Fast methods for low dimensional non-linear least-squares optimization

CORTOPT 2016

August 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

[1]Aalborg University
Dept. of Electronic Systems

[2]Bang & Olufsen A/S
Denmark

[3]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

Introduction
- Grid-size Selection in the Scalar Case

Fundamental Frequency Estimation
- Background on Fundamental Frequency Estimation
- Fast NLS
- Results
- Summary and Demo

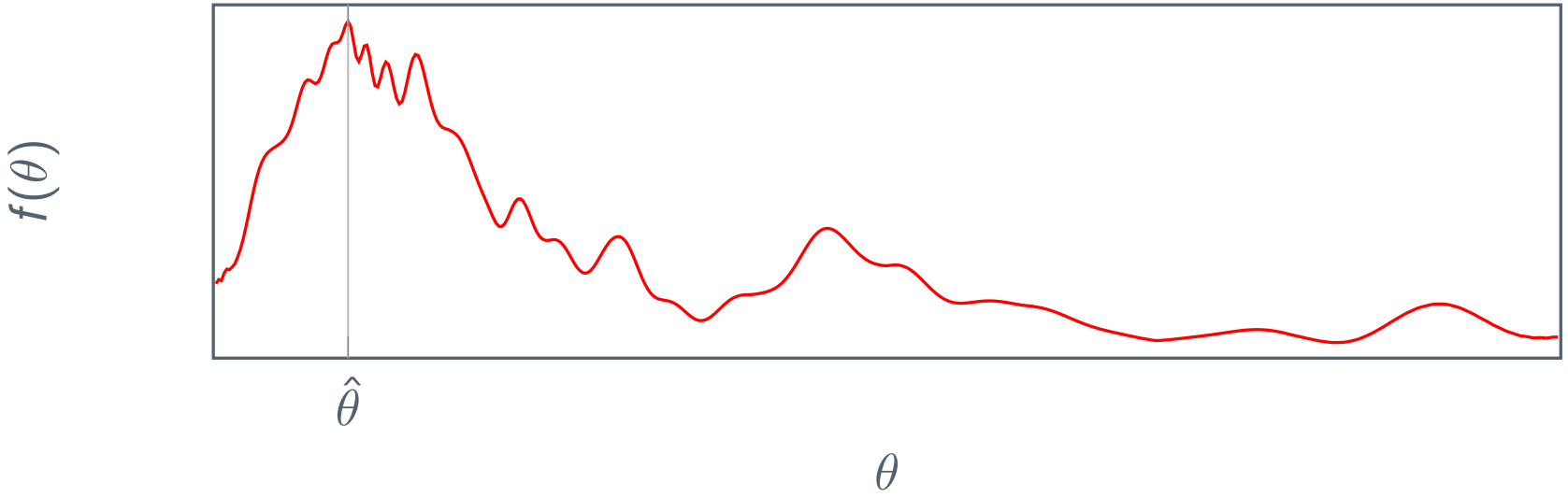
Source Localisation
- Grid Size Selection in the Diagonal Case
- Demo

Summary

Introduction

Problem

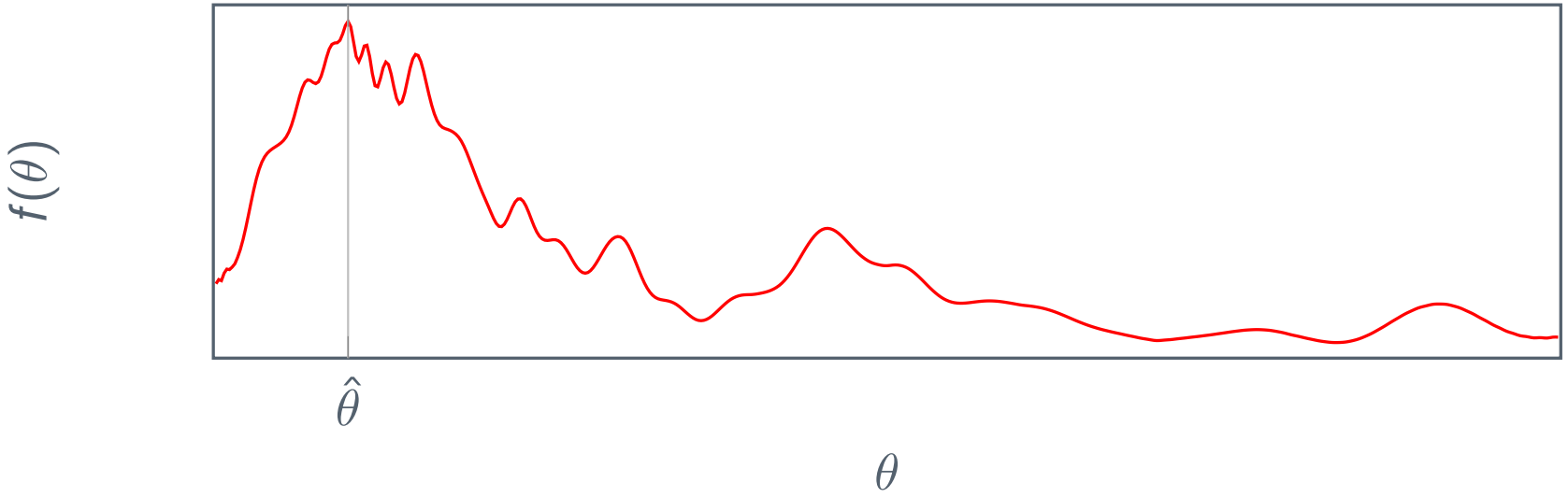
$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\operatorname{argmax}}\, f(\boldsymbol{\theta}) \tag{1}$$



Introduction

Problem

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\operatorname{argmax}}\, f(\boldsymbol{\theta}) \tag{1}$$



How can we solve this problem - fast?

Introduction

Goals of any solution

1. As few cost function evaluations as possible (to solve the problem)
2. Each cost function evaluation should be as cheap as possible

Introduction

Goals of any solution

1. As few cost function evaluations as possible (to solve the problem)
2. Each cost function evaluation should be as cheap as possible

Possible solutions

- Interval Analysis
- Branch and bound methods
- Approximate methods
- Grid method (with refinement)
  - Conceptually simple
  - FFT can often be used

Introduction
Example: Single complex-valued sinusoid

Signal Model

$$x(n) = \alpha \exp(j\omega n) + e(n) \ , \quad \text{for } n = 0, 1, \cdots, N-1 \tag{2}$$

In vector form

$$\boldsymbol{x} = \boldsymbol{z}(\omega)\alpha + \boldsymbol{e} \tag{3}$$

Introduction

Example: Single complex-valued sinusoid

Signal Model

$$x(n) = \alpha \exp(j\omega n) + e(n)\ , \quad \text{for } n = 0, 1, \cdots, N-1 \tag{2}$$

In vector form

$$\boldsymbol{x} = \boldsymbol{z}(\omega)\alpha + \boldsymbol{e} \tag{3}$$

Nonlinear least squares (NLS) estimator

$$\hat{\omega} = \underset{\omega \in [0, 2\pi)}{\operatorname{argmax}}\, f(\omega) \tag{4}$$

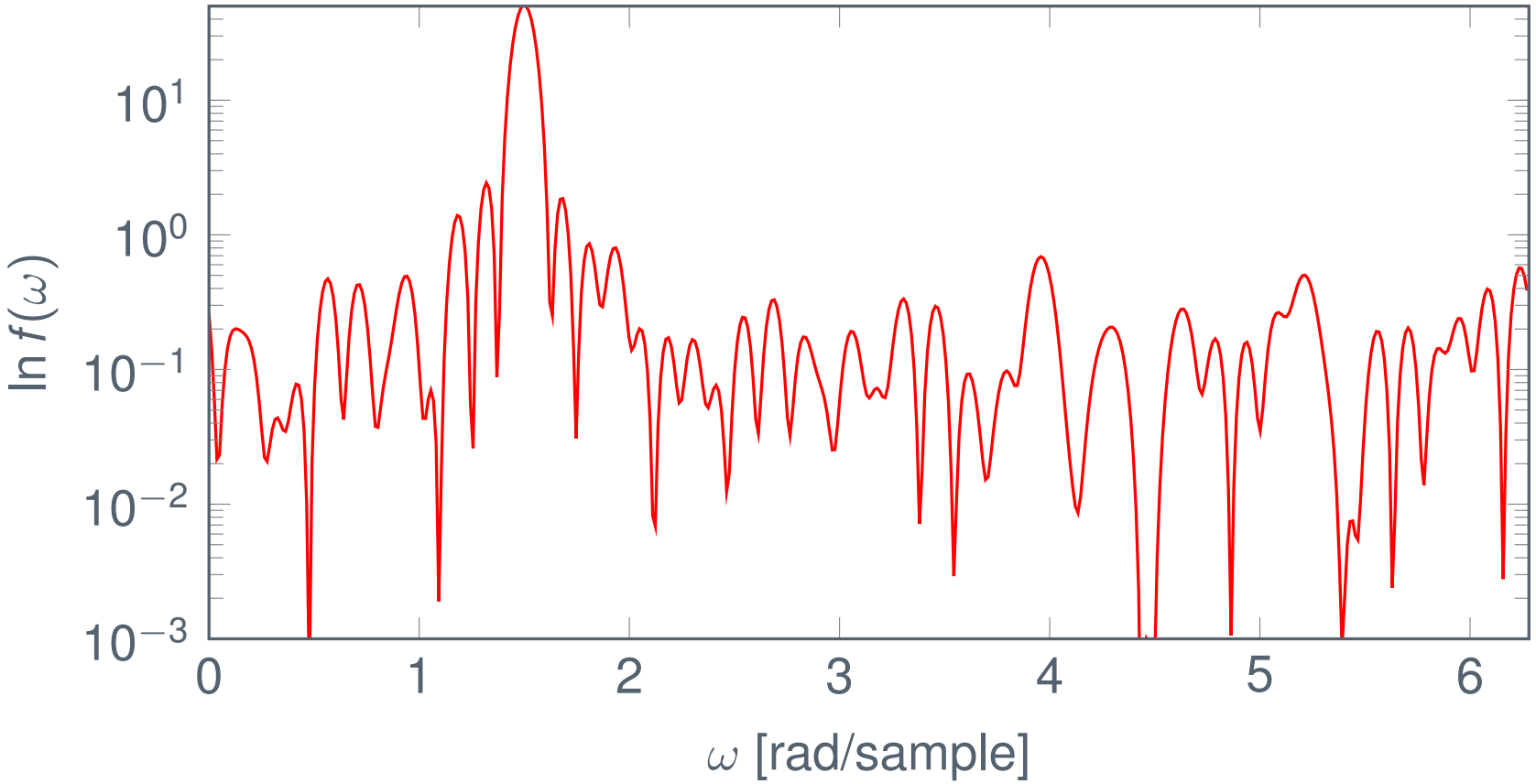
where

$$f(\omega) = \boldsymbol{x}^T \boldsymbol{z}(\omega) \left[\boldsymbol{z}^T(\omega)\boldsymbol{z}(\omega)\right]^{-1} \boldsymbol{z}^T(\omega)\boldsymbol{x} = \frac{1}{N}\left|\sum_{n=0}^{N-1} x(n)\mathrm{e}^{-j\omega n}\right|^2 \tag{5}$$

Introduction

Example: Single complex-valued sinusoid

Signal-to-noise ratio (SNR) of 10 dB.



Introduction

How should we select the grid size?

1. Select according to the Cramér-Rao lower bound (CRLB) at some SNR or desired accuracy (many cost function evaluations)

Introduction

How should we select the grid size?

1. Select according to the Cramér-Rao lower bound (CRLB) at some SNR or desired accuracy (many cost function evaluations)
2. Find a coarse estimate on a grid and refine it by exploiting local concavity

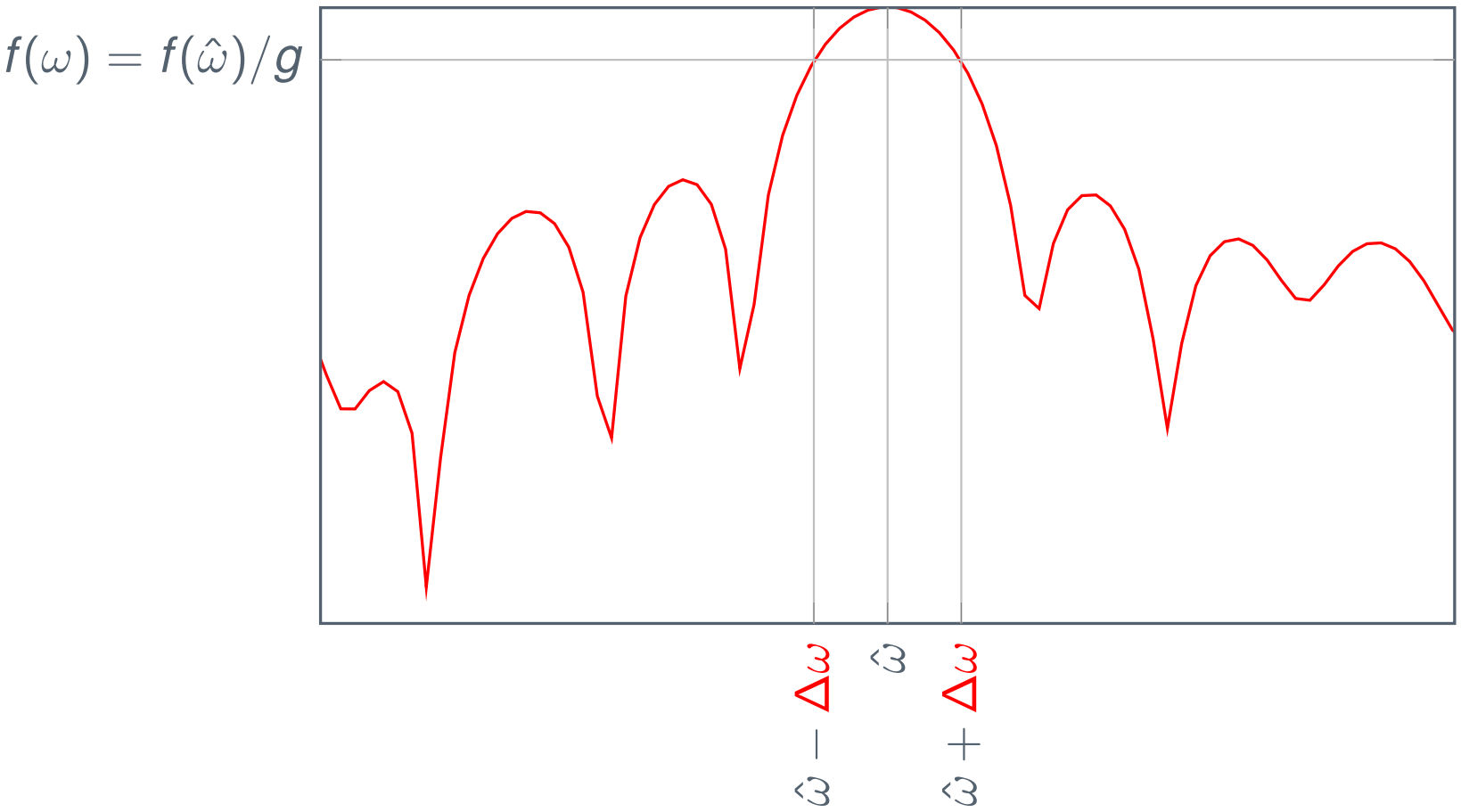
Introduction

How should we select the grid size?

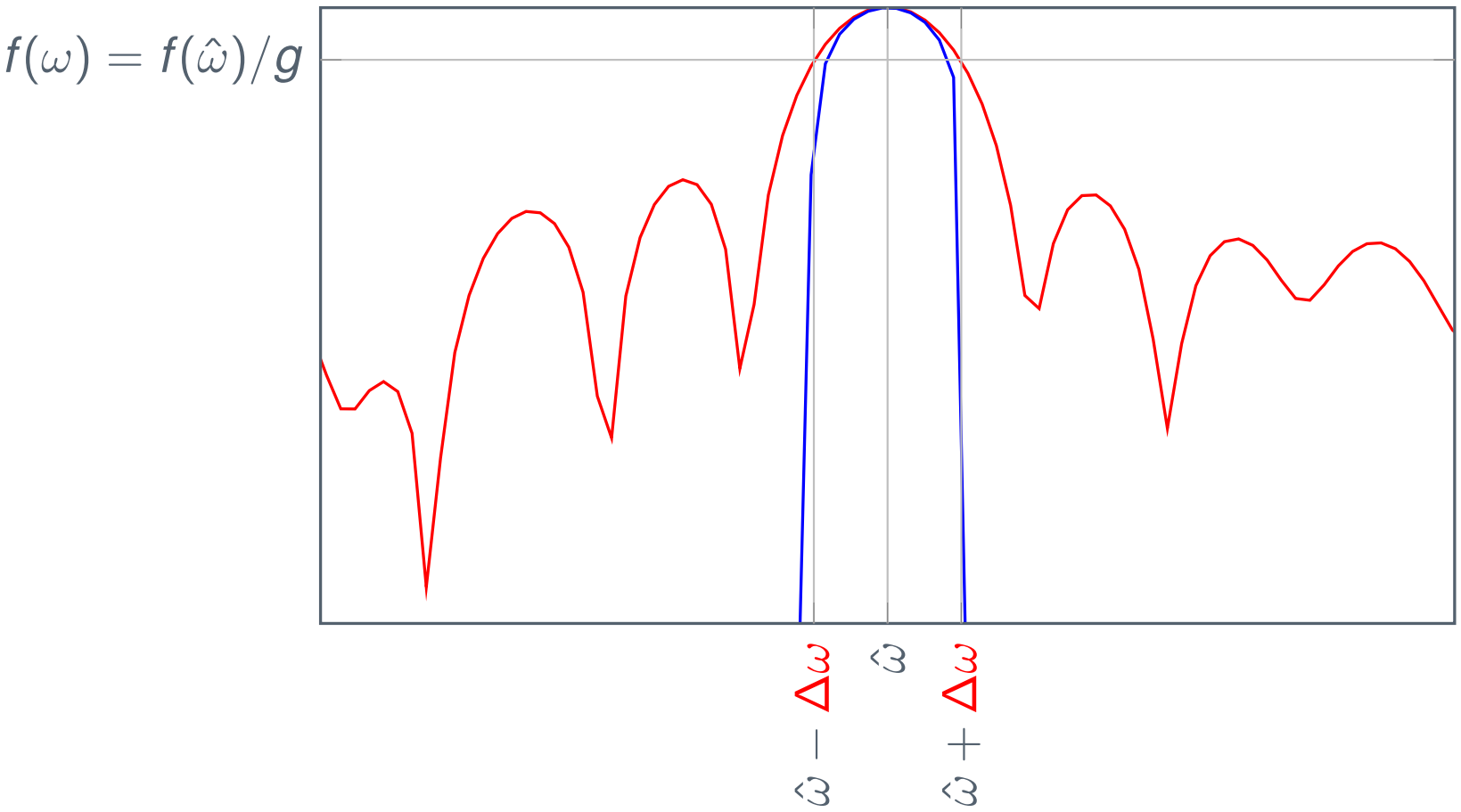
1. Select according to the Cramér-Rao lower bound (CRLB) at some SNR or desired accuracy (many cost function evaluations)
2. Find a coarse estimate on a grid and refine it by exploiting local concavity

But how coarse can the grid be?

Grid-size Selection in the Scalar Case



Grid-size Selection in the Scalar Case



Grid-size Selection in the Scalar Case

Approach

- Let $f : S \mapsto \mathbb{R}$ in C^2

Grid-size Selection in the Scalar Case

Approach

- Let $f : S \mapsto \mathbb{R}$ in C^2
- Let $\hat{\theta} \in S \subseteq \mathbb{R}$ be a global maximiser

Grid-size Selection in the Scalar Case

Approach

- Let $f : S \mapsto \mathbb{R}$ in C^2
- Let $\hat{\theta} \in S \subseteq \mathbb{R}$ be a global maximiser
- Let $f'(\hat{\theta}) = 0$ (a boundary point is not a solution).

Grid-size Selection in the Scalar Case

Approach

- Let $f : S \mapsto \mathbb{R}$ in C^2
- Let $\hat{\theta} \in S \subseteq \mathbb{R}$ be a global maximiser
- Let $f'(\hat{\theta}) = 0$ (a boundary point is not a solution).
- Make a second order Taylor approximation around the mode

$$f(\theta) \approx f(\hat{\theta}) + \frac{1}{2}(\theta - \hat{\theta})^T H(\hat{\theta})(\theta - \hat{\theta}) \tag{6}$$

where $H(\hat{\theta}) = \nabla^2 f(\theta)$ is the Hessian.

Grid-size Selection in the Scalar Case

Approach

For $g > 1$, select the grid size $\Delta\theta$ so that

$$f(\theta) = \frac{f(\hat{\theta})}{g} \approx f(\hat{\theta}) + \frac{1}{2}(\theta - \hat{\theta})^T H(\hat{\theta})(\theta - \hat{\theta}) \tag{7}$$

Grid-size Selection in the Scalar Case

Approach

For $g > 1$, select the grid size $\Delta\theta$ so that

$$f(\theta) = \frac{f(\hat{\theta})}{g} \approx f(\hat{\theta}) + \frac{1}{2}(\theta - \hat{\theta})^T H(\hat{\theta})(\theta - \hat{\theta}) \tag{7}$$

Solving for θ gives

$$\theta = \hat{\theta} \pm \Delta\theta = \hat{\theta} \pm \sqrt{2\frac{1-g}{g}\frac{f(\hat{\theta})}{H(\hat{\theta})}} \tag{8}$$

Grid-size Selection in the Scalar Case

Example: Single complex-valued sinusoid

We have that

$$f(\hat{\omega}) = \frac{1}{N} \left| \boldsymbol{z}^T(\hat{\omega}) \boldsymbol{x} \right|^2 \tag{9}$$

$$H(\hat{\omega}) \approx -\frac{N}{6} \left| \boldsymbol{z}^T(\hat{\omega}) \boldsymbol{x} \right|^2 \tag{10}$$

Grid-size Selection in the Scalar Case

Example: Single complex-valued sinusoid

We have that

$$f(\hat{\omega}) = \frac{1}{N}\left|\boldsymbol{z}^T(\hat{\omega})\boldsymbol{x}\right|^2 \tag{9}$$

$$H(\hat{\omega}) \approx -\frac{N}{6}\left|\boldsymbol{z}^T(\hat{\omega})\boldsymbol{x}\right|^2 \tag{10}$$

Thus,

$$\Delta\omega = \sqrt{2\frac{1-g}{g}\frac{f(\hat{\omega})}{H(\hat{\omega})}} \approx \sqrt{12\frac{g-1}{g}}\frac{1}{N} \tag{11}$$

Grid-size Selection in the Scalar Case

Example: Single complex-valued sinusoid

We have that

$$f(\hat{\omega}) = \frac{1}{N}\left|\boldsymbol{z}^T(\hat{\omega})\boldsymbol{x}\right|^2 \tag{9}$$

$$H(\hat{\omega}) \approx -\frac{N}{6}\left|\boldsymbol{z}^T(\hat{\omega})\boldsymbol{x}\right|^2 \tag{10}$$

Thus,

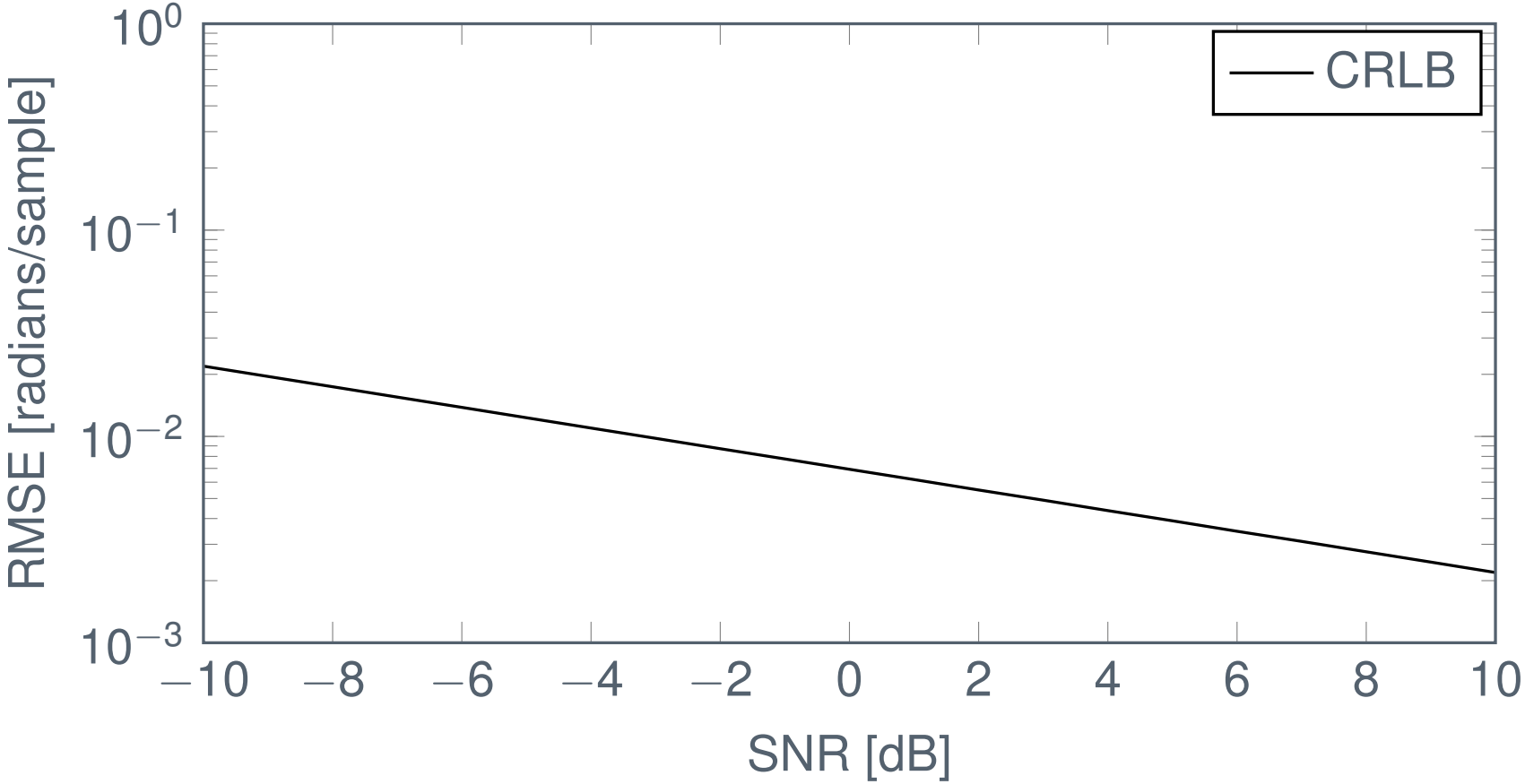
$$\Delta\omega = \sqrt{2\frac{1-g}{g}\frac{f(\hat{\omega})}{H(\hat{\omega})}} \approx \sqrt{12\frac{g-1}{g}}\frac{1}{N} \tag{11}$$

How do we select the value for g?

Grid-size Selection in the Scalar Case

Example: Single complex-valued sinusoid

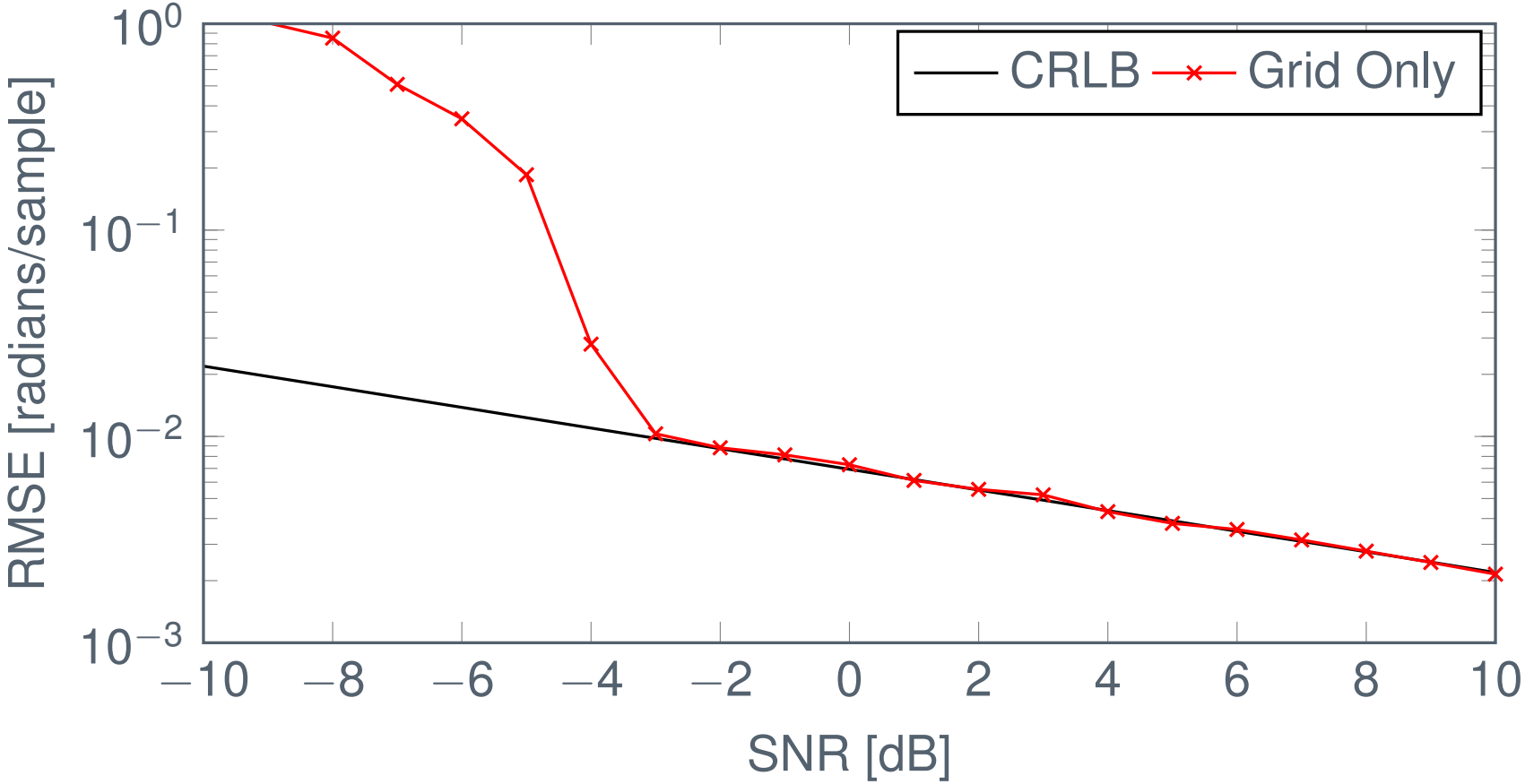
Setup: $N = 50$, 1000 Monte Carlo runs for each g and SNR, and a Fibonacci search for refinement.



Grid-size Selection in the Scalar Case

Example: Single complex-valued sinusoid

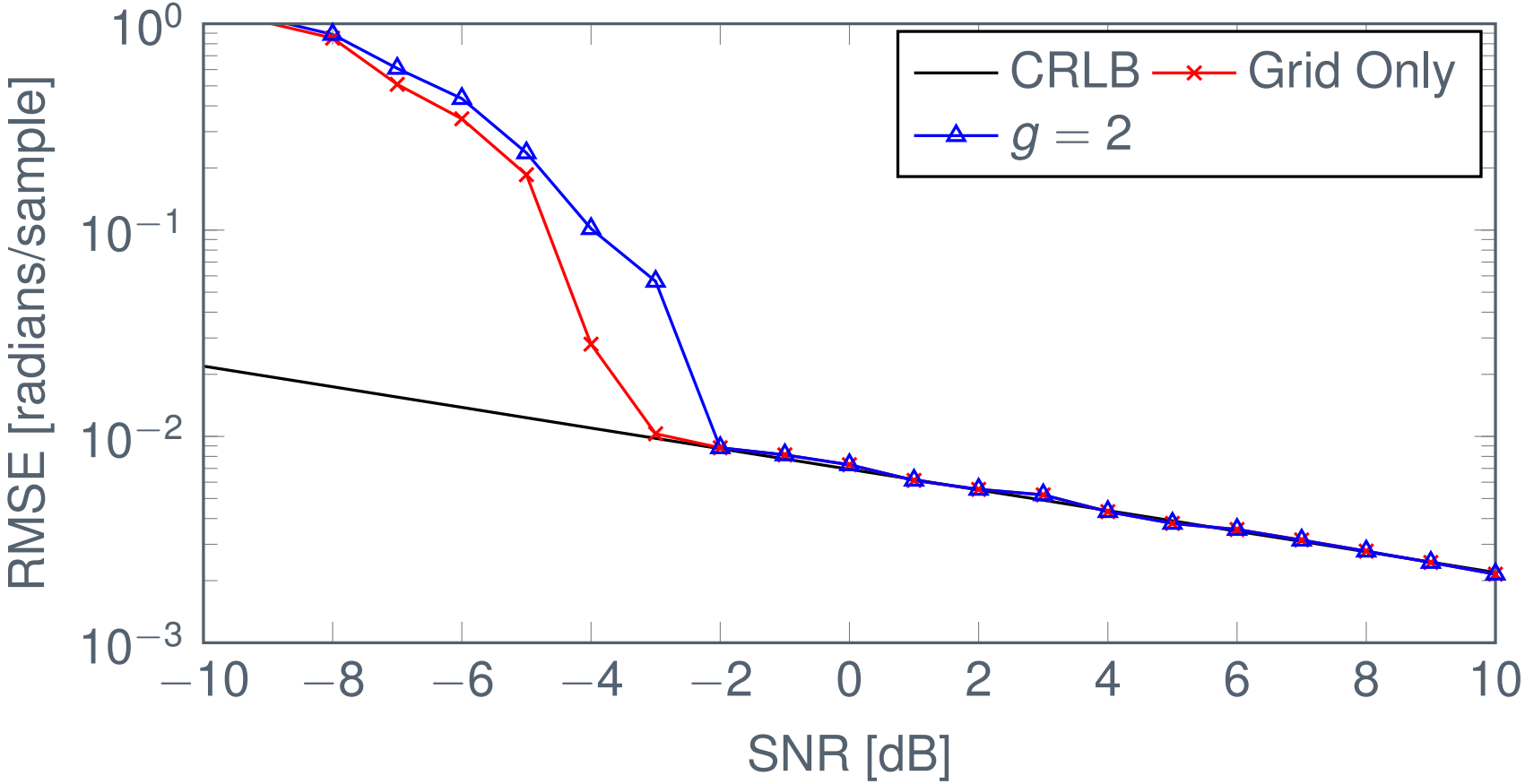
Setup: $N = 50$, 1000 Monte Carlo runs for each g and SNR, and a Fibonacci search for refinement.



Grid-size Selection in the Scalar Case

Example: Single complex-valued sinusoid

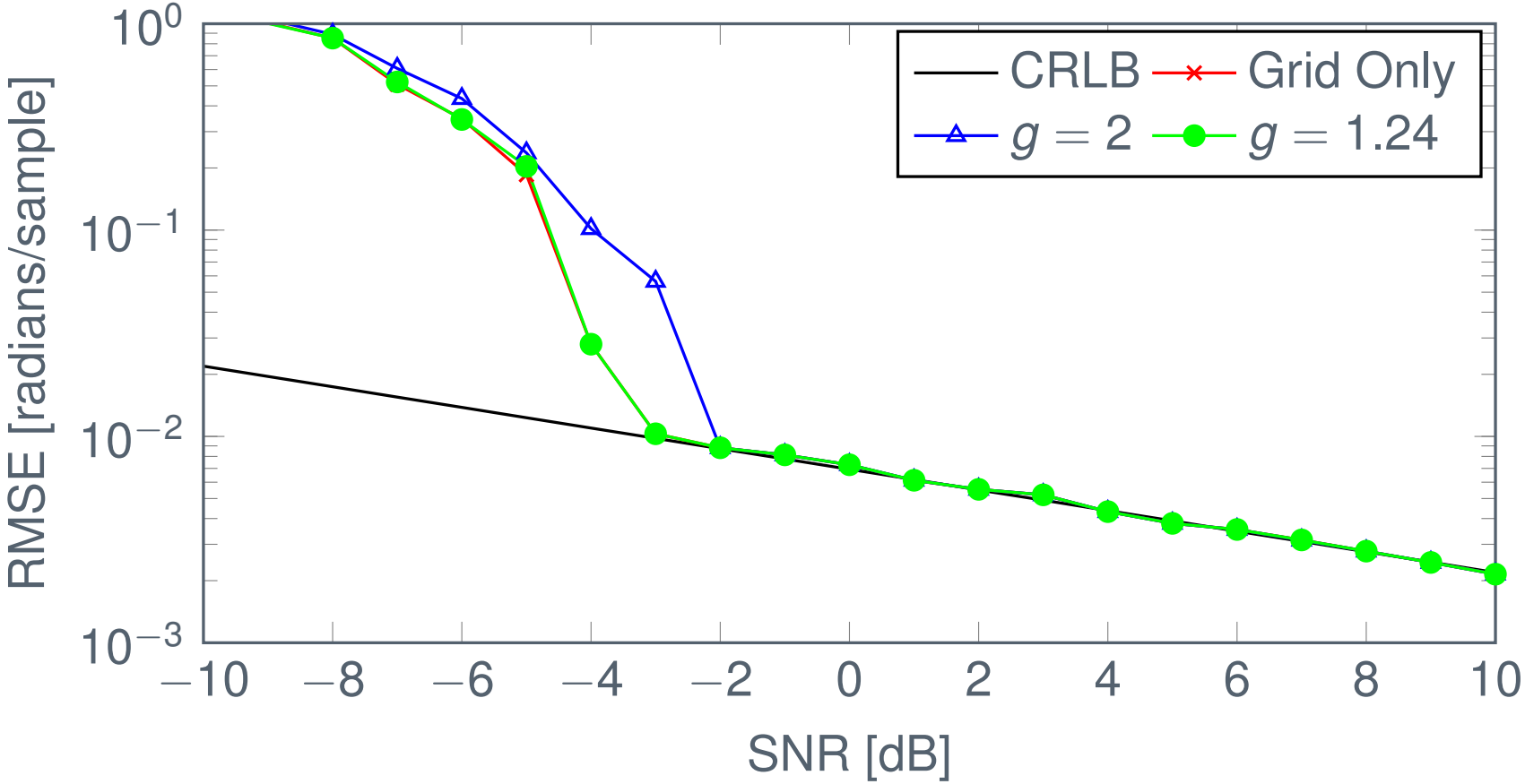
Setup: $N = 50$, 1000 Monte Carlo runs for each g and SNR, and a Fibonacci search for refinement.



Grid-size Selection in the Scalar Case

Example: Single complex-valued sinusoid

Setup: $N = 50$, 1000 Monte Carlo runs for each g and SNR, and a Fibonacci search for refinement.



Grid-size Selection in the Scalar Case

Example: Single complex-valued sinusoid

g	2	1.24	1.024	1.0024	1.00024	Grid Only
# grid points	65	100	301	946	2989	28673
τ [ms]	1.27	1.25	1.21	1.19	1.27	3.00

Grid-size Selection in the Scalar Case

Example: Single complex-valued sinusoid

g	2	1.24	1.024	1.0024	1.00024	Grid Only
# grid points	65	100	301	946	2989	28673
τ [ms]	1.27	1.25	1.21	1.19	1.27	3.00

The grid only approach is approximately 2.5 times slower than the combined approach for $g = 1.24$.

Agenda

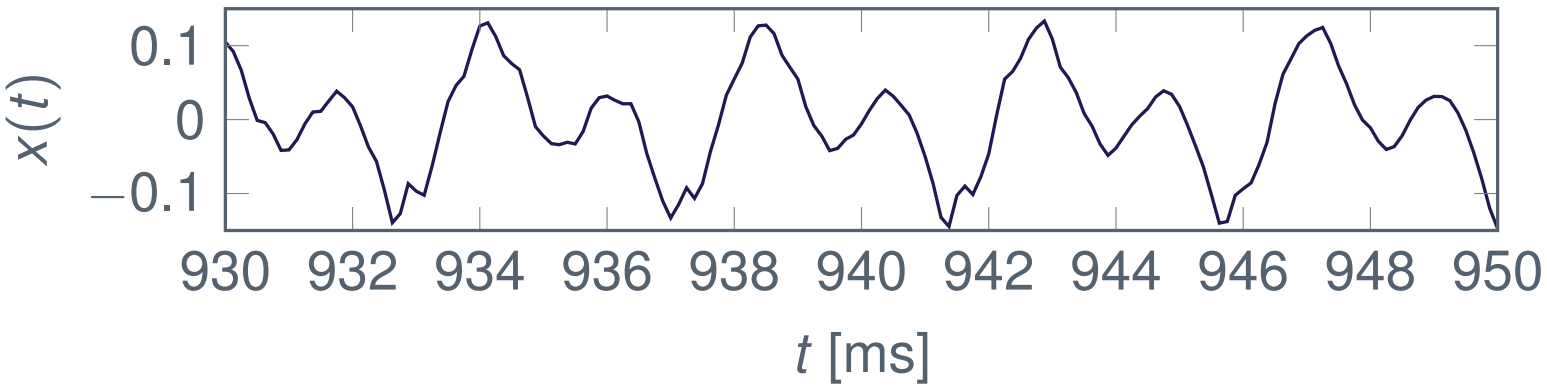
Introduction
- Grid-size Selection in the Scalar Case

Fundamental Frequency Estimation
- Background on Fundamental Frequency Estimation
- Fast NLS
- Results
- Summary and Demo

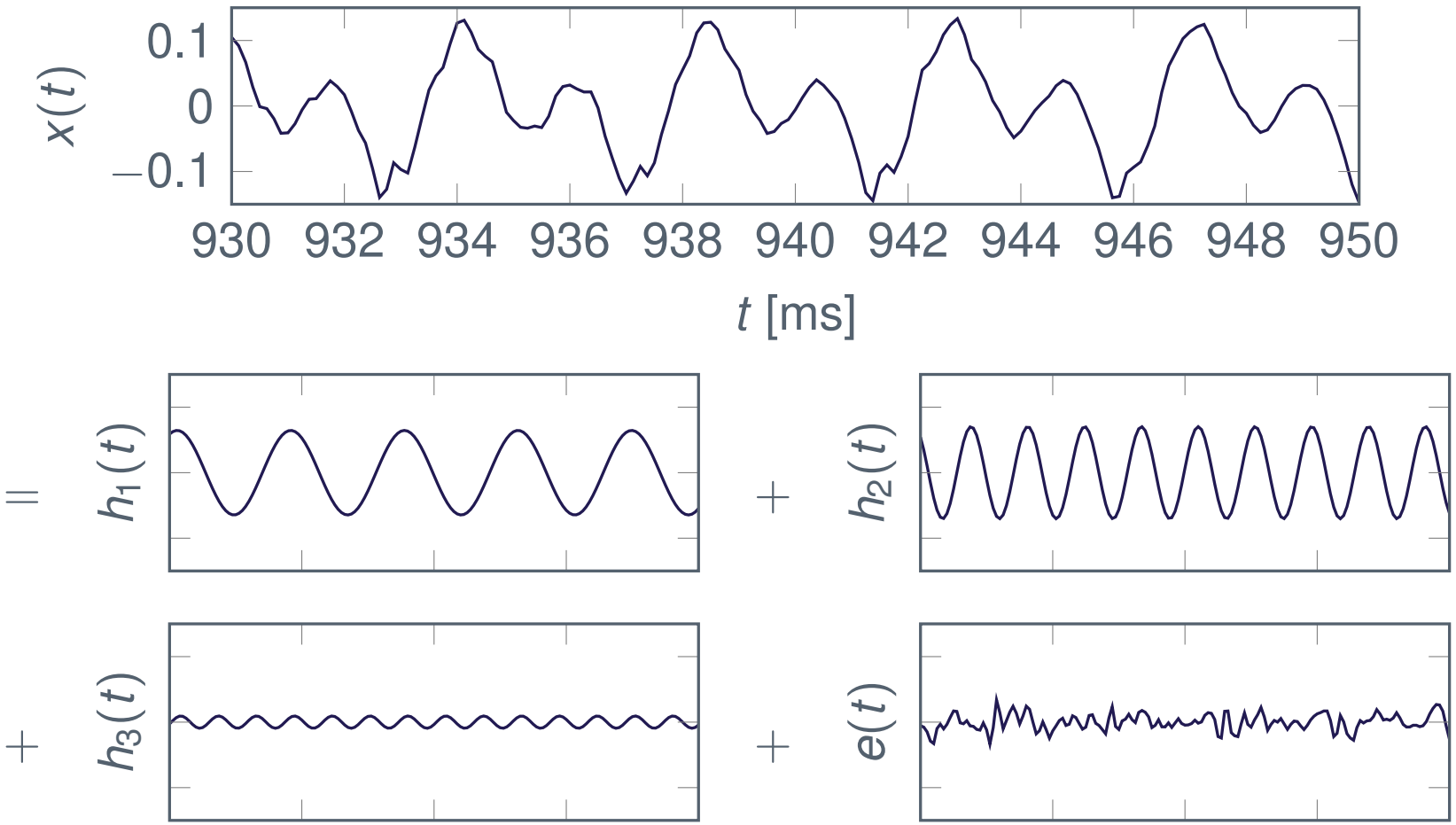
Source Localisation
- Grid Size Selection in the Diagonal Case
- Demo

Summary

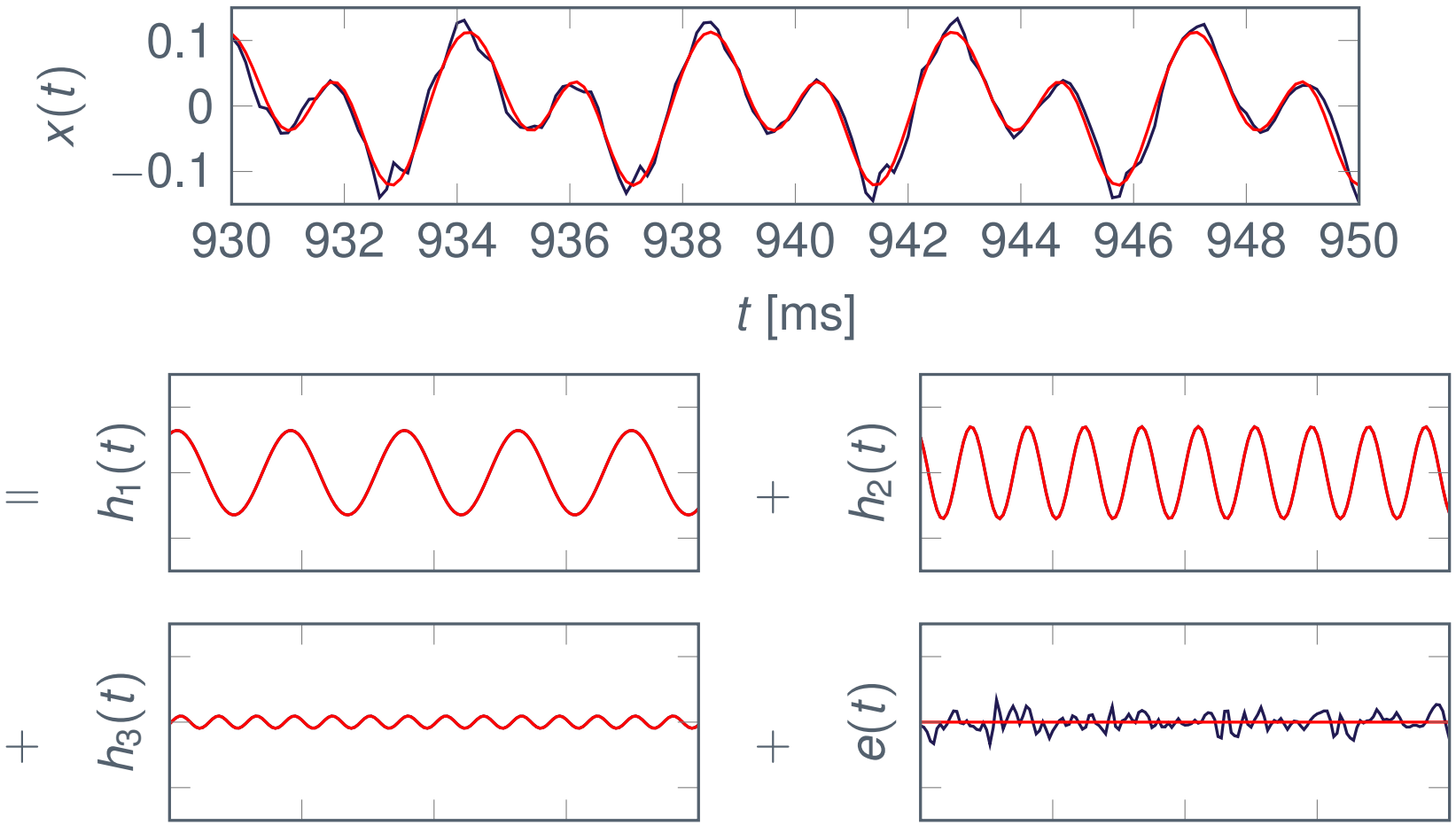
Speech Signal Example



Speech Signal Example



Speech Signal Example



Mathematical Model

Mathematical Model

$$x(t) = \sum_{i=1}^{I} h_i(t) + e(t) = \sum_{i=1}^{I} A_i \cos(2\pi f_0 i t + \phi_i) + e(t) \qquad (12)$$

where

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

Mathematical Model

Mathematical Model

$$x(t) = \sum_{i=1}^{I} h_i(t) + e(t) = \sum_{i=1}^{I} A_i \cos(2\pi f_0 i t + \phi_i) + e(t) \qquad (12)$$

where

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

Analysis problem: Get f_0 (and I) from the data.

Estimation Methods

Correlation Methods

A periodic signal satisfies that

$$x(t) = x(t + T) \tag{13}$$

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

$$x(t) = x(t + T) \tag{13}$$

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

$$x(t) = x(t + T) \tag{13}$$

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

$$x(t) = \sum_{i=1}^{I} A_i \cos(i2\pi f_0 t + \phi_i) + e(t) \tag{14}$$

$$= \sum_{i=1}^{I} \left[a_i \cos(i2\pi f_0 t) - b_i \sin(i2\pi f_0 t) \right] + e(t) \tag{15}$$

Estimation Methods

Parametric Methods

Estimate the parameters in

$$x(t) = \sum_{i=1}^{I} A_i \cos(i2\pi f_0 t + \phi_i) + e(t) \tag{14}$$

$$= \sum_{i=1}^{I} \left[a_i \cos(i2\pi f_0 t) - b_i \sin(i2\pi f_0 t) \right] + e(t) \tag{15}$$

+ 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

$$x(t) = \sum_{i=1}^{I} A_i \cos(i2\pi f_0 t + \phi_i) + e(t) \tag{14}$$

$$= \sum_{i=1}^{I} \left[a_i \cos(i2\pi f_0 t) - b_i \sin(i2\pi f_0 t) \right] + e(t) \tag{15}$$

+ 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

The sampled signal model

$$x(n) = \sum_{i=1}^{I} \left[a_i \cos(i\omega_0 n) - b_i \sin(i\omega_0 n) \right] + e(n) \tag{16}$$

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

$$\boldsymbol{x} = \boldsymbol{Z}_I(\omega_0)\alpha_I + \boldsymbol{e}\,. \tag{17}$$

where

$$\begin{aligned}
\boldsymbol{Z}_I(\omega) &= \begin{bmatrix} \boldsymbol{c}(\omega) & \boldsymbol{c}(2\omega) & \cdots & \boldsymbol{c}(I\omega) & \boldsymbol{s}(\omega) & \boldsymbol{s}(2\omega) & \cdots & \boldsymbol{s}(I\omega) \end{bmatrix} \\
\boldsymbol{c}(\omega) &= \begin{bmatrix} \cos(\omega n_0) & \cdots & \cos(\omega(n_0 + N - 1)) \end{bmatrix}^T \\
\boldsymbol{s}(\omega) &= \begin{bmatrix} \sin(\omega n_0) & \cdots & \sin(\omega(n_0 + N - 1)) \end{bmatrix}^T \\
\alpha_I &= \begin{bmatrix} \boldsymbol{a}_I^T & -\boldsymbol{b}_I^T \end{bmatrix}^T,\ \boldsymbol{a}_I = \begin{bmatrix} a_1 & \cdots & a_I \end{bmatrix}^T,\ \boldsymbol{b}_I = \begin{bmatrix} b_1 & \cdots & b_I \end{bmatrix}^T\,.
\end{aligned}$$

Maximum Likelihood Estimation

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

$$\hat{\omega}_0 = \underset{\omega_0}{\operatorname{argmax}}\, \boldsymbol{x}^T \boldsymbol{Z}_l(\omega_0) \left[\boldsymbol{Z}_l^T(\omega_0)\boldsymbol{Z}_l(\omega_0)\right]^{-1} \boldsymbol{Z}_l^T(\omega_0)\boldsymbol{x} \tag{18}$$

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

Maximum Likelihood Estimation

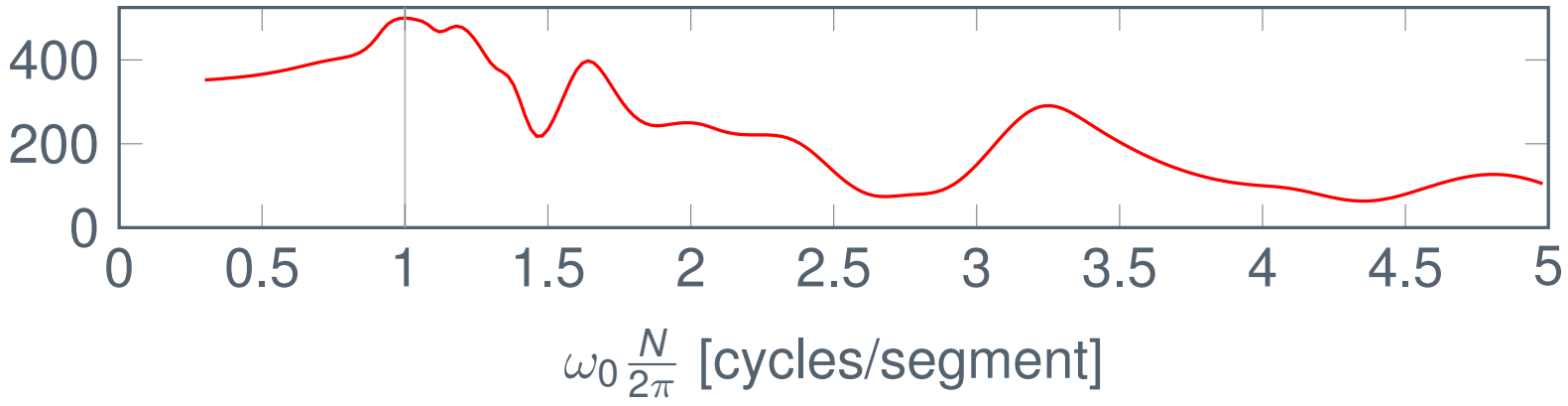
The maximum likelihood estimate (MLE) for the fundamental frequency is

$$\hat{\omega}_0 = \underset{\omega_0}{\operatorname{argmax}}\, \boldsymbol{x}^T \boldsymbol{Z}_l(\omega_0) \left[\boldsymbol{Z}_l^T(\omega_0)\boldsymbol{Z}_l(\omega_0)\right]^{-1} \boldsymbol{Z}_l^T(\omega_0)\boldsymbol{x} \tag{18}$$

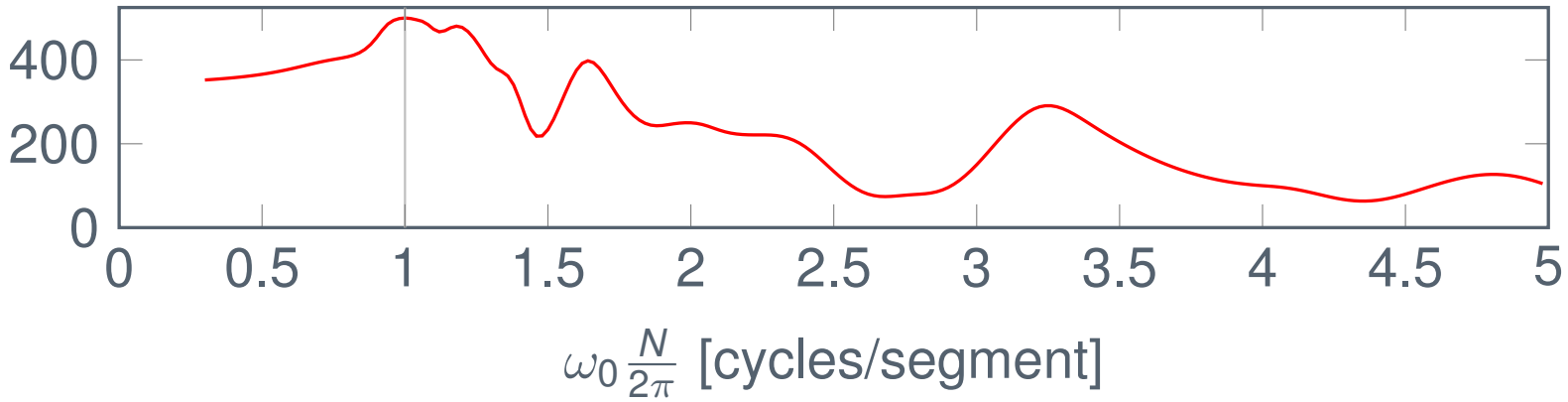
and is also known as the nonlinear least squares (NLS) estimate.

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

Maximum Likelihood Estimation



Maximum Likelihood Estimation

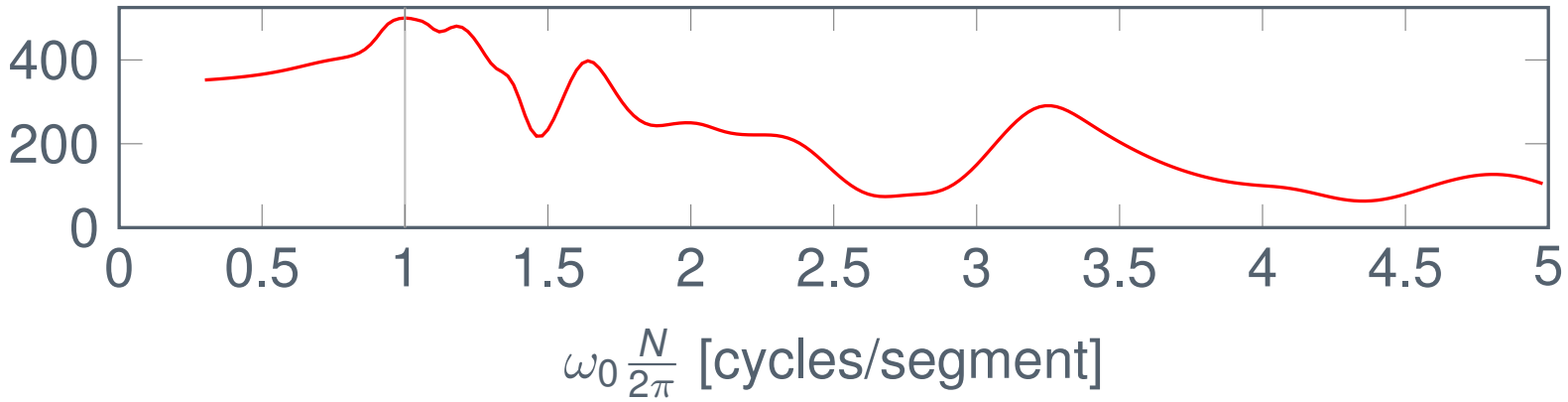


1. Compute NLS cost function

$$\hat{\omega}_0 = \underset{\omega_0 \in (0, \pi/l)}{\operatorname{argmax}} \; \boldsymbol{x}^T \boldsymbol{Z}_l(\omega_0) \left[\boldsymbol{Z}_l^T(\omega_0) \boldsymbol{Z}_l(\omega_0) \right]^{-1} \boldsymbol{Z}_l^T(\omega_0) \boldsymbol{x} \tag{19}$$

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

Maximum Likelihood Estimation

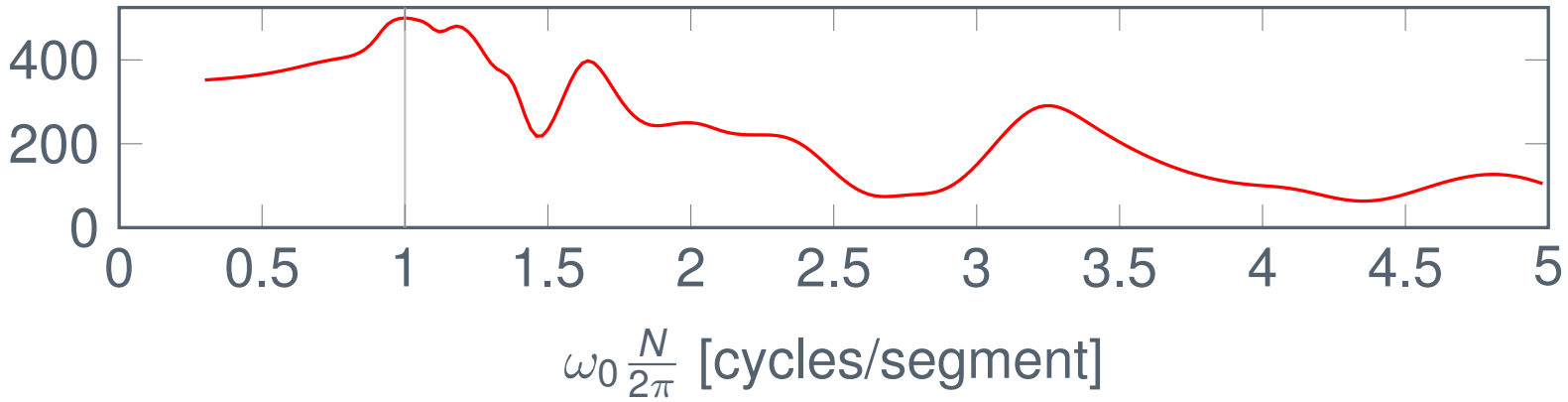


1. Compute NLS cost function

$$\hat{\omega}_0 = \underset{\omega_0 \in (0, \pi/l)}{\operatorname{argmax}} \, \boldsymbol{x}^T \boldsymbol{Z}_l(\omega_0) \left[\boldsymbol{Z}_l^T(\omega_0) \boldsymbol{Z}_l(\omega_0) \right]^{-1} \boldsymbol{Z}_l^T(\omega_0) \boldsymbol{x} \tag{19}$$

   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

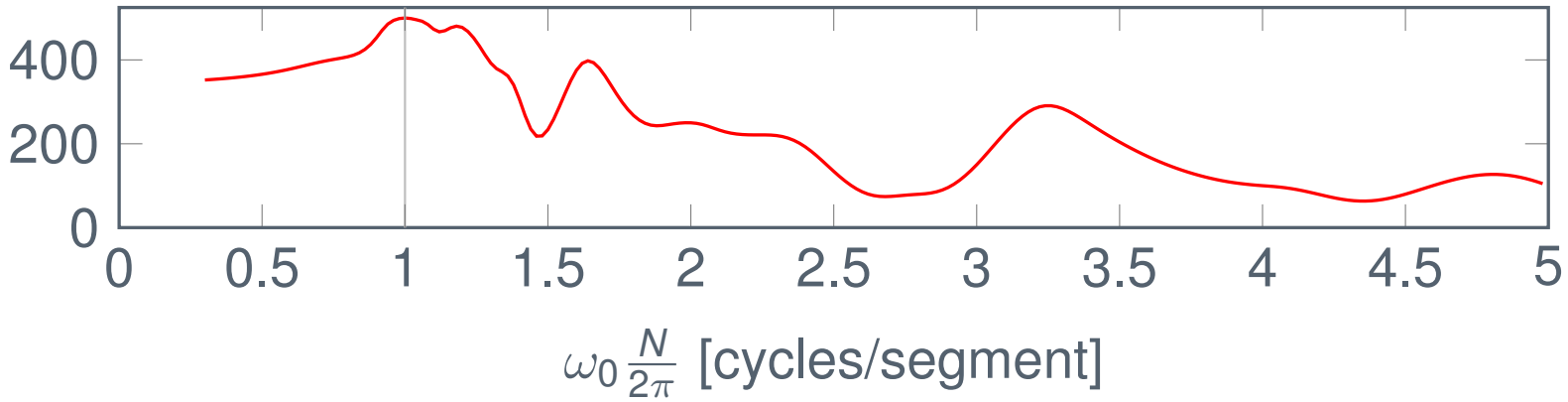


1. Compute NLS cost function

$$\hat{\omega}_0 = \underset{\omega_0 \in (0, \pi/l)}{\operatorname{argmax}} \; \boldsymbol{x}^T \boldsymbol{Z}_l(\omega_0) \left[\boldsymbol{Z}_l^T(\omega_0) \boldsymbol{Z}_l(\omega_0) \right]^{-1} \boldsymbol{Z}_l^T(\omega_0) \boldsymbol{x} \tag{19}$$

   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

$$\hat{\omega}_0 = \underset{\omega_0 \in (0, \pi/l)}{\operatorname{argmax}} \; \boldsymbol{x}^T \boldsymbol{Z}_l(\omega_0) \left[\boldsymbol{Z}_l^T(\omega_0) \boldsymbol{Z}_l(\omega_0) \right]^{-1} \boldsymbol{Z}_l^T(\omega_0) \boldsymbol{x} \tag{19}$$

   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

$$\hat{\omega}_0 \approx \underset{\omega_0 \in (0,\pi/I)}{\operatorname{argmax}} \; \boldsymbol{x}^T \boldsymbol{Z}_I(\omega_0) \left[N \boldsymbol{I}_I / 2 \right]^{-1} \boldsymbol{Z}_I^T(\omega_0) \boldsymbol{x} \; . \tag{20}$$

Harmonic Summation

The harmonic summation (HS) estimator

$$\hat{\omega}_0 \approx \underset{\omega_0 \in (0,\pi/I)}{\operatorname{argmax}} \; \boldsymbol{x}^T \boldsymbol{Z}_I(\omega_0) \left[N \boldsymbol{I}_I / 2 \right]^{-1} \boldsymbol{Z}_I^T(\omega_0) \boldsymbol{x} \; . \tag{20}$$

Complexities

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

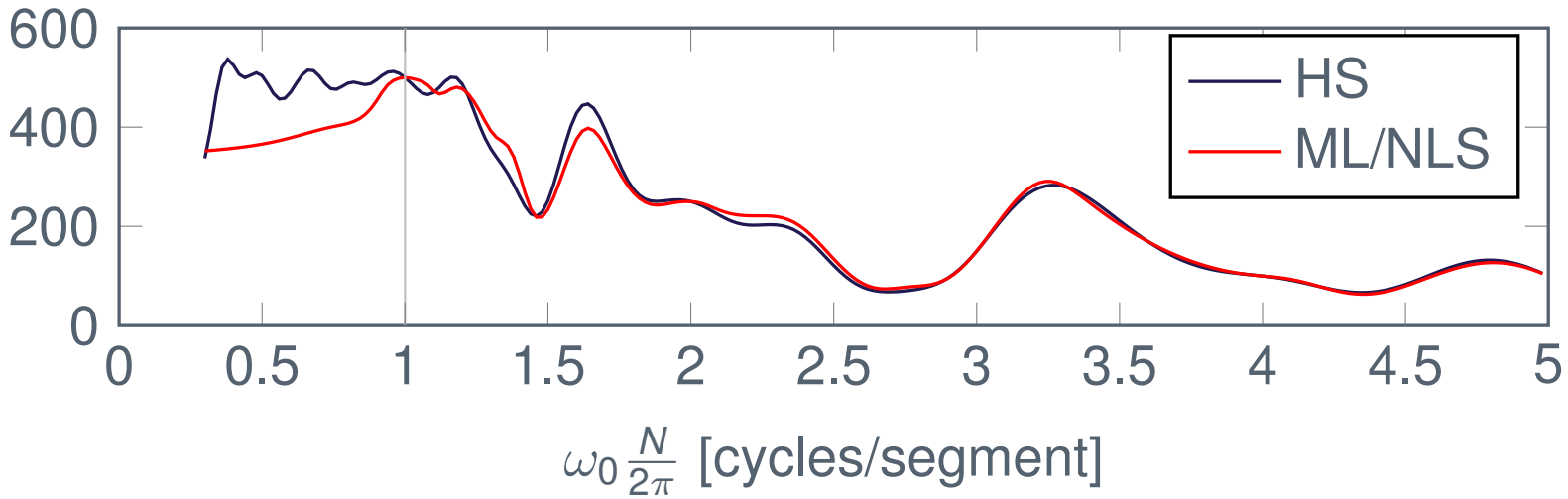
Harmonic Summation

The harmonic summation (HS) estimator

$$\hat{\omega}_0 \approx \underset{\omega_0 \in (0,\pi/l)}{\operatorname{argmax}} \; \boldsymbol{x}^T \boldsymbol{Z}_l(\omega_0) \left[\boldsymbol{N}\boldsymbol{I}_l/2\right]^{-1} \boldsymbol{Z}_l^T(\omega_0)\boldsymbol{x} \; . \tag{20}$$

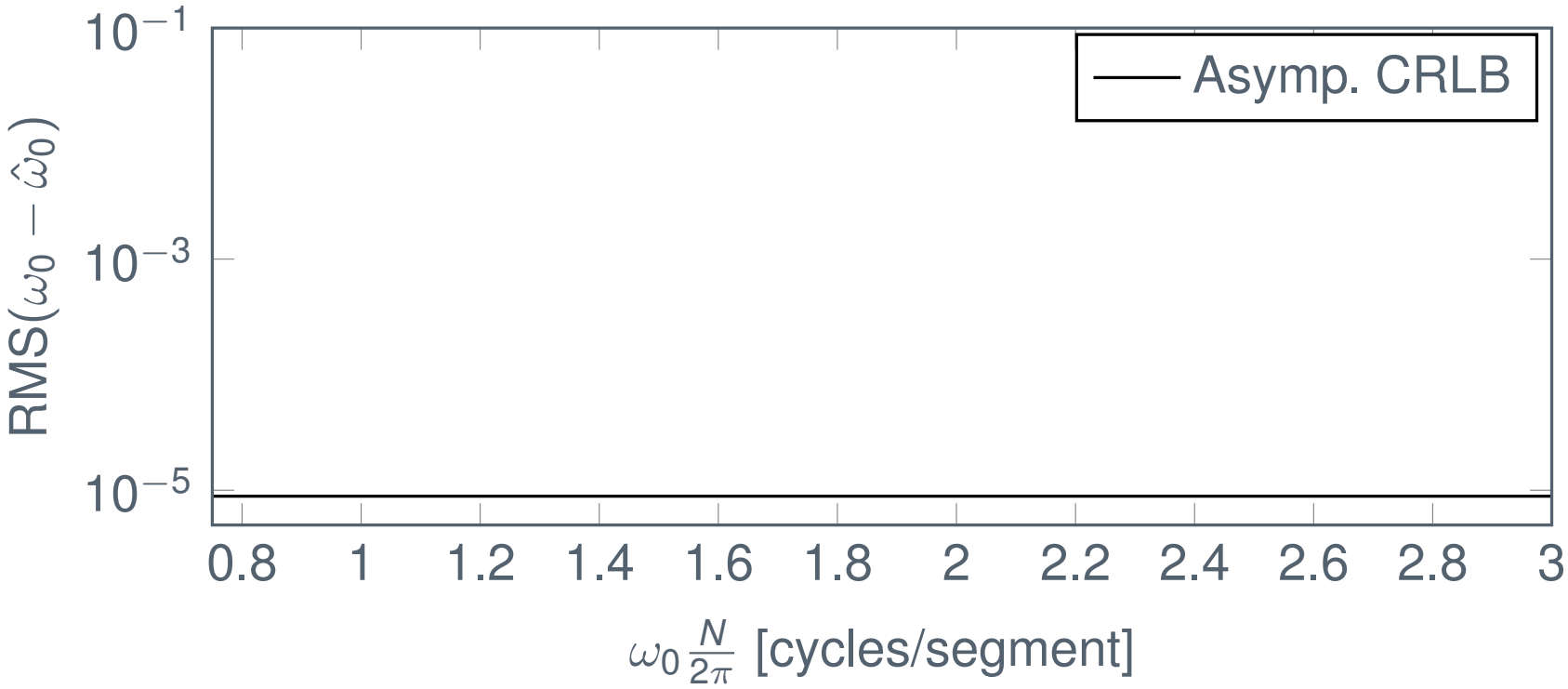
Complexities

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



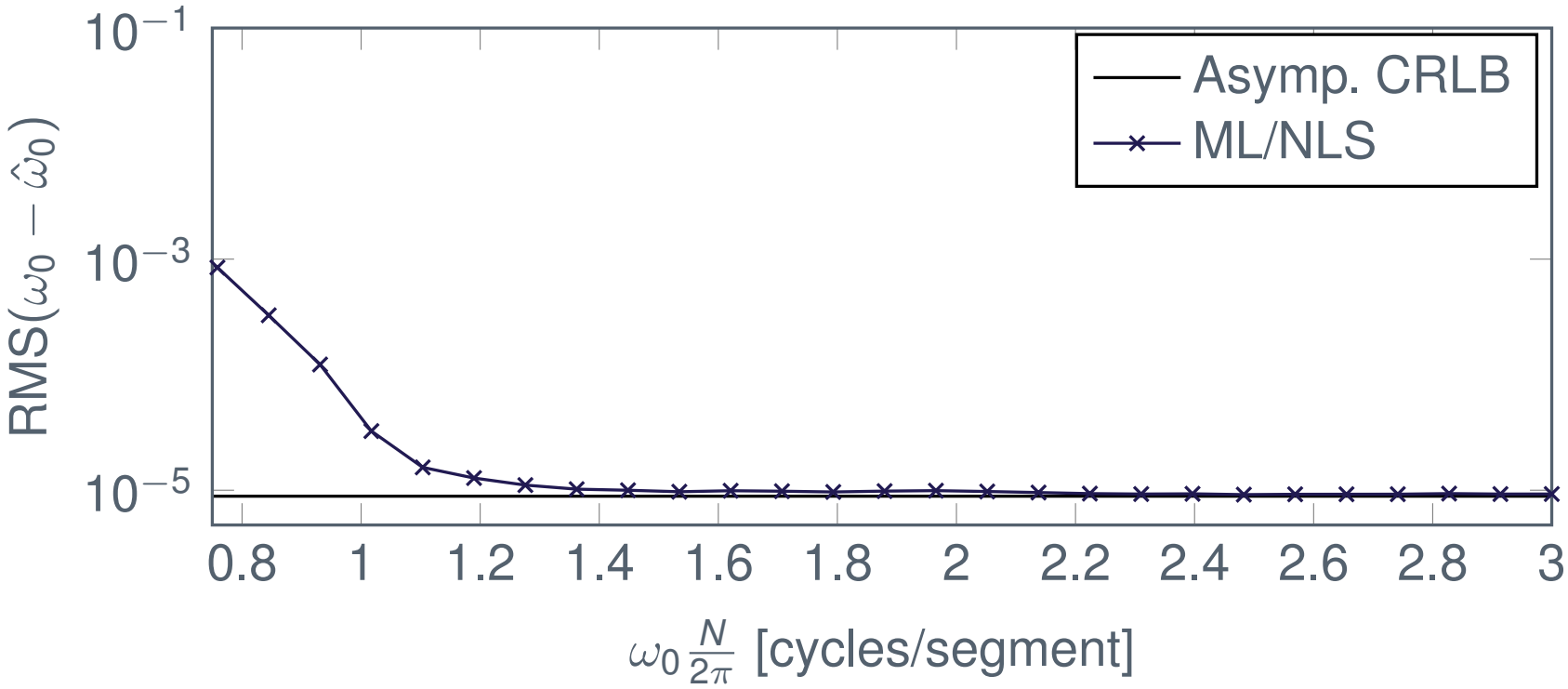
Estimation accuracy for a high fundamental frequency

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



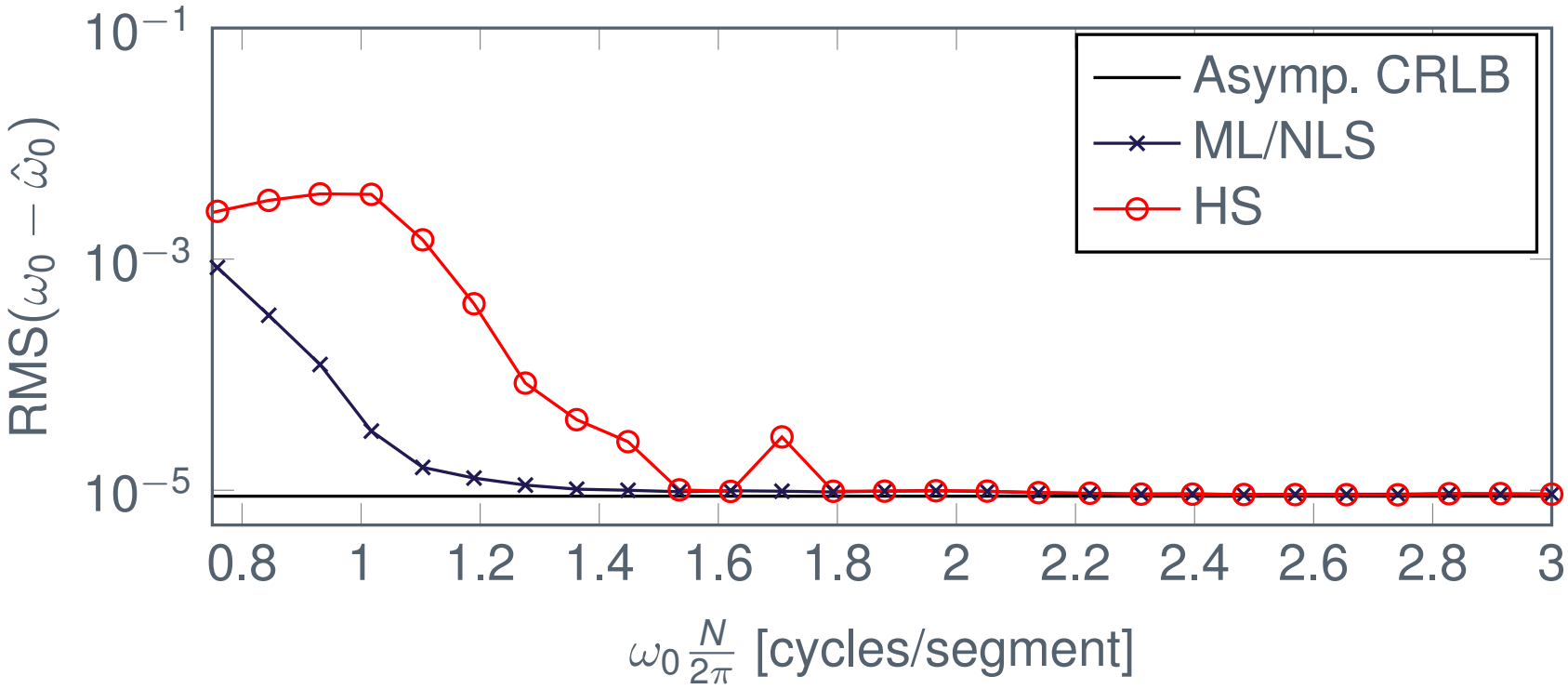
Estimation accuracy for a high fundamental frequency

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



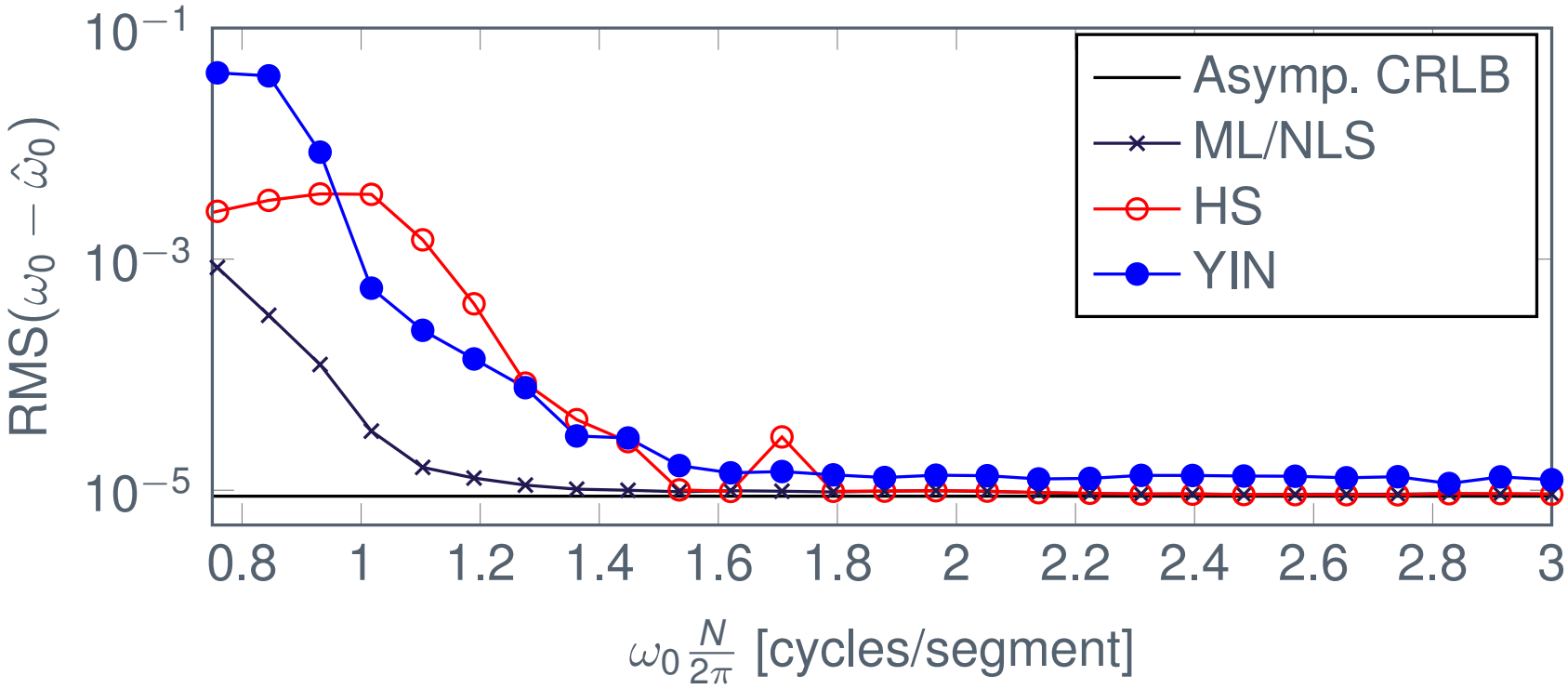
Estimation accuracy for a high fundamental frequency

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



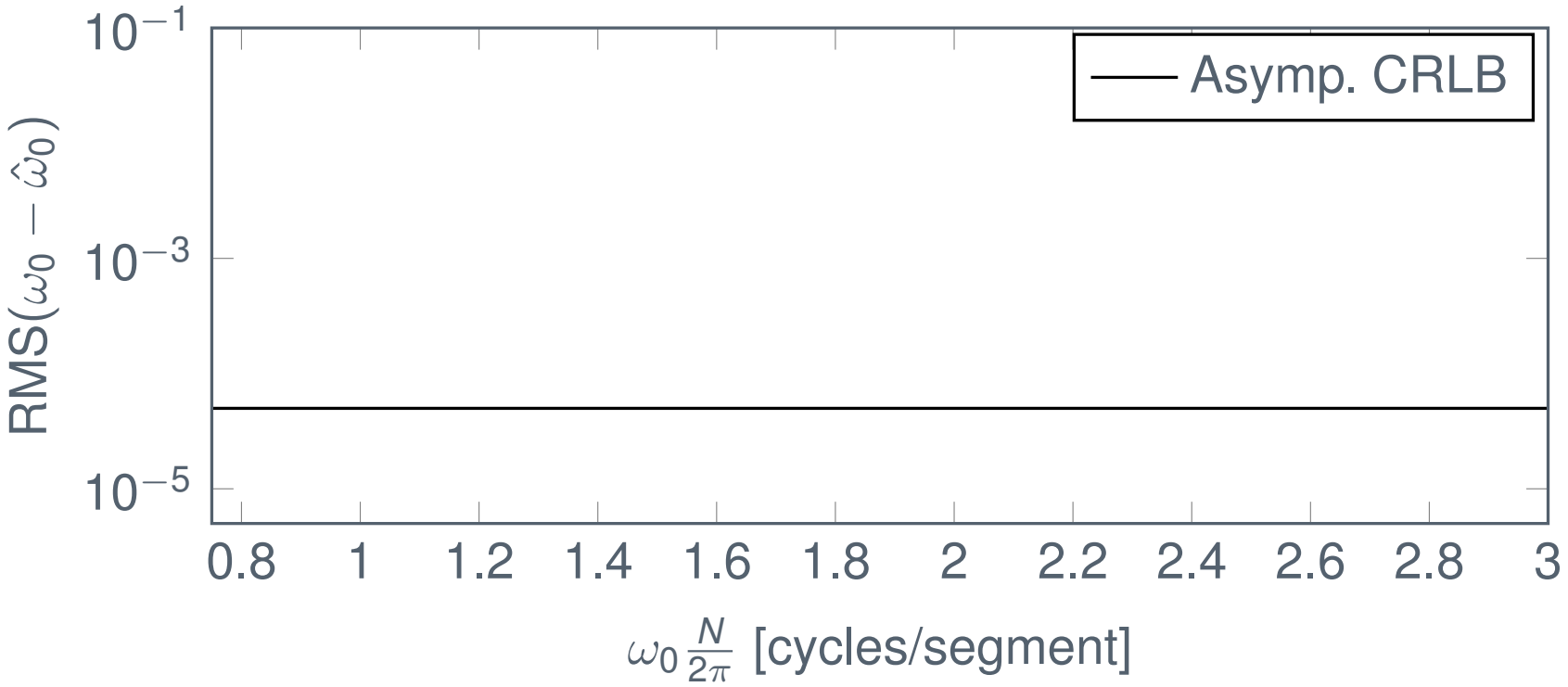
Estimation accuracy for a high fundamental frequency

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



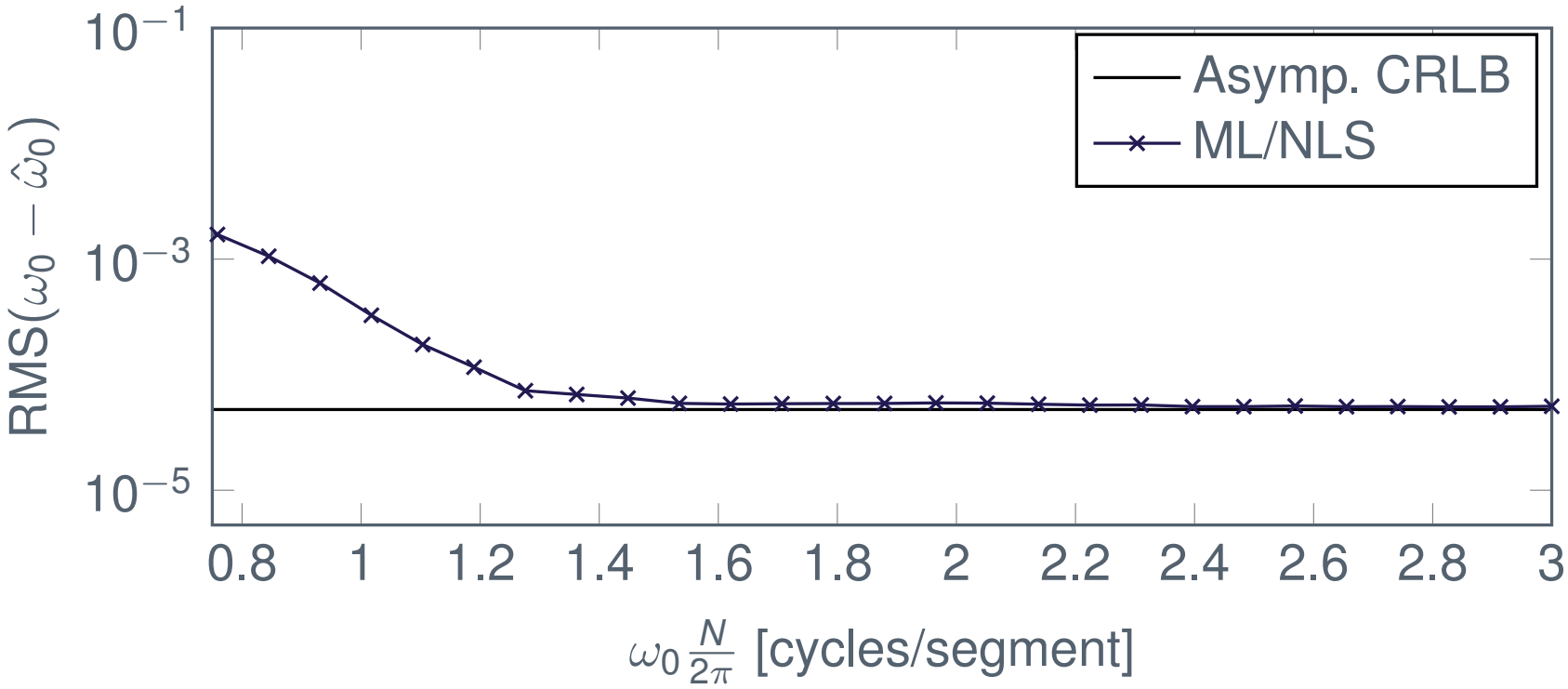
Estimation accuracy for a low fundamental frequency

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



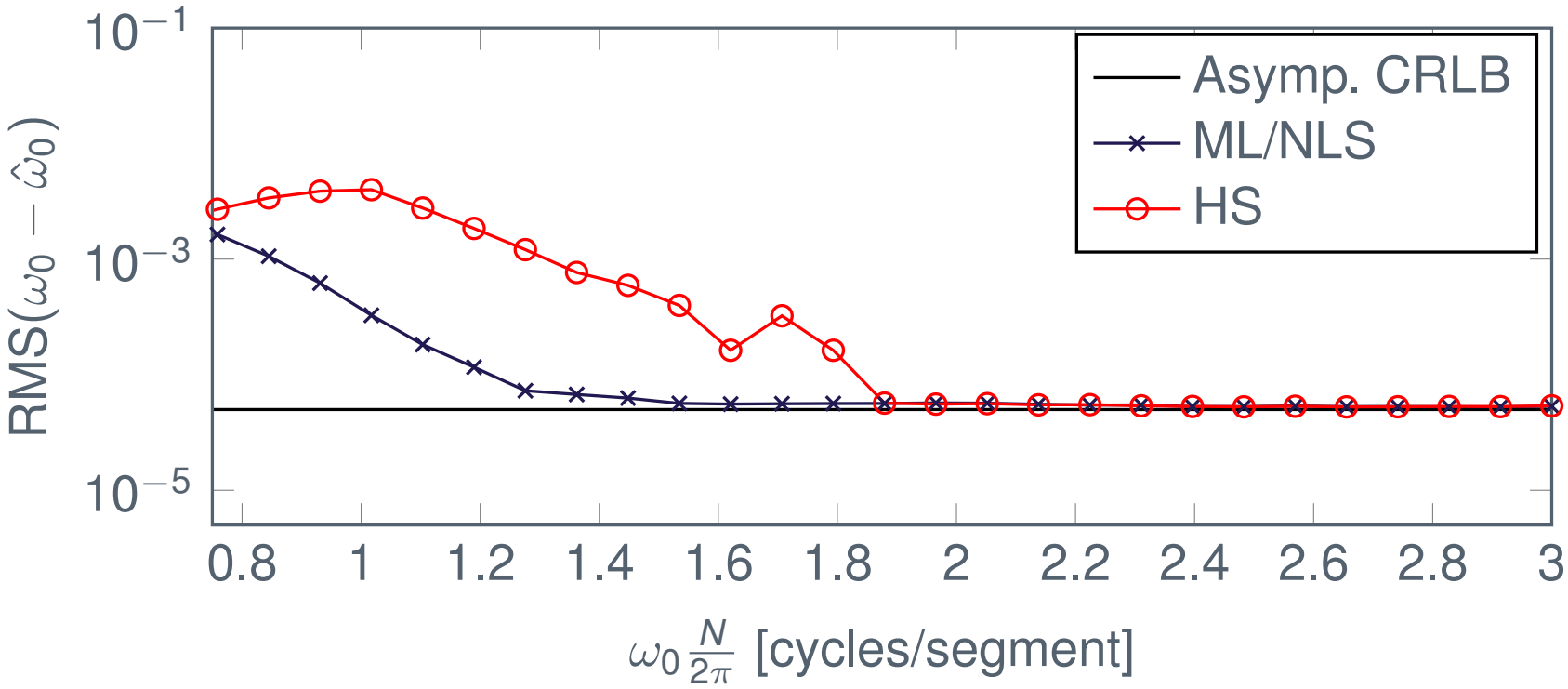
Estimation accuracy for a low fundamental frequency

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



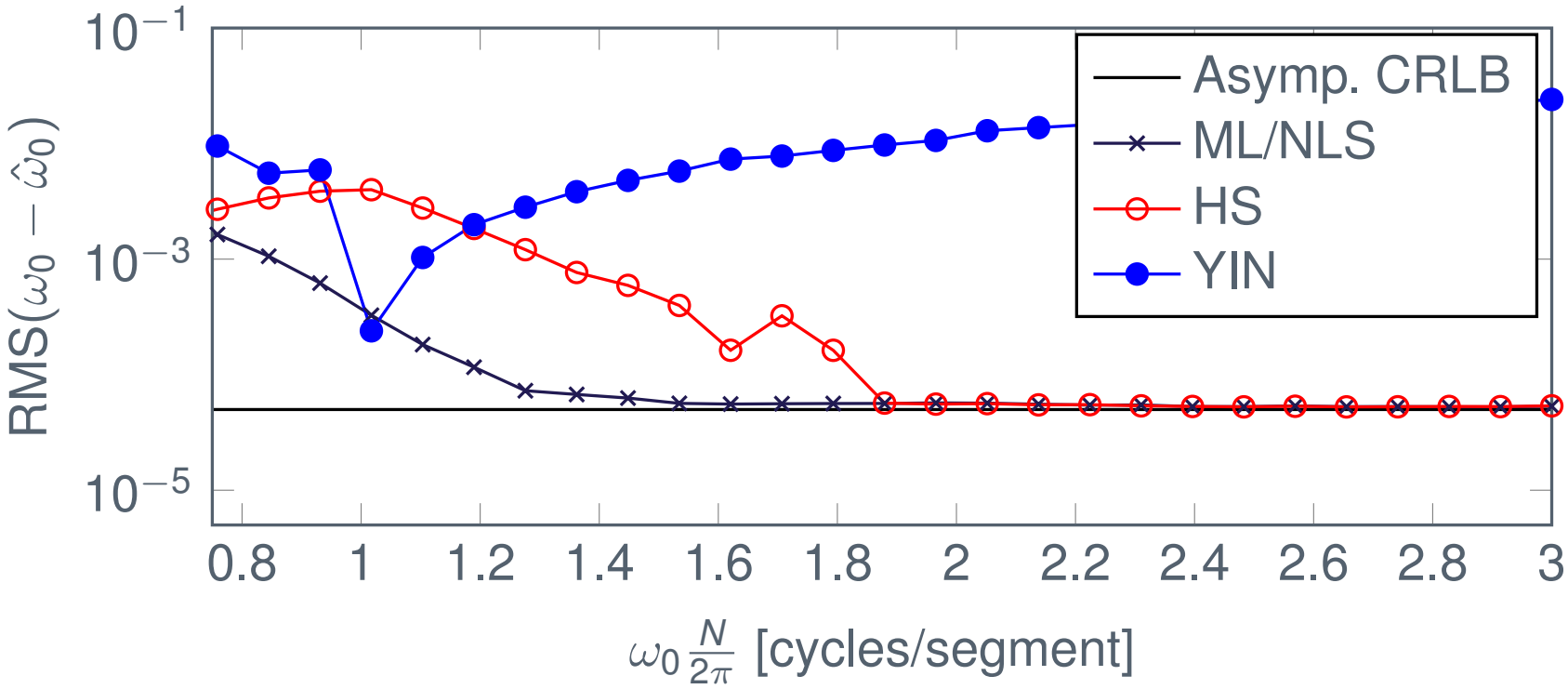
Estimation accuracy for a low fundamental frequency

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 more than approx. 2 cycles/segment, HS and ML/NLS produce nearly the same estimates.

Summary So Far

- When the fundamental frequency is more than approx. 2 cycles/segment, 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 more than approx. 2 cycles/segment, 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 more than approx. 2 cycles/segment, 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

1. ML/NLS and HS: Decrease F (fewer function evaluations)
2. ML/NLS: $\mathcal{O}(F \log F) + \mathcal{O}(FL)$ (faster function evaluations)
3. (Model comparison can be based on numerical integration)

Agenda

Introduction
- Grid-size Selection in the Scalar Case

Fundamental Frequency Estimation
- Background on Fundamental Frequency Estimation
- Fast NLS
- Results
- Summary and Demo

Source Localisation
- Grid Size Selection in the Diagonal Case
- Demo

Summary

Fast ML/NLS

Contribution 1: Fewer function evaluations

We have that

$$f(\hat{\omega}_0) \approx \frac{N}{2} \sum_{i=1}^{I} |\hat{\alpha}_i|^2 \tag{21}$$

$$H(\hat{\omega}_0) \approx -\frac{N^3}{12} \sum_{i=1}^{I} |\hat{\alpha}_i|^2 i^2 \tag{22}$$

Fast ML/NLS

Contribution 1: Fewer function evaluations

We have that

$$f(\hat{\omega}_0) \approx \frac{N}{2} \sum_{i=1}^{I} |\hat{\alpha}_i|^2 \tag{21}$$

$$H(\hat{\omega}_0) \approx -\frac{N^3}{12} \sum_{i=1}^{I} |\hat{\alpha}_i|^2 i^2 \tag{22}$$

Thus,

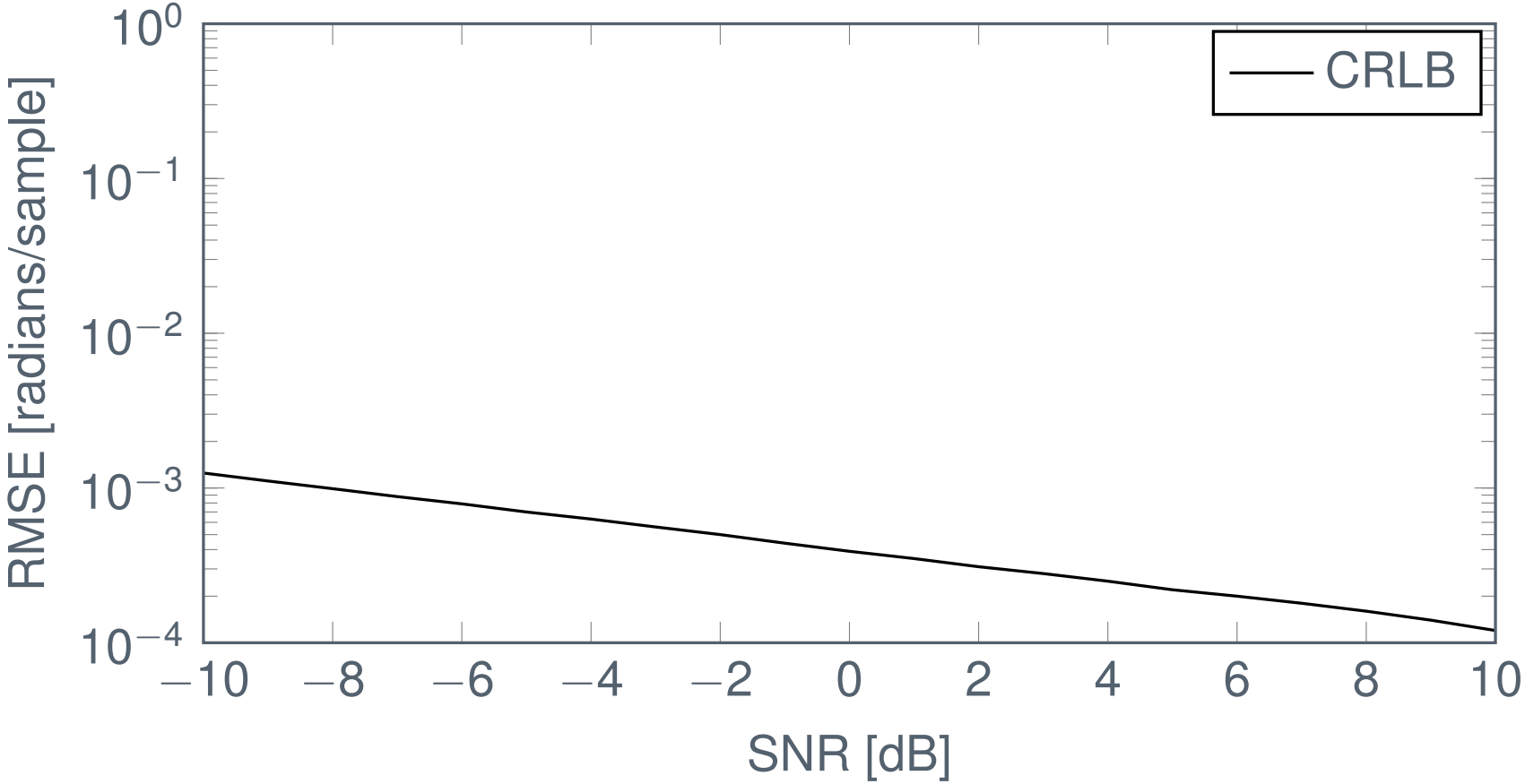
$$\Delta\omega_0 = \sqrt{2\frac{1-g}{g}\frac{f(\hat{\omega}_0)}{H(\hat{\omega}_0)}} \approx \sqrt{12\frac{1-g}{g}\frac{\sum_{i=1}^{I}|\hat{\alpha}_i|^2}{N^2\sum_{i=1}^{I}|\hat{\alpha}_i|^2 i^2}} \tag{23}$$

$$> \sqrt{12\frac{g-1}{g}}\frac{1}{NI} \tag{24}$$

Fast ML/NLS

Contribution 1: Fewer function evaluations

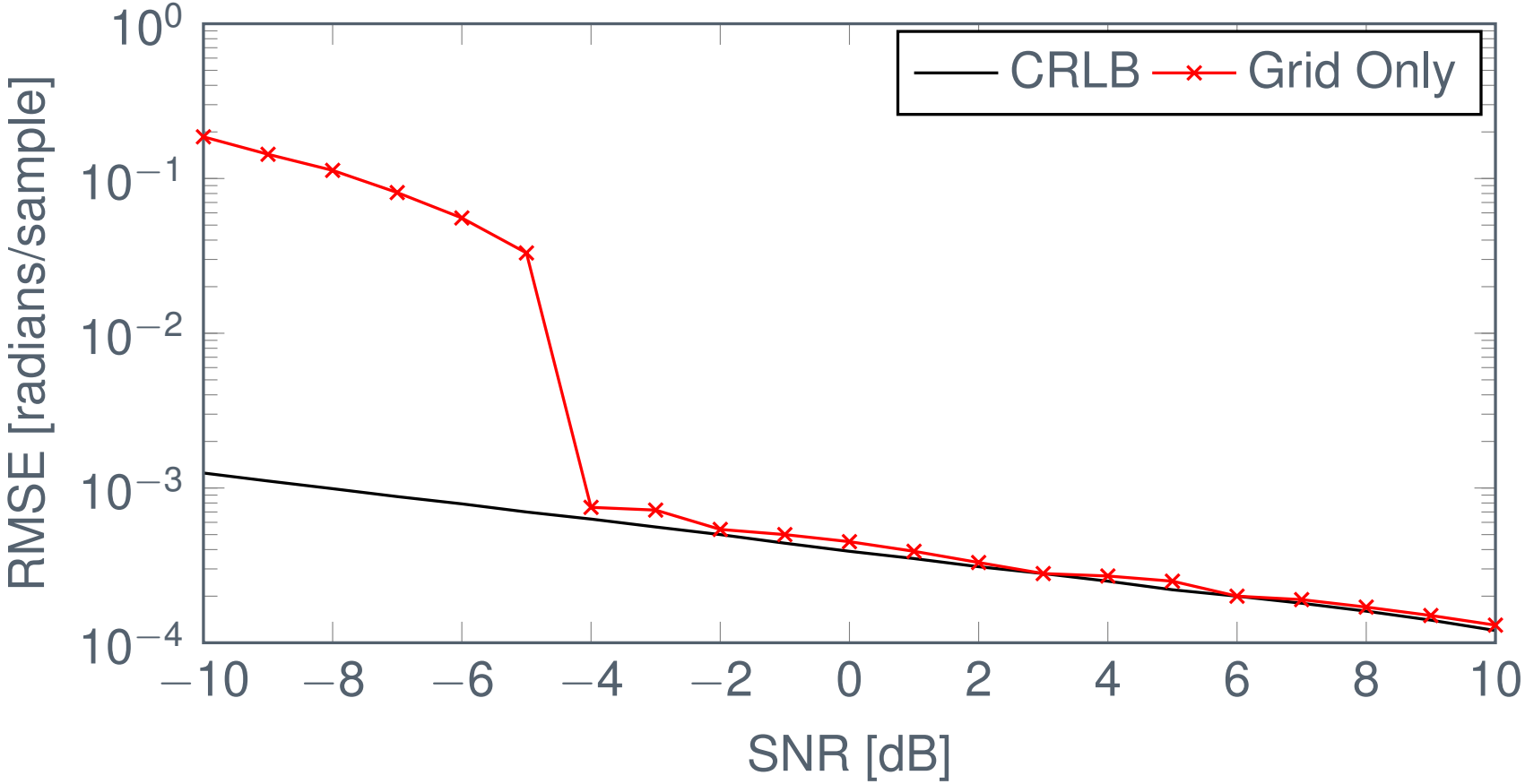
Setup: $N = 100$, $l = 10$, 250 Monte Carlo runs for each g and SNR, and a Fibonacci search for refinement.



Fast ML/NLS

Contribution 1: Fewer function evaluations

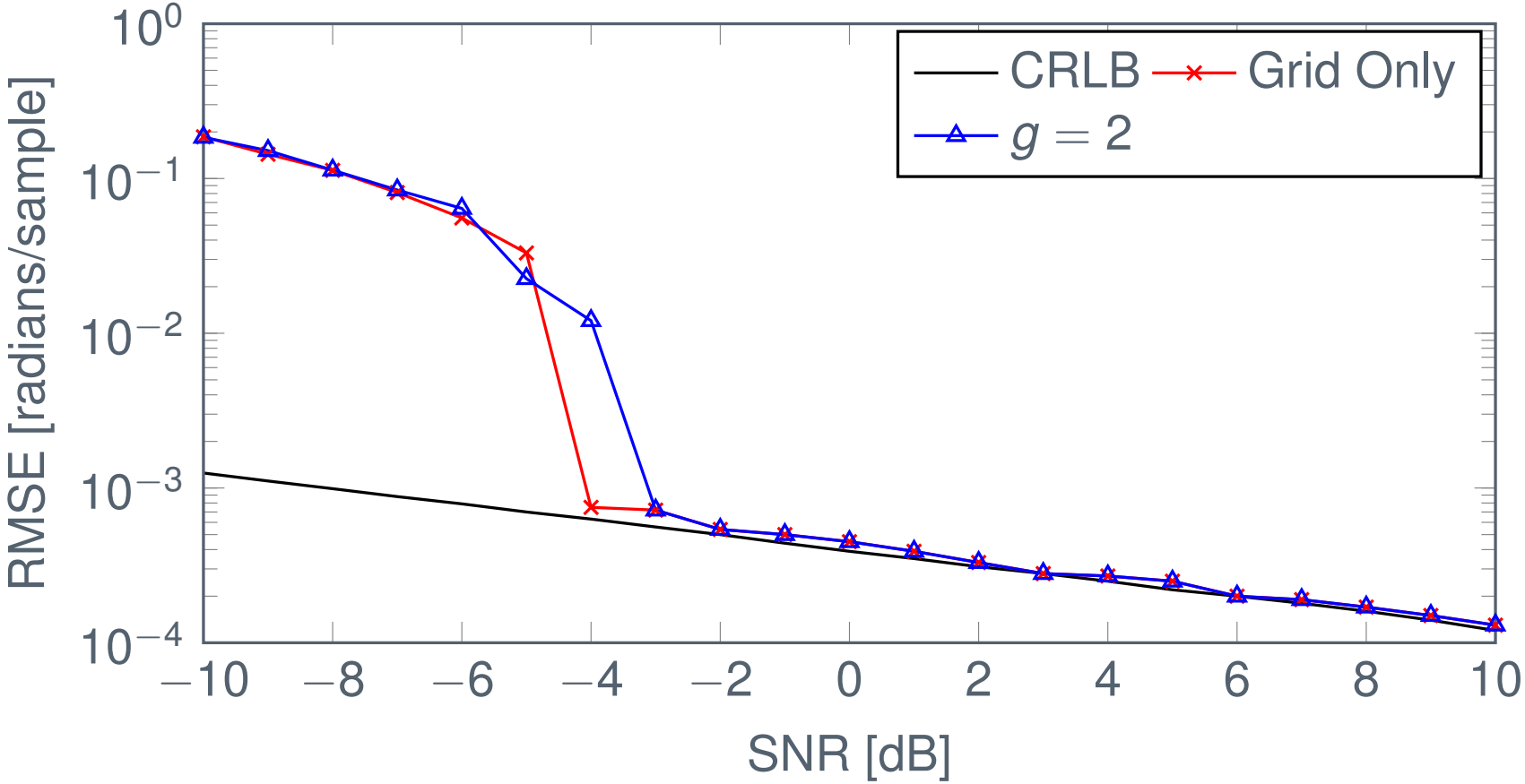
Setup: $N = 100$, $l = 10$, 250 Monte Carlo runs for each g and SNR, and a Fibonacci search for refinement.



Fast ML/NLS

Contribution 1: Fewer function evaluations

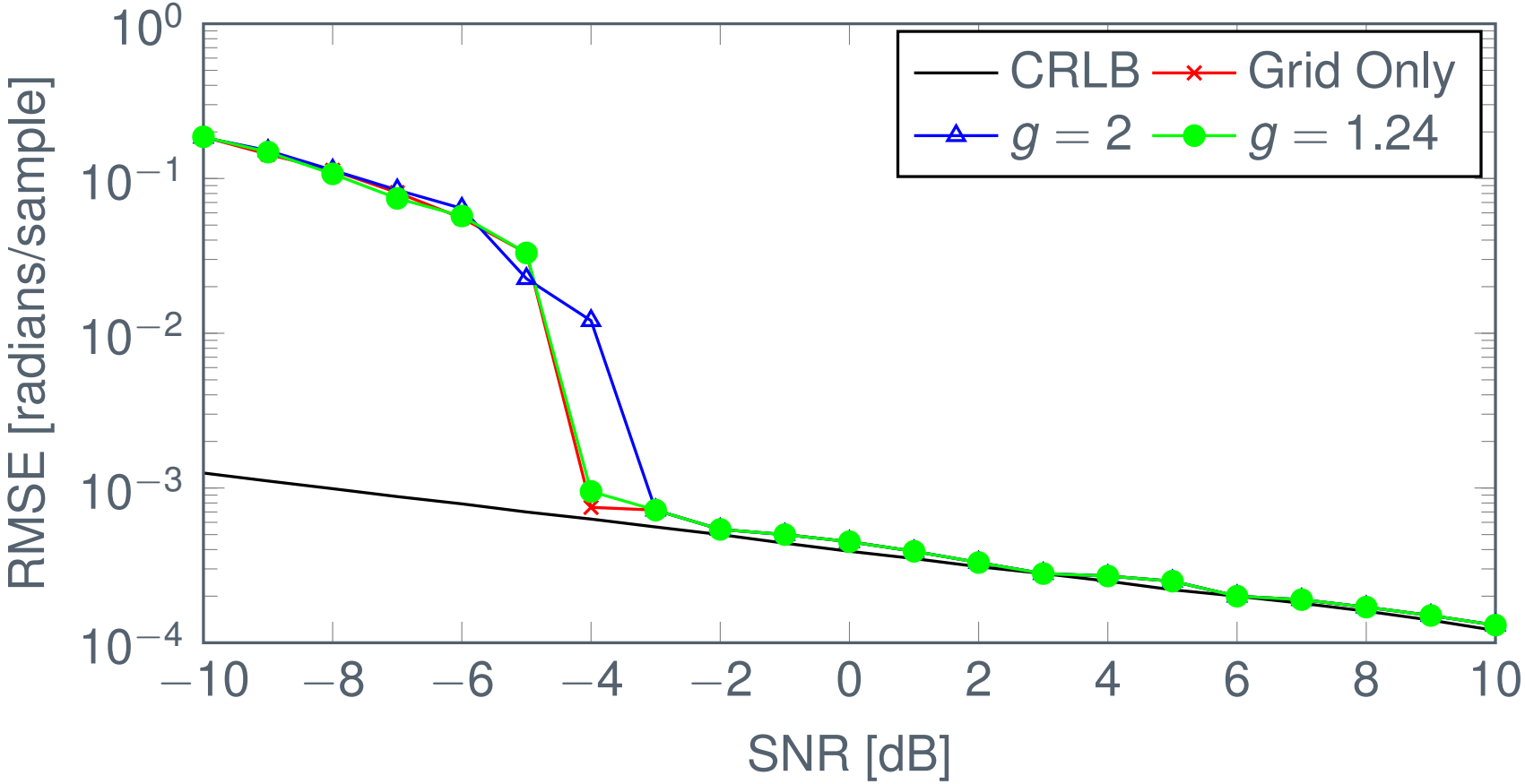
Setup: $N = 100$, $l = 10$, 250 Monte Carlo runs for each g and SNR, and a Fibonacci search for refinement.



Fast ML/NLS

Contribution 1: Fewer function evaluations

Setup: $N = 100$, $l = 10$, 250 Monte Carlo runs for each g and SNR, and a Fibonacci search for refinement.



Fast ML/NLS

Contribution 1: Fewer function evaluations

g	2	1.24	1.024	1.0024	1.00024	Grid Only
# grid points	1285	2000	6011	18911	59770	811116
τ [s]	0.03	0.03	0.03	0.05	0.12	1.42

Fast ML/NLS

Contribution 1: Fewer function evaluations

g	2	1.24	1.024	1.0024	1.00024	Grid Only
# grid points	1285	2000	6011	18911	59770	811116
τ [s]	0.03	0.03	0.03	0.05	0.12	1.42

The grid only approach is approximately 50 times slower than the combined approach for $g = 1.24$.

Fast ML/NLS

Contribution 2: Faster function evaluations

$$J_l(\omega) = \boldsymbol{x}^T \boldsymbol{Z}_l(\omega_0) \left[\boldsymbol{Z}_l^T(\omega_0) \boldsymbol{Z}_l(\omega_0) \right]^{-1} \boldsymbol{Z}_l^T(\omega_0) \boldsymbol{x} \tag{25}$$

Fast ML/NLS

Contribution 2: Faster function evaluations

$$J_l(\omega) = \boldsymbol{x}^T \boldsymbol{Z}_l(\omega_0) \left[\boldsymbol{Z}_l^T(\omega_0)\boldsymbol{Z}_l(\omega_0)\right]^{-1} \boldsymbol{Z}_l^T(\omega_0)\boldsymbol{x} \tag{25}$$

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

Contribution 2: Faster function evaluations

$$J_l(\omega) = \boldsymbol{x}^T \boldsymbol{Z}_l(\omega_0) \left[\boldsymbol{Z}_l^T(\omega_0)\boldsymbol{Z}_l(\omega_0)\right]^{-1} \boldsymbol{Z}_l^T(\omega_0)\boldsymbol{x} \tag{25}$$

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

Contribution 2: Faster function evaluations

$$J_l(\omega) = \boldsymbol{x}^T \boldsymbol{Z}_l(\omega_0) \left[\boldsymbol{Z}_l^T(\omega_0)\boldsymbol{Z}_l(\omega_0)\right]^{-1} \boldsymbol{Z}_l^T(\omega_0)\boldsymbol{x} \tag{25}$$

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

$$\boldsymbol{Z}_l^T(\omega_0)\boldsymbol{Z}_l(\omega_0) = \begin{bmatrix} \boldsymbol{T}_l(\omega_0) & -\tilde{\boldsymbol{T}}_l(\omega_0) \\ \tilde{\boldsymbol{T}}_l(\omega_0) & \boldsymbol{T}_l(\omega_0) \end{bmatrix} + \begin{bmatrix} \boldsymbol{H}_l(\omega_0) & \tilde{\boldsymbol{H}}_l(\omega_0) \\ \tilde{\boldsymbol{H}}_l(\omega_0) & -\boldsymbol{H}_l(\omega_0) \end{bmatrix} \tag{26}$$

Fast ML/NLS

Contribution 2: Faster function evaluations

$$J_l(\omega) = \boldsymbol{x}^T \boldsymbol{Z}_l(\omega_0) \left[\boldsymbol{Z}_l^T(\omega_0)\boldsymbol{Z}_l(\omega_0)\right]^{-1} \boldsymbol{Z}_l^T(\omega_0)\boldsymbol{x} \tag{25}$$

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

$$\boldsymbol{Z}_l^T(\omega_0)\boldsymbol{Z}_l(\omega_0) = \begin{bmatrix} \boldsymbol{T}_l(\omega_0) & -\tilde{\boldsymbol{T}}_l(\omega_0) \\ \tilde{\boldsymbol{T}}_l(\omega_0) & \boldsymbol{T}_l(\omega_0) \end{bmatrix} + \begin{bmatrix} \boldsymbol{H}_l(\omega_0) & \tilde{\boldsymbol{H}}_l(\omega_0) \\ \tilde{\boldsymbol{H}}_l(\omega_0) & -\boldsymbol{H}_l(\omega_0) \end{bmatrix} \tag{26}$$

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

Fast ML/NLS

Contribution 2: Faster function evaluations

$$J_l(\omega) = \boldsymbol{x}^T \boldsymbol{Z}_l(\omega_0) \left[\boldsymbol{Z}_l^T(\omega_0)\boldsymbol{Z}_l(\omega_0)\right]^{-1} \boldsymbol{Z}_l^T(\omega_0)\boldsymbol{x} \tag{25}$$

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

$$\boldsymbol{Z}_l^T(\omega_0)\boldsymbol{Z}_l(\omega_0) = \begin{bmatrix} \boldsymbol{T}_l(\omega_0) & -\tilde{\boldsymbol{T}}_l(\omega_0) \\ \tilde{\boldsymbol{T}}_l(\omega_0) & \boldsymbol{T}_l(\omega_0) \end{bmatrix} + \begin{bmatrix} \boldsymbol{H}_l(\omega_0) & \tilde{\boldsymbol{H}}_l(\omega_0) \\ \tilde{\boldsymbol{H}}_l(\omega_0) & -\boldsymbol{H}_l(\omega_0) \end{bmatrix} \tag{26}$$

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

$$[\boldsymbol{T}_l(\omega_0) + \boldsymbol{H}_l(\omega_0)]\, \boldsymbol{a}_l(\omega_0) = \bar{\boldsymbol{w}}_l(\omega_0) \tag{27}$$

$$[\boldsymbol{T}_l(\omega_0) - \boldsymbol{H}_l(\omega_0)]\, \boldsymbol{b}_l(\omega_0) = -\tilde{\boldsymbol{w}}_l(\omega_0)\,. \tag{28}$$

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

Contribution 2: Faster function evaluations

$$[\boldsymbol{T}_l(\omega_0) + \boldsymbol{H}_l(\omega_0)]\, \boldsymbol{a}_l(\omega_0) = \bar{\boldsymbol{w}}_l(\omega_0) \tag{29}$$

Fast ML/NLS

Contribution 2: Faster function evaluations

$$[\boldsymbol{T}_l(\omega_0) + \boldsymbol{H}_l(\omega_0)]\, \boldsymbol{a}_l(\omega_0) = \bar{\boldsymbol{w}}_l(\omega_0) \tag{29}$$

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

Contribution 2: Faster function evaluations

$$[\boldsymbol{T}_l(\omega_0) + \boldsymbol{H}_l(\omega_0)]\, \boldsymbol{a}_l(\omega_0) = \bar{\boldsymbol{w}}_l(\omega_0) \tag{29}$$

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 (29) is a solution to a lower order.

Fast ML/NLS

Contribution 2: Faster function evaluations

$$[\boldsymbol{T}_l(\omega_0) + \boldsymbol{H}_l(\omega_0)]\, \boldsymbol{a}_l(\omega_0) = \bar{\boldsymbol{w}}_l(\omega_0) \tag{29}$$

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 (29) 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

Contribution 2: Faster function evaluations

$$[\boldsymbol{T}_l(\omega_0) + \boldsymbol{H}_l(\omega_0)]\, \boldsymbol{a}_l(\omega_0) = \bar{\boldsymbol{w}}_l(\omega_0) \tag{29}$$

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 (29) 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 (29) therefore has a time complexity of $\mathcal{O}(l)$ when we have the solution to (29) for $l-1$.

Fast ML/NLS

Contribution 2: Faster function evaluations

$$[\boldsymbol{T}_l(\omega_0) + \boldsymbol{H}_l(\omega_0)]\, \boldsymbol{a}_l(\omega_0) = \bar{\boldsymbol{w}}_l(\omega_0) \tag{29}$$

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 (29) 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 (29) therefore has a time complexity of $\mathcal{O}(l)$ when we have the solution to (29) for $l - 1$.
10. The total time complexity is reduced to

$$\mathcal{O}(F \log F) + \mathcal{O}(FL) \tag{30}$$

Fast ML/NLS

Contribution 2: Faster function evaluations

Toeplitz-plus-Hankel Solver

$$[\boldsymbol{T}_l(\omega_0) + \boldsymbol{H}_l(\omega_0)]\, \boldsymbol{a}_l(\omega_0) = \bar{\boldsymbol{w}}_l(\omega_0) \tag{31}$$

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

Fast ML/NLS

Contribution 2: Faster function evaluations

Toeplitz-plus-Hankel Solver

$$[\boldsymbol{T}_l(\omega_0) + \boldsymbol{H}_l(\omega_0)]\, \boldsymbol{a}_l(\omega_0) = \bar{\boldsymbol{w}}_l(\omega_0) \tag{31}$$

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

Fast ML/NLS

Contribution 2: Faster function evaluations

Toeplitz-plus-Hankel Solver

$$[\boldsymbol{T}_l(\omega_0) + \boldsymbol{H}_l(\omega_0)]\, \boldsymbol{a}_l(\omega_0) = \bar{\boldsymbol{w}}_l(\omega_0) \tag{31}$$

- 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

$$[\boldsymbol{T}_l(\omega_0) + \boldsymbol{H}_l(\omega_0)]\, \gamma_l(\omega_0) = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix}^T . \tag{32}$$

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

Fast ML/NLS

Contribution 2: Faster function evaluations

Toeplitz-plus-Hankel Solver

$$[\boldsymbol{T}_l(\omega_0) + \boldsymbol{H}_l(\omega_0)]\, \boldsymbol{a}_l(\omega_0) = \bar{\boldsymbol{w}}_l(\omega_0) \tag{31}$$

- 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

$$[\boldsymbol{T}_l(\omega_0) + \boldsymbol{H}_l(\omega_0)]\, \gamma_l(\omega_0) = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix}^T . \tag{32}$$

  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

Introduction
- Grid-size Selection in the Scalar Case

Fundamental Frequency Estimation
- Background on Fundamental Frequency Estimation
- Fast NLS
- Results
- Summary and Demo

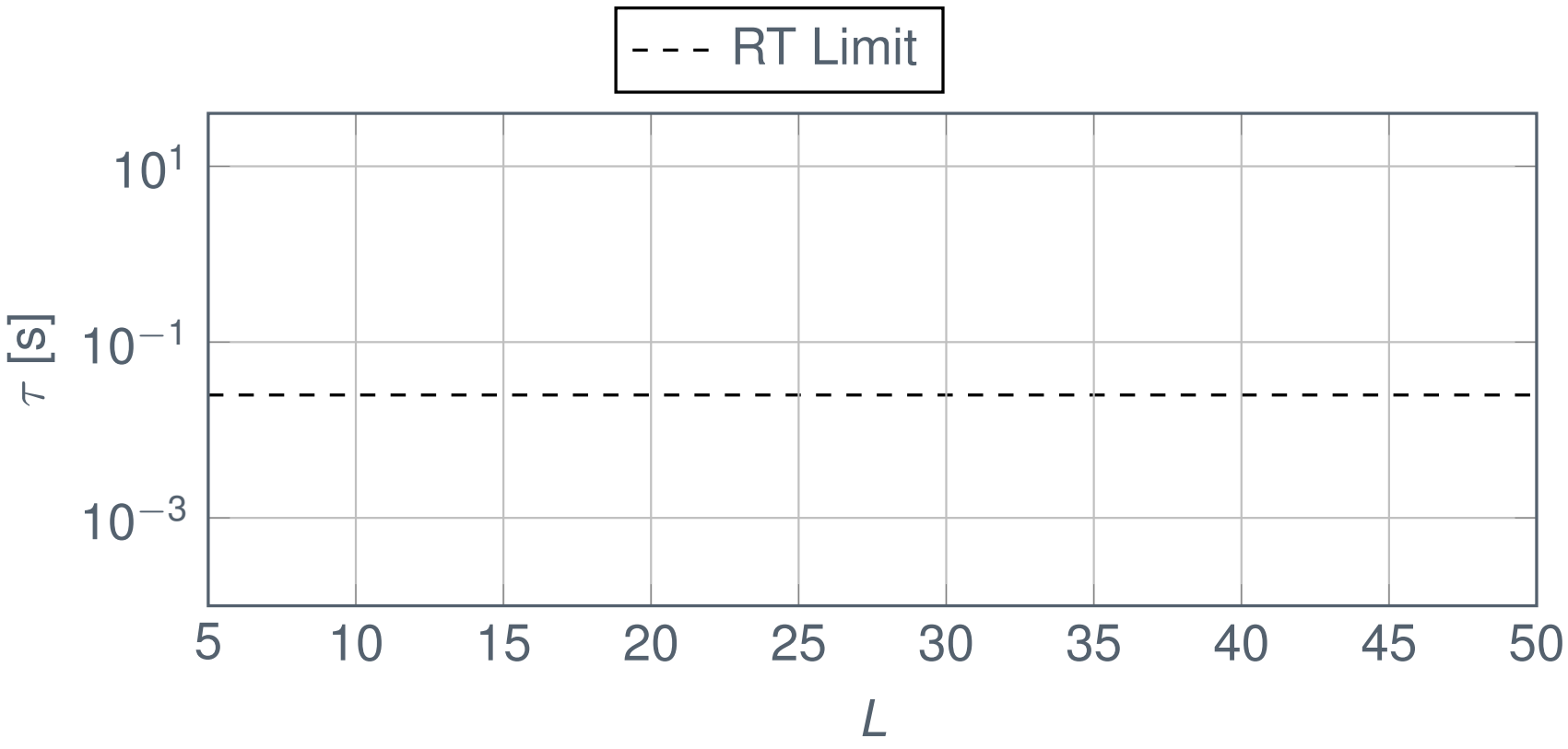
Source Localisation
- Grid Size Selection in the Diagonal Case
- Demo

Summary

Computation Time vs. Model Order

MATLAB Implementation

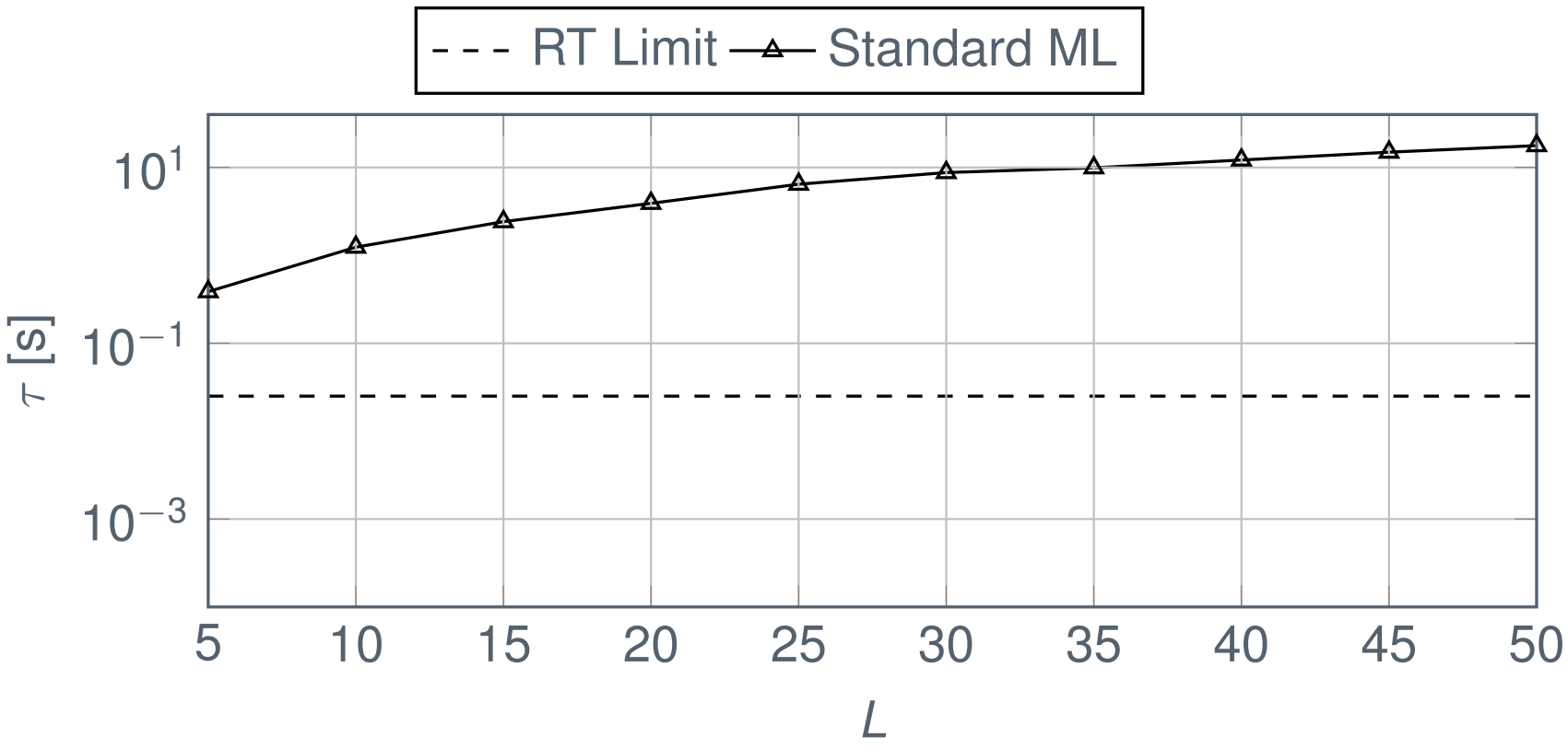
Setup: $N = 200$ (25 ms @ $f_s = 8000$ Hz), $F = 5NL$, T420 laptop



Computation Time vs. Model Order

MATLAB Implementation

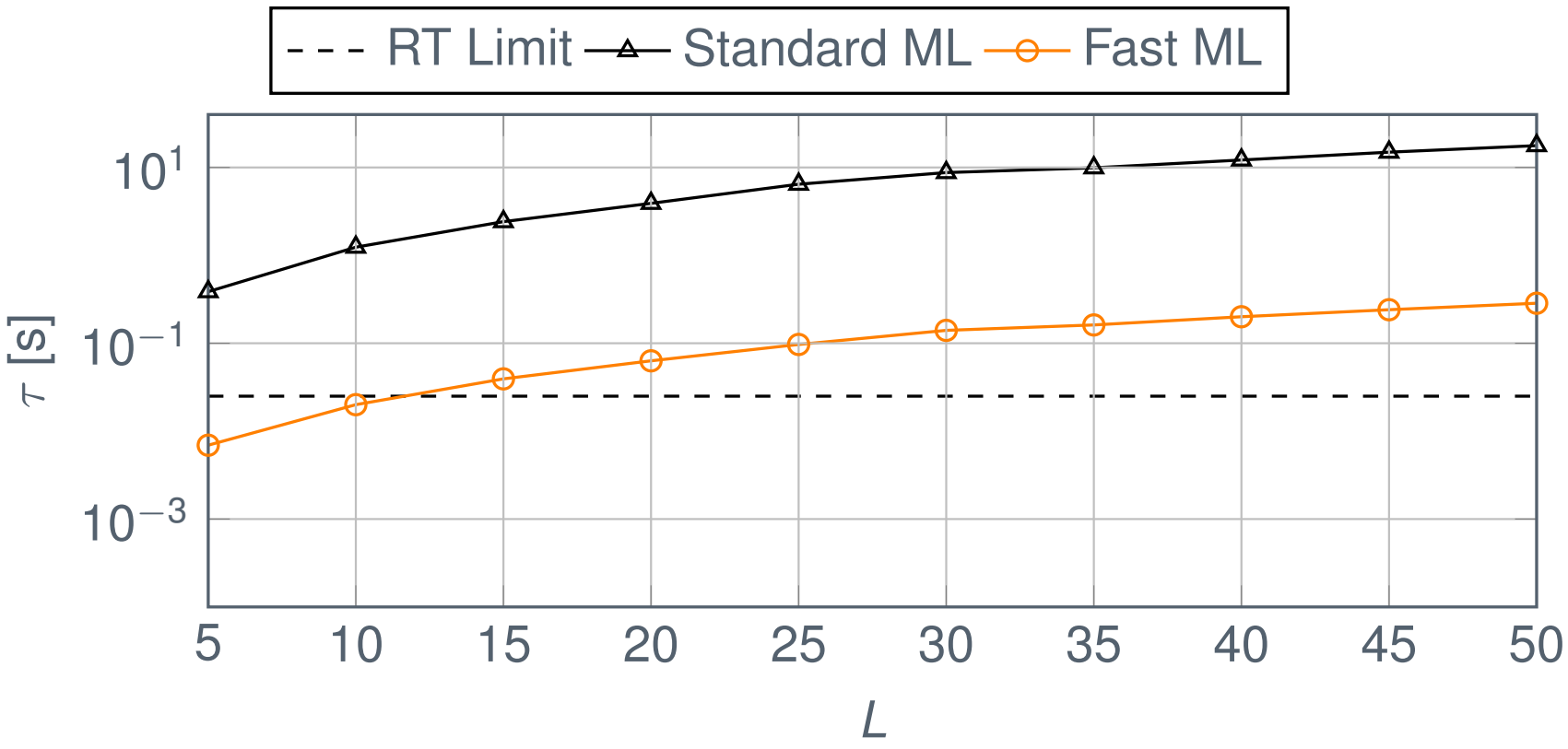
Setup: $N = 200$ (25 ms @ $f_s = 8000$ Hz), $F = 5NL$, T420 laptop



Computation Time vs. Model Order

MATLAB Implementation

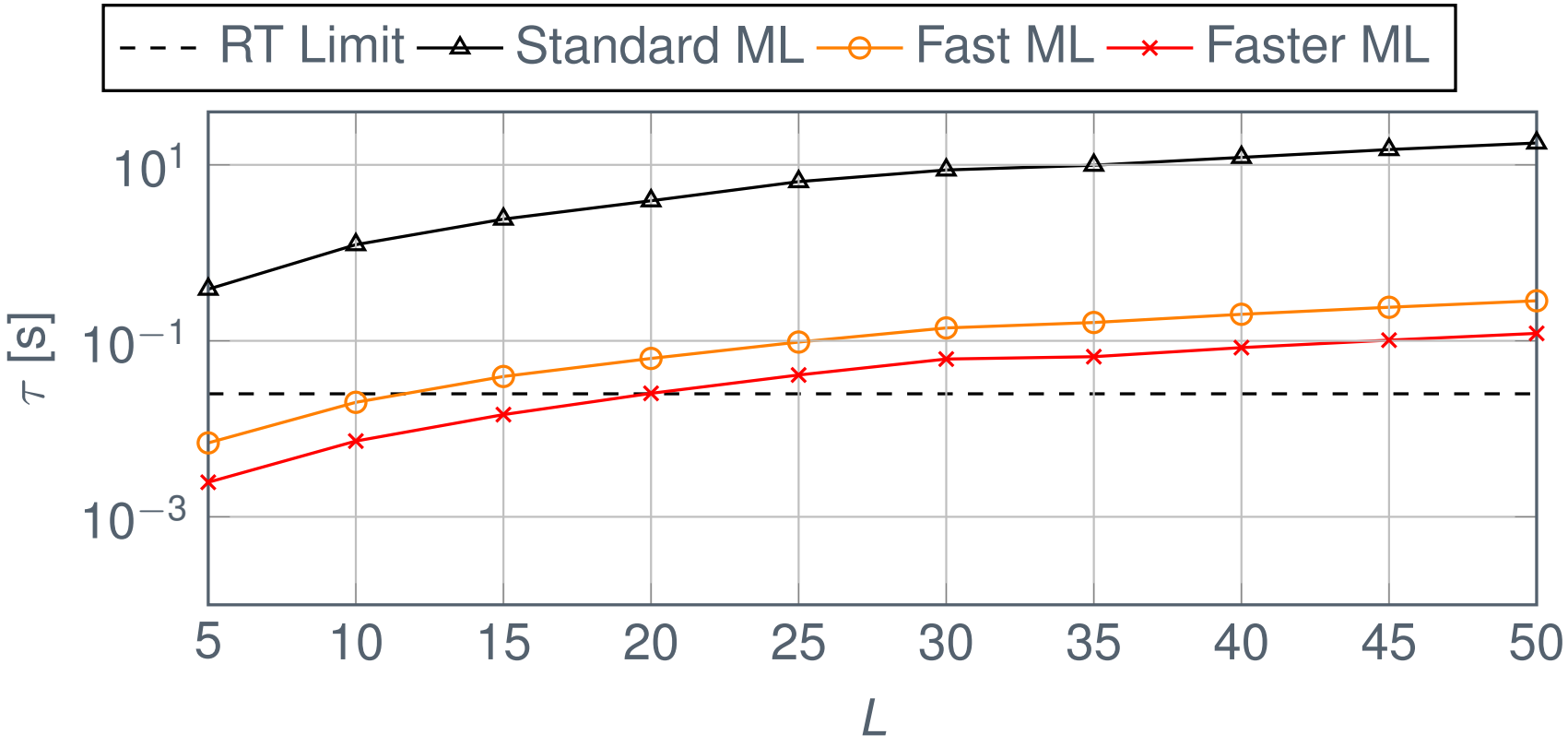
Setup: $N = 200$ (25 ms @ $f_s = 8000$ Hz), $F = 5NL$, T420 laptop



Computation Time vs. Model Order

MATLAB Implementation

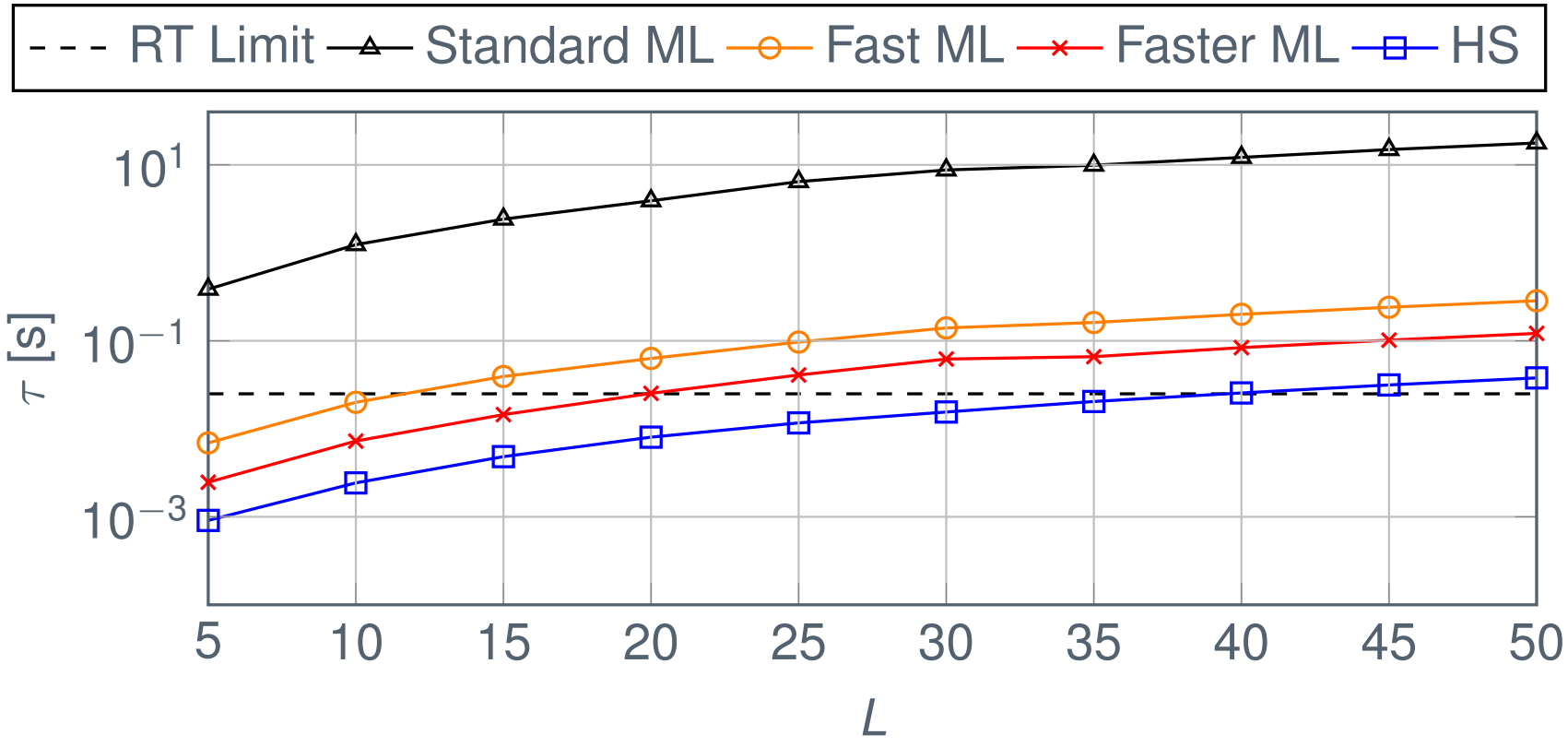
Setup: $N = 200$ (25 ms @ $f_s = 8000$ Hz), $F = 5NL$, T420 laptop



Computation Time vs. Model Order

MATLAB Implementation

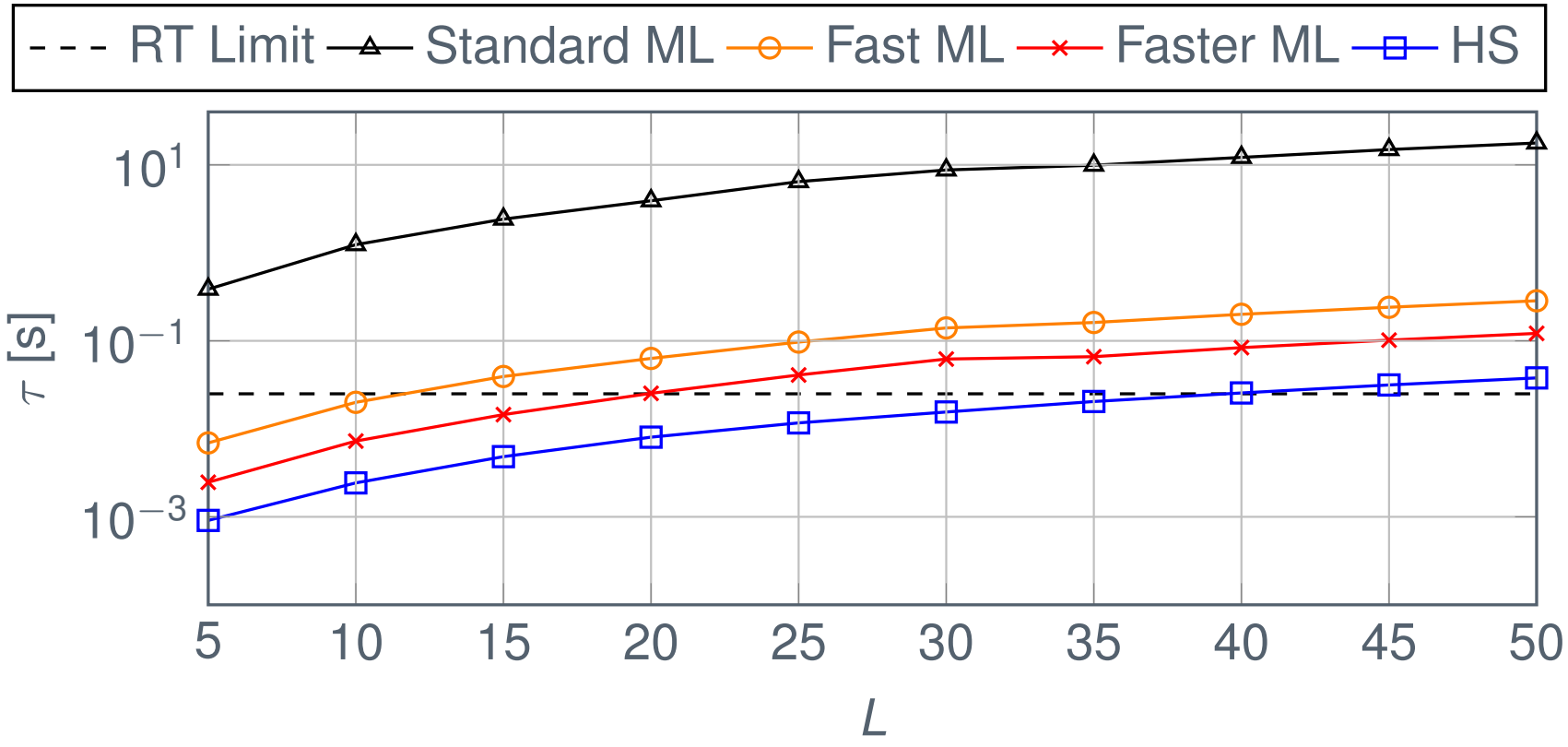
Setup: $N = 200$ (25 ms @ $f_s = 8000$ Hz), $F = 5NL$, T420 laptop



Computation Time vs. Model Order

MATLAB Implementation

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

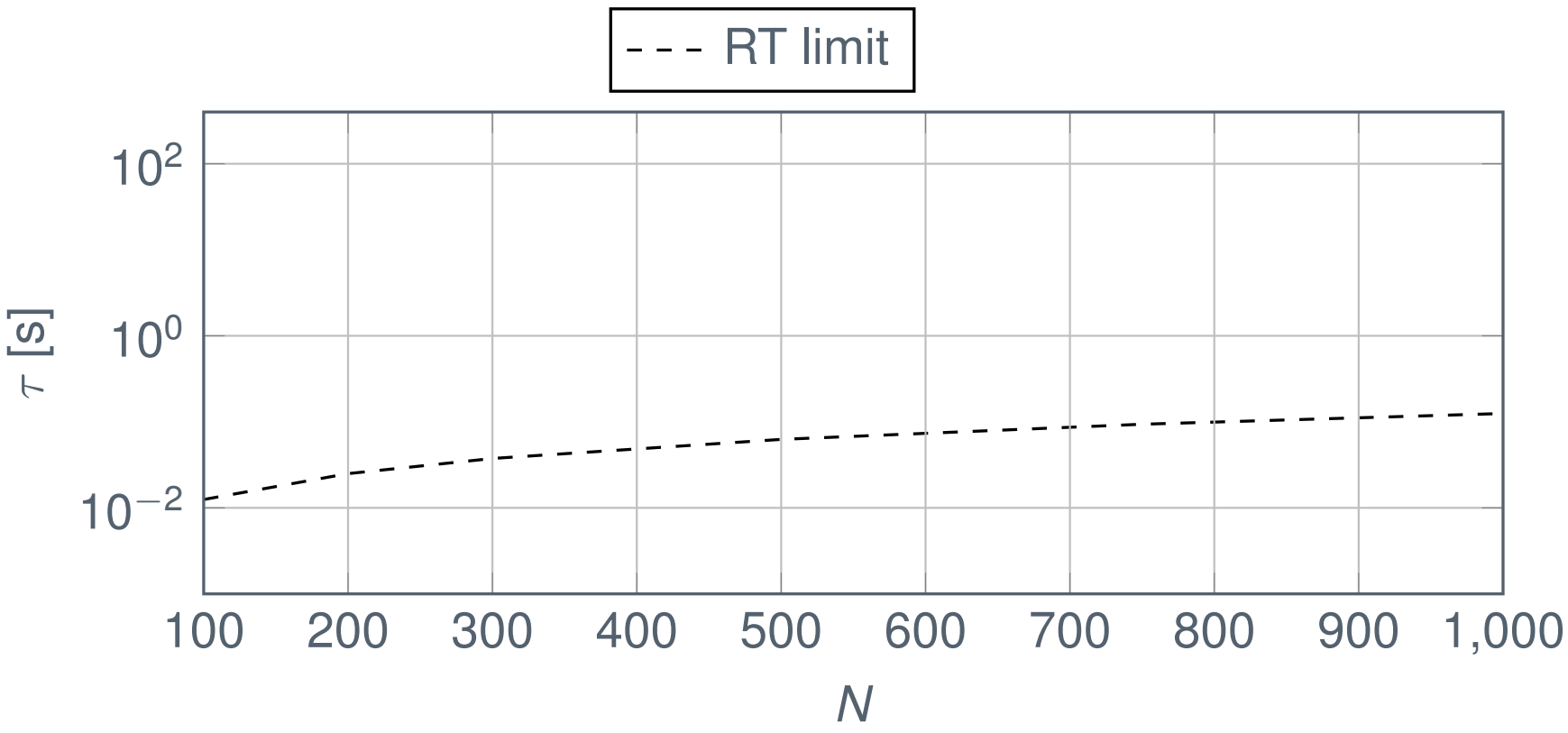


$$\tau(\text{Standard ML}) \approx 60\tau(\text{Fast ML}) \approx 150\tau(\text{Faster ML}) \approx 500\tau(\text{HS})$$

Computation Time vs. Data Size

MATLAB Implementation

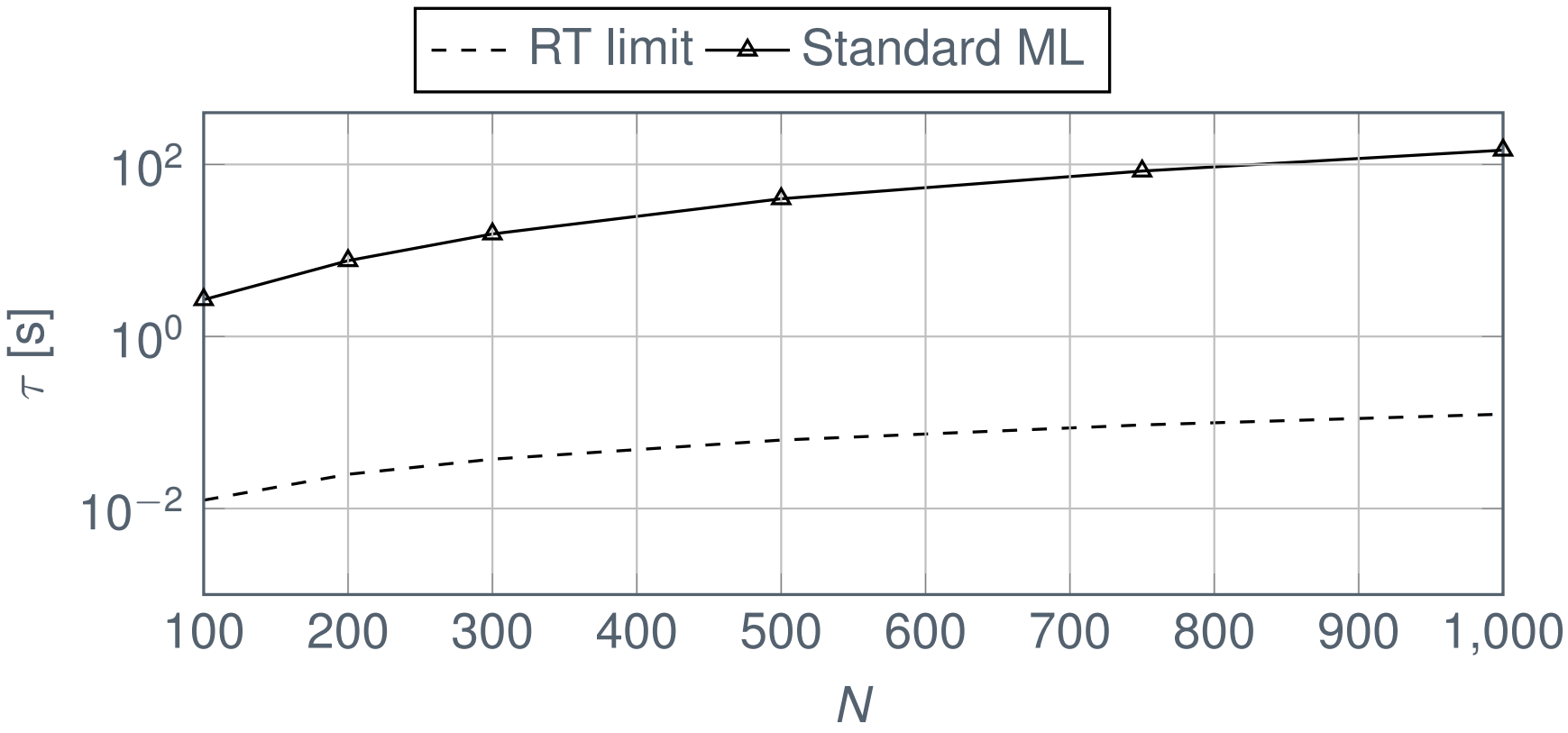
Setup: $L = 30$, $F = 5NL$, T420 laptop



Computation Time vs. Data Size

MATLAB Implementation

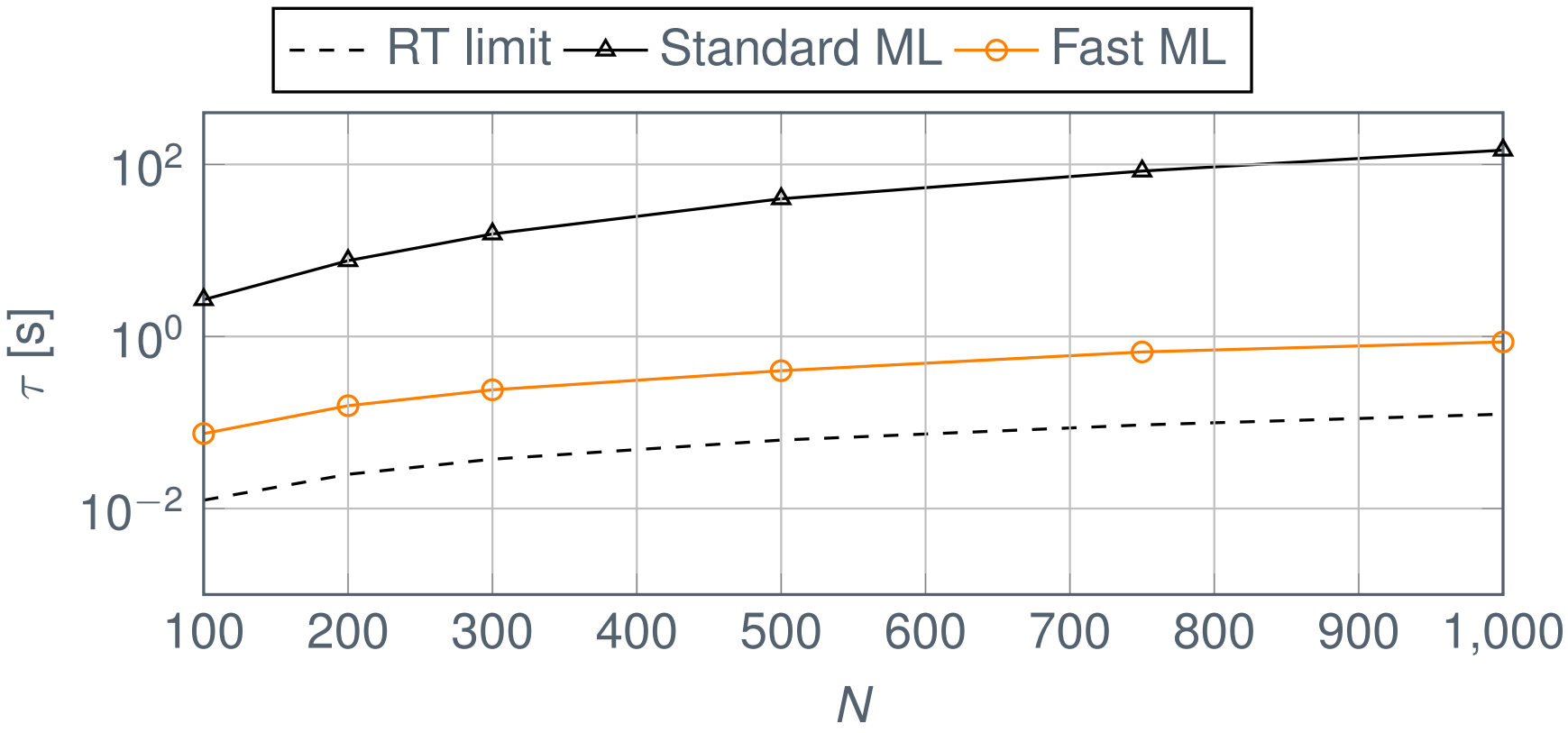
Setup: $L = 30$, $F = 5NL$, T420 laptop



Computation Time vs. Data Size

MATLAB Implementation

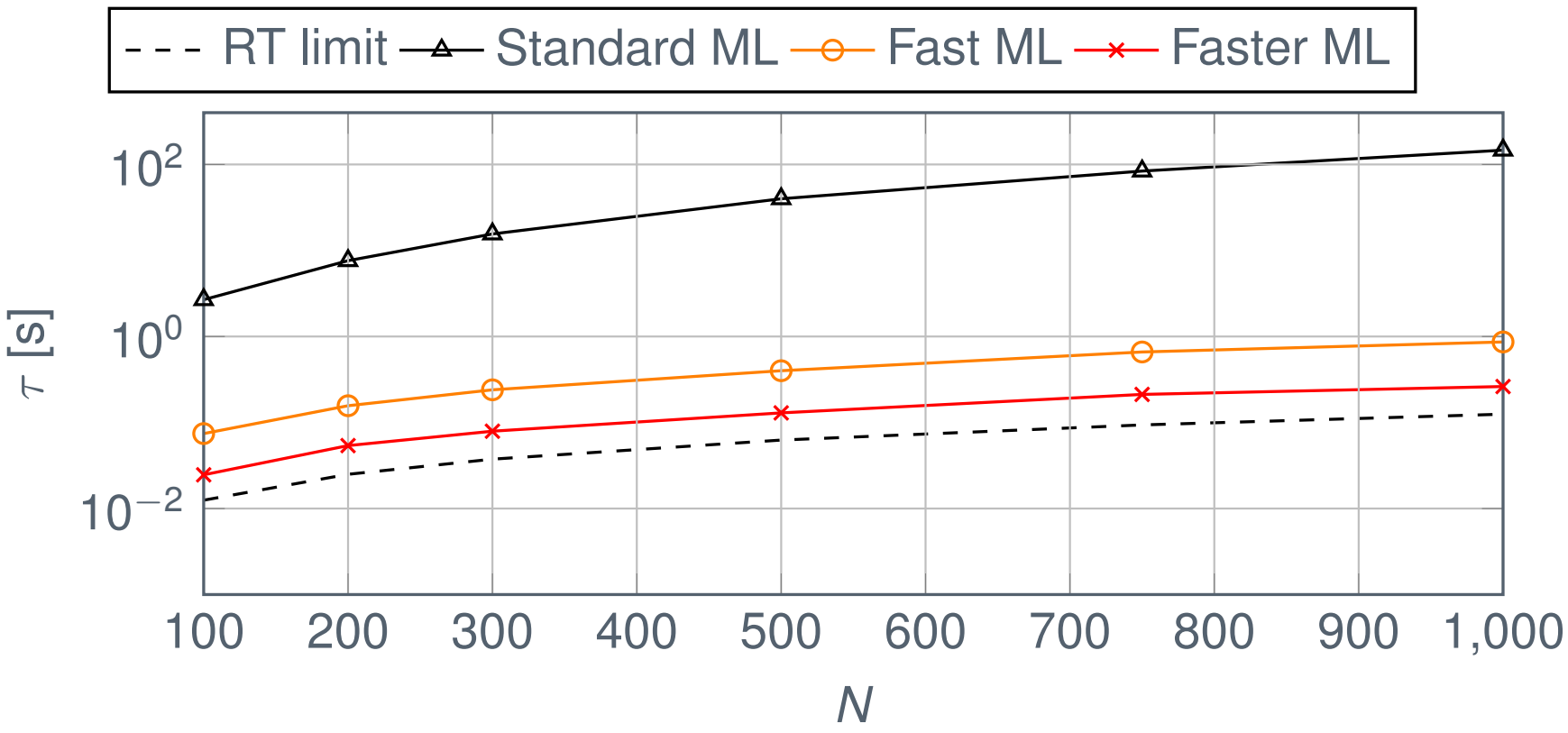
Setup: $L = 30$, $F = 5NL$, T420 laptop



Computation Time vs. Data Size

MATLAB Implementation

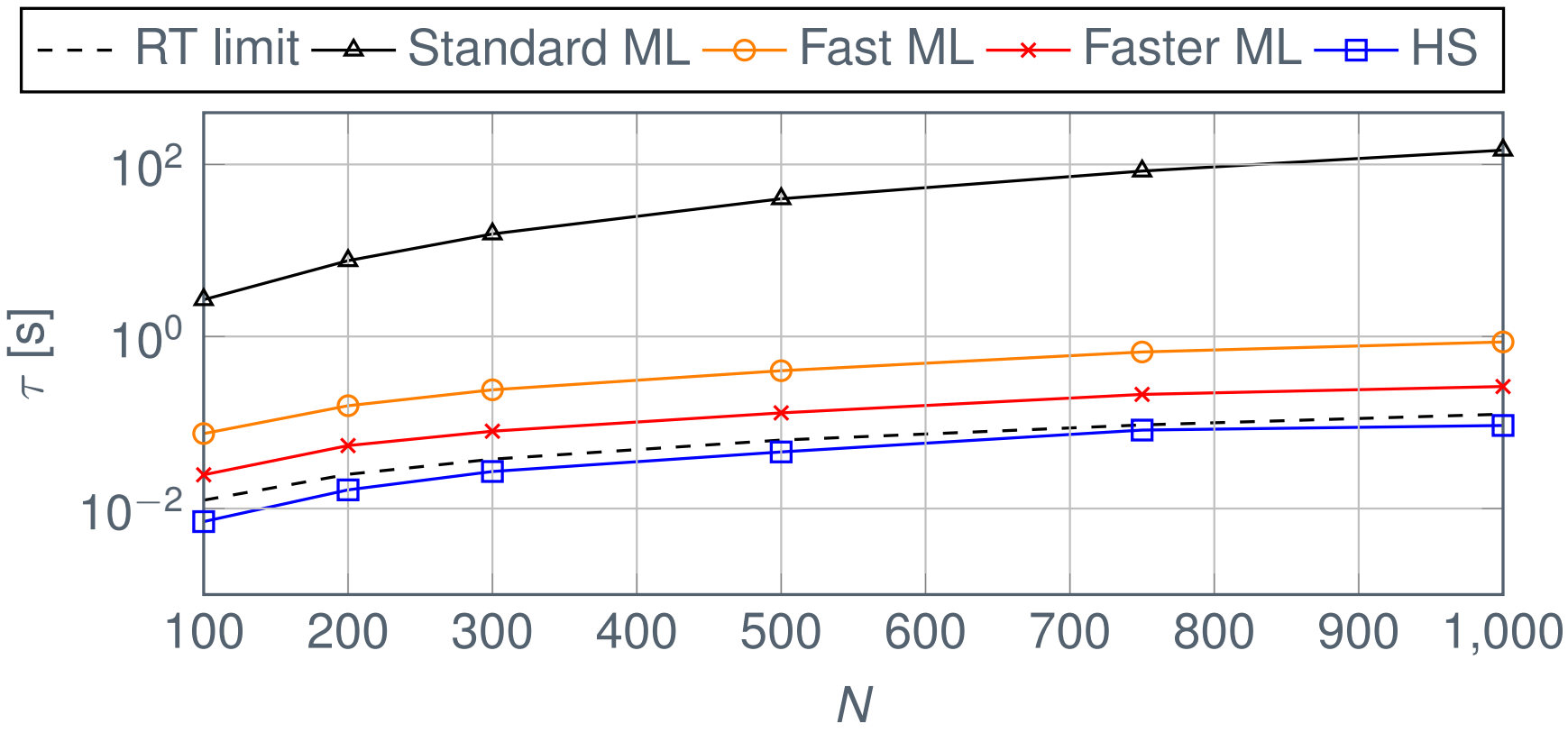
Setup: $L = 30$, $F = 5NL$, T420 laptop



Computation Time vs. Data Size

MATLAB Implementation

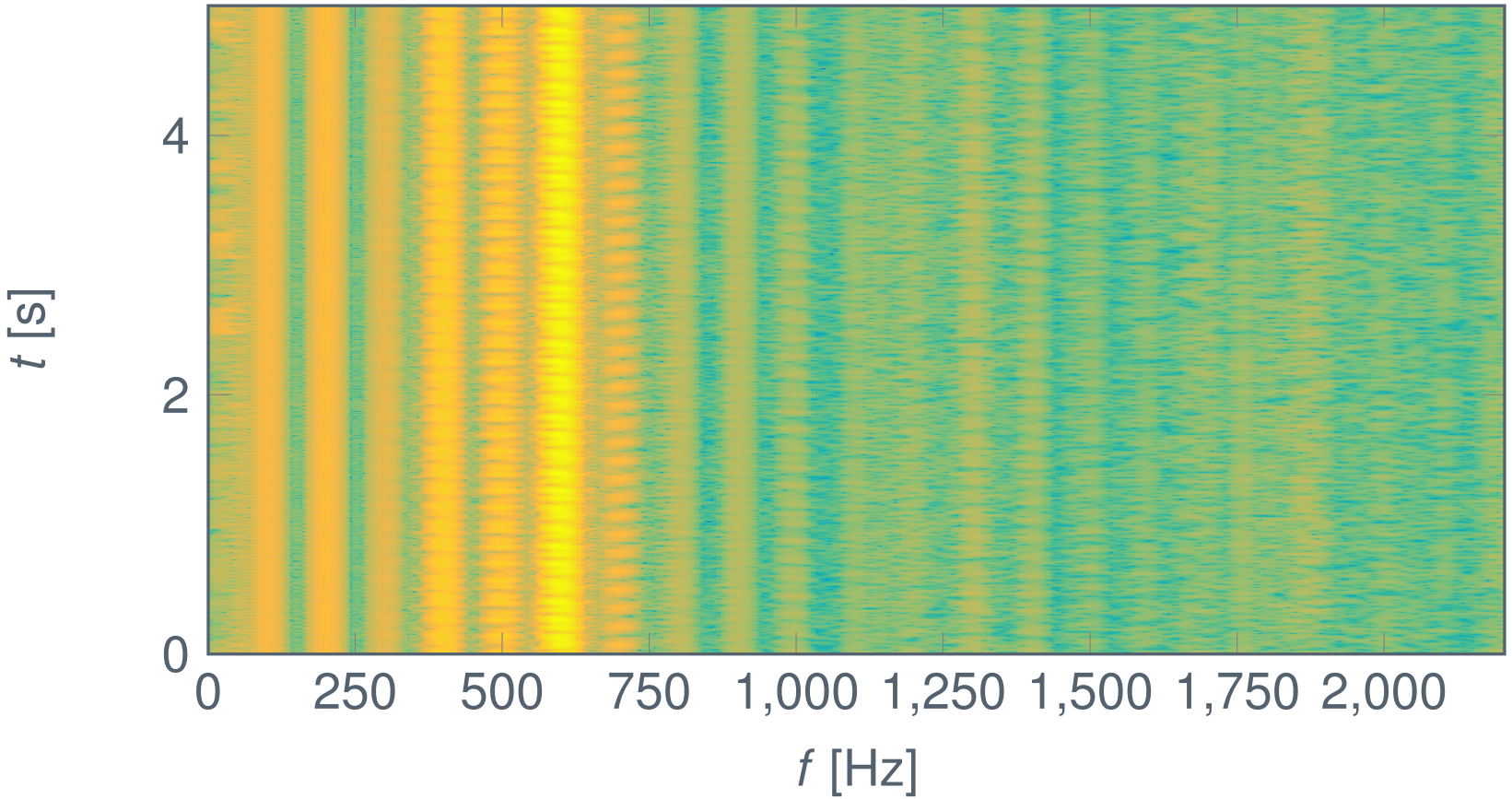
Setup: $L = 30$, $F = 5NL$, T420 laptop



Washing Machine Example

Acoustic measurements by Brüel & Kjær

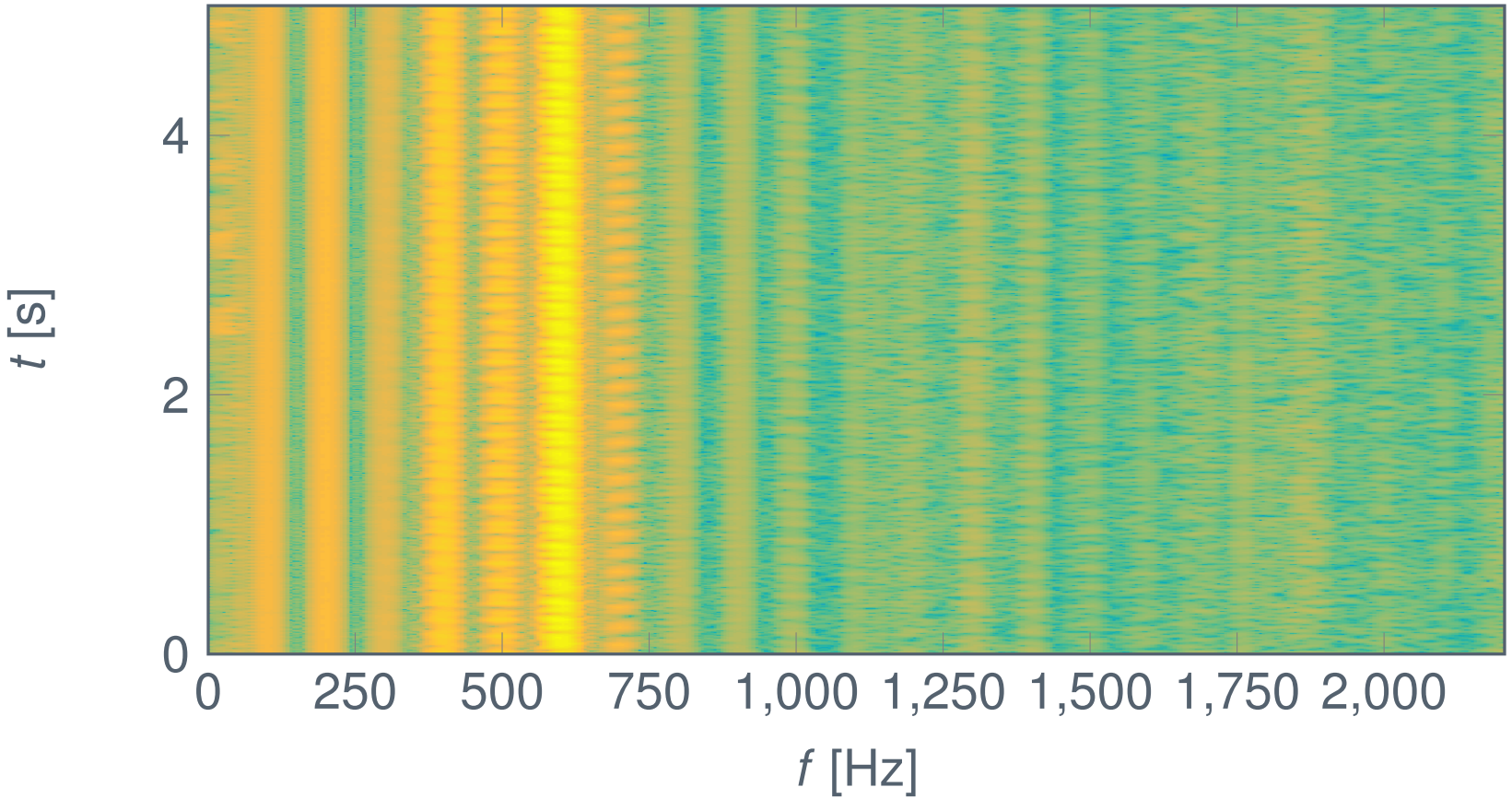
- $f_s = 44.1$ kHz



Washing Machine Example

Acoustic measurements by Brüel & Kjær

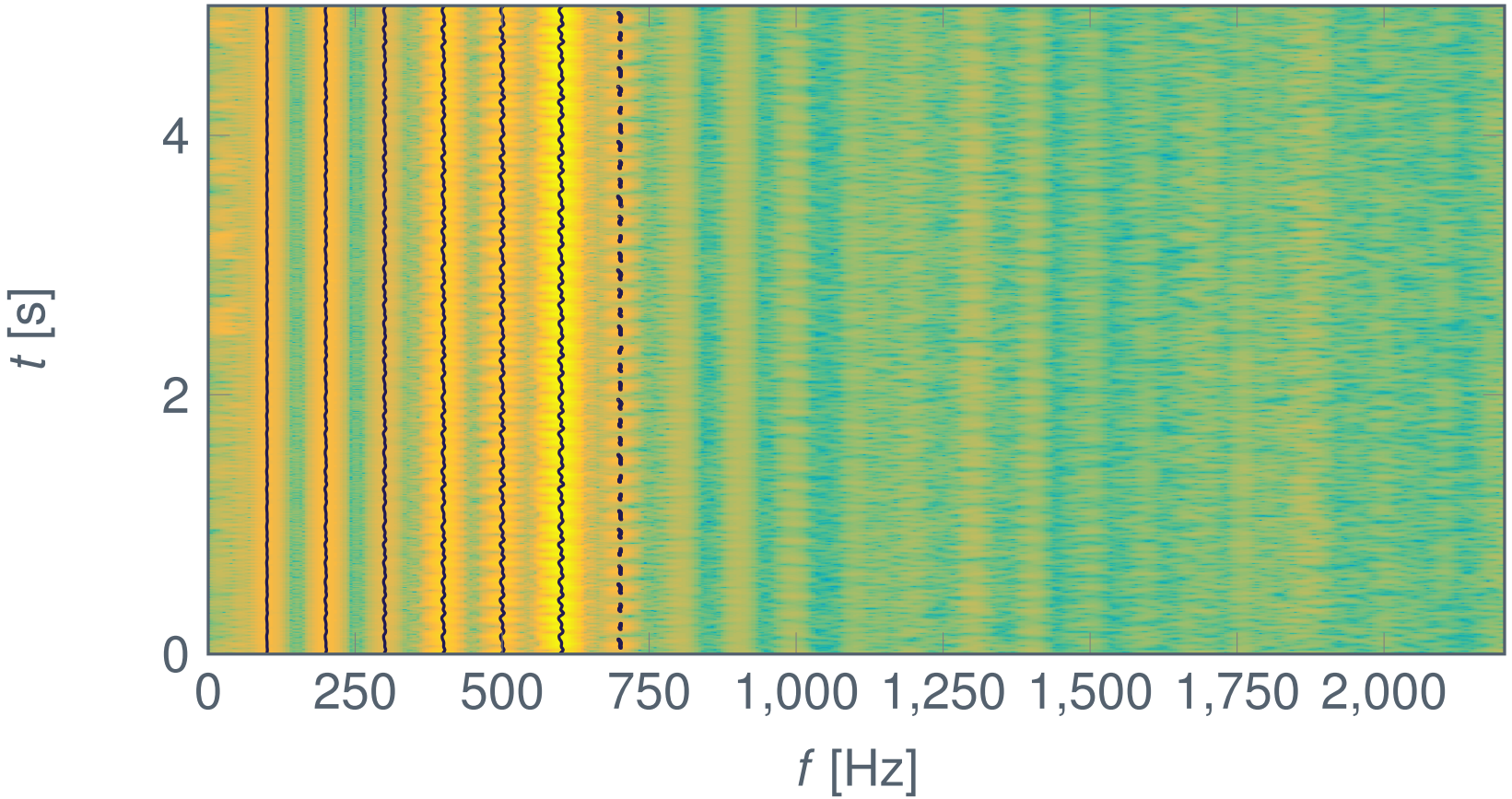
- $f_s = 44.1$ kHz
- $f_{rs} = 4410$ Hz, 60 ms windows, 15/16 overlap, and $L = 15$



Washing Machine Example

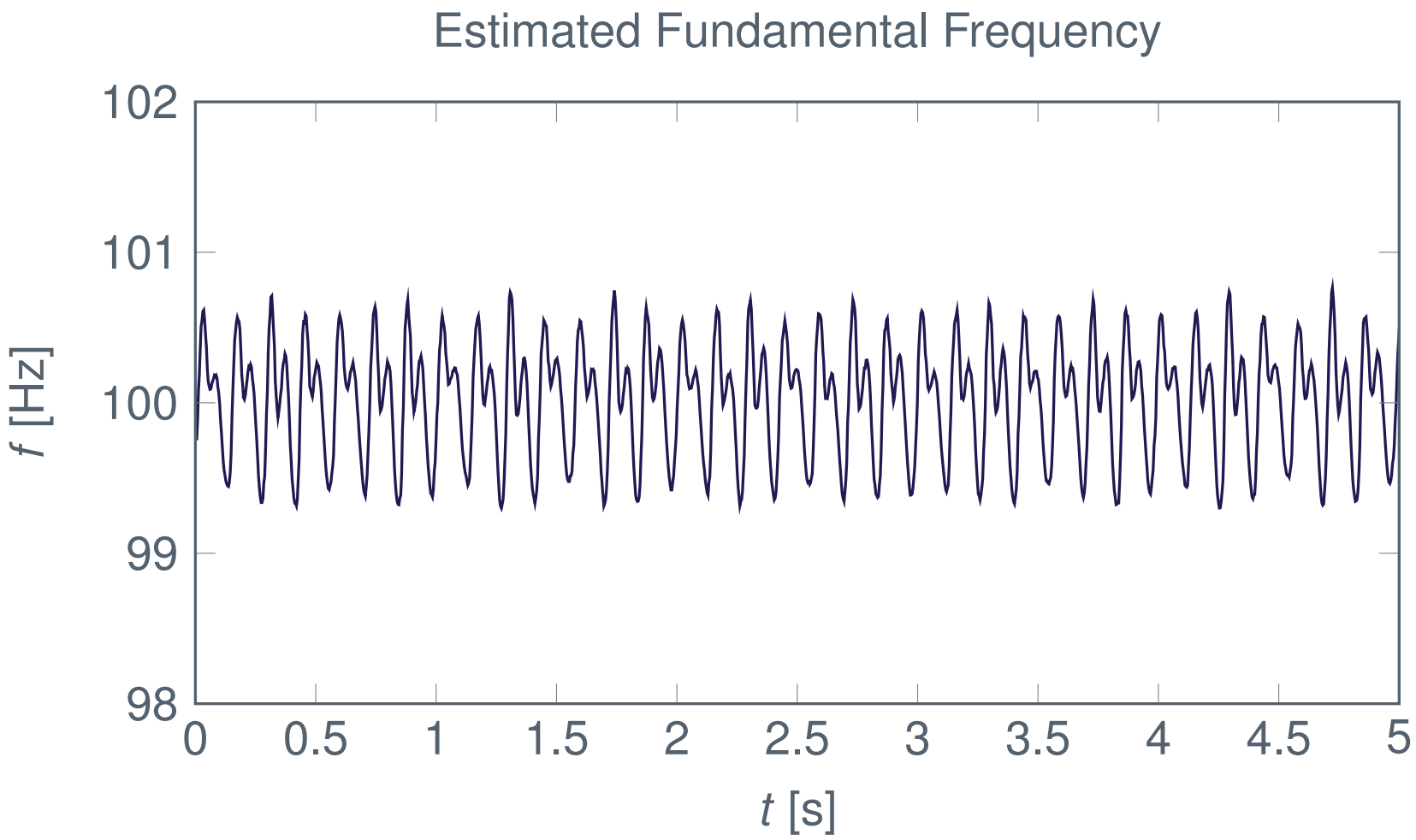
Acoustic measurements by Brüel & Kjær

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



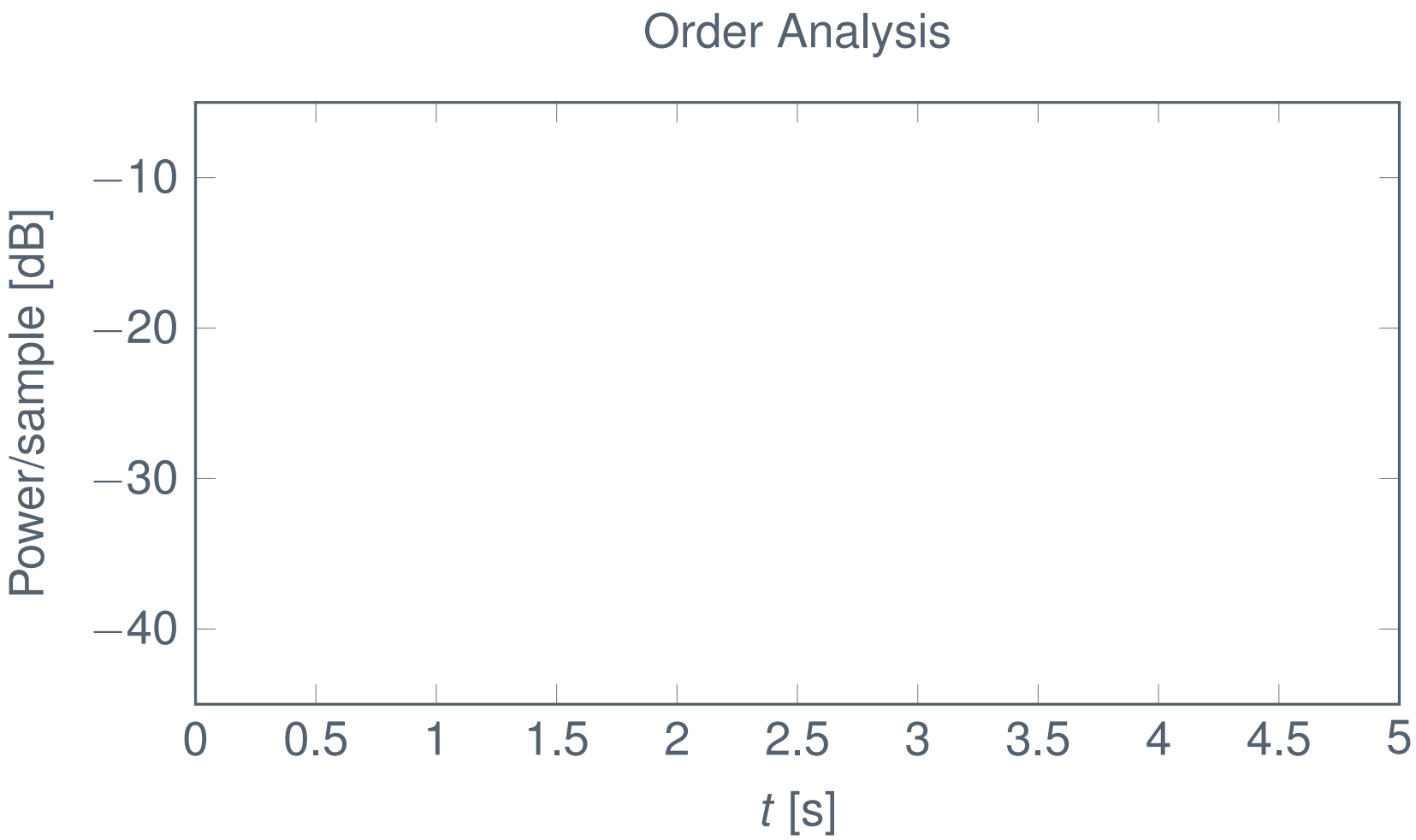
Washing Machine Example

Acoustic measurements by Brüel & Kjær



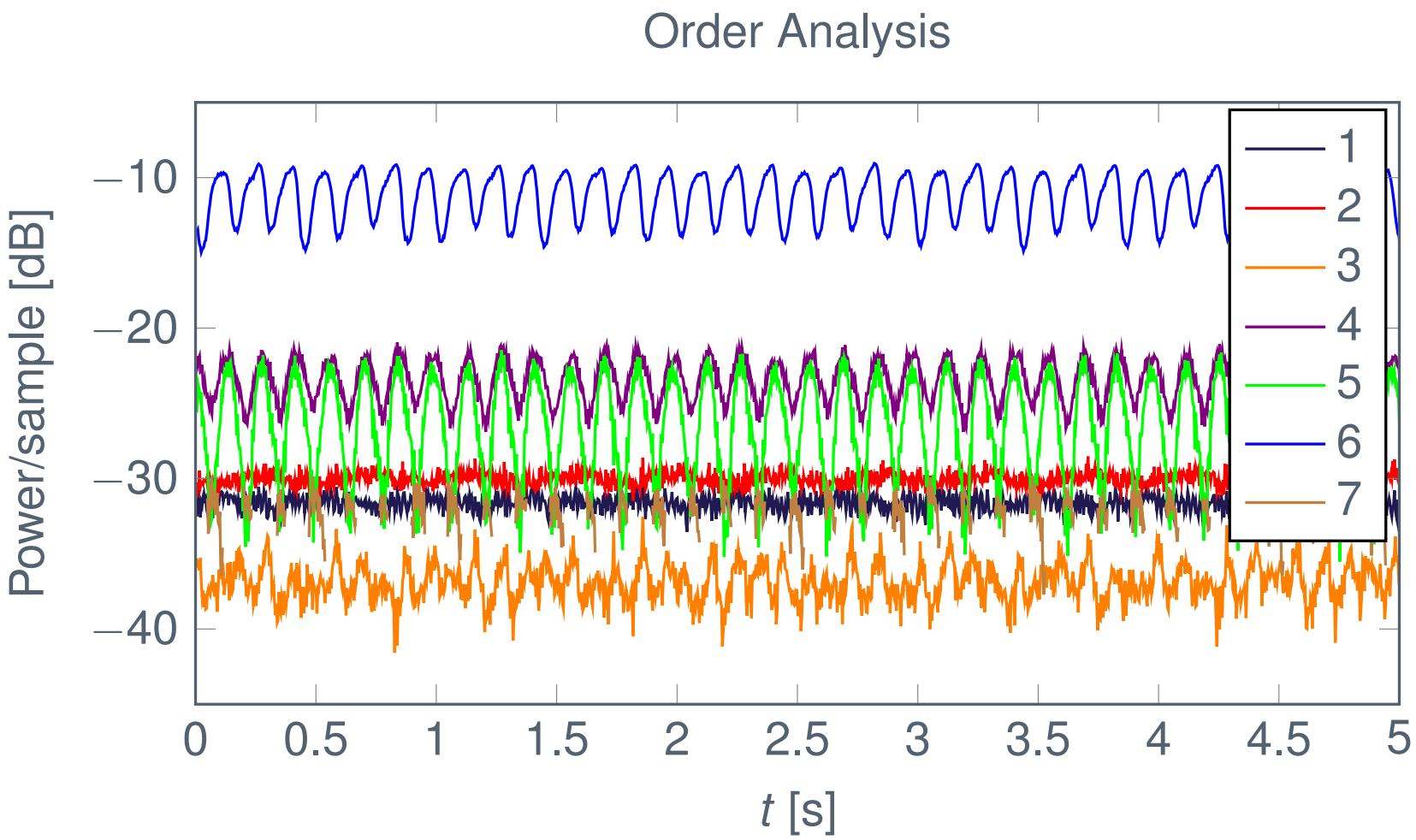
Washing Machine Example

Acoustic measurements by Brüel & Kjær



Washing Machine Example

Acoustic measurements by Brüel & Kjær



Agenda

Introduction
- Grid-size Selection in the Scalar Case

Fundamental Frequency Estimation
- Background on Fundamental Frequency Estimation
- Fast NLS
- Results
- Summary and Demo

Source Localisation
- Grid Size Selection in the Diagonal Case
- Demo

Summary

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

$$\mathcal{O}(F \log F) + \mathcal{O}(FL^3) \tag{33}$$

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

$$\mathcal{O}(F \log F) + \mathcal{O}(FL^3) \tag{33}$$

  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

$$\mathcal{O}(F \log F) + \mathcal{O}(FL) \tag{34}$$

  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

$$\mathcal{O}(F \log F) + \mathcal{O}(FL^3) \tag{33}$$

  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

$$\mathcal{O}(F \log F) + \mathcal{O}(FL) \tag{34}$$

  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.

Agenda

Introduction
- Grid-size Selection in the Scalar Case

Fundamental Frequency Estimation
- Background on Fundamental Frequency Estimation
- Fast NLS
- Results
- Summary and Demo

Source Localisation
- Grid Size Selection in the Diagonal Case
- Demo

Summary

Source Localisation

Signal Model

Free-field signal model for a uniform circular array (UCA):

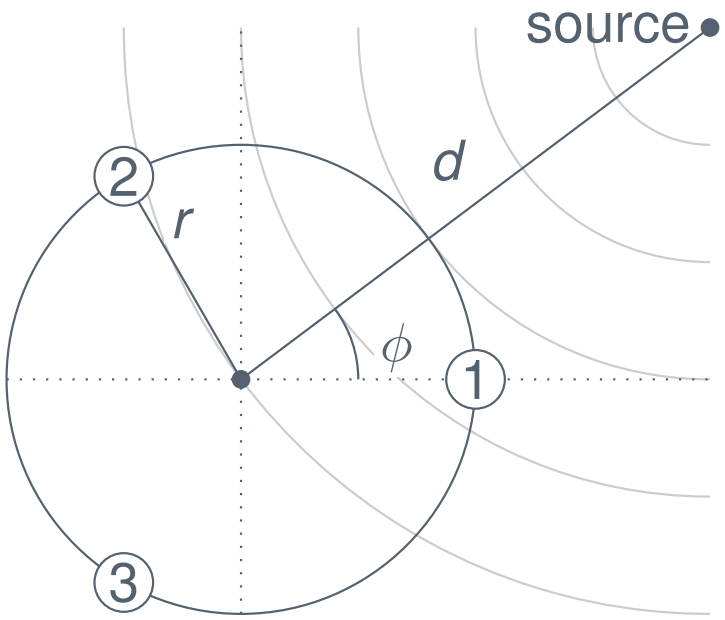
$$\boldsymbol{x}_k = \beta_k \boldsymbol{s}(\eta_k) + \boldsymbol{e}_k \tag{35}$$

for $k = 1, \ldots, K$ where

$$\beta_k = \beta$$

$$\eta_k = \frac{f_s}{c}\left[d - r\cos\left(\phi - 2\pi\frac{k-1}{K}\right)\right]$$

when the source is in the far field.



Source Localisation
Signal Model

Periodic source signal

A periodic source signal can be decomposed as

$$\boldsymbol{s}(\eta_k) = \sum_{i=-I}^{I} \alpha_i \boldsymbol{z}(i\omega_0) \exp(-ji\omega_0\eta_k) \tag{36}$$

where $I = \lfloor N/2 \rfloor$ and $\omega_0 = 2\pi/N$.

Source Localisation
Signal Model

Periodic source signal

A periodic source signal can be decomposed as

$$\boldsymbol{s}(\eta_k) = \sum_{i=-I}^{I} \alpha_i \boldsymbol{z}(i\omega_0) \exp(-ji\omega_0\eta_k) \tag{36}$$

where $I = \lfloor N/2 \rfloor$ and $\omega_0 = 2\pi/N$. Define that

$$\xi = d\frac{\omega_0 f_s}{c} \in [0, 2\pi) \tag{37}$$

$$\rho = r\frac{\omega_0 f_s}{c} \in [0, 2\pi) \tag{38}$$

so that

$$\omega_0\eta_k = \xi - \rho\cos\left(\phi - 2\pi\frac{k-1}{K}\right) \tag{39}$$

Source Localisation

NLS Estimator

$$(\hat{\xi}, \hat{\phi}) = \underset{\xi \in [0,2\pi), \phi \in [0,2\pi)}{\operatorname{argmax}} \quad J(\xi, \phi) \tag{40}$$

$$J(\xi, \phi) = \left| \sum_{k=1}^{K} \boldsymbol{s}^T(\eta_k) \boldsymbol{x} \right|^2 = \left| \boldsymbol{f}^H(\xi) \boldsymbol{A}^H \sum_{k=1}^{K} \boldsymbol{D}_k^H(\phi) \boldsymbol{F}^H \boldsymbol{x}_k \right|^2 \tag{41}$$

Source Localisation

NLS Estimator

$$(\hat{\xi}, \hat{\phi}) = \underset{\xi \in [0,2\pi), \phi \in [0,2\pi)}{\operatorname{argmax}} \quad J(\xi, \phi) \tag{40}$$

$$J(\xi, \phi) = \left| \sum_{k=1}^{K} \boldsymbol{s}^T(\eta_k) \boldsymbol{x} \right|^2 = \left| \boldsymbol{f}^H(\xi) \boldsymbol{A}^H \sum_{k=1}^{K} \boldsymbol{D}_k^H(\phi) \boldsymbol{F}^H \boldsymbol{x}_k \right|^2 \tag{41}$$

where we have defined

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{z}(-2\pi l/N) & \cdots & \boldsymbol{z}(2\pi l/N) \end{bmatrix} \tag{42}$$

$$\boldsymbol{f}(\xi) = \begin{bmatrix} \exp(jl\xi) & \cdots & \exp(-jl\xi) \end{bmatrix}^T \tag{43}$$

$$\boldsymbol{D}_k(\phi) = \operatorname{diag}\left(\boldsymbol{f}\left(-\rho \cos\left(\phi - 2\pi \frac{k-1}{K} \right) \right) \right) \tag{44}$$

$$\boldsymbol{A} = \operatorname{diag}(\alpha_{-l}, \ldots, \alpha_l) = N^{-1} \operatorname{diag}\left(\boldsymbol{F}^H \boldsymbol{s}(0) \right) \tag{45}$$

Source Localisation

NLS Estimator

Task

We wish to solve

$$(\hat{\xi}, \hat{\phi}) = \underset{\xi \in [0,2\pi), \phi \in [0,2\pi)}{\operatorname{argmax}} \; J(\xi, \phi) \tag{46}$$

using

1. a coarse 2D grid search
2. a 2D refinement

Source Localisation
NLS Estimator

Task
We wish to solve

$$(\hat{\xi}, \hat{\phi}) = \underset{\xi \in [0,2\pi), \phi \in [0,2\pi)}{\operatorname{argmax}} \; J(\xi, \phi) \tag{46}$$

using

1. a coarse 2D grid search
2. a 2D refinement

Question
How should we select the grid sizes for ξ and ϕ?

Agenda

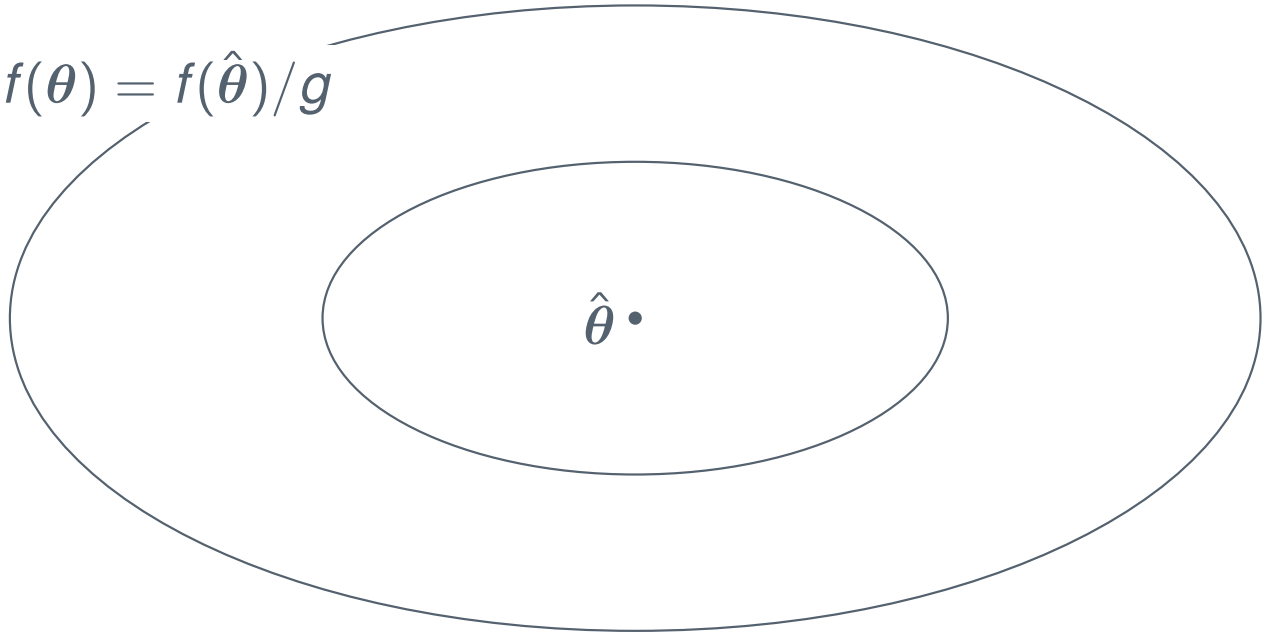
Introduction
- Grid-size Selection in the Scalar Case

Fundamental Frequency Estimation
- Background on Fundamental Frequency Estimation
- Fast NLS
- Results
- Summary and Demo

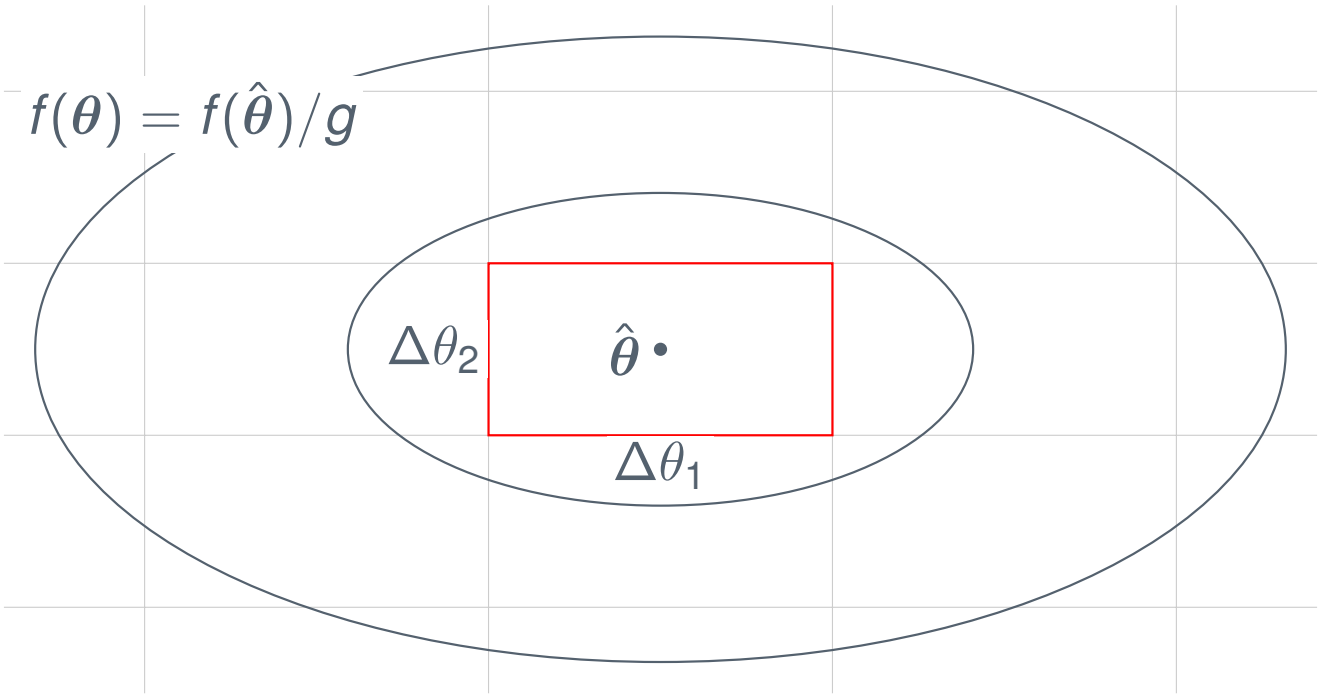
Source Localisation
- Grid Size Selection in the Diagonal Case
- Demo

Summary

Grid Size Selection in the Diagonal Case



Grid Size Selection in the Diagonal Case



Grid Size Selection in the Diagonal Case

Taylor approximation

$$f(\boldsymbol{\theta}) = \frac{f(\hat{\boldsymbol{\theta}})}{g} \approx f(\hat{\boldsymbol{\theta}}) + \frac{1}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T \boldsymbol{H}(\hat{\boldsymbol{\theta}})(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) \tag{47}$$

Grid Size Selection in the Diagonal Case

Taylor approximation

$$f(\boldsymbol{\theta}) = \frac{f(\hat{\boldsymbol{\theta}})}{g} \approx f(\hat{\boldsymbol{\theta}}) + \frac{1}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T \boldsymbol{H}(\hat{\boldsymbol{\theta}})(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) \tag{47}$$

If the Hessian is diagonal, we have that

$$\theta_p = \hat{\theta}_p \pm \Delta\theta_p = \hat{\theta}_p \pm \sqrt{2\frac{1-g}{g}\frac{f(\hat{\boldsymbol{\theta}})}{\left[\boldsymbol{H}(\hat{\boldsymbol{\theta}})\right]_{pp}}} \tag{48}$$

Grid Size Selection in the Diagonal Case

Example: Source Localisation

We have that

$$f(\hat{\xi}, \hat{\phi}) = K^2 \hat{\beta}^2 \left[\boldsymbol{s}^T(0) \boldsymbol{s}(0) \right]^2 \tag{49}$$

$$\boldsymbol{H}(\hat{\xi}, \hat{\phi}) \approx -2K^2 \hat{\beta}^2 \boldsymbol{s}^T(0) \boldsymbol{s}(0) N \sum_{i=-I}^{I} |\alpha_i|^2 i^2 \begin{bmatrix} 1 & 0 \\ 0 & \rho^2/2 \end{bmatrix} \tag{50}$$

Grid Size Selection in the Diagonal Case

Example: Source Localisation

We have that

$$f(\hat{\xi}, \hat{\phi}) = K^2 \hat{\beta}^2 \left[\boldsymbol{s}^T(0) \boldsymbol{s}(0) \right]^2 \tag{49}$$

$$\boldsymbol{H}(\hat{\xi}, \hat{\phi}) \approx -2K^2 \hat{\beta}^2 \boldsymbol{s}^T(0) \boldsymbol{s}(0) N \sum_{i=-I}^{I} |\alpha_i|^2 i^2 \begin{bmatrix} 1 & 0 \\ 0 & \rho^2/2 \end{bmatrix} \tag{50}$$

Thus,

$$\Delta\xi \approx \sqrt{\frac{g-1}{g} \frac{N^{-1} P_s}{\sum_{i=-I}^{I} |\alpha_i|^2 i^2}} > \sqrt{2 \frac{g-1}{g} \frac{1}{N}} \tag{51}$$

$$\Delta\phi \approx \sqrt{\frac{g-1}{g} \frac{2N^{-1} P_s}{\rho^2 \sum_{i=-I}^{I} |\alpha_i|^2 i^2}} > \sqrt{2 \frac{g-1}{g} \frac{\sqrt{2}}{\rho N}} \tag{52}$$

Grid Size Selection in the Diagonal Case

Example: Source Localisation

We have that

$$f(\hat{\xi}, \hat{\phi}) = K^2 \hat{\beta}^2 \left[\boldsymbol{s}^T(0) \boldsymbol{s}(0) \right]^2 \tag{49}$$

$$\boldsymbol{H}(\hat{\xi}, \hat{\phi}) \approx -2K^2 \hat{\beta}^2 \boldsymbol{s}^T(0) \boldsymbol{s}(0) N \sum_{i=-I}^{I} |\alpha_i|^2 i^2 \begin{bmatrix} 1 & 0 \\ 0 & \rho^2/2 \end{bmatrix} \tag{50}$$

Thus,

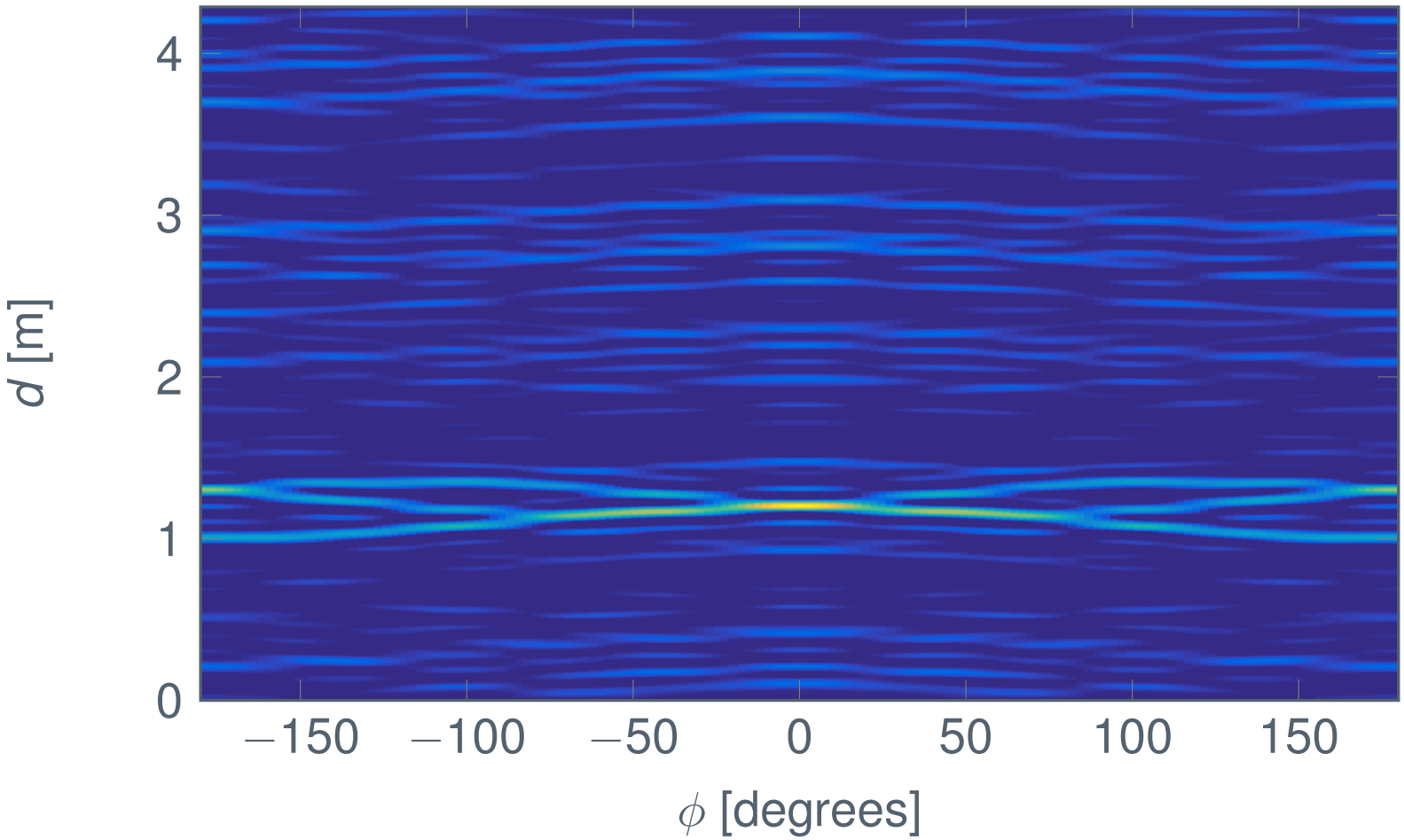
$$\Delta\xi \approx \sqrt{\frac{g-1}{g} \frac{N^{-1} P_s}{\sum_{i=-I}^{I} |\alpha_i|^2 i^2}} > \sqrt{2 \frac{g-1}{g} \frac{1}{N}} \tag{51}$$

$$\Delta\phi \approx \sqrt{\frac{g-1}{g} \frac{2N^{-1} P_s}{\rho^2 \sum_{i=-I}^{I} |\alpha_i|^2 i^2}} > \sqrt{2 \frac{g-1}{g} \frac{\sqrt{2}}{\rho N}} \tag{52}$$

The grid size does not depend on the number of sensors K!

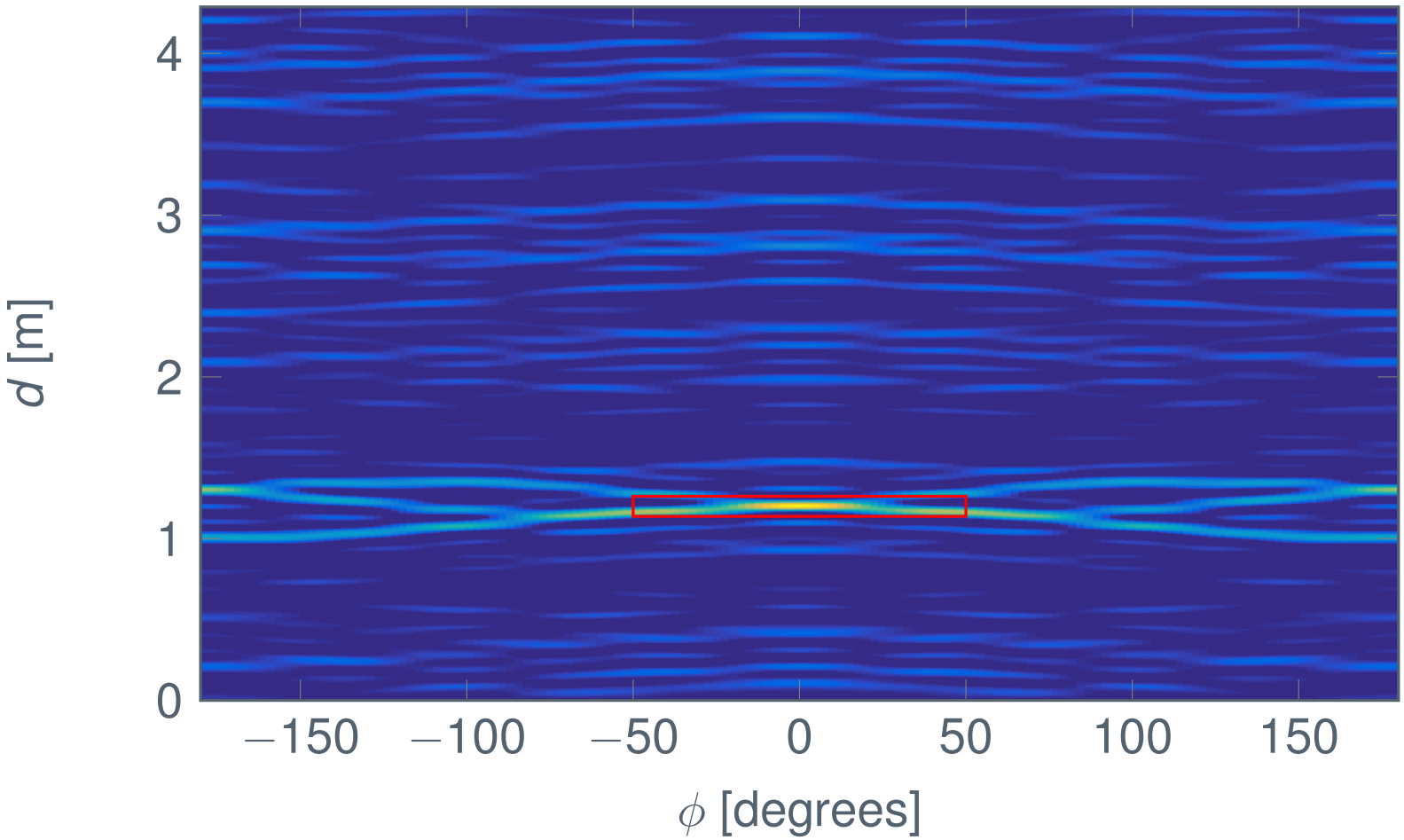
Grid Size Selection in the Diagonal Case

Example: Source Localisation



Grid Size Selection in the Diagonal Case

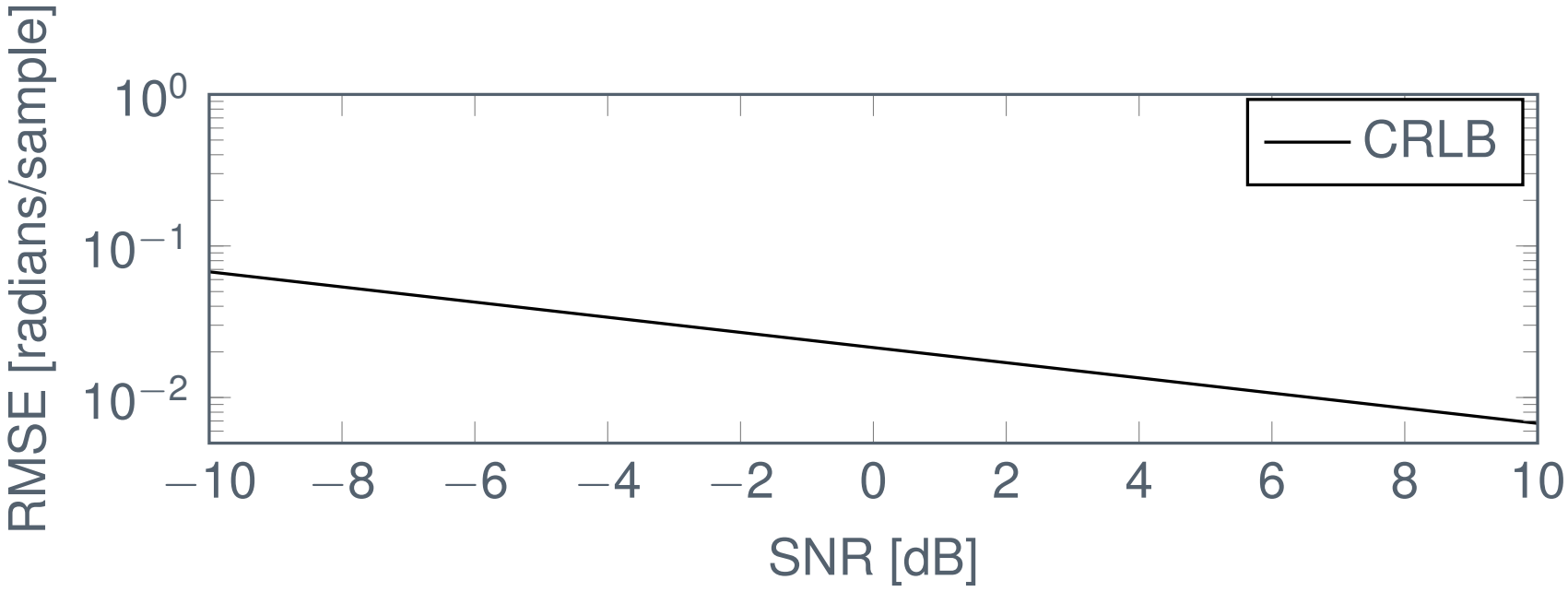
Example: Source Localisation



Grid Size Selection in the Diagonal Case

Example: Source Localisation

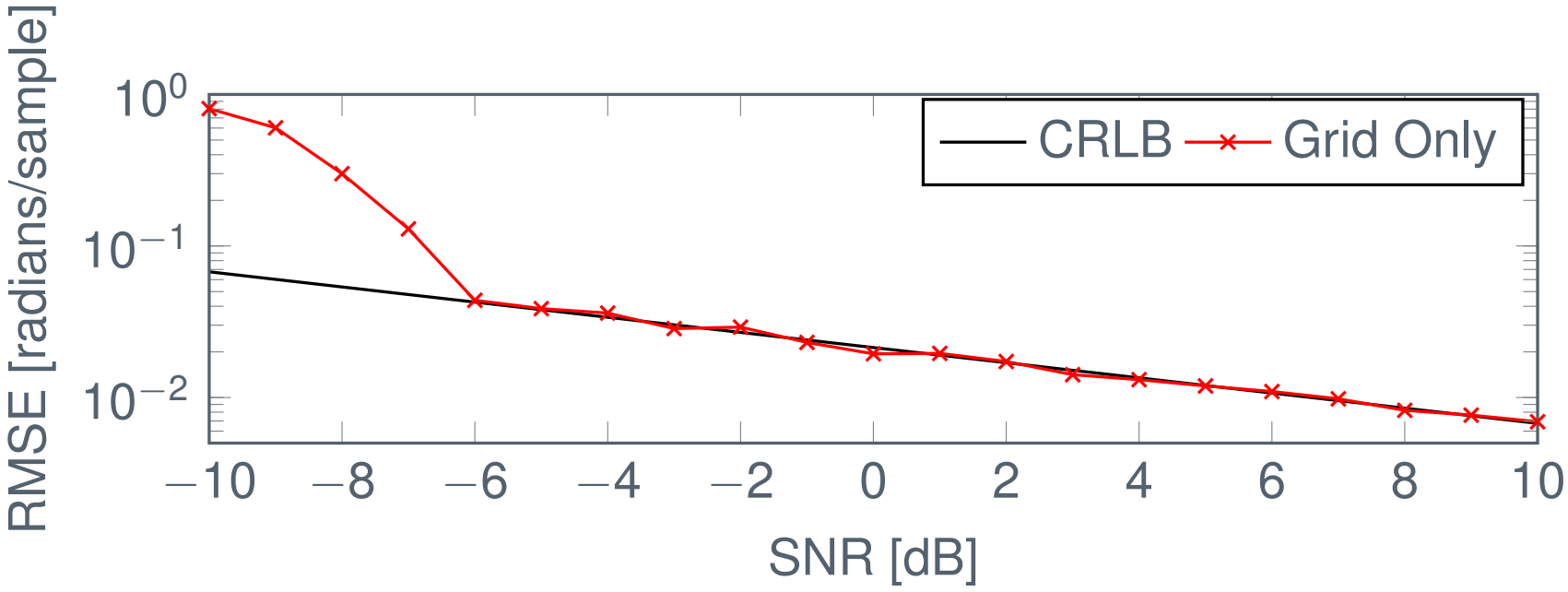
Setup: $N = 100$, $K = 3$, 250 Monte Carlo runs for every g and SNR, and $r = 0.1$.



Grid Size Selection in the Diagonal Case

Example: Source Localisation

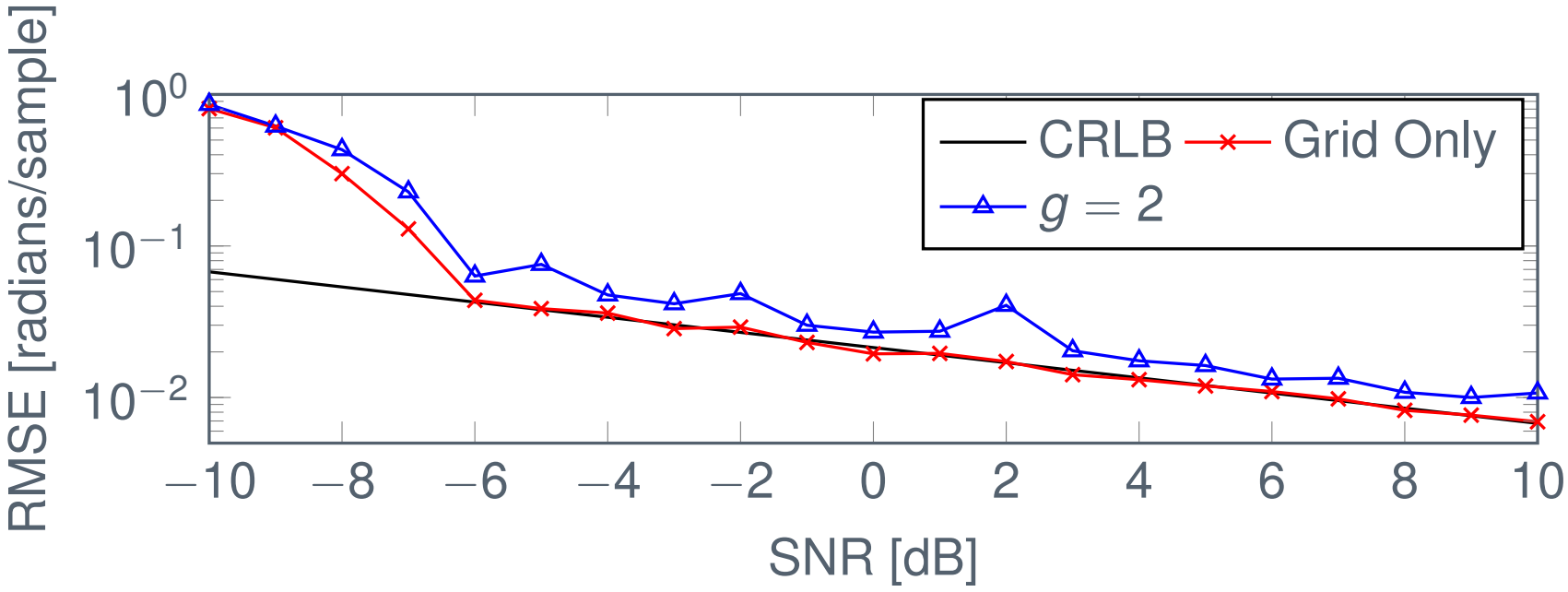
Setup: $N = 100$, $K = 3$, 250 Monte Carlo runs for every g and SNR, and $r = 0.1$.



Grid Size Selection in the Diagonal Case

Example: Source Localisation

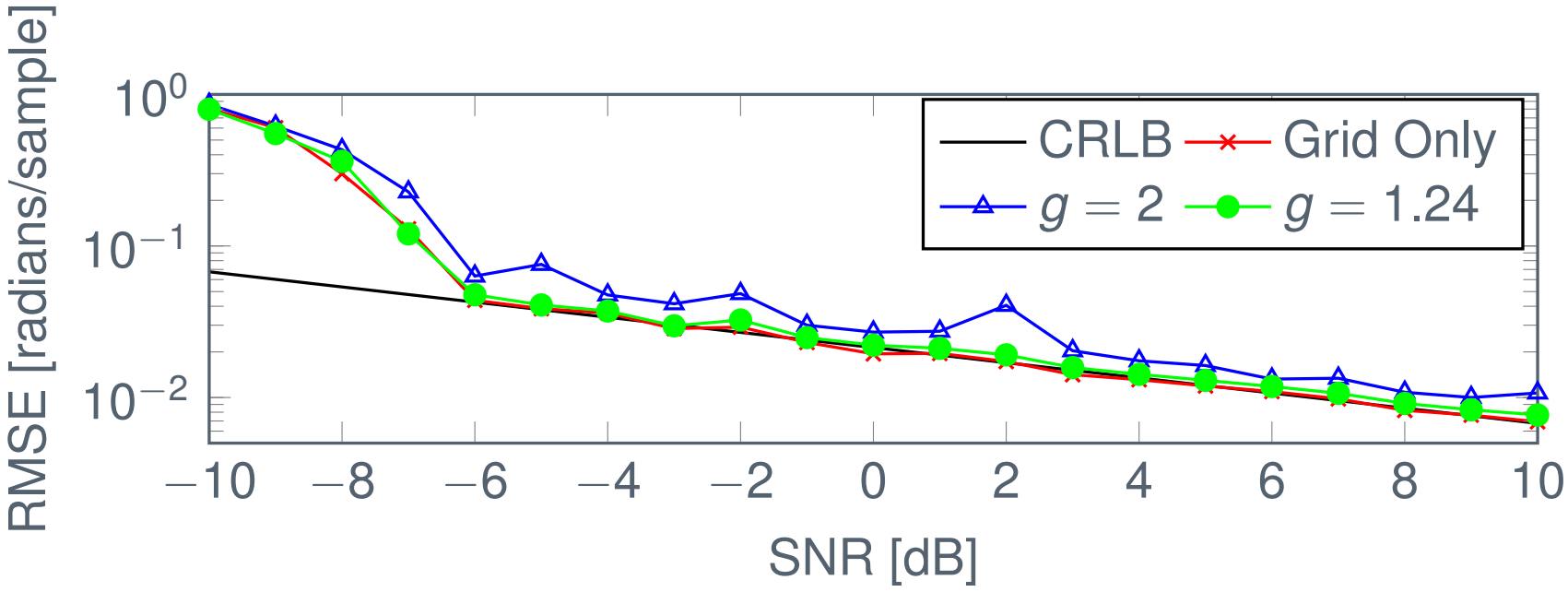
Setup: $N = 100$, $K = 3$, 250 Monte Carlo runs for every g and SNR, and $r = 0.1$.



Grid Size Selection in the Diagonal Case

Example: Source Localisation

Setup: $N = 100$, $K = 3$, 250 Monte Carlo runs for every g and SNR, and $r = 0.1$.



Grid Size Selection in the Diagonal Case

Example: Source Localisation

g	10	2	1.24	1.024	1.0024	Grid Only
# grid points ξ	192	258	402	1208	3798	16126
# grid points ϕ	15	19	30	89	279	1671
τ [s]	0.004	0.004	0.006	0.018	0.105	1.985

Grid Size Selection in the Diagonal Case

Example: Source Localisation

g	10	2	1.24	1.024	1.0024	Grid Only
# grid points ξ	192	258	402	1208	3798	16126
# grid points ϕ	15	19	30	89	279	1671
τ [s]	0.004	0.004	0.006	0.018	0.105	1.985

The grid only approach is approximately 350 times slower than the combined approach for $g = 1.24$.

Agenda

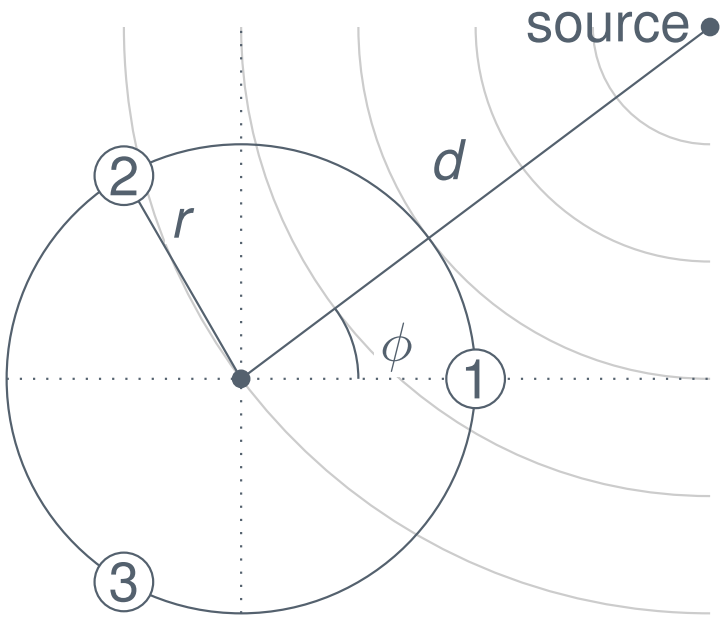
Introduction
- Grid-size Selection in the Scalar Case

Fundamental Frequency Estimation
- Background on Fundamental Frequency Estimation
- Fast NLS
- Results
- Summary and Demo

Source Localisation
- Grid Size Selection in the Diagonal Case
- Demo

Summary

Demo
Source Localisation



Agenda

Introduction
- Grid-size Selection in the Scalar Case

Fundamental Frequency Estimation
- Background on Fundamental Frequency Estimation
- Fast NLS
- Results
- Summary and Demo

Source Localisation
- Grid Size Selection in the Diagonal Case
- Demo

Summary

Summary

- In many spectral array processing problems, the grid method is attractive due to the FFT (fast cost function evaluations).

Summary

- In many spectral array processing problems, the grid method is attractive due to the FFT (fast cost function evaluations).
- A two-step approach consisting of
  1. a coarse grid search and
  2. a local refinement

  is potentially several orders of magnitude faster than a fine grid search (fewer cost function evaluations).

Summary

- In many spectral array processing problems, the grid method is attractive due to the FFT (fast cost function evaluations).
- A two-step approach consisting of
  1. a coarse grid search and
  2. a local refinement

  is potentially several orders of magnitude faster than a fine grid search (fewer cost function evaluations).
- We get (nearly) the same accuracy.

Thanks for your attention!

AALBORG UNIVERSITY
DENMARK