Fast and Statistically Efficient
Fundamental Frequency Estimation
ICASSP 2016

March 23, 2016

J. K. Nielsen\textsuperscript{1,2}, T. L. Jensen\textsuperscript{1}, J. R. Jensen\textsuperscript{3},
M. G. Christensen\textsuperscript{3}, and S. H. Jensen\textsuperscript{1}

\{jkn,tlj,shj\}@es.aau.dk, \{jrj,mgc\}@create.aau.dk

\textsuperscript{2}Aalborg University
Dept. of Electronic Systems

\textsuperscript{3}Bang & Olufsen A/S
Denmark

\textsuperscript{1}Aalborg University
Audio Analysis Lab, AD:MT

Supported by the Danish Council for Independent Research, the InnovationsFonden, and the Villum Foundation.
Agenda

Fundamental Frequency Estimation
   Maximum Likelihood Estimation

Fast Maximum Likelihood Estimation

Results

Summary
Speech Signal Example

\[ x(t) = h_1(t) + h_2(t) + h_3(t) + e(t) \]
Speech Signal Example

\[ x(t) = h_1(t) + h_2(t) + h_3(t) + e(t) \]
Speech Signal Example

\[ x(t) = h_1(t) + h_2(t) + h_3(t) + e(t) \]
Mathematical Model

$x(t) = \sum_{i=1}^{l} h_i(t) + e(t) = \sum_{i=1}^{l} A_i \cos(i2\pi f_0 t + \phi_i) + e(t)$

where

- $A_i$ real amplitude of the $i$th harmonic
- $\phi_i$ phase of the $i$th harmonic
- $f_0$ fundamental frequency in Hz
- $l$ the number of harmonics/model order
- $e(t)$ white Gaussian noise with variance $\sigma^2$
Mathematical Model

\[ x(t) = \sum_{i=1}^{l} h_i(t) + e(t) = \sum_{i=1}^{l} A_i \cos(i2\pi f_0 t + \phi_i) + e(t) \]  

(1)

where

- \( A_i \) real amplitude of the \( i \)th harmonic
- \( \phi_i \) phase of the \( i \)th harmonic
- \( f_0 \) fundamental frequency in Hz
- \( l \) the number of harmonics/model order
- \( e(t) \) white Gaussian noise with variance \( \sigma^2 \)

Analysis problem: Get \( f_0 \) and \( l \) from the data.
Estimation Methods

Correlation Methods
A periodic signal satisfies that

\[ x(t) = x(t + T) \]  

(2)

where \( T = 1/f_0 \) is the period. Thus, the autocorrelation function of \( x(t) \) has a peak for a lag of \( T \).
Estimation Methods

Correlation Methods
A periodic signal satisfies that

$$x(t) = x(t + T)$$  \hspace{1cm} (2)

where $T = 1/f_0$ is the period. Thus, the autocorrelation function of $x(t)$ has a peak for a lag of $T$.

+ Intuitive and simple
+ Low computational complexity
+ No need to estimate the model order
- Fail for low fundamental frequencies
- Are very sensitive to noise

Correlation methods such as YIN and RAPT are very popular.
Estimation Methods

Correlation Methods
A periodic signal satisfies that

\[ x(t) = x(t + T) \]  

where \( T = 1/f_0 \) is the period. Thus, the autocorrelation function of \( x(t) \) has a peak for a lag of \( T \).

+ Intuitive and simple
+ Low computational complexity
+ No need to estimate the model order
- Fail for low fundamental frequencies
- Are very sensitive to noise

Correlation methods such as YIN and RAPT are very popular.
Estimation Methods

Parametric Methods
Estimate the parameters in

\[
x(t) = \sum_{i=1}^{l} A_i \cos(i2\pi f_0 t + \phi_i) + e(t)
\]

\[
= \sum_{i=1}^{l} \left[ a_i \cos(i2\pi f_0 t) - b_i \sin(i2\pi f_0 t) \right] + e(t)
\]
Estimation Methods

Parametric Methods

Estimate the parameters in

\[ x(t) = \sum_{i=1}^{l} A_i \cos(i2\pi f_0 t + \phi_i) + e(t) \]  

\[ = \sum_{i=1}^{l} \left[ a_i \cos(i2\pi f_0 t) - b_i \sin(i2\pi f_0 t) \right] + e(t) \]

+ High estimation accuracy
+ Work very well in even noisy conditions
+ Work for low fundamental frequencies
- The model order has to be estimated
- High computational complexity
Estimation Methods

Parametric Methods
Estimate the parameters in

\[ x(t) = \sum_{i=1}^{l} A_i \cos(i2\pi f_0 t + \phi_i) + e(t) \]  \hspace{1cm} (3)

\[ = \sum_{i=1}^{l} \left[ a_i \cos(i2\pi f_0 t) - b_i \sin(i2\pi f_0 t) \right] + e(t) \]  \hspace{1cm} (4)

+ High estimation accuracy
+ Work very well in even noisy conditions
+ Work for low fundamental frequencies
- The model order has to be estimated
- High computational complexity (for MLE/NLS)
Periodic Signals
Vector Signal Model

The sampled signal model

\[ x(n) = \sum_{i=1}^{l} \left[ a_i \cos(i\omega_0 n) - b_i \sin(i\omega_0 n) \right] + e(n) \quad (5) \]

for \( n = n_0, n_0 + 1, \ldots, n_0 + N - 1 \) can be written as

\[ x = Z_l(\omega_0) \alpha_l + e. \quad (6) \]

where

\[ Z_l(\omega) = \begin{bmatrix} c(\omega) & c(2\omega) & \cdots & c(l\omega) & s(\omega) & s(2\omega) & \cdots & s(l\omega) \end{bmatrix} \]

\[ c(\omega) = \begin{bmatrix} \cos(\omega n_0) & \cdots & \cos(\omega(n_0 + N - 1)) \end{bmatrix}^T \]

\[ s(\omega) = \begin{bmatrix} \sin(\omega n_0) & \cdots & \sin(\omega(n_0 + N - 1)) \end{bmatrix}^T \]

\[ \alpha_l = \begin{bmatrix} a_l^T & -b_l^T \end{bmatrix}^T, \quad a_l = \begin{bmatrix} a_1 & \cdots & a_l \end{bmatrix}^T, \quad b_l = \begin{bmatrix} b_1 & \cdots & b_l \end{bmatrix}^T. \]
The maximum likelihood estimate (MLE) for the fundamental frequency is

\[ \hat{\omega}_0 = \arg\max_{\omega_0} x^T Z(I(\omega_0)) \left[ Z(I(\omega_0)) Z(I(\omega_0)) \right]^{-1} Z(I(\omega_0)) x \]

and is also known as the nonlinear least squares (NLS) estimate.
The maximum likelihood estimate (MLE) for the fundamental frequency is

\[ \hat{\omega}_0 = \arg\max_{\omega_0} x^T Z_l(\omega_0) \left[ Z_l^T(\omega_0) Z_l(\omega_0) \right]^{-1} Z_l^T(\omega_0) x \] (7)

and is also known as the nonlinear least squares (NLS) estimate.

- The ML/NLS estimator has been known since Quinn and Thomson (1991), but is costly to compute.
Maximum Likelihood Estimation

\( \omega_0 \frac{N}{2\pi} \) [cycles/segment]
Maximum Likelihood Estimation

1. Compute NLS cost function

\[ \hat{\omega}_0 = \arg\max_{\omega_0 \in (0, \pi/l)} x^T Z_I(\omega_0) \left[ Z_I^T(\omega_0) Z_I(\omega_0) \right]^{-1} Z_I^T(\omega_0) x \]  

(8)

on an \( F/l \)-point uniform grid for all model orders \( l \in \{1, \ldots, L\} \).
1. Compute NLS cost function

\[ \hat{\omega}_0 = \arg \max_{\omega_0 \in (0, \pi/l)} x^T Z_l(\omega_0) \left[ Z_l^T(\omega_0) Z_l(\omega_0) \right]^{-1} Z_l^T(\omega_0) x \]  

on an \( F/l \)-point uniform grid for all model orders \( l \in \{1, \ldots, L\} \).

2. Optionally refine the \( L \) grid estimates.
Maximum Likelihood Estimation

1. Compute NLS cost function

\[
\hat{\omega}_0 = \arg\max_{\omega_0 \in (0, \pi/l)} x^T Z_l(\omega_0) \left[ Z_l^T(\omega_0) Z_l(\omega_0) \right]^{-1} Z_l^T(\omega_0) x
\] (8)

on an \( F/l \)-point uniform grid for all model orders \( l \in \{1, \ldots, L\} \).

2. Optionally refine the \( L \) grid estimates.

3. Do model comparison.
1. Compute NLS cost function

\[
\hat{\omega}_0 = \arg\max_{\omega_0 \in (0, \pi/l)} x^T Z_l(\omega_0) \left[ Z_l^T(\omega_0) Z_l(\omega_0) \right]^{-1} Z_l^T(\omega_0) x
\]

on an \(F/l\)-point uniform grid for all model orders \(l \in \{1, \ldots, L\}\).

2. Optionally refine the \(L\) grid estimates.

3. Do model comparison.
Harmonic Summation

The harmonic summation (HS) estimator

\[ \hat{\omega}_0 \approx \argmax_{\omega_0 \in (0, \pi/l)} x^T Z_l(\omega_0) \left[ N I_l / 2 \right]^{-1} Z_l^T(\omega_0) x. \]  

(9)
Harmonic Summation

The harmonic summation (HS) estimator

$$\hat{\omega}_0 \approx \operatorname*{argmax}_{\omega_0 \in (0, \pi/l)} x^T Z_l(\omega_0) \left[NI_l/2\right]^{-1} Z_l^T(\omega_0)x.$$ \hspace{1cm} (9)

Complexities

<table>
<thead>
<tr>
<th>Method</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML/NLS</td>
<td>$O(F \log F) + O(FL^3)$</td>
</tr>
<tr>
<td>HS</td>
<td>$O(F \log F) + O(FL)$</td>
</tr>
</tbody>
</table>
Harmonic Summation

The harmonic summation (HS) estimator

$$\hat{\omega}_0 \approx \operatorname{argmax}_{\omega_0 \in (0, \pi/l)} x^T Z_l(\omega_0) \left[NI/2\right]^{-1} Z_l^T(\omega_0)x.$$  \hspace{1cm} (9)

Complexities

<table>
<thead>
<tr>
<th>Method</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML/NLS</td>
<td>$O(F \log F) + O(FL^3)$</td>
</tr>
<tr>
<td>HS</td>
<td>$O(F \log F) + O(FL)$</td>
</tr>
</tbody>
</table>
Estimation accuracy for a high fundamental frequency

Setup: \( N = 200, l = 10, 10000 \) repetitions, random phases, and constant amplitudes, and SNR of 15 dB
Estimation accuracy for a high fundamental frequency

Setup: $N = 200$, $l = 10$, 10000 repetitions, random phases, and constant amplitudes, and SNR of 15 dB
Estimation accuracy for a high fundamental frequency

Setup: $N = 200$, $l = 10$, 10000 repetitions, random phases, and constant amplitudes, and SNR of 15 dB
Estimation accuracy for a high fundamental frequency

Setup: $N = 200$, $l = 10$, 10000 repetitions, random phases, and constant amplitudes, and SNR of 15 dB
Estimation accuracy for a low fundamental frequency

Setup: \( N = 200, \ l = 10, \) 10000 repetitions, random phases, and constant amplitudes, and SNR of 0 dB
Estimation accuracy for a low fundamental frequency

Setup: \( N = 200, \ l = 10, \) 10000 repetitions, random phases, and constant amplitudes, and SNR of 0 dB

\[
\text{RMS}(\omega_0 - \hat{\omega}_0)
\]

\[
\omega_0 \frac{N}{2\pi} \quad \text{[cycles/segment]}
\]
Estimation accuracy for a low fundamental frequency

Setup: $N = 200$, $l = 10$, 10000 repetitions, random phases, and constant amplitudes, and SNR of 0 dB

\[
\text{RMS}(\omega_0 - \hat{\omega}_0)
\]

- Asymp. CRLB
- ML/NLS
- HS

\[
\omega_0 \frac{N}{2\pi} \quad \text{[cycles/segment]}
\]
Estimation accuracy for a low fundamental frequency

Setup: $N = 200$, $l = 10$, 10000 repetitions, random phases, and constant amplitudes, and SNR of 0 dB
Summary So Far

- When the fundamental frequency is **not low** (less than approx. 2 cycles/sample), HS and ML/NLS produce nearly the same estimates.
Summary So Far

- When the fundamental frequency is not low (less than approx. 2 cycles/sample), HS and ML/NLS produce nearly the same estimates.
- ML/NLS is much more accurate for low fundamental frequencies.
Summary So Far

- When the fundamental frequency is not low (less than approx. 2 cycles/sample), HS and ML/NLS produce nearly the same estimates.
- ML/NLS is much more accurate for low fundamental frequencies.
- For an $F$-point grid and a maximum candidate model order of $L$, the complexities of the grid search are:
  - $\mathcal{O}(F \log F) + \mathcal{O}(F L^3)$ for ML/NLS
  - $\mathcal{O}(F \log F) + \mathcal{O}(F L)$ for HS
Summary So Far

- When the fundamental frequency is not low (less than approx. 2 cycles/sample), HS and ML/NLS produce nearly the same estimates.
- ML/NLS is much more accurate for low fundamental frequencies.
- For an $F$-point grid and a maximum candidate model order of $L$, the complexities of the grid search are:
  - ML/NLS: $O(F \log F) + O(FL^3)$
  - HS: $O(F \log F) + O(FL)$

Contribution
ML/NLS: $O(F \log F) + O(FL)$
Agenda

Fundamental Frequency Estimation
Maximum Likelihood Estimation

Fast Maximum Likelihood Estimation

Results

Summary
Fast ML/NLS

\[ J_l(\omega) = x^T Z_l(\omega_0) \left[ Z_l^T(\omega_0)Z_l(\omega_0) \right]^{-1} Z_l^T(\omega_0)x \] (10)
Fast ML/NLS

\[ J_l(\omega) = x^T Z_l(\omega_0) \left[ Z_l^T(\omega_0)Z_l(\omega_0) \right]^{-1} Z_l^T(\omega_0)x \]  

1. Solve \( Z_l^T(\omega_0)Z_l(\omega_0)\alpha_l = Z_l^T(\omega_0)x \) efficiently for \( \alpha_l \).
Fast ML/NLS

\[ J_l(\omega) = x^T Z_l(\omega_0) \left[ Z_l^T(\omega_0)Z_l(\omega_0) \right]^{-1} Z_l^T(\omega_0)x \]  

(10)

1. Solve \( Z_l^T(\omega_0)Z_l(\omega_0)\alpha_l = Z_l^T(\omega_0)x \) efficiently for \( \alpha_l \).

2. \( Z_l^T(\omega_0)x \) can be computed at the complexity of a single FFT.
Fast ML/NLS

\[ J_l(\omega) = x^T Z_l(\omega_0) \left[ Z_l^T(\omega_0)Z_l(\omega_0) \right]^{-1} Z_l^T(\omega_0)x \] (10)

1. Solve \( Z_l^T(\omega_0)Z_l(\omega_0)\alpha_l = Z_l^T(\omega_0)x \) efficiently for \( \alpha_l \).
2. \( Z_l^T(\omega_0)x \) can be computed at the complexity of a single FFT.
3. The coefficient matrix has a block Toeplitz-plus-Hankel structure

\[ Z_l^T(\omega_0)Z_l(\omega_0) = \begin{bmatrix} T_l(\omega_0) & -\tilde{T}_l(\omega_0) \\ \tilde{T}_l(\omega_0) & T_l(\omega_0) \end{bmatrix} + \begin{bmatrix} H_l(\omega_0) & \tilde{H}_l(\omega_0) \\ \tilde{H}_l(\omega_0) & -H_l(\omega_0) \end{bmatrix} \] (11)
Fast ML/NLS

\[ J_l(\omega) = x^T Z_l(\omega_0) \left[ Z_l^T(\omega_0) Z_l(\omega_0) \right]^{-1} Z_l^T(\omega_0)x \] (10)

1. Solve \( Z_l^T(\omega_0) Z_l(\omega_0) \alpha_l = Z_l^T(\omega_0)x \) efficiently for \( \alpha_l \).
2. \( Z_l^T(\omega_0)x \) can be computed at the complexity of a single FFT.
3. The coefficient matrix has a block Toeplitz-plus-Hankel structure

\[
Z_l^T(\omega_0) Z_l(\omega_0) = \begin{bmatrix}
\tilde{T}_l(\omega_0) & -\tilde{T}_l(\omega_0) \\
\tilde{T}_l(\omega_0) & T_l(\omega_0)
\end{bmatrix} + \begin{bmatrix}
H_l(\omega_0) & \tilde{H}_l(\omega_0) \\
\tilde{H}_l(\omega_0) & -H_l(\omega_0)
\end{bmatrix}
\] (11)

4. The cost function \( J_l(\omega) \) does not depend on the start index \( n_0 \).
\[ J_l(\omega) = x^T Z_l(\omega_0) \left[ Z_l^T(\omega_0)Z_l(\omega_0) \right]^{-1} Z_l^T(\omega_0)x \]  

1. Solve \( Z_l^T(\omega_0)Z_l(\omega_0)\alpha_l = Z_l^T(\omega_0)x \) efficiently for \( \alpha_l \).
2. \( Z_l^T(\omega_0)x \) can be computed at the complexity of a single FFT.
3. The coefficient matrix has a block Toeplitz-plus-Hankel structure

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

4. The cost function \( J_l(\omega) \) does not depend on the start index \( n_0 \).
5. If \( n_0 = -(N-1)/2 \), then \( \tilde{T}_l(\omega_0) = \tilde{H}_l(\omega_0) = 0 \), and

\[
\begin{bmatrix} T_l(\omega_0) + H_l(\omega_0) \end{bmatrix} a_l(\omega_0) = \tilde{w}_l(\omega_0) \\
\begin{bmatrix} T_l(\omega_0) - H_l(\omega_0) \end{bmatrix} b_l(\omega_0) = -\tilde{w}_l(\omega_0).
\]

where \( Z_l^T(\omega_0)x = [\tilde{w}_l^T(\omega_0), \tilde{w}_l^T(\omega_0)]^T \).
Fast ML/NLS

\[
[T_i(\omega_0) + H_i(\omega_0)] a_i(\omega_0) = \bar{w}_i(\omega_0)
\] (14)
Fast ML/NLS

\[ [T_l(\omega_0) + H_l(\omega_0)] a_l(\omega_0) = \bar{w}_l(\omega_0) \]  

(14)

Fast ML/NLS

\[ [T_i(\omega_0) + H_i(\omega_0)] \mathbf{a}_i(\omega_0) = \mathbf{w}_i(\omega_0) \]  \hspace{1cm} (14)

6. Fast algorithms for Toeplitz-plus-Hankel: Reduces time complexity from \( O(l^3) \) to \( O(l^2) \) (Merchant and Parks (1982), Gohberg and Koltracht (1989), Kailath and Sayed (1995)).

7. The solutions to all upper-left subsystems of (14) is a solution to a lower order.

7. The solutions to all upper-left subsystems of (14) is a solution to a lower order.

8. Thus, solving the system for $l = L$ using a recursive Toeplitz-plus-Hankel solver gives the solutions for $l = 1, \ldots, L - 1$ for free in the process.
Fast ML/NLS

\[
\begin{bmatrix}
    T_i(\omega_0) + H_i(\omega_0)
\end{bmatrix}
\begin{bmatrix}
    a_i(\omega_0)
\end{bmatrix}
= \bar{w}_i(\omega_0)
\]  

(14)


7. The solutions to all upper-left subsystems of (14) is a solution to a lower order.

8. Thus, solving the system for \(l = L\) using a recursive Toeplitz-plus-Hankel solver gives the solutions for \(l = 1, \ldots, L - 1\) for free in the process.

9. Solving (14) therefore has a time complexity of \(O(l)\) when we have the solution to (14) for \(l - 1\).
Fast ML/NLS

\[
[T_l(\omega_0) + H_l(\omega_0)] a_l(\omega_0) = \bar{w}_l(\omega_0) \quad (14)
\]


7. The solutions to all upper-left subsystems of (14) is a solution to a lower order.

8. Thus, solving the system for \(l = L\) using a recursive Toeplitz-plus-Hankel solver gives the solutions for \(l = 1, \ldots, L - 1\) for free in the process.

9. Solving (14) therefore has a time complexity of \(O(l)\) when we have the solution to (14) for \(l - 1\).

10. The total time complexity is reduced to

\[
O(F \log F) + O(FL) \quad (15)
\]
\[ \begin{bmatrix} T_l(\omega_0) + H_l(\omega_0) \end{bmatrix} a_l(\omega_0) = \bar{w}_l(\omega_0) \] (16)

- We use the recursive solver by Gohberg and Koltracht (1989).
Fast ML/NLS
Toeplitz-plus-Hankel Solver

\[
[T_l(\omega_0) + H_l(\omega_0)] a_l(\omega_0) = \bar{w}_l(\omega_0)
\] (16)

- We use the recursive solver by Gohberg and Koltracht (1989).
- Consists of a data independent and a data dependent step.
Fast ML/NLS
Toeplitz-plus-Hankel Solver

\[
[T_l(\omega_0) + H_l(\omega_0)] a_l(\omega_0) = \bar{w}_l(\omega_0)
\]  

(16)

- We use the recursive solver by Gohberg and Koltracht (1989).
- Consists of a data independent and a data dependent step.
- The data independent step consists in solving

\[
[T_l(\omega_0) + H_l(\omega_0)] \gamma_l(\omega_0) = [0 \cdots 0 1]^T
\]

(17)

for \( \gamma_l(\omega_0) \) for \( l = 1, \ldots, L \).
Fast ML/NLS
Toeplitz-plus-Hankel Solver

\[
[T_l(\omega_0) + H_l(\omega_0)] a_l(\omega_0) = \bar{w}_l(\omega_0)
\] (16)

- We use the recursive solver by Gohberg and Koltracht (1989).
- Consists of a data independent and a data dependent step.
- The data independent step consists in solving

\[
[T_l(\omega_0) + H_l(\omega_0)] \gamma_l(\omega_0) = [0 \cdots 0 1]^T
\] (17)

for \( \gamma_l(\omega_0) \) for \( l = 1, \ldots, L \).
- The data independent step can be computed off-line. Requires memory to store \( \gamma_l(\omega_0) \) for all frequencies and model orders.
Agenda

Fundamental Frequency Estimation
Maximum Likelihood Estimation

Fast Maximum Likelihood Estimation

Results

Summary
Computation Time vs. Model Order
MATLAB Implementation

Setup: $N = 200$ (25 ms @ $f_s = 8000$ Hz), $F = 5NL$, T420 laptop
Computation Time vs. Model Order
MATLAB Implementation

Setup: \( N = 200 \) (25 ms @ \( f_s = 8000 \) Hz), \( F = 5NL \), T420 laptop

\[
\tau [s] \quad 10^{-3} \quad 10^{-1} \quad 10^{1}
\]

\( L \)

- - - RT Limit \quad \triangle \text{Standard ML}

\( \tau \text{(Standard ML)} \approx 60 \)

\( \tau \text{(Fast ML)} \approx 150 \)

\( \tau \text{(Faster ML)} \approx 500 \)
Computation Time vs. Model Order
MATLAB Implementation

Setup: $N = 200$ (25 ms @ $f_s = 8000$ Hz), $F = 5NL$, T420 laptop

- - - RT Limit  ▲ Standard ML  ○ Fast ML

$\tau [s]$

$10^{-3}$  $10^{-1}$  $10^{1}$

5 10 15 20 25 30 35 40 45 50

$L$

$\tau (\text{Standard ML}) \approx 60$

$\tau (\text{Fast ML}) \approx 150$

$\tau (\text{Fast ML}) \approx 500$
Setup: \( N = 200 \) (25 ms @ \( f_s = 8000 \) Hz), \( F = 5NL \), T420 laptop

\[ \tau [s] \]

\[ 10^{-3} \quad 10^{-1} \quad 10^{1} \]

\[ 5 \quad 10 \quad 15 \quad 20 \quad 25 \quad 30 \quad 35 \quad 40 \quad 45 \quad 50 \]

- - - RT Limit
- Standard ML
- Fast ML
- Faster ML

\( \tau (\text{Standard ML}) \approx 60 \)
\( \tau (\text{Fast ML}) \approx 150 \)
\( \tau (\text{Faster ML}) \approx 500 \)
Computation Time vs. Model Order
MATLAB Implementation

Setup: $N = 200$ (25 ms @ $f_s = 8000$ Hz), $F = 5NL$, T420 laptop

- - - RT Limit  ▲  Standard ML  ○  Fast ML  ■  Faster ML  ■  HS

$\tau$ [s]

\begin{align*}
\tau(\text{Standard ML}) & \approx 60 \\
\tau(\text{Fast ML}) & \approx 150 \\
\tau(\text{Faster ML}) & \approx 500 \\
\tau(\text{HS}) &
\end{align*}
Computation Time vs. Model Order
MATLAB Implementation

Setup: \( N = 200 \) (25 ms @ \( f_s = 8000 \) Hz), \( F = 5NL \), T420 laptop

\[ \tau(\text{Standard ML}) \approx 60\tau(\text{Fast ML}) \approx 150\tau(\text{Faster ML}) \approx 500\tau(\text{HS}) \]
Setup: $L = 30$, $F = 5NL$, T420 laptop
Computation Time vs. Data Size
MATLAB Implementation

Setup: $L = 30$, $F = 5NL$, T420 laptop

--- RT limit  Standard ML
Computation Time vs. Data Size
MATLAB Implementation

Setup: $L = 30$, $F = 5NL$, T420 laptop
Computation Time vs. Data Size
MATLAB Implementation

Setup: $L = 30$, $F = 5NL$, T420 laptop
Computation Time vs. Data Size
MATLAB Implementation

Setup: \( L = 30, F = 5NL, \) T420 laptop
Washing Machine Example
Acoustic measurements by Brüel & Kjær

$\mathbf{\triangleright} \quad f_s = 44.1 \text{ kHz}$
Washing Machine Example
Acoustic measurements by Brüel & Kjær

- $f_s = 44.1$ kHz
- $f_{rs} = 4410$ Hz, 60 ms windows, 15/16 overlap, and $L = 15$
Washing Machine Example
Acoustic measurements by Brüel & Kjær

- $f_s = 44.1$ kHz
- Computation time: 28 s (50 % overlap: 3.8 s)
- $f_{rs} = 4410$ Hz, 60 ms windows, 15/16 overlap, and $L = 15$
Washing Machine Example
Acoustic measurements by Brüel & Kjær

Estimated Fundamental Frequency

\[ f \text{ [Hz]} \]
\[ t \text{ [s]} \]

\[ 98 \quad 99 \quad 100 \quad 101 \quad 102 \]
Washing Machine Example
Acoustic measurements by Brüel & Kjær

Order Analysis

Power/sample [dB]

$t$ [s]
Agenda

Fundamental Frequency Estimation
   Maximum Likelihood Estimation

Fast Maximum Likelihood Estimation

Results

Summary
The ML/NLS estimator works well for a low SNR and/or a low fundamental frequency.
Summary

- The ML/NLS estimator works well for a low SNR and/or a low fundamental frequency.
- Although the ML/NLS estimator has been known for at least 25 years, no algorithm with a complexity lower than
  \[ O(F \log F) + O(FL^3) \]  
  has been proposed for computing the fundamental frequency estimate on an \( F \)-point uniform grid for all model orders up to \( L \).
Summary

- The ML/NLS estimator works well for a low SNR and/or a low fundamental frequency.
- Although the ML/NLS estimator has been known for at least 25 years, no algorithm with a complexity lower than
  \[ \mathcal{O}(F \log F) + \mathcal{O}(F L^3) \] (18)
  has been proposed for computing the fundamental frequency estimate on an \( F \)-point uniform grid for all model orders up to \( L \).
- We have proposed an algorithm that lower the complexity to
  \[ \mathcal{O}(F \log F) + \mathcal{O}(F L) \] (19)
  which is the same as that for harmonic summation.
Summary

The ML/NLS estimator works well for a low SNR and/or a low fundamental frequency.

Although the ML/NLS estimator has been known for at least 25 years, no algorithm with a complexity lower than

\[ O(F \log F) + O(FL^3) \]  \hspace{1cm} (18)

has been proposed for computing the fundamental frequency estimate on an \( F \)-point uniform grid for all model orders up to \( L \).

We have proposed an algorithm that lower the complexity to

\[ O(F \log F) + O(FL) \]  \hspace{1cm} (19)

which is the same as that for harmonic summation.

For a typical configuration, simulation studies show that the proposed algorithm is approximately 60-150 faster than the standard algorithm and 4 – 10 times slower than harmonic summation.
Thanks for your attention!