A Parametric Method for Multi-Pitch Estimation

Filip Elvander

Master's thesis
2015:E27

Lund University

Faculty of Engineering
Centre for Mathematical Sciences
Mathematical Statistics

Centrum Scientiarum Mathematicarum

Abstract

This thesis proposes a novel method for multi-pitch estimation. The method operates by posing pitch estimation as a sparse recovery problem which is solved using convex optimization techniques. In that respect, it is an extension of an earlier presented estimation method based on the group-LASSO. However, by introducing an adaptive total variation penalty, the proposed method requires fewer user supplied parameters, thereby simplifying the estimation procedure. The method is shown to have comparable to superior performance in low noise environments when compared to three standard multi-pitch estimation methods as well as the predecessor method. Also presented is a scheme for automatic selection of the regularization parameters, thereby making the method more user friendly. Used together with this scheme, the proposed method is shown to yield accurate, although not statistically efficient, pitch estimates when evaluated on synthetic speech data.

Contents

1 Introduction

This thesis is concerned with estimating the fundamental frequencies, or pitch frequencies, of multi-pitch signals. A pitch is defined as a set of harmonically related sinusoids, i.e., sinusoids whose frequencies all are integer multiples of a single common frequency. This means that the frequency content of a pitch can be expressed as the set

$$\mathbf{\Omega}_k \subseteq \left\{\omega_{k,\ell} \mid \omega_{k,\ell} = \omega_{k,1}\ell \, , \ell = 1, 2, \ldots, L_k\right\} \tag{1}$$

where $\omega_{k,1}$ is referred to as the angular fundamental frequency of the pitch $\mathbf{\Omega}_k$ and the individual frequency components constituting the pitch are referred to as harmonics or partials. Furthermore, L_k is the harmonic order, i.e., the highest order harmonic of the pitch. Pitch estimation can mean both estimation of the fundamental frequency $\omega_{k,1}$ and all the harmonics $\omega_{k,\ell}$, i.e., the set of coefficients $\{\ell\}$. Often, only the former is considered and the terms pitch estimation and fundamental frequency estimation are then used interchangeably. In this thesis, we are concerned with estimating the fundamental frequencies $\omega_{1,1}, \omega_{2,1}, \ldots, \omega_{K,1}$ of a given K-pitch signal without assuming *a priori* knowledge of neither the number of pitches K nor the number of harmonics for each pitch. In many of the experiments in this thesis, temporal frequencies $f_{k,\ell}$ instead of angular frequencies $\omega_{k,\ell}$ will be used. The connection between the two is

$$f_{k,\ell} = \frac{\omega_{k,\ell}}{2\pi} f_s \tag{2}$$

where f_s is the sampling frequency. In those cases, pitch estimation refers to estimating the temporal fundamental frequencies $f_{1,1}, f_{2,1}, \ldots, f_{K,1}$. When there is no risk of confusion, both angular and temporal fundamental frequencies will be referred to as fundamental frequencies or pitch frequencies.

1.1 Applications of pitch estimation

Pitch estimation is a problem arising in a variety of fields, not least in audio processing. It is a fundamental building block in several music information retrieval applications such as automatic music transcription, i.e., automatic sheet music generation from audio [1]. Pitch estimation could also be used as a component in methods for cover song detection and music querying, possibly improving currently available services. For example, the popular query service Shazam [2] operates by matching hashed portions of spectrograms of user provided samples against a large music database. As a change of instrumentation would alter the spectrogram of a song, such algorithms can only identify recordings of a song that are very similar to the actual recording present in the data base. Thus, services such as Shazam might fail to identify, e.g., acoustic alternate versions of rock songs. A query algorithm based on pitch estimation could on the other

hand correctly match the acoustic version to the original electrified one as it would recognize, e.g., the main melody. The applicability of pitch estimation to music is due to the fact that the notes produced by many instruments used in Western tonal music, e.g., woodwind instruments such as the clarinet, exhibit a structure that is fairly true to the harmonic sinusoidal structure in (1) [3]. However, for some plucked stringed instruments such as the guitar and the piano, the tension of the string results in the harmonics deviating from perfect integer multiples of the fundamental frequency, a phenomenon called inharmonicity. For some instruments, such as the piano, there are models describing the structure of the inharmonicity based on physical properties of the instrument [4]. Dealing with inharmonicity in full generality is a research area in its own right and will not be considered in this work.

1.2 Previous research

Estimating the fundamental frequencies of multi-pitch signals is generally a hard problem. There are a lot of methods available, see, e.g., [5], but many of them require *a priori* model order knowledge, i.e., they require knowledge of the number of pitches present in the signal, as well as the number of active harmonics for each pitch. Three such methods will be used in this thesis as reference estimators. The first method, here referred to as ORTH, exploits orthogonality between the signal and noise subspaces to form pitch frequency estimates. The second method is an optimal filtering method based on the Capon estimator and therefore here referred to as Capon. The third method is an approximate non-linear least squares method, here referred to as ANLS. All three methods are described in detail in chapters 4.7, 3.5, and 2.7 in [5], respectively.

Methods not requiring *a priori* model order knowledge have also been proposed. For example, [6] uses a sparse dictionary representation of the signal and regularization penalties to implicitly choose the model order. A similar, but less general, method was introduced in [7], which used a dictionary specifically tailored to piano notes for estimating pitch frequencies generated by pianos. Other source specific methods include [8] and [9]. In [9] and [10], pitch estimation is based on the assumption of spectral smoothness, i.e., the amplitudes of the harmonics within a pitch are assumed to be of comparable magnitudes. This is an assumption that will be used also in this thesis. Another field of research is performing multi-pitch estimation, often in the context of automatic music transcription, by decomposing the spectrogram of the signal into two matrices, one that describes the frequency content of the signal and one that describes the time activation of the frequency components. This method makes use of the non-negative matrix factorization, first introduced in this context in [11] and since then widely used, such as, e.g., [12]. There are also more statistical approaches to multi-pitch estimation, posing the estimation as a Bayesian inference problem (see e.g. [13]).

1.3 Organisation of the thesis

This thesis is organized as follows: Section 2 introduces some concepts and methods from convex optimization that in this thesis are used to perform multi-pitch estimation. Section 3 describes the assumed signal model as well as the proposed estimation method. In the same section, some numerical results obtained using Monte Carlo simulation are presented and comparisons to other pitch estimation methods are made. As the proposed method requires user defined regularization parameter, Section 4 explores ways of systematically choosing these parameters. The same section presents a self-regularizing version of the proposed method. The refined method is then evaluated using synthetic data modelled on authentic speech signals and is compared to other pitch estimation methods. Section 5 offers some conclusions based on the findings and Section 6 suggests further research areas.

2 Convex optimization

In this thesis, multi-pitch estimation will be posed as solving a convex optimization problem. Therefore, this section briefly presents some methods for solving such problems. Convex optimization problems can be solved by using publicly available convex minimizers such as the interior point methods SeDuMi [14] or SDPT3 [15]. However, increasing the number of data samples, these methods will become computationally cumbersome. Therefore, the Alternating Direction Method of Multipliers, abbreviated ADMM, will be used in this work. This algorithm class attempts to merge the effectiveness of dual ascent with the robustness of augmented Lagrangian methods. Below is an outline of these two methods and how they are combined into the ADMM. For more details, see e.g. [15].

2.1 Dual ascent and augmented Lagrangians

Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ be a convex function, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{A} \in \mathbb{R}^{m \times n}$ and consider the convex optimization problem

$$\begin{aligned} &\min_{\mathbf{x}} \ f(\mathbf{x}) \\ &\text{s.t } \mathbf{A}\mathbf{x} = \mathbf{b} \end{aligned} \tag{3}$$

The Lagrangian $L(\mathbf{x}, \mathbf{y})$ and its dual function, $g(\mathbf{y})$, with dual variable $\mathbf{y} \in \mathbb{R}^m$, are defined as

$$\begin{aligned} L(\mathbf{x}, \mathbf{y}) &= f(\mathbf{x}) + \mathbf{y}^T(\mathbf{A}\mathbf{x} - \mathbf{b}) \\ g(\mathbf{y}) &= \inf_{\mathbf{x}} L(\mathbf{x}, \mathbf{y}) = -f^*(-\mathbf{A}^T\mathbf{y}) - \mathbf{b}^T\mathbf{y} \end{aligned} \tag{4}$$

where f^* is the complex conjugate of f. Assuming that strong duality holds, we have that

$$\min_{\mathbf{x}} f(\mathbf{x}) = \max_{\mathbf{y}} g(\mathbf{y}) \tag{5}$$

Thus, under the assumption that f is strictly convex, the optimal primal point $\mathbf{x}^\star$ can be retrieved from the optimal dual point $\mathbf{y}^\star$ as

$$\mathbf{x}^\star = \operatorname*{argmin}_{\mathbf{x}} L(\mathbf{x}, \mathbf{y}^\star) \tag{6}$$

The dual ascent method attempts to solve the dual and primal problems using gradient ascent of the dual problem. Assuming that g is differentiable, its gradient is given by $\nabla g(\mathbf{y}) = \mathbf{A}\mathbf{x} - \mathbf{b}$, i.e., the residual of the primal equality constraint. The dual ascent method operates by iteratively solving

$$\begin{aligned} \mathbf{x}^{(k+1)} &= \operatorname*{argmin}_{\mathbf{x}} L(\mathbf{x}, \mathbf{y}^{(k)}) \\ \mathbf{y}^{(k+1)} &= \mathbf{y}^{(k)} + \alpha^{(k)} \nabla g(\mathbf{y}^{(k)}) = \mathbf{y}^{(k)} + \alpha^{(k)}(\mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{b}) \end{aligned} \tag{7}$$

where $\alpha^{(k)}$ is the step size of the algorithm at step k. The strength of the dual ascent method is that it allows for splitting the problem in a number of simpler subproblems in the case of f being separable in the variable $\mathbf{x}$, that is if $f(\mathbf{x})$ can be written as

$$f(\mathbf{x}) = \sum_{i=1}^{N} f_i(\mathbf{x}_i) \tag{8}$$

where $\mathbf{x}_i \in \mathbb{R}^{n_i}$ and $\sum_{i=1}^{N} = n$. If one partitions the matrix $\mathbf{A}$ according to this separation, i.e., $\mathbf{A} = [\mathbf{A}_1, \ldots, \mathbf{A}_N]$, the Lagrangian can be decomposed as

$$L(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N} L_i(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N} \Big(f_i(\mathbf{x}_i) + \mathbf{y}^T \mathbf{A}_i \mathbf{x}_i - \frac{1}{N} \mathbf{y}^T \mathbf{b} \Big) \tag{9}$$

Thus, the updating of $\mathbf{x}^{(k+1)}$ in (7) can be split into N subproblems as

$$\begin{aligned} \mathbf{x}_i^{(k+1)} &= \operatorname*{argmin}_{\mathbf{x}} L_i(\mathbf{x}, \mathbf{y}^{(k)}),\ i = 1, \ldots, N \\ \mathbf{y}^{(k+1)} &= \mathbf{y}^{(k)} + \alpha^{(k)}(\mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{b}) \end{aligned} \tag{10}$$

which may, for example, be distributed to and solved separately by different CPUs. However, the drawback of dual ascent is that it requires rather strong assumptions about the convexity of f to guarantee convergence. To improve the robustness of the method, (3) may instead be augmented as

$$\begin{aligned} &\min_{\mathbf{x}} f(\mathbf{x}) + \rho/2 ||\mathbf{A}\mathbf{x} - \mathbf{b}||_2^2 \\ &\text{s.t } \mathbf{A}\mathbf{x} = \mathbf{b} \end{aligned} \tag{11}$$

where ρ is a positive scalar. This problem obviously has the same solution as the original one, as the added penalty term will be zero for any feasible point $\mathbf{x}$. The Lagrangian of this problem is

$$L_\rho(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}) + \rho/2||\mathbf{A}\mathbf{x} - \mathbf{b}||_2^2 + \mathbf{y}^T(\mathbf{A}\mathbf{x} - \mathbf{b}) \tag{12}$$

with associated dual function $g_\rho(\mathbf{y}) = \inf_{\mathbf{x}} L_\rho(\mathbf{x}, \mathbf{y})$. This formulation requires less restrictive assumptions for the dual function to be differentiable than the original formulation. Also, the primal and dual problems can be solved as before using the iterative scheme

$$\begin{aligned} \mathbf{x}^{(k+1)} &= \operatorname*{argmin}_{\mathbf{x}} L_\rho(\mathbf{x}, \mathbf{y}^{(k)}) \\ \mathbf{y}^{(k+1)} &= \mathbf{y}^{(k)} + \rho(\mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{b}) \end{aligned} \tag{13}$$

where the step size $\alpha^{(k)}$ in (7) now is fixed and identical to ρ for all k. Convergence under this formulation, which is called the Method of Multipliers, can be shown to be more robust than under the original scheme. The reason for choosing ρ as step size is that this automatically yields dual feasibility. Primal and dual feasibility for a point $(\mathbf{x}, \mathbf{y})$ in (3) is, respectively

$$\begin{aligned} \mathbf{A}\mathbf{x} - \mathbf{b} &= \mathbf{0} \\ \nabla f(\mathbf{x}) + \mathbf{A}^T\mathbf{y} &= \mathbf{0} \end{aligned} \tag{14}$$

As we from (13) have

$$\mathbf{x}^{(k+1)} = \operatorname*{argmin}_{\mathbf{x}} L_\rho(\mathbf{x}, \mathbf{y}^{(k)}) \tag{15}$$

it follows that

$$\nabla_{\mathbf{x}} L_\rho(\mathbf{x}^{(k+1)}, \mathbf{y}^{(k)}) = \mathbf{0} \tag{16}$$

Using (12), it follows directly that

$$\begin{aligned} \mathbf{0} &= \nabla_{\mathbf{x}} L_\rho(\mathbf{x}^{(k+1)}, \mathbf{y}^{(k)}) \\ &= \nabla f(\mathbf{x}^{(k+1)}) + \mathbf{A}^T\big(\mathbf{y}^{(k)} + \rho(\mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{b})\big) \\ &= \nabla f(\mathbf{x}^{(k+1)}) + \mathbf{A}^T\mathbf{y}^{(k+1)} \end{aligned} \tag{17}$$

i.e., dual feasibility is obtained in every iteration of (13). Eventually, the optimal point $(\mathbf{x}^\star, \mathbf{y}^\star)$ is reached as the primal residual $\mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{b}$ converges to zero. However, it should be noted that the augmented Lagrangian L_ρ will no longer be separable in $\mathbf{x}$ if f is, meaning that the problem of finding $\operatorname*{argmin}_{\mathbf{x}} L_\rho(\mathbf{x}, \mathbf{y}^{(k)})$ cannot be split. In the following section, it will be presented how the ADMM attempts to remedy this.

2.2 Alternating Direction Method of Multipliers (ADMM)

Consider the problem

$$\begin{aligned} &\min_{\mathbf{x},\mathbf{z}} \; f_1(\mathbf{x}) + f_2(\mathbf{z}) \\ &\text{s.t } \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{c} \end{aligned} \tag{18}$$

where $\mathbf{x} \in \mathbb{R}$, $\mathbf{z} \in \mathbb{R}^m$, $\mathbf{A} \in \mathbb{R}^{p \times n}$, $\mathbf{B} \in \mathbb{R}^{p \times m}$, $\mathbf{c} \in \mathbb{R}^p$, and f_1 and f_2 are convex functions. The augmented Lagrangian of this problem is

$$L_\rho(\mathbf{x}, \mathbf{z}, \mathbf{y}) = f_1(\mathbf{x}) + f_2(\mathbf{z}) + \mathbf{y}^T(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}) + (\rho/2) \left\| \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c} \right\|_2^2 \tag{19}$$

The corresponding method of multipliers algorithm for this problem would be

$$\begin{aligned} (\mathbf{x}^{(k+1)}, \mathbf{z}^{(k+1)}) &= \underset{\mathbf{x},\mathbf{z}}{\operatorname{argmin}} \; L_\rho(\mathbf{x}, \mathbf{z}, \mathbf{y}^{(k)}) \\ \mathbf{y}^{(k+1)} &= \mathbf{y}^{(k)} + \rho(\mathbf{A}\mathbf{x}^{(k+1)} + \mathbf{B}\mathbf{z}^{(k+1)} - \mathbf{c}) \end{aligned} \tag{20}$$

The ADMM, on the other hand, does not update $\mathbf{x}$ and $\mathbf{z}$ jointly and instead uses the scheme

$$\mathbf{x}^{(k+1)} = \underset{\mathbf{x}}{\operatorname{argmin}} \; L_\rho(\mathbf{x}, \mathbf{z}^{(k)}, \mathbf{y}^{(k)}) \tag{21}$$

$$\mathbf{z}^{(k+1)} = \underset{\mathbf{z}}{\operatorname{argmin}} \; L_\rho(\mathbf{x}^{(k+1)}, \mathbf{z}, \mathbf{y}^{(k)}) \tag{22}$$

$$\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)} + \rho(\mathbf{A}\mathbf{x}^{(k+1)} + \mathbf{B}\mathbf{z}^{(k+1)} - \mathbf{c}) \tag{23}$$

As (21)–(23) is a variation of the method of multipliers, with the difference that the updating of $\mathbf{x}$ and $\mathbf{z}$ is done sequentially, the method is called Alternating Directions Method of Multipliers. In order to make the implementation of the ADMM simpler, a scaled version of (21)–(23) can be used. To this end, introduce the scaled dual variable $\mathbf{u} = \rho^{-1}\mathbf{y}$ and define the primal residual $\mathbf{r}$ as

$$\mathbf{r} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c} \tag{24}$$

From this, we have that

$$\mathbf{y}^T\mathbf{r} + (\rho/2) \left\| \mathbf{r} \right\| = (\rho/2) \left\| \mathbf{r} + \mathbf{u} \right\|_2^2 - (\rho/2) \left\| \mathbf{u} \right\|_2^2 \tag{25}$$

which allows us to re-write (21)–(23) as

$$\mathbf{x}^{(k+1)} = \underset{\mathbf{x}}{\operatorname{argmin}} \; f(\mathbf{x}) + (\rho/2) \left\| \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z}^{(k)} - \mathbf{c} + \mathbf{u}^{(k)} \right\|_2^2 \tag{26}$$

$$\mathbf{z}^{(k+1)} = \underset{\mathbf{z}}{\operatorname{argmin}} \; f_2(\mathbf{z}) + (\rho/2) \left\| \mathbf{A}\mathbf{x}^{(k+1)} + \mathbf{B}\mathbf{z} - \mathbf{c} + \mathbf{u}^{(k)} \right\|_2^2 \tag{27}$$

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \mathbf{A}\mathbf{x}^{(k+1)} + \mathbf{B}\mathbf{z}^{(k+1)} - \mathbf{c} \tag{28}$$

To ensure convergence of (26)–(28), two assumptions are needed:

- The functions $f_1 : \mathbb{R}^n \cup \{+\infty\} \mapsto \mathbb{R}$ and $f_2 : \mathbb{R}^m \cup \{+\infty\} \mapsto \mathbb{R}$ are closed, proper, and convex, and
- $\exists(\mathbf{x}^\star, \mathbf{z}^\star, \mathbf{y}^\star) : L_0(\mathbf{x}^\star, \mathbf{z}^\star, \mathbf{y}) \leq L_0(\mathbf{x}^\star, \mathbf{z}^\star, \mathbf{y}^\star) \leq L_0(\mathbf{x}, \mathbf{y}, \mathbf{y}^\star)\ \forall \mathbf{x}, \mathbf{z}, \mathbf{y}$, i.e., the unagmented Lagrangian L_0 has at least one saddle-point $(\mathbf{x}^\star, \mathbf{z}^\star, \mathbf{y}^\star)$.

Let

$$p^\star = \inf\{f_1(\mathbf{x}) + f_2(\mathbf{z}) \mid \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} = \mathbf{c}\} \tag{29}$$

denote the optimal value of (18) and define the residual of the constraint in the kth iteration of (26)–(28) as

$$\mathbf{r}^{(k)} = \mathbf{A}\mathbf{x}^{(k)} + \mathbf{B}\mathbf{z}^{(k)} - \mathbf{c} \tag{30}$$

Then, if the above stated assumptions hold, we have

- *Residual convergence*: $\lim_{k\to+\infty} \mathbf{r}^{(k)} = \mathbf{0}$, i.e., the iterates approach primal feasibility.
- *Objective convergence*: $\lim_{k\to+\infty} f_1(\mathbf{x}^{(k)}) + f_2(\mathbf{z}^{(k)}) = p^\star$, i.e., the objective function of the iterates approach the optimal value.
- *Dual variable converge*: $\lim_{k\to+\infty} \mathbf{y}^{(k)} = \mathbf{y}^\star$, i.e., the dual iterates approaches a dual optimal point.

Note that objective converges only states that the objective value converges to $p^\star$. To guarantee $\lim_{k\to+\infty} \mathbf{x}^{(k)} = \mathbf{x}^\star$ and $\lim_{k\to+\infty} \mathbf{z}^{(k)} = \mathbf{z}^\star$, additional assumptions are needed. The reader is referred to [15] for the proof of the above statement and additional details.

3 Multi-pitch estimation

3.1 Signal model

Consider a complex-valued[1] signal consisting of K pitches, where the kth pitch is constituted by a set of L_k harmonically related sinusoids, defined by the component having the lowest frequency, ω_k, according to (1), such that

$$x(t) = \sum_{k=1}^{K} \sum_{\ell=1}^{L_k} a_{k,\ell} e^{i\omega_k \ell t} \tag{31}$$

for $t = 1, \ldots, N$, where $\omega_k \ell$ is the frequency of the ℓth harmonic in the kth pitch, and with $a_{k,l}$ denoting its magnitude and phase. The occurrence of such harmonic signals is often in combination with non-sinusoidal components, such

[1] For notational simplicity and computational efficiency, we here use the discrete-time analytical signal formed from the measured (real-valued) signal.

as, for instance, colored broadband noise or non-stationary impulses. In this work, only the narrowband components of the signal are considered, although noting that audio signals often also contain other features of notable perceptual importance such as the signal's timbre. In general, selecting model orders in (31) is a daunting task, with both the number of sources, K, and the number of harmonics in each of these sources, L_k, being unknown, as well as often being structured such that different sources may have spectrally overlapping overtones. In order to remedy this, this work proposes a relaxation of the model onto a predefined grid of $P \gg K$ candidate fundamentals, each having $L_{\max} \geq \max_k L_k$ harmonics. Assume that the candidate fundamentals are chosen so numerous and so closely spaced that the approximation

$$x(t) \approx \sum_{p=1}^{P} \sum_{\ell=1}^{L_{\max}} a_{p,\ell} e^{i\omega_p \ell t} \tag{32}$$

holds. As only K pitches are present in the actual signal, we want to derive an estimator of the amplitudes $a_{p,\ell}$ such that only few, ideally $\sum_{k=1}^{K} L_k$, dictionary elements are non-zero. This approach may be seen as a sparse linear regression problem reminiscent of the one in [16] and has been thoroughly examined in the context of pitch estimation in, e.g., [6, 17, 18]. For notational convenience, define the set of all amplitude parameters to be estimated as

$$\mathbf{\Psi} = \left\{ \mathbf{\Psi}_{\omega_1}, \ldots, \mathbf{\Psi}_{\omega_P} \right\} \tag{33}$$

$$\mathbf{\Psi}_{\omega_k} = \left\{ a_{k,1}, \ldots, a_{k,L_{\max}} \right\} \tag{34}$$

where, as described above, most of the $a_{k,\ell}$ in $\mathbf{\Psi}$ will be zero. Note that the structure of $\mathbf{\Psi}$ will be sparse, i.e., having few non-zero elements. Also, the pattern of this sparsity will be group wise, meaning that if a pitch with fundamental frequency ω_p is not present, then neither will any of its harmonics, i.e., $\mathbf{\Psi}_{\omega_p} = \mathbf{0}$. Due to the harmonic structure of the signal, candidate pitches having fundamental frequencies at fractions of the present pitches fundamentals will have a partial fit of their harmonics. This may cause misclassification, i.e., erroneously identifying a present pitch as one or more non-present candidate pitches. This is the cause of the so-called halfling problem, which is mistaking the true pitch with fundamental frequency ω_p for the candidate pitch with fundamental frequency $\omega_p/2$. This may occur if the candidate set $\mathbf{\Psi}$ is structured such that the halfling pitch may perfectly model the true pitch, which is when $L_{\max} \geq 2L_p$. This is illustrated in Figure 1, displaying a pitch with fundamental frequency 100 Hz and four harmonics and as well as its halfling, i.e., a pitch with fundamental frequency 50 Hz and eight harmonics where only the even-numbered harmonics are non-zero. Relating to music signals, this is the same as mistaking a pitch for the pitch an octave below it. Thus, when estimating the elements of $\mathbf{\Psi}$, one also has to take into account of some structure of the block sparsity in order to avoid erroneously selecting halflings. A method for doing this will be presented in the following section.

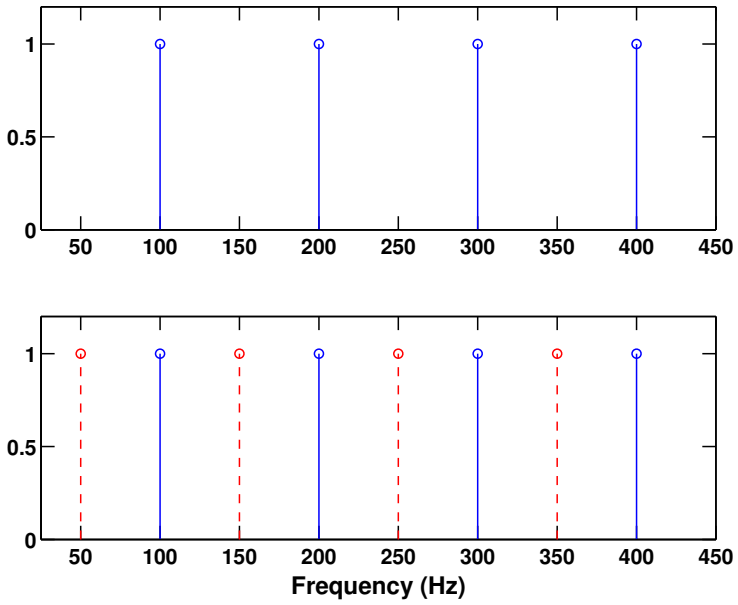

Figure 1: The upper picture depicts a pitch with fundamental frequency 100 Hz and four harmonics. The lower picture depicts a pitch with fundamental frequency 50 Hz and eight harmonics where all odd-numbered harmonics are zero (marked with dashed red).

3.2 Proposed estimation algorithm

Assume that we have measured a time frame of the signal in (31), for $t = 1, \ldots, N$, and that the observations are corrupted by an additive broadband noise, $e(t)$, such that our measurements are well modeled as $y(t) = x(t) + e(t)$. A straightforward approach to estimate $\mathbf{\Psi}$ would then be to minimize the residual cost function

$$g_1(\mathbf{\Psi}) = \frac{1}{2} \sum_{t=1}^{N} \left| y(t) - \sum_{p=1}^{P} \sum_{\ell=1}^{L_{\max}} a_{p,\ell} e^{i\omega_p \ell t} \right|^2 \tag{35}$$

However, setting

$$\hat{\mathbf{\Psi}} = \underset{\mathbf{\Psi}}{\operatorname{argmin}}\ g_1(\mathbf{\Psi}) \tag{36}$$

will not yield the desired sparsity structure of $\mathbf{\Psi}$ and will be prone to also model the noise $e(t)$. A solution to this would be to add terms penalizing solutions $\hat{\mathbf{\Psi}}$ that are not sparse, for example as

$$\hat{\mathbf{\Psi}} = \underset{\mathbf{\Psi}}{\operatorname{argmin}}\ g_1(\mathbf{\Psi}) + \lambda ||\mathbf{\Psi}||_0 \tag{37}$$

where $||\mathbf{\Psi}||_0$ is the pseudo-norm counting the number of non-zero elements in $\mathbf{\Psi}$ and λ is a regularization parameter. However, this in general leads to a combinatorial problem that is NP-hard to solve. To avoid this, one can approximate the ℓ_0 penalty by the convex function

$$g_2(\mathbf{\Psi}) = \sum_{p=1}^{P} \sum_{\ell=1}^{L_{\max}} |a_{p,\ell}| \tag{38}$$

The resulting problem

$$\min_{\mathbf{\Psi}} \; g_1(\mathbf{\Psi}) + \lambda g_2(\mathbf{\Psi}) \tag{39}$$

is known as the LASSO [19]. In fact, it can be shown that under some restrictions on $\mathbf{\Psi}$, (see also [20]), the LASSO is guaranteed to retrieve the non-zero indices of $\mathbf{\Psi}$ with high probability, although these conditions are not assumed to be met here. To encourage the group-sparse behavior of $\hat{\mathbf{\Psi}}$, one can further introduce

$$g_3(\mathbf{\Psi}) = \sum_{p=1}^{P} \sqrt{\sum_{\ell=1}^{L_{\max}} |a_{p,\ell}|^2} \tag{40}$$

which is also a convex function. The inner sum corresponds to the ℓ_2-norm, and does not enforce sparsity within each pitch, whereas instead the outer sum, corresponding to the ℓ_1-norm, enforces sparsity between pitches. Thereby, adding the $g_3(\mathbf{\Psi})$ constraint will penalize the number of non-zero pitches. However, if we for some p have $2L_p \leq L_{\max}$, the above penalties have no way of discriminating between the correct pitch candidate ω_p and the spurious halfling candidate $\omega_p/2$. However, as the candidates will differ in that the halfling will only contribute to the harmonic signal at every other frequency in the block, as was seen in Figure 1, one may reduce the risk of such a misclassification by adding the further penalty

$$\breve{g}_4(\mathbf{\Psi}) = \sum_{q=2}^{PL_{\max}} \Big| |a_q| - |a_{q-1}| \Big| \tag{41}$$

where the reparametrization is $q = (p-1)L_{\max} + \ell$, which would add a cost to blocks where there are notable magnitude variations between neighboring harmonics. Unfortunately, (41) is not convex, but a simple convex approximation would be $\tilde{g}_4$, detailed as

$$\tilde{g}_4(\mathbf{\Psi}) = \sum_{q=2}^{PL_{\max}} \big| a_q - a_{q-1} \big| \tag{42}$$

which would be a good approximation of (41) if all the harmonics had the same phase. Clearly, this may not be the case, resulting in that the penalty in

(42) would also penalize the correct candidate. An illustration of this is found by considering the worst-case scenario, when all the adjacent harmonics are completely out of phase and have the same magnitudes, i.e., $a_{p,\ell+1} = a_{p,\ell}e^{i\pi}$ with magnitude $|a_{p,\ell}| = r$, for $\ell = 1, \ldots, L_p - 1$. Then, the penalty in (42) will yield a cost of $\tilde{g}_4(\mathbf{\Psi}_{\omega_p}) = 2rL_p$ rather than the desired $\breve{g}_4(\mathbf{\Psi}_{\omega_p}) = 2r$. The cost may also be compared with that of (38), which is $g_2(\mathbf{\Psi}_{\omega_p}) = rL_p$, suggesting that this would add a relatively large penalty. More interestingly, for the halfling candidate pitch, the cost will be just as large, i.e., if $\omega_{p'} = \omega_p/2$, then $\tilde{g}_4(\mathbf{\Psi}_{\omega_{p'}}) = 2rL_p$ provided that $L_{\max} \geq 2L_p$, thereby offering no possibility of discriminating between the true pitch and its halfling. Obviously, such a worst case scenario is just as unlikely as all harmonics having the same phase, if assuming that the phases are uniformly distributed on $[0, 2\pi)$. Instead, the $\tilde{g}_4$ penalty of the true pitch will be slightly smaller than its halfling counterpart, on average, and together with (40), the scales tip in favour of the true pitch, as shown in [6]. One may thus conclude that the combination of g_3 and $\tilde{g}_4$ provides a block sparse solution where halflings are usually discouraged. However, it should be noted that such a solution requires the tuning of two functions to control the block sparsity.

This work proposes to simplify the described estimator by improving the approximation in (42), by using an adaptive penalty approach. In order to do so, let $\varphi_{k,\ell}$ denote the phase of the component with frequency $\omega_{k,\ell}$ and collect these phases in the parameter set

$$\mathbf{\Phi} = \left\{\mathbf{\Phi}_{\omega_1}, \ldots, \mathbf{\Phi}_{\omega_P}\right\} \tag{43}$$

$$\mathbf{\Phi}_{\omega_k} = \left\{\varphi_{k,1}, \ldots, \varphi_{k,L_{\max}}\right\} \tag{44}$$

The penalty function in (42) may then be modified to

$$g_4(\mathbf{\Psi}, \mathbf{\Phi}) = \sum_{q=1}^{PL_{\max}-1} \left|a_{q+1}e^{-\varphi_{q+1}} - a_q e^{-\varphi_q}\right| \tag{45}$$

thus penalizing only differences in magnitude. In order to do so, the phases $\varphi_{k,\ell}$ need to be estimated as the arguments of the latest available amplitude estimates $a_{k,\ell}$. As a result, (45) yields an improved approximation of (41), avoiding the issues of (42) described above, and also promotes a block sparse solution. The block sparsity is promoted due to the reparametrization of the amplitude indices: as the dictionary resolution of the dictionary is high, we do not expect adjacent candidate pitches to be present in the signal. In effect, this introduces a penalty for activating a pitch block. As a result, the block-norm penalty function g_3 may be omitted, which simplifies the algorithm noticeably. Thus, we form the parameter estimates by solving

$$\hat{\mathbf{\Psi}} = \arg\min_{\mathbf{\Psi}} \; g_1(\mathbf{\Psi}) + \lambda_2 g_2(\mathbf{\Psi}) + \lambda_4 g_4(\mathbf{\Psi}, \mathbf{\Phi}) \tag{46}$$

where λ_2 and λ_4 are user-defined regularization parameters that weigh the importance of each penalty function with that of the residual cost. To form the

convex criteria and to facilitate the implementation, consider the signal expressed in matrix notation as

$$\mathbf{y} = \left[\begin{array}{cccc} y(1) & ... & y(N) \end{array} \right]^T \tag{47}$$

$$= \sum_{p=1}^{P} \mathbf{W}_p \, \mathbf{a}_p + \mathbf{e} \triangleq \mathbf{W}\mathbf{a} + \mathbf{e} \tag{48}$$

where

$$\mathbf{W} = \left[\begin{array}{ccc} \mathbf{W}_1 & \dots & \mathbf{W}_P \end{array} \right] \tag{49}$$

$$\mathbf{W}_p = \left[\begin{array}{ccc} \mathbf{z}^1 & \dots & \mathbf{z}^{L_{\max}} \end{array} \right] \tag{50}$$

$$\mathbf{z}_p = \left[\begin{array}{ccc} e^{i\omega_p 1} & \dots & e^{i\omega_p N} \end{array} \right]^T \tag{51}$$

$$\mathbf{a} = \left[\begin{array}{ccc} \mathbf{a}_1^T & \dots & \mathbf{a}_P^T \end{array} \right]^T \tag{52}$$

$$\mathbf{a}_p = \left[\begin{array}{ccc} a_{p,1} & \dots & a_{p,L_{\max}} \end{array} \right]^T \tag{53}$$

The dictionary matrix $\mathbf{W}$ is constructed of P horizontally stacked blocks, or dictionary atoms $\mathbf{W}_p$, where each is a matrix with $L_{\max}$ columns and N rows. In order to obtain an acceptable approximation of (41), the problem must be solved iteratively, where the last solution is used to improve the next. To pursue an even sparser solution, a re-weighting procedure is simultaneously used for $g_2(\mathbf{\Psi})$, similar to the one used in [21]. The solution is thus found at the k-th iteration by solving

$$\hat{\mathbf{a}}^{(k)} = \arg\min_{\mathbf{a}} \sum_{j=1,2,4} g_j(\mathbf{H}_j^{(k)}\mathbf{a}, \lambda_j) \tag{54}$$

where

$$\mathbf{H}_1^{(k)} = \mathbf{W} \tag{55}$$

$$\mathbf{H}_2^{(k)} = \mathrm{diag}\big(1/(\left\|\hat{\mathbf{a}}^{(k-1)}\right\|_1 + \epsilon)\big) \tag{56}$$

$$\mathbf{H}_4^{(k)} = \mathbf{F}\,\mathrm{diag}(\arg\left(\hat{\mathbf{a}}^{(k-1)}\right))^{-1} \tag{57}$$

and with

$$g_1(\mathbf{H}_1^{(k)}\mathbf{a}, 1) = \frac{1}{2}\left|\left|\mathbf{y} - \mathbf{H}_1^{(k)}\mathbf{a}\right|\right|_2^2 \tag{58}$$

$$g_2(\mathbf{H}_2^{(k)}\mathbf{a}, \lambda_2) = \lambda_2 \left|\left|\mathbf{H}_2^{(k)}\mathbf{a}\right|\right|_1 \tag{59}$$

$$g_4(\mathbf{H}_4^{(k)}\mathbf{a}, \lambda_4) = \lambda_4 \left|\left|\mathbf{H}_4^{(k)}\mathbf{a}\right|\right|_1 \tag{60}$$

where $\mathrm{diag}(\cdot)$ denotes a diagonal matrix formed with the given vector along its diagonal, $\arg(\cdot)$ is the element-wise complex argument, and $\epsilon \ll 1$. Also, $\mathbf{I}$ denotes the identity matrix, and $\mathbf{F}$ is a first order difference matrix, having elements $\mathbf{F}\{n,n\} = 1$, $\mathbf{F}\{n,n+1\} = -1$, for $n = 1, \dots, PL_{\max}-1$, and zeros everywhere else. As intended, the minimization in (54) is convex, and may be solved

using one of many convex solvers publicly available, such as, for instance, the interior point methods SeDuMi [14] or SDPT3 [15]. However, as mentioned earlier, these methods are quite computationally burdensome and will scale poorly with increased data length and larger grids. Instead, we here propose an efficient implementation using ADMM. The problem in (54) may be implemented in a similar manner as was done in [22], thus requiring only two tuning parameters, λ_2 and λ_4. The proposed method compares to the PEBS and PEBS-TV algorithms introduced in [6] as improving upon the former, and requiring fewer tuning parameters than the latter. The proposed method is therefore termed a light and improved version of PEBS, here denoted the PEBSI-Lite algorithm.

3.3 ADMM implementation

In order to solve (54), an ADMM implementation is needed. Therefore, (54) has to be written on the form (18). To this end, introduce the auxiliary variables $\mathbf{z} \in \mathbb{C}^{PL_{\max}}$, $\mathbf{u}_1 \in \mathbb{C}^N$, $\mathbf{u}_2 \in \mathbb{C}^{PL_{\max}}$, and $\mathbf{u}_4 \in \mathbb{C}^{PL_{\max}-1}$ and let

$$\mathbf{G}^{(k)} = \begin{bmatrix} \mathbf{H}_1^{(k)T} & \mathbf{H}_2^{(k)T} & \mathbf{H}_4^{(k)T} \end{bmatrix}^T \tag{61}$$

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}_1^T & \mathbf{u}_2^T & \mathbf{u}_4^T \end{bmatrix}^T \tag{62}$$

Note that the earlier ADMM results which were derived for real variables also hold in the complex case as complex numbers can be represented as real vectors. Thus, we want to solve

$$\min_{\mathbf{z}} f(\mathbf{G}^{(k)}\mathbf{z}) \tag{63}$$

where

$$f(\mathbf{G}^{(k)}\mathbf{z}) = \frac{1}{2}\left\|\mathbf{y} - \mathbf{H}_1^{(k)}\mathbf{z}\right\|_2^2 + \lambda_2\left\|\mathbf{H}_2^{(k)}\mathbf{z}\right\|_1 + \lambda_4\left\|\mathbf{H}_4^{(k)}\mathbf{z}\right\|_1 \tag{64}$$

Using the auxiliary variabel $\mathbf{u}$, one may equivalently solve

$$\begin{aligned} &\min_{\mathbf{z},\mathbf{u}} f(\mathbf{u}) + \frac{\mu}{2}\left\|\mathbf{G}^{(k)}\mathbf{z} - \mathbf{u}\right\|_2^2 \\ &\text{subject to } \mathbf{G}^{(k)}\mathbf{z} - \mathbf{u} = \mathbf{0} \end{aligned} \tag{65}$$

where μ is a positive scalar, as the added term is zero for any feasible point. Introducing the (scaled) dual variable

$$\mathbf{d} = \begin{bmatrix} \mathbf{d}_1^T & \mathbf{d}_2^T & \mathbf{d}_4^T \end{bmatrix}^T \tag{66}$$

where $\mathbf{d}_1 \in \mathbb{C}^N$, $\mathbf{d}_2 \in \mathbb{C}^{PL_{\max}}$, and $\mathbf{d}_4 \in \mathbb{C}^{PL_{\max}-1}$, the Lagrangian of the problem is

$$L_\mu(\mathbf{z},\mathbf{u},\mathbf{d}) = f(\mathbf{u}) + \frac{\mu}{2}\left\|\mathbf{G}^{(k)}\mathbf{z} - \mathbf{u} - \mathbf{d}\right\|_2^2 - \frac{\mu}{2}\|\mathbf{d}\|_2^2 \tag{67}$$

Algorithm 1 The proposed PEBSI-Lite algorithm

initiate $k := 0$, $\mathbf{H}_4^{(0)} = \mathbf{F}$, and
  $\mathbf{a}^{(0)} = \mathbf{z}_{\text{save}} = \mathbf{d}_{\text{save}} = \mathbf{0}^{PL_{\max}\times 1}$
repeat {adaptive penalty scheme}
  initiate $\ell := 0$, $\mathbf{u}_2(0) = \mathbf{a}^{(k)}$,
    $\mathbf{z}(0) = \mathbf{z}_{\text{save}}$, and $\mathbf{d}(0) = \mathbf{d}_{\text{save}}$
  repeat {ADMM scheme}
    $\mathbf{z}(\ell) = \left(\mathbf{G}^{(k)H}\mathbf{G}^{(k)}\right)^{-1}\mathbf{G}^{(k)H}\left(\mathbf{u}(\ell)+\mathbf{d}(\ell)\right)$
    $\mathbf{u}_1(\ell+1) = \frac{\mathbf{y}-\mu\left(\mathbf{H}_1^{(k)}\mathbf{z}(\ell+1)-\mathbf{d}_1(\ell)\right)}{1+\mu}$
    $\mathbf{u}_2(\ell+1) = \mathbf{T}\left(\mathbf{H}_2^{(k)}\mathbf{z}(\ell+1)-\mathbf{d}_2(\ell), \frac{\lambda_2}{\mu}\right)$
    $\mathbf{u}_4(\ell+1) = \mathbf{T}\left(\mathbf{H}_4^{(k)}\mathbf{z}(\ell+1)-\mathbf{d}_4(\ell), \frac{\lambda_4}{\mu}\right)$
    $\mathbf{d}(\ell+1) = \mathbf{d}(\ell) - \left(\mathbf{G}^{(k)}\mathbf{z}(\ell+1)-\mathbf{u}(\ell+1)\right)$
    $\ell \leftarrow \ell+1$
  until convergence
  store $\mathbf{a}^{(k)} = \mathbf{u}_2(\text{end}), \mathbf{z}_{\text{save}} = \mathbf{z}(\text{end})$, and $\mathbf{d}_{\text{save}} = \mathbf{d}(\text{end})$
  update $\mathbf{H}_4^{(k+1)} = \mathbf{F}\text{diag}\left(\arg\left(\mathbf{a}^{(k)}\right)\right)^{-1}$
  $k \leftarrow k+1$
until convergence

The Lagrangian (67) is separable in the variables $\mathbf{z}$, $\mathbf{u}_1$, $\mathbf{u}_2$, and $\mathbf{u}_4$ and one may thus form an updating scheme similar to (26)–(28) as

$$\mathbf{z}(j+1) = \underset{\mathbf{z}}{\text{argmin}} \ \left\|\mathbf{G}^{(k)}\mathbf{z}-\mathbf{u}(j)-\mathbf{d}(j)\right\|_2^2 \tag{68}$$

$$\mathbf{u}_1(j+1) = \underset{\mathbf{u}_1}{\text{argmin}} \ \frac{1}{2}\left\|\mathbf{y}-\mathbf{u}_1\right\|_2^2 + \frac{\mu}{2}\left\|\mathbf{H}_1^{(k)}\mathbf{z}(j+1)-\mathbf{u}_1-\mathbf{d}_1(j)\right\|_2^2 \tag{69}$$

$$\mathbf{u}_2(j+1) = \underset{\mathbf{u}_2}{\text{argmin}} \ \lambda_2\left\|\mathbf{u}_2\right\|_1 + \frac{\mu}{2}\left\|\mathbf{H}_2^{(k)}\mathbf{z}(j+1)-\mathbf{u}_2-\mathbf{d}_2(j)\right\|_2^2 \tag{70}$$

$$\mathbf{u}_4(j+1) = \underset{\mathbf{u}_4}{\text{argmin}} \ \lambda_4\left\|\mathbf{u}_4\right\|_1 + \frac{\mu}{2}\left\|\mathbf{H}_4^{(k)}\mathbf{z}(j+1)-\mathbf{u}_4-\mathbf{d}_4(j)\right\|_2^2 \tag{71}$$

$$\mathbf{d}(j+1) = \mathbf{d}(j) - \left(\mathbf{G}^{(k)}\mathbf{z}(j+1)-\mathbf{u}(j+1)\right) \tag{72}$$

The updates of $\mathbf{z}$ and $\mathbf{u}$ are given by

$$\mathbf{z}(j+1) = \underset{\mathbf{z}}{\text{argmin}} \ \left\|\mathbf{G}^{(k)}\mathbf{z}-\mathbf{u}(j)-\mathbf{d}(j)\right\|_2^2 \tag{73}$$

$$= \left(\mathbf{G}^{(k)H}\mathbf{G}^{(k)}\right)^{-1}\mathbf{G}^{(k)H}\left(\mathbf{u}(j)+\mathbf{d}(j)\right) \tag{74}$$

and

$$\mathbf{u}_1(j+1) = \underset{\mathbf{u}_1}{\operatorname{argmin}} \frac{1}{2}\left\|\mathbf{y}-\mathbf{u}_1\right\|_2^2 + \frac{\mu}{2}\left\|\mathbf{H}_1^{(k)}\mathbf{z}(j+1)-\mathbf{u}_1-\mathbf{d}_1(j)\right\|_2^2 \tag{75}$$

$$= \frac{\mathbf{y}-\mu\big(\mathbf{H}_1^{(k)}\mathbf{z}(j+1)-\mathbf{d}_1(j)\big)}{1+\mu} \tag{76}$$

respectively. Finally, using the element-wise shrinkage function from [6],

$$\mathbf{T}\left(\mathbf{x},\xi\right) = \frac{\max(|\mathbf{x}|-\xi,0)}{\max(|\mathbf{x}|-\xi,0)+\xi}\odot\mathbf{x} \tag{77}$$

one may update $\mathbf{u}_2$ and $\mathbf{u}_4$ as

$$\mathbf{u}_2(j+1) = \underset{\mathbf{u}_2}{\operatorname{argmin}}\ \lambda_2\left\|\mathbf{u}_2\right\|_1 + \frac{\mu}{2}\left\|\mathbf{H}_2^{(k)}\mathbf{z}(j+1)-\mathbf{u}_2-\mathbf{d}_2(j)\right\|_2^2 \tag{78}$$

$$= \mathbf{T}\left(\mathbf{H}_2^{(k)}\mathbf{z}(j+1)-\mathbf{d}_2(j),\frac{\lambda_2}{\mu}\right) \tag{79}$$

and

$$\mathbf{u}_4(j+1) = \underset{\mathbf{u}_4}{\operatorname{argmin}}\ \lambda_4\left\|\mathbf{u}_4\right\|_1 + \frac{\mu}{2}\left\|\mathbf{H}_4^{(k)}\mathbf{z}(j+1)-\mathbf{u}_4-\mathbf{d}_4(j)\right\|_2^2 \tag{80}$$

$$= \mathbf{T}\left(\mathbf{H}_4^{(k)}\mathbf{z}(j+1)-\mathbf{d}_4(j),\frac{\lambda_4}{\mu}\right) \tag{81}$$

respectively. Having this in place, the full algorithm is presented in Algorithm 1, where the solution is given as $\hat{\mathbf{a}} = \mathbf{a}^{(k_{\text{end}})}$ where k_{end} is the last iteration index of the outer loop.

3.4 Numerical results

In order to examine the performance of the proposed algorithm, it was evaluated using a simulated dual-pitch signal, measured in white Gaussian noise at different Signal-to-Noise Ratios (SNRs), ranging from −5 dB to 20 dB in steps of 5 dB. The SNR is here defined as

$$\text{SNR} = 10\log_{10}\frac{\sigma_x^2}{\sigma_e^2} \tag{82}$$

where σ_x^2 and σ_e^2 is the variance of the signal and the noise, respectively. For a pitch signal generated by (31), under the simplifying assumption of distinct sinusoidal components, the variance of the signal is given by

$$\sigma_x^2 = \sum_{k=1}^{K}\sum_{\ell=1}^{L_k}\frac{|a_{k,\ell}|^2}{2} \tag{83}$$

At each SNR, 200 Monte Carlo simulations were performed, each simulation generating a signal with fundamental frequencies of 600 and 700 Hz. To reflect

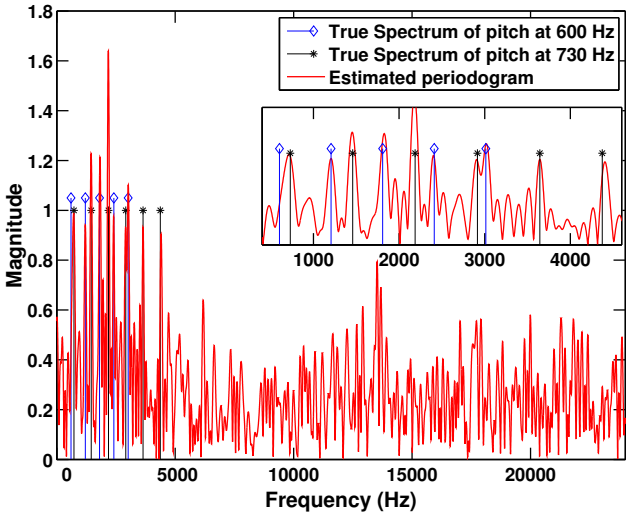

Figure 2: The periodogram estimate and the true signal studied in Figure 3.

the performance in presence of off-grid effects, the fundamental frequencies were randomly chosen at each simulation uniformly on $600 \pm d/2$ and $700 \pm d/2$, where d is the grid point spacing. The phases of the harmonics in each pitch were chosen uniformly on $[0, 2\pi)$, whereas all had unit magnitude. The signal was sampled at $f_s = 48$ kHz on a time frame of 10 ms, yielding $N = 480$ samples per frame. As a result, the pitches were spaced by approximately f_s/N Hz, which is the resolution limit of the periodogram. This is also seen in Figure 2, illustrating the resolution of the periodogram as well as the frequencies of the harmonics, at SNR $= -5$ dB. From the figure, it may be concluded that the signal contains more than one harmonic source, as the observed peaks are not harmonically related. Furthermore, it is clear that the fundamental frequencies are not separated by the periodogram, indicating that any pitch estimation algorithm based on the periodogram would suffer notable difficulties. In order to form the estimates, the estimation procedure began by using a coarse dictionary, with candidate pitches uniformly distributed on the interval $[280, 1500]$ Hz, thus also including $\omega_p/2$ and $2\omega_p$ for both pitches. The coarse resolution was $d = 10$ Hz, i.e., still a super-resolution of $f_s/10N$. After estimation on this grid, a zooming step was taken where a new grid with spacing $d/10$ was laid $\pm 2d$ around each pitch having non-zero power. This zooming approach was taken for the proposed method, as well as for PEBS and PEBS-TV. The regularization parameter values used for PEBSI-Lite, PEBS-TV, and PEBS are presented in Tables 1, 2, and 3 respectively. The values where selected using manual cross-validation for similar signals. Comparisons were also made with the ANLS, ORTH, and the harmonic Capon estimators, which had been given the oracle model orders (see [5] for more details on these methods). The simulation and

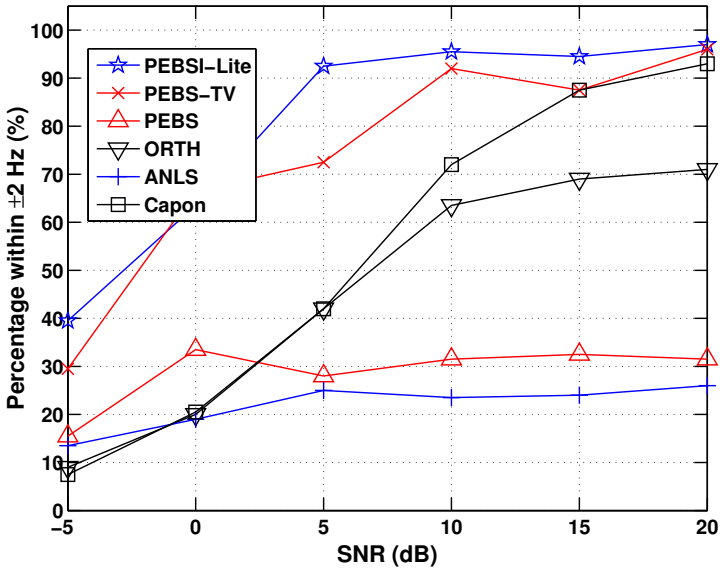

Figure 3: Percentage of estimated pitches where both fundamental frequencies lie at most 2 Hz, or $d/5 = 1/50N$, from the ground truth, plotted as a function of SNR. Here, the pitches have $[5, 6]$ harmonics, respectively, and $L_{\max} = 10$.

estimation procedure was performed for two cases; one where the number of harmonics L_k were set to 5 and 6, and one where L_k were set to 10 and 11. In the former case, $L_{\max} = 10$ and in the latter $L_{\max} = 20$, i.e., well above the true number of harmonics. Figures 3 and 4 show the percentage of pitch estimates where both lie within ± 2 Hz from the true values for the six compared methods, for the case of 5 and 6 as well as 10 and 11 harmonics, respectively. As is clear from the figures, the proposed method performs as well, or better, than the PEBS-TV algorithm, although requiring fewer tuning parameters. In this setting, PEBS performs poorly, as the generous choices of $L_{\max}$ allows it to pick the halfling, as predicted.

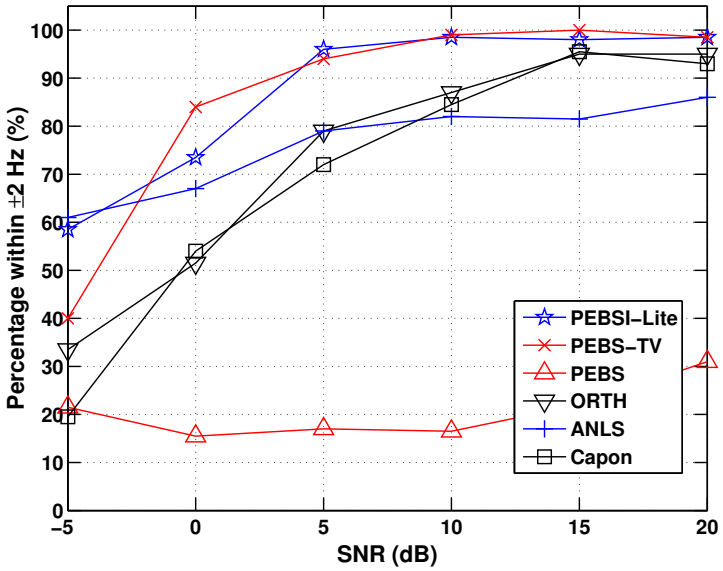

Figure 4: Percentage of estimated pitches where both fundamental frequencies lie at most 2 Hz, or $d/5 = 1/50N$, from the ground truth, plotted as a function of SNR. Here, the pitches have $[10, 11]$ harmonics, respectively, and $L_{\max} = 20$.

SNR (dB)	-5	0	5	10	15	20
λ_2	0.4	0.4	0.2	0.2	0.2	0.2
λ_4	0.4	0.4	0.2	0.2	0.2	0.2

Table 1: Regularization parameter values for PEBSI-Lite.

SNR (dB)	-5	0	5	10	15	20
λ_2	0.2	0.2	0.2	0.15	0.1	0.1
λ_3	0.3	0.3	0.3	0.2	0.2	0.15
λ_4	0.1	0.1	0.1	0.75	0.75	0.05

Table 2: Regularization parameter values for PEBS-TV.

SNR (dB)	-5	0	5	10	15	20
λ_2	0.2	0.2	0.2	0.15	0.15	0.1
λ_4	0.4	0.4	0.4	0.3	0.3	0.2

Table 3: Regularization parameter values for PEBS.

4 Choosing the regularization parameters

The pitch estimates produced by PEBSI-Lite in the preceding section were highly dependent on the values of the regularization parameters λ_2 and λ_4, which had to be hand-tuned to produce good results. In general, large values of λ_2 encourage sparse solutions while large values of λ_4 encourage solutions that are smooth within blocks. As the model order is unknown, it is generally hard to determine how sparse the solution should be in order to be considered the desired one. Therefore, one often determines the values of the regularization parameters using cross-validation schemes, making the performance of the methods user dependent. Thus, one would like to have a systematic and preferable automatic method for choosing λ_2 and λ_4, and thereby the model order. A common approach to solving model order problems is to use information criteria such as AIC or BIC [23], which measure the fit of the model to the data, while penalizing high model orders, resulting in a trade-off criterion that should take its optimal (minimal for AIC and BIC) for the correct model order. For the LASSO problem, there have been suggestions of appropriate model order criteria [24], [25]. In [6], the authors suggest a BIC-style criterion for multi-pitch estimation. However, this criterion can only be applied to a single estimate, i.e., one PEBS-TV solution, to determine which of the found pitches are true and which are spurious. Thus, it cannot be used to choose between different estimates. To the author's knowledge, there are no good model order selection rules available that are applicable to PEBSI-Lite.

Also, even if one has an efficient criterion for choosing between different models, one first has to form a set of candidate models, in effect running Algorithm 1 for different values of λ_2 and λ_4. As the set of possible choices is $\{(\lambda_2, \lambda_4) | (\lambda_2, \lambda_4) \in \mathbf{R}_+ \times \mathbf{R}_+\}$ one also needs a strategy for choosing a smaller set of (λ_2, λ_4) candidates. Ideally, one would like to only fit one model per sparsity level, with sparsity meaning either the number of activated blocks, i.e., pitches, or elements, i.e., sinusoidal components. For each sparsity level one would like to fit a model having the least biased estimates of the sinusoidal amplitudes. This means that one, for a given sparsity level, would like to find the smallest pair (λ_2, λ_4) resulting in that sparsity level.

Figure 5 shows a plot of the number of pitches present in the solution when applying PEBSI-Lite to a three pitch signal for a grid of parameter values (λ_2, λ_4). The number of harmonics of each pitch is 4, 5, and 4 respectively, resulting in a total of 13 sinusoidal components. In the figure, ridges on the solution surface where the number of present pitches changes can be seen. To find our set of candidate models, we would therefore like to find these ridges without having to solve PEBSI-Lite for the whole plane of regularization parameter values. In an attempt to understand how to construct such a path algorithm, the next section presents a variation of an algorithm published in [25] for the simpler LASSO case.

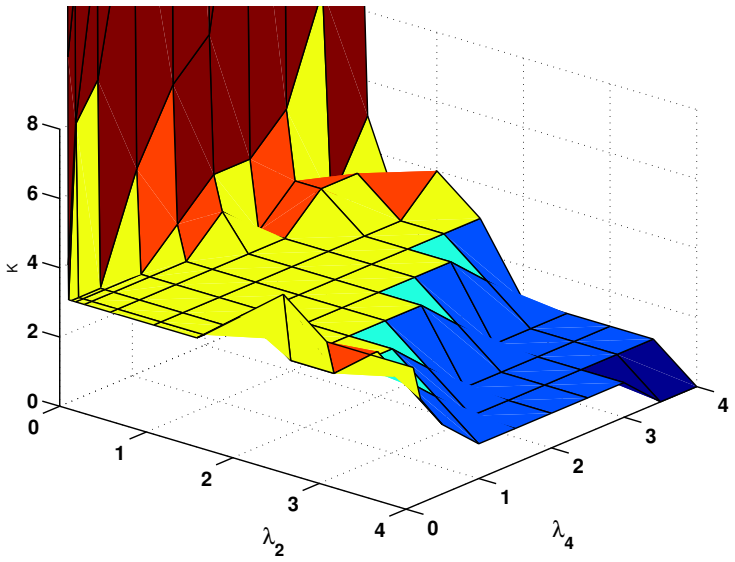

Figure 5: The number of pitches present, (K), in solutions obtained when applying the PEBSI-Lite algorithm to a three pitch signal with a total of 13 sinusoidal components for varying values of (λ_2, λ_4).

4.1 Candidate model selection for the LASSO

The LASSO problem is often stated as

$$\min_{\mathbf{a}} ||\mathbf{y} - \mathbf{W}\mathbf{a}||_2^2 + \lambda||\mathbf{a}||_1 \tag{84}$$

which, as noted above, is a relaxation of the, in general, NP-hard problem

$$\min_{\mathbf{a}} ||\mathbf{y} - \mathbf{W}\mathbf{a}||_2^2 + \lambda||\mathbf{a}||_0 \tag{85}$$

where $\mathbf{y}$ is an N-vector, $\mathbf{a}$ is an P-vector, and $\mathbf{W}$ is an $N \times P$ matrix. Often, $\mathbf{W}$ is an over-complete dictionary with $P \gg N$. As mentioned earlier, the term $\lambda||\mathbf{a}||_1$ acts as an approximation of the non-convex penalty $\lambda||\mathbf{a}||_0$ and induces sparsity on the solution vector $\mathbf{a}$. If $||\mathbf{a}||_0 = K$, $\mathbf{a}$ is said to be K-sparse. If the true model order is unknown, one might be interested in solving the problem for values of λ inducing different sparsity levels K and then apply some model order selection criterion in order to choose the correct model order. There exists algorithms that solve (84) for all values $\lambda \in [0, +\infty)$ in the case of real variables and matrices, the probably most well-known method being called LARS [26]. However, instead of solving (84) for regularization parameters along the positive real line, one might be interested, perhaps for computational reasons, to solve (84) for only one value of λ per sparsity level K. In [25], a method for finding

these values λ was proposed for the slightly different problem

$$\min_{\mathbf{a}} ||\mathbf{y} - \mathbf{W}\mathbf{a}||_2 + \lambda||\mathbf{a}||_1 \tag{86}$$

The authors introduce the term singular point to denote values λ for which a slight increase in λ changes the sparsity of the solution to (86). This means that the k-th singular point $\lambda^{(k)}$ is defined as

$$\lambda^{(k)} = \max\{\lambda \mid \|\mathbf{a}(\lambda)\|_0 = k\} \tag{87}$$

where $\mathbf{a}(\lambda)$ is the solution to (86). As the problems (84) and (86) are not the same, we here present some modifications of the results given in [25] as to fit (84). Assume that (84) has been solved for a value of λ and denote the solution $\mathbf{a}(\lambda)$. In order to derive a condition for $\mathbf{a}(\lambda)$ to be a solution to (84), we consider the real valued counterpart of (84). Let $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_P]$ be the column representation of $\mathbf{W}$ and introduce $\tilde{\mathbf{a}}(\lambda)$, $\tilde{\mathbf{y}}$, $\tilde{\mathbf{w}}_j$, and $\tilde{\mathbf{W}}$ as

$$\tilde{\mathbf{a}}(\lambda) = \begin{bmatrix} \mathfrak{Re}\{\mathbf{a}(\lambda)\}^T & \mathfrak{Im}\{\mathbf{a}(\lambda)\}^T \end{bmatrix}^T \tag{88}$$

$$\tilde{\mathbf{y}} = \begin{bmatrix} \mathfrak{Re}\{\mathbf{y}\}^T & \mathfrak{Im}\{\mathbf{y}\}^T \end{bmatrix}^T \tag{89}$$

$$\tilde{\mathbf{w}}_j = \begin{bmatrix} \mathfrak{Re}\{\mathbf{w}_j\} & -\mathfrak{Im}\{\mathbf{w}_j\} \\ \mathfrak{Im}\{\mathbf{w}_j\} & \mathfrak{Re}\{\mathbf{w}_j\} \end{bmatrix} \tag{90}$$

$$\tilde{\mathbf{W}} = \begin{bmatrix} \mathfrak{Re}\{\mathbf{W}\} & -\mathfrak{Im}\{\mathbf{W}\} \\ \mathfrak{Im}\{\mathbf{W}\} & \mathfrak{Re}\{\mathbf{W}\} \end{bmatrix} \tag{91}$$

where $\mathfrak{Re}\{\cdot\}$ and $\mathfrak{Im}\{\cdot\}$ denote the real and imaginary parts of the vectors and matrices, respectively. Using this notation, we can reformulate (84) as

$$\min_{\tilde{\mathbf{a}}} \frac{1}{2} \left\| \tilde{\mathbf{y}} - \tilde{\mathbf{W}}\tilde{\mathbf{a}} \right\|_2^2 + \lambda \sum_{j=1}^{P} \left\| [\tilde{a}_j, \tilde{a}_{j+P}]^T \right\|_2 \tag{92}$$

where $[\tilde{a}_j, \tilde{a}_{j+P}]^T$ is the column vector representation of the complex number a_j. Thus, we have effectively transformed (84) to an equivalent group-LASSO problem in real variables with P groups. Further, $\mathbf{a}(\lambda)$ is a solution to (84) if its real counterpart $\tilde{\mathbf{a}}(\lambda)$ solves (92). The objective function (92) is convex which means that $\tilde{\mathbf{a}}(\lambda)$ is a solution if and only if

$$-\tilde{\mathbf{w}}_j^T \left(\tilde{\mathbf{y}} - \tilde{\mathbf{W}}\tilde{\mathbf{a}}(\lambda) \right) + \lambda \tilde{\mathbf{s}}_j = \mathbf{0} \ , \ j = 1, \ldots, P \tag{93}$$

where $\tilde{\mathbf{s}}_j$ is the sub differential of $\left\| [\tilde{a}_j, \tilde{a}_{j+P}]^T \right\|_2$, i.e.,

$$\tilde{\mathbf{s}}_j \in \begin{cases} \frac{[\tilde{a}_j, \tilde{a}_{j+P}]^T}{\|[\tilde{a}_j, \tilde{a}_{j+P}]^T\|_2} & \text{if } [\tilde{a}_j, \tilde{a}_{j+P}]^T \neq \mathbf{0} \\ \mathbf{v} & \text{if } [\tilde{a}_j, \tilde{a}_{j+P}]^T = \mathbf{0} \end{cases} \tag{94}$$

where $\mathbf{v}$ is a 2×1 vector such that $\|\mathbf{v}\|_2 \leq 1$. As is shown in [27], the condition on $\mathbf{v}$ can be strengthened to $\|\mathbf{v}\|_2 < 1$. Using (93), we can state the equivalent conditions for the complex valued case, which are

$$-\mathbf{w}_j^H \left(\mathbf{y} - \mathbf{W}\mathbf{a}(\lambda) \right) + \lambda s_j = 0 \ , \ j = 1, \ldots, P \tag{95}$$

Algorithm 2 LASSO singular points algorithm

initiate $\lambda^{(1)} = \max_{j\in\{1,\dots,P\}} |\mathbf{w}_j^H \mathbf{y}|$
for $k = 1, \dots, K-1$ **do**
  $\mathbf{a}(\lambda^{(k)}) = \underset{\mathbf{a}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{y} - \mathbf{W}\mathbf{a}\|_2^2 + \lambda^{(k)} \|\mathbf{a}\|_1$
  $I_k = \{j \mid a_j(\lambda^{(k)}) \neq 0\}$
  for $j' \notin I_k$ **do**
    initiate $\ell = 1$
    repeat
      $\Lambda_{j'}^{(\ell)} = R\big(\mathbf{w}_{j'}^H \mathbf{P}_{W_{I_k}}^{\perp} \mathbf{y}\ ,\ \mathbf{w}_{j'}^H \mathbf{W}_{I_k} (\mathbf{W}_{I_k}^H \mathbf{W}_{I_k})^{-1} \boldsymbol{\gamma}^{(\ell)}\big)$
      $\mathbf{a}_{I_k}^{(\ell)} = (\mathbf{W}_{I_k}^H \mathbf{W}_{I_k})^{-1} (\mathbf{W}_{I_k}^H \mathbf{y} - \Lambda_{j'}^{(\ell)} \boldsymbol{\gamma}^{(\ell)})$
      $\gamma_l^{(\ell+1)} = \frac{a_l^{(\ell)}}{|a_l^{(\ell)}|}\ , l = 1, \dots, |I_k|$
      $\ell \leftarrow \ell + 1$
    until convergence
    $\Lambda_{j'} = \Lambda_{j'}(\text{end})$
  end for
  $\lambda^{(k+1)} = \max_{j'\notin I_k} \Lambda_{j'}$
end for

where

$$s_j \in \begin{cases} \frac{a_j}{|a_j|} & \text{if } a_j \neq 0 \\ v & \text{if } a_j = 0 \end{cases} \tag{96}$$

with $|v| < 1$. Let I denote the set of indices corresponding to non-zero components of $\mathbf{a}(\lambda)$, i.e., $I = \{j \mid a_j(\lambda) \neq 0\}$ and let $\mathbf{W}_I = [\mathbf{w}_{j_1} \dots \mathbf{w}_{j_K}]$ be the part of the dictionary corresponding to these non-zero components. Then, if $\mathbf{a}(\lambda)$ is a solution to (84), it must hold that

$$\mathbf{w}_j^H (\mathbf{y} - \mathbf{W}\mathbf{a}(\lambda)) = \lambda \frac{a_j}{|a_j|}\ , j \in I \tag{97}$$

$$|\mathbf{w}_j^H (\mathbf{y} - \mathbf{W}\mathbf{a}(\lambda))| < \lambda\ , j \notin I \tag{98}$$

From (98), we see that the correlation between a vector $\mathbf{w}_j$ and the model residual $\mathbf{y} - \mathbf{W}\mathbf{a}(\lambda)$ decides whether a_j will be set to zero or not. Further, $\lambda' < \lambda$ is a singular point if it is the largest λ' that changes condition (98) to equality for one $j \notin I$, i.e.,

$$\lambda' = \max_{j\notin I} |\mathbf{w}_j^H (\mathbf{y} - \mathbf{W}\mathbf{a}(\lambda')| \tag{99}$$

where $\mathbf{a}(\lambda')$ solves (84) for $\lambda = \lambda'$. Let $\lambda^{(1)}$ be the largest λ yielding a non-zero solution. By definition $\mathbf{a}(\lambda) = \mathbf{0}$, for $\lambda > \lambda^{(1)}$, and we therefore see from (99) that

$$\lambda^{(1)} = \max_{j\in\{1,\dots,P\}} |\mathbf{w}_j^H \mathbf{y}| \tag{100}$$

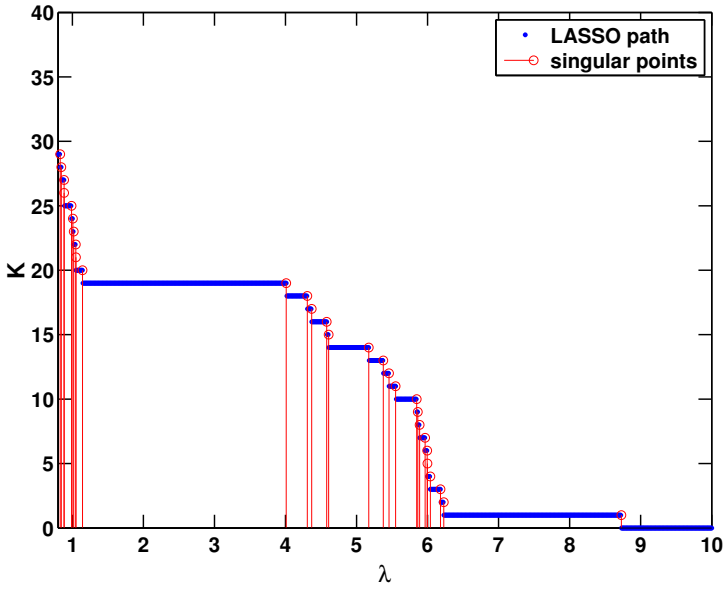

Figure 6: True LASSO path for a test signal consisting of four pitches with a total of 20 harmonics with 19 unique sinusoidal components. Also plotted are the 29 first singular points as given by Algorithm 2. Note that the LASSO path has not been computed for a fine enough grid to produce solutions for sparsity levels $K = 5, 21$, and 26.

Having $\lambda^{(1)}$, [25] proposes a scheme for determining $\lambda^{(k)}$ for $k > 1$ for the problem in (86). We here present the corresponding scheme for (84). Assume that we have obtained a singular point $\lambda^{(k)}$. Solving (84) for $\lambda^{(k)}$ yields the solution $\mathbf{a}(\lambda^{(k)})$ from which the set $I_k = \{j \mid a_j(\lambda^{(k)} \neq 0\}$ of active indices can be determined. Let $\mathbf{a}_{I_k}$ denote part of the vector $\mathbf{a}$ restricted to the index set I_k. Then, the next singular point $\lambda^{(k+1)}$ is determined by for each $j' \notin I_k$ solving the set of equations

$$\mathbf{w}_j^H(\mathbf{y} - \mathbf{W}\mathbf{a}_{I_k}(\Lambda_{j'})) = \Lambda_{j'} \frac{a_j(\Lambda_{j'})}{|a_j(\Lambda_{j'})|} \quad , j \in I_k \tag{101}$$

$$|\mathbf{w}_{j'}^H(\mathbf{y} - \mathbf{W}\mathbf{a}_{I_k}(\Lambda_{j'}))| = \Lambda_{j'} \tag{102}$$

and obtaining the next singular point as $\lambda^{(k+1)} = \max_{j' \notin I_k} \Lambda_{j'}$. In order to solve the counterpart equations to (101) and (102), a numerical iterative scheme is

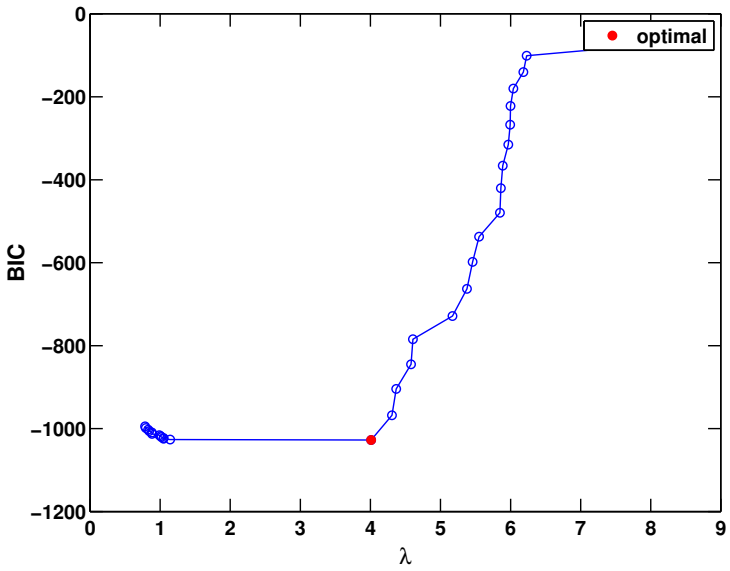

Figure 7: BIC computed for the models corresponding to the sparsity levels $K = 1, \ldots, 29$, plotted against corresponding values of the regularization parameter λ. The candidate models have been obtained using Algorithm 2. The optimal value of the regularization parameter is $\lambda = 4$. The signal consists of four pitches with a total of 20 harmonics of which 19 are unique.

proposed in [25]. The corresponding scheme here is to iterate

$$\begin{aligned} \Lambda_{j'}^{(\ell)} &= R\big(\mathbf{w}_{j'}^H \mathbf{P}_{W_{I_k}}^{\perp} \mathbf{y}\,,\ \mathbf{w}_{j'}^H \mathbf{W}_{I_k} (\mathbf{W}_{I_k}^H \mathbf{W}_{I_k})^{-1} \boldsymbol{\gamma}^{(\ell)}\big) \\ \mathbf{a}_{I_k}^{(\ell)} &= (\mathbf{W}_{I_k}^H \mathbf{W}_{I_k})^{-1} (\mathbf{W}_{I_k}^H \mathbf{y} - \Lambda_{j'}^{(\ell)} \boldsymbol{\gamma}^{(\ell)}) \\ \gamma_l^{(\ell+1)} &= \frac{a_l^{(\ell)}}{|a_l^{(\ell)}|} \quad , l = 1, \ldots, |I_k| \end{aligned} \tag{103}$$

where $\gamma_l^{(1)} = \frac{a_l(\lambda^{(k)})}{|a_l(\lambda^{(k)})|}$ for $l = 1, \ldots, |I_k|$, with $|I_k|$ denoting the cardinality of the set I_k, $R(a, b)$ the root of the equation $r = |a + rb|$, and

$$\mathbf{P}_{W_{I_k}}^{\perp} = \mathbf{I} - \mathbf{W}_{I_k} (\mathbf{W}_{I_k}^H \mathbf{W}_{I_k})^{-1} \mathbf{W}_{I_k}^H \tag{104}$$

Note that $\boldsymbol{\gamma}^{(1)}$ is initialized by the LASSO solution $\mathbf{a}_{I_k}(\lambda^{(k)})$. Note also that we here have to require $\mathbf{W}_{I_k}$ to have full column rank. The algorithm for finding the first K singular points is presented in Algorithm 2. The performance of this scheme is illustrated in Figure 6, where it has been used to determine the singular points $\lambda^{(k)}$, for $k = 1, \ldots, 29$, for a signal consisting of four pitches with a total of 20 harmonics of which 19 are unique. Also plotted in the figure is the sparsity level of the LASSO path evaluated for a grid of values of λ. As can be

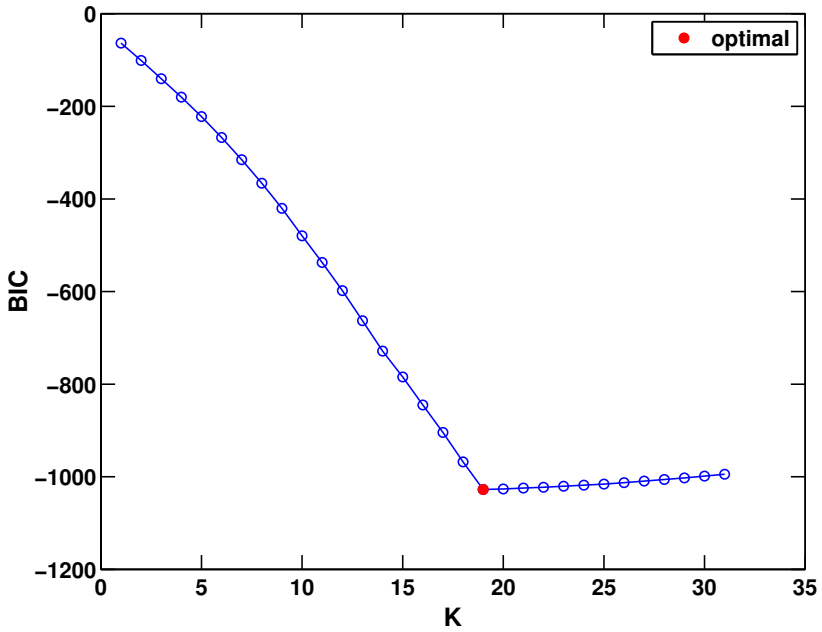

Figure 8: BIC computed according to (105) for models corresponding to different sparsity levels K. The candidate models have been obtained using Algorithm 2. The signal consists of four pitches with a total of 20 harmonics of which 19 are unique. Note that the BIC criterion correctly selects sparsity level $K = 19$ as the optimal one.

seen, the true LASSO path and the singular points determined by the iterative scheme agree. Note that the LASSO path has not been evaluated for a fine enough grid to identify sparsity levels $k = 5, 21$, and 26, whereas the iterative scheme correctly identifies all singular points. Also, considering computational speed, the iterative scheme for finding the singular points is much faster than finding the whole LASSO path as (84) only has to be solved for the singular points $\lambda^{(k)}$. Having fitted these K models with their respective sparsity level, one might choose the optimal model, and thereby the optimal λ, as the model minimizing the BIC criterion [23]

$$\mathrm{BIC}(\lambda^{(k)}) = 2N \log \hat{\sigma}^2(\lambda^{(k)}) + (5k + 1) \log N \tag{105}$$

where $\hat{\sigma}^2(\lambda^{(k)})$ is the MLE of the residual variance corresponding to the model determined by $\lambda^{(k)}$. Figures 7 and 8 present plots of $\mathrm{BIC}(\lambda^{(k)})$ computed according to (105) for the 29 singular points $\lambda^{(k)}$ also presented in Figure 6. Figure 7 plots $\mathrm{BIC}(\lambda^{(k)})$ against $\lambda^{(k)}$, whereas Figure 8 plots $\mathrm{BIC}(\lambda^{(k)})$ against the implied sparsity level k. Note that the BIC criterion correctly selects the singular point $\lambda = 4$ as the optimal, which corresponds to a solution $\mathbf{a}(\lambda)$ with 19 non-zero components.

4.2 Candidate model selection for PEBSI-Lite

Returning to our original problem of multi-pitch estimation, we want to choose λ_2 and λ_4, and given these regularization parameter values form our amplitude estimate from (46). One would therefore like to use something similar to Algorithm 2 to find a set of candidate models specified by (λ_2, λ_4) and from that set of models choose the optimal one according to some metric. The analog to finding singular points would here be to find singular ridges, i.e., curves through the $\lambda_2 - \lambda_4$-plane separating different sparsity levels. However, we now have two different candidate interpretations of sparsity; sparsity meaning the number of active components, i.e., harmonics, or block sparsity meaning the number active sources or pitches. Also, the equations that the optimal solution has to satisfy are now

$$-\mathbf{w}_j^H\big(\mathbf{y} - \mathbf{W}\mathbf{a}(\lambda)\big) + \lambda_2 s_j + \lambda_4(\tau_j - \tau_{j-1}) = 0 \qquad j = 2, \ldots, P-1 \tag{106}$$

$$-\mathbf{w}_j^H\big(\mathbf{y} - \mathbf{W}\mathbf{a}(\lambda)\big) + \lambda_2 s_j + \lambda_4 \tau_j = 0 \qquad j = 1 \tag{107}$$

$$-\mathbf{w}_j^H\big(\mathbf{y} - \mathbf{W}\mathbf{a}(\lambda)\big) + \lambda_2 s_j - \lambda_4 \tau_{j-1} = 0 \qquad j = P \tag{108}$$

where

$$s_j \in \begin{cases} \frac{a_j}{|a_j|} & \text{if } a_j \neq 0 \\ v_s & \text{if } a_j = 0 \end{cases} \tag{109}$$

$$\tau_j \in \begin{cases} \frac{a_j - a_{j-1}}{|a_j - a_{j-1}|} & \text{if } a_j - a_{j-1} \neq 0 \\ v_\tau & \text{if } a_j - a_{j-1} = 0 \end{cases} \tag{110}$$

where $|v_s| \leq 1$ and $|v_\tau| \leq 1$. Attempting something similar to Algorithm 2 would therefore be quite complicated. There have been suggestions on how to compute the solution path for the real variable counter part of our problem, called the sparse fused LASSO [28], which is formulated as

$$\min_{\boldsymbol{\beta}} \frac{1}{2} \|\mathbf{y} - \mathbf{W}\boldsymbol{\beta}\|_2^2 + \lambda_2 \|\boldsymbol{\beta}\|_1 + \lambda_4 \sum_{j=1}^{P-1} |\beta_j - \beta_{j+1}| \tag{111}$$

where $\mathbf{y} \in \mathbb{R}^N$, $\boldsymbol{\beta} \in \mathbb{R}^P$, and $\mathbf{W} \in \mathbb{R}^{N \times P}$. In [29], the authors present a very elegant way of computing the solution path in the case when $P = N$ and $\mathbf{W} = \mathbf{I}$, for which (111) is known as the Fused Lasso Signal Approximator in which the solution $\boldsymbol{\beta}$ is a smoothed version of the signal $\mathbf{y}$. However, for a general matrix $\mathbf{W}$, the algorithm becomes considerably complex. Also, the presented algorithm demands $\mathbf{W}$ to have full column rank, something that is not true for our dictionary $\mathbf{W}$. In [30], the authors present an approach to find the solution path of

$$\min_{\boldsymbol{\beta}} \frac{1}{2} \|\mathbf{y} - \mathbf{W}\boldsymbol{\beta}\|_2^2 + \lambda \|\mathbf{D}\boldsymbol{\beta}\|_1 \tag{112}$$

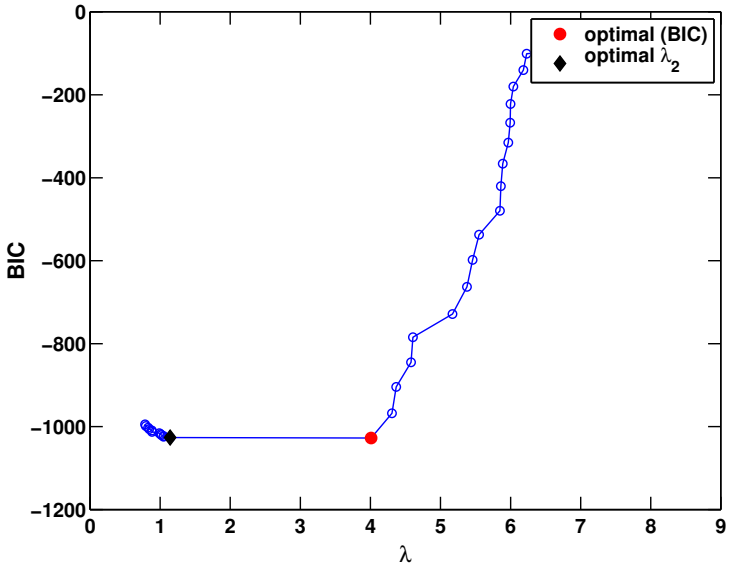

Figure 9: The BIC computed for the models corresponding to the sparsity levels $K = 1, \ldots, 29$ plotted against corresponding values of the regularization parameter λ. The candidate models have been obtained using Algorithm 2. The optimal value of the regularization parameter for the LASSO problem is $\lambda = 4$. However, for the pitch estimation problem, we set $\lambda_2 = 1.1427$, slightly larger than the next singular point. The signal consists of four pitches with a total of 20 harmonics of which 19 are unique.

for the real-variable case with a general penalty matrix $\mathbf{D}$ by considering the solution paths of the dual variable. Unfortunately, this is only for the one-dimensional case, i.e., for the case when the minimization has only a single regularization parameter, λ.

Instead of trying to determine appropriate values of (λ_2, λ_4) simultaneously, one could try to decouple the problem by first determining the value of one of the parameters and then move on to determine the value of the other. Having a fast path algorithm for the LASSO problem in Algorithm 2, a simple idea would be to first solve (84), set λ_2 to the optimal λ as determined by (105), and then conduct a line search for λ_4. It should be noted that the dictionary $\mathbf{W}$ used in (47) contains columns that are potentially identical as two or more candidate pitches may have overlapping harmonics. Using such a $\mathbf{W}$ in (84) renders the problem ill-posed, with infinitely many solutions. To remedy this, one might construct $\breve{\mathbf{W}}$ as the dictionary containing only unique columns of $\mathbf{W}$. Assume that we have run Algorithm 2 using $\breve{\mathbf{W}}$ and have determined k to be the optimal number of sinusoids with corresponding singular point $\lambda^{(k)}$. These k sinusoids, corresponding to k unique frequencies, might be a superposition of a larger num-

ber of harmonics, having overlapping frequencies. Together with the fact that we introduce an additional penalty with λ_4 when performing pitch estimation, setting $\lambda_2 = \lambda^{(k)}$ might therefore yield only solutions with all amplitudes a_j set to zero. A more conservative choice would instead be to set $\lambda_2 = \lambda^{(k+1)} + \epsilon$ for some $\epsilon > 0$, as $\lambda \in (\lambda^{(k+1)}, \lambda^{(k)}]$ yields k non-zero components when solving (84). This is illustrated in Figure 9. That would leave only a line search to determine λ_4. Also, as performing this line search with the full dictionary $\mathbf{W}$ would be computationally cumbersome, we could exploit the knowledge gained from the solution of (84). As we know the present sinusoidal components, we can discard the pitches in $\mathbf{W}$ that have no harmonics that correspond to any of the k detected sinusoids, resulting in a smaller problem that can be solved faster.

Although this approach seems attractive at first glance and works in some standardized cases, it turns out that we in practice cannot use Algorithm 2 to determine λ_2 and thereby reduce the complexity of our problem. This is due to that Algorithm 2 only converges for a sufficiently large number of singular points if the dictionary $\breve{\mathbf{W}}$ is moderately resolved in frequency. Constructing $\breve{\mathbf{W}}$ from the unique columns from our standard pitch dictionary $\mathbf{W}$ yields too high frequency resolution, causing Algorithm 2 to break down. Thus, using this approach as a preprocessing step will only work if one has prior knowledge of the frequency content of the signal, i.e., there are no off-grid effects, and if the frequencies are not too closely spaced, something that we cannot assume to hold.

4.3 Adaptive dictionary construction

As finding an efficient path algorithm for PEBSI-Lite proves elusive, an alternative approach could be to conduct a grid search to find appropriate values of λ_2 and λ_4. As noted earlier, such a search using the full pitch dictionary $\mathbf{W}$ would be computationally cumbersome. Therefore, this section proposes a signal dependent dictionary construction aimed at forming a dictionary that is smaller than the default dictionary, $\mathbf{W}$, but better suited to the signal to be analysed.

The dictionary construction begins by estimating the frequency content of the signal without imposing a harmonic structure, i.e., we are just estimating multiple sinusoids in noise. This estimation could be performed by standard methods such as ESPRIT. As the number of sinusoidal components is unknown, estimates corresponding to different model orders can be evaluated using the BIC criterion (105) in order to choose a suitable model order. As the only interesting pitch candidates are those having at least one harmonic corresponding to a present sinusoidal component, we can design a considerably reduced dictionary, containing only pitches with such matching harmonics. If one has some prior knowledge of the nature of the signal, one could impose stronger assumptions on the candidate pitches in order to reduce the dictionary further, e.g.,

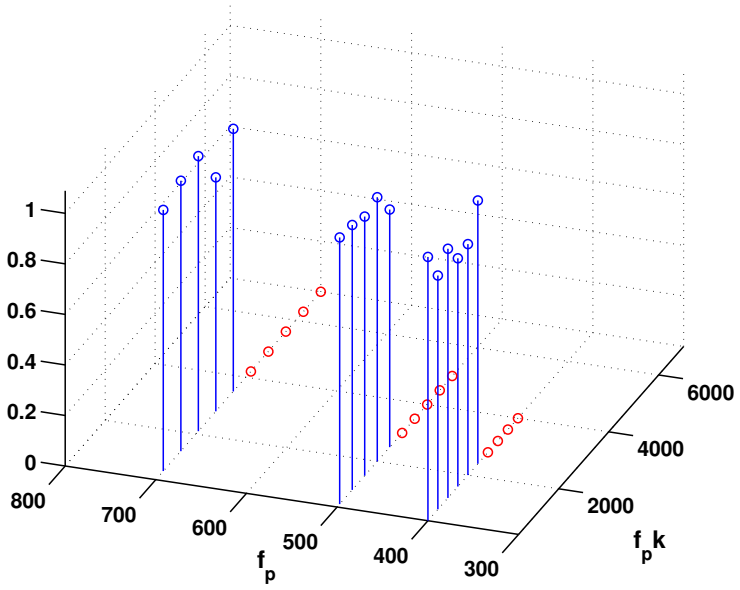

Figure 10: True pitch spectrum of a three pitch signal with fundamental frequencies 400, 500 and 700 Hz with 6, 5, and 5 harmonics, respectively. The f_p-axis gives the fundamental frequency of each pitch and the $f_p k$-axis the frequencies of the harmonics.

by allowing only pitches whose first harmonic is found in the set of estimated sinusoids. Using the obtained dictionary, one could then proceed to conduct a search for λ_2 and λ_4. However, with this smaller dictionary, the total variation penalty as formulated in (45) might result in erroneous solutions. This is illustrated in Figures 10 and 11. Figure 10 displays the true pitch spectrum of a three pitch signal with fundamental frequencies 400, 500, and 700 Hz and 6, 5, and 5 harmonics, respectively. When constructing a dictionary based on estimated frequency content using ESPRIT, the candidate pitches with fundamental frequencies 350 and 400 Hz are placed in adjacent blocks. Subsequently, when performing pitch estimation with PEBSI-Lite the optimal solution is the one presented in Figure 11. Note that the fifth harmonic of the 700 Hz pitch now has been mapped to the tenth harmonic of the 350 Hz candidate pitch. This erroneous solution is due to that the total variation penalty in (45) penalizes amplitude variation not only within pitches, but also between adjacent pitches. When using a large dictionary, this does not pose a problem as it is unlikely that adjacent pitches will have good fit to the signal. However, the dictionary considered now is constructed of only pitches with reasonable signal fit. In order to remedy this, the total variation penalty has to be modified. Note that (45) resulted in not only smoothness, but also block sparsity in the case of a large dictionary $\mathbf{W}$. In order to keep the block sparse effect, one can change the total

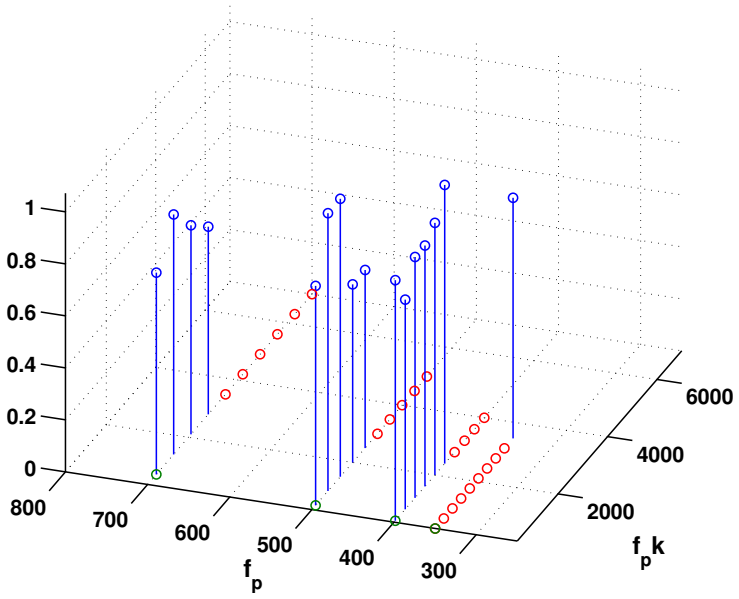

Figure 11: Estimated pitch spectrum of a three pitch signal with fundamental frequencies 400, 500 and 700 Hz with 6, 5, and 5 harmonics, respectively. The f_p-axis gives the fundamental frequency of each pitch and the $f_p k$-axis the frequencies of the harmonics.

variation penalty to

$$g_4(\mathbf{\Psi}, \mathbf{\Phi}) = \sum_{p=1}^{P} \sum_{\ell=0}^{L_{\max}} \left| a_{p,\ell+1} e^{-\varphi_{p,\ell+1}} - a_{p,\ell} e^{-\varphi_{p,\ell}} \right| \tag{113}$$

$$a_{p,0} e^{-\varphi_{p,0}} = a_{p,L_{\max}+1} e^{-\varphi_{p,L_{\max}+1}} = 0 \quad , \forall p \tag{114}$$

In matrix notation, this means that the matrix $\mathbf{H}_4^{(k)}$ in (57) is modified by redefining the matrix $\mathbf{F}$ as a $P(L_{\max}+1) \times PL_{\max}$ matrix $\mathbf{F} = \text{diag}(\mathbf{F}_1, \ldots, \mathbf{F}_P)$, where each block $\mathbf{F}_p$ is a $(L_{\max}+1) \times L_{\max}$ matrix with elements

$$f_{k,\ell} = \begin{cases} 1 & \text{if } k = \ell = 1 \\ -1 & \text{if } k = \ell \neq 1 \\ 1 & \text{if } k = \ell + 1 \\ 0 & \text{otherwise} \end{cases} \tag{115}$$

Algorithm 1 remains the same, except for the auxiliary variable $\mathbf{u}_4$ and dual variable $\mathbf{d}_4$ which now both are in $\mathbb{C}^{P(L_{\max}+1)}$. Using this modified total variation penalty, pitch estimates are independent of the ordering of the candidate pitches in the dictionary and the reduced dictionary can be used without experiencing the above mentioned problem. Having this in place, the search for

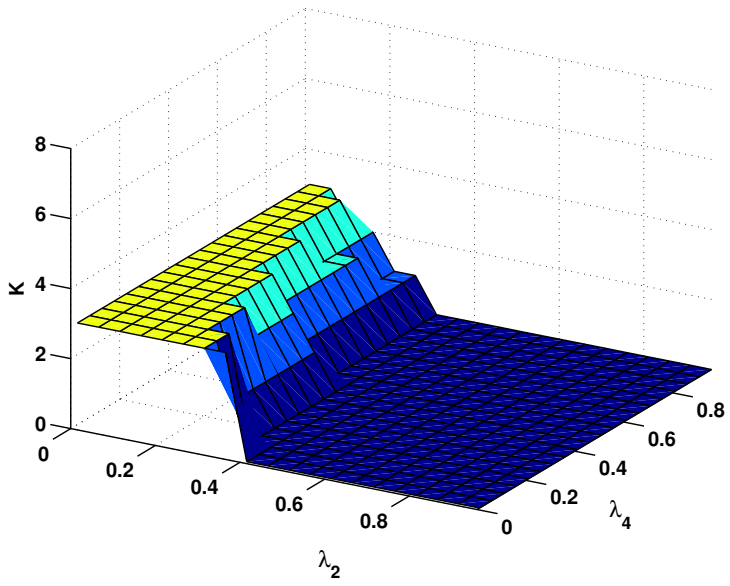

Figure 12: Number of pitches, (K), present in the solution of PEBSI-Lite for different values (λ_2, λ_4) when applied to a three pitch signal with 4, 8, and 12 harmonics, respectively.

(λ_2, λ_4) can be conducted. Although considerably cheaper as compared to when performed using a full dictionary, a complete evaluation of the $\lambda_2 - \lambda_4$ plane is still quite expensive. To avoid a full grid search, the following heuristic concerning the connection between λ_2 and λ_4 can be used. Assume that we have a single pitch signal where all L_k harmonics have equal magnitude r. Further, assume that when setting $\lambda_4 = 0$, λ_2' is the largest value of λ_2 resulting in a nonzero solution, where each harmonic amplitude is estimated to r'. If we would instead set $\lambda_2 = 0$ and consider which value of λ_4 that should result in the same solution, this value should be

$$\lambda_4' = \frac{L_k}{2}\lambda_2' \tag{116}$$

as this would result in precisely the same penalty as with $\lambda_4 = 0$, $\lambda_2 = \lambda_2'$. If we assume (116) to be true, we should, for spectrally smooth signals, expect to see ridges in the solution surface where the number of pitches present in the solution changes, and the shapes of the ridges in the $\lambda_2 - \lambda_4$ plane should be described by lines similar to (116). This indeed seems to be the case. Figure 12 presents a plot of the number of pitches present in the solution for different values (λ_2, λ_4) for a signal consisting of three pitches with 4, 8, and 12 harmonics, where each harmonic amplitude has been drawn uniformly on $(0.9, 1.1)$. On the plateau with two pitches, the pitch with four harmonics have been set to zero, whereas

Algorithm 3 Self-Regularized PEBSI-Lite

initiate $\hat{\boldsymbol{\omega}} = \emptyset$, $\ell = 1$
repeat {sinusoidal component estimation}
  $\hat{\boldsymbol{\omega}}_\ell \leftarrow \ell$ sinusoidal components from ESPRIT
  $\text{BIC}_\ell \leftarrow 2N \log \hat{\sigma}^2(\hat{\boldsymbol{\omega}}_\ell) + (5\ell + 1) \log N$
until $\text{BIC}_\ell > \text{BIC}_{\ell-1}$
construct dictionary $\mathbf{W}$ from $\hat{\boldsymbol{\omega}}_{\ell-1}$
$L \leftarrow$ largest number of active harmonics among candidate pitches in $\mathbf{W}$
initiate $\lambda = \epsilon$, $k = 1$
$\hat{\sigma}_y^2 \leftarrow \text{Var}(y)$
repeat {regularization parameter line search}
  $\lambda_2 \leftarrow \lambda$, $\lambda_4 \leftarrow \frac{L}{2}\lambda$
  form amplitude estimate $\hat{\mathbf{a}}^{(k)}$ from Algorithm 1
  estimate the variance of the model residual $\hat{\sigma}^2(\lambda_2, \lambda_4)$
  $\lambda \leftarrow \lambda + \epsilon$
  $k \leftarrow k + 1$
until $\hat{\sigma}^2(\lambda_2, \lambda_4) > \tau \hat{\sigma}_y^2$
$\hat{\mathbf{a}} \leftarrow \hat{\mathbf{a}}^{(k-1)}$

on the plateau with one pitch present, only the pitch with twelve harmonics is present. Note the shape of the different plateaus: seen in the $\lambda_2 - \lambda_4$ plane, the slopes of the ridges seem to be proportional to (116) where $L_k = 4, 8$, and 12, for the three ridges corresponding to changes from three to two, from two to one, and from one to zero pitches, respectively. The signal corresponding to Figure 12 has a relatively low noise level, with SNR = 20 dB. Decreasing the SNR-level, the least regularized solutions, i.e., with λ_2 and λ_4 close to zero, results in more than three non-zero pitches. Guided by this observation, one could by a re-parametrization reduce the search for (λ_2, λ_4) from a two-dimensional to a one dimensional search. Keeping the plateaus in Figure 12 and our assumption of spectral smoothness in mind, we should expect a desirable solution to correspond to a (λ_2, λ_4)-pair with $\lambda_2 \leq \lambda_4$. In order to get solutions regularized with respect to spectral smoothness, while keeping the risk of getting only zero solutions low, the following parametrization can be used. Let λ denote the only free parameter and set

$$\lambda_2 = \lambda \tag{117}$$

$$\lambda_4 = \frac{L}{2}\lambda \tag{118}$$

where L is the largest number of harmonics among the pitches present in the signal. Although L is unknown, it can be estimated during the dictionary construction phase as we have access to ESPRIT estimates of the signal's sinusoidal components. Having this in place, a line search can be conducted for the value of λ. As for choosing an optimal λ from the set of candidates, we unfortunately, as noted earlier, do not have a fully functioning BIC criterion that takes into

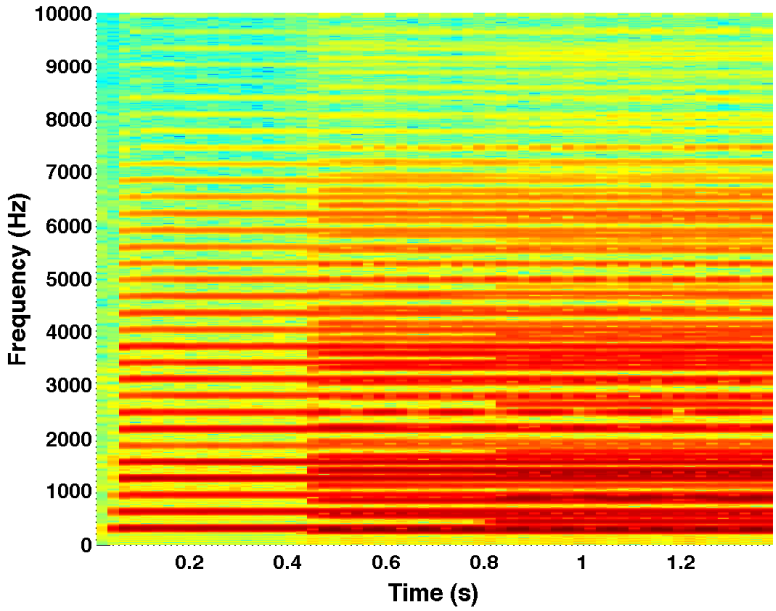

Figure 13: Spectrogram for a signal consisting of one, two and lastly three MIDI-saxophones playing notes with fundamental frequencies 311, 277, and 440 Hz, respectively.

account the harmonic structure of the signal and that would allow us to choose between candidate solutions. Though, assuming that we obtain at least one solution that correctly retrieves the support of the true pitches, one could make the model order choice based on the MLE of the residual variance σ_λ^2 as follows. Having obtained a solution with PEBSI-Lite using the regularization parameter λ, the residual variance σ_λ^2 can be estimated by least squares and the unique frequencies of that solution. In low noise environments, we expect false pitches that model noise to not contribute much to the signal power. Thus, the first significant rise in residual variance is expected to occur when one of the true pitches are set to zero. Therefore, we propose keeping only models that correspond to lower values of σ_λ^2 and then choosing the optimal model as the one having the least number of active pitches. This might be expected to work for high values of SNR but might break down when the power of the noise is close to that of the clean signal. The complete algorithm for the dictionary construction, line search, and pitch estimation is outlined in Algorithm 3, where ϵ denotes the step size of the line search and $\tau \in (0, 1)$ is a threshold for detecting an increase in model residual variance. Figure 13 shows a plot of the spectrogram of a signal consisting of three MIDI-saxophones playing notes with fundamental frequencies 311, 277, and 440 Hz. The 311 Hz saxophone starts out alone and is after 0.45 seconds joined by the 277 Hz saxophone and after 0.95 seconds by the

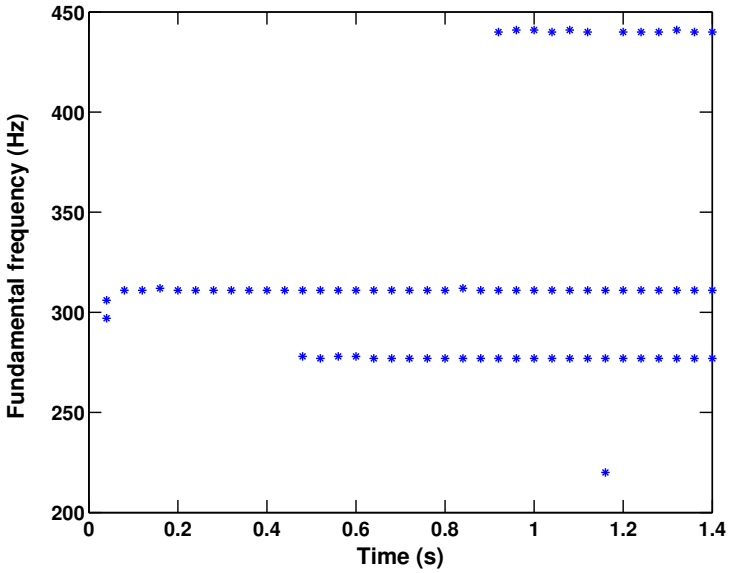

Figure 14: Frame wise pitch estimates for a signal consisting of one, two and lastly three MIDI-saxophones playing notes with fundamental frequencies 311, 277, and 440 Hz, respectively.

440 Hz saxophone. The image is quite blurred for the later parts of the signal, but for the first half second, one can clearly see the harmonic structure of the saxophone pitch. It is worth noting that a large number harmonics is present. Figure 14 shows pitch estimates produced by Algorithm 3, using $\tau = 0.1$, when applied to the same signal. As can be seen, the estimates are quite accurate, with the exception of the beginning of the first tone and for a single frame where the 440 Hz pitch is mistaken for a 220 Hz pitch.

4.4 Numerical results

In order to examine the performance of Algorithm 3, it was evaluated using a simulated triple-pitch signal, measured in white Gaussian noise at different SNR levels, ranging from 0 dB to 25 dB, in steps of 5 dB. To make the simulations realistic, the spectral envelopes of the three pitch components were constructed from periodograms of three different speech recordings. The formants of the three pitches are displayed in Figure 15. The pitches had fundamental frequencies 200, 350, and 530 Hz, and 7, 8, and 11 harmonics, respectively. At each level of SNR, 1000 Monte Carlo simulations were performed, where the fundamental frequencies were chosen uniformly on 200 ± 2.5, 350 ± 2.5, and 530 ± 2.5 Hz, respectively, and the phase of each harmonic was chosen uniformly on $[0, 2\pi)$. The signal was sampled in a 40 ms window at a sampling frequency of 20 kHz,

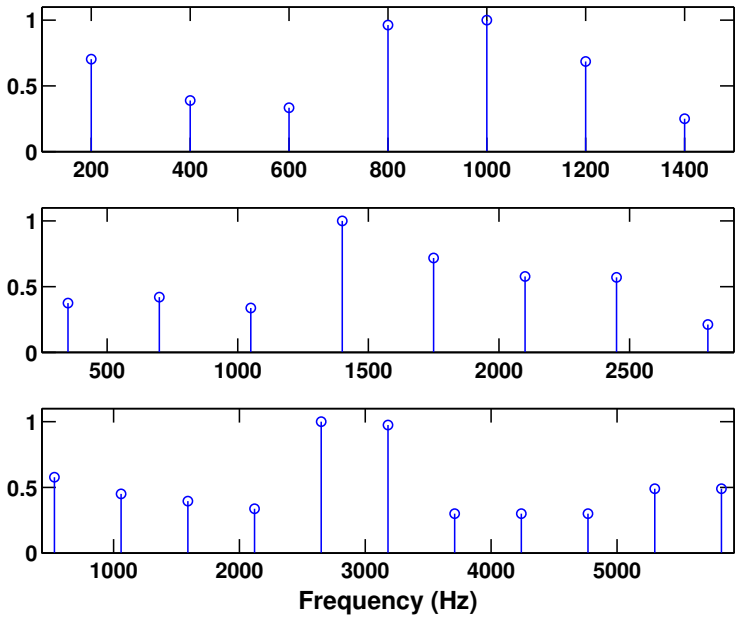

Figure 15: Formants for the three pitches constituting the test signal for the Monte Carlo simulations.

generating 800 samples of the signal. The algorithm settings were $\tau = 0.1$ and $\epsilon = 0.05$. Here, Algorithm 3 was compared to the ANLS, ORTH, harmonic Capon, as well as PEBS-TV estimators. The three first comparison methods were given the oracle model orders. To illustrate the fact that the choice of regularization parameter values is not universal, the values in Table 2 were used initially. However, this resulted in such poor performance that the parameter values had to be slightly altered in order to make PEBS-TV an interesting reference method. As a compromise, the parameter values corresponding to SNR 20 dB in Table 2 were used for all SNRs in this simulation setting. For the dictionaries of PEBSI-Lite and PEBS-TV, $L_{\max} = 16$ is used. Figure 16 shows the percentage of the pitch estimates where all three pitch estimates lie within ± 2 Hz of the true values for the five different methods. As can be seen, the performance of PEBSI-Lite is poor for low SNRs while improving considerably in lower noise settings. The low scoring for PEBSI-Lite for low SNRs is mainly due to selection of wrong model orders. This is illustrated in Figure 17, which shows the percentage of the estimates in which PEBSI-Lite and PEBS-TV selects the correct number of pitches. As can be seen, for an SNR of 0 dB, PEBSI-Lite selects the true model order in less than 10% of the simulations. Mostly, a too high model order is selected, which is to be expected as the model order choice is based on the power of the model residual and that the pitch estimates depend on the accuracy of the initial ESPRIT estimates. Though, for higher SNRs, PEBSI-Lite clearly outperforms the reference methods. Arguably, one

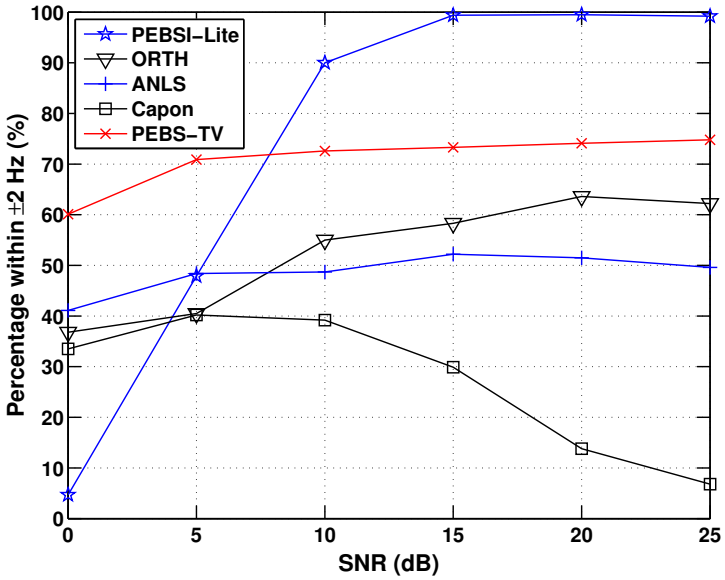

Figure 16: Percentage of estimated pitches where all three fundamental frequencies lie at most 2 Hz from the ground truth.

could improve on these results by either using prior knowledge of the noise level or by estimating it, and based on this make the model order selection scheme more robust. Figure 18 shows a plot of the root mean squared error (RMSE) for the estimated fundamental frequencies. Instead of presenting three separate RMSE plots, Figure 18 shows an aggregate version where the MSE for the three pitches have been summed. In order to construct the RMSE values for PEBSI-Lite and PEBS-TV, estimates where the model order has not been correctly determined have been discarded. Thus, for SNR level 0 dB, the RMSE values for PEBSI-Lite is based on quite few samples. However, as PEBSI-Lite finds the correct model order for high SNR levels with high probability, the corresponding RMSE values are more trustworthy in these regions. For the reference methods ORTH, ANLS, Capon, and PEBS-TV, some of the estimates deviate from the true pitch frequencies with as much as 100 Hz, resulting in very large RMSE values should all estimates be used in their computation. Thus, in order to obtain RMSE values comparable to that of the PEBSI-Lite estimates, only estimates found within 2 Hz of the true pitch frequencies are used when computing RMSE for the reference methods. With this, as can be seen in Figure 18, PEBSI-Lite performs worse than the reference methods for SNRs of 0 to 10 dB, while outperforming all reference methods except Capon for SNRs of 20 and 25 dB. Though, one should bear in mind that the RMSE values for Capon for these SNRs are based on only 15% respectively 8% of the available pitch estimates, as can be seen in Figure 16.

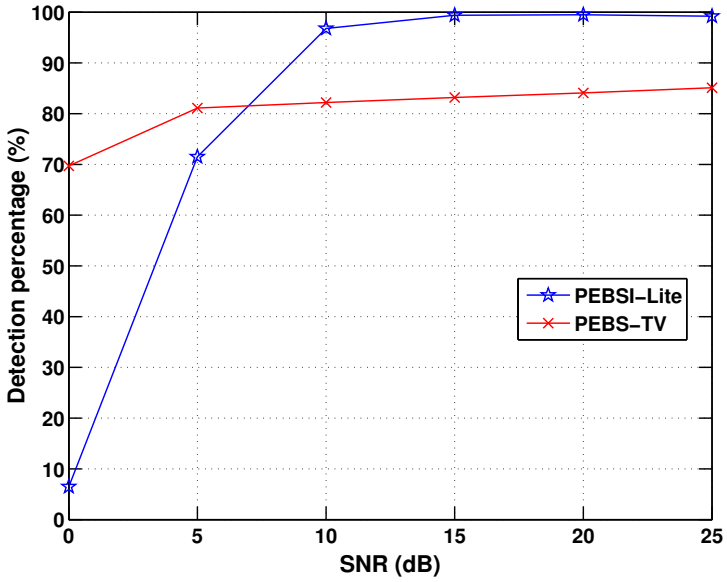

Figure 17: Estimated probability of PEBSI-Lite determining the correct number of pitches for the triple pitch test signal.

Also presented in Figure 18 is the root Cramér-Rao lower bound (CRLB) for the estimates of the pitch frequencies. As the frequencies of the harmonics in this case are distinct and the additive noise is Gaussian, the lower limit for the variance of an unbiased pitch frequency estimat $\hat{f}_k$ is given by [5]

$$\mathrm{Var}(\hat{f}_k) \geq \frac{6\sigma^2(f_s/2\pi)^2}{N(N^2-1)\sum_{\ell=1}^{L_k}|a_{k,\ell}|^2\ell^2} \tag{119}$$

where σ^2 is the variance of the additive noise, $a_{k,\ell}$ is the amplitude of harmonic ℓ of pitch k, N is the number of data samples, and f_s is the sampling frequency. In analog with the summed MSE values for the pitch estimates, the root CRLB curve presented here is the sum of the three separate limits, i.e.,

$$\mathrm{CRLB} = \sum_{k=1}^{3} \frac{6\sigma^2(f_s/2\pi)^2}{N(N^2-1)\sum_{\ell=1}^{L_k}|a_{k,\ell}|^2\ell^2} \tag{120}$$

As can bee seen in Figure 18, PEBSI-Lite, as well as the other methods, fail to reach the CRLB. In an attempt to improve the PEBSI-Lite estimates for SNR levels above and including 15 dB, a non-linear least squares (NLS) search was performed. This means that we obtain refined estimates of the pitch frequencies f_k contained in the vector $\mathbf{f}$ as (see, e.g, [31])

$$\mathbf{f} = \underset{\mathbf{f}}{\mathrm{argmax}}\ \mathbf{y^H B}(\mathbf{B^H B})^{-1}\mathbf{B^H y} \tag{121}$$

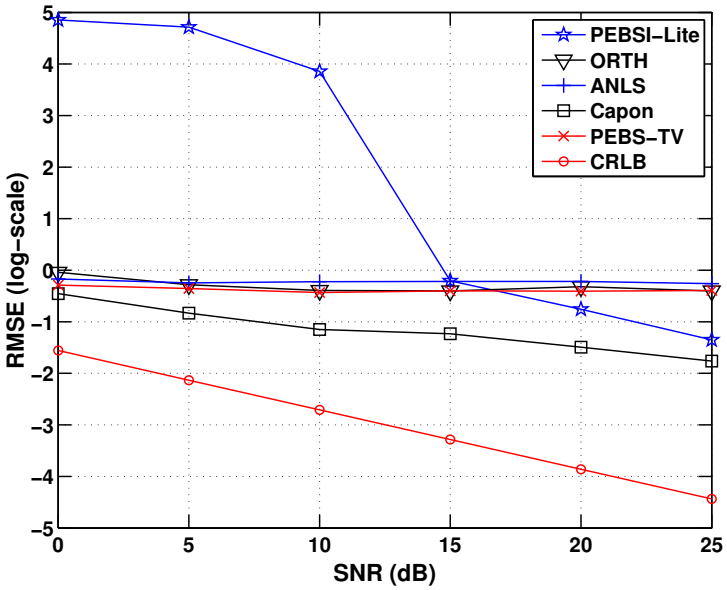

Figure 18: The RMSE for the fundamental frequency estimates for the triple pitch test signal. Also plotted is the (root) CRLB. For PEBSI-Lite and PEBS-TV, only estimates where the number of pitches is found are considered. For the reference methods ORTH, ANLS, Capon, and PEBS-TV only estimates where all estimated pitch frequencies lie within 2 Hz of the true pitch frequencies are considered.

where $\mathbf{B}$ is a block matrix consisting of K blocks, $\mathbf{B} = [\mathbf{B}_1, \ldots, \mathbf{B}_K]$, where each block $\mathbf{B}_j$ corresponds to a pitch and is constructed as

$$\mathbf{B}_j = \begin{bmatrix} e^{i2\pi f_j/f_s t_1} & \ldots & e^{i2\pi L_j f_j/f_s t_1} \\ \vdots & & \vdots \\ e^{i2\pi f_j/f_s t_N} & \ldots & e^{i2\pi L_j f_j/f_s t_N} \end{bmatrix} \tag{122}$$

Given that the PEBSI-Lite estimates are fairly close to the true pitch frequencies, we expect the NLS scheme to converge if we solve (121) using routines like MATLAB's *fminsearch* initialized with the PEBSI-Lite estimates. However, the success of such a scheme is not only dependent on good initial frequency estimates, we also need the true number of harmonics L_j for each pitch. Figure 19 presents a plot of the average absolute error in the number of detected harmonics for each pitch for the test signal when using PEBSI-Lite. As can be seen, the number of detected harmonics is only correct for the third pitch even for the largest SNRs. The errors in number of harmonics for the first and second pitches are due to the relatively small amplitudes of both pitches highest order harmonics, as shown in Figure 15, making these harmonics prone to being cancelled out by the PEBSI-Lite regularization penalties. Using erroneous harmonic orders

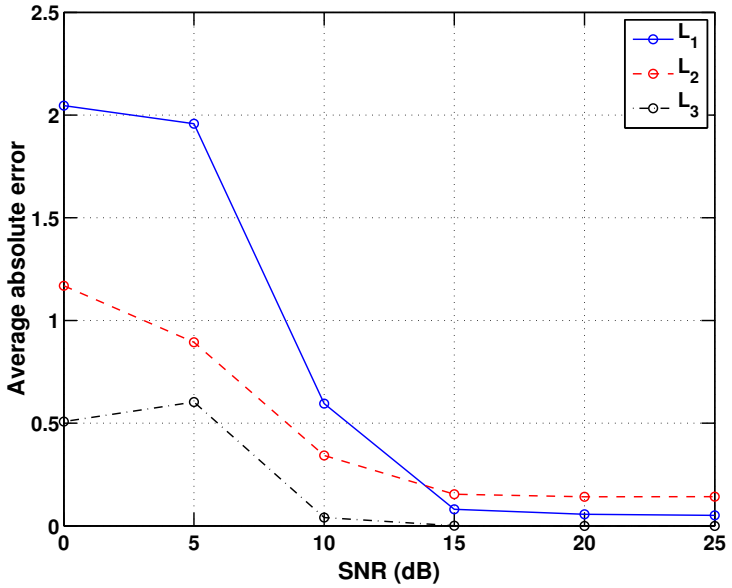

Figure 19: The average absolute error in the number of detected harmonics (L_1, L_2, L_3) for the three pitches of the test signal when using PEBSI-Lite. Only estimates where the right number of pitches is found are considered.

as input to the NLS search, we expect the resulting pitch frequency estimates to be somewhat biased. Indeed, this is what happens. Figure 20 presents a plot of the RMSE of the pitch frequency estimates when the PEBSI-Lite estimates for SNRs above and including 15, 20, and 25 dB have been post-processed using NLS. As can be seen, we still fail to reach the CRLB, although the estimation errors have become smaller. Note also that the slopes of the RMSE curve for PEBSI-Lite and CRLB now are somewhat different, which is due to that the erroneous harmonic orders induces varying degrees of bias in the estimates.

Considering computational complexity, ANLS and ORTH are by far the fastest methods, with average running times of 0.03 and 1.6 seconds per estimation cycle, respectively. For Capon and PEBS-TV, the corresponding running times are 6.1 and 6.4 seconds, respectively, while running PEBSI-Lite using Algorithm 3 requires on average 40.1 seconds per estimation cycle. Although Algorithm 3 is considerably more expensive to run than the reference methods, it should be noted that the method does not require any user input in terms of regularization parameter values. PEBS-TV could arguably be tuned to perform on par with PEBSI-Lite if one is allowed to change the values of its regularization parameters. However, PEBS-TV needs the setting of three parameter values and after trying only seven such triplets, the computational time is the same as running Algorithm 3.

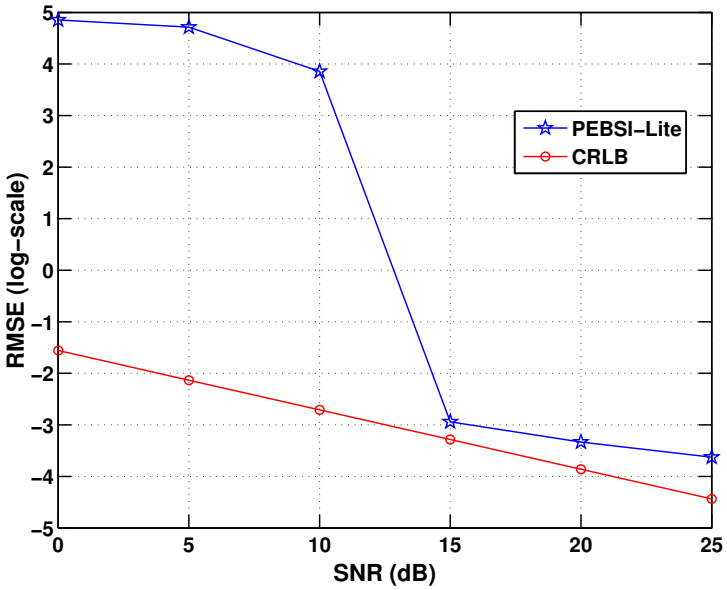

Figure 20: The RMSE for the fundamental frequency estimates where the estimates obtained using PEBSI-Lite have been improved using NLS for SNR levels 15, 20, and 25 dB, as compared to the (root) CRLB. Only estimates where the number of pitches is found are considered.

5 Discussion and conclusions

The proposed algorithm PEBSI-Lite has been shown to be an accurate method for multi-pitch estimation. The method was shown to perform as good as, or better than, state of the art methods when evaluated using Monte Carlo simulations of a smooth synthetic two-pitch signal. The advantage over similar methods presented in [6] is that fewer regularization parameters are needed, simplifying the calibration of the model. However, automatically choosing the regularization parameters proved to be a difficult task. A method for finding an appropriate model, and thereby the regularization parameters, by means of a line search was presented, combined with a scheme for constructing an efficient dictionary of candidate pitches. Combined with this scheme, PEBSI-Lite was shown to outperform other multi-pitch estimation methods for high levels of SNR, while breaking down in too noisy settings. Also, even if this scheme would fail to select the correct model order, the obtained efficient dictionary facilitates a more rigorous grid search in terms of computational complexity. Such a grid search could also exploit information about the solution surface obtained from the line search. Although the results are encouraging, it should be noted that the design of the method prohibits any economics of scale: the dictionary and regularization parameters are tailored specifically to the signal, so when performing pitch estimation over several frames, consecutive frames cannot necessarily share the same dictionaries as different pitches may be present in different frames. Also, with the present design of the method, it seems hard to obtain statistically efficient pitch frequency estimates.

6 Future research

In its present formulation, PEBSI-Lite does not allow for any sharing of information between consecutive time frames, even if the algorithm should be applied to multi-frame signals for, e.g., music transcription. Arguably, the method could be enhanced both in terms of accuracy and in terms of speed, could it incorporate information about pitch estimates from preceding time frames when forming pitch estimates. An issue not addressed in this thesis is inharmonicity. As noted earlier, the perfectly harmonic model of pitches fail to capture the frequency content of, e.g., the piano. An interesting area of investigation would therefore be finding ways of adaptively adjusting the harmonic frequencies of the dictionary, should one want to remain in the field of sparse recovery. There are methods available that are able to handle inharmonicity in the case of known sources, such as pianos, so elaborating on such approaches to make them more general would be an interesting topic. As for the issue of choosing regularization parameters, a path algorithm for efficiently finding solutions for all such choices to sparse recovery problems should be of great interest also outside the field of multi-pitch estimation.

References

[1] M. Müller, D. P. W. Ellis, A. Klapuri, and G. Richard, "Signal Processing for Music Analysis," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1088–1110, 2011.

[2] A. Wang, "An Industrial Strength Audio Search Algorithm," in *4th International Conference on Music Information Retrieval*, Baltimore, Maryland USA, Oct. 26-30 2003.

[3] N. H. Fletcher and T. D. Rossing, *The Physics of Musical Instruments*, Springer-Verlag, New York, NY, 1988.

[4] H. Fletcher, "Normal vibration frequencies of stiff piano string," *Journal of the Acoustical Society of America*, vol. 36, no. 1, 1962.

[5] M. Christensen and A. Jakobsson, *Multi-Pitch Estimation*, Morgan & Claypool, 2009.

[6] 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.

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

[8] 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 Processing Letters*, vol. 21, no. 12, pp. 1536–1540, December 2014.

[9] 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 Language Processing*, vol. 18, no. 6, pp. 1643–1654, Aug. 2010.

[10] A. Klapuri, "Multiple fundamental frequency estimation based on harmonicity and spectral smoothness," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 11, no. 6, pp. 804–816, 2003.

[11] 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.

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

[13] 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, France, Aug. 31-Sept. 4 2015.

[14] R. H. Tutuncu, K. C. Toh, and M. J. Todd, "Solving semidefinite-quadratic-linear programs using SDPT3," *Mathematical Programming Ser. B*, vol. 95, pp. 189–217, 2003.

[15] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Op-

timization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

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

[17] T. Kronvall, S. I. Adalbjörnsson, and A. Jakobsson, "Joint DOA and Multi-Pitch Estimation Using Block Sparsity," in *39th IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Florence, May 4-9 2014.

[18] T. Kronvall, S. I. Adalbjörnsson, and A. Jakobsson, "Joint DOA and Multi-pitch estimation via Block Sparse Dictionary Learning," in *22nd European Signal Processing Conference (EUSIPCO)*, Lisbon, Sept. 1-5 2014.

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

[20] E. J. Candes, J. Romberg, and T. Tao, "Robust Uncertainty Principles: Exact Signal Reconstruction From Highly Incomplete Frequency Information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[21] E. J. Candes, 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.

[22] M. A. T. Figueiredo and J. M. Bioucas-Dias, "Algorithms for imaging inverse problems under sparsity regularization," in *Proc. 3rd Int. Workshop on Cognitive Information Processing*, May 2012, pp. 1–6.

[23] P. Stoica and Y. Selén, "Model-order Selection — A Review of Information Criterion Rules," *IEEE Signal Processing Magazine*, vol. 21, no. 4, pp. 36–47, July 2004.

[24] C. D. Austin, R. L. Moses, J. N. Ash, and E. Ertin, "On the Relation Between Sparse Reconstruction and Parameter Estimation With Model Order Selection," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, pp. 560–570, 2010.

[25] A. Panahi and M. Viberg, "Fast Candidate Point Selection in the LASSO Path," *IEEE Signal Processing Letters*, vol. 19, no. 2, pp. 79–82, Feb 2012.

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

[27] 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.

[28] 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.

[29] H. Hoefling, "A Path Algorithm for the Fused Lasso Signal Approximator," *Journal of Computational and Graphical Statistics*, vol. 19, no. 4, pp. 984–1006, December 2010.

[30] R.J. Tibshirani and J. Taylor, "The Solution Path of the Generalized

Lasso," *The Annals of Statistics*, vol. 39, no. 3, pp. 1335–1371, June 2011.
[31] P. Stoica and R. Moses, *Spectral Analysis of Signals*, Prentice Hall, Upper Saddle River, N.J., 2005.

Master's Theses in Mathematical Sciences 2015:E27
ISSN 1404-6342

LUTFMS-3281-2015

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

http://www.maths.lth.se/