LUND UNIVERSITY

Online Estimation of Multiple Harmonic Signals

FILIP ELVANDER, JOHAN SWÄRD, AND ANDREAS JAKOBSSON

Published in: IEEE Transactions on Audio, Speech and Language Processing
doi:10.1109/TASLP.2016.2634118

Lund 2017

Mathematical Statistics
Centre for Mathematical Sciences
Lund University

Online Estimation of Multiple Harmonic Signals

Filip Elvander, Johan Swärd, and Andreas Jakobsson

Abstract—**In this paper, we propose a time-recursive multi-pitch estimation algorithm using a sparse reconstruction framework, assuming that only a few pitches from a large set of candidates are active at each time instant. The proposed algorithm does not require any training data, and instead utilizes a sparse recursive least squares formulation augmented by an adaptive penalty term specifically designed to enforce a pitch structure on the solution. The amplitudes of the active pitches are also recursively updated, allowing for a smooth and more accurate representation. When evaluated on a set of ten music pieces, the proposed method is shown to outperform other general purpose multi-pitch estimators in either accuracy or computational speed, although not being able to yield performance as good as the state-of-the art methods, which are being optimally tuned and specifically trained on the present instruments. However, the method is able to outperform such a technique when used without optimal tuning, or when applied to instruments not included in the training data.**

Index Terms—**Adaptive signal processing, dictionary learning, group sparsity, multi-pitch estimation, sparse recursive least squares**

I. Introduction

The problem of estimating the fundamental frequency, or pitch, arises in a variety of fields, such as in speech and audio processing, non-destructive testing, and biomedical modeling (see, e.g., [1]–[6], and the references therein). In such applications, the measured signal may often result from several partly simultaneous sources, meaning that both the number of pitches, and the number of overtones of each such pitch, may be expected to vary over the signal. Such would be the case, for instance, in most forms of audio signals. The resulting multi-pitch estimation problem is in general difficult, with one of the most notorious issues being the so-called sub-octave problem, i.e., distinguishing between pitches whose fundamental frequencies are related by powers of two. Both non-parametric, such as methods based on autocorrelation (see, e.g., [7] and references therein), and parametric multi-pitch estimators (see, e.g., [2]) have been suggested, where the latter are often more robust to the sub-octave problem, but rely heavily on accurate *a priori* model order information of both the number of pitches present and the number of harmonic overtones for each pitch Regrettably, the need for accurate model order information is a significant drawback, as such information is typically difficult to obtain and may vary rapidly over the signal. In order to alleviate this, several sparse reconstruction algorithms tailored for multi-pitch estimation have recently been proposed, allowing for estimators that do not require explicit knowledge of the number of sources or their harmonics; for example, in [8], the so-called PEBS estimator was introduced, exploiting the block-sparse structure of the pitch signal. This estimator was then further developed in [9], such that the likelihood of erroneously selecting a sub-octave in place of the true pitch was lowered, while also introducing a self-regularization technique for selecting the penalty parameters. Both these estimators form implicit model order decisions based on one or more tuning parameters that dictate the relative weight of various penalties. As shown in the above cited works, the resulting estimators are able to allow for (rapidly) varying model orders, without significant loss of performance. Earlier works based on sparse representations of signals also include works such as [10], which considers atomic decomposition of audio signals in both the time and the frequency domains.

There have also been methods proposed for multi-pitch estimation and tracking that are source specific, i.e., tailored specifically to sources, e.g., musical instruments, that are known to be present in the signal. In [11], the authors perform multi-pitch estimation on music mixtures by, via a probabilistic framework, matching the signal to a pre-learned dictionary of spectral basis vectors that correspond to instruments known to be present in the signal. A similar source specific idea was used in [12], where pitch estimation was performed by matching the signal to spectral templates learned from individual piano keys. Other methods specifically designed to handle multi-pitch estimation for pianos include [13]–[15]. Another field of research is designing multi-pitch estimators based on a two-matrix factorisation of the short-time Fourier transform, i.e., a non-negative matrix factorization (see, e.g., [16]–[18]). The method has also been used in the sparse reconstruction framework, for instance to learn atoms in order to decompose the signal [19]. A common assumption is also that of spectral smoothness within each pitch, which may also be exploited in order to improve the estimation performance (see, e.g., [13], [17], [20], [21]).

In many audio processing applications, pitch tracking is of great interest and despite being a problem that has been studied for a long time, it still attracts a lot of attention. Over the years, there have been many different approaches for tracking pitches; some of the more recent include particle filters [22], neural networks [23], and Bayesian filtering [24]. Many of these methods require *a priori* model order information, and/or are limited to the single pitch case. The sparse pitch estimators in [8], [9] are robust to these model assumptions, and allow for multiple pitches. However, these estimators process each data frame separately, treating each as an isolated and stationary measurement, without exploiting the information obtained from earlier data frames when forming the estimates. To allow

This work was supported in part by the Swedish Research Council and the Royal Physiographic Society in Lund.

Parts of the material herein has been accepted for publication at the Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, November 2016.

F. Elvander, J. Swärd, and A. Jakobsson are with the Center for Mathematical Sciences, Lund University, SE-22100 Lund, Sweden (email: filip.elvander@matstat.lu.se; js@maths.lth.se, aj@maths.lth.se).

for such correlation over time, the PEBS estimator introduced in [8] was recently extended to exploit the previous pitch estimates, as well as the power distribution of the following frame, when processing the current data frame [25]. In this work, we extend on this effort, but instead propose a fully time-recursive problem formulation using the sparse recursive least squares (RLS) estimator. The resulting estimator does not only allow for more stable pitch estimates as compared to earlier sparse multi-pitch estimators, as more information is used at each time-point, but also decreases the computational burden of each update, as new estimates are formed by updating already available ones. On the other hand, sparse adaptive filtering is a field attracting steadily increasing attention, with, for instance, the sparse RLS algorithm being explored for adaptive filtering in, e.g., [26]–[28]. Other related studies include [29], wherein the authors use a projection approach to solve a recursive LASSO-type problem, and [30], which introduced an online recursive method allowing for an underlying dynamical signal model and the use of sparsity-inducing penalties. Recursive algorithms designed for group-sparse systems have also been introduced, such as the ones presented in [31]–[33], but to the best of our knowledge, no such technique has so-far been applied to the multi-pitch estimation problem. This is the problem we strive to address in this paper. It should be noted that the here presented work differs from many other multi-pitch estimators in that it only exploits the assumption that the signal of interest is generated by a harmonic sinusoidal model. Recently, quite a few methods for multi-pitch estimation adhering to the machine learning paradigm have been proposed (see, e.g., [34], [35]). In these methods, a model is trained on labeled signals, such as, e.g., notes played by individual music instruments, extracting features from the training data that are then used for classification in the estimation stage. As opposed to this, the method presented here is not dependent on being trained on any dataset prior to the estimation.

Our earlier efforts on multi-pitch estimation based on sparse modeling, such as the PEBS [8] and PEBSI-Lite [9] algorithms, have focused on frame-based multi-pitch estimation techniques, with PEBS introducing the use of block sparsity to form the pitch estimates, and PEBSI-Lite refining these ideas and introducing a self-regularization technique to select the required user parameters. In this work, we build on the insights from these algorithms, and expand these ideas by introducing a method that allows for a sample-by-sample updating, in the form of an RLS-like sparse estimator, thereby allowing the estimates to also exploit information available in earlier data samples. The sub-octave problems experienced by PEBS and later alleviated by PEBSI-Lite, with the use of a total-variation penalty enforcing spectral smoothness, is here addressed using an adaptively re-weighted block penalty. Furthermore, we introduce a signal-adaptive updating scheme for the dictionary frequency atoms that allows the proposed method to, e.g., track frequency modulated signals, and alleviates grid mismatches otherwise commonly experienced by dictionary based methods.

The remainder of this paper is organized as follows; in the next section, we introduce the multi-pitch signal model and its corresponding dictionary formulation. Then, in Section III, we introduce the group sparse RLS formulation for multi-pitch estimation, followed by a scheme for decreasing the bias of the harmonic amplitude estimates in Section IV. Section V presents a discussion about various algorithmic considerations. Section VI contains numerical examples illustrating the performance of the proposed estimator on various audio signals. Finally, Section VII concludes upon the work.

A. Notation

In this work, we use lower case non-bold letters such as x to denote scalars and lower case boldface letter such as $\mathbf{x}$ to denote vectors. Upper case bold face letters such as $\mathbf{X}$ are used for matrices. We let diag $(\mathbf{x})$ denote a diagonal matrix formed with the vector $\mathbf{x}$ along its diagonal. Sets are denoted using upper case calligraphic letters such as $\mathcal{A}$. If $\mathcal{A}$ and $\mathcal{B}$ are sets of integers, then $\mathbf{x}_{\mathcal{A}}$ denotes the sub-vector of $\mathbf{x}$ indexed by $\mathcal{A}$. For matrices, $\mathbf{X}_{\mathcal{A},\mathcal{B}}$ denotes the matrix constructed using the rows indexed by $\mathcal{A}$ and columns indexed by $\mathcal{B}$. We use the shorthand $\mathbf{X}_{\mathcal{A}}$ to denote $\mathbf{X}_{\mathcal{A},\mathcal{A}}$. Furthermore, $\bar{[\cdot]}$, $[\cdot]^H$, and $[\cdot]^T$ denotes complex conjugation, conjugate transpose, and transpose, respectively. Also, $|\mathcal{A}|$ is the cardinality of the set $\mathcal{A}$, and $|\mathbf{x}|$ denotes the number of elements in the vector $\mathbf{x}$, unless otherwise stated. Finally, we for vectors $\mathbf{x} \in \mathbb{C}^n$ let $\|\mathbf{x}\|_\ell$ denote the ℓ-norm, defined as

$$\|\mathbf{x}\|_\ell = \left(\sum_{j=1}^{n} |x_j|^\ell \right)^{1/\ell} \tag{1}$$

and use $i = \sqrt{-1}$.

II. Signal model

Consider a measured signal[1], $y(t)$, that is generated according to the model $y(t) = x(t) + e(t)$, where

$$x(t) = \sum_{k=1}^{K(t)} \sum_{\ell=1}^{L_k(t)} w_{k,\ell}(t) \mathrm{e}^{i2\pi f_k(t)\ell t} \tag{2}$$

with $K(t)$ denoting the number of pitches at time t, with fundamental frequencies $f_k(t)$, having $L_k(t)$ harmonics, $w_{k,\ell}(t)$ the complex-valued amplitude of the ℓth harmonic of the kth pitch, and where $e(t)$ denotes a broad-band additive noise. It should be stressed that the number of pitches, as well as their fundamental frequencies, and the number of harmonics for each source, may vary over time. It is worth noting that we here assume a harmonic signal, such as detailed in (2); however, as shown in the numerical section, the proposed method does also work well for somewhat inharmonic signals, such as, e.g., those resulting from a piano.

We here attempt to approximate the measured signal using a sparse representation in an over-complete harmonic basis, see, e.g., [36]. Specifically, as in [8], [9], the signal sources are approximated using a sparse modeling framework containing P

[1]For notational and computational simplicity, we here consider the discrete-time analytic signal of any real-valued measured signal.

candidate pitches, each allowed to have up to $L_{\max}$ harmonics, such that

$$x(t) \approx \sum_{p=1}^{P} \sum_{\ell=1}^{L_{\max}} w_{p,\ell}(t) \mathrm{e}^{i2\pi f_p(t)\ell t} \tag{3}$$

where the dictionary is selected large enough so that (at least) $K(t)$ candidate pitches, $f_p(t)$, reasonably well approximate the true pitch frequencies (see also, e.g., [37], [38]), i.e., such that $P \gg \max_t K(t)$ and $L_{\max} \gg \max_{t,k} L_k(t)$. It should be noted that as the signal is assumed to contain relatively few pitches at each time instance, the resulting amplitude vector will be sparse, although with a harmonic structure reflecting the overtones of the pitches. Furthermore, it may be noted that the frequency grid-points, $f_p(t)$, are allowed to vary with time, which will here be implemented using an adaptive dictionary learning scheme. Using this framework, the pitches present in the signal at time t may be implicitly estimated by identifying the non-zero amplitude coefficients, $w_{p,\ell}(t)$.

III. Group-sparse RLS for pitches

Exploiting the structure of the signal, we introduce the group-sparse adaptive filter, $\mathbf{w}(t)$, which at time t is divided into P groups according to

$$\mathbf{w}(t) = \left[\begin{array}{ccc} \mathbf{w}_1^T(t) & \dots & \mathbf{w}_P^T(t) \end{array} \right]^T \tag{4}$$

$$\mathbf{w}_p(t) = \left[\begin{array}{ccc} w_{p,1}(t) & \dots & w_{p,L_{\max}}(t) \end{array} \right]^T \tag{5}$$

implying that, ideally, only $K(t)$ sub-vectors $\mathbf{w}_p(t)$ will be non-zeros at time t. In order to achieve this, the filter is formed as

$$\hat{\mathbf{w}}(t) = \arg\min_{\mathbf{w}} \; g_t(\mathbf{w}) + h_t(\mathbf{w}) \tag{6}$$

where $\hat{\mathbf{w}}(t)$ denotes the solution of (6), $g_t(\mathbf{w})$ the regular RLS criterion, (see, e.g., [39]), formed as

$$g_t(\mathbf{w}) = \frac{1}{2} \sum_{\tau=1}^{t} \lambda^{t-\tau} \left| y(\tau) - \mathbf{w}^T \mathbf{a}(\tau) \right|^2 \tag{7}$$

and $h_t(\mathbf{w})$ a sparsity inducing penalty function. Note that a similar adaptive filter formulation for estimating sparse data structures was introduced in [27]. However, whereas [27] considered sparse signals, we in this work expand this approach to also consider block sparsity, and specifically the pitch structure. As a result, the dictionary is here formed as

$$\mathbf{a}(t) = \left[\begin{array}{ccc} \mathbf{a}_1^T(t) & \dots & \mathbf{a}_P^T(t) \end{array} \right]^T \tag{8}$$

$$\mathbf{a}_p(t) = \left[\begin{array}{ccc} \mathrm{e}^{i2\pi f_p(t) t} & \dots & \mathrm{e}^{i2\pi f_p(t) L_{\max} t} \end{array} \right]^T \tag{9}$$

and $\lambda \in (0,1)$ being a user-determined forgetting factor. The choice of the forgetting factor λ will reflect assumptions on the variability of the spectral content of the signal, with λ close to 1 implying an almost stationary signal, whereas a smaller value will allow for a quicker adaption to changes in the spectral content. The sparsity inducing function, $h_t(\mathbf{w})$, should be selected as to encourage a pitch-structure in the solution; in [9], which considered multi-pitch estimation on isolated time frames, this function, which then was not a function of time, was selected as

$$h(\mathbf{w}) = \gamma_1 \left\| \mathbf{w} \right\|_1 + \gamma_2 \sum_{p=1}^{P} \left\| \mathbf{F} \mathbf{w}_{\mathcal{G}_p} \right\|_1 \tag{10}$$

where $\mathbf{F}$ is the first difference matrix and $\mathcal{G}_p$ is the set of indices corresponding to the harmonics of the candidate pitch p. The second term of this penalty function is the ℓ_1-norm of the differences between consecutive harmonics and acts as a total variation penalty on the spectral envelope of each pitch. Often referred to as the sparse fused LASSO [40], this penalty was in [9] used to promote solutions with spectral smoothness in each pitch, although requiring some additional refinements to achieve this. To allow for a fast implementation, we will here instead consider the time-varying penalty function

$$h_t(\mathbf{w}) = \gamma_1(t) \left\| \mathbf{w} \right\|_1 + \sum_{p=1}^{P} \gamma_{2,p}(t) \left\| \mathbf{w}_{\mathcal{G}_p} \right\|_2 \tag{11}$$

where $\gamma_1(t)$ and $\gamma_{2,p}(t)$ are non-negative regularization parameters. This penalty, often called the sparse group LASSO [41] when combined with a squared ℓ_2-norm model fit term, is reminiscent of the one used in the PEBS method introduced in [8], and belongs to the class of methods utilizing mixed norms for sparse signal estimation (see, e.g., [42]). The second term of this penalty function, the pitch-wise ℓ_2-norm, has a group-sparsifying effect, encouraging solutions where active harmonics are grouped together into a few number of pitches. As the frequency content of different pitches may be quite similar due to overlapping, or close to overlapping, harmonics, the group penalty thus prevents erroneous activation of isolated harmonics, while still allowing the different groups to retain harmonics shared by different sources (see also [8], [9]). In the case of overlapping harmonics in the signal, i.e., the presence of two pitches which share at least one harmonic, the ℓ_2-norm will favor solutions of the optimization problem (6) in which the powers of these harmonics are shared among the two pitches. The precise level of sharing is decided by the relative powers of the unique harmonics of each pitch so that the pitch having unique harmonics with more power will also be assigned a larger share of the power corresponding to the overlapping harmonics. In the case of the the two pitches having unique harmonics with equal combined power, the power of the overlapping harmonics will also be shared equally. However, when, as in [8], using fixed penalty parameters $\gamma_1(t)$ and $\gamma_{2,p}(t)$, the resulting estimate has been shown to be prone to mistaking a pitch for its sub-octave (see also [9]). In order to discourage this type of erroneous solutions, we will herein introduce a way of adaptively choosing the group sparsity parameter, $\gamma_{2,p}(t)$, as further discussed below.

We note that $g_t(\mathbf{w})$, as defined in (7), may be expressed in matrix form as

$$g_t(\mathbf{w}) = \frac{1}{2} \left\| \mathbf{\Lambda}_{1:t}^{1/2} \mathbf{y}_{1:t} - \mathbf{\Lambda}_{1:t}^{1/2} \mathbf{A}_{1:t} \mathbf{w} \right\|_2^2 \tag{12}$$

where

$$\mathbf{y}_{\tau:t} = \left[\begin{array}{ccc} y(\tau) & \dots & y(t) \end{array} \right]^T \tag{13}$$

$$\mathbf{A}_{\tau:t} = \left[\begin{array}{ccc} \mathbf{a}(\tau) & \dots & \mathbf{a}(t) \end{array} \right]^T \tag{14}$$

and with $\mathbf{\Lambda}_{1:t} = \text{diag}\left(\left[\begin{array}{cccc} \lambda^{t-1} & \lambda^{t-2} & \ldots & 1 \end{array}\right]\right)$. To simplify notation, define

$$\mathbf{R}(t) \triangleq \mathbf{A}_{1:t}^H \mathbf{\Lambda}_{1:t} \mathbf{A}_{1:t} \quad (15)$$

$$\mathbf{r}(t) \triangleq \mathbf{A}_{1:t}^H \mathbf{\Lambda}_{1:t} \mathbf{y}_{1:t} \; . \quad (16)$$

With these definitions, the minimization in (6) may be formed using proximal gradient iterations, (see, e.g., [43]), such that the jth iteration may be expressed as

$$\hat{\mathbf{w}}^{(j+1)}(t) = \arg\min_{\mathbf{w}} \frac{1}{2s(t)} \left\| \boldsymbol{\nu}^{(j)} - \mathbf{w} \right\|_2^2 + h_t(\mathbf{w}) \quad (17)$$

where

$$\boldsymbol{\nu}^{(j)} = \hat{\mathbf{w}}^{(j)}(t) + s(t) \left[\mathbf{r}(t) - \mathbf{R}(t) \hat{\mathbf{w}}^{(j)}(t) \right] \quad (18)$$

with $s(t)$ denoting the step-size. We note that this update is reminiscent of the one presented in [27], which considers the problem of ℓ_1-regularized recursive least squares, although it should be noted that the ℓ_1-norm for complex vectors in [27] is defined to be the sum of the absolute values of the real and imaginary parts separately, whereas we here use the more common definition, as given by (1). In [27], the authors motivate their minimization algorithm by casting it as an EM-algorithm using reasoning from [44], as well as some further assumptions about properties of the signal. By studying the zero sub-differential equations for (17), it can be shown that the closed form solution for each group p can be computed separately as (see, e.g., equations (54)-(55) and (32)-(38) in [8]; for further details, see also [41])

$$\tilde{\boldsymbol{\nu}}_{\mathcal{G}_p}^{(j)} = S_1 \left(\boldsymbol{\nu}_{\mathcal{G}_p}^{(j)}, s(t)\gamma_1(t) \right) \quad (19)$$

$$\hat{\mathbf{w}}_{\mathcal{G}_p}^{(j+1)}(t) = S_2 \left(\tilde{\boldsymbol{\nu}}_{\mathcal{G}_p}^{(j)}, s(t)\gamma_{2,p}(t) \right) \quad (20)$$

where $S_1(\cdot)$ and $S_2(\cdot)$ are the soft thresholding operators corresponding to the ℓ_1- and ℓ_2-norms, respectively, i.e.,

$$S_1(\mathbf{z}, \alpha) = \frac{\max(|\mathbf{z}| - \alpha, 0)}{\max(|\mathbf{z}| - \alpha, 0) + \alpha} \odot \mathbf{z} \quad (21)$$

$$S_2(\mathbf{z}, \alpha) = \frac{\max(\|\mathbf{z}\|_2 - \alpha, 0)}{\max(\|\mathbf{z}\|_2 - \alpha, 0) + \alpha} \mathbf{z} \quad (22)$$

where, in (21), $|\mathbf{z}|$ denotes the vector obtained by taking the absolute value of each element of the vector $\mathbf{z}$, the max function operates element-wise on the vector $\mathbf{z}$, and $\odot$ denotes element-wise multiplication. Furthermore, as $\mathbf{R}(t)$ and $\mathbf{r}(t)$ can be expressed as

$$\mathbf{R}(t) = \sum_{\tau=1}^{t} \lambda^{t-\tau} \mathbf{a}(\tau) \mathbf{a}^H(\tau) \quad (23)$$

$$\mathbf{r}(t) = \sum_{\tau=1}^{t} \lambda^{t-\tau} y(\tau) \bar{\mathbf{a}}(\tau) \quad (24)$$

these entities can be updated according to

$$\mathbf{R}(t) = \lambda \mathbf{R}(t-1) + \mathbf{a}(t)\mathbf{a}^H(t) \quad (25)$$

$$\mathbf{r}(t) = \lambda \mathbf{r}(t-1) + y(t)\bar{\mathbf{a}}(t) \; , \quad (26)$$

when new samples become available. Here, $\bar{(\cdot)}$ denotes complex conjugation.

IV. Refined amplitude estimates

In general, the sparsity promoting penalty function $h_t(\mathbf{w})$ will introduce a downward bias on the magnitude of the amplitude estimates formed by (6). However, as the support of $\hat{\mathbf{w}}(t)$ will reflect the fundamental frequencies present in the signal, we can refine the amplitude estimates by minimizing a least squares criterion. As this problem only considers amplitudes of harmonics of pitches that are believed to be in the signal, we do not need to use any sparsity inducing penalties and can therefore avoid the magnitude bias. This will be analogous to estimating the amplitudes of each harmonic using recursive least squares assuming that the support of the filter is known. To this end, let

$$\mathcal{S}(t) = \bigcup_{p \in \mathcal{A}(t)} \mathcal{G}_p \quad (27)$$

$$\mathcal{A}(t) = \left\{ p \mid \left\| \hat{\mathbf{w}}_{\mathcal{G}_p}(t) \right\|_2 > 0 \right\} \; , \quad (28)$$

i.e., $\mathcal{A}(t)$ is the set of active pitches determined by the sparse filter $\hat{\mathbf{w}}(t)$, at time t, and $\mathcal{S}(t)$ is the index set corresponding to the harmonics of these pitches. Let $\breve{\mathbf{w}}(t)$ denote the refined amplitude estimates at time t. Given $\hat{\mathbf{w}}(t)$, and thereby $\mathcal{S}(t)$, we update this filter according to

$$\breve{\mathbf{w}}_k(t) = 0 \; , k \notin \mathcal{A}(t) \quad (29)$$

$$\breve{\mathbf{w}}_{\mathcal{S}(t)}(t) = \arg\min_{\mathbf{w} \in \mathbb{C}^{|\mathcal{S}(t)|}} \mathbf{w}^H \mathbf{R}_{\mathcal{S}(t)} \mathbf{w} - \mathbf{w}^H \mathbf{r}_{\mathcal{S}(t)} - \mathbf{r}_{\mathcal{S}(t)}^H \mathbf{w} + \xi \left\| \mathbf{w} - \breve{\mathbf{w}}_{\mathcal{S}(t)}(t-1) \right\|_2^2 \quad (30)$$

where $\mathbf{R}_{\mathcal{S}(t)}(t)$ is the $|\mathcal{S}(t)| \times |\mathcal{S}(t)|$ matrix constructed by the rows and columns of $\mathbf{R}(t)$ indexed by $\mathcal{S}(t)$ and $\mathbf{r}_{\mathcal{S}(t)}(t)$ is the $|\mathcal{S}(t)|$ dimensional vector constructed by the elements of $\mathbf{r}(t)$, indexed by $\mathcal{S}(t)$. The second term of (30) is a proximal term that will promote a smooth trajectory for the magnitude of the filter coefficients, where the parameter $\xi > 0$ controls the smoothness. This type of smoothness-promoting penalty has earlier been used, for instance, to enforce temporal continuity in NMF applications [45]. To avoid inverting large matrices, we split the solving of (30) into $\mathcal{A}(t)$ problems of size L_{max} using a cyclic coordinate descent scheme (see also, e.g., [26]). To this end, define the index sets

$$\mathcal{Q}_p = \mathcal{S}(t) \setminus \mathcal{G}_p \; , p \in \mathcal{A}(t) \; , \quad (31)$$

i.e., the indices corresponding to harmonics that are not part of pitch p. Considering only terms in the cost function in (30) that depend on harmonics of the pth pitch, we can form an update of the corresponding filter coefficients according to

$$\breve{\mathbf{w}}_{\mathcal{G}_p}(t) = \arg\min_{\mathbf{w} \in \mathbb{C}^{L_{\text{max}}}} \mathbf{w}^H \mathbf{R}_{\mathcal{G}_p} \mathbf{w} - \mathbf{w}^H \mathbf{r}^{(p)} - \mathbf{r}^{(p)H} \mathbf{w} + \xi \left\| \mathbf{w} - \breve{\mathbf{w}}_{\mathcal{G}_p}(t-1) \right\|_2^2 \quad (32)$$

where

$$\mathbf{r}^{(p)} = \mathbf{r}_{\mathcal{G}_p} - \mathbf{R}_{\mathcal{G}_p, \mathcal{Q}_p} \tilde{\mathbf{w}}_{\mathcal{Q}_p} \; . \quad (33)$$

The vector $\tilde{\mathbf{w}}_{\mathcal{Q}_p} \in \mathbb{C}^{|\mathcal{Q}_p|}$ contains the (partially updated) filter coefficients that correspond to other pitches than p, i.e.,

$$\tilde{\mathbf{w}}_{\mathcal{G}_q} = \begin{cases} \breve{\mathbf{w}}_{\mathcal{G}_q}(t) & \text{if updated} \\ \breve{\mathbf{w}}_{\mathcal{G}_q}(t-1) & \text{if not updated} \end{cases} \quad (34)$$

for $q \neq p$. By setting the gradient of (32) with respect to $\mathbf{w}$ to zero, we find the update of $\check{\mathbf{w}}_{\mathcal{G}_p}(t)$ to be

$$\check{\mathbf{w}}_{\mathcal{G}_p}(t) = \left(\mathbf{R}_{\mathcal{G}_p} + \xi\,\mathbf{I}\right)^{-1}\left(\mathbf{r}^{(p)} + \xi\,\check{\mathbf{w}}_{\mathcal{G}_p}(t-1)\right). \tag{35}$$

V. Algorithmic considerations

We proceed to examine some implementation aspects of the presented algorithm, first discussing the appropriate choice of the penalty parameters, then possible computational speed-ups, as well as ways of adaptively updating the used pitch dictionary.

A. Parameter choices

In order to discourage solutions containing erroneous sub-octaves, we here propose to update the group penalty parameter, in iteration j of the filter update (17), as

$$\gamma_{2,p}(t) = \gamma_2(t)\max\left(1, \frac{1}{\left|\hat{w}_{p,1}^{j-1}(t)\right| + \epsilon}\right) \tag{36}$$

where $\left|\hat{w}_{p,1}^{j-1}(t)\right|$ is the estimated amplitude of the first harmonic of group p, obtained in iteration $j-1$, with $\epsilon \ll 1$ being a user-specified parameter selected to avoid a division by zero. In this paper, we use $\epsilon = 10^{-5}$. As sub-octaves will typically have missing first harmonics, such a choice will encourage shifting power from the sub-octave to the proper pitch. Similar types of re-weighted penalties have earlier been used to enhance sparsity in the estimated signal (see, e.g., [46], [47]). Studies using many different kinds of pitch signals indicate that the overall performance of the algorithm is relatively insensitive to the choice of the parameter $s(t)$, which may typically be selected in the range $s(t) \in \left[10^{-5}, 10^{-3}\right]$. Here, we use $s(t) = 10^{-4}$. The choice of the penalty parameters $\gamma_1(t)$ and $\gamma_2(t)$ can be made using inner-products between the dictionary and the signal. Letting Δ denote the time-lag, define

$$\eta(t, \mu) = \mu\left\|\mathbf{\Lambda}_{1:\Delta}\mathbf{A}_{t-\Delta:t}^{H}\mathbf{y}_{t-\Delta:t}\right\|_{\infty} \tag{37}$$

where $\mu \in (0, 1)$. A good rule of thumb is choosing $\gamma_1(t)$ in the neighborhood of (37) with $\mu = 0.1$, whereas a corresponding reasonable value for $\gamma_2(t)$ is $\mu = 1$. Empirically, the performance of the algorithm has been seen to be robust to variations of these choices of μ. This method emulates choosing the values of the penalty parameters based on the correlation between the signal and the dictionary in a finite window. Here, the window length, Δ, is determined by the forgetting factor, λ, and by how much correlation one is willing to lose as a result from the truncation. For example, selecting

$$\Delta = \frac{\log(0.01)}{\log\lambda} \tag{38}$$

will yield a window such that the excluded samples will contribute to less than 0.01 of the correlation. It should be noted that for smoothly varying signals, $\gamma_1(t)$ and $\gamma_2(t)$ only need to be updated infrequently.

B. Iteration speed-up

As the signal is assumed to have a sparse representation in the dictionary $\mathbf{a}(t)$, one may expect updates of the coefficients of many groups, here indexed by q, to result in zero amplitude estimates. As such groups do not contribute to the pitch estimates, these groups would preferably be excluded from the updates in (17)-(18). If assuming the support of $\mathbf{w}(t)$ to be constant for all t, one could thus sequentially discard such groups from the updating step, and thereby decrease computation time. However, as generally pitches may disappear and then re-appear, as well as drift in frequency over time, we will here only exclude the groups q from the updating steps temporarily. That is, if at time τ, we have $\left\|\hat{\mathbf{w}}_{\mathcal{G}_q}\right\|_2 < \tilde{\epsilon}$, where $\tilde{\epsilon} \ll 1$, the group q is considered not to be present in the signal and is therefore excluded from the updating steps for a waiting period, T. After that period, it is again included in the updates, allowing it to again appear in the signal. Defining the set $\mathcal{U}$, indexing the groups that are considered active, the group q is adaptively included and excluded from $\mathcal{U}$ depending on the size of $\left\|\hat{\mathbf{w}}_{\mathcal{G}_q}\right\|_2$. If the signal can be assumed to have slowly varying spectral content, meaning that the support of $\mathbf{w}(t)$ is also varying slowly, the waiting period T may be chosen to be quite long, as to improve the computational efficiency. In general, choosing T as to correspond to a few milliseconds allows for a speed-up of the algorithm while at the same time enabling it to track the time evolution of $\mathbf{w}(t)$.

C. Dictionary learning

In general, a signal's pitch frequencies may vary over time, for instance, due to vibrato. Applying the filter updating scheme using fixed grid-points will therefore result in rapidly changing support of the filter or energy leakage between adjacent blocks of the filter, here indexed by p. In order to overcome this problem, and to allow for smooth tracking of pitches over time, we propose a scheme for adaptively updating the dictionary of candidate pitches. This adaptive adjustment scheme also allows for the use of a grid with coarser resolution than would otherwise be possible. Let $\mathcal{T} = \{\tau_k\}_k$ be the set of time points in which the dictionary is updated. As only groups $\hat{\mathbf{w}}_{\mathcal{G}_p}(\tau_k)$ with non-zero power are considered to be present in the signal, one only has to adjust the fundamental frequencies of these. Assuming that the current estimate of such a candidate pitch frequency is $f_p(\tau_{k-1})$, one only needs to consider adjusting it on the interval $f_p(\tau_{k-1}) \pm \frac{1}{2}\delta_{f,k}(t)$, where $\delta_{f,k}(t)$ denotes the current grid-point spacing. The update can be formed using the approximate non-linear least squares method in [48], [2], where, instead of L_{max}, one uses the harmonic order corresponding to the non-zero components of $\hat{\mathbf{w}}_{\mathcal{G}_p}(\tau_k)$. This refined estimate is obtained by first forming the residual, and adding back the current group of harmonics, whereafter the approximate non-linear least squares method is applied to update the frequencies. The adjusted frequency $f_p(\tau_k)$ is then used to update the dictionary on the time interval $[\tau_k, \tau_{k+1})$. After updating the dictionary, the filter coefficient estimates will, due to the recursive nature of the method, be partly based on the old dictionary and partly on the updated one. It is thus very likely that after the dictionary update

Algorithm 1 The PEARLS algorithm

Initialise $\hat{\mathbf{w}}(0) \leftarrow \mathbf{0}$, $\mathbf{R}(0) \leftarrow \mathbf{0}$, $\mathbf{r}(0) \leftarrow \mathbf{0}$
$t \leftarrow 1$
repeat {Recursive update scheme}
  $\mathbf{R}(t) \leftarrow \lambda \mathbf{R}(t-1) + \mathbf{a}(t)\mathbf{a}^H(t)$
  $\mathbf{r}(t) \leftarrow \lambda \mathbf{r}(t-1) + y(t)\bar{\mathbf{a}}(t)$
  $j \leftarrow 0$
  $\hat{\mathbf{w}}^{(j)}(t) \leftarrow \hat{\mathbf{w}}(t-1)$
  repeat {Proximal gradient update}
    $\boldsymbol{\nu}^{(j)} \leftarrow \hat{\mathbf{w}}^{(j)}(t) + s(t)\left[\mathbf{r}(t) - \mathbf{R}(t)\hat{\mathbf{w}}^{(j)}(t)\right]$
    $\hat{\mathbf{w}}^{(j+1)}(t) \leftarrow \arg\min_{\mathbf{w}} \frac{1}{2s(t)} \left\|\boldsymbol{\nu}^{(j)} - \mathbf{w}\right\|_2^2 + h_t(\mathbf{w})$
    $j \leftarrow j+1$
  until convergence
  $\hat{\mathbf{w}}(t) \leftarrow \hat{\mathbf{w}}^{(j)}(t)$
  Determine $\mathcal{A}(t)$ and $\mathcal{S}(t)$
  $\breve{\mathbf{w}}_k(t) \leftarrow 0$, $k \notin \mathcal{A}(t)$
  $\breve{\mathbf{w}}_{\mathcal{S}(t)}(t) = \arg\min_{\mathbf{w}\in\mathbb{C}^{|\mathcal{S}(t)|}} \mathbf{w}^H \mathbf{R}_{\mathcal{S}(t)} \mathbf{w} - \mathbf{w}^H \mathbf{r}_{\mathcal{S}(t)} - \mathbf{r}^H_{\mathcal{S}(t)}\mathbf{w}$
  $+\xi \left\|\mathbf{w} - \breve{\mathbf{w}}_{\mathcal{S}(t)}(t-1)\right\|_2^2$
  Update active set $\mathcal{U}$
  if $t \in \mathcal{T}$ **then**
    Update dictionary
  end if
  $t \leftarrow t+1$
until end of signal

the phase component of the two filter coefficient parts will differ. To avoid this, we instead incorporate the phase into the dictionary, thus obtaining a filter coefficient with zero phase. This is accomplished by estimating the phases at the same time as the frequencies are updated in the dictionary updating step. Each estimated phase is then multiplied with the corresponding column of the dictionary, thus including the phases into the dictionary. This update corresponds to changing (8) and (9) to

$$\mathbf{a}(t,\boldsymbol{\phi}) = \left[\begin{array}{ccc} \mathbf{a}_1^T(t,\boldsymbol{\phi}_1) & \dots & \mathbf{a}_P^T(t,\boldsymbol{\phi}_P) \end{array}\right]^T \quad (39)$$

$$\mathbf{a}_p(t,\boldsymbol{\phi}_p) = \left[\begin{array}{ccc} e^{i2\pi f_p(t)t + i\pi\phi_{p_1}} & \dots & e^{i2\pi f_p(t)L_{\max}t + i\pi\phi_{p_{L_{\max}}}} \end{array}\right]^T \quad (40)$$

where

$$\boldsymbol{\phi} = \left[\begin{array}{ccc} \boldsymbol{\phi}_1^T & \dots & \boldsymbol{\phi}_P^T \end{array}\right]^T \quad (41)$$

$$\boldsymbol{\phi}_p = \left[\begin{array}{ccc} \phi_{p_1}^T & \dots & \phi_{p_{L_{\max}}} \end{array}\right]^T \quad (42)$$

with ϕ_{p_ℓ} denoting the phase of the ℓth harmonic of the pth pitch. With this formulation the phases are incorporated into the dictionary, thus rendering the amplitudes real valued.

Together with the discussed algorithmic considerations, the presented time-recursive multi-pitch estimator is detailed in Algorithm 1. The algorithm is termed the Pitch Estimation using dictionary-Adaptive Recursive Least Squares (PEARLS) method[2].

[2]An implementation in MATLAB may be found at http://www.maths.lu.se/staff/andreas-jakobsson/publications/.

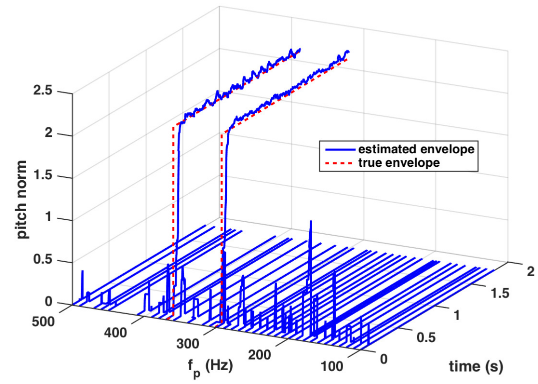

Fig. 1. Pitch frequency and pitch norm estimates, i.e., estimates of $f_p(t)$ and $\left\|\breve{\mathbf{w}}_{\mathcal{G}_p}(t)\right\|_2$ as produced by PEARLS when applied to a simulated two-pitch signal with fundamental frequencies 302 and 369 Hz, respectively, deviating from the original dictionary grid points by 2 and 1 Hz respectively.

VI. Numerical results

In this section, we evaluate the performance of the proposed PEARLS algorithm using both simulated signals and real audio recordings.

A. Simulated signals

To demonstrate the effect of the smoothing parameter, ξ, as well as the ability of PEARLS to smoothly track the amplitudes of pitches, we first consider an illustrative example with a two-pitch signal. Figure 1 shows the time evolution of the pitch frequency and pitch norm estimates, i.e., estimates of $f_p(t)$ and $\left\|\breve{\mathbf{w}}_{\mathcal{G}_p}(t)\right\|_2$, as produced by PEARLS when applied to a two-pitch signal with fundamental frequencies 302 and 369 Hz, respectively, where both pitches are constituted by 5 harmonics each. Both pitches enter the signal after 90 ms, reaching their maximum amplitudes momentarily and keeping them for the rest of the signal duration. The signal was sampled at 11 kHz. The settings for PEARLS was $L_{\max} = 10$, $\lambda = 0.995$, and the smoothing parameter was $\xi = 10^4$. The original pitch frequency grid was chosen so that the true pitch frequencies deviated from the closest grid points by 2 and 1 Hz, respectively. As can be seen from the figure, the estimate initially, before the pitch signals appear, contains several spurious pitch estimates, but then quickly finds the pitch signals when these appear in the data. At this point, the spurious peaks are suppressed and the estimates are seen to well follow the true pitch envelopes. It is worth noting that both the response time and the steady state variance of the estimates will be influenced by the choice of the smoothing parameter, ξ. Figures 2 and 3 illustrate this effect by considering the response time, defined as the time required for the PEARLS amplitude estimate to reach 95% of its peak value, and the steady state amplitude variance, respectively. The signal considered is the same as in Figure 1. As can be seen from the figures, a higher value of ξ implies a longer response time for PEARLS, while at the same time promoting a more smooth pitch norm trajectory, just as could be expected.

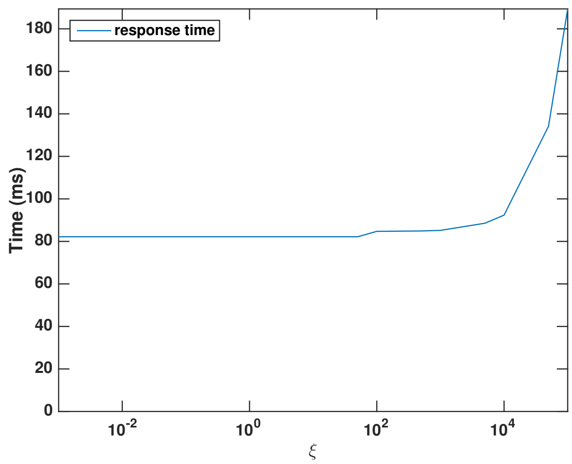

Fig. 2. Respone time for different values of the smoothing parameter ξ.

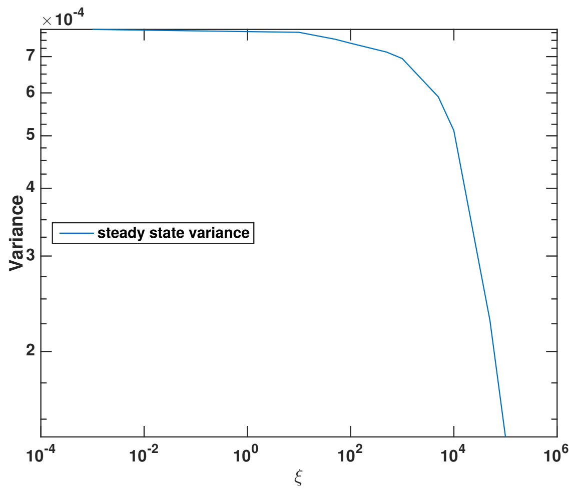

Fig. 3. Steady state variance of the pitch norm estimate for different values of the smoothing parameter ξ.

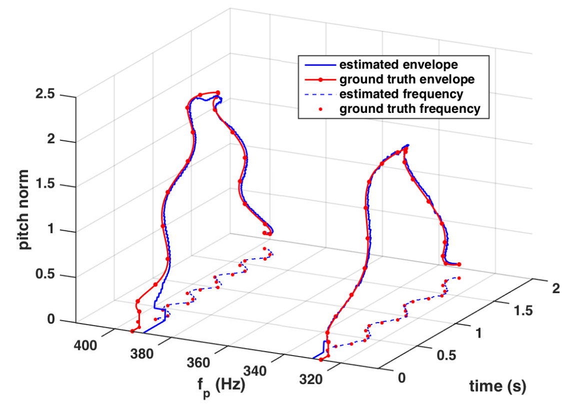

Fig. 4. Pitch frequency and pitch norm estimates, i.e., estimates of $f_p(t)$ and $\left\|\breve{\mathbf{w}}_{\mathcal{G}_p}(t)\right\|_2$, as produced by PEARLS when applied to a simulated two-pitch signal with fundamental frequencies that oscillate according to sine waves.

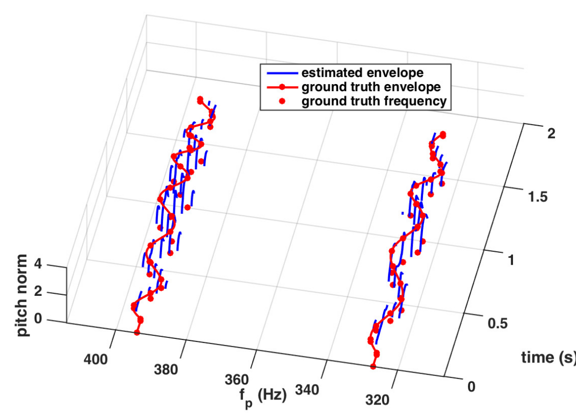

Fig. 5. Pitch frequency and pitch norm estimates, i.e., estimates of $f_p(t)$ and $\left\|\breve{\mathbf{w}}_{\mathcal{G}_p}(t)\right\|_2$, as produced by PEARLS when applied to a simulated two-pitch signal with fundamental frequencies that oscillate according to sine waves. Here, the dictionary learning scheme is excluded from Algorithm 1.

The PEARLS algorithm is not restricted to form estimates of stationary pitches; it is also able to cope with amplitude and frequency modulated signals. In Figure 4, PEARLS has been applied to a two-pitch signal with fundamental frequencies that oscillate according to sine waves with frequencies 2 and 3 Hz on the intervals 327 ± 2 Hz and 394 ± 3 Hz, respectively. Also, the pitch norms are not constant, but are amplitude modulated according to a Hamming window. As can be seen, PEARLS is able to track the two pitches smoothly both in frequency and in pitch norm. Here, the pitches consisted of 5 and 7 harmonics, respectively. The signal was sampled at 11 kHz, with PEARLS using the same settings as above. As comparison, Figure 6 presents a corresponding plot for the multi-pitch estimator ESACF [7], using recommended settings. As ESACF only estimates pitch frequencies, pitch norm estimates have been obtained using least squares, assuming known harmonic orders. ESACF is a frame based estimator and the signal was therefore here subdivided into 30ms windows. As can be seen, the ESACF estimates deviate from the true pitch frequencies, causing the amplitude estimates to degrade. Figure 5 demonstrates the usefulness of using the dictionary learning procedure. In this figure, PEARLS is again applied to the signal with two frequency modulated pitches, but this time the dictionary learning scheme is excluded from Algorithm 1. As can be seen in the figure, PEARLS is still able to estimate the frequency content, as well as the pitch norms, but the tracking is now performed by different elements of $\breve{\mathbf{w}}(t)$, as the frequency modulation causes the different candidate pitches to become activated and then deactivated, with the activation-deactivation cycles following the periods of the frequency modulation. Also, there is some power-sharing between adjacent pitch groups of $\breve{\mathbf{w}}(t)$ at time points where the frequency modulating sinusoids change sign. In contrast, the dictionary learning scheme allows for a much smoother tracking as the movable dictionary elements counters the activation-deactivation phenomenon, which can be observed in Figure 4.

B. Real audio

We proceed to evaluate the performance of PEARLS on the Bach10 dataset [49]. This dataset consists of ten excerpts from chorals composed by J. S. Bach, and have been

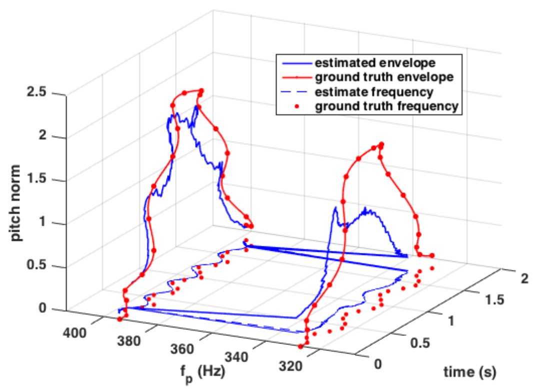

Fig. 6. Pitch frequency, i.e., estimates of $f_p(t)$, as produced by ESACF when applied to a simulated two-pitch signal with fundamental frequencies that oscillate according to sine waves. The pitch norms, i.e., $\left\|\check{\mathbf{w}}_{\mathcal{G}_p}(t)\right\|_2$, have been estimated by applying least squares to the ESACF pitch frequency estimates using oracle harmonic orders.

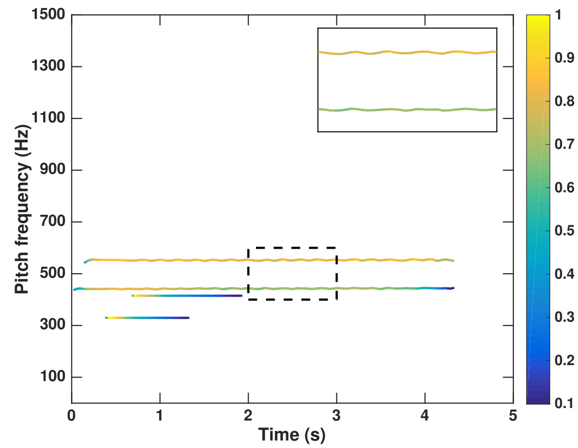

Fig. 7. Ground truth for a signal consisting of two trumpets and two pianos. The amplitude of each pitch, i.e., the pitch norm, is illustrated by the color of each track. The amplitudes have been normalized so that the maximal amplitude is 1.

TABLE I
PERFORMANCE MEASURES FOR THE PEARLS, PEBSI-LITE, BW15, AND ESACF ALGORITHMS, WHEN EVALUATED ON THE BACH10 DATASET.

	PEARLS	PEBSI-Lite	BW15	ESACF
Accuracy	0.437	0.449	0.515	0.269
Precision	0.683	0.631	0.684	0.471
Recall	0.548	0.609	0.675	0.386

arranged to be performed by an ensemble consisting of a violin, a clarinet, a saxophone, and a bassoon, with each excerpt being 25-42 seconds long. The algorithm settings for PEARLS were $\lambda = 0.985$, $\xi = 10^3$, $L_{\text{max}} = 6$, and the dictionary was updated every 10 ms using 45 ms of past signal samples. Each music piece, originally sampled at 44.1 kHz, was down-sampled to 11.025 kHz. The PEARLS estimates were compared to ground truth values with a time-resolution of one reference point every 30 ms. The ground truth fundamental frequencies were obtained by applying the single-pitch estimator YIN [50] to each separate channel with manual correction of obvious errors. The results are presented in Table I, presenting values of the performance measures *Accuracy*, *Precision*, and *Recall*, as defined in [51]. As in [51], an estimated fundamental frequency is associated with a ground truth fundamental frequency if it lies within a quarter-tone, or 3%, of the ground truth fundamental frequency. For comparison, Table I also includes corresponding performance measures for the PEBSI-Lite [9] and ESACF algorithms. The values for PEBSI-Lite and ESACF were originally presented in [9], and the settings for these algorithms are the same as is presented there. Also presented in Table I are performance measures obtained when applying the method presented in [35], hereafter referred to as BW15, after the authors and year of publication, to the same dataset. Being trained on databases of music instrument, this method uses probabilistic latent component analysis to produce pitch estimates and is specifically tailored to estimate pitches in music signals. The frequency resolution of the obtained estimates corresponds to that of the Western chromatic scale, i.e., to the keys of the piano. As can be seen, PEARLS clearly outperforms ESACF and performs on par with PEBSI-Lite when considering these measures, although it should be stressed that PEARLS has significantly lower computational complexity than PEBSI-Lite. The BW15 methods performs better than the other presented methods, including PEARLS, for this dataset. This is as the performance of the BW15 estimate was formed when using an *a posteriori* thresholding of the obtained estimate, optimally selecting the threshold level as to maximize the performance measures; this in order to illustrate the best possible performance achievable for BW15. However, several other choices of possible threshold levels resulted in BW15 performing worse than both PEARLS and PEBSI-Lite. Furthermore, the BW15 estimator is sensitive to mismatches between the examined signal and the training dataset used to construct its priors. This is illustrated by applying the BW15 and PEARLS estimators to a signal consisting of two (harmonic) trumpet notes and two (inharmonic) piano notes. The trumpets are playing the notes A4 and D♭5, corresponding to the fundamental frequencies 440 and 554.37 Hz, whereas the pianos are playing the notes E4 and G♯4, corresponding to the fundamental frequencies 329.65 and 415.3 Hz. The signal was sampled at 11.025 kHz. The ground truth pitches can be seen in Figure 7. Here, the amplitude, i.e., the pitch norm, of each pitch is illustrated by the color of each track. The amplitude has been normalized so that the maximum amplitude is equal to one. The corresponding estimates produced by PEARLS (using the same settings as for the Bach10 dataset) and BW15 are presented in Figures 8 and 9, respectively. As can be seen from Figure 8, PEARLS is able to correctly identify both the trumpet and the piano pitches, despite the pianos being inharmonic and thereby differing from the assumed signal model, as given in (2). Note that PEARLS is also able to smoothly track the frequency modulation caused by that trumpets are playing with vibrato, which can be more clearly seen from the zoomed-in portions of Figures 7 and 8. In contrast, as seen in Figure 9, BW15 is able to correctly identify the piano pitches (note that pianos were

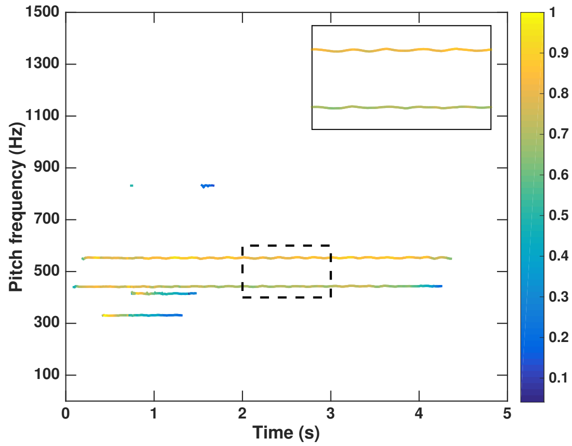

Fig. 8. Estimates produced by PEARLS when applied to a signal with two trumpets as well as two pianos. The amplitude of each pitch, i.e., the pitch norm, is illustrated by the color of each track. The amplitudes have been normalized so that the maximal amplitude is 1.

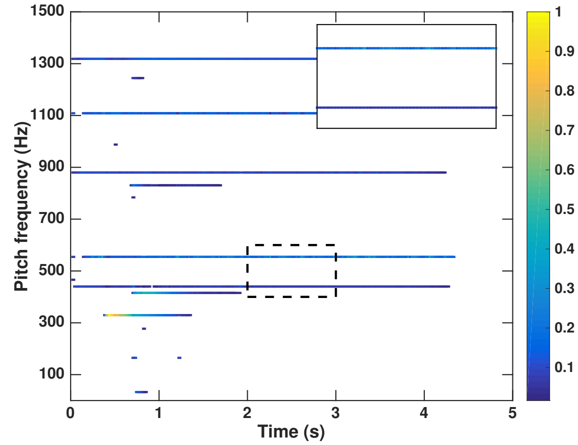

Fig. 9. Estimates produced by BW15 when applied to a signal with two trumpets as well as two pianos. The magnitudes of the estimates are illustrated by the color of the pitch tracks. The magnitudes have been normalised so that the maximal magnitude is 1.

included in the training dataset used by the authors of [35]), but instead of identifying the sinusoidal content corresponding to the trumpets (which are not in the training dataset) as originating from only two pitches, several of the individual harmonics are instead being assigned individual pitches. It may be noted that the method does not accurately represent the vibratos; this as the estimates of BW15 are restricted to correspond to the keys of the piano. It should further be noted that the pitches indicated as being the most significant by BW15 are not those corresponding to the true fundamental frequencies, but instead higher order harmonics. This problem is arguably due to the mismatch between the content of the signal and the database used to train the method. Thus, for this example, it is not possible to recover the true pitches by thresholding the solution of BW15, as the thresholding would eliminate true pitch candidates before getting rid of the erroneous ones. Although the estimates produced by BW15 could arguably be improved by extending its training data to also include trumpets, this example illustrates that basing estimation on exploiting the features of a signal model, as PEARLS does, can be beneficial in terms of the generality of the estimator, even in the face of slight deviations from the assumed signal model, which in this case takes the form of inharmonicity for the pianos. It can be noted that an interesting future development would be to combine the benefits from training a hidden Markov model, as is done in BW15, with the more robust approach in PEARLS.

Another recent method that would be of interest to consider in this respect would be the one presented in [21], which also exhibits some conceptual similarities with the herein presented algorithm. Notably, the sparsifying role played by the ℓ_1-norm herein is in [21] formed by instead determining the significant spectral peaks using an estimate of the noise floor. The pitch selection, herein formed using the group-wise ℓ_2-norm, is in [21] made by matching spectral content with that of components in a large training data set, which is also used to measure the power concentration for low-order harmonics, as well as a synchronicity measure. The relative weighting of these components is selected using training data. Using a greedy approach, the method in [21] then iteratively adds candidate pitches to the estimate; the power allocation between pitches that have overlapping harmonics is resolved using an interpolation scheme utilizing the power of harmonics unique to each candidate pitch. In contrast, the number of active pitches is herein decided by the optimal point of (6), where candidate pitches not contained in the signal should be assigned zero power. It can also be noted that the optimization problem presented here does not favor spectral smoothness; rather, the ℓ_2-norm will favor collecting as much power as possible into a few candidate pitches. The power of overlapping harmonics will therefore tend to be allocated to pitches with more prominent unique harmonics.

Using a MATLAB implementation of PEARLS on a 2.68 GHz PC, the average running time for the Bach pieces was 20 minutes. The Bach pieces were on average 33 seconds long[3]. For PEBSI-Lite, the average running time was 54 minutes, with the signal being divided into non-overlapping frames of length 30 ms.

As an illustration of the performance of PEARLS on the Bach10 dataset, Figures 10 and 11 present the estimated fundamental frequencies obtained using ESACF and PEARLS, respectively, for the piece *Ach, Gott und Herr*, as compared to the ground truth for each instrument. Here, in order to make a fair comparison of the computational complexities of the estimators, the ESACF estimate was computed on windows of length 30 ms, where two consecutive windows overlapped in all but one sample. Although ESACF can arguably be applied to windows with smaller overlap, this setup meant that ESACF would produce pitch tracks with the same time resolution as PEARLS. This resulted in an average running time of 11 minutes per music piece, that is, about half that of PEARLS. As can be seen from the figures, PEARLS is considerably

[3]We note that the current implementation has not exploited that the filter updating step (17) can be done for all P candidate pitches in parallel. Similarly, the computations for PEBSI-Lite can also be parallelized, as each time frame can be processed in isolation.

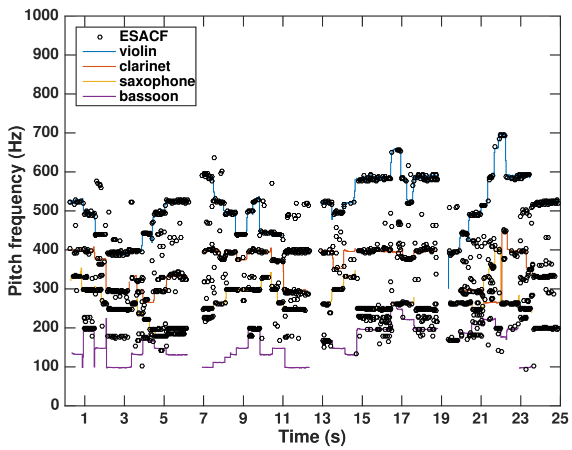

Fig. 10. Pitch tracks produced by ESACF when applied to a 25 seconds excerpt of J. S. Bach's *Ach, Gott und Herr* performed by a violin, a clarinet, a saxophone, and a bassoon.

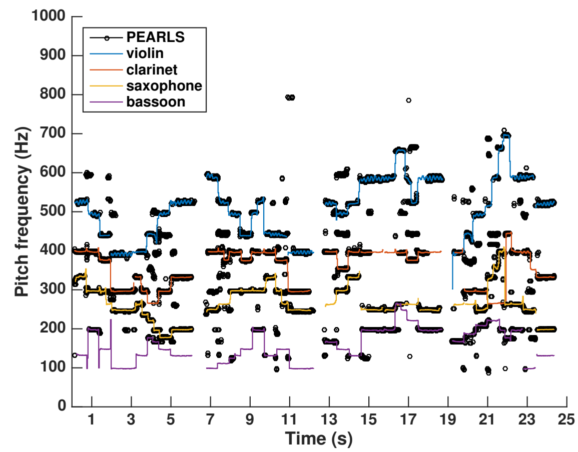

Fig. 11. Pitch tracks produced by PEARLS when applied to a 25 seconds excerpt of J. S. Bach's *Ach, Gott und Herr* performed by a violin, a clarinet, a saxophone, and a bassoon.

better at tracking the instruments than ESACF. In Figure 12, the corresponding results for BW15 are shown. The figure has been truncated at 1000 Hz to simplify inspection, although pitch estimates with fundamental frequencies higher than 1000 Hz did occur repeatedly. From the figure, it is clear that BW15 is better able to track the bassoon (which is included in the method's training data) than either PEARLS or ESACF. It can also be noted that the discrete nature of the BW15 estimator prevents it from tracking smaller frequency variations, such as vibratos.

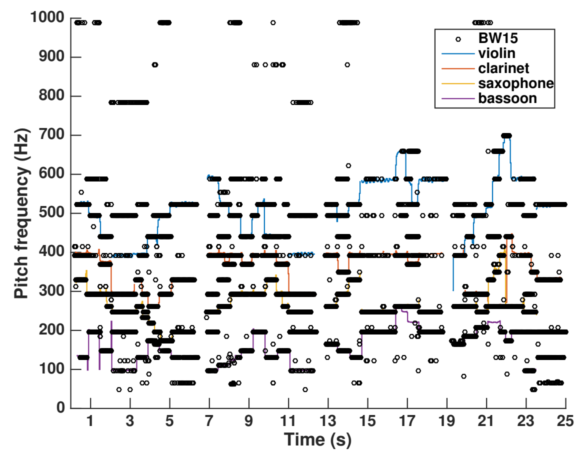

Fig. 12. Pitch tracks produced by BW15 when applied to a 25 seconds excerpt of J. S. Bach's *Ach, Gott und Herr* performed by a violin, a clarinet, a saxophone, and a bassoon.

VII. Conclusions

In this work, we have presented a time-recursive multi-pitch estimation algorithm, based on a both sparse and group-sparse reconstruction technique. The method has been shown to be able to accurately track multiple pitches over time, in fundamental frequency as well as in amplitude, without requiring prior knowledge of the number of pitches nor the number of harmonics present in the signal. Furthermore, we have presented a scheme for adaptively changing the signal dictionary, thereby providing robustness against grid mismatch, as well as allowing for smooth tracking of frequency modulated signals. We have shown that the proposed method yields accurate results when applied to real data, outperforming other general purpose multi-pitch estimators in either estimation accuracy and/or computational speed. The method has further been shown to be robust to deviations from the assumed signal model, although it is not able to yield performance as good as that achievable by a state-of-the art method being optimally tuned and specifically trained on the present instruments. However, the method is able to outperform such a technique when used without optimal tuning, or when applied to instruments not included in the training data.

References

[1] A. Klapuri and M. Davy, *Signal Processing Methods for Music Transcription*, Springer, 2006.

[2] M. Christensen and A. Jakobsson, *Multi-Pitch Estimation*, Morgan & Claypool, San Rafael, Calif., 2009.

[3] M. Müller, *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*, Springer International Publishing, 2015.

[4] R. B. Randall, *Vibration-Based Condition Monitoring: Industrial, Aerospace and Automotive Applications*, John Wiley & Sons, Chichester, UK, 2011.

[5] M. A. Little, P. E. McSharry, E. J. Hunter, J. Spielman, and L. O. Ramig, "Suitability of Dysphonia Measurements for Telemonitoring of Parkinson's disease," *IEEE Trans. Biomed. Eng.*, vol. 56, no. 4, pp. 1015–102, April 2009.

[6] P. Somervuo, A. Harma, and S. Fagerlund, "Parametric Representations of Bird Sounds for Automatic Species Recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 6, pp. 2252–2263, November 2006.

[7] T. Tolonen and M. Karjalainen, "A computationally efficient multipitch analysis model," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 6, pp. 708–716, 2000.

[8] S. I. Adalbjörnsson, A. Jakobsson, and M. G. Christensen, "Multi-Pitch Estimation Exploiting Block Sparsity," *Elsevier Signal Processing*, vol. 109, pp. 236–247, April 2015.

[9] F. Elvander, T. Kronvall, S. I. Adalbjörnsson, and A. Jakobsson, "An Adaptive Penalty Multi-Pitch Estimator with Self-Regularization," *Elsevier Signal Processing*, vol. 127, pp. 56–70, October 2016.

[10] M. D. Plumbley, S. A. Abdallah, T. Blumensath, and M. E. Davies, "Sparse representations of polyphonic music," *Signal Processing*, vol. 86, no. 3, pp. 417–431, March 2006.

[11] M. Bay, A.F. Ehmann, J.W. Beauchamp, P. Smaragdis, and J.S. Downie, "Second Fiddle is Important Too: Pitch Tracking Individual Voices in Polyphonic music," in *13th Annual Conference of the International Speech Communication Association*, Portland, September 2012, pp. 319–324.

[12] A. Dessein, A. Cont, and G. Lemaitre, "Real-Time Polyphonic Music Transcription With Non-Negative Matrix Factorisation and Beta-Divergence," in *Proceedings of the 11th International Society for Music*

Information Retrieval Conference, Utrecht, NL, August 2010, pp. 489–494.

[13] V. Emiya, R. Badeau, and B. David, "Multipitch Estimation of Piano Sounds Using a New Probabilistic Spectral Smoothness Principle," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 18, no. 6, pp. 1643–1654, August 2010.

[14] C. Kim, W. Chang, S-H. Oh, and S-Y. Lee, "Joint Estimation of Multiple Notes and Inharmoncity Coefficient Based on f0-Triplet for Automatic Piano Transcription," *IEEE Signal Process. Lett.*, vol. 21, no. 12, pp. 1536–1540, December 2014.

[15] A. Cogliati, Z. Duan, and B. Wohlberg, "Piano Music Transcription with Fast Convolutional Sparse Coding," in *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing*, Boston, MA, Sept 2015, pp. 1–6.

[16] P. Smaragdis and J.C. Brown, "Non-Negative Matrix Factorization for Polyphonic Music Transcription," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2003, pp. 177–180.

[17] E. Vincent, N. Bertin, and R. Badeau, "Adaptive Harmonic Spectral Decomposition for Multiple Pitch Estimation," *IEEE Trans. Acoust., Speech, Language Process.*, vol. 18, no. 3, pp. 528–537, March 2010.

[18] N. Bertin, R. Badeau, and E. Vincent, "Enforcing Harmonicity and Smoothness in Bayesian Non-Negative Matrix Factorization Applied to Polyphonic Music Transcription," *IEEE Trans. Acoust., Speech, Language Process.*, vol. 18, no. 3, pp. 538–549, 2010.

[19] M. Genussov and I. Cohen, "Multiple fundamental frequency estimation based on sparse representations in a structured dictionary," *Digit. Signal Process.*, vol. 23, no. 1, pp. 390–400, Jan. 2013.

[20] A. Klapuri, "Multiple fundamental frequency estimation based on harmonicity and spectral smoothness," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 6, pp. 804–816, 2003.

[21] C. Yeh, A. Roebel, and X. Rodet, "Multiple Fundamental Frequency Estimation and Polyphony Inference of Polyphonic Music Signals," *IEEE Trans. Acoust., Speech, Language Process.*, vol. 18, no. 6, pp. 1116–1126, August 2010.

[22] G. Zhang and S. Godsill, "Tracking Pitch Period Using Particle Filters," in *IEEE Workhop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, New York, October 2013.

[23] K. Han and D. Wang, "Neural Networks For Supervised Pitch Tracking in Noise," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, May 2014.

[24] H. Hajimolahoseini, R. Amirfattahi, S. Gazor, and H. Soltanian-Zadeh, "Robust Estimation and Tracking of Pitch Period Using an Efficient Bayesian Filter," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 7, pp. 1219–1229, July 2016.

[25] S. Karimian-Azari, A. Jakobsson, J. R. Jensen, and M. G. Christensen, "Multi-Pitch Estimation and Tracking using Bayesian Inference in Block Sparsity," in *23rd European Signal Processing Conference*, Nice, Aug. 31-Sept. 4 2015.

[26] D. Angelosante, J. A. Bazerque, and G. B. Giannakis, "Online Adaptive Estimation of Sparse Signals: Where RLS meets the ℓ_1-Norm," *IEEE Trans. Signal Process.*, vol. 58, 2010.

[27] B. Babadi, N. Kalouptsidis, and V. Tarokh, "SPARLS: The Sparse RLS Algorithm," *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 4013–4025, August 2010.

[28] N. Vaswani and J. Zhan, "Recursive Rrecovery of Sparse Signal Sequences From Compressive Measurements: A Reveiew," *IEEE Trans. Signal Process.*, vol. 64, no. 13, pp. 3523–3549, July 2016.

[29] Y. Kopsinis, K. Slavakis, and S. Theodoridis, "Online Sparse System Identification and Signal Reconstruction Using Projections Onto Weighted ℓ_1 Balls," *IEEE Trans. Signal Process.*, vol. 59, no. 3, pp. 936–952, March 2011.

[30] E. C. Hall and R. M. Willett, "Online Convex Optimization in Dynamic Environments," *IEEE J. Sel. Topics in Signal Processing*, vol. 9, no. 4, pp. 647–662, June 2015.

[31] Y. Chen and A. O. Hero, "Recursive $\ell_{1,\infty}$ Group Lasso," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 3978–3987, Aug. 2012.

[32] E. Eksioglu, "Group sparse RLS algorithms," *International Journal of Adaptive Control and Signal Processing*, vol. 28, pp. 1398–1412, 2014.

[33] S. Jiang and Y. Gu, "Block-Sparsity-Induced Adaptive Filter for Multi-Clustering System Identification," *IEEE Trans. Signal Process.*, vol. 63, no. 20, pp. 5318–5330, October 2015.

[34] B. Fuentes, R. Badeau, and G. Richard, "Harmonic Adaptive Latent Component Analysis of Audio and Application to Music Transcription," *IEEE Trans. Acoust., Speech, Language Process.*, vol. 21, no. 9, pp. 1854–1866, September 2013.

[35] E. Benetos and T. Weyde, "An Efficient Temporally-Constrained Probabilistic Model for Multiple-Instrument Music Transcription," in *Proceedings of the 16th International Society for Music Information Retrieval Conference*, Malaga, Spain, October 2015.

[36] R. Gribonval and E. Bacry, "Harmonic decomposition of audio signals with matching pursuit," *IEEE Trans. Signal Process.*, vol. 51, no. 1, pp. 101–111, jan. 2003.

[37] 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.

[38] 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.

[39] S. Haykin, *Adaptive Filter Theory (4th edition)*, Prentice Hall, Inc., Englewood Cliffs, N.J., 2002.

[40] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and Smoothness via the Fused Lasso," *Journal of the Royal Statistical Society B*, vol. 67, no. 1, pp. 91–108, January 2005.

[41] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, "A Sparse-Group Lasso," *Journal of Computational and Graphical Statistics*, vol. 22, no. 2, pp. 231–245, 2013.

[42] M. Kowalski, "Sparse Regression Using Mixed Norms," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 303 – 324, 2009.

[43] P. Combettes and V. Wajs, "Signal recovery by proximal forward-backward splitting," *SIAM Jour. Multiscale Modeling & Simulation*, vol. 4, pp. 1168–1200, 2005.

[44] M. A. T. Figueiredo and R. D. Nowak, "An EM Algorithm for Wavelet-Based Image Restoration," *IEEE Trans. Image Process.*, vol. 12, no. 8, pp. 906–916, August 2003.

[45] T. Virtanen, "Monaural Sound Source Separation by Nonnegative Matrix Factorization With Temporal Continuity and Sparseness Criteria," *IEEE Trans. Acoust., Speech, Language Process.*, vol. 15, no. 3, pp. 1066–1074, March 2007.

[46] 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.

[47] 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.

[48] M. G. Christensen, P. Stoica, A. Jakobsson, and S. H. Jensen, "Multi-pitch estimation," *Signal Processing*, vol. 88, no. 4, pp. 972–983, April 2008.

[49] Z. Duan and B. Pardo, "Bach10 dataset," http://music.cs.northwestern.edu/data/Bach10.html, Accessed December 2015.

[50] A. de Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *J. Acoust. Soc. Am.*, vol. 111, no. 4, pp. 1917–1930, 2002.

[51] M. Bay, A. F. Ehmann, and J. S. Downie, "Evaluation of Multiple-F0 Estimation and Tracking Systems," in *International Society for Music Information Retrieval Conference*, Kobe, Japan, October 2009.

Filip Elvander received his M.Sc. from Lund University in Industrial Engineering and Management in 2015, Sweden, and is currently working towards a Ph.D. in Mathematical Statistics at Lund University. His research interests include sparse modeling, robust estimation, and convex modeling and approximation techniques in statistical signal processing and spectral analysis.

Johan Swärd (S'12) received his M.Sc. in Industrial Engineering and Management, and his Licenciate of Technology from Lund University, Sweden, in 2012 and 2015, respectively,. He is currently working towards a Ph.D. in Mathematical Statistics at Lund University. He has been a visiting researcher at the Department of Systems Innovations at Osaka University, Japan, and Stevens Institute of Technology, New Jersey, USA. His research interests include machine learning and applications of sparse and convex modeling in statistical signal processing and spectral analysis.

Andreas Jakobsson (S'95-M'00-SM'06) received his M.Sc. from Lund Institute of Technology and his Ph.D. in Signal Processing from Uppsala University in 1993 and 2000, respectively. Since, he has held positions with Global IP Sound AB, the Swedish Royal Institute of Technology, King's College London, and Karlstad University, as well as held an Honorary Research Fellowship at Cardiff University. He has been a visiting researcher at King's College London, Brigham Young University, Stanford University, Katholieke Universiteit Leuven, and University of California, San Diego, as well as acted as an expert for the IAEA. He is currently Professor and Head of Mathematical Statistics at Lund University, Sweden. He has published his research findings in about 200 refereed journal and conference papers, and has filed five patents. He has also authored a book on time series analysis (Studentlitteratur, 2013 and 2015), and co-authored (together with M. G. Christensen) a book on multi-pitch estimation (Morgan & Claypool, 2009). He is a member of The Royal Swedish Physiographic Society, a member of the EURASIP Special Area Team on Signal Processing for Multisensor Systems (2015-), a Senior Member of IEEE, and an Associate Editor for Elsevier Signal Processing. He has previously also been a member of the IEEE Sensor Array and Multichannel (SAM) Signal Processing Technical Committee (2008-2013), an Associate Editor for the IEEE Transactions on Signal Processing (2006-2010), the IEEE Signal Processing Letters (2007-2011), the Research Letters in Signal Processing (2007-2009), and the Journal of Electrical and Computer Engineering (2009-2014). His research interests include statistical and array signal processing, detection and estimation theory, and related application in remote sensing, telecommunication and biomedicine.