# STRUCTURED AND INCOHERENT PARAMETRIC DICTIONARY DESIGN

*Mehrdad Yaghoobi*[†], *Laurent Daudet*[‡], *Michael E. Davies*[†]

[†] Institute for Digital Communications (IDCom), The University of Edinburgh, EH9 3JL, UK
[‡] Institut Langevin (LOA), UMR 7587, Université Paris Diderot - Paris 7, ESPCI, 10 rue Vauquelin 75231 Paris Cedex 05, France.
{ m.yaghoobi-vaighan, mike.davies}@ed.ac.uk, laurent.daudet@espci.fr

## ABSTRACT

A new dictionary selection approach for sparse coding, called parametric dictionary design, has recently been introduced. The aim is to choose a dictionary from a class of admissible dictionaries which can be presented parametrically. The designed dictionary satisfies a constraint, here the incoherence property, which can help conventional sparse coding methods to find sparser solutions in average. In this paper, an extra constraint will be applied on the parametric dictionaries to find a structured dictionary. Various structures can be imposed on dictionaries to promote a correlation between the atoms. We choose a useful structure which lets us to implement the dictionary using a set of filter banks. This indeed helps to implement the dictionary-signal multiplications more efficiently. The price we pay for the extra structure is that the designed dictionary is not as incoherent as unstructured parametric designed dictionaries.

***Index Terms***— Sparse Approximation, Dictionary Selection, Parametric Dictionary Design, Structured Dictionary.

## 1. INTRODUCTION

Solving an underdetermined linear system inducing a sparsity constraint on the representation has found various applications recently. Often it is assumed that the generative model is known *a priori*. The generative model is often represented by a matrix, called a *dictionary*, $\mathbf{D}_{d\times N} \in \mathbb{C}^{d\times N} : d < N$, which can be used to generate the given signal $\mathbf{y}$ by $\mathbf{y} \approx \mathbf{Dx}$. Each column of $\mathbf{D}$ is called an *atom*. Here we *only* consider real atoms and signals. The sparse approximation would be,

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \ \mathrm{s.t.}\ \|\mathbf{y} - \mathbf{Dx}\|_2^2 \leq \xi, \qquad (1)$$

where the operator $\|.\|_0$ counts the number of non-zero coefficients and $\xi \in \mathbb{R}^+$ is a small constant. Optimization of (1) is very difficult in general and we often use some kind of relaxations or approximations to make it tractable, see [1] for a survey on different sparse coding methods.

When the dictionary is unknown, it can be adapted to a set of training samples using dictionary learning methods, see for example [2,3]. Alternatively one can generate a dictionary which satisfies some mathematical properties to facilitate the use of dictionary with conventional sparse coding algorithms. Parametric Dictionary Design (PDD) [4] is proposed in such a framework, in which the dictionary is specified by a set of parameters. The aim is to find a set of parameters subject to the incoherence of dictionary. The (mutual) coherence [5] $\mu_{\mathbf{D}}$ of a column normalized dictionary $\mathbf{D}$ is defined as follows,

$$\mu_{\mathbf{D}} = \max_{i,j:j\neq i} \{|\langle \mathbf{d}_i, \mathbf{d}_j\rangle|\}.$$

A dictionary is incoherent if its coherence is small and the largest inner-product of two distinct atoms is thus small. The greedy pursuit and basis pursuit algorithms are more successful in Perfect, or Exact, Recovery [5] and the representations are often sparser using incoherent dictionaries, which is a motivation for the incoherent PDD. By letting the dictionary lie in the parametric space we promote the availability of sparse approximations and by minimizing the coherence of the dictionary we improve the performance of practical sparse coding algorithms.

A drawback of the PDD is that the designed dictionary does not have a useful structure, for example, to enable fast implementation. A structured dictionary is in general a dictionary in which the atoms are correlated. A simple example of structured dictionaries is a shift-invariant dictionary in which the atoms are time-shifted versions of a set of mother atoms. A parametric dictionary is called “structured”, if there exist at least two distinct atoms that depend on the values of a single non-empty set of parameters. In this setting, the dictionary is not column separable based on the parameters (the value of a single parameter can change more than one atom). The number of parameters is also reduced as a result, which can help to free up some memory in practice. Although the PDD framework in [4] includes structured dictionaries, they will here be considered with more detail. A case study will be presented later to practically demonstrate the advantages of the proposed method. A new approach for the PDD is also presented which can be used in structured and non-structured scenarios. It simplifies the parameter update step by reducing the problem order from quartic to quadratic form.

The contributions of current paper are twofold:

1. *Presenting a new practical algorithm for solving the parameter update step of PDD:* In the previous reports [4, 6], we introduced a gradient descent based algorithm for the parameter update. Although it works well in some applications, by constraining the search space to the space of rank-d matrices, the parameter update step would be easier. A technique to project onto such a space followed by updating the parameters will later be explored in this paper.

2. *Applying a structure to the parametric dictionaries to accelerate dictionary implementations:* A shift-resilience structure is proposed here. The modified PDD, which is called Structured PDD, is presented and the designed dictionary is compared with the initial and the unstructured dictionary by some simulations.

This work is supported by EU FP7, FET-Open grant number 225913. MED acknowledges support of his position from the Scottish Funding Council and their support of the Joint Research Institute with the Heriot-Watt University as a component part of the Edinburgh Research Partnership.

**Algorithm 1** *Parametric Dictionary Design*

```
1: initialization: $k = 1, \mathbf{D}_{\Gamma_1} \in \mathscr{D}, \{\alpha_i\}_{1 \le i \le K} : 0 < \alpha_i \le 1$
2: while $k \le K$ do
3:     $\mathbf{G}_{\Gamma_k} = \mathbf{D}_{\Gamma_k}^T \mathbf{D}_{\Gamma_k}$
4:     $\mathbf{G}_{P_{k+1}} = \arg\min_{\mathbf{G} \in \Lambda^N} \|\mathbf{G}_{\Gamma_k} - \mathbf{G}\|_F$
5:     $\mathbf{G}_{R_{k+1}} = \alpha_k \mathbf{G}_{P_{k+1}} + (1 - \alpha_k)\mathbf{G}_{\Gamma_k}$
6:     $\mathbf{D}_{\Gamma_{k+1}} \in \mathbf{D}_{\Gamma_k} \cup \{\forall \mathbf{D} \in \mathscr{D} : \|\mathbf{D}^T\mathbf{D} - \mathbf{G}_{R_{k+1}}\|_F < \|\mathbf{G}_{\Gamma_k} - \mathbf{G}_{R_{k+1}}\|_F\}$
7:     $k = k + 1$
8: end while
```

**Algorithm 2** *Parameter Update Step*

```
$\mathbf{G} = \mathbf{G}_{R_{k+1}}$
$\mathbf{G}^{\frac{1}{2}} = \Sigma_d^{\frac{1}{2}} \mathbf{U} \ : \ \mathbf{G} = \mathbf{U}\Sigma\mathbf{U}^T$
$\mathbf{A}^* = \mathbf{V}\mathbf{W}^T \ : \ \mathbf{D}_{\Gamma_k} \mathbf{G}^{\frac{T}{2}} = \mathbf{V}\Delta\mathbf{W}^T$
$\mathbf{D}^* = \arg\min_{\mathbf{D} \in \mathscr{D}} \|\mathbf{D} - \mathbf{A}^*\mathbf{G}^{\frac{1}{2}}\|_F$
$\mathbf{D}_{\Gamma_{k+1}} = \begin{cases} \mathbf{D}^* \\ \mathbf{D}_{\Gamma_k} \end{cases}$ see (7) for the criteria.
Updating $\Gamma_{k+1}$ with the parameters of $\mathbf{D}_{\Gamma_{k+1}}$
```

## 2. PARAMETRIC DICTIONARY DESIGN

Let $\mathbf{D}_\Gamma \in \mathcal{D}$ be a column normalized[1] parametric dictionary where $\Gamma \in \Upsilon$ is a collection of parameters and $\mathcal{D}$ is an admissible set. In a simple setting, $\Gamma$ is a matrix in $\mathbb{R}^{p \times N}$ and each atom $\mathbf{d}_i$ can be generated using a column of parameter matrix $\gamma_i$. The aim of PDD is to find $\Gamma^*$ such that the designed dictionary $\mathbf{D}_{\Gamma^*}$ is *incoherent*, *i.e.* $\mu$ is small. The inner-product of two atoms of $\mathbf{D}$ represents the angle between those atoms. A dictionary with uniform angles between each pair of distinct atoms is called an Equiangular Tight Frame (ETF), which has the minimum coherence [7]. Let $\mathbf{G} := \mathbf{D}^T\mathbf{D}$ be the Gram matrix of $\mathbf{D}$. The Gram matrix $\mathbf{G}_G$ of an ETF has unit values on the main diagonal and the absolute values of the off-diagonal elements are $\mu_G$, which is defined as,

$$\mu_G := \sqrt{\frac{N-d}{d(N-1)}}. \tag{2}$$

Let the linear space of full rank matrices in $\mathbb{R}^{d \times N}$ be equipped with the trace inner product, *i.e.* $\forall \mathbf{A}, \mathbf{B} \in \mathbb{R}^{d \times N} \langle \mathbf{A}, \mathbf{B} \rangle = \mathrm{tr}\{\mathbf{A}^T\mathbf{B}\}$. The PDD can be defined as finding a dictionary with a Gram matrix close to the set of Gram matrices of ETF's, $\Theta_d^N$. An ETF *can* exist if $N \le \frac{d(d+1)}{2}$, which it means that there is no ETF for some pairs of $(d, N)$'s. To simplify the problem and resolve the issue of empty $\Theta_d^N$'s for some $(d, N)$'s, $\Theta_d^N$ is replaced by a convex set $\Lambda^N$ [8], which includes $\Theta_d^N$, as follows,

$$\Lambda^N = \{\mathbf{G} \in \mathbb{R}^{N \times N} : \mathbf{G} = \mathbf{G}^T, \mathrm{diag}\,\mathbf{G} = 1, \max_{i \ne j} |g_{i,j}| \le \mu_G\}.$$

The PDD problem can now be reformulated as an optimization problem,

$$\inf_{\Gamma \in \Upsilon, \mathbf{G} \in \Lambda^N} \|\mathbf{D}_\Gamma^T \mathbf{D}_\Gamma - \mathbf{G}\|_F^2 \tag{3}$$

In this paper we assume that $\Upsilon$ is a compact set, which lets us to use the "min" operator instead of "inf" in (3). Solving (3) is not easy in general. To simplify the problem and find an approximate solution, we assume that $\mathbf{D}_\Gamma$ is continuously differentiable, *i.e.* class $C^1$, then apply a relaxed version of the alternating minimization method. In the alternating minimization method, $\Gamma$ and $\mathbf{G}$ are updated alternatingly to reduce the objective of (3), while the other parameter is kept fixed. The stability, *i.e.* boundedness, of the algorithm is thus guaranteed. A relaxed version of such method has been used in [4] in which $\Gamma$ is updated to reduce the distance of the Gram matrix and a point between current Gram matrix and the current $\mathbf{G} \in \Lambda^N$. The relaxation is controlled by a scalar parameter $\alpha$. This point might be outside of both $\Upsilon$ and $\Lambda^N$. A pseudocode for this algorithm is presented in Algorithm 1. It has two important steps, line 4 and 6. $\mathbf{G}$ is updated in line 4 with the closest point in $\Lambda^N$ to the current $\mathbf{G}_{\Gamma_k}$. As long as $\Lambda^N$ is convex, $\mathbf{G}_{P_{k+1}}$ is unique and it can be found by projecting $\mathbf{G}_{\Gamma_k}$ onto $\Lambda^N$ using the following operator [8],

$$g_{P_{i,j}} = \begin{cases} \mathrm{sign}(g_{D_{i,j}})\mu_G & i \ne j \\ 1 & o.w.\ , \end{cases} \tag{4}$$

where $g_{D_{i,j}}$ is the $(i,j)^{th}$ component of $\mathbf{G}_{\Gamma_k}$. The parameter update step of line 6 can be done using a gradient descent method as introduced in [6]. A difficulty is that the gradient is a tensor and applying conventional optimization methods become difficult in this setting. Here we introduce an alternative technique to update parameters.

Let the set of symmetric rank-d matrices in $\mathbb{R}^{N \times N}$ be noted by $\mathcal{S}^+(d, N)$ [9], which is shown to be equivalent to the set of Gram matrices of full-rank matrices in $\mathbb{R}^{d \times N}$ [10, Proposition 1.1]. $\mathcal{S}^+(d, N)$ has some interesting features which might be useful for the PDD and we left it for an individual research in the future. The first step of the parameter update step can be to find the orthogonal projection of $\mathbf{G}_{R_{k+1}}$ onto $\mathcal{S}^+(d, N)$. If $\mathbf{G}_{R_{k+1}} = \mathbf{U}\Sigma\mathbf{U}^T$, the projection onto $\mathcal{S}^+(d, N)$ can be found by $\mathcal{P}_{\mathcal{S}^+(d,N)}\{\mathbf{G}_{R_{k+1}}\} = \mathbf{U}\Sigma_d\mathbf{U}^T$ where $\Sigma_d = \mathrm{diag}\{\sigma_i\}_{i \in \mathcal{I}_d}$ and $\mathcal{I}_d$ is the set of $d$ largest eigenvalues of $\mathbf{G}_{R_{k+1}}$ [8]. We can now restrict the search space to $\mathcal{S}^+(d, N)$ and find an update which is closer to $\mathcal{P}_{\mathcal{S}^+(d,N)}\{\mathbf{G}_{R_{k+1}}\}$.

$\mathcal{S}^+(d, N) = \{\mathbf{D}^T\mathbf{D} : \mathbf{D}^T \in \mathbb{R}_*^{N \times d}\}$ where $\mathbb{R}_*^{N \times d}$ is the set of all full-rank real $N \times d$ matrices [10]. A further simplification can be to use a mapping from $\mathcal{S}^+(d, N)$ to $\mathbb{R}_*^{N \times d}$ and use a new metric, *i.e.* $\|.\|_F$ in $\mathbb{R}_*^{N \times d}$. This mapping is not unique which is caused by the fact that the Gram matrix is invariant to the left rotation of $\mathbf{D}$. This mapping can be found in two steps, first by calculating $\mathbf{G}^{\frac{1}{2}}$, for example, using eigenvalue decomposition of $\mathbf{G} = \mathbf{U}\Sigma_d\mathbf{U}^T$, *i.e.* $\mathbf{G}^{\frac{1}{2}} = \Sigma_{d\ d \times N}^{\frac{1}{2}}\mathbf{U}^T$, where $\Sigma_d^{\frac{1}{2}} = \mathrm{diag}\{\sigma_i^{1/2}\}_{i \in \mathcal{I}}$ is $d \times N$ diagonal matrix. Then finding the best rotation by minimizing the following objective,

$$\mathbf{A}^* = \arg \min_{\mathbf{A} \in \mathbf{R}_*^{d \times d} : \mathbf{A}^T\mathbf{A} = \mathbf{I}_d} \|\mathbf{D}_{\Gamma_k} - \mathbf{A}\mathbf{G}^{\frac{1}{2}}\|_F. \tag{5}$$

This is a standard optimization problem which can be solved exactly [11, Example 7.4.8] as $\mathbf{A}^* = \mathbf{V}\mathbf{W}^T$, where $\mathbf{D}_{\Gamma_k}\mathbf{G}^{\frac{T}{2}} = \mathbf{V}\Delta\mathbf{W}^T$ is a singular value decomposition. Using these two steps we can find a mapping $f : \mathcal{S}^+(d, N) \to \mathbb{R}_*^{d \times N}$ by $f(\mathbf{G}) = \mathbf{A}^*\mathbf{G}^{\frac{1}{2}}$. Let $d(\mathbf{D}, \mathbf{G}) = \|\mathbf{D}^T\mathbf{D} - \mathbf{G}\|_F$. The parameter update can now be found as follows,

$$\mathcal{D}^* = \{\forall \mathbf{D}^* : \mathbf{D}^* = \arg\min_{\mathbf{D} \in \mathscr{D}} \|\mathbf{D} - f(\mathbf{G})\|_F\}, \tag{6}$$

$$\mathbf{D}_{\Gamma_{k+1}} \begin{cases} \in \mathcal{D}^* & d(\mathbf{D}^*, \mathbf{G}_{R_{k+1}}) < d(\mathbf{D}_{\Gamma_k}, \mathbf{G}_{R_{k+1}}) \\ = \mathbf{D}_{\Gamma_k} & o.w. \end{cases} \tag{7}$$

[1] In this paper we assume that the dictionary is always column normalized.

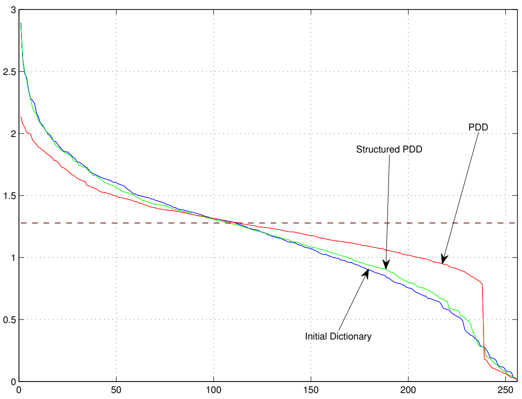

**Fig. 1**. Eigenvalues of the parametric dictionaries.

where $\mathscr{D}$ is the set of parametric dictionaries. Note that the solution of (6) might not be unique. In this case we can update with one of the solutions. The reason that we use (7) instead of directly updating $\mathbf{D}_{\Gamma_{k+1}}$ with a $\mathbf{D}^* \in \mathcal{D}^*$ is to prevent a continuum of solutions. There is a wide range of methods to approximately minimize (6), *e.g.* gradient descent, Newton's and Gauss-Newton's methods. The dictionary update step also provides a parameter update which is used in the PDD. A pseudocode for the new parameter update step in line 6 is presented in Algorithm 2.

### 2.1. Structured Parametric Dictionary Design

A parametric dictionary is called structured if a single parameter affects more than one atom. This framework is general and we only consider a special case, in which the dictionary is partitioned into disjoint sets of uncorrelated atoms. In other words, changing a single parameter can only change the atoms of a partition. An example of such dictionaries will be presented in the next section. Such a dictionary can be presented as $\mathbf{D}_\Gamma = [\mathbf{D}_{\gamma_k}]_{k\in\mathcal{K}}$, where the operator $[.]_{k\in\mathcal{K}}$ is the concatenation of operands. A step in most optimization techniques, which is used for the line 4, is to calculate the gradient of $\mathbf{D}_\Gamma$ with respect to $\{\gamma_k\}_{k\in\mathcal{K}}$ which can be simplified as $\frac{\partial}{\partial\Gamma}\mathbf{D} = [\partial/\partial\gamma_k \mathbf{D}_{\gamma_k}]_{k\in\mathcal{K}}$. In this setting if the number of parameters in each $\gamma_k$ is fixed, *e.g.* $p$, we can generate a parameter matrix $\Gamma_{p\times N}$ by putting $\gamma_k$'s as the columns of $\Gamma_{p\times N}$. In the next section, it will be shown that such a setting can be used to generate a shift-resilient Gammatone parametric dictionary.

## 3. CASE STUDY: STRUCTURED GAMMATONE DICTIONARY

The Gammatone filterbanks have been shown to be closely related to the human auditory system [12] and the dictionary learned using audio training samples [13]. This model will be used here to find a reasonable size incoherent dictionary which has a shift-resilience structure for a more efficient dictionary implementation using filter banks. The generative function for a Gammatone dictionary is as follows,

$$g(t) = at^{n-1}e^{-2\pi bBt}\cos(2\pi f_c t), \tag{8}$$

where $B = f_c/Q + b_{min}$, $f_c$ is the center frequency and $a$, $b$, $Q$, $b_{min}$ and $n$ are some constants. The dictionary is generated by sampling the parameters of $g(t - t_c)$, where $t_c$ is the time-shift. To induce the structure on the dictionary, let $t_c$ be generated with a linear model, *i.e.* $t_c = t_0 + l\Delta : l \in \mathbb{N}_0$, where $t_0 \in [0, \Delta)$, $\Delta$ and $\mathbb{N}_0$ are the time-offset, the time-shift step size and non-negative integers. In this paper we assume that $\Delta$ is fixed during dictionary design, as letting $\Delta$ change, the PDD becomes very complicated. The difficulty is mainly caused by the fact that changing $\Delta$ can change the size of the dictionary. $\gamma_k = [t_{0k}\ f_{c_k}\ n_k\ b_k]^T$ are thus the $k^{th}$ optimization parameters. A set of atoms is generated using $\gamma_k$ and $\{l : l \in \mathcal{L}\}$ followed by discretizing the atoms, see [4] for more detail on discretization. $l$ is upperbounded such that $t_c$ is always smaller than the atom length. We can choose an upper bound for the magnitude of each parameter to generate a bounded admissible set. By including the boundary values, $\Upsilon$ becomes a compact set. The parametric dictionary $\mathbf{D}_\Gamma$ is finally generated by concatenating $\mathbf{D}_{\gamma_k}$'s. The derivation of the dictionary with respect to $\Gamma$ can be derived, using the structure explained in subsection 2.1, for each dictionary block $\mathbf{D}_{\gamma_k}$ similar to [4, Appendix B].

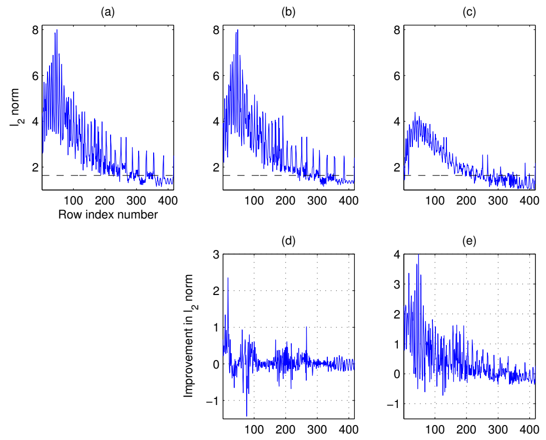

**Fig. 2**. $\ell_2$ norms of the initial (a), the structured designed (b) and unstructured designed dictionaries (c). The improvement of the $\ell_2$ norms w.r.t an ETF for the structured designed (d) and unstructured designed (e) dictionaries.

### 3.1. Simulation Results

The simulations are intended to first show the performance of the PDD algorithm in a structured setting, then demonstrate the advantages of designed dictionary in sparse approximation of audio signals using MP. The simulation parameters are presented in Table 1. The parameters in the first row and $\Delta$ are fixed and the others are used to generate the initial dictionaries, which might change throughout the PDD.

The simulations were run with two settings, 1) *unstructured:* no constraint on $t_c$ and 2) *structured:* $t_c$ follows the model $t_c = t_0 + l\Delta$. In the first experiment we designed the dictionary and showed the eigenvalues of the Gram matrices in Figure 1. The eigenvalues of a tight frame is also shown with a dashed line. Although the improvement of the eigenvalues, toward a tight frame, is not significant, it is changed in the right direction and is between the original and an unstructured parametric designed dictionary. The $\ell_2$ norms of the columns of the Gram matrices of the mentioned dictionaries, which can show how much the corresponding atoms are correlated to the

**Table 1**. The parameters of the Structured PDD.

| $d$ | $N$ | $\lvert\mathcal{K}\rvert$ | $b_{min}$ | $Q$ | $K$ | $\alpha$ |
|---|---|---|---|---|---|---|
| 256 | 418 | 35 | 24.7 | 9.26 | 100 | 0.5 |

| $t_0$ | n | b | $f_c$ | $\Delta$ |
|---|---|---|---|---|
| 0 | 4 | 1 | $50 + .27kB$ | $\arg\max_t \lvert g(t)\rvert$ |

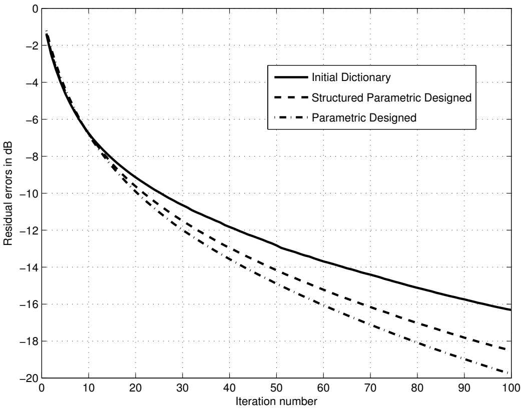

**Fig. 3**. Average approximation errors using MP for the initial, structured and unstructured designed dictionaries.

other atoms, are shown in the first row of Figure 2. The changes of the norms is obvious in the unstructured designed dictionary. To show that it is improved in the structured dictionary we also showed the reduction of the norms toward an ETF in the second row. Although the improvements in norms are small, most of the graph is in the positive orthant, which shows a reduction of the norm to a reference ETF.

Figures 1 and 2 show only a small achievement by structured PDD. This might be caused by selecting a highly restrictive structure for the dictionary. It is also relevant to investigate the performance of the structured parametric designed dictionary in sparse approximation of some sparsly structured signals. Some audio signals recorded from BBC Radio 3, which often plays classical music, have been used to evaluate the dictionaries. The average approximation errors, using 100 randomly selected audio samples, of the sparse approximations by applying MP algorithm are shown in Figure 3. The Structured dictionary shows a promising performance in this experiment.

## 4. CONCLUSION

Imposing a structure on the parametric dictionary to facilitate the implementation of the designed dictionaries was investigated in this paper. A general form was introduced and a special case was investigated in more detail by using a case study. Another method was also presented to let the PDD be solved using conventional optimization techniques. Finally by some simulations on the Gammatone parametric dictionary, we showed that the designed dictionary is superior to the initial dictionary in sparse approximations of some selected audio signals.

One possible structure was explored in this report. There have been various structures introduced for dictionaries in dictionary learning problem. An independent research on these structures is left for future work. A structured parametric dictionary model can also be used in the dictionary learning problem. It preserves the structure of dictionary while adapting the dictionary to a given data.

The proposed algorithm for the parameter update needs to calculate the objective value in each iteration. It is a necessary step to guarantee the stability of the algorithm. Furthur investigations on the proposed algorithm might guarantee the stability of the overall algorithm without an explicit calculation of the objective.

## 5. REFERENCES

[1] J.E. Tropp and S.J. Wright, “Computational methods for sparse solution of linear inverse problems,” Tech. Rep. 2009-01, California Institute of Technology, March 2009.

[2] M. Aharon, E. Elad, and A.M. Bruckstein, “K-SVD: an algorithm for designining of overcomplete dictionaries for sparse representation,” *IEEE Trans. on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[3] M. Yaghoobi, T. Blumensath, and M. Davies, “Dictionary learning for sparse approximations with the majorization method,” *IEEE Trans. on Signal Processing*, vol. 57, no. 6, pp. 2178–2191, 2009.

[4] M. Yaghoobi, L. Daudet, and M. Davies, “Parametric dictionary design for sparse coding,” accepted for publication in IEEE Transactions on Signal Processing, 2009.

[5] D.L. Donoho and X. Huo, “Uncertainty principles and ideal atomic decomposition,” *IEEE Trans. on Information Theory*, vol. 47, no. 7, pp. 2845–2862, 2001.

[6] M. Yaghoobi, L. Daudet, and M. Davies, “Parametric dictionary design for sparse coding,” in *Workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS09)*, 2009.

[7] T. Strohmer and R.W. Heath, “Grassmannian frames with applications to coding and communication,” *Applied and Computational Harmonic Analysis*, vol. 14, no. 3, pp. 257–275, 2003.

[8] J.A. Tropp, I.S. Dhillon, R.W. Heath Jr., and T. Strohmer, “Designing structural tight frames via an alternating projection method,” *IEEE Trans. on Information Theory*, vol. 51, no. 1, pp. 188–209, 2005.

[9] S. Bonnabel and R. Sepulchre, “Geometric distance and mean for positive semi-definite matrices of fixed rank,” Accepted for publication in SIAM Journal of Matrix Analysis. Available at `http://arxiv.org/abs/0807.4462`.

[10] B. Vandereycken, P.-A. Absil, and S. Vandewalle, “Embedded geometry of the set of symmetric positive semidefinite matrices of fixed rank,” in *IEEE Workshop on Statistical Signal Processing*, Cardiff, UK, September 2009, pp. 389–392.

[11] R.A. Horn and C.R. Johnson, *Matrix Analysis*, Cambridge University Press, 1985.

[12] R.D. Patterson, I. Nimmo-Smith, J. Holdsworth, and P. Rice, “An efficient auditory filterbank based on the Gammatone function,” Tech. Rep., APU Report, 1988.

[13] E.C. Smith and M. S. Lewicki, “Efficient auditory coding,” *Nature*, vol. 439, pp. 978–982, 2006.