Convolutional Sparse Coding with Multipath Orthogonal Matching Pursuit

Yanis Gomes[1], Charles Truong[1], Jean-Philippe Saut[2], Fikri Hafid[1,3], Pascale Prieur[3], Laurent Oudre[1]
[1] *Université Paris Saclay, Université Paris Cité, ENS Paris Saclay, CNRS, SSA, INSERM, Centre Borelli,* F-91190, Gif-sur-Yvette, France
[2] *Eurobios Mews Labs,* F-94230, Cachan, France
[3] *RTE, Direction R&D,* F-92073, Paris La Défense, France

***Abstract*—Finding patterns in time series is crucial to understanding physical or physiological phenomena monitored with sensors. Convolutional sparse coding (CSC) methods, which approximate signals by a sparse combination of short signal templates (also called atoms), are well-suited for this task. Nevertheless, sparsity results in intractable non-convex optimization problems. This paper introduces an algorithm, based on Multipath Matching Pursuit, which is novel in convolutional settings, to efficiently and accurately estimate atoms' localizations in time series. We describe a principled way to improve this greedy procedure by returning several candidate solutions instead of one. Our approach yields better localization and signal reconstruction on simulated data and in a real-world use case, which consists in automatically detect damages, such as cracks and broken wires, in overhead power lines.**

***Index Terms*—convolutional sparse coding, orthogonal matching pursuit, compressed sensing, greedy optimization, sparse coding, sparse representation, deconvolution**

I. Introduction

When studying complex physical or physiological phenomena, a common approach consists in monitoring a system or a subject with sensors. Often, finding characteristic patterns in the resulting time series helps to understand the phenomena at hand [1]–[3]. In this work, we study overhead power lines, an essential part of electricity transmission systems that transmit electrical energy along large distances. For instance, RTE (Réseau de Transport d'Électricité), the French electricity transmission system operator, manages more than 100,000 km of such lines. To detect cracks and broken wires due to fretting fatigue, the Magnetic Flux Leakage (MFL) method is a well-known non-destructive testing approach [4]. Indeed, the magnetic field measured around a defect has an identifiable and reproducible shape [5]–[8]; see Fig. 1 for an example. However, simple pattern detection approaches fail when several defects are close together and overlap; the contributions of each defect to the signal are difficult to disentangle. In this article, we will speak about the "time axis" for generality, although it is rather a "position axis" for this particular application.

Related work. Detecting characteristic local patterns is often referred to as Convolutional Sparse Coding (CSC) [3], [9], [10]. In this setting, the signal is modelled as a linear combination of a few short prototypical patterns called atoms, and the objective is to estimate from noisy observations the temporal location and type of the atoms [11]. In our application, the dictionary is known beforehand as it is given by physical modelling of the overhead power lines. One issue with CSC is that it yields a non-convex optimization problem because of the strict sparsity (enforced by a ℓ_0-norm). The CSC literature has roughly two approaches to address this problem.

This work was funded by the Industrial Chair PhLAMES.

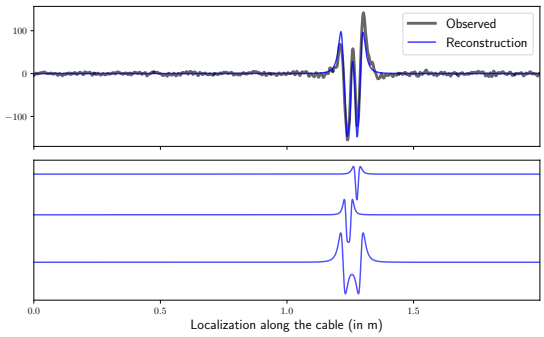

Fig. 1. Top: example of a two-meter-long noisy MFL response signal (in arbitrary units) and its reconstruction as a linear combination of three prototypical defects. Bottom: the three prototypical defects, ordered from largest amplitude (bottom) to smallest (top).

In the convolutional setting, several articles propose to solve a convex relaxation composed of a reconstruction error and a ℓ_1-based regularization to enforce sparsity [11]. The resulting optimization problem is exactly and efficiently solved with coordinate descent algorithms [12], [13] or proximal gradient descent [14]. Nevertheless, the solutions to the relaxed problem often contain spurious detections and post-processing is needed to have sparser solutions [15].

The second type of approach includes greedy methods which extend to the convolutional setting, Matching Pursuit (MP) and Orthogonal Matching Pursuit (OMP) techniques [16]–[18] They are iterative methods that estimate the position and type of a single atom at each iteration. Greedy methods only provide a suboptimal solution, which is problematic when the atoms are correlated [19], [20], and overlap, as in our data.

Approaches to mitigate the suboptimality of greedy techniques have been explored *in a non-convolutional setting*. All compute several candidate solutions and return the one with the lowest reconstruction error. Certain authors propose

to estimate several atoms at each iteration [21], [22]. Others randomly draw atoms at each iteration [23]. A slightly different algorithm uses a so-called multipath approach, where instead of taking the best atom, the 2nd or 3rd best are also considered [24]–[26]. Those methods do not adapt easily to the convolutional setting because we need to estimate both the atom and its location, resulting in many possible combinations.

Contributions. We propose an improvement on the OMP algorithm for the CSC problem that relies on the multipath approach [24] introduced in a *non-convolutional* setting. We adapt this methodology to take into account the fact that we have to estimate both the location and the atom. We also describe a novel, efficient strategy that saves computations between similar candidate solutions. Our approach is empirically compared to ℓ_1-based and greedy algorithms and is shown to outperform them on simulated and real data significantly. We also provide an easy-to-install Python implementation (github.com/deepcharles/convmmp) and an online app (convmmp.streamlit.app).

II. Background on Convolutional Sparse Coding

This section formally introduces the CSC model and briefly explains the convolutional OMP, which will be helpful when our method is described. (In the following, $[\![a]\!]$ is the integer interval $\{1, \dots, a\}$.)

A. Model and Task

Let $\boldsymbol{y} = [y_1, \dots, y_T] \in \mathbb{R}^T$ be a real-valued signal with T observations. The linear convolutional model [11] assumes that the signal $\boldsymbol{y}$ satisfies $\boldsymbol{y} = \sum_{i=1}^{N_d} \boldsymbol{d}_i * \boldsymbol{z}_i^\star + \boldsymbol{\varepsilon}$ where $\boldsymbol{a} * \boldsymbol{b}$ is the convolution product between two real signals $\boldsymbol{a}$ and $\boldsymbol{b}$, the $\boldsymbol{d}_i \in \mathbb{R}^{T_d}$ are N_d unit norm signals ($\|\boldsymbol{d}_i\| = 1$) of length $T_d \ll T$ called atoms, the $\boldsymbol{z}_i^\star \in \mathbb{R}^{T_z}$ are N_d (unknown) signals of length T_z called activation signals, and $\boldsymbol{\varepsilon} \in \mathbb{R}^T$ is a Gaussian white noise of same length as $\boldsymbol{y}$. The length T_z of the activation signals is such that the $\boldsymbol{d}_i * \boldsymbol{z}_i$ have length T.

In the literature, the set of atoms $\{\boldsymbol{d}_i\}_{i \in [\![N_d]\!]}$ is often called the dictionary. Each atom $\boldsymbol{d}_i$ is activated at time positions encoded by the activation signal $\boldsymbol{z}_i^\star$. Generally, it is assumed that only a few positions are activated, meaning that the signals $\boldsymbol{z}_i^\star$ are sparse. The objective of CSC algorithms is to recover the true activations $\boldsymbol{z}_i^\star$ from the noisy observations $\boldsymbol{y}$. This estimation amounts to solving the following optimization problem:

$$\min_{\substack{\boldsymbol{z}_i \in \mathbb{R}^{T_z} \\ i \in [\![N_d]\!]}} \left[\left\| \boldsymbol{y} - \sum_{i=1}^{N_d} \boldsymbol{d}_i * \boldsymbol{z}_i \right\|^2 + \lambda \sum_{i=1}^{N_d} \|\boldsymbol{z}_i\|_0 \right] \tag{1}$$

where $\|\boldsymbol{z}_i\|_0$ is the so-called ℓ_0 (pseudo-)norm, which counts the number of non-zero elements of $\boldsymbol{z}_i$ and $\lambda > 0$ is a user-defined parameter controlling the trade-off between the data-fidelity term and the activations' sparsity. Recall that, here, the dictionary is assumed known.

B. Background on `conv-OMP`

Let `conv-OMP` denote the OMP procedure in a convolutional setting. This is is a greedy iterative procedure that finds an approximate solution to the CSC problem (1) [18]. At each iteration, it finds and records the most correlated atom with a residual signal that is then updated by removing the contribution of all previously found atoms. The next iteration repeats the same operation on the new residual.

Formally, initiate the residual signal $\boldsymbol{r} = \boldsymbol{y}$ and the set $\widehat{\mathcal{T}} = \emptyset$ of selected pairs $(t, i) \in [\![T_z]\!] \times [\![N_d]\!]$ of time position and atom index. At iteration $k \geq 1$, we select one atom and one time position by solving

$$\hat{t}, \hat{\imath} \leftarrow \underset{t \in [\![T]\!], i \in [\![N_d]\!]}{\arg\max} \langle \boldsymbol{r}, \boldsymbol{\delta}_t * \boldsymbol{d}_i \rangle^2 \tag{2}$$

where $\boldsymbol{\delta}_t \in \mathbb{R}^{T_z}$ is the (activation) signal that is zero everywhere except at time position t, where it is equal to 1. The set $\mathcal{T}$ becomes $\mathcal{T} \leftarrow \mathcal{T} \cup \{(\hat{t}, \hat{\imath})\}$. To update the residual for the following iteration, define $\tilde{\boldsymbol{y}} \in \mathbb{R}^T$ the orthogonal projection of $\boldsymbol{y}$ onto $\textsc{Span}\left(\{\boldsymbol{\delta}_{\hat{t}} * \boldsymbol{d}_{\hat{\imath}} \text{ for all } (\hat{t}, \hat{\imath}) \in \mathcal{T}\}\right) \subset \mathbb{R}^T$, the linear space spanned by the selected atoms and time positions up to iteration k. The residual signal is updated: $\boldsymbol{r} \leftarrow \boldsymbol{y} - \tilde{\boldsymbol{y}}$. The iterations end when a user-defined stopping criterion is met, for instance, when the desired number of atoms is detected or the norm of the residual falls below a threshold.

III. Method

This section describes our approach, called `conv-MMP` for Convolutional Multipath Orthogonal Matching Pursuit.

A. Algorithm's Overview

Our method is a greedy procedure that estimates the solution of the CSC problem (1) by iteratively finding atoms correlated with the residual, as in `conv-OMP`. However, instead of only considering the most correlated atom at each iteration, our approach also looks at the 2nd, 3rd, 4th... most correlated atoms. The multipath approach defines *paths*, which are sequences of rankings, e.g. (2nd, 1st, 4th). For this particular path, our approach starts by finding the 2nd most correlated position and atom index and adds them to the set of detected atoms. Then the contribution of this atom is removed, as in `conv-OMP`, and the "1st most" correlated (with the residual) position and atom index is added to the set of detected atoms, and the residual is updated. In the last iteration, the 4th most correlated (with the residual) position and atom index is added to the set of detected atoms. Notice that `conv-OMP` amounts to applying our method on the path (1st, 1st, ...). Also, the path length is the number of iterations.

Usually, several paths are considered, and the one that minimizes the reconstruction error is the output of our algorithm. Contrary to the classical multipath approach [24], in the convolutional setting, it is not obvious to rank pairs (t, i) of time position and atom index according to their correlations: part of our contribution is to define a relevant ranking (see Sec. III-B). Also, we describe how to save computations between similar paths (e.g. (2nd, 1st, 1st) and (2nd, 1st, 2nd)),

resulting in a faster algorithm, compared to the naive version that computes all paths independently (see Sec. III-C).

B. Ordering Correlations

At a particular iteration, we have a residual signal $\boldsymbol{r}$. The objective is to estimate the j-th most correlated (with the residual) pair (t, i) of time position and atom index. A naive approach is to order pairs $(t, i) \in [\![T_z]\!] \times [\![N_d]\!]$ of time position and atom according to their correlation with the residual: the j-th best pair would the one associated with the j-th largest correlation $\langle \boldsymbol{r}, \boldsymbol{\delta}_t * \boldsymbol{d}_i \rangle^2$. However, this strategy does not produce a diversified set of candidates because $\langle \boldsymbol{r}, \boldsymbol{\delta}_t * \boldsymbol{d}_i \rangle^2$ can be closely approximated by $\langle \boldsymbol{r}, \boldsymbol{\delta}_{t+1} * \boldsymbol{d}_i \rangle^2$ or $\langle \boldsymbol{r}, \boldsymbol{\delta}_{t-1} * \boldsymbol{d}_i \rangle^2$ (notice the time shift by one sample). Indeed,

$$\begin{aligned}\langle \boldsymbol{r}, \boldsymbol{\delta}_t * \boldsymbol{d}_i \rangle - \langle \boldsymbol{r}, \boldsymbol{\delta}_{t-1} * \boldsymbol{d}_i \rangle &= \sum_{s=1}^{T_d} r_{t+s} d_{i,s} - \sum_{s=1}^{T_d} r_{t-1+s} d_{i,s} \\ &= r_{t+T_d} d_{i,T_d} - r_t d_{i,1} + \sum_{s=1}^{T_d-1} r_{t+s}(d_{i,s} - d_{i,s+1}).\end{aligned}$$

If the atoms are smooth ($d_{i,s} - d_{i,s+1} \approx 0$) and vanish to zero on the edges ($d_{i,1} \approx 0$ and $d_{i,T_d} \approx 0$) –both assumptions are verified in our real-world use-case, see Fig. 1– then $\langle \boldsymbol{r}, \boldsymbol{\delta}_t * \boldsymbol{d}_i \rangle \approx \langle \boldsymbol{r}, \boldsymbol{\delta}_{t-1} * \boldsymbol{d}_i \rangle$. Therefore, if $\langle \boldsymbol{r}, \boldsymbol{\delta}_t * \boldsymbol{d}_i \rangle^2$ is the largest correlation, the second largest is often $\langle \boldsymbol{r}, \boldsymbol{\delta}_{t+1} * \boldsymbol{d}_i \rangle^2$ or $\langle \boldsymbol{r}, \boldsymbol{\delta}_{t-1} * \boldsymbol{d}_i \rangle^2$. The naive correlation ordering only produces, at first, almost identical candidate pairs.

To explore atoms that are different (in position) from each other, we introduce a novel partial order between pairs $(t, i) \in [\![T_z]\!] \times [\![N_d]\!]$ of time position and atom. The first pair, denoted $(t^{(1)}, i^{(1)})$, is defined by

$$t^{(1)}, i^{(1)} := \underset{t \in [\![T]\!], i \in [\![N_d]\!]}{\arg\max} \langle \boldsymbol{r}, \boldsymbol{\delta}_t * \boldsymbol{d}_i \rangle^2 \tag{3}$$

where $\boldsymbol{\delta}_t$ is defined in (2). Note that $(t^{(1)}, i^{(1)})$ is the pair selected by the `conv-OMP` iteration. Now assume that j pairs have been ordered, $(t^{(1)}, i^{(1)}), \ldots, (t^{(j)}, i^{(j)})$. The $(j+1)$-th pair is defined as the one that maximizes the correlation with the residual while being located at a certain time distance $\Delta > 0$ from previous pairs:

$$t^{(j+1)}, i^{(j+1)} := \underset{\substack{t \in [\![T]\!], i \in [\![N_d]\!] \\ \text{s.t. } \forall l \in [\![j]\!],\ |t - t^{(l)}| > \Delta}}{\arg\max} \langle \boldsymbol{r}, \boldsymbol{\delta}_t * \boldsymbol{d}_i \rangle^2 . \tag{4}$$

This definition is natural in the sense that $\boldsymbol{\delta}_{t^{(j)}} * \boldsymbol{d}_{i^{(j)}}$ is more correlated with the residual signal than $\boldsymbol{\delta}_{t^{(j+1)}} * \boldsymbol{d}_{i^{(j+1)}}$, for any $j \geq 1$. In addition, the constraint on the time position ensures that $|t^{(j)} - t^{(j')}| > \Delta$ for any (j, j'). This prevents selecting very close atoms and guarantees that the candidate solutions are diversified.

C. Computational considerations

Enumerating paths. To apply `conv-MMP`, we need to enumerate promising paths. We adopt the path ordering scheme of the original (non-convolutional) multipath approach [24]. With this scheme, called Depth First Search, paths with low ranked atoms in the early iterations come first. The width W is the lowest rank that is explored. For instance, the first four paths with width $W = 3$ are (1st, 1st, 1st), (2nd, 1st, 1st), (3rd, 1st, 1st), and (1st, 2nd, 1st). For more details, see [24].

Complexity. For a single path, `conv-MMP` has the same complexity as `conv-OMP`: $N_{\text{atom}}(C_{\text{cor}} + C_{\text{proj}})$ where N_{atom} is the number of atoms to estimate, C_{cor} is the complexity of finding the best correlation (2) and C_{proj} is the complexity of updating the residual. (For simplicity, assume the same complexity for finding the best correlation or the W best ones, although there is a slight overhead.) Straightforwardly, we have $C_{\text{cor}} = \mathcal{O}(N_d T \log T_d)$ (using FFT-based convolutions). A naive implementation of `conv-MMP` has complexity $N_{\text{path}} N_{\text{atom}}(C_{\text{cor}} + C_{\text{proj}})$ where N_{path} is the number of paths. However, paths often have operations that can be shared. For instance, in the first iteration of (1st, 1st) and (2nd, 1st), the same correlations must be computed. Also, (1st, 1st) and (1st, 2nd) have the same first residual update. Our implementation takes advantage of those shared operations. If we assume that $N_{\text{path}} = W^P$ for a certain P, several combinatorial manipulations yield a complexity $(W^P - 1)(C_{\text{cor}} + W C_{\text{proj}})/(W - 1)$; the complexity gain over the naive implementation is larger than $W^P C_{\text{cor}} = N_{\text{path}} C_{\text{cor}}$.

IV. Experiments

This section describes the experiments, simulated and real, where our method is compared to baseline approaches.

Baselines. We compare our method `conv-MMP` to three baseline methods: `conv-OMP`, described in Sec. II-B, Convolutional Matching Pursuit, `conv-MP`, which is similar to `conv-OMP`, with a more straightforward residual update step (no orthogonal projection), and `conv-L1`, which solves a convex approximation of the CSC problem (1) (implementation of [27]). For our method, we explore different numbers of paths (3, 9, 27, 81), resulting in four versions of our method to be evaluated: `conv-MMP3`, `conv-MMP9`, `conv-MMP27` and `conv-MMP81`. Recall that `conv-OMP` is equivalent to `conv-MMP1`. The time distance Δ (4) is set to ten samples, and the width to three.

Dictionary. In the simulated and real experiments, we use the same dictionary derived from the physical model that describes the real-world data. As in [5], [6], [28], the magnetic leakage field from a steel surface is modelled by a magnetic dipole model: the radial component H_r of the magnetic field at position x along the overhead line is

$$H_r(x, y, b, \sigma) = \frac{-8\sigma b x y}{[(x+b)^2 + y^2]\,[(x-b)^2 + y^2]} \tag{5}$$

where $2b$ is the width of the defect (in m), σ is the linear density of magnetic charge (in G.m), and y is the height (in m) of the sensor probe measured from the surface of the material. In practice, our magnetic field sensor measures $\partial H_r / \partial x$, up to a multiplicative constant. An atom is defined as the signal $\partial H_r / \partial x$ measured over a cable of length 0.5 m according to (5), for various parameter combinations (y, b, σ), chosen by experts to reflect the kind of defects found in the real data.

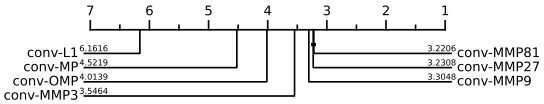

Fig. 2. Critical difference diagram showing algorithms' rankings according to the F1 score on the simulated data. A thick horizontal black line joins algorithms that do not have significantly different rankings (Friedman test with a post-hoc Wilcoxon-Holm correction, level 5%). Here `conv-MMP27` and `conv-MMP81` are not significantly different.

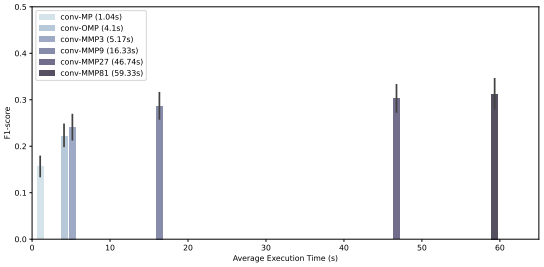

Fig. 3. Trade-off between performance and computation time. Black thick vertical lines indicate the variance of the F1 score.

The resulting dictionary has 143 atoms of length 492 samples (4.92 m); see Fig. 1 for a few atom examples.

A. Simulated experiments

Data. Data are generated by randomly selecting atoms from the dictionary and positions along the cable. The number of atoms is random, between 2 and 5. We add a Gaussian white noise, with a Signal-to-Noise Ratio (SNR) in $\{0, 5, 10, 15\}$ dB. For each pair of (atom number, SNR), we generate 200 signals, resulting in 3200 synthetic signals, all of length $T = 738$.

Metrics. A true positive is a well-detected atom, both in position and shape. Formally, an atom $\boldsymbol{d}_i$ at position t is a true positive if there is an estimated atom $\boldsymbol{d}_{\hat{i}}$ at position $\hat{t}$ such that $|t - \hat{t}|$ is below a position margin and $|\langle \boldsymbol{d}_i, \boldsymbol{d}_{\hat{i}} \rangle|$ is above a correlation margin. (We add the constraint that an estimated atom can only detect one true atom.) We set the position margin to five samples, and the correlation margin to 0.95. We can compute the F1 score from the number of true positives. We rank the algorithms for each signal by their F1 scores and report the average rank in a Critical Difference Diagram [29]. Here, the true number of atoms is known.

Results. Average rankings are shown on Fig. 2. First, `conv-L1` and `conv-MP` are the least accurate methods. Second, we observe that more paths lead to better F1 scores. However, there is no statistical difference between `conv-MMP27` and `conv-MMP81`, meaning that increasing the number of explored paths above a certain point is unnecessary.

Fig. 3 shows how the F1 score evolves with the computation time. (We restrict ourselves to signals containing three atoms for this plot.) The first observation is that exploring more paths leads to better F1 scores but larger computation times. Again, we observe that the F1 score quickly reaches a plateau.

The second observation is that the computation time is not strictly linear in the number of explored paths. Indeed, `conv-MMP3` (5.17s) is faster than running sequentially three times `conv-OMP` (which is equivalent to `conv-MMP1` and takes 4.1s); the same applies to `conv-MMP9` and `conv-MMP27`. Indeed, thanks to our efficient algorithmic strategy, we can save intermediate results between similar paths. Note that the amount of saved paths is a complex combinatorial function of the number of paths; between `conv-MMP3` and `conv-MMP9`, there is not much gain.

TABLE I
APPROXIMATION QUALITY OF `conv-OMP` AND `conv-MMP9` ON THE REAL-WORLD DATA SET

Method	PSNR (dB)	Paired Student's t-test
`conv-OMP`	26.56 (±2.97)	Statistics: 3.17, p-value: 0.2%.
`conv-MMP9`	26.79 (±2.81)	

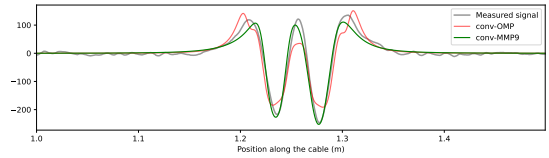

Fig. 4. Example of a 50cm-long MFL response signal (in arbitrary units) and its reconstruction by `conv-MMP9` and `conv-OMP`.

B. Real-world application

Data. The real data set comprises 32 cable portions of 2 m in length, all containing various defects that illustrate the challenges in energy infrastructure maintenance. Maintenance engineers' objective is to detect the position and type of defects using only the MFL signal to intervene in the correct area of an overhead power line (which can be several hundred meters long). Here, no ground-truth information is available about the positions and characteristics of the defects. However, we inspected the cables three times, i.e. the same measurement procedure has been applied three times on each cable. In total, there are 96 signals (32×3).

Metrics. Without ground truth, we only compare the reconstruction quality. In that regard, the Peak Signal-to-Noise Ratio (PSNR) is a common measure. It is expressed in dB and defined, for a true signal $\boldsymbol{u}$ with T samples and its approximation $\hat{\boldsymbol{u}}$, by

$$\mathrm{PSNR}(\boldsymbol{u}, \hat{\boldsymbol{u}}) := 20 \log_{10}(\max \boldsymbol{u}) - 10 \log_{10} \left(\|\hat{\boldsymbol{u}} - \boldsymbol{u}\|^2 / T \right).$$

The "true" signal is the average over the three measurements for each cable portion; we estimate three atoms for each signal.

Results. The results are shown in Tab. I: only `conv-OMP` is compared to `conv-MMP` as we have shown that is the best non-multipath approach for CSC on simulated data. The reconstruction is better for `conv-MMP9` (larger PSNR). Using a paired Student's t-test, we show that our method is even significantly better (p-value 0.2%). Fig. 4 displays the difference in reconstruction on an illustrating example. On this overhead line portion, `conv-MMP9` can find a qualitatively better atom combination than the baseline approach `conv-OMP`.

V. CONCLUSION

We have introduced an efficient algorithm to perform CSC on signals that can be decomposed with a few overlapping prototypical atoms. Our method outperforms common baselines on simulated and real-world data.

REFERENCES

[1] B. J. Owen and B. S. Sathyaprakash, "Matched filtering of gravitational waves from inspiraling compact binaries: Computational cost and template placement," *Physical Review D*, vol. 60, no. 2, pp. 022002, June 1999.

[2] Y. Shi, Z. Nenadic, and X. Xu, "Novel Use of Matched Filtering for Synaptic Event Detection and Extraction," *PLoS ONE*, vol. 5, no. 11, pp. e15517, Nov. 2010.

[3] T. Dupré La Tour, T. Moreau, M. Jas, and A. Gramfort, "Multivariate convolutional sparse coding for electromagnetic brain signals," in *Advances in Neural Information Processing Systems (Neurips)*, Montréal, Canada, 2018, pp. 3292–3302.

[4] S. Liu, Y. Sun, X. Jiang, and Y. Kang, "A Review of Wire Rope Detection Methods, Sensors and Signal Processing Techniques," *Journal of Nondestructive Evaluation*, vol. 39, no. 85, 2020.

[5] K. Mandal, D. Dufour, T. W. Krause, and D. L. Atherton, "Investigations of magnetic flux leakage and magnetic Barkhausen noise signals from pipeline steel," *Journal of Physics D: Applied Physics*, vol. 30, no. 6, pp. 962, 1997.

[6] H.-M. Lei and G.-Y. Tian, "Broken wire detection in coated steel belts using the magnetic flux leakage method," *Insight-Non-Destructive Testing and Condition Monitoring*, vol. 55, no. 3, pp. 126–131, 2013.

[7] S. L. Huang, L. Peng, S. Wang, and W. Zhao, "A basic signal analysis approach for magnetic flux Leakage response," *IEEE Transactions on Magnetics*, vol. 54, no. 10, 2018.

[8] B. Feng, J. Wu, H. Tu, J. Tang, and Y. Kang, "A review of magnetic flux leakage nondestructive testing," *Materials*, vol. 15, no. 20, pp. 7362, 2022.

[9] R. Grosse, R. Raina, H. Kwong, and A. Y. Ng, "Shift-Invariant Sparse Coding for Audio Classification," in *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2007, vol. 23, pp. 149–158.

[10] Y. Zhu and S. Lucey, "Convolutional sparse coding for trajectory reconstruction," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 37, no. 3, pp. 529–540, 2015.

[11] Jean-Baptiste Malagnoux and Matthieu Kowalski, "From convolutional sparse coding to *-nmf factorization of time-frequency coefficients," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 5530–5534.

[12] T. Moreau, L. Oudre, and N. Vayatis, "DICOD: Distributed Convolutional Sparse Coding," in *International Conference on Machine Learning (ICML)*, Stockohlm, Sweden, 2018, pp. 3626–3634, PMLR (80).

[13] E. Zisselman, J. Sulam, and M. Elad, "A local block coordinate descent algorithm for the convolutional sparse coding model," in *Proceedings of the EEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2018, pp. 8200–8209.

[14] B. Wohlberg, "Efficient Algorithms for Convolutional Sparse Representations," *IEEE Transactions on Image Processing*, vol. 25, no. 1, 2016.

[15] S. Van De Geer, "On tight bounds for the lasso," *Journal of Machine Learning Research (JMLR)*, vol. 19, pp. 1–48, 2018.

[16] E. Plaut and R. Giryes, "Matching Pursuit Based Convolutional Sparse Coding," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018, pp. 6847–6851, IEEE.

[17] E. Plaut and R. Giryes, "A greedy approach to $\ell_{\{0,\infty\}}$-based convolutional sparse coding," *SIAM Journal on Imaging Sciences*, vol. 12, no. 1, pp. 186–210, 2019.

[18] A H Song, F J Flores, and D Ba, "Convolutional dictionary learning with grid refinement," *IEEE Transactions on Signal Processing*, vol. 68, pp. 2558–2573, 2020.

[19] V. Papyan, J. Sulam, and M. Elad, "Working locally thinking globally: Theoretical guarantees for convolutional sparse coding," *IEEE Transactions on Signal Processing*, vol. 65, no. 21, pp. 5687–5701, 2017.

[20] Y. Fu, T. Zhu, Y. Xiang, Z. Chen, and L. Cai, "Stability Analysis of $\ell_{0,\infty}$-Norm Based Convolutional Sparse Coding Using Stripe Coherence," *IEEE Transactions on Signal Processing*, vol. 68, pp. 5810–5823, 2020.

[21] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 58, pp. 1094–1121, 2 2012.

[22] J. Wang and P. Li, "Recovery of sparse signals using multiple orthogonal least squares," *IEEE Transactions on Signal Processing*, vol. 65, pp. 2049–2062, 10 2017.

[23] S. Li and L. Fang, "Signal denoising with random refined orthogonal matching pursuit," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, pp. 26–34, 1 2012.

[24] Suhyuk Kwon, Jian Wang, and Byonghyo Shim, "Multipath Matching Pursuit," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2986–3001, 2014.

[25] Pengbo Geng, Jian Wang, and Wengu Chen, "Multipath least squares algorithm and analysis," *Signal Processing*, vol. 174, pp. 107633, 2020.

[26] S. K. Chaitanya and K. Srinivasan, "Equivalent source method based near field acoustic holography using multipath orthogonal matching pursuit," *Applied Acoustics*, vol. 187, pp. 108501, 2022.

[27] M. Jas, T. Dupré La Tour, U. Şimşekli, and A. Gramfort, "Learning the morphology of brain signals using alpha-stable convolutional sparse coding," in *Advances in Neural Information Processing Systems (Neurips)*, Long Beach, CA, USA, 2017, pp. 1100–1109.

[28] N. N. Zatsepin and V. E. Shcherbinin, "Calculation of the magnetostatic field of surface defects Part 1. Field topography of defect models," *Defectoscopy*, vol. 5, no. 50, 1966.

[29] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller, "Deep learning for time series classification: a review," *Data Mining and Knowledge Discovery*, vol. 33, no. 4, pp. 917–963, 2019.