbol{\vartheta}^{(i)}}{2} + \tfrac{1}{2}\sum_j g_j^i \left(\tfrac{\boldsymbol{\gamma}+\mathbf{c}}{2}\right)\left(\boldsymbol{\gamma}^{(j)} - \mathbf{c}^{(j)}\right) \\ &+ \tfrac{1}{8}\sum_{k,l} g_{kl}^i \left(\tfrac{\boldsymbol{\gamma}+\mathbf{c}}{2}\right)\left(\boldsymbol{\gamma}^{(k)} - \mathbf{c}^{(k)}\right)\left(\boldsymbol{\gamma}^{(l)} - \mathbf{c}^{(l)}\right) + \mathcal{O}\left(\|\boldsymbol{\gamma} - \mathbf{c}\|^3\right) \end{aligned} \tag{32}$$

where g_j^i and g_{kl}^i are the first and second order derivatives of $g^{(i)}$ with respect to $\mathbf{c}^{(j)}$. Similarly to (31) and (32), we approximate g at $\boldsymbol{\theta} = g(\mathbf{c})$ and obtain

$$\left\|\frac{\boldsymbol{\theta} - \boldsymbol{\vartheta}}{2}\right\|^2 = \left\|\boldsymbol{\theta} - \frac{\boldsymbol{\theta} + \boldsymbol{\vartheta}}{2}\right\|^2 = \sum_i \left(\boldsymbol{\theta}^{(i)} - \frac{\boldsymbol{\theta}^{(i)} + \boldsymbol{\vartheta}^{(i)}}{2}\right)^2. \tag{33}$$

and

$$\begin{aligned} \boldsymbol{\theta}^{(i)} &= \tfrac{\boldsymbol{\theta}^{(i)}+\boldsymbol{\vartheta}^{(i)}}{2} + \tfrac{1}{2}\sum_j g_j^i \left(\tfrac{\boldsymbol{\gamma}+\mathbf{c}}{2}\right)\left(\mathbf{c}^{(j)} - \boldsymbol{\gamma}^{(j)}\right) \\ &+ \tfrac{1}{8}\sum_{k,l} g_{kl}^i \left(\tfrac{\boldsymbol{\gamma}+\mathbf{c}}{2}\right)\left(\mathbf{c}^{(k)} - \boldsymbol{\gamma}^{(k)}\right)\left(\mathbf{c}^{(l)} - \boldsymbol{\gamma}^{(l)}\right) + \mathcal{O}\left(\|\boldsymbol{\gamma} - \mathbf{c}\|^3\right) \end{aligned} \tag{34}$$

Combining (31) and (33) yields

$$\begin{aligned} \|\boldsymbol{\vartheta} - \boldsymbol{\theta}\|^2 &= 2\left\|\tfrac{\boldsymbol{\vartheta}-\boldsymbol{\theta}}{2}\right\|^2 + 2\left\|\tfrac{\boldsymbol{\theta}-\boldsymbol{\vartheta}}{2}\right\|^2 \\ &= 2\sum_i \left(\boldsymbol{\vartheta}^{(i)} - \tfrac{\boldsymbol{\theta}^{(i)}+\boldsymbol{\vartheta}^{(i)}}{2}\right)^2 + 2\sum_i \left(\boldsymbol{\theta}^{(i)} - \tfrac{\boldsymbol{\theta}^{(i)}+\boldsymbol{\vartheta}^{(i)}}{2}\right)^2 \end{aligned} \tag{35}$$

and by substituting (32) and (34) into (35), we get

$$\|\boldsymbol{\theta} - \boldsymbol{\vartheta}\|^2 = (\mathbf{c} - \boldsymbol{\gamma})^T \left[\left(\mathbf{J}\mathbf{J}^T\right)^{-1}\left(\frac{\boldsymbol{\gamma} + \mathbf{c}}{2}\right)\right](\mathbf{c} - \boldsymbol{\gamma}) + \mathcal{O}\left(\|\boldsymbol{\gamma} - \mathbf{c}\|^4\right). \tag{36}$$

It is shown in [10] that

$$\left(\mathbf{J}\mathbf{J}^T\right)^{-1}\left(\frac{\boldsymbol{\gamma} + \mathbf{c}}{2}\right) = 2\left[\left(\mathbf{J}\mathbf{J}^T\right)(\mathbf{c}) + \left(\mathbf{J}\mathbf{J}^T\right)(\boldsymbol{\gamma})\right]^{-1} + \mathcal{O}\left(\|\boldsymbol{\gamma} - \mathbf{c}\|^4\right). \tag{37}$$

Now, substituting (37) into (36) yields a second-order approximation of the squared Euclidean distance in the parametric space

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

REFERENCES

[1] A. Singer and R. Coifmain, "Non-linear independent component analysis with diffusion maps," *Appl. Comput. Harmon. Anal.*, 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] A. Singer, "Spectral independent component analysis," *Appl. Comput. Harmon. Anal.*, 2006.

[5] 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.

[6] 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.

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

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

[9] L. Rosasco and M. Belkin, "On learning with intergral operators," *J. Mach. Learn. Res.*, pp. 905–934, 2010.

[10] D. Kushnir, A. Haddad, and R. Coifman, "Anisotropic diffusion on sub-manifolds with application to earth structure classification," *submitted*, 2010.

[11] 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.

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

[13] I. Cohen, "Relative transfer function identification using speech signals," *IEEE Trans. Speech Audio Process.*, vol. 12, pp. 451–459, 2004.

[14] 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.

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

[16] S. Gannot and I. Cohen, "Speech enhancement based on the general transfer function gsc and postfiltering," *Speech and Audio Processing, IEEE Transactions on*, vol. 12, no. 6, pp. 561 – 571, 2004.

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

[18] R. Talmon, I. Cohen, and S. Gannot, "Convolutive transfer function generalized sidelobe canceler," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 17, no. 7, pp. 1420 –1434, 2009.

[19] 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.

[20] J. G. Proakis and Manolakis D. K., *Digital Signal Processing: Principles, Algorithms, and Applications*, 4th edition. Prentice Hall, 2006.

[21] F. R. K. Chung, *Spectral Graph Theory*, CBMS-AMS, 1997.

[22] B. Nadler, S. Lafon, R. Coifman, and I. G. Kevrekidis, "Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators," *Neural Information Process. Systems (NIPS)*, vol. 18, 2005.

[23] 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.

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

[25] A. Singer, "From graph to manifold laplacian: The convergence rate," *Appl. Comput. Harmon. Anal.*, vol. 21, pp. 128–134, 2006.

[26] T. F. Quatieri, *Discrete-time speech signal processing: principles and practice*, Prentice-Hall signal processing series. Prentice Hall, 2005.

[27] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small room acoustics," *Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943950, 1979.

[28] E. A. P. Habets, "Room impulse response (RIR) generator," http://home.tiscali.nl/ehabets/rir_generator.html, Jul. 2006.

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

[30] J. Benesty, "Adaptive eigenvalue decomposition algorithm for passive acoustic source localization," *The Journal of the Acoustical Society of America*, vol. 107, no. 1, pp. 384–391, 2000.

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

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

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

[34] 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.

[35] J. P. Dmochowski and J. Benesty, "Steered beamforming approaches for acoustic source localization," pp. 307–337, 2010.

[36] K. Ito, *On Stochastic Differential Equations*, American Mathematical Society, Memoirs Series, Books on Demand, 1951.

[37] K. Ito and H. P. McKean, *Diffusion Processes and Their Simple Paths*, Academic Press, 1965.