



Fast and Statistically Efficient Fundamental Frequency Estimation

Nielsen, Jesper Kjær; Jensen, Tobias Lindstrøm; Jensen, Jesper Rindom; Christensen, Mads Græsbøll; Jensen, Søren Holdt

Published in:
Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on

DOI (link to publication from Publisher):
[10.1109/ICASSP.2016.7471642](https://doi.org/10.1109/ICASSP.2016.7471642)

Publication date:
2016

Document Version
Accepted author manuscript, peer reviewed version

[Link to publication from Aalborg University](#)

Citation for published version (APA):
Nielsen, J. K., Jensen, T. L., Jensen, J. R., Christensen, M. G., & Jensen, S. H. (2016). Fast and Statistically Efficient Fundamental Frequency Estimation. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on* (pp. 86-90). IEEE. <https://doi.org/10.1109/ICASSP.2016.7471642>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

Take down policy

If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

FAST AND STATISTICALLY EFFICIENT FUNDAMENTAL FREQUENCY ESTIMATION

Jesper Kjær Nielsen¹², Tobias Lindstrøm Jensen¹, Jesper Rindom Jensen³, Mads Græsbøll Christensen³,
and Søren Holdt Jensen¹

¹Aalborg University, Denmark
Dept. of Electronic Systems
{jkn,tlj,shj}@es.aau.dk

²Bang & Olufsen A/S
Struer, Denmark

³Aalborg University, Denmark
Audio Analysis Lab, AD:MT
{jrj,mgc}@create.aau.dk

ABSTRACT

Fundamental frequency estimation is a very important task in many applications involving periodic signals. For computational reasons, fast autocorrelation-based estimation methods are often used despite parametric estimation methods having superior estimation accuracy. However, these parametric methods are much more costly to run. In this paper, we propose an algorithm which significantly reduces the computational cost of an accurate maximum likelihood-based estimator for real-valued data. The computational cost is reduced by exploiting the matrix structure of the problem and by using a recursive solver. Via benchmarks, we demonstrate that the computation time is reduced by approximately two orders of magnitude. The proposed fast algorithm is available for download online.

Index Terms— Fundamental frequency estimation, Toeplitz-plus-Hankel solver, fast algorithm

1. INTRODUCTION

Many processes are cyclical in nature and, therefore, generate periodic signals. Examples of such processes are encountered in a large variety of application such as music processing [1, 2], speech processing [3, 4], sonar [5], order analysis [6], and electrocardiography (ECG) [7, 8]. An analysis of periodic signals often involves the estimation of the fundamental frequency which describes the lowest rate at which the process repeats itself. Consequently, many fundamental frequency estimators have been suggested in the scientific literature, and these can, rather coarsely, be dichotomised into two groups of methods. In one group, the methods are non-parametric and based on the autocorrelation function [9]. These methods are typically computationally cheap, but have a suboptimal estimation accuracy and are not robust to noise. In the second group, the methods are based on a signal model whose parameters are typically estimated based on some statistical criterion [2]. They have a good estimation accuracy, but are in general computationally expensive. The maximum likelihood (ML) estimator [10, 11] is a good example of this since it is optimal from an estimation theoretic perspective, but very costly to compute. Primarily for this reason, the scientific literature on speech and audio processing mainly employs the methods from the first group. In this paper, however, we reduce the computational cost of the ML estimator significantly, thus making it a viable alternative to the autocorrelation-based methods.

The work by J.K. Nielsen was supported by the Danish Innovations-Fonden. The work by T.L. Jensen and J.R. Jensen was partly supported by the Independent Research Council for Technology and Production grant no. 4005-00122 and 1337-00084, respectively. The work by M.G. Christensen was supported by the Villum Foundation

The signal model for a uniformly sampled and periodic signal in additive, white Gaussian noise $e(n)$ is given by

$$x(n) = \sum_{i=1}^l [a_i \cos(i\omega_0 n) - b_i \sin(i\omega_0 n)] + e(n) \quad (1)$$

where a_i and b_i are the linear weights of the i 'th harmonic component and ω_0 is the fundamental frequency in radians per sample. If an N -dimensional data set $\{x(n)\}_{n=n_0}^{n_0+N-1}$ is observed, the signal model can be written in a vector form as

$$\mathbf{x} = \mathbf{Z}_l(\omega_0)\boldsymbol{\alpha}_l + \mathbf{e} \quad (2)$$

where we have defined

$$\mathbf{x} = [x(n_0) \ \cdots \ x(n_0 + N - 1)]^T \quad (3)$$

$$\mathbf{e} = [e(n_0) \ \cdots \ e(n_0 + N - 1)]^T \quad (4)$$

$$\mathbf{Z}_l(\omega) = [\mathbf{C}_l(\omega) \ \mathbf{S}_l(\omega)] \quad (5)$$

$$\mathbf{C}_l(\omega) = [\mathbf{c}(\omega) \ \mathbf{c}(2\omega) \ \cdots \ \mathbf{c}(l\omega)] \quad (6)$$

$$\mathbf{S}_l(\omega) = [\mathbf{s}(\omega) \ \mathbf{s}(2\omega) \ \cdots \ \mathbf{s}(l\omega)] \quad (7)$$

$$\mathbf{c}(\omega) = [\cos(\omega n_0) \ \cdots \ \cos(\omega(n_0 + N - 1))]^T \quad (8)$$

$$\mathbf{s}(\omega) = [\sin(\omega n_0) \ \cdots \ \sin(\omega(n_0 + N - 1))]^T \quad (9)$$

$$\boldsymbol{\alpha}_l = [\mathbf{a}_l^T \ -\mathbf{b}_l^T]^T \quad (10)$$

$$\mathbf{a}_l = [a_1 \ \cdots \ a_l]^T \quad (11)$$

$$\mathbf{b}_l = [b_1 \ \cdots \ b_l]^T. \quad (12)$$

The scalars n_0 and l are the start index and the unknown model order, respectively. The former is going to play an important role later. For the model in (2), the ML estimate can be derived as in [12] to

$$\hat{\omega}_0 = \underset{\omega_0 \in \Omega_l}{\operatorname{argmax}} J_{\text{NLS}}(\omega_0, l) \quad (13)$$

where Ω_l is a subset on $(0, \pi/l)$ and

$$J_{\text{NLS}}(\omega_0, l) = \mathbf{x}^T \mathbf{Z}_l(\omega_0) \left[\mathbf{Z}_l^T(\omega_0) \mathbf{Z}_l(\omega_0) \right]^{-1} \mathbf{Z}_l^T(\omega_0) \mathbf{x} \quad (14)$$

is the non-linear least squares (NLS) cost function for a model order of l . We have here opted for the term NLS since this term is conventionally used for the cost function in (14). Note, however, that the NLS and the ML estimators are identical for the model in (2). Unfortunately, the NLS cost function has numerous sharp peaks so gradient based methods work very poorly for finding the maximiser. Instead, the cost function is often evaluated on a grid followed by a line search to refine the estimate [13]. When the model order is

Algorithm 1 The standard algorithm for computing the NLS cost function matrix $[J]_{l,f} = J_{\text{NLS}}(\omega_0, l)$ where the row indices are the L model orders and the column indices are the F fundamental frequencies on the Fourier grid. The notation \odot and $[\cdot]_{i,k}$ denotes element-wise multiplication and element (i, k) , respectively. Note that the vector function $\mathbf{m}_l(\omega_0, \mathbf{f})$ is a selection function that forms the vector $\mathbf{w}_l(\omega_0)$ as described in Sec. 2.2.1.

```

1:  $\mathbf{f} = \text{fft}(\mathbf{x})$   $\triangleright \mathcal{O}(F \log F)$ 
2:  $[J]_{1,1:F} = 2N^{-1}(\mathbf{f}^* \odot \mathbf{f})^T$   $\triangleright \mathcal{O}(F)$ 
3: for  $l \in \{2, 3, \dots, L\}$  do
4:   for  $f \in \{2, 3, \dots, \lceil F/(2l) \rceil\}$  do
5:      $\omega_0 = 2\pi(f-1)/F$   $\triangleright \mathcal{O}(1)$ 
6:      $\mathbf{w}_l(\omega_0) = \mathbf{m}_l(\omega_0, \mathbf{f})$   $\triangleright \mathcal{O}(1)$ 
7:     Solve  $\mathbf{Z}_l^T(\omega_0)\mathbf{Z}_l(\omega_0)\boldsymbol{\alpha}_l = \mathbf{w}_l(\omega_0)$  for  $\boldsymbol{\alpha}_l$   $\triangleright \mathcal{O}(l^3)$ 
8:      $[J]_{l,f} = \mathbf{w}_l^T(\omega_0)\boldsymbol{\alpha}_l$   $\triangleright \mathcal{O}(l)$ 
9:   end for
10: end for

```

unknown, we have to compute an estimate for all candidate model orders before we can use model detection criteria such as AIC, MDL, and BIC [14–18] to determine the fundamental frequency estimate of the most likely model. Thus, if the set of candidate model orders is $\{1, \dots, L\}$, L NLS cost functions of the form in (14) should be computed on a grid. In this paper, we propose an algorithm for speeding up these computationally costly grid-based evaluations.

In (2), we have used the real-valued signal model for a periodic signal. Although most periodic signals are real-valued, a complex-valued representation is often used instead for analytical and computational reasons. We recently demonstrated the latter point in [19] where we reduced the computational complexity of the ML estimator for the complex-valued signal model corresponding to (2). This was achieved by exploiting the Toeplitz structure of the complex analogue to $\mathbf{Z}_l^T(\omega_0)\mathbf{Z}_l(\omega_0)$. In the real-valued case, however, $\mathbf{Z}_l^T(\omega_0)\mathbf{Z}_l(\omega_0)$ has a much more complicated block Toeplitz-plus-Hankel structure so the proposed algorithm in [19] cannot directly be applied to the real-valued case. Instead, we reformulate the problem so that an efficient and recursive Toeplitz-plus-Hankel solver can be used. This reduces the computational complexity of computing the fundamental frequency estimate on an F -point uniform grid for all model orders $l \in \{1, 2, \dots, L\}$ from $\mathcal{O}(F \log F) + \mathcal{O}(FL^3)$ (see Algorithm 1) to just $\mathcal{O}(F \log F) + \mathcal{O}(FL)$. The latter order of complexity is the same as that of the popular harmonic summation (HS) method [20, 21] which can be interpreted as an approximation to the ML estimator (see more in Sec. 2.1).

2. FAST EVALUATION OF THE NLS COST FUNCTION

In this section, we derive the algorithm for significantly reducing the cost of computing the NLS cost function on an F -point uniform grid for all model orders $l \in \{1, 2, \dots, L\}$. This step is the most costly step in most NLS- and MUSIC-based fundamental frequency estimators such as the recently proposed Bayesian estimator in [13]. First, however, we briefly outline the standard algorithm for making the above computations and outline how the harmonic summation method reduces the cost significantly by making a simple approximation.

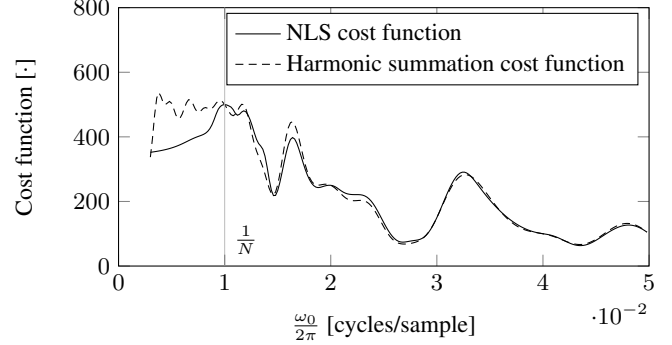


Fig. 1. Example of the exact NLS and HS cost functions. $N = 100$, $L = 10$, $\omega_0 = 2\pi/N$, and constant amplitude $\sqrt{a_i^2 + b_i^2} = 1$, $i \in \{1, \dots, L\}$.

2.1. Standard Algorithm and Harmonic Summation

The standard algorithm for computing the NLS cost function on an F -point uniform grid for all model orders $l \in \{1, 2, \dots, L\}$ is outlined in Algorithm 1. Unless L is very small (relative to N), the main cost is to solve the problem in line 7. The total cost of running Algorithm 1 is $\mathcal{O}(F \log F) + \mathcal{O}(FL^3)$. In HS, line 7 is replaced by $(N/2)\boldsymbol{\alpha}_l = \mathbf{w}_l$ which is justified by the asymptotic result that $\lim_{N \rightarrow \infty} 2N^{-1}\mathbf{Z}_l^T(\omega_0)\mathbf{Z}_l(\omega_0) = \mathbf{I}_{2l}$ where \mathbf{I}_{2l} is the $2l \times 2l$ identity matrix. This reduces the cost of solving the problem in line 7 from $\mathcal{O}(l^3)$ to $\mathcal{O}(l)$. Thus, the total cost of HS is $\mathcal{O}(F \log F) + \mathcal{O}(FL)$. In Fig. 1, examples of the NLS and HS cost functions are shown. Clearly, HS is inaccurate when the fundamental frequency is low (see much more on this in [12]), but this is currently the price to pay to reduce the computational complexity. In the next section, however, we show how the NLS cost function can be computed with the same order of complexity as HS. That is, we derive an algorithm for solving the problem in line 7 of Algorithm 1 in linear complexity.

2.2. A Fast Algorithm

The proposed fast algorithm is based on five key observations which are described in details in the next five sections. To keep the notation simpler, we define $\mathbf{A}_l(\omega_0) = \mathbf{Z}_l^T(\omega_0)\mathbf{Z}_l(\omega_0)$.

2.2.1. Computing $\mathbf{w}_l(\omega_0)$

The vector $\mathbf{w}_l(\omega_0)$ is given by

$$\mathbf{w}_l(\omega_0) = \mathbf{Z}_l^T(\omega_0)\mathbf{x} = \mathbf{C}_l^T(\omega_0)\mathbf{x} + \mathbf{S}_l^T(\omega_0)\mathbf{x} \quad (15)$$

and can be computed efficiently from \mathbf{f} in line 1 of Algorithm 1 by extracting the appropriate elements from the vector \mathbf{f} . Specifically from (5)–(9), the elements of $\mathbf{w}_l(\omega_0)$ can all be computed from

$$\mathbf{c}^T(\omega_0 l)\mathbf{x} = \Re \left[\exp(-j\omega_0 l n_0) [\mathbf{f}]_{l(f-1)+1} \right] \quad (16)$$

$$\mathbf{s}^T(\omega_0 l)\mathbf{x} = -\Im \left[\exp(-j\omega_0 l n_0) [\mathbf{f}]_{l(f-1)+1} \right] \quad (17)$$

where ω_0 and the integer f are related through $\omega_0 = 2\pi(f-1)/F$.

2.2.2. The structure of $\mathbf{A}_l(\omega_0)$

The matrix $\mathbf{A}_l(\omega_0)$ has a block Toeplitz-plus-Hankel like structure given by

$$\mathbf{A}_l(\omega_0) = \begin{bmatrix} \mathbf{T}_l(\omega_0) & -\tilde{\mathbf{T}}_l(\omega_0) \\ \tilde{\mathbf{T}}_l(\omega_0) & \mathbf{T}_l(\omega_0) \end{bmatrix} + \begin{bmatrix} \mathbf{H}_l(\omega_0) & \tilde{\mathbf{H}}_l(\omega_0) \\ \tilde{\mathbf{H}}_l(\omega_0) & -\mathbf{H}_l(\omega_0) \end{bmatrix} \quad (18)$$

where $\mathbf{T}_l(\omega_0) = \{t_{i-k}(\omega_0)\}_{i,k=1}^l$, $\mathbf{H}_l(\omega_0) = \{h_{i+k}(\omega_0)\}_{i,k=1}^l$, $\tilde{\mathbf{T}}_l(\omega_0) = \{\tilde{t}_{i-k}(\omega_0)\}_{i,k=1}^l$, $\tilde{\mathbf{H}}_l(\omega_0) = \{\tilde{h}_{i+k}(\omega_0)\}_{i,k=1}^l$ and the elements of the Toeplitz and Hankel matrices are for $i, k \in \{1, \dots, l\}$ given by

$$t_{i-k}(\omega_0) = t_{k-i}(\omega_0) = \begin{cases} \frac{1}{2} \cos(\omega_0(i-k) \left[n_0 + \frac{N-1}{2} \right]) \times \frac{\sin(\omega_0(i-k)N/2)}{\sin(\omega_0(i-k)/2)} & i \neq k \\ N/2 & i = k \end{cases} \quad (19)$$

$$h_{i+k}(\omega_0) = \frac{1}{2} \cos\left(\omega_0(i+k) \left[n_0 + \frac{N-1}{2} \right]\right) \times \frac{\sin(\omega_0(i+k)N/2)}{\sin(\omega_0(i+k)/2)} \quad (20)$$

$$\tilde{t}_{i-k}(\omega_0) = -\tilde{t}_{k-i}(\omega_0) = \begin{cases} \frac{1}{2} \sin(\omega_0(i-k) \left[n_0 + \frac{N-1}{2} \right]) \times \frac{\sin(\omega_0(i-k)N/2)}{\sin(\omega_0(i-k)/2)} & i \neq k \\ 0 & i = k \end{cases} \quad (21)$$

$$\tilde{h}_{i+k}(\omega_0) = \frac{1}{2} \sin\left(\omega_0(i+k) \left[n_0 + \frac{N-1}{2} \right]\right) \times \frac{\sin(\omega_0(i+k)N/2)}{\sin(\omega_0(i+k)/2)}. \quad (22)$$

2.2.3. Selection of the start index n_0

The cost function $J_{\text{NLS}}(\omega_0, l)$ can be shown not to depend on the start index n_0 . This is a key observation since this allows us to select it as $n_0 = -(N-1)/2$ so that $\tilde{t}_{i-k}(\omega_0) = \tilde{h}_{i+k}(\omega_0) = 0$ for $i, k \in \{1, \dots, l\}$ and any $\omega_0 \in \mathbb{R}$ (see (21) and (22)). This means that we can write the problem in line 7 of Algorithm 1 as two separate Toeplitz-plus-Hankel systems as

$$(\mathbf{T}_l(\omega_0) + \mathbf{H}_l(\omega_0))\mathbf{a}_l(\omega_0) = \tilde{\mathbf{w}}_l(\omega_0) \quad (23)$$

$$(\mathbf{T}_l(\omega_0) - \mathbf{H}_l(\omega_0))\mathbf{b}_l(\omega_0) = -\tilde{\mathbf{w}}_l(\omega_0) \quad (24)$$

where $\mathbf{w}_l(\omega_0) = [\tilde{\mathbf{w}}_l^T(\omega_0) \quad \tilde{\mathbf{w}}_l^T(\omega_0)]^T$ is partitioned as in (18).

2.2.4. Solving the two linear systems for a given model order

The two linear systems in (23) and (24) can be solved using a Toeplitz-plus-Hankel solver such as those suggested in [22, 23]. The complexity of fast Toeplitz-plus-Hankel solvers is $\mathcal{O}(l^2)$ for a model order of l . Problems involving a Toeplitz-plus-Hankel structure can be interpreted as a special case of matrix problems with displacement structures (see, e.g., the overview work [24]).

2.2.5. Solving the two linear systems for all model orders

The two linear systems in (23) and (24) do not only have a Toeplitz-plus-Hankel structure, but any upper-left subsystem corresponds to

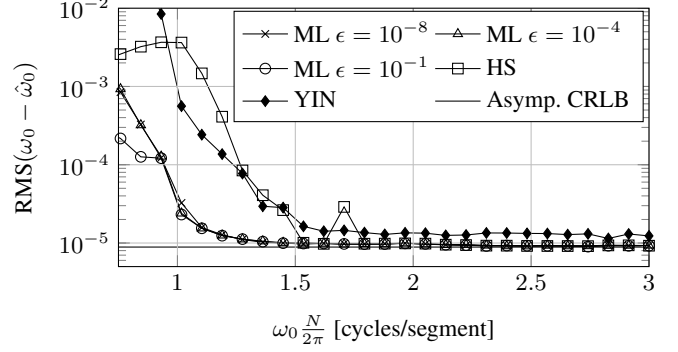


Fig. 2. The estimation accuracy as a function of frequency for $N = 200$, $l = 10$, and an SNR of 15 dB.

a linear system of a lower order. That is, if we solve the two linear systems for the maximum model order L using a recursive solver as the one in [22], we get the solutions to all the systems of size $l = 1, \dots, L-1$ for free in the process. The complexity of solving the two systems (23) and (24) for a model order of $l+1$ is, therefore, only $\mathcal{O}(l)$ if we already have the solution to the two systems for l .

Based on these key observations, the time complexity of solving the problem in line 7 of Algorithm 1 can be reduced from $\mathcal{O}(l^3)$ to just $\mathcal{O}(l)$ which is the same order of complexity as that of HS. However, note that no approximations are involved in solving the problem. In addition to the observations above, we also note the following about the algorithm.

- To further speed up the algorithm, it can be vectorised over the frequencies, and this is also done in our MATLAB implementation.
- When the fundamental frequency is very low, the columns of $\mathbf{Z}_l(\omega_0)$ are nearly linearly dependent [12]. The matrix $\mathbf{A}_l(\omega_0)$ is therefore ill-conditioned and may give numerical problems. However, $\mathbf{A}_l(\omega_0)$ can easily be regularised by considering $\mathbf{A}_l(\omega_0) + \epsilon \mathbf{I}_{2l}$ instead where ϵ is a small positive constant. Since $\epsilon \mathbf{I}_{2l}$ is Toeplitz and diagonal, the regularisation can easily be included in the fast algorithm by adding ϵ to $t_0(\omega_0)$. Note also that the regularised algorithm can be viewed as an intermediate solution between the NLS method ($\epsilon = 0$) and the harmonic summation method ($\epsilon = \infty$).

3. SIMULATIONS

In this section, two things are demonstrated. First, we demonstrate what Fig. 1 suggests; that the ML estimator has a superior estimation accuracy compared to HS for low fundamental frequencies. For comparison, we also include the YIN method [25] which is the most popular and most cited autocorrelation-based estimator. Second, and most importantly, we show that the proposed fast algorithm reduces the computational cost significantly of computing the NLS cost function on a uniform F -point grid for all model orders up to L .

Fig. 2 and Fig. 3 show the estimation accuracy of the ML estimator, HS, and YIN as a function of the frequency for an SNR of 15 dB and 0 dB, respectively. The estimation accuracy was assessed via a Monte Carlo simulation consisting of 10,000 runs for each fundamental frequency point. The data were generated from the model in (2), and the grid-based fundamental frequency estimate was refined using a line search [26, Ch. 4]. In the simulations, the

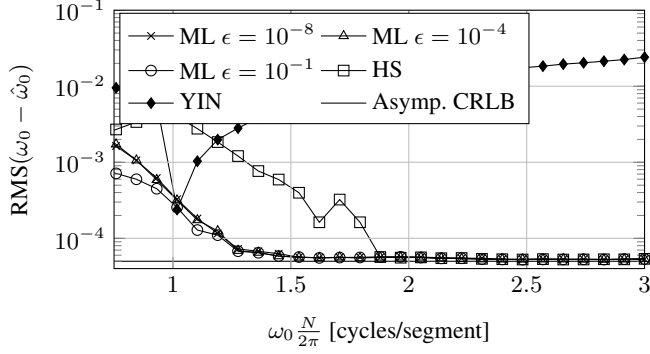


Fig. 3. The estimation accuracy as a function of frequency for $N = 200$, $l = 10$, and an SNR of 0 dB.

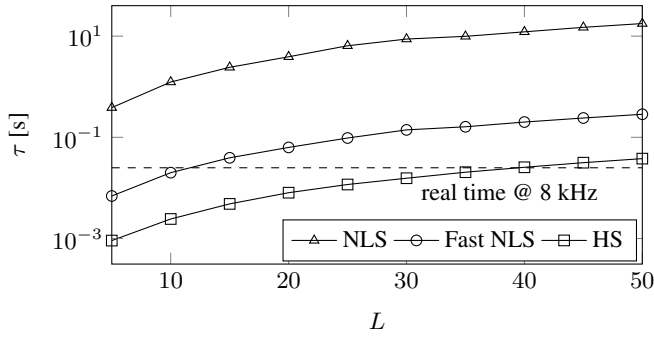


Fig. 4. The computation time for three different ways of computing either the exact or the approximate NLS cost function for $N = 200$ and $F = 5NL$.

fundamental frequency was kept low relative to N and this also explains why the unit on the x -axis is cycles/segment (of data). Several interesting things can be seen from the two figures. First, YIN does not work for low fundamental frequencies and in noisy environments, and it has a suboptimal performance for higher frequencies even when the noise-level is low. On the other hand, HS attains the Cramér-Rao lower bound (CRLB) and is an efficient estimator when the fundamental cycle is not lower than 1.75 cycles/segment in this simulation setup. For low fundamental frequencies, the ML estimator clearly outperforms HS and YIN. Note that the asymptotic CRLB is way too optimistic for low fundamental frequencies (see also [12]). The accuracy of the ML estimator has been plotted for three different values of the regularisation parameter. There is not any major performance difference, but we have noted that the NLS cost function contains spurious peaks at low frequencies, caused by numerical problems, when no regularisation is used.

Fig. 4 and Fig. 5 show the computation times of computing the cost functions on a uniform F -point grid for all model orders up to L as a function of the number of data points N and the maximum model order L , respectively. In addition to the computation times, the real-time limit is also marked as a dashed line for a sampling frequency of 8 kHz. The timing benchmarks were performed on MATLAB implementations of the various algorithms, and these MATLAB implementations as well as the rest of the simulation code is available from <http://kom.aau.dk/~jkn/publications/publications.php>. The timings were obtained by executing the algorithms 10^5 times using the

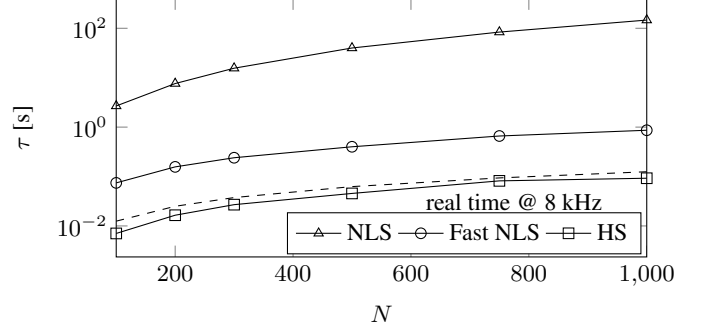


Fig. 5. The computation time for three different ways of computing either the exact or the approximate NLS cost function for $L = 30$ and $F = 5NL$.

smallest $i \in \{0, 1, 2, \dots\}$ such that the execution time $\tau_0 \geq 0.2$ s. For this i , three repetitions of 10^i executions were then run and timed, producing the three repetition times τ_1 , τ_2 and τ_3 . The reported execution times in Fig. 4 and Fig. 5 were then calculated as $\tau = \min(\tau_1, \tau_2, \tau_3)/10^i$. This procedure is the same as the default settings in Python's `timeit` module [27]. All timings were executed on an Intel(R) Dual Core(TM) i5-2410M CPU at 2.3 GHz with Ubuntu Linux kernel 3.13.0-24-generic and MATLAB 8.4.0. Note that we did not include YIN in these benchmarks since we cannot separate YIN in the same grid and refinement step as the NLS and HS methods can. Moreover, YIN's time complexity is also data dependent since all peaks of the cost function must be interpolated [25].

In Fig. 4, the timings for $N = 200$ data points and a variable model order are shown. The proposed fast algorithm was approximately 55–70 times faster compared to the standard algorithm in Algorithm 1. On the other hand, the fast algorithm was approximately 7–9 times slower than HS in this implementation. Even though HS and the proposed fast algorithm have the same order of complexity, this was expected since the constant for the latter algorithm is higher. In Fig. 5, the timings were reported for a fixed $L = 30$ and a variable number of data points. We observe a similar behaviour to that in Fig. 4. However, the relative speed up of the proposed fast algorithm to the standard algorithm increased as a function of N to approximately 150 at $N = 1000$.

4. CONCLUSION

In this paper, we proposed a fast algorithm for a maximum likelihood-based (ML) fundamental frequency estimator for real-valued data. The algorithm was derived by selecting the start index n_0 in a way so that a block Toeplitz-plus-Hankel system could be written as two separate Toeplitz-plus-Hankel systems. A recursive and fast Toeplitz-plus-Hankel algorithm was then used to compute the non-linear least squares cost function on a uniform F -point grid for all model orders up to L . This reduced the computational complexity from $\mathcal{O}(F \log F) + \mathcal{O}(FL^3)$ (see Algorithm 1) to just $\mathcal{O}(F \log F) + \mathcal{O}(FL)$ where the latter complexity is the same as the popular harmonic summation (HS) method. Via simulations, we first demonstrated that the ML estimator outperformed HS and the autocorrelation-based YIN method for both a low fundamental frequency and in noisy conditions. We also found that the proposed algorithm reduced the computation time of approximately two orders of magnitude for standard sized problems.

5. REFERENCES

- [1] N. H. Fletcher and T. D. Rossing, *The Physics of Musical Instruments*, Springer, 2 edition, Jun. 1998.
- [2] M. G. Christensen and A. Jakobsson, *Multi-Pitch Estimation*, San Rafael, CA, USA: Morgan & Claypool, 2009.
- [3] H. Dudley, “The carrier nature of speech,” *Bell System Technical Journal*, vol. 19, no. 4, pp. 495–515, Oct. 1940.
- [4] R. J. Sluijter, *The Development of Speech Coding and the First Standard Coder for Public Mobile Telephony*, Ph.D. thesis, Technische Universiteit Eindhoven, 2005.
- [5] G. L. Ogden, L. M. Zurk, M. E. Jones, and M. E. Peterson, “Extraction of small boat harmonic signatures from passive sonar,” *J. Acoust. Soc. Am.*, vol. 129, no. 6, pp. 3768–3776, Jun. 2011.
- [6] S. Gade, H. Herlufsen, H. Konstantin-Hansen, and N. J. Wismer, “Order tracking analysis,” Technical review no. 2, Brüel & Kjær A/S, 1995.
- [7] V. K. Murthy, L. J. Haywood, J. Richardson, R. Kalaba, S. Salzberg, G. Harvey, and D. Vereeke, “Analysis of power spectral densities of electrocardiograms,” *Mathematical Biosciences*, vol. 12, no. 1–2, pp. 41–51, Oct. 1971.
- [8] T. Schäck, C. Sledz, M. Muma, and A. M. Zoubir, “A new method for heart rate monitoring during physical exercise using photoplethysmographic signals,” in *Proc. European Signal Processing Conf.*, 2015.
- [9] L. R. Rabiner, “On the use of autocorrelation analysis for pitch detection,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 25, no. 1, pp. 24–33, Feb. 1977.
- [10] B. G. Quinn and P. J. Thomson, “Estimating the frequency of a periodic function,” *Biometrika*, vol. 78, no. 1, pp. 65–74, Mar. 1991.
- [11] M. G. Christensen, P. Stoica, A. Jakobsson, and S. H. Jensen, “Multi-pitch estimation,” *Signal Processing*, vol. 88, no. 4, pp. 972–983, Apr. 2008.
- [12] M. G. Christensen, “Accurate estimation of low fundamental frequencies from real-valued measurements,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 10, pp. 2042–2056, 2013.
- [13] J. K. Nielsen, M. G. Christensen, and S. H. Jensen, “Default Bayesian estimation of the fundamental frequency,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 3, pp. 598–610, Mar. 2013.
- [14] H. Akaike, “A new look at the statistical model identification,” *IEEE Trans. Autom. Control*, vol. 19, no. 6, pp. 716–723, Dec. 1974.
- [15] G. Schwarz, “Estimating the dimension of a model,” *Ann. Stat.*, vol. 6, no. 2, pp. 461–464, Mar. 1978.
- [16] J. Rissanen, “Modeling by shortest data description,” *Automatica*, vol. 14, no. 5, pp. 465–471, Sep. 1978.
- [17] P. M. Djuric, “Asymptotic MAP criteria for model selection,” *IEEE Trans. Signal Process.*, vol. 46, no. 10, pp. 2726–2735, Oct. 1998.
- [18] J. K. Nielsen, M. G. Christensen, A. T. Cemgil, and S. H. Jensen, “Bayesian model comparison with the g-prior,” *IEEE Trans. Signal Process.*, vol. 62, no. 1, pp. 225–238, 2014.
- [19] J. K. Nielsen, T. L. Jensen, J. R. Jensen, M. G. Christensen, and S. H. Jensen, “A fast algorithm for maximum likelihood-based fundamental frequency estimation,” in *Proc. European Signal Processing Conf.*, 2015.
- [20] A. M. Noll, “Pitch determination of human speech by the harmonic product spectrum, the harmonic sum spectrum, and a maximum likelihood estimate,” in *Proc. of the symposium on computer process. commun.*, 1969, vol. 779.
- [21] D. J. Hermes, “Measurement of pitch by subharmonic summation,” *J. Acoust. Soc. Am.*, vol. 83, no. 1, pp. 257–264, 1988.
- [22] I. Gohberg and I. Koltracht, “Efficient algorithm for Toeplitz plus Hankel matrices,” *Integral Equations Operator Theory*, vol. 12, no. 1, pp. 136–142, 1989.
- [23] G. Heinig and K. Rost, “Fast algorithms for Toeplitz and Hankel matrices,” *Linear Algebra and its Applications*, vol. 435, no. 1, pp. 1–59, 2011.
- [24] T. Kailath and A. H. Sayed, “Displacement structure: Theory and applications,” *SIAM Review*, vol. 37, no. 3, pp. 297–386, 1995.
- [25] A. de Cheveigné and H. Kawahara, “YIN, a fundamental frequency estimator for speech and music,” *J. Acoust. Soc. Am.*, vol. 111, no. 4, pp. 1917–1930, Apr. 2002.
- [26] Andreas Antoniou and Wu-Sheng Lu, *Practical Optimization: Algorithms and Engineering Applications*, Springer, Mar. 2007.
- [27] G. van Rossum and the Python development team, *The Python Library Reference*, Python Software Foundation, 2016, v. 2.7.11, <https://docs.python.org/2/index.html>.