FAST AND STATISTICALLY EFFICIENT FUNDAMENTAL FREQUENCY ESTIMATION

Jesper Kjær Nielsen\textsuperscript{1,2}, Tobias Lindstrøm Jensen\textsuperscript{1}, Jesper Rindom Jensen\textsuperscript{3}, Mads Græsbøll Christensen\textsuperscript{3}, and Søren Holdt Jensen\textsuperscript{1}

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

\textsuperscript{2}Bang & Olufsen A/S
Struer, Denmark

\textsuperscript{3}Aalborg 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 of occurrence. 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].

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

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

where \( a_i \) and \( b_i \) are the linear weights of the \( i \)’th harmonic component and \( \omega_0 \) is the fundamental frequency in radians per sample. If an \( N \)-dimensional data set \( \{x(n)\}_{n=0}^{N-1} \) is observed, the signal model can be written in a vector form as

\[
x = Z_1(\omega_0) \alpha + e
\]

where we have defined

\[
x = [x(n_0) \cdots x(n_0 + N - 1)]^T
\]

\[
e = [e(n_0) \cdots e(n_0 + N - 1)]^T
\]

\[
Z_1(\omega) = [C_1(\omega) \ S_1(\omega)]
\]

\[
C_1(\omega) = [c(\omega) \ c(2\omega) \cdots c(\omega)]
\]

\[
S_1(\omega) = [s(\omega) \ s(2\omega) \cdots s(\omega)]
\]

\[
c(\omega) = [\cos(\omega n_0) \cdots \cos(\omega(n_0 + N - 1))]^T
\]

\[
s(\omega) = [\sin(\omega n_0) \cdots \sin(\omega(n_0 + N - 1))]^T
\]

\[
\alpha = [\alpha_1^T \ -b_1^T]^T
\]

\[
a_i = [a_1 \cdots a_l]^T
\]

\[
b_i = [b_1 \cdots b_l]^T
\]

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

\[
\hat{\omega}_0 = \arg \max_{\omega_0 \in \Omega_l} J_{NLS}(\omega_0, l)
\]

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

\[
J_{NLS}(\omega_0, l) = x^T Z_1(\omega_0) \left[ Z_1^T(\omega_0) Z_1(\omega_0) \right]^{-1} Z_1^T(\omega_0) x
\]

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

The work by J.K. Nielsen was supported by the Danish InnovationsFonden. 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 1337-00064, respectively. The work by M.G. Christensen was supported by the Villum Foundation.
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, 2, \ldots, 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 $Z(\omega_0)Z(\omega_0)\alpha_t = \mathbf{w}(\omega_0)$ for $\alpha_t$ $\mathcal{O}(l^3)$.

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 $l \in \{1, 2, \ldots, L\}$. This 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.

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 $[\cdot]_{l,k}$ denotes element-wise multiplication and element $(l, k)$, respectively. Note that the vector function $\mathbf{m}_l(\omega_0, f)$ is a selection function that forms the vector $\mathbf{w}_l(\omega_0)$ as described in Sec. 2.2.1.

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

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 $l \in \{1, 2, \ldots, L\}$ 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 $N/2\alpha_t = \mathbf{w}$ which is justified by the asymptotic result that $\lim_{N \to \infty} 2N^{-1}Z(\omega_0)Z(\omega_0) = I_{2l}$ where $I_{2l}$ 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 we 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 $A_l(\omega_0) = Z(l)^T(\omega_0)Z(l)(\omega_0)$.

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

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

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

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

$$e^T(l\omega_0)x = \Re \left[ \exp(-j\omega_0ln_0(f)(f-1)+1) \right]$$ (16)
$$s^T(l\omega_0)x = -\Im \left[ \exp(-j\omega_0ln_0(f)(f-1)+1) \right]$$ (17)

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

![Fig. 1. Example of the exact NLS and HS cost functions. $N = 100$, $L = 10$, $\omega_0 = 2\pi/N$, and constant amplitude $\sqrt{a^2 + b^2} = 1, i \in \{1, \ldots, L\}$.](image)
2.2.2. The structure of \( A_l(\omega_0) \)

The matrix \( A_l(\omega_0) \) has a block Toeplitz-plus-Hankel like structure given by

\[
A_l(\omega_0) =
\begin{bmatrix}
T_l(\omega_0) & -T_l(\omega_0) \\
\bar{T}_l(\omega_0) & T_l(\omega_0)
\end{bmatrix}
+ \begin{bmatrix}
H_l(\omega_0) & \bar{H}_l(\omega_0) \\
\bar{H}_l(\omega_0) & H_l(\omega_0)
\end{bmatrix}
\]

(18)

where \( T_l(\omega_0) = \{ t_{i+k}(\omega_0) \}_{i,k=1}^l \), \( H_l(\omega_0) = \{ h_{i+k}(\omega_0) \}_{i,k=1}^l \), \( \bar{T}_l(\omega_0) = \{ \bar{t}_{i+k}(\omega_0) \}_{i,k=1}^l \), \( \bar{H}_l(\omega_0) = \{ \bar{h}_{i+k}(\omega_0) \}_{i,k=1}^l \) and the elements of the Toeplitz and Hankel matrices are for \( i,k \in \{ 1, \ldots, l \} \) given by

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

(19)

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

(20)

\[
\bar{t}_{i-k}(\omega_0) = -\bar{t}_{k-i}(\omega_0) =
\begin{cases}
\frac{1}{2} \cos \left( \frac{\sin(\omega_0(i-k))}{\sin(\omega_0(i-k)/2)} \frac{n_0 + \frac{N-1}{2}}{N/2} \right) & i \neq k \\
\frac{1}{2} \sin \left( \frac{\sin(\omega_0(i-k))}{\sin(\omega_0(i-k)/2)} \frac{n_0 + \frac{N-1}{2}}{N/2} \right) & i = k
\end{cases}
\]

(21)

\[
\bar{h}_{i+k}(\omega_0) =
\begin{cases}
\frac{1}{2} \cos \left( \frac{\sin(\omega_0(i+k))}{\sin(\omega_0(i+k)/2)} \frac{n_0 + \frac{N-1}{2}}{N/2} \right) & i \neq k \\
\frac{1}{2} \sin \left( \frac{\sin(\omega_0(i+k))}{\sin(\omega_0(i+k)/2)} \frac{n_0 + \frac{N-1}{2}}{N/2} \right) & i = k
\end{cases}
\]

(22)

2.2.3. Selection of the start index \( n_0 \)

The cost function \( J_{\text{RMS}}(\omega_0,l) \) 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 \( t_{i-k}(\omega_0) = h_{i+k}(\omega_0) = 0 \) for \( i,k \in \{ 1, \ldots, l \} \) 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

\[
(T_l(\omega_0) + H_l(\omega_0)) \mathbf{a}_l(\omega_0) = \mathbf{w}_l(\omega_0)
\]

(23)

\[
(T_l(\omega_0) - H_l(\omega_0)) \mathbf{b}_l(\omega_0) = \mathbf{-w}_l(\omega_0)
\]

(24)

where \( \mathbf{w}_l(\omega_0) = [\mathbf{w}_l^T(\omega_0) \quad \mathbf{\bar{w}}_l^T(\omega_0)]^T \) 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 \( O(l^2) \) 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 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 \( l = 1, \ldots, L - 1 \) 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 \( O(l) \) 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 \( O(l^3) \) to just \( O(l) \) 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 \( A_l(\omega_0) \) is therefore ill-conditioned and may give numerical problems. However, \( A_l(\omega_0) \) can easily be regularised by considering \( A_l(\omega_0) + \epsilon I_2 \) instead where \( \epsilon \) is a small positive constant. Since \( \epsilon I_2 \) is Toeplitz and diagonal, the regularisation can easily be included in the fast algorithm by adding \( \epsilon \) to \( t_{0,l}(\omega_0) \). 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. 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
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 he fundamental cycle is not lower than 1.75 cycles/segment.

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 nonlinear least squares cost function on a uniform grid for all model orders up to \( L \), 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 \( 10^4 \) times using 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 nonlinear least squares cost function on a uniform grid for all model orders up to \( L \). This reduced the computational complexity from \( O(F \log F) + O(F L) \) (see Algorithm 1) to just \( O(F \log F) + O(F L) \) 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


