LUND UNIVERSITY

High Resolution Sparse Estimation of Exponentially Decaying N-dimensional Signals

J. SWÄRD, S. I. ADALBJÖRNSSON, AND
A. JAKOBSSON

Published in: Elsevier Signal Processing

doi:10.1016/j.sigpro.2016.04.002

Lund 2016

Mathematical Statistics

Centre for Mathematical Sciences

Lund University

High Resolution Sparse Estimation of Exponentially Decaying N-dimensional Signals ☆

Johan Swärd*, Stefan I. Adalbjörnsson*, Andreas Jakobsson*

Mathematical Statistics, Centre for Mathematical Sciences, Lund University, P.O. Box 118, SE-221 00 Lund, Sweden

Abstract

In this work, we consider the problem of high-resolution estimation of the parameters detailing an N-dimensional (N-D) signal consisting of an unknown number of exponentially decaying sinusoidal components. Since such signals are not sparse in an oversampled Fourier matrix, earlier approaches typically exploit large dictionary matrices that include not only a finely spaced frequency grid, but also a grid over the considered damping factors. Even in the 2-D case, the resulting dictionary is typically very large, resulting in a computationally cumbersome optimization problem. Here, we introduce a sparse modeling framework for N-dimensional exponentially damped sinusoids using the Kronecker structure inherent in the model. Furthermore, we introduce a novel dictionary learning approach that iteratively refines the estimate of the candidate frequency and damping coefficients for each component, thus allowing for smaller dictionaries, and for frequency and damping parameter that are not restricted to a grid. The performance of the proposed method is illustrated using simulated data, clearly showing the improved performance as compared to previous techniques.

Keywords: Sparse signal modeling, spectral analysis, sparse reconstruction, parameter estimation, dictionary learning, damped sinusoids.

☆This work was supported in part by the Swedish Research Council, the Crafoord's and Carl Trygger's foundations, and the Royal Physiographic Society in Lund. This work has been presented in part at the ICASSP and EUSIPCO conferences [1, 2].

*Corresponding author. Phone: +46462228544.

Email addresses: js@maths.lth.se (Johan Swärd), sia@maths.lth.se (Stefan I. Adalbjörnsson), aj@maths.lth.se (Andreas Jakobsson)

1. Introduction

High-dimensional decaying sinusoidal signals occur in a wide variety of fields, such as spectroscopy, geology, sonar, and radar, and given the importance of such signals in a variety of applications, the topic has attracted notable attention in the recent literature (see, e.g. [1–11]). Common solutions include subspace-based algorithms[3–8], which are typically making relatively strong model assumptions, or the use of high-dimensional representations necessitating an iterative zooming procedure over multiple dimensions, such as the technique introduced in [11]. These kind of approaches often suffer from high complexity and sub-optimal performance, typically requiring an accurate initialization or model order information to yield reliable results, information which is commonly not available in many of the discussed applications.

Often, the measurements are also assumed to be uniformly sampled, which may well be undesired in applications such as, for instance, spectroscopy. Furthermore, the number of modes present in the signal is generally unknown, or may vary over time, typically necessitating some form of model order selection decision. Given such difficulties, it is often of interest to formulate non-parametric or semi-parametric modeling techniques, imposing only mild assumptions of the *a priori* knowledge of the signal structure. Popular solutions include the so-called dCapon, dAPES, and dIAA spectral estimators, which all form generalized spectral estimates of the signal, constructing spectral representations over both the frequency and damping dimensions [12, 13] (see also [14, 15]). Although this form of techniques are robust to the made model order assumptions, they suffer difficulties in separating closely spaced modes from each other, and typically require notable computational efforts if not implemented carefully [15].

As an alternative, one may use sparse modeling of the signal, forming a large dictionary of all potential frequencies and damping candidates, thus generally having vastly more columns than rows. For a given signal and the resulting

dictionary matrix, one thus wishes to find the sparsest solution to the resulting linear set of equations, mapping the signal to a linear combination of a few of the columns of the dictionary. Such techniques have successfully been applied to line spectral data, and the topic has attracted notable attention in the recent literature (see, e.g., [16–22]). Although these algorithms appear quite different from each other, they share the property that the considered dictionary grid should be selected sufficiently fine to allow for a sparse signal representation (see also [23, 24]), which, if extended to also consider damped modes, necessitates a large dictionary matrix containing elements with a sufficiently fine grid over the range of both the potential frequencies and damping candidates (see, e.g., [13, 25, 26]); this will be particularly noticeable if treating large data sets, or data sets with multiple measurement dimensions. In order to mitigate this problem, we here introduce a tensor representation of the signal model, allowing us to exploit the resulting inherent Kronecker structure, which may be exploited to significantly reduce the required complexity as compared to a naive implementation of the sparse modeling framework.

Furthermore, we propose a novel dictionary learning approach, wherein one iteratively decomposes the signal with a fixed small dictionary, adaptively learning the dictionary elements best suited to enhance sparsity. To this effect, we initially form a coarsely spaced dictionary with undamped modes over the range of considered frequency candidates, iteratively adapting both the frequency and damping settings for the dictionary elements, thereby also allowing for both a reduction and an expansion of the number of dictionary elements considered in each iteration of the optimization. In order to further reduce complexity, we propose a computationally efficient implementation based on the concept of the alternating direction method of multipliers (ADMM) (see, e.g., [27]), where the Kronecker structure of the resulting dictionary matrices may be exploited to dramatically decrease the cost of each iteration.

The remainder of the paper is organized as follows: in the next section, we introduce the considered data model. Then, in Section 3, we introduce the idea behind decoupling the search dimensions. Section 4 introduces the

ADMM formulation of the estimator, and Section 5 illustrates the performance of the proposed estimator using simulated data. Finally, Section 6 contains our conclusions. In the remainder of the paper, we use the following notation: scalars are represented using lower case letters, whereas vectors are represented with lower case bold-face letters. Matrices are represented with capital bold-face letters, tensors with capital bold Euler script letter, $(\cdot)^T$ denotes the transpose, and $(\cdot)^H$ the conjugate transpose.

2. *N*-Dimensional Signal model

Consider an N-dimensional signal consisting of a sum of K modes, i.e., K N-dimensional damped sinusoids such that observation $x_{\boldsymbol{\tau}}$ at a sampling point τ, where

$$\boldsymbol{\tau} = \begin{bmatrix} t_{i_1}^{(1)} & t_{i_2}^{(2)} & \dots & t_{i_N}^{(N)} \end{bmatrix}^T \tag{1}$$

and $t_{i_\ell}^{(\ell)}$ denotes the i_ℓ:th sampling point in dimension ℓ, may be well modeled as

$$x_{\boldsymbol{\tau}} = \sum_{k=1}^{K} g_k \prod_{\ell=1}^{N} \xi_{k,\ell}^{t_{i_\ell}^{(\ell)}} + \epsilon_{\boldsymbol{\tau}} \tag{2}$$

where

$$\xi_{k,\ell} = e^{j\omega_k^{(\ell)} - \beta_k^{(\ell)}} \tag{3}$$

and with g_k denoting the complex amplitude of mode k, and $\epsilon_{\boldsymbol{\tau}}$ is an additive noise term, here for simplicity assumed to be an independent identically distributed circularly symmetric Gaussian random variable. Assuming the signal is observed over $t_{i_n}^{(n)}$, for $i_n = 1, \dots, I_n$, and $n = 1, \dots, N$, the entire sequence may be stored in an N-way tensor $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{I_1 \times I_2 \times \cdots \times I_N}$. It is worth noting that this formulation makes no restriction on any of the dimensions to have a sampling scheme that is equidistant, thus encompassing both missing data scenarios as well as irregular sampling. The entire model may thus be written in tensor

format as the sum of K rank one tensors, such that

$$\boldsymbol{\mathcal{X}} = \sum_{k=1}^{K} g_k \tilde{\mathbf{a}}^{(1)}(k) \circ \tilde{\mathbf{a}}^{(2)}(k) \cdots \circ \tilde{\mathbf{a}}^{(N)}(k) + \boldsymbol{\mathcal{E}} \tag{4}$$

where $\circ$ denotes the outer product defined such that element $\boldsymbol{\tau}$ of $\boldsymbol{\mathcal{X}}$ corresponds to equation (2), $\boldsymbol{\mathcal{E}}$ is the tensor containing the noise terms, and

$$\tilde{\mathbf{a}}^{(n)}(k) = \left[\begin{array}{ccc} \xi_{k,n}^{t_1^{(n)}} & \cdots & \xi_{k,n}^{t_{I_n}^{(n)}} \end{array} \right]^T \tag{5}$$

For an overview of tensor algebra sufficient for the here discussed results see, e.g., [28], which also use a notation consistent with the one used in this article. The model thus contain $(2N+1)K+1$ unknown parameters, namely

$$\boldsymbol{\theta} \triangleq \left[\begin{array}{cc} \{\{\omega_k^{(n)}, \beta_k^{(n)}\}_{n=1}^{N}, g_k\}_{k=1}^{K}, & K \end{array} \right]^T \tag{6}$$

of which $2NK$ are non-linear parameters. Clearly, one could, in theory, form a non-linear least squares (LS) minimization over these parameters, as well as form a model order estimate from the resulting model order residuals for varying possible candidate model sizes. However, such a solution would in most practical situations be computationally unfeasible, even for low dimensional data sets, especially as the optimization is well known to have numerous local minima [29]. To avoid this, we introduce a sparse modeling heuristic to approximate the model. This can be done by creating a large dictionary of candidate parameters, selected from a grid fine enough such that each true parameter lies sufficiently close to some grid point. For instance, if, to simplify our notation, one considers a single N-dimensional sinusoid and fix all but the first frequency and damping coefficients, then one may approximate (4) using a dictionary containing P_1 and J_1 candidate elements along the (first) frequency and damping dimension, respectively, such as

$$\boldsymbol{\mathcal{X}} \approx \sum_{p=1}^{P_1} \sum_{j=1}^{J_1} g_{p,j} \mathbf{a}_{\omega_p}^{(1)}(\beta_j) \circ \mathbf{a}_{\omega_2}^{(2)}(\beta_2) \circ \cdots \circ \mathbf{a}_{\omega_N}^{(N)}(\beta_N) \tag{7}$$

where $\omega_2, \ldots, \omega_N$ and $\beta_2, \ldots, \beta_N$ denote the (for simplicity) fixed frequency and damping coefficients along the 2nd to N:th dimensions,

$$\mathbf{a}_{\omega}^{(n)}(\beta) = \left[\begin{array}{ccc} \xi_n^{t_1^{(n)}} & \cdots & \xi_n^{t_{I_n}^{(n)}} \end{array} \right]^T$$

where

$$\xi_n = e^{j\omega^{(n)} - \beta^{(n)}} \tag{8}$$

and $g_{k,\ell}$ denotes the contribution of each of these dictionary elements in the approximation. Thus, as long as P_1 and J_1 are selected sufficiently large to allow for a grid of dictionary elements such that the true frequency and damping coefficients lie close to one of the grid points, only one $g_{p,j}$ should be non-zero for each of the K modes. By similarly extending the dictionary for each of the frequency and damping dimensions, such that $g_{p_1,\ldots,p_N,j_1,\ldots,j_N}$ denotes the contribution of the corresponding dictionary elements for the p_q:th and j_r:th frequency and damping dictionary elements, where $q, r \in \{1, \ldots, N\}$, the resulting (very large) dictionary would allow for a sparse approximative solution of the unknown parameters, such that most of the dictionary elements would not contribute to the approximation. Given such an approximative solution, the number of modes, K, may be estimated as the number of elements with non-zero contribution to the approximation. The non-linear parameters may then be estimated correspondingly, such that for any non-zero variables, e.g., $g_{h_1,\ldots,h_N,i_1,\ldots,i_N}$, the non-linear parameters are estimated as the frequency and damping coefficient that correspond to the found coefficients. Such a solution may be obtained by reformulating the problem using the *vec* operator, defined here for tensors such that it is the usual *vec* operation on the mode-1 matricization, or unfolding (see also [28]), of a given tensor, i.e.,

$$vec\left(\mathcal{X}\right) \triangleq vec\left(\mathbf{X}_{(1)}\right) \tag{9}$$

This allows for a sparse LS solution to be found by solving

$$\min_{\tilde{\mathbf{g}}} \|vec\left(\mathcal{X}\right) - \tilde{\mathbf{A}}\tilde{\mathbf{g}}\|_2^2 + \rho(\tilde{\mathbf{g}}) \tag{10}$$

where $\tilde{\mathbf{g}} = vec\left(\mathcal{G}\right)$, with $\mathcal{G} \in \mathbb{C}^{P_1 \times \cdots \times J_N}$ denoting the tensor formed from the amplitudes of all of the dictionary elements, and the i:th column of $\tilde{\mathbf{A}}$ is formed as

$$\tilde{\mathbf{A}}_{:i} = vec\left(\mathbf{a}^{(1)}_{\omega_{p_1}}(\beta_{j_1}) \circ \mathbf{a}^{(2)}_{\omega_{p_2}}(\beta_{j_2}) \cdots \circ \mathbf{a}^{(N)}_{\omega_{p_N}}(\beta_{j_N})\right) \tag{11}$$

where the notation $\mathbf{A}_{:i}$ denotes the ith column of the matrix $\mathbf{A}$. The penalty term $\rho(\cdot)$ is added in (10) as the grid is typically chosen such that the number of elements in $vec(\mathcal{X})$ is smaller than the number of columns in $\tilde{\mathbf{A}}$; thus, if assuming that $\tilde{\mathbf{A}}$ is of full rank, the system of equations is under-determined, with infinitely many solution, out of which one is interested in finding one that appropriately weighs sparsity and model fit. Ideally, $\rho(\cdot)$ could be chosen as a function counting the number of non-zero elements. However, the resulting optimization problem is well known to be combinatorial in nature and will be unfeasible to solve even for moderate problem sizes. Common approximative choices include the scaled ℓ_1 norm [17, 30], ℓ_q penalties [16, 31], and the reweighted ℓ_1 approach, which may be seen to correspond to the log penalty [32]. Herein, we consider the ℓ_1 and the log penalty. It is worth noting that the above sparsity restrictions allow for solutions having multiple damping coefficients for a given frequency. Such solutions imply that the component is not an exponentially damped sinusoid; as this is not relevant for the here considered application, we proceed to refine the constraint such that it will only yield unique frequency-damping pairs for each component. To this end, we propose an iterative dictionary learning approach such that the damping parameters for each sinusoidal component is held fixed during the sparse LS step, after which the damping parameters are found using the residual from the sparse LS step, one mode at the time, thus allowing for damping and frequency estimation to be performed with a non-linear optimization algorithm, e.g., Newton's method. Thus, we initially fix all damping parameters to zero, modifying (7) such that the dictionary is only formed over the unknown frequencies, i.e.,

$$\mathcal{X} \approx \sum_{p_1=1}^{P_1} \cdots \sum_{p_N=1}^{P_N} g_{p_1,\ldots,p_N} \mathbf{a}^{(1)}_{\omega_{p_1}}(\beta_{p_1}) \circ \cdots \circ \mathbf{a}^{(N)}_{\omega_{p_N}}(\beta_{p_N}) \tag{12}$$

The resulting minimization with respect to the K unknown frequencies, which may then be used to estimate the damping components, iteratively finding each

of the set of estimates. To allow for a computationally efficient solution, the considered frequency and damping grids, respectively, are updated in each iteration, such that the dictionary is refined in each step of the iteration. However, even with such a reduction in complexity, the iterative optimization problems are clearly daunting, being formed over $J_1 \times \cdots \times J_N$ and $P_1 \times \cdots \times P_N$ dimensions, respectively. In the next two sections, we therefore proceed to examine how these minimizations may be performed in an efficient manner utilizing the Kronecker structure of the dictionary matrices for the sparse LS step, and by solving the non-linear damping parameter estimation one mode at a time.

3. ADMM implementation

The minimization problem considered in (10) may be solved using an approximation of the form

$$\min_{\tilde{\mathbf{g}}} \|vec(\mathcal{X}) - \tilde{\mathbf{A}}\tilde{\mathbf{g}}\|_2^2 + \sum_{k=1}^{P_1 \times \cdots \times J_N} \lambda_k |\tilde{g}_k| \tag{13}$$

where λ_k denotes a set of tuning parameters, for $k = 1, \ldots, P_1 \times \cdots \times J_N$. In case these tuning parameters are all selected equal and the penalty is included as an inequality constraint, the resulting minimization is equivalent with the regular ℓ_1 penalized LS problem, often called basis pursuit denoising [33], or the LASSO [30]. For highly correlated dictionary elements, as may be required for high resolution N-D spectra, one may obtain sparser solutions using a reweighted LASSO formulation [32], such that the λ_k:s are instead selected as

$$\lambda_k = \frac{\phi}{|\tilde{g}_k(\ell)| + \varepsilon} \tag{14}$$

where the constant ϵ is included to avoid numerical problems when $g_k(\ell)$ is close to zero. Here, $\tilde{g}_k(\ell)$ denotes the value of g_k at iteration ℓ, and with $\phi > 0$ denoting a tuning parameter controlling the sparsity at the solution. A general efficient iterative algorithm for solving problems such as (10), using an ADMM implementation was proposed in [27], and may be easily adapted to

the here considered reweighted scenario. The steps involved are summarized in Algorithm 1, where the Ψ operator is a shrinkage operator, defined as

$$\Psi(\mathbf{x}, \gamma) = \mathbf{x}(1 - \gamma/|\mathbf{x}|)^{+} \tag{15}$$

where $(\cdot)^{+}$ denotes the positive part of a scalar. In Algorithm 1, $\tilde{\mathbf{g}}$ has been split up into two separate variables $\mathbf{z}$ and $\mathbf{u}$. Furthermore, $\mathbf{d}$ denotes the scaled dual variable, (see, e.g., [27] for a detailed discussion). The complexity of each iteration in the resulting algorithm is approximately $\mathcal{O}(n^2 p)$, where p and n denote the columns and rows of $\mathbf{a}$, respectively. This is about the same as the computational cost for many LASSO solvers (see e.g. [34]). In the N-dimensional case, the overall computational complexity is about $\mathcal{O}(\prod_{n=1}^{N} J_n P_n \prod_{n=1}^{N} I_n^2)$, implying that even a 3-dimensional problem with 100 grid points in each dimension would result in a cost of approximately $100^{12} I_1 I_2$ operations, in each step, where I_n denotes the number of samples in dimension n. Fortunately, this complexity may be significantly reduced by exploiting the inherent Kronecker structure of the model. In order to do so, we rewrite (4) using tensor products as

$$\boldsymbol{\mathcal{X}} \;=\; \boldsymbol{\mathcal{G}} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)} + \boldsymbol{\mathcal{E}} \tag{16}$$

where the operator $\times_n$ represents the n-mode product of a tensor with a matrix, and the dictionary matrix for dimension n is given as

$$\mathbf{A}^{(n)} \triangleq \left[\begin{array}{ccc} \mathbf{a}_{\omega_{k_1}}^{(n)}(\beta_{k_1}) & \ldots & \mathbf{a}_{\omega_{K_1}}^{(n)}(\beta_{K_1}) \end{array} \right] \tag{17}$$

Expressed in this form, one may note that the matricization may be accomplished via Kronecker products instead (see, e.g.,[28], [35]), yielding

$$\mathbf{X}_{(1)} = \mathbf{A}^{(1)} \mathbf{G}_{(1)} \left(\mathbf{A}^{(N)} \otimes \mathbf{A}^{(N-1)} \otimes \cdots \otimes \mathbf{A}^{(2)} \right)^{T} \tag{18}$$

where $\otimes$ denotes the Kronecker product, and $\mathbf{X}_{(1)} \in \mathbb{C}^{I_1 \times \prod_{n=2}^{N} I_n}$ is obtained by stacking all the mode-1 slices of $\boldsymbol{\mathcal{X}}$, and with $\mathbf{G}_{(1)}$ defined similarly. Vectorizing the resulting mode-1 slices yields (see, e.g.,[36]),

$$\mathrm{vec}\left(\mathbf{X}_{(1)}\right) = \left(\mathbf{A}^{(N)} \otimes \cdots \otimes \mathbf{A}^{(2)} \otimes \mathbf{A}^{(1)} \right) \mathrm{vec}\left(\mathbf{G}_{(1)}\right) \tag{19}$$

allowing us to express the parameters in (10) as

$$\tilde{\mathbf{g}} \triangleq \text{vec}\left(\mathbf{G}_{(1)}\right) \in \mathbb{C}^{\tilde{K}\times 1} \tag{20}$$

$$\tilde{\mathbf{A}} \triangleq \left(\mathbf{A}^{(N)} \otimes \cdots \otimes \mathbf{A}^{(2)} \otimes \mathbf{A}^{(1)}\right) \in \mathbb{C}^{\tilde{I}\times \tilde{K}} \tag{21}$$

As a result, the full $\tilde{\mathbf{A}}$ matrix does not need to be formed, and vector multiplication of the form $\tilde{\mathbf{A}}\mathbf{x}$ and $\tilde{\mathbf{A}}^H\mathbf{y}$, for any appropriately sized vector $\mathbf{x}$ and $\mathbf{y}$, may be computed iteratively by each sub-matrix $\mathbf{A}^{(n)}$, and by then reshaping the resulting elements (see, e.g.,[37, p. 28] for further details). This allows for a dramatic complexity reduction. To illustrate this, consider the case where each $\mathbf{A}^{(\ell)}$ matrix is $n \times n$. Then, the operation $\tilde{\mathbf{A}}\mathbf{x}$, which would require about $\mathcal{O}(n^{2N})$ multiplications if first forming $\tilde{\mathbf{A}}$ and then computing the inner-product using this matrix, may instead be formed using only $\mathcal{O}(Nn^{N+1})$ (see, e.g., [38]) operations. Furthermore, the LS step in the ADMM algorithm for solving (10) may also be computed significantly cheaper by utilizing its Kronecker structure, simply by calculating the singular value decomposition of each sub-matrix $\mathbf{A}^{(n)} = \mathbf{U}_n\mathbf{\Sigma}_n\mathbf{V}_n^H$, and then utilizing that the singular value decomposition of $\tilde{\mathbf{A}}$ is given by (see, e.g., [36, p. 246])

$$\tilde{\mathbf{A}} = \mathbf{U}_{\tilde{\mathbf{A}}}\mathbf{\Sigma}_{\tilde{\mathbf{A}}}\mathbf{V}_{\tilde{\mathbf{A}}}^H \tag{22}$$

where

$$\mathbf{U}_{\tilde{\mathbf{A}}} = \mathbf{U}_1 \otimes \cdots \otimes \mathbf{U}_N \tag{23}$$

$$\mathbf{\Sigma}_{\tilde{\mathbf{A}}} = \mathbf{\Sigma}_1 \otimes \cdots \otimes \mathbf{\Sigma}_N \tag{24}$$

$$\mathbf{V}_{\tilde{\mathbf{A}}}^H = \mathbf{V}_1^H \otimes \cdots \otimes \mathbf{V}_N^H \tag{25}$$

As a result, one may solve step 3 in Algorithm 1 by solving the equivalent LS problem

$$\min_{\tilde{\mathbf{z}}} \left\| \begin{bmatrix} \mathbf{U}_{\tilde{\mathbf{A}}}^H\mathbf{y} \\ \mathbf{V}_{\tilde{\mathbf{A}}}^H\boldsymbol{\xi} \end{bmatrix} - \begin{bmatrix} \mathbf{\Sigma}_{\tilde{\mathbf{A}}} \\ \sqrt{\mu}\mathbf{I} \end{bmatrix} \tilde{\mathbf{z}} \right\|_2^2 \tag{26}$$

where

$$\tilde{\mathbf{z}} = \left(\mathbf{\Sigma}_{\tilde{\mathbf{A}}}^2 + \mu\mathbf{I}\right)^{-1}\left(\mathbf{\Sigma}_{\tilde{\mathbf{A}}}\mathbf{U}_{\tilde{\mathbf{A}}}^H\mathbf{y} + \sqrt{\mu}\mathbf{V}_{\tilde{\mathbf{A}}}^H\boldsymbol{\xi}\right) \tag{27}$$

with $\tilde{\mathbf{z}} = \mathbf{V}_{\tilde{\mathbf{A}}}^H \mathbf{z}$ and $\boldsymbol{\xi} = \sqrt{\mu}(\mathbf{u}(\ell) - \mathbf{d}(\ell))$. Thus, the LS step can be solved by three matrix vector multiplications, two Hadamard products between vectors, one scalar multiplication of a vector, and a vector-vector addition, which may all be calculated using their inherent Kronecker structure, significantly reducing the computational cost. For example if each $\mathbf{A}^{(\ell)}$ is $n \times n$, the cost for our approach is approximately $\mathcal{O}(Nn^{N+1})$ versus $\mathcal{O}(n^{3N})$ for a solution that does not use the inherent structure of the equations.

4. Sparse dictionary learning

As noted above, the considered grid over the candidate frequency and damping coefficients are updated in alternating fashion. Let $\hat{K}$ denote the number of non-zero amplitudes after the sparse LS step. Then, the dictionary learning may be done by forming the residual[1]

$$\boldsymbol{\mathcal{R}} = \boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{G}} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)} \tag{28}$$

Using a relaxation-based procedure (see also [39]), one then iteratively adds back one mode at a time to the residual in (28), and form an estimate of the frequency and damping of this mode using an N-dimensional single mode solver, such as, for instance, the standard nonlinear least squares estimator or, in the case of uniformly sampled data, an estimator such as the PUMA estimator [40]. Using the refined parameter estimates, the mode is then subtracted again, and the next mode is refined similarly. The procedure is summarized in Algorithm 2. Using the refined modes, the dictionary is then updated, such that it is separated into N dictionaries, one over each dimension, with each dictionary being centered in a fine grid around each of the found frequencies. As a result, the unused dictionary elements, having zero-amplitudes, are excluded from the updated dictionary (unless being close to one of the found modes). This also implies that closely spaced modes may yield overlapping dictionary elements;

[1] To simplify our notation, we have here suppressed the dependencies on the frequency ω and the damping β.

such duplicated dictionary elements are removed to avoid collinearity in the dictionary. For each grid point, the dictionary element is scaled according to the found damping coefficient of the corresponding mode, to ensure that all dictionary elements have the same norm, thus refining the dictionary iteratively over both frequencies and damping coefficients. We coin the resulting method the Sparse Exponential Mode Analysis (SEMA) algorithm.

5. Numerical examples

We proceed to examine the performance of the proposed method using simulated data. To simplify the presentation, we focus on the 1-D, 2-D, and 3-D cases, since problems of these dimensions offer more intuitive results that are also easier to analyze. Considering first the 1-D case, we illustrate the performance of the proposed method using simulated data. We initially consider a data vector containing $N = 128$ samples of a three mode signal, where the frequencies and damping parameters are chosen uniformly over $[0,\ 1]$ and $[0,\ 0.025]$, respectively. We note that we here use normalized frequencies, lying in the interval $[0,\ 1]$, denoted by the letter f. For now, we ensure that no modes are closer in frequency than $1/N$. Figures 1 and 2 depict the resulting performance of the SEMA algorithm, as compared to the non-parametric damped-Capon (dCapon) estimate [12, 15], as a function of the signal-to-noise-ratio (SNR), defined as $\log_{10}(||\mathbf{y}||_2^2/N\sigma^2)$, where σ^2 denotes the variance of the noise. The two figures show the root mean squared error (RMSE) of the frequency and damping estimates, defined as

$$\text{RMSE} = \sqrt{\frac{1}{MK}\sum_{m=1}^{M}\sum_{k=1}^{K}\left(\theta_{m,k} - \hat{\theta}_{m,k}\right)^2} \tag{29}$$

where $\theta_{m,k}$ denotes the estimate of either the frequency or the damping of mode k for Monte-Carlo simulation m, M is the total number of Monte-Carlo simulations, and K the number of modes. These results have been obtained using $M = 175$ Monte-Carlo simulations. In this example, dCapon has a frequency

grid that is selected to be 6000×6000, uniformly covering frequencies and damping factors in $[0, 1]$ and $[0, 0.025]$, respectively, and where the recommended filter length of $N/4$ is used. The SEMA algorithm on the other hand uses a dictionary containing only 128 elements in the first iteration, and, thereafter, uses only 40 grid points for each found mode when updating the dictionary in each subsequent iteration. As can be seen from the figures, the proposed SEMA algorithm yields notably better estimates than the dCapon estimator, without requiring a large dictionary grid over both dimensions, thereby allowing for a substantially faster implementation. It is also worth noting that the dCapon estimation errors are here larger than the smallest possible error that is attainable given the current grid size, implying that the grid size does not in itself limit the quality of the estimates.

Next, we examine the ability of the methods to resolve two closely spaced spectral lines. In this case, we consider a signal containing two sinusoidal components with frequencies, $f_1 = 0.6417$ and $f_2 = 0.6456$, i.e., separated by $0.5/N$, with damping constants being 0.010 for both modes. Figure 3 illustrates the resulting frequency estimates as obtained from 5 Monte-Carlo simulations, and SNR = 20 dB. For comparison, the figure also shows the estimates obtained using 1-D SEMA, dCapon, dIAA [41], and for a Lasso method with a dictionary containing both frequencies and damping factors, and exploiting a zooming similar to the one used in SEMA. Here, to speed-up the computations, the frequency grid for dCapon has been selected to only be formed on $[0.63, 0.67]$, allowing the method notable *a priori* information on the frequency region of interest. The damping grid ranges over $[0, 0.025]$ and has size 500 for all methods, except for the used Lasso method, where, due to complexity reasons, it is set to 10. As seen in the figure, the proposed method clearly manages to resolve the two peaks, whereas the Lasso and dIAA estimates are only partly succeeding to do so, while dCapon yields noticeably biased estimates. In the figure, the (red) square indicates the region $1/(2N)$ centered around the true frequencies.

We proceed to examine the performance of the SEMA algorithm for 2-D sim-

ulated data, examining the RMSE of two well separated peaks, showing that the proposed method has similar performance to the statistically efficient PUMA method [7], using simulated data mimicking a 2-D Nuclear Magnetic Resonance (NMR) signal, simulated using (2), containing two damped sinusoids and having 33×31 samples. Figures 4-7 illustrates the performance of the SEMA estimator as compared to the parametric PUMA estimator and the corresponding Cramér-Rao lower bound (CRLB) [42]. The frequencies were randomly selected in the interval from 0 to 1 in normalized frequencies, and selected such that components were separated by at least $3/N$ in each dimension. If the spacing between the peaks is smaller, the estimation will degenerate for all methods. The damping parameters were set to $\beta_1 = (0.05 \quad 0.02)$ and $\beta_2 = (0.01 \quad 0.04)$ for all simulations. Each mode was normalized in amplitude, thus making sure that both peaks were equally dominant. The PUMA algorithm was, as for all examples, allowed 100 iterations, as well as oracle model order information, and the initial grid for the proposed 2-D method was, as for the following examples, set to 100. The proposed method was allowed two iterations and used 33 grid points to zoom in on each found mode. The choice of λ governs the number of peaks that may be found. If set too high, peaks with low amplitude will be suppressed, and if set too low, peaks that originate from the noise will not be suppressed. However, due to the reweighting step, a too small λ will be compensated for, and therefore the algorithm is relatively robust to the choice of λ, as long as it is not set too large. Therefore, it is preferable to set λ to a small value. In these examples, we set λ equal to the tenth largest peak found in the periodogram. One could argue that we thereby limit the number of peaks that may be found, but that is easily avoided. If λ were set to equal the amplitude of the r:th largest peak and, when using the method, we found r peaks, one would run the algorithm a second time but with a somewhat smaller λ value. In this way, we make sure that we do not in fact limit the algorithm to a specified number of peaks. The test was performed using 250 Monte-Carlo simulations, for each value of the considered SNR. Figures 4-7 illustrate the total RMSE of all the unknown parameters. As can be seen from the figure, both the para-

metric PUMA, which has been allowed oracle model order information, and the proposed semi-parametric SEMA algorithms yield statistically efficient parameter estimates especially for larger larger SNR. Here, if the proposed algorithm did not manage to estimate the number of modes correctly, that estimate was then removed from the RMSE calculations for all methods. This happened two times out of 1500 Monte-Carlo simulations. The average computation times for 100 simulations on SNR level 20 was around 0.6 seconds for SEMA and 0.005 seconds for PUMA.

We proceed to examine the methods ability to resolve two closely spaced peaks. This was done by fixing the first mode at frequency $f_1 = (0.4, \quad 0.6)$, and letting the second mode gradually approach the first. The modes were initially separated by $1/N_1$ and $1/N_2$ in each frequency dimension, and the test was stopped when the modes were separated by $0.1/N_1$ and $0.1/N_2$. The data size for this example was again 33×31. The same SEMA settings as above were used. We also compare the estimates to that of a zero-padded 2-D periodogram, where 2^{13} zeros were padded in each dimension, but zoomed in on the correct frequencies (± 0.1 in each frequency). The damping parameters where fixed to 0.02 for all modes and dimensions, and the SNR was set to 10 dB. Furthermore, PUMA was again allowed complete knowledge of the number of peaks. To determine whether or not two peaks were resolved, we ensured that the method fulfilled at least two separation criteria: First, the peaks that were found had to be at least within a rectangle of size $1/N_1 \times 1/N_2$ from the correct frequencies; Secondly, the power of the valley between the peaks were allowed to be at most 90% of the average power of the peaks. If these two criteria were met, the modes were deemed to be resolved. The results are shown in Figure 8, where the x-axis should be interpreted as the distance divided by N_1, i.e., 0.1 means that the distance between the modes is $0.1/N_1$. As may be seen from the figure, the periodogram's ability to distinguish the two modes drastically decreases as the modes become closer. As may be expected. the PUMA method on the other hand manages to separate the modes very well until they are about 0.3 apart from each other. As can be seen from the figure,

the SEMA method achieves about the same performance as PUMA until the distance is less than 0.4. It should be stressed that the PUMA estimator is given perfect prior knowledge about the number of modes, whereas the 2-D SEMA has no such prior information. As is clear from the figure, the SEMA estimate seems to be able to separate closely spaced modes almost as well as the parametric and statistically efficient PUMA estimator, without imposing any a priori model order information, as well as yielding far better performance than the periodogram estimate. A typical result is shown in Figures 9 and 10, where the peaks are separated by $0.5/N1$. It clearly shows how SEMA manages to separate the two peaks, whereas the periodogram only shows one peak.

In the next example, we investigate how well SEMA works on non-uniformly sampled data. We made 100 Monte-Carlo simulations on a simulated NMR signal containing 33×31 sample points, where the second dimension was sampled uniformly and the first dimension was sampled in a non-uniformly manner, mimicking a typical high-dimensional NMR experiment. The frequency was randomly selected and separated at least $3/N_1$ from each other, whereas the β parameters were set to $\beta_1 = (0.01, 0.02)$ and $\beta_2 = (0.04, 0.03)$ throughout the simulation. Again, each mode was normalized in amplitude. In each dimension, 100 frequency grid points were used, and SEMA was allowed one iteration. Since PUMA does not work with non-uniformly sampled data, we instead applied an NLS search for the frequency and damping parameters in the mode estimation stage (Algorithm 2). Figure 11 shows the result where the frequency and damping parameters RMSE are shown together with the corresponding CRB.

Finally, to also illustrate the performance for higher dimensional data, we examine a 3-D data sets containing two unit amplitude damped modes at frequencies drawn uniformly from $(0, 1)$, with damping parameters fixed to $\beta_1 = (0.01, 0.06)$, $\beta_2 = (0.02, 0.05)$, and $\beta_3 = (0.03, 0.04)$, and having $13 \times 13 \times 13$ samples. The modes were created so that they were separated at least by $1/N_1$ in all dimensions. The summed RMSE of the six frequency components was computed using 100 Monte-Carlo simulations for each considered SNR-level; ranging from -10 dB to 10 dB in steps of 5 dB. These estimates were compared to the

N-dimensional PUMA estimator [10], and a 3-D periodogram zero-padded to 512 samples in each dimension, and . On our computer, it was not possible to allow for more zero-padding due to memory constraints. Furthermore, the estimates from the periodogram were selected as the two largest peaks in a cube of size $0.1 \times 0.1 \times 0.1$ around each of the true frequencies, thereby disallowing the periodogram to return any frequencies outside this area. The SEMA estimator were given an initial frequency grid of $15 \times 15 \times 15$ and allowed only one iteration. The user parameter λ was set to either 0.35 or to the mean of the all but the ten largest peaks in the nonzero-padded periodogram, depending on which value was the smallest. The results can be seen in Figure 12, showing the log RMSE for the frequency estimates for the three methods, clearly showing the preferable performance of the SEMA algorithm as compared to the periodogram, and similar performance to the N-dimensional PUMA estimator, even though this has been allowed oracle model order knowledge. The figure also shows the log RMSE for the SEMA and PUMA damping estimates, obtained as a part of the procedure. We note that the used frequency resolution is not limiting the quality of the periodogram estimates via grid effects. It is also worth noting that the evaluation time for the periodogram, implemented using Matlab's optimized FFT command, is only four times faster than using our proposed SEMA 3-D implementation, even though SEMA is implemented using standard Matlab code as well as estimating the damping parameters.

6. Conclusions

In this work, we have introduced a semi-parametric separable sparse model for N-dimensional damped sinusoidal signal components, forming a computationally efficient implementation exploiting the inherent structure of the resulting tensors, which allows us to treat the dictionary for each sampling dimension seperately. The proposed SEMA algorithms is found to yield highly accurate estimates of the frequency and damping coefficients of the signal modes, without imposing a priori knowledge on the number of modes present in the signal, a

difficult for previously proposed parametric methods. To further reduce computational complexity, the proposed method reduces the 2-D dictionary into a sequence of 1-D dictionary learning problems, specifically exploiting the properties of the damping coefficients in a novel dictionary learning approach. The performance of the method is illustrated using 1-, 2-, and 3-D simulated data as compared to the (parametric) PUMA estimator, the Cramér-Rao lower bound, and a zero-padded periodogram estimate, as well as the corresponding non-parametric Capon and IAA based estimators, and a LASSO-based estimator, clearly illustrating the achievable performance gain.

7. Acknowledgment

The authors would like to thank the authors of [7], [10], and [40] for providing their implementation of the PUMA algorithm.

[1] J. Swärd, S. I. Adalbjörnsson, and A. Jakobsson, "High Resolution Sparse Estimation of Exponentially Decaying Signals," in *39th IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Florence, Italy, May 4-9 2014.

[2] S. I. Adalbjörnsson, J. Swärd, and A. Jakobsson, "High Resolution Sparse Estimation of Exponentially Decaying Two-dimensionalimensional Signals," in *22nd European Signal Processing Conference*, Lisbon, Portugal, 2014.

[3] J. Liu and X. Liu, "An Eigenvector-Based Approach for Multidimensional Frequency Estimation With Improved Identifiability," *IEEE Trans. Signal Process.*, vol. 54, pp. 4543–4556, 2006.

[4] Y. Hua, "Estimating Two-Dimensional Frequencies by Matrix Enhancement and Matrix Pencil," *IEEE Transactions on Signal Processing*, vol. 40, no. 9, pp. 2267–2280, September 1992.

[5] J. Sacchini, W. Steedly, and R. Moses, "Two-dimensional Prony modeling and parameter estimation," *IEEE Transactions on Signal Processing*, vol. 41, no. 11, pp. 3127–3137, November 1993.

[6] S. Rouquette and M. Najim, "Estimation of Frequencies and Damping Factors by Two-Dimensional ESPRIT Type Methods," *IEEE Transactions on Signal Processing*, vol. 49, no. 49, pp. 237–245, January 2001.

[7] F. K. W. Chan, H. C. So, and W. Sun, "Subspace approach for two-dimensional parameter estimation of multiple damped sinusoids," *Signal Process.*, vol. 92, pp. 2172 – 2179, 2012.

[8] M. Haardt, F. Roemer, and G. Del Galdo, "Higher-Order SVD-Based Subspace Estimation to Improve the Parameter Estimation Accuracy in Multidimensional Harmonic Retrieval Problems," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3198–3213, July 2008.

[9] Y. Li, J. Razavilar, and K. J. R. Liu, "A High-Resolution Technique for Multidimensional NMR Spectroscopy," *IEEE Trans. Biomed. Eng.*, vol. 45, no. 1, pp. 78–86, 1998.

[10] W. Sun and H. C. So, "Accurate and Computationally Efficient Tensor-Based Subspace Approach for Multidimensional Harmonic Retrieval,"

IEEE Trans. Signal Process., vol. 60, no. 10, pp. 5077–5088, Oct. 2012.

[11] S. Sahnoun, E. H. Djermoune, and D. Brie, "Sparse Modal Estimation of 2-D NMR Signals," in *38th IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Vancouver, Canada, May 26-31 2013.

[12] P. Stoica and T. Sundin, "Nonparametric NMR Spectroscopy," *J. Magn. Reson.*, vol. 152, pp. 57–69, 2001.

[13] E. Gudmundson, P. Stoica, J. Li, A. Jakobsson, M. D. Rowe, J. A. S. Smith, and J. Ling, "Spectral Estimation of Irregularly Sampled Exponentially Decaying Signals with Applications to RF Spectroscopy," *J. Magn. Reson.*, vol. 203, no. 1, pp. 167–176, March 2010.

[14] F. J. Frigo, J. A. Heinen, J. A. Hopkins, T. Niendorf, and B. J. Mock, "Using Peak-Enhanced 2D-Capon Analysis with Single Voxel Proton Magnetic Resonance Spectroscopy to Estimate T2* for Metabolites," in *Proc. of ISMRM*, 2004, vol. 12, p. 2437.

[15] G. O. Glentis and A. Jakobsson, "Computationally efficient damped Capon and APES spectral estimation," in *21st European Signal Processing Conference*, Marrakech, Morocco, Sept. 9-13 2013.

[16] I. F. Gorodnitsky and B. D. Rao, "Sparse Signal Reconstruction from Limited Data Using FOCUSS: A Re-weighted Minimum Norm Algorithm," *IEEE Trans. Signal Process.*, vol. 45, no. 3, pp. 600–616, March 1997.

[17] J. J. Fuchs, "On the Use of Sparse Representations in the Identification of Line Spectra," in *17th World Congress IFAC*, Seoul, jul 2008, pp. 10225–10229.

[18] P. Stoica, Jian Li, and Hao He, "Spectral Analysis of Nonuniformly Sampled Data: A New Approach Versus the Periodogram," *IEEE Trans. Signal Process.*, vol. 57, no. 3, pp. 843–858, March 2009.

[19] T. Yardibi, J. Li, P. Stoica, M. Xue, and A. B. Baggeroer, "Source Localization and Sensing: A Nonparametric Iterative Approach Based on Weighted Least Squares," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 46, no. 1, pp. 425–443, January 2010.

[20] X. Tan, W. Roberts, J. Li, and P. Stoica, "Sparse Learning via Iterative

Minimization With Application to MIMO Radar Imaging," *IEEE Trans. Signal Process.*, vol. 59, no. 3, pp. 1088–1101, March 2011.

[21] P. Stoica, P. Babu, and J. Li, "SPICE : a novel covariance-based sparse estimation method for array processing," *IEEE Trans. Signal Process.*, vol. 59, no. 2, pp. 629 –638, Feb. 2011.

[22] P. Stoica and P. Babu, "SPICE and LIKES: Two hyperparameter-free methods for sparse-parameter estimation," *Signal Processing*, vol. 92, no. 7, pp. 1580–1590, July 2012.

[23] Y. Chi, L. L. Scharf, A. Pezeshki, and A. R. Calderbank, "Sensitivity to Basis Mismatch in Compressed Sensing," *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 2182 –2195, May 2011.

[24] P. Stoica and P. Babu, "Sparse Estimation of Spectral Lines: Grid Selection Problems and Their Solutions," *IEEE Trans. Signal Process.*, vol. 60, no. 2, pp. 962–967, Feb. 2012.

[25] S. I. Adalbjörnsson and A. Jakobsson, "Sparse Estimation of Spectroscopic Signals," in *19th European Signal Processing Conference, EUSIPCO 2011*, Barcelona, Spain, 2011.

[26] S. Sahnoun, E. Djermoune, C. Soussen, and D. Brie, "Sparse multidimensional modal analysis using a multigrid dictionary refinement," *EURASIP J. Applied SP*, vol. 60, pp. 1–10, 2012.

[27] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[28] T. G. Kolda and B. W. Bader, "Tensor Decompositions and Applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[29] P. Stoica and R. Moses, *Spectral Analysis of Signals*, Prentice Hall, Upper Saddle River, N.J., 2005.

[30] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *Journal of the Royal Statistical Society B*, vol. 58, no. 1, pp. 267–288, 1996.

[31] R. Chartrand, "Exact reconstruction of sparse signals via nonconvex min-

imization," *IEEE Signal Process. Lett.*, vol. 14, no. 10, pp. 707–710, Oct. 2007.

[32] E. J. Candès, M. B. Wakin, and S. Boyd, "Enhancing Sparsity by Reweighted l_1 Minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 877–905, Dec. 2008.

[33] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic Decomposition by Basis Pursuit," *SIAM Review*, vol. 43, pp. 129–159, 2001.

[34] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani, "Least angle regression," *The Annals of Statistics*, vol. 32, no. 2, pp. 407–499, April 2004.

[35] R. L. Bishop and S. I. Goldberg, *Tensor Analysis on Manifolds*, Dover Publications, Inc., New York, 1968.

[36] R. A. Horn and C. A. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, England, 1991.

[37] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The John Hopkins University Press, 4^{th} edition, 2013.

[38] C. Tadonki and B. Philippe, "Parallel Numerical Linear Algebra," chapter Parallel Multiplication of a Vector by a Kronecker Product of Matrices, pp. 71–89. Nova Science Publishers, Inc., Commack, NY, USA, 2001.

[39] J. Li and P. Stoica, "Efficient Mixed-Spectrum Estimation with Applications to Target Feature Extraction," *IEEE Trans. Signal Process.*, vol. 44, no. 2, pp. 281–295, February 1996.

[40] H. C. So, F. Chanand W. H. Lau, and C. Chan, "An Efficient Approach for Two-Dimensional Parameter Estimation of a Single-Tone," *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 1999–2009, April 2010.

[41] E. Gudmundson, Jun Ling, P. Stoica, Jian Li, and A. Jakobsson, "Spectral Estimation of Damped Sinusoids in the Case of Irregularly Sampled Data," in *Proceedings of the 9th International Symposium on Signals, Circuits and Systems (ISSCS 2009)*, Iasi, Romania, July 9-10 2009.

[42] A. Månsson, A. Jakobsson, and M. Akke, "Multidimensional Cramer-Rao Lower Bound for Non-Uniformly Sampled NMR Signals," in *22nd European*

Signal Processing Conference, Lisbon, Sept. 1-5 2014.

Algorithm 1 Sparse LS via ADMM

1: Initiate $\mathbf{z} = \mathbf{z}(0), \mathbf{u} = \mathbf{u}(0)$, and $\ell = 0$

2: **repeat**

3: $\quad \mathbf{z}(\ell+1) = \left(\tilde{\mathbf{A}}^H \tilde{\mathbf{A}} + \mu \mathbf{I}\right)^{-1} \left(\tilde{\mathbf{A}}^H \mathbf{y} + \mu(\mathbf{u}(\ell) - \mathbf{d}(\ell))\right)$

4: $\quad \mathbf{u}(\ell+1) = \Psi\left(\mathbf{z}(\ell+1) + \mathbf{d}(\ell+1), \frac{\lambda}{\mu}\right)$

5: $\quad \mathbf{d}(\ell+1) = \mathbf{d}(\ell) + \mathbf{z}(\ell+1) - \mathbf{u}(\ell+1)$

6: $\quad \ell \leftarrow \ell + 1$

7: **until** convergence

[Figure 1 about here.]

[Figure 2 about here.]

[Figure 3 about here.]

[Figure 4 about here.]

[Figure 5 about here.]

[Figure 6 about here.]

[Figure 7 about here.]

[Figure 8 about here.]

[Figure 9 about here.]

[Figure 10 about here.]

[Figure 11 about here.]

[Figure 12 about here.]

Algorithm 2 Mode estimation

1: Initialize all damping coefficients to zero and use (10) to form initial estimates $\left\{g_{\boldsymbol{\gamma}_k}\right\}_{k=1}^{\hat{K}}$

2: **for** $i = 1, \ldots, iter_{max}$ **do**

3: Compute the residual according to (28)

4: **for** $k = 1, \ldots, \hat{K}$ **do**

5: Add the current mode to the residual:
$\boldsymbol{\mathcal{Y}}_k = \boldsymbol{\mathcal{R}}_k + g_{\boldsymbol{\gamma}_k} \mathbf{a}_k^{(1)} \circ \cdots \circ \mathbf{a}_k^{(N)}$

6: Estimate the frequencies and the dampings for the mode

7: Remove the current mode:
$\boldsymbol{\mathcal{R}}_k = \boldsymbol{\mathcal{Y}}_k - g_{\boldsymbol{\gamma}_k} \mathbf{a}_k^{(1)} \circ \cdots \circ \mathbf{a}_k^{(N)}$

8: **end for**

9: Use the found frequencies and damping coefficient to create new dictionaries and re-solve (10).

10: **end for**

List of Figures

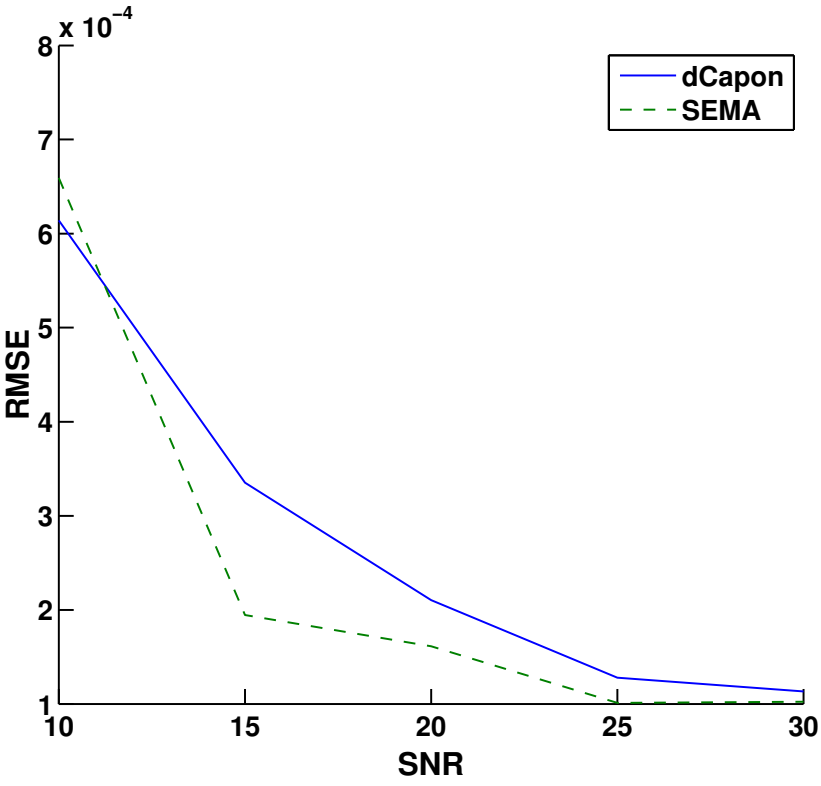

Figure 1: The RMSE of the frequency estimation as a function of SNR.

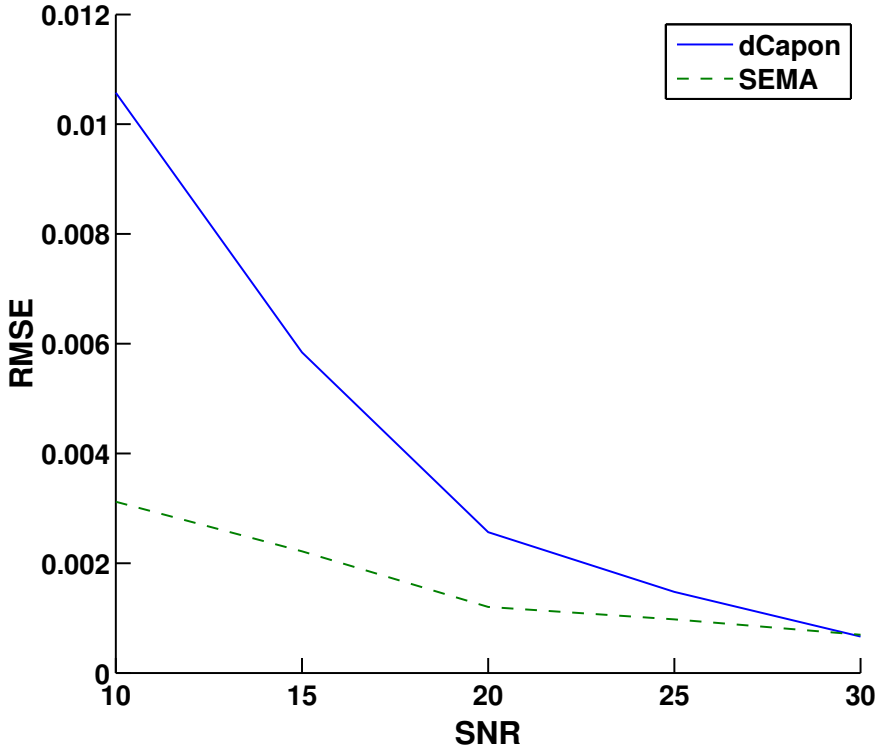

Figure 2: The RMSE of the damping estimation as a function of SNR.

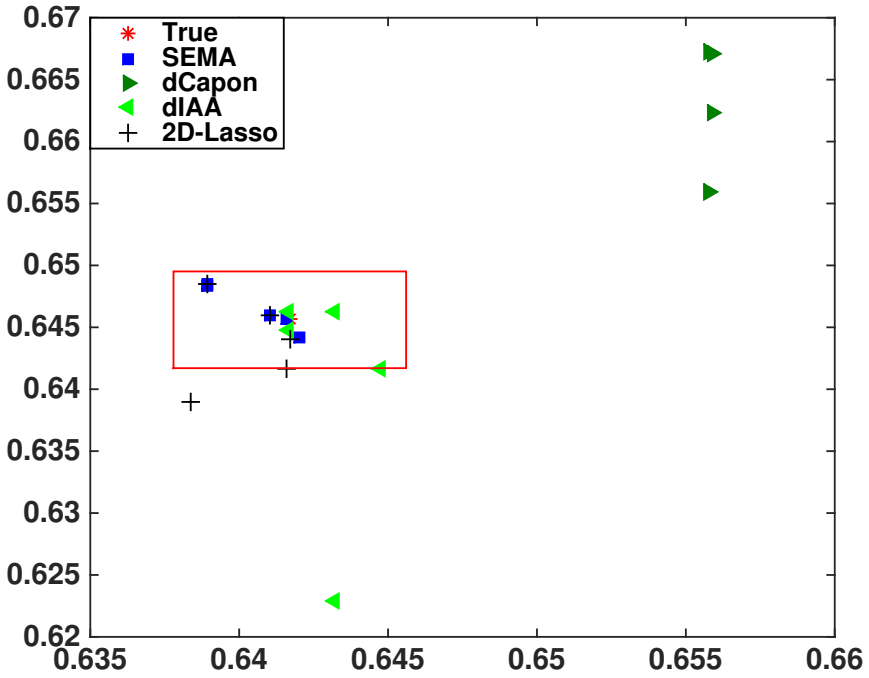

Figure 3: The result of resolving two closely spaced spectral peaks. The (red) square indicates the distance $1/(2N)$ from the true frequencies.

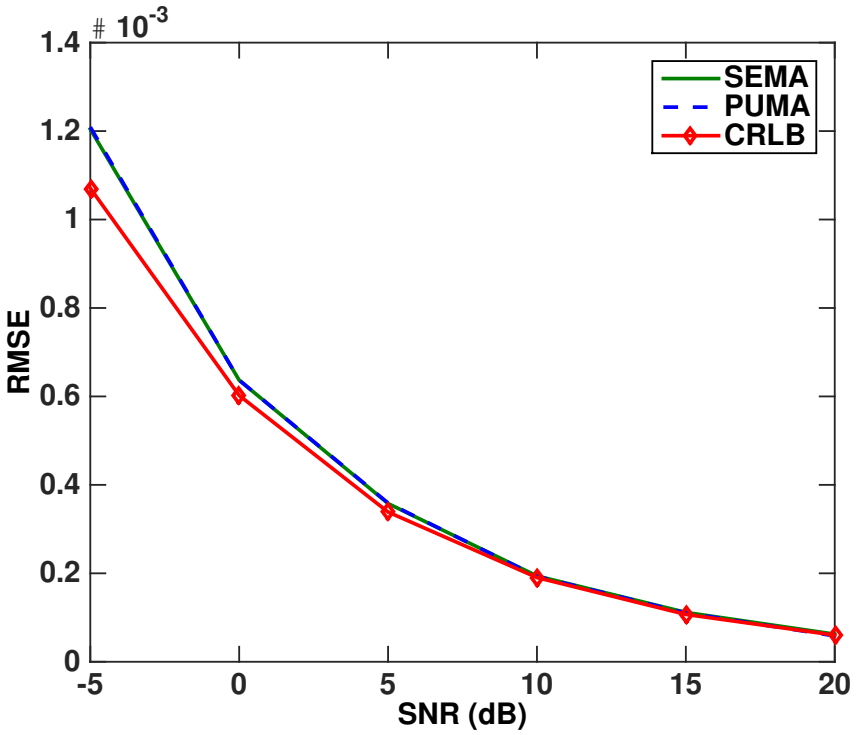

Figure 4: The average RMSE of $f_1^{(1)}$ and $f_2^{(1)}$ as a function of SNR.

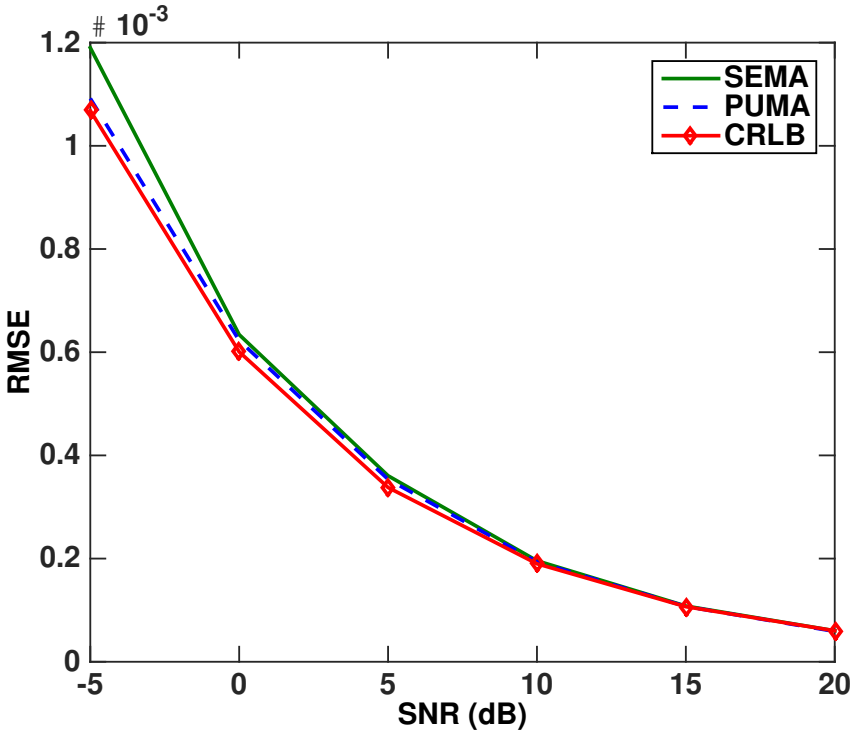

Figure 5: The average RMSE of $f_1^{(2)}$ and $f_2^{(2)}$ as a function of SNR.

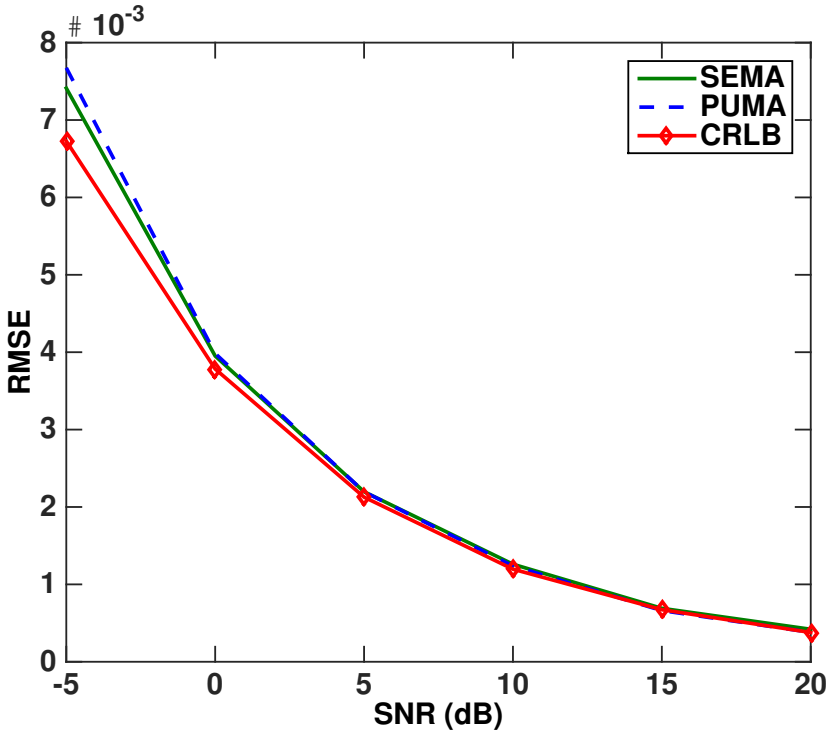

Figure 6: The average RMSE of $\beta_1^{(1)}$ and $\beta_2^{(1)}$ as a function of SNR.

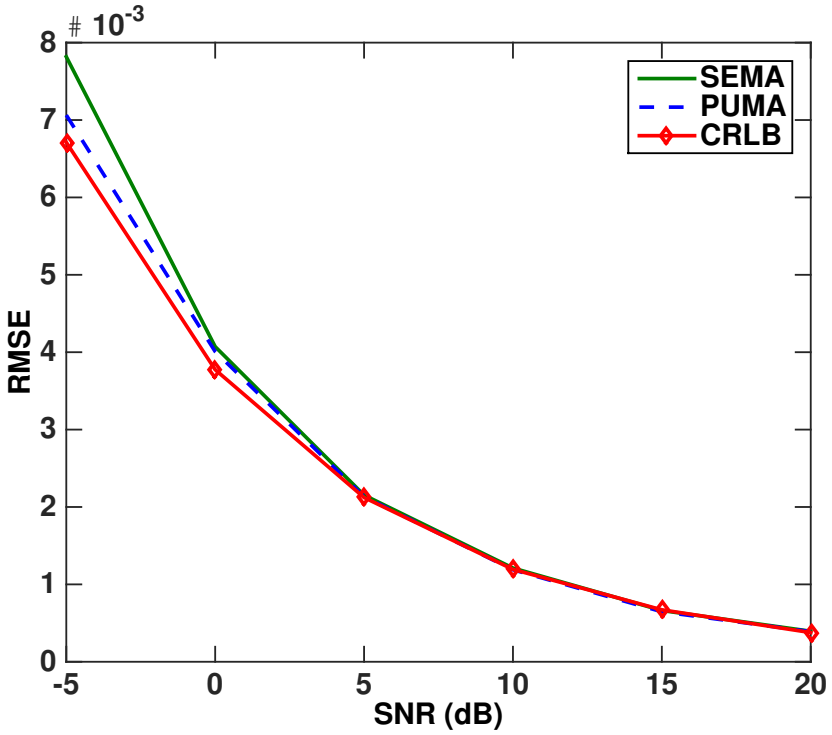

Figure 7: The average RMSE of $\beta_1^{(2)}$ and $\beta_2^{(2)}$ as a function of SNR.

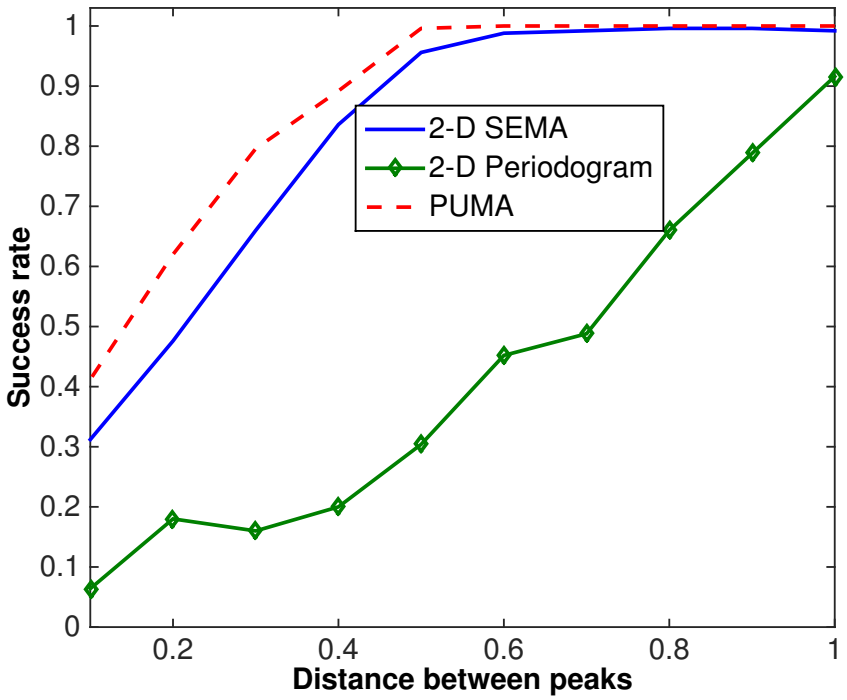

Figure 8: Ability to resolve two peaks as a function of the peak separation.

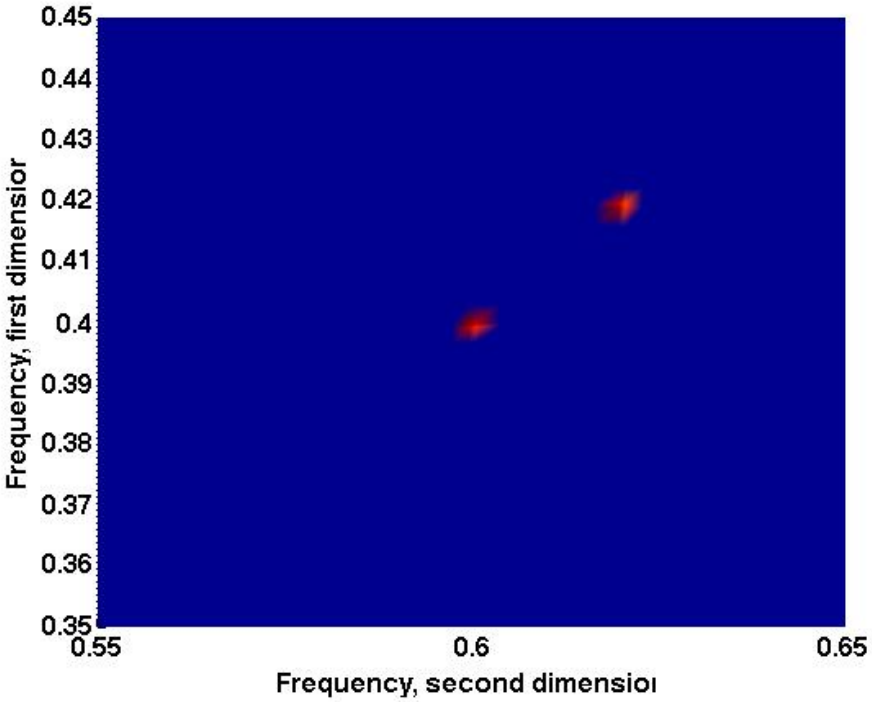

Figure 9: Resulting estimates using 2-D SEMA on two closely spaced modes.

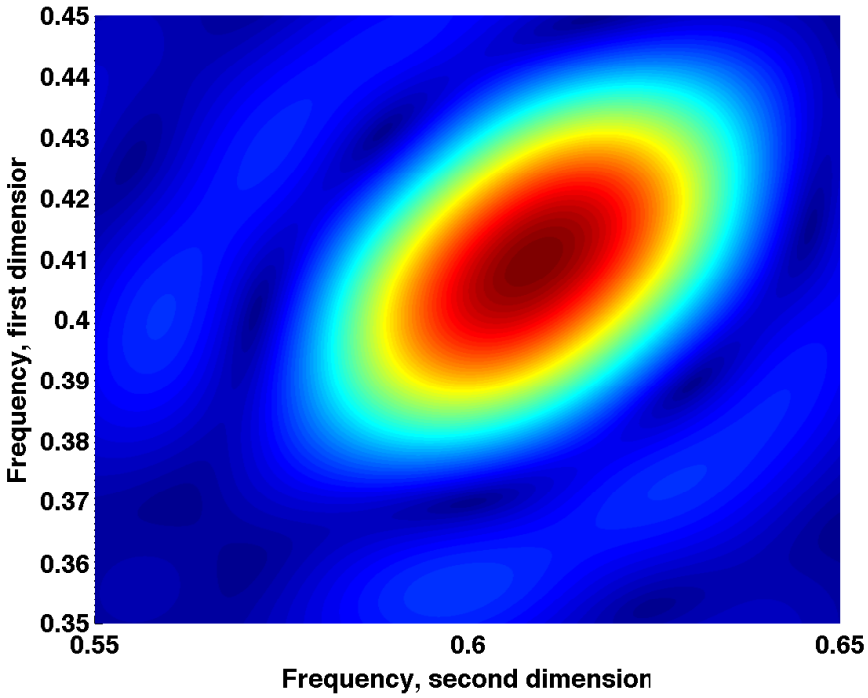

Figure 10: Resulting estimates using two dimensional periodogram on two closely spaced modes.

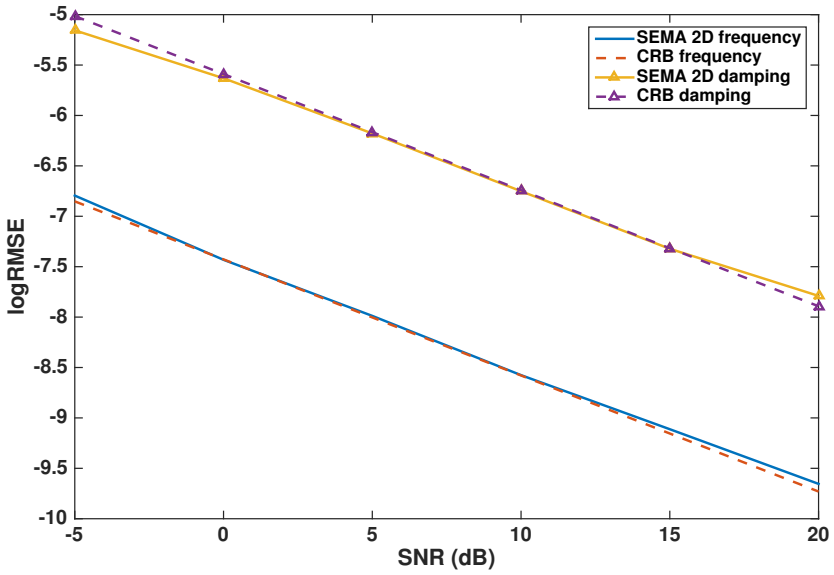

Figure 11: The RMSE for the frequency and damping estimates using SEMA for a non-unifomly sampled signal.

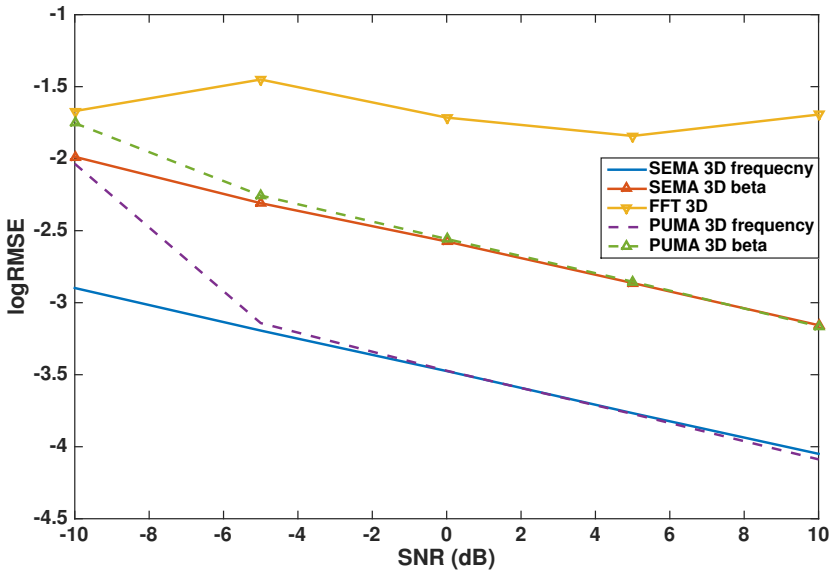

Figure 12: The log RMSE for the frequency estimates using SEMA and 3-D periodogram, together with the log RMSE for the damping estimates yielded from SEMA.