Fast and Statistically Efficient Fundamental Frequency Estimation

ICASSP 2016

March 23, 2016

J. K. Nielsen[1,2], T. L. Jensen[1], J. R. Jensen[3], M. G. Christensen[3], and S. H. Jensen[1]

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

[2]Aalborg University
Dept. of Electronic Systems

[3]Bang & Olufsen A/S
Denmark

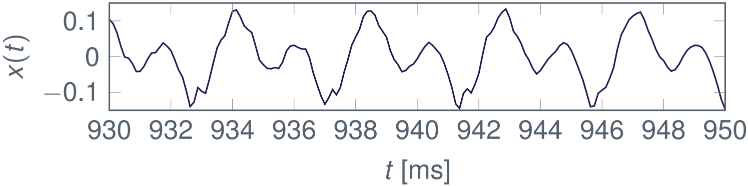
[1]Aalborg University
Audio Analysis Lab, AD:MT

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

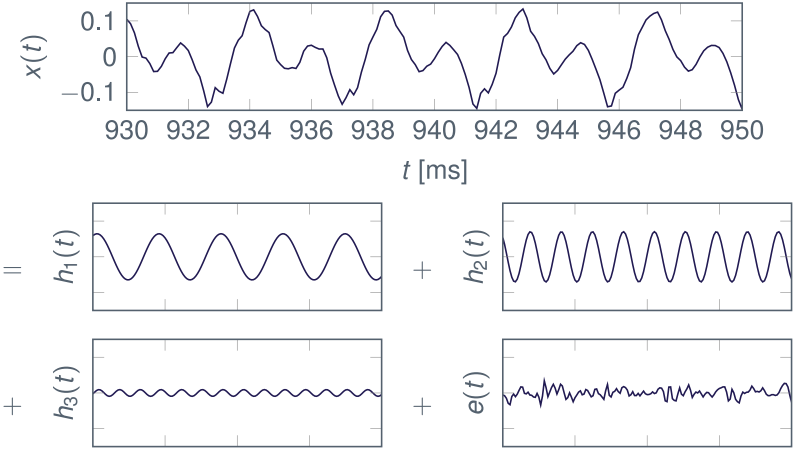
AALBORG UNIVERSITY
DENMARK

Agenda

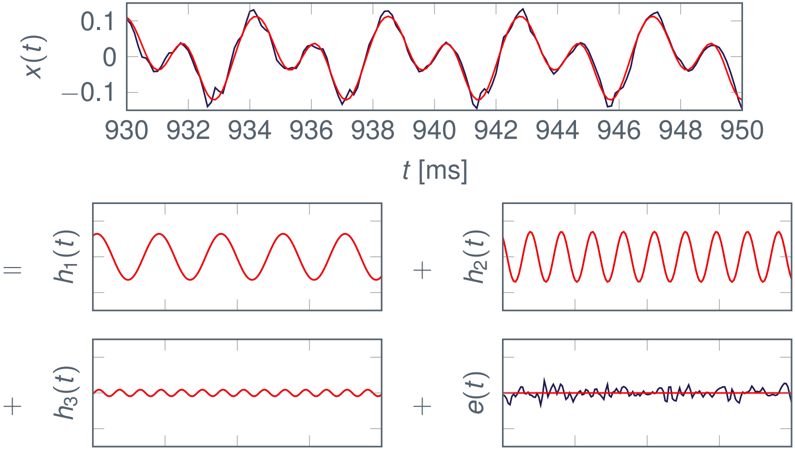
Speech Signal Example



Speech Signal Example



Speech Signal Example



Mathematical Model

Mathematical Model

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

where

- A_i real amplitude of the *i*th harmonic
- ϕ_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 σ^2

Mathematical Model

Mathematical Model

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

where

- A_i real amplitude of the ith harmonic
- ϕ_i phase of the ith harmonic
- f_0 fundamental frequency in Hz
- l the number of harmonics/model order
- $e(t)$ white Gaussian noise with variance σ^2

Analysis problem: Get f_0 and l from the data.

Estimation Methods

Correlation Methods

A periodic signal satisfies that

$$x(t) = x(t + T) \tag{2}$$

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

$$x(t) = x(t + T) \tag{2}$$

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

Estimation Methods

Correlation Methods

A periodic signal satisfies that

$$x(t) = x(t + T) \tag{2}$$

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

Parametric Methods

Estimate the parameters in

$$x(t) = \sum_{i=1}^{l} A_i \cos(i2\pi f_0 t + \phi_i) + e(t) \tag{3}$$

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

Estimation Methods

Parametric Methods

Estimate the parameters in

$$x(t) = \sum_{i=1}^{l} A_i \cos(i2\pi f_0 t + \phi_i) + e(t) \tag{3}$$

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

- \+ 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

Estimation Methods

Parametric Methods

Estimate the parameters in

$$x(t) = \sum_{i=1}^{l} A_i \cos(i2\pi f_0 t + \phi_i) + e(t) \tag{3}$$

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

- \+ 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

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

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

$$\boldsymbol{x} = \boldsymbol{Z}_l(\omega_0)\boldsymbol{\alpha}_l + \boldsymbol{e}\,. \tag{6}$$

where

$$\boldsymbol{Z}_l(\omega) = \begin{bmatrix} \boldsymbol{c}(\omega) & \boldsymbol{c}(2\omega) & \cdots & \boldsymbol{c}(l\omega) & \boldsymbol{s}(\omega) & \boldsymbol{s}(2\omega) & \cdots & \boldsymbol{s}(l\omega) \end{bmatrix}$$

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

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

$$\boldsymbol{\alpha}_l = \begin{bmatrix} \boldsymbol{a}_l^T & -\boldsymbol{b}_l^T \end{bmatrix}^T, \ \boldsymbol{a}_l = \begin{bmatrix} a_1 & \cdots & a_l \end{bmatrix}^T, \ \boldsymbol{b}_l = \begin{bmatrix} b_1 & \cdots & b_l \end{bmatrix}^T.$$

Maximum Likelihood Estimation

The maximum likelihood estimate (MLE) for the fundamental frequency is

$$\hat{\omega}_0 = \underset{\omega_0}{\operatorname{argmax}}\, \boldsymbol{x}^T \boldsymbol{Z}_l(\omega_0) \left[\boldsymbol{Z}_l^T(\omega_0)\boldsymbol{Z}_l(\omega_0)\right]^{-1} \boldsymbol{Z}_l^T(\omega_0)\boldsymbol{x} \tag{7}$$

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

Maximum Likelihood Estimation

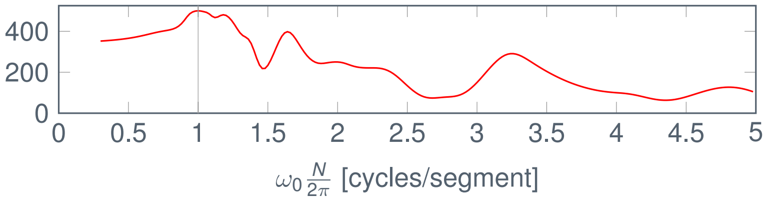
The maximum likelihood estimate (MLE) for the fundamental frequency is

$$\hat{\omega}_0 = \underset{\omega_0}{\operatorname{argmax}}\, \boldsymbol{x}^T \boldsymbol{Z}_l(\omega_0) \left[\boldsymbol{Z}_l^T(\omega_0)\boldsymbol{Z}_l(\omega_0)\right]^{-1} \boldsymbol{Z}_l^T(\omega_0)\boldsymbol{x} \qquad (7)$$

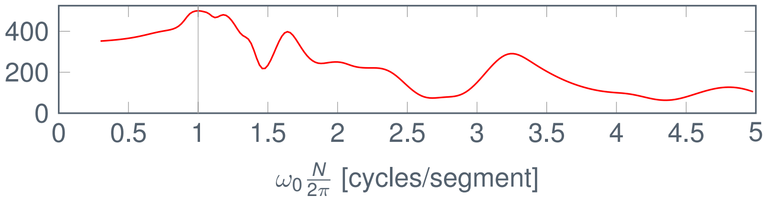
and is also known as the nonlinear least squares (NLS) estimate.

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

Maximum Likelihood Estimation



Maximum Likelihood Estimation

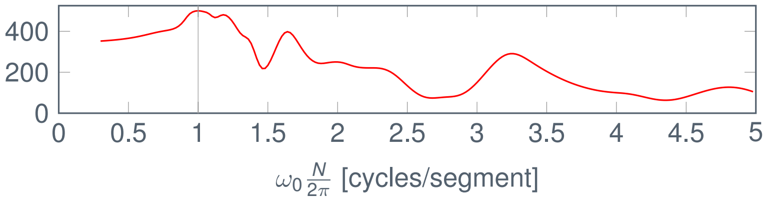


1. Compute NLS cost function

$$\hat{\omega}_0 = \underset{\omega_0 \in (0, \pi/l)}{\operatorname{argmax}} \; \boldsymbol{x}^T \boldsymbol{Z}_l(\omega_0) \left[\boldsymbol{Z}_l^T(\omega_0) \boldsymbol{Z}_l(\omega_0) \right]^{-1} \boldsymbol{Z}_l^T(\omega_0) \boldsymbol{x} \qquad (8)$$

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

Maximum Likelihood Estimation



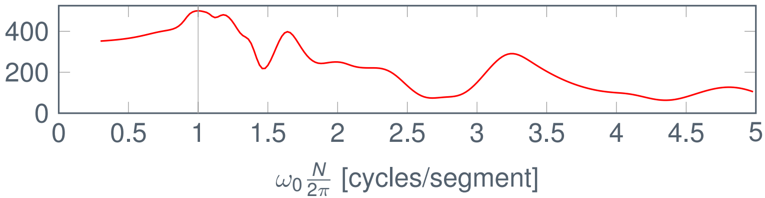
1. Compute NLS cost function

$$\hat{\omega}_0 = \underset{\omega_0 \in (0, \pi/l)}{\operatorname{argmax}} \; \boldsymbol{x}^T \boldsymbol{Z}_l(\omega_0) \left[\boldsymbol{Z}_l^T(\omega_0) \boldsymbol{Z}_l(\omega_0) \right]^{-1} \boldsymbol{Z}_l^T(\omega_0) \boldsymbol{x} \quad (8)$$

   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

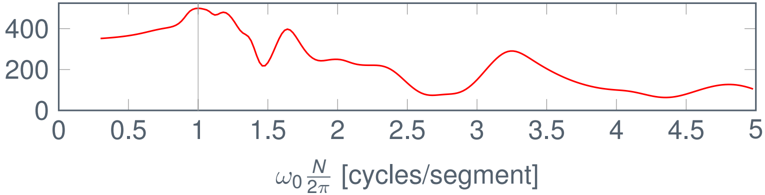


1. Compute NLS cost function

$$\hat{\omega}_0 = \underset{\omega_0 \in (0, \pi/l)}{\text{argmax}}\ \boldsymbol{x}^T \boldsymbol{Z}_l(\omega_0) \left[\boldsymbol{Z}_l^T(\omega_0)\boldsymbol{Z}_l(\omega_0)\right]^{-1} \boldsymbol{Z}_l^T(\omega_0)\boldsymbol{x} \qquad (8)$$

   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.

Maximum Likelihood Estimation



1. Compute NLS cost function

$$\hat{\omega}_0 = \underset{\omega_0 \in (0, \pi/l)}{\operatorname{argmax}} \, \boldsymbol{x}^T \boldsymbol{Z}_l(\omega_0) \left[\boldsymbol{Z}_l^T(\omega_0) \boldsymbol{Z}_l(\omega_0) \right]^{-1} \boldsymbol{Z}_l^T(\omega_0) \boldsymbol{x} \qquad (8)$$

   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.

Harmonic Summation

The harmonic summation (HS) estimator

$$\hat{\omega}_0 \approx \underset{\omega_0 \in (0, \pi / l)}{\operatorname{argmax}} \ \boldsymbol{x}^T \boldsymbol{Z}_l(\omega_0) \left[N \boldsymbol{I}_l / 2 \right]^{-1} \boldsymbol{Z}_l^T(\omega_0) \boldsymbol{x} \ . \qquad (9)$$

Harmonic Summation

The harmonic summation (HS) estimator

$$\hat{\omega}_0 \approx \underset{\omega_0 \in (0, \pi/l)}{\operatorname{argmax}} \, \boldsymbol{x}^T \boldsymbol{Z}_l(\omega_0) \left[N \boldsymbol{I}_l / 2\right]^{-1} \boldsymbol{Z}_l^T(\omega_0) \boldsymbol{x} \, . \tag{9}$$

Complexities

ML/NLS	$\mathcal{O}(F \log F) + \mathcal{O}(FL^3)$
HS	$\mathcal{O}(F \log F) + \mathcal{O}(FL)$

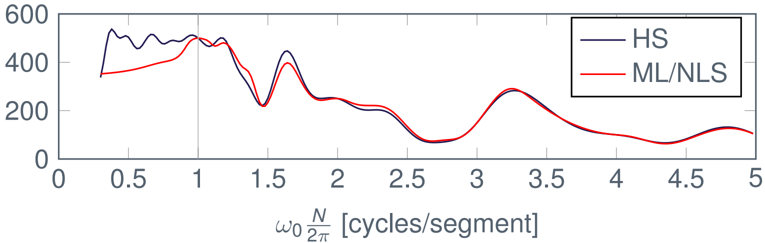
Harmonic Summation

The harmonic summation (HS) estimator

$$\hat{\omega}_0 \approx \underset{\omega_0 \in (0, \pi / l)}{\operatorname{argmax}} \; \boldsymbol{x}^T \boldsymbol{Z}_l(\omega_0) \left[N \boldsymbol{I}_l / 2\right]^{-1} \boldsymbol{Z}_l^T(\omega_0) \boldsymbol{x} \, . \qquad (9)$$

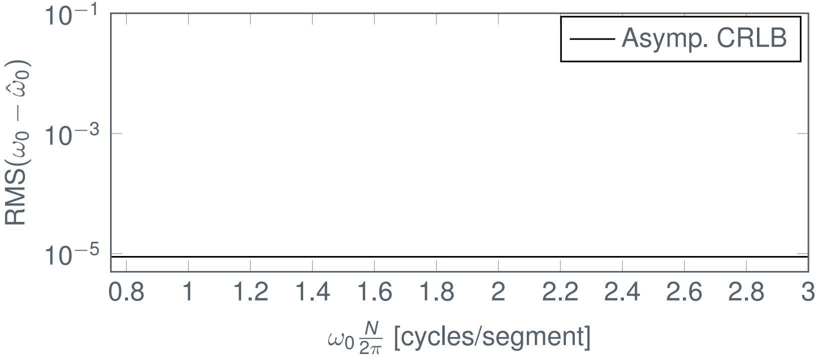
Complexities

ML/NLS	$\mathcal{O}(F \log F) + \mathcal{O}(FL^3)$
HS	$\mathcal{O}(F \log F) + \mathcal{O}(FL)$



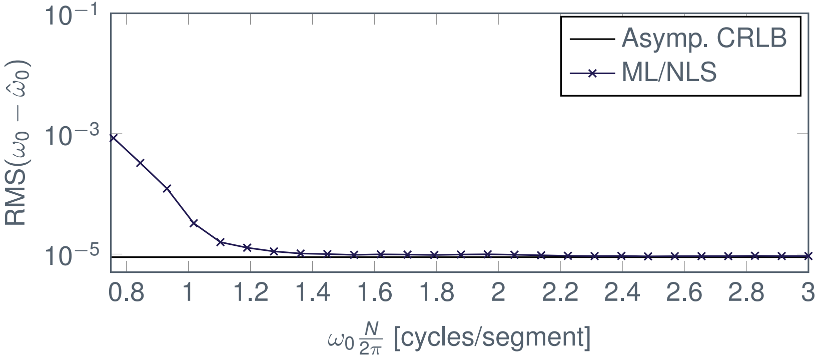
Estimation accuracy for a high fundamental frequency

Setup: $N = 200$, $l = 10$, 10000 repetitions, random phases, and constant amplitudes, and SNR of 15 dB



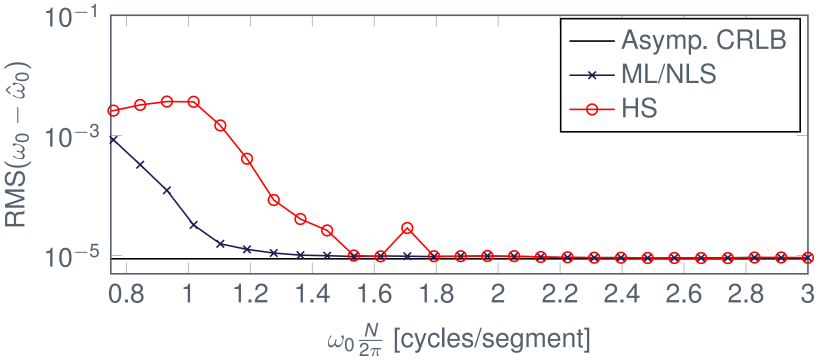
Estimation accuracy for a high fundamental frequency

Setup: $N = 200$, $l = 10$, 10000 repetitions, random phases, and constant amplitudes, and SNR of 15 dB



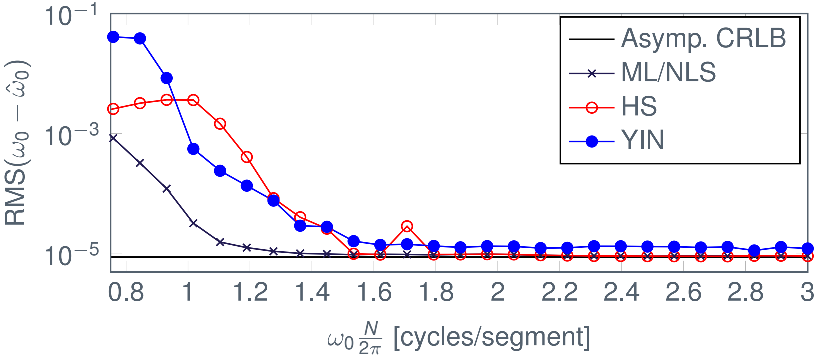
Estimation accuracy for a high fundamental frequency

Setup: $N = 200$, $l = 10$, 10000 repetitions, random phases, and constant amplitudes, and SNR of 15 dB



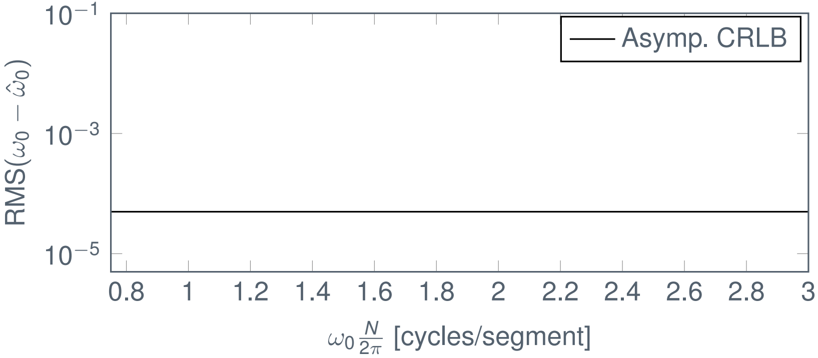
Estimation accuracy for a high fundamental frequency

Setup: $N = 200$, $l = 10$, 10000 repetitions, random phases, and constant amplitudes, and SNR of 15 dB



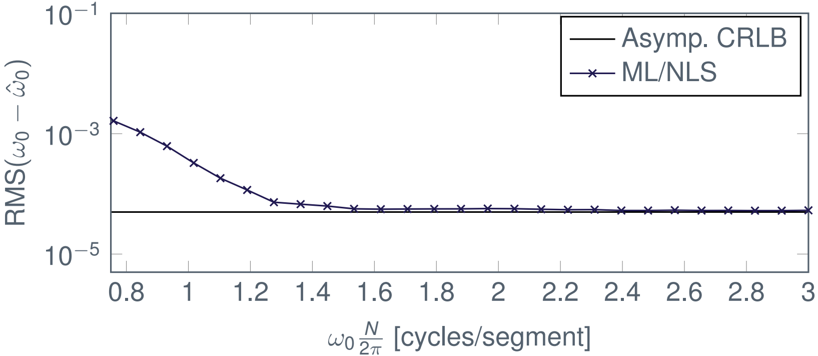
Estimation accuracy for a low fundamental frequency

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



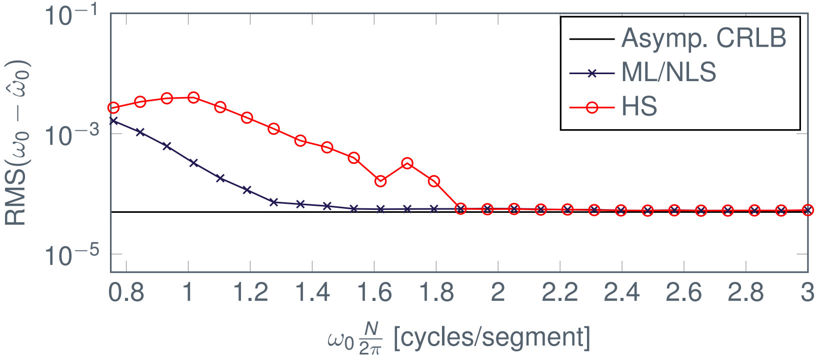
Estimation accuracy for a low fundamental frequency

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



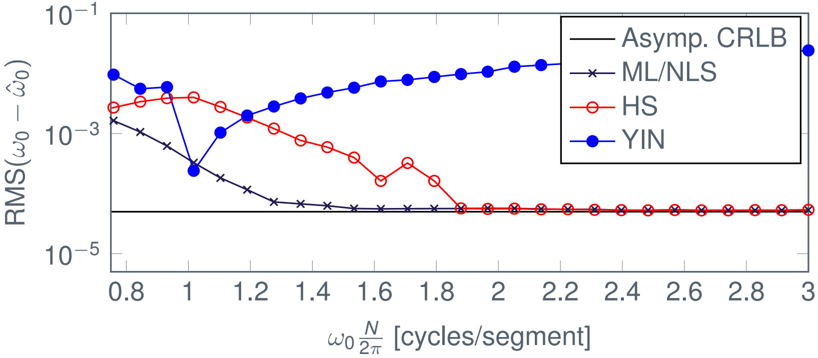
Estimation accuracy for a low fundamental frequency

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



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

ML/NLS	$\mathcal{O}(F \log F) + \mathcal{O}(FL^3)$
HS	$\mathcal{O}(F \log F) + \mathcal{O}(FL)$

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	$\mathcal{O}(F \log F) + \mathcal{O}(FL^3)$
HS	$\mathcal{O}(F \log F) + \mathcal{O}(FL)$

Contribution

ML/NLS: $\mathcal{O}(F \log F) + \mathcal{O}(FL)$

Agenda

Fundamental Frequency Estimation

Maximum Likelihood Estimation

Fast Maximum Likelihood Estimation

Results

Summary

Fast ML/NLS

$$J_l(\omega) = \boldsymbol{x}^T \boldsymbol{Z}_l(\omega_0) \left[\boldsymbol{Z}_l^T(\omega_0) \boldsymbol{Z}_l(\omega_0) \right]^{-1} \boldsymbol{Z}_l^T(\omega_0) \boldsymbol{x} \tag{10}$$

Fast ML/NLS

$$J_l(\omega) = \boldsymbol{x}^T \boldsymbol{Z}_l(\omega_0) \left[\boldsymbol{Z}_l^T(\omega_0)\boldsymbol{Z}_l(\omega_0)\right]^{-1} \boldsymbol{Z}_l^T(\omega_0)\boldsymbol{x} \tag{10}$$

1. Solve $\boldsymbol{Z}_l^T(\omega_0)\boldsymbol{Z}_l(\omega_0)\boldsymbol{\alpha}_l = \boldsymbol{Z}_l^T(\omega_0)\boldsymbol{x}$ efficiently for $\boldsymbol{\alpha}_l$.

Fast ML/NLS

$$J_l(\omega) = \boldsymbol{x}^T \boldsymbol{Z}_l(\omega_0) \left[\boldsymbol{Z}_l^T(\omega_0) \boldsymbol{Z}_l(\omega_0) \right]^{-1} \boldsymbol{Z}_l^T(\omega_0) \boldsymbol{x} \tag{10}$$

1. Solve $\boldsymbol{Z}_l^T(\omega_0)\boldsymbol{Z}_l(\omega_0)\boldsymbol{\alpha}_l = \boldsymbol{Z}_l^T(\omega_0)\boldsymbol{x}$ efficiently for $\boldsymbol{\alpha}_l$.
2. $\boldsymbol{Z}_l^T(\omega_0)\boldsymbol{x}$ can be computed at the complexity of a single FFT .

Fast ML/NLS

$$J_l(\omega) = \boldsymbol{x}^T \boldsymbol{Z}_l(\omega_0) \left[\boldsymbol{Z}_l^T(\omega_0)\boldsymbol{Z}_l(\omega_0)\right]^{-1} \boldsymbol{Z}_l^T(\omega_0)\boldsymbol{x} \tag{10}$$

1. Solve $\boldsymbol{Z}_l^T(\omega_0)\boldsymbol{Z}_l(\omega_0)\boldsymbol{\alpha}_l = \boldsymbol{Z}_l^T(\omega_0)\boldsymbol{x}$ efficiently for $\boldsymbol{\alpha}_l$.
2. $\boldsymbol{Z}_l^T(\omega_0)\boldsymbol{x}$ can be computed at the complexity of a single FFT .
3. The coefficient matrix has a block Toeplitz-plus-Hankel structure

$$\boldsymbol{Z}_l^T(\omega_0)\boldsymbol{Z}_l(\omega_0) = \begin{bmatrix} \boldsymbol{T}_l(\omega_0) & -\tilde{\boldsymbol{T}}_l(\omega_0) \\ \tilde{\boldsymbol{T}}_l(\omega_0) & \boldsymbol{T}_l(\omega_0) \end{bmatrix} + \begin{bmatrix} \boldsymbol{H}_l(\omega_0) & \tilde{\boldsymbol{H}}_l(\omega_0) \\ \tilde{\boldsymbol{H}}_l(\omega_0) & -\boldsymbol{H}_l(\omega_0) \end{bmatrix} \tag{11}$$

Fast ML/NLS

$$J_l(\omega) = \boldsymbol{x}^T \boldsymbol{Z}_l(\omega_0) \left[\boldsymbol{Z}_l^T(\omega_0)\boldsymbol{Z}_l(\omega_0)\right]^{-1} \boldsymbol{Z}_l^T(\omega_0)\boldsymbol{x} \qquad (10)$$

1. Solve $\boldsymbol{Z}_l^T(\omega_0)\boldsymbol{Z}_l(\omega_0)\boldsymbol{\alpha}_l = \boldsymbol{Z}_l^T(\omega_0)\boldsymbol{x}$ efficiently for $\boldsymbol{\alpha}_l$.
2. $\boldsymbol{Z}_l^T(\omega_0)\boldsymbol{x}$ can be computed at the complexity of a single FFT .
3. The coefficient matrix has a block Toeplitz-plus-Hankel structure

$$\boldsymbol{Z}_l^T(\omega_0)\boldsymbol{Z}_l(\omega_0) = \begin{bmatrix} \boldsymbol{T}_l(\omega_0) & -\tilde{\boldsymbol{T}}_l(\omega_0) \\ \tilde{\boldsymbol{T}}_l(\omega_0) & \boldsymbol{T}_l(\omega_0) \end{bmatrix} + \begin{bmatrix} \boldsymbol{H}_l(\omega_0) & \tilde{\boldsymbol{H}}_l(\omega_0) \\ \tilde{\boldsymbol{H}}_l(\omega_0) & -\boldsymbol{H}_l(\omega_0) \end{bmatrix} \qquad (11)$$

4. The cost function $J_l(\omega)$ does not depend on the start index n_0.

Fast ML/NLS

$$J_l(\omega) = \boldsymbol{x}^T \boldsymbol{Z}_l(\omega_0) \left[\boldsymbol{Z}_l^T(\omega_0)\boldsymbol{Z}_l(\omega_0)\right]^{-1} \boldsymbol{Z}_l^T(\omega_0)\boldsymbol{x} \tag{10}$$

1. Solve $\boldsymbol{Z}_l^T(\omega_0)\boldsymbol{Z}_l(\omega_0)\boldsymbol{\alpha}_l = \boldsymbol{Z}_l^T(\omega_0)\boldsymbol{x}$ efficiently for $\boldsymbol{\alpha}_l$.
2. $\boldsymbol{Z}_l^T(\omega_0)\boldsymbol{x}$ can be computed at the complexity of a single FFT .
3. The coefficient matrix has a block Toeplitz-plus-Hankel structure

$$\boldsymbol{Z}_l^T(\omega_0)\boldsymbol{Z}_l(\omega_0) = \begin{bmatrix} \boldsymbol{T}_l(\omega_0) & -\tilde{\boldsymbol{T}}_l(\omega_0) \\ \tilde{\boldsymbol{T}}_l(\omega_0) & \boldsymbol{T}_l(\omega_0) \end{bmatrix} + \begin{bmatrix} \boldsymbol{H}_l(\omega_0) & \tilde{\boldsymbol{H}}_l(\omega_0) \\ \tilde{\boldsymbol{H}}_l(\omega_0) & -\boldsymbol{H}_l(\omega_0) \end{bmatrix} \tag{11}$$

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

$$[\boldsymbol{T}_l(\omega_0) + \boldsymbol{H}_l(\omega_0)]\, \boldsymbol{a}_l(\omega_0) = \bar{\boldsymbol{w}}_l(\omega_0) \tag{12}$$

$$[\boldsymbol{T}_l(\omega_0) - \boldsymbol{H}_l(\omega_0)]\, \boldsymbol{b}_l(\omega_0) = -\tilde{\boldsymbol{w}}_l(\omega_0)\,. \tag{13}$$

where $\boldsymbol{Z}_l^T(\omega_0)\boldsymbol{x} = [\bar{\boldsymbol{w}}_l^T(\omega_0), \tilde{\boldsymbol{w}}_l^T(\omega_0)]^T$.

Fast ML/NLS

$$[\boldsymbol{T}_l(\omega_0) + \boldsymbol{H}_l(\omega_0)]\,\boldsymbol{a}_l(\omega_0) = \bar{\boldsymbol{w}}_l(\omega_0) \tag{14}$$

Fast ML/NLS

$$[\boldsymbol{T}_l(\omega_0) + \boldsymbol{H}_l(\omega_0)]\, \boldsymbol{a}_l(\omega_0) = \bar{\boldsymbol{w}}_l(\omega_0) \tag{14}$$

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

Fast ML/NLS

$$[\boldsymbol{T}_l(\omega_0) + \boldsymbol{H}_l(\omega_0)]\, \boldsymbol{a}_l(\omega_0) = \bar{\boldsymbol{w}}_l(\omega_0) \tag{14}$$

6. Fast algorithms for Toeplitz-plus-Hankel: Reduces time complexity from $\mathcal{O}(l^3)$ to $\mathcal{O}(l^2)$ (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.

Fast ML/NLS

$$[\boldsymbol{T}_l(\omega_0) + \boldsymbol{H}_l(\omega_0)]\,\boldsymbol{a}_l(\omega_0) = \bar{\boldsymbol{w}}_l(\omega_0) \tag{14}$$

6. Fast algorithms for Toeplitz-plus-Hankel: Reduces time complexity from $\mathcal{O}(l^3)$ to $\mathcal{O}(l^2)$ (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.
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

$$[\boldsymbol{T}_l(\omega_0) + \boldsymbol{H}_l(\omega_0)]\,\boldsymbol{a}_l(\omega_0) = \bar{\boldsymbol{w}}_l(\omega_0) \qquad (14)$$

6. Fast algorithms for Toeplitz-plus-Hankel: Reduces time complexity from $\mathcal{O}(l^3)$ to $\mathcal{O}(l^2)$ (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.
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 $\mathcal{O}(l)$ when we have the solution to (14) for $l - 1$.

Fast ML/NLS

$$[\boldsymbol{T}_l(\omega_0) + \boldsymbol{H}_l(\omega_0)]\,\boldsymbol{a}_l(\omega_0) = \bar{\boldsymbol{w}}_l(\omega_0) \tag{14}$$

6. Fast algorithms for Toeplitz-plus-Hankel: Reduces time complexity from $\mathcal{O}(l^3)$ to $\mathcal{O}(l^2)$ (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.
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 $\mathcal{O}(l)$ when we have the solution to (14) for $l-1$.
10. The total time complexity is reduced to

$$\mathcal{O}(F \log F) + \mathcal{O}(FL) \tag{15}$$

Fast ML/NLS

Toeplitz-plus-Hankel Solver

$$[\boldsymbol{T}_l(\omega_0) + \boldsymbol{H}_l(\omega_0)]\,\boldsymbol{a}_l(\omega_0) = \bar{\boldsymbol{w}}_l(\omega_0) \tag{16}$$

- We use the recursive solver by Gohberg and Koltracht (1989).

Fast ML/NLS

Toeplitz-plus-Hankel Solver

$$[\boldsymbol{T}_l(\omega_0) + \boldsymbol{H}_l(\omega_0)]\,\boldsymbol{a}_l(\omega_0) = \bar{\boldsymbol{w}}_l(\omega_0) \qquad (16)$$

- 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

$$[\boldsymbol{T}_l(\omega_0) + \boldsymbol{H}_l(\omega_0)]\,\boldsymbol{a}_l(\omega_0) = \bar{\boldsymbol{w}}_l(\omega_0) \tag{16}$$

- 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

$$[\boldsymbol{T}_l(\omega_0) + \boldsymbol{H}_l(\omega_0)]\,\gamma_l(\omega_0) = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix}^T . \tag{17}$$

for $\gamma_l(\omega_0)$ for $l = 1, \dots, L$.

Fast ML/NLS

Toeplitz-plus-Hankel Solver

$$[\boldsymbol{T}_l(\omega_0) + \boldsymbol{H}_l(\omega_0)]\, \boldsymbol{a}_l(\omega_0) = \bar{\boldsymbol{w}}_l(\omega_0) \quad (16)$$

- 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

$$[\boldsymbol{T}_l(\omega_0) + \boldsymbol{H}_l(\omega_0)]\, \gamma_l(\omega_0) = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix}^T . \quad (17)$$

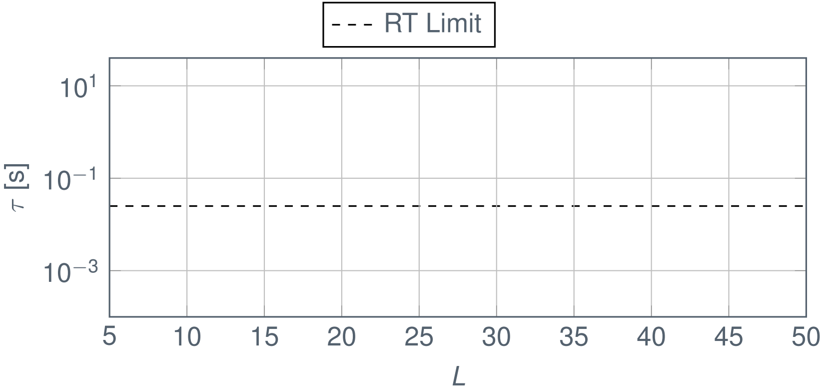
  for $\gamma_l(\omega_0)$ for $l = 1, \ldots, L$.
- The data independent step can be computed off-line. Requires memory to store $\gamma_l(\omega_0)$ for all frequencies and model orders.

Agenda

Computation Time vs. Model Order

MATLAB Implementation

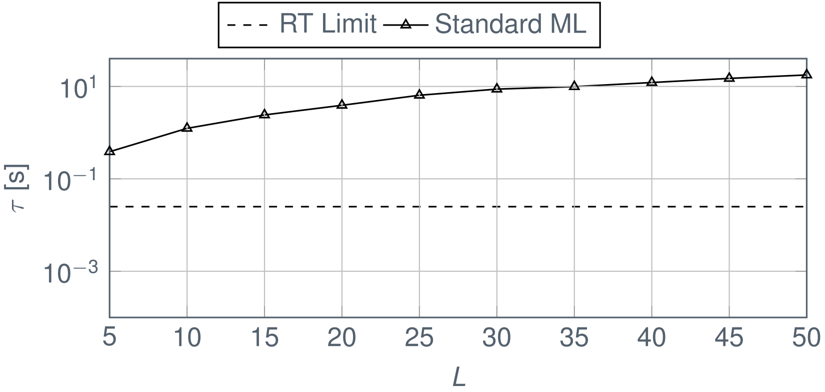
Setup: $N = 200$ (25 ms @ $f_s = 8000$ Hz), $F = 5NL$, T420 laptop



Computation Time vs. Model Order

MATLAB Implementation

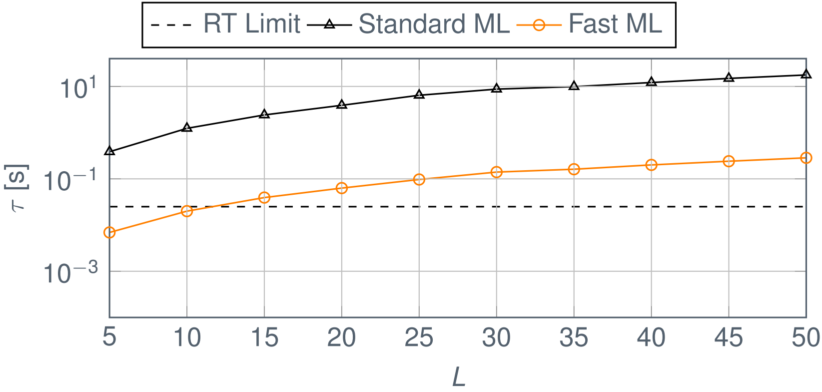
Setup: $N = 200$ (25 ms @ $f_s = 8000$ Hz), $F = 5NL$, T420 laptop



Computation Time vs. Model Order

MATLAB Implementation

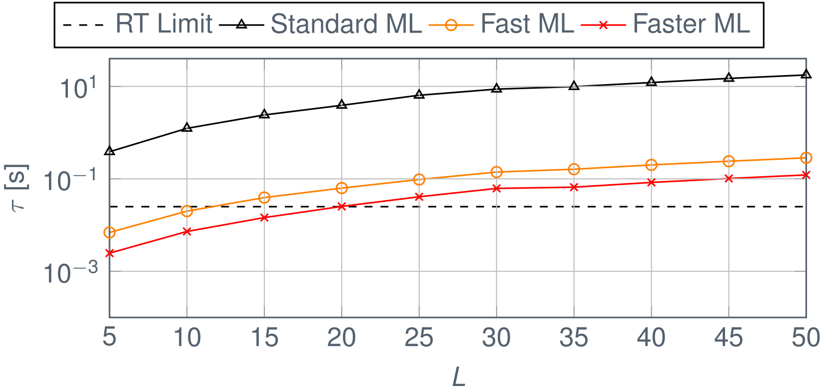
Setup: $N = 200$ (25 ms @ $f_s = 8000$ Hz), $F = 5NL$, T420 laptop



Computation Time vs. Model Order

MATLAB Implementation

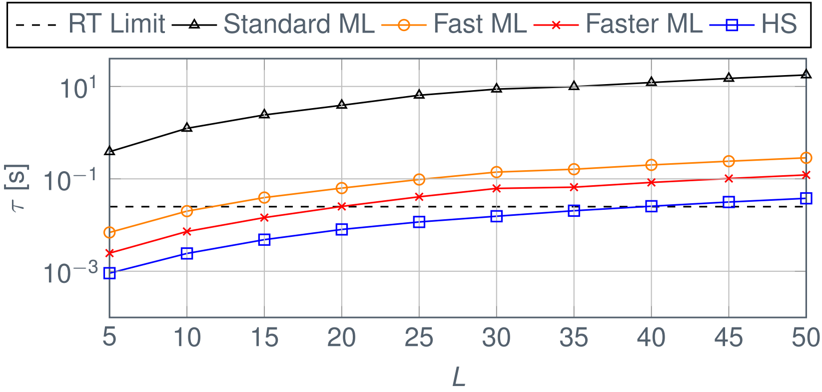
Setup: $N = 200$ (25 ms @ $f_s = 8000$ Hz), $F = 5NL$, T420 laptop



Computation Time vs. Model Order

MATLAB Implementation

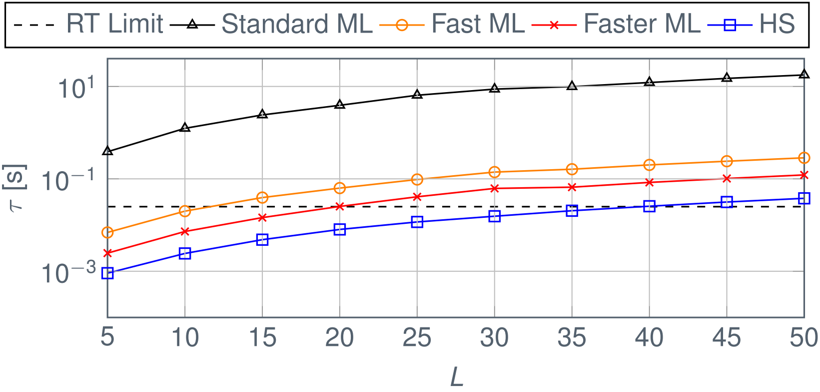
Setup: $N = 200$ (25 ms @ $f_s = 8000$ Hz), $F = 5NL$, T420 laptop



Computation Time vs. Model Order

MATLAB Implementation

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

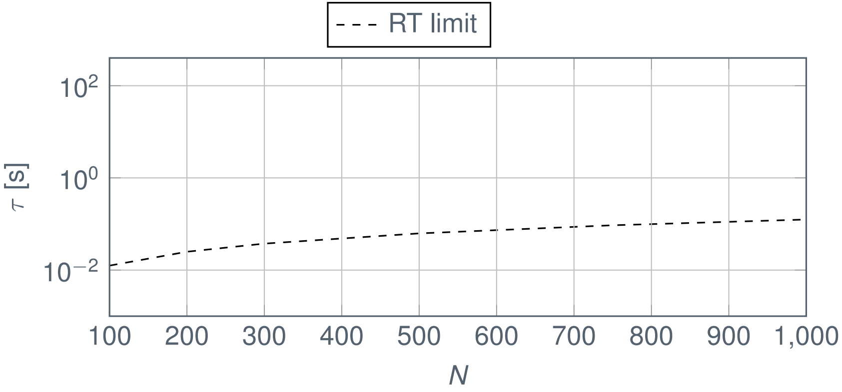


$\tau(\text{Standard ML}) \approx 60\tau(\text{Fast ML}) \approx 150\tau(\text{Faster ML}) \approx 500\tau(\text{HS})$

Computation Time vs. Data Size

MATLAB Implementation

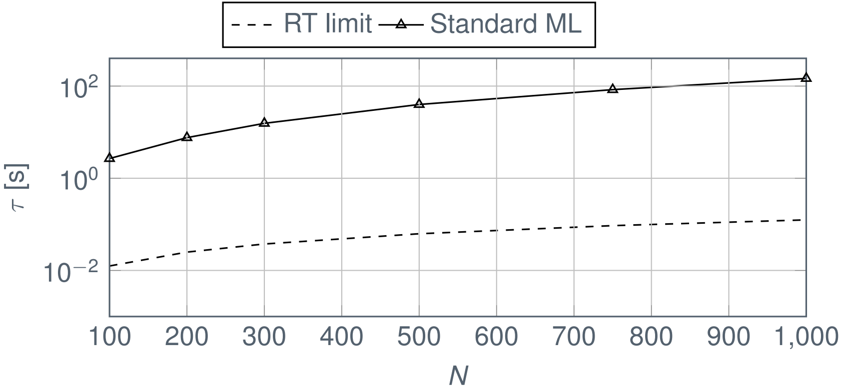
Setup: $L = 30$, $F = 5NL$, T420 laptop



Computation Time vs. Data Size

MATLAB Implementation

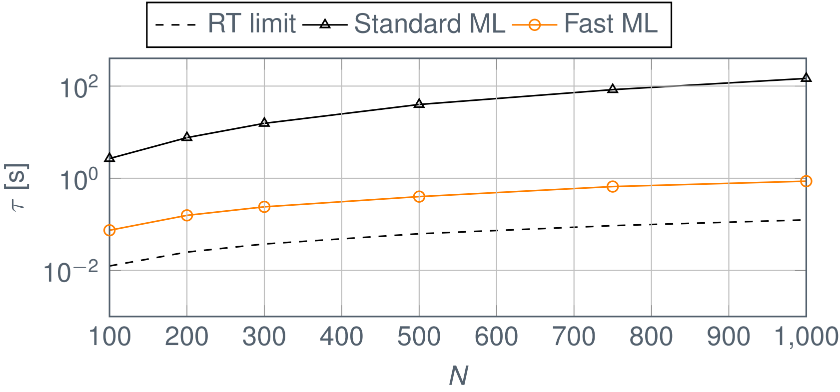
Setup: $L = 30$, $F = 5NL$, T420 laptop



Computation Time vs. Data Size

MATLAB Implementation

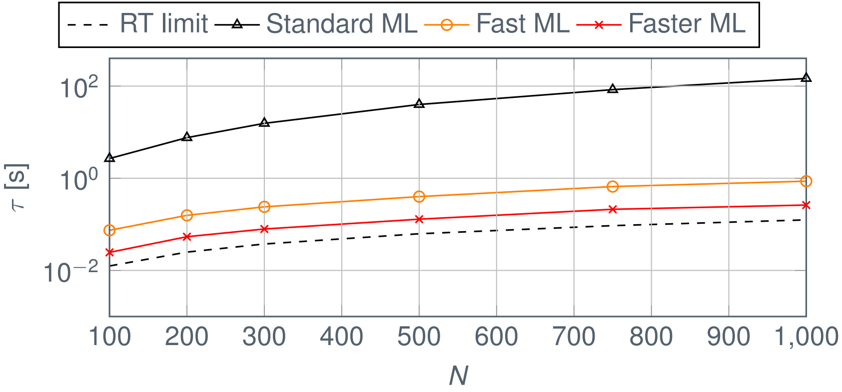
Setup: $L = 30$, $F = 5NL$, T420 laptop



Computation Time vs. Data Size

MATLAB Implementation

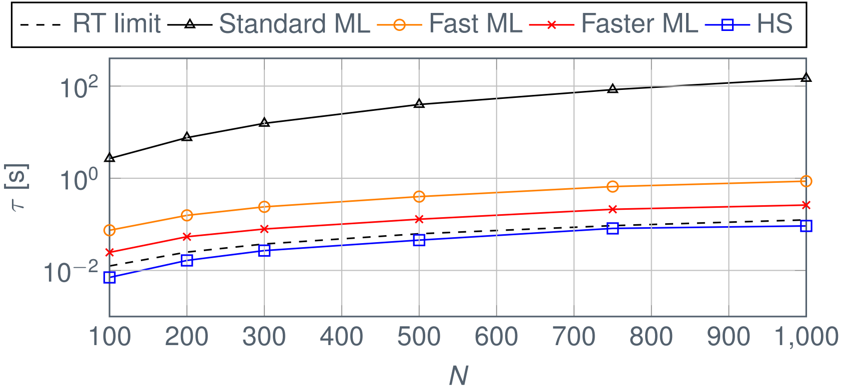
Setup: $L = 30$, $F = 5NL$, T420 laptop



Computation Time vs. Data Size

MATLAB Implementation

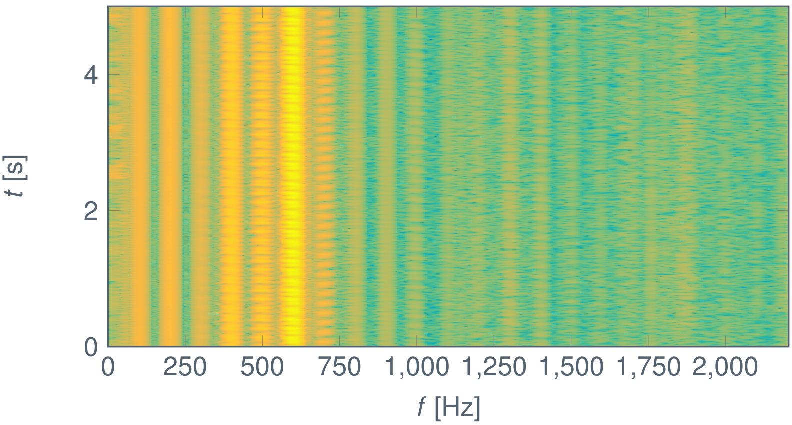
Setup: $L = 30$, $F = 5NL$, T420 laptop



Washing Machine Example

Acoustic measurements by Brüel & Kjær

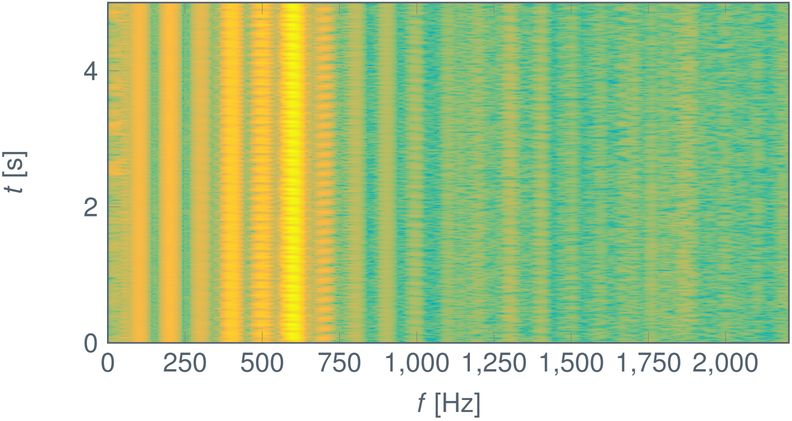
- $f_s = 44.1$ kHz



Washing Machine Example

Acoustic measurements by Brüel & Kjær

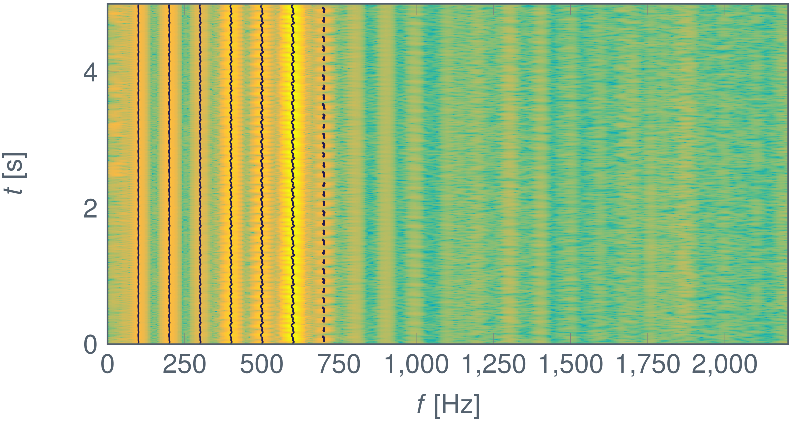
- $f_s = 44.1$ kHz
- $f_{rs} = 4410$ Hz, 60 ms windows, 15/16 overlap, and $L = 15$



Washing Machine Example

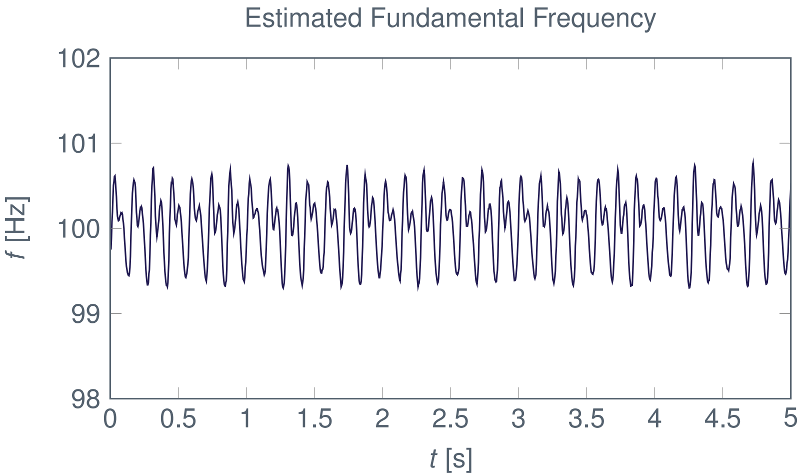
Acoustic measurements by Brüel & Kjær

- $f_s = 44.1$ kHz   Computation time: 28 s (50 % overlap: 3.8 s)
- $f_{rs} = 4410$ Hz, 60 ms windows, 15/16 overlap, and $L = 15$

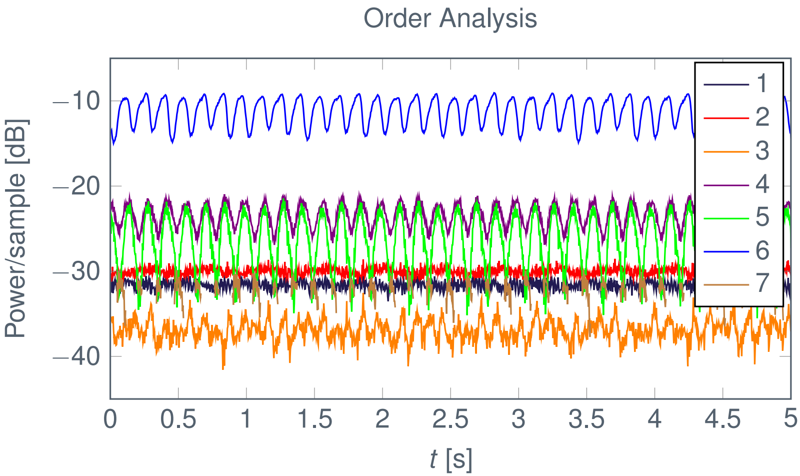


Washing Machine Example

Acoustic measurements by Brüel & Kjær



Washing Machine Example
Acoustic measurements by Brüel & Kjær



Agenda

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

$$\mathcal{O}(F \log F) + \mathcal{O}(FL^3) \tag{18}$$

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

$$\mathcal{O}(F \log F) + \mathcal{O}(FL^3) \quad (18)$$

  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

$$\mathcal{O}(F \log F) + \mathcal{O}(FL) \quad (19)$$

  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

$$\mathcal{O}(F \log F) + \mathcal{O}(FL^3) \tag{18}$$

  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

$$\mathcal{O}(F \log F) + \mathcal{O}(FL) \tag{19}$$

  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!

AALBORG UNIVERSITY
DENMARK