Signal Processing for Interactive Systems

Lecture 5

Cumhur Ekut

cer@create.aau.dk

AALBORG UNIVERSITY

DENMARK

Agenda

Agenda

The harmonic model

Motivation

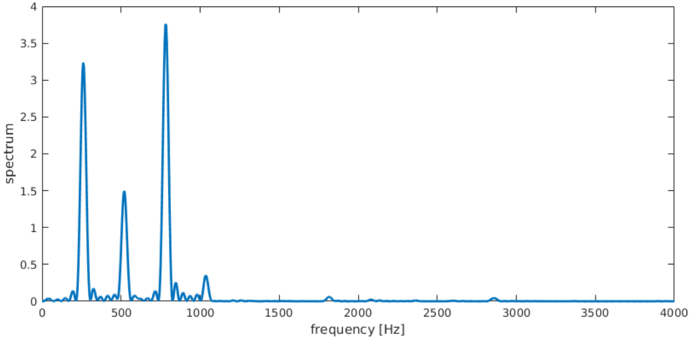
In about 20 minutes, you will know

- how the spectrum of a periodic signal is
- how periodic signals can be modelled
- the matrix form of the harmonic model

The harmonic model

Structure of the spectrum of periodic signals

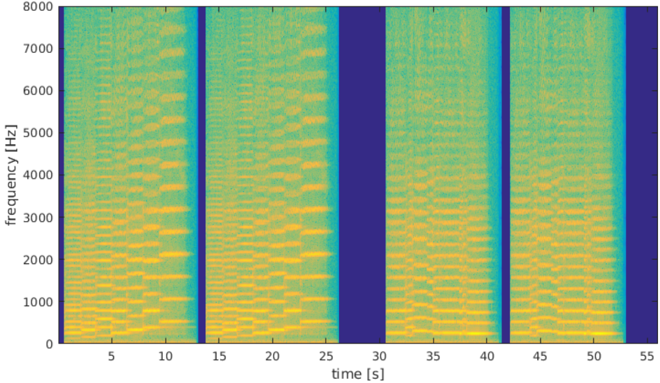
What is the structure in the spectrum of the music signal?



The harmonic model

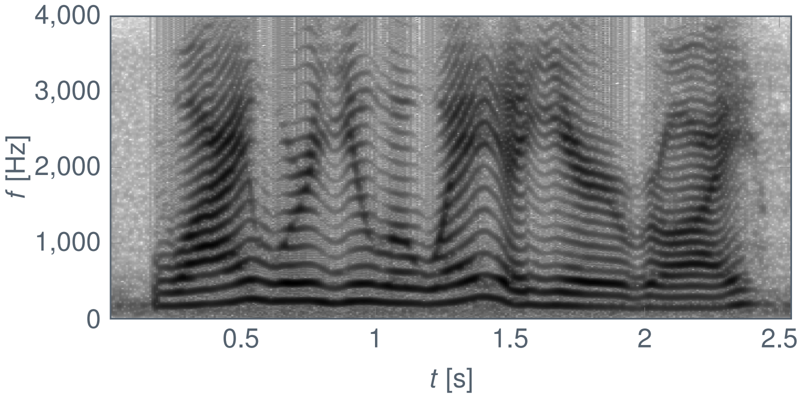
Structure of the spectrum of periodic signals

What is the structure in the spectrum of the music signal?

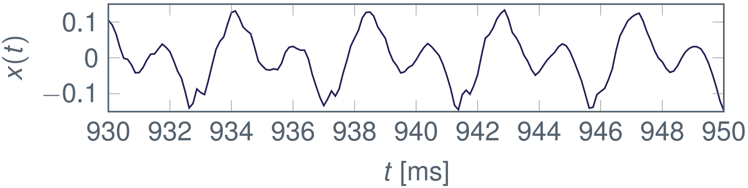


The harmonic model

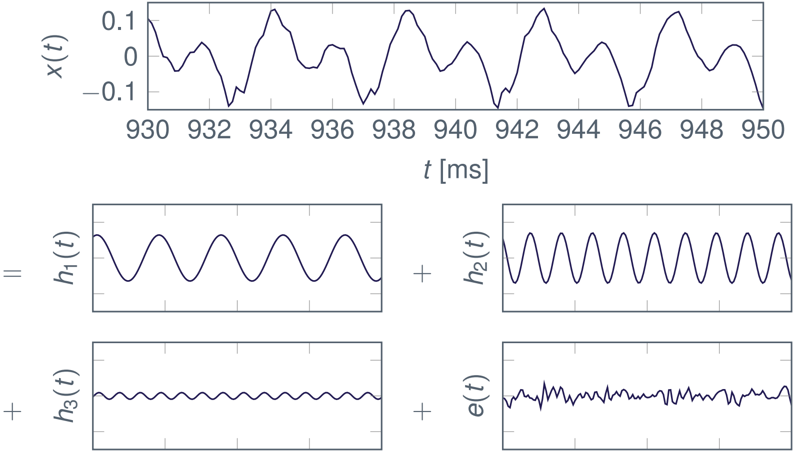
Example (Speech signal)



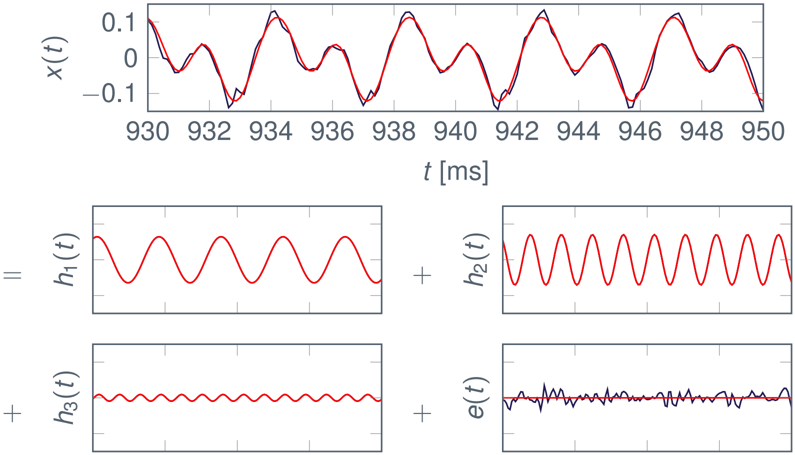
The harmonic model



The harmonic model



The harmonic model



The harmonic model

The signal model for any periodic signal is

$$s(n) = \sum_{l=1}^{L} h_l(n) = \sum_{l=1}^{L} A_l \cos(l\omega_0 n + \phi_l) \tag{1}$$

where

- ω_0 fundamental frequency or pitch in radians/sample
- L number of harmonic components (or model order)
- A_l amplitude of l'th harmonic component
- ϕ_l phase of l'th harmonic component

The harmonic model

The harmonic model is

$$s(n) = \sum_{l=1}^{L} h_l(n) = \sum_{l=1}^{L} A_l \cos(l\omega_0 n + \phi_l) \ . \tag{2}$$

What are we assuming?

The signal is perfectly periodic

The pitch is constant

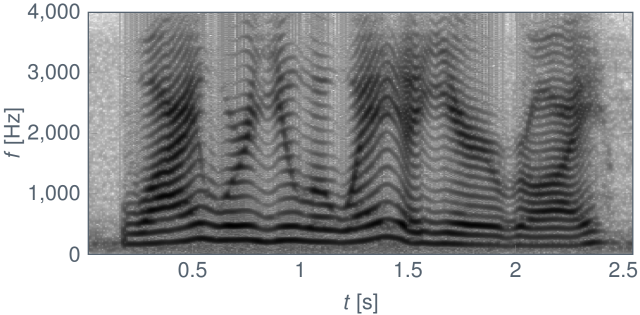
The amplitudes are constant

The number of harmonic components are constant

Is this model good enough?

The harmonic model

Example (Speech signal)



Hypothesis: For short enough segments, the harmonic model is an accurate representation of voiced speech.

The harmonic model

Matrix form of the harmonic model

The harmonic model can be rewritten as

$$\begin{aligned} s(n) &= \sum_{l=1}^{L} A_l \cos(l\omega_0 n + \phi_l) && (3) \\ &= \sum_{l=1}^{L} \left[\underbrace{A_l \cos(\phi_l)}_{=a_l} \cos(l\omega_0 n) - \underbrace{A_l \sin(\phi_l)}_{=b_l} \sin(l\omega_0 n) \right] && (4) \\ &= \sum_{l=1}^{L} \begin{bmatrix} \cos(l\omega_0 n) & \sin(l\omega_0 n) \end{bmatrix} \begin{bmatrix} a_l \\ -b_l \end{bmatrix} && (5) \end{aligned}$$

where

- $a_l = A_l \cos(\phi_l)$ and $b_l = A_l \sin(\phi_l)$ are linear parameters
- the fundamental frequency ω_0 is a nonlinear parameter

The harmonic model

Matrix form of the harmonic model

We have that

$$s(n) = \sum_{l=1}^{L} \begin{bmatrix} \cos(l\omega_0 n) & \sin(l\omega_0 n) \end{bmatrix} \begin{bmatrix} a_l \\ -b_l \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\omega_0 n) & \cdots & \cos(L\omega_0 n) & \sin(\omega_0 n) & \cdots & \sin(L\omega_0 n) \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_L \\ -b_1 \\ \vdots \\ -b_L \end{bmatrix}$$

The harmonic model

Matrix form of the harmonic model

For $n = 0, 1, \ldots, N-1$, this can be written as

$$\boldsymbol{s} = \boldsymbol{Z}_L(\omega_0)\boldsymbol{\alpha}_L \tag{6}$$

where

$$\boldsymbol{s} = \begin{bmatrix} s(0) & \cdots & s(N-1) \end{bmatrix}^T$$

$$\boldsymbol{Z}_L(\omega) = \begin{bmatrix} \boldsymbol{z}_\mathrm{c}(\omega) & \boldsymbol{z}_\mathrm{c}(2\omega) & \cdots & \boldsymbol{z}_\mathrm{c}(L\omega) & \boldsymbol{z}_\mathrm{s}(\omega) & \boldsymbol{z}_\mathrm{s}(2\omega) & \cdots & \boldsymbol{z}_\mathrm{s}(L\omega) \end{bmatrix}$$

$$\boldsymbol{z}_\mathrm{c}(\omega) = \begin{bmatrix} \cos(\omega 0) & \cdots & \cos(\omega(N-1)) \end{bmatrix}^T$$

$$\boldsymbol{z}_\mathrm{s}(\omega) = \begin{bmatrix} \sin(\omega 0) & \cdots & \sin(\omega(N-1)) \end{bmatrix}^T$$

$$\boldsymbol{\alpha}_L = \begin{bmatrix} \boldsymbol{a}_L^T & -\boldsymbol{b}_L^T \end{bmatrix}^T, \; \boldsymbol{a}_L = \begin{bmatrix} a_1 & \cdots & a_L \end{bmatrix}^T, \; \boldsymbol{b}_L = \begin{bmatrix} b_1 & \cdots & b_L \end{bmatrix}^T$$

The harmonic model

Matrix form of the harmonic model

We have now shown how the harmonic model can be written as

$$\boldsymbol{s} = \boldsymbol{Z}_L(\omega_0)\boldsymbol{\alpha}_L \ , \tag{7}$$

i.e., a matrix vector product. Some comments:

- The vector $\boldsymbol{\alpha}_L \in \mathbb{R}^{2L\times 1}$ contains $2L$ linear parameters
- The matrix $\boldsymbol{Z}_L(\omega_0) \in \mathbb{R}^{N\times 2L}$ depends on one nonlinear parameter
- We do not know the sizes of the matrix and vector since the number L of harmonic components is unknown.

The harmonic model

Summary - part I

- We can model any periodic signal using the harmonic model given by

$$s(n) = \sum_{l=1}^{L} A_l \cos(l\omega_0 n + \phi_l) \tag{8}$$

$$= \sum_{l=1}^{L} \left[\underbrace{A_l \cos(\phi_l)}_{=a_l} \cos(l\omega_0 n) - \underbrace{A_l \sin(\phi_l)}_{=b_l} \sin(l\omega_0 n) \right] \tag{9}$$

where

ω_0	fundamental frequency or pitch in radians/sample
L	number of harmonic components (or model order)
A_l	amplitude of l'th harmonic component
ϕ_l	phase of l'th harmonic component

The harmonic model

Summary - part II

- For $n = 0, 1, \ldots, N-1$, the harmonic model can be written as

$$\boldsymbol{s} = \boldsymbol{Z}_L(\omega_0)\boldsymbol{\alpha}_L \tag{10}$$

where $\boldsymbol{\alpha}_L$ is an unknown vector and $\boldsymbol{Z}_L(\omega_0)$ a matrix parametrised by the fundamental frequency ω_0.

Agenda

The harmonic model

Fundamental frequency estimation II

Fundamental frequency estimation III

Comparison of Methods

Fundamental frequency estimation II

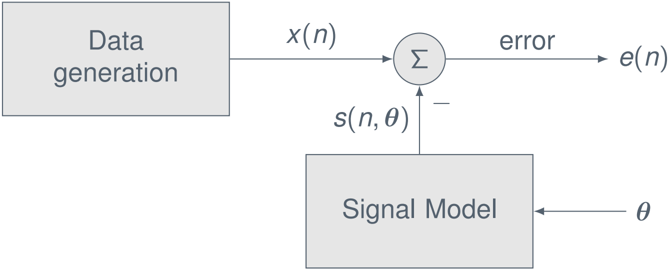
Motivation

In about 20 minutes, you will know

- how we estimate the fundamental frequency using nonlinear least-squares (NLS)

Fundamental frequency estimation II

The method of least-squares



- The vector θ contains the model parameters
- The signal $s(n, \theta)$ is produced by the signal model
- The signal $x(n)$ is the observed data
- The error consists of noise and model inaccuracies

Fundamental frequency estimation II

From the figure (on the previous slide), we have that

$$e(n) = x(n) - s(n, \theta) \ , \qquad n = 0, 1, \ldots, N-1 \tag{11}$$

where $s(n, \theta)$ is a periodic signal model given by

$$s(n, \theta) = \sum_{l=1}^{L} \Big[a_l \cos(l \omega_0 n) - b_l \sin(l \omega_0 n) \Big] \tag{12}$$

$$\theta = \begin{bmatrix} a_1 & \cdots & a_L & b_1 & \cdots & b_L & \omega_0 \end{bmatrix}^T \tag{13}$$

Fundamental frequency estimation II

The nonlinear least squares (NLS) method is that of solving

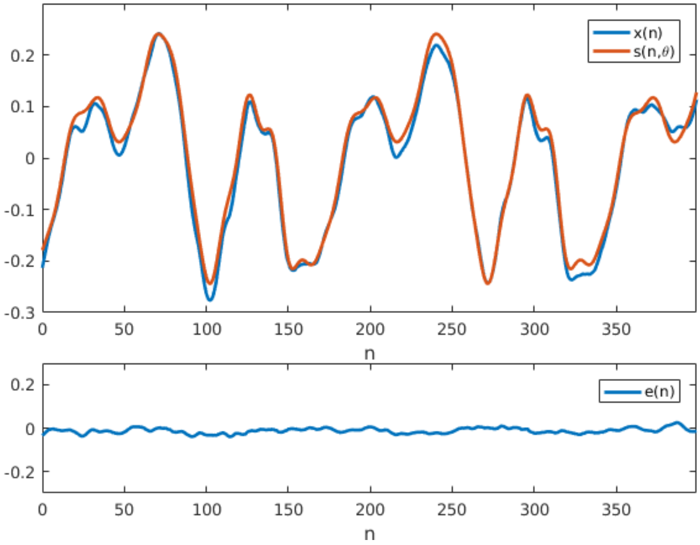
$$\hat{\boldsymbol{\theta}} = \operatorname*{argmin}_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \tag{14}$$

where $J(\boldsymbol{\theta})$ measures the squared error

$$J(\boldsymbol{\theta}) = \sum_{n=0}^{N-1} |e(n)|^2 = \sum_{n=0}^{N-1} |x(n) - s(n, \boldsymbol{\theta})|^2 \tag{15}$$

- Solving this problem naïvely is very computationally demanding since the fundamental frequency is a nonlinear parameter.
- Asymptotically, however, an efficient solution exists which for historical reasons is called harmonic summation (Noll, 1969).

Fundamental frequency estimation II



Fundamental frequency estimation II

We can model a periodic signal $s(n)$ observed in noise $e(n)$ as

$$x(n) = \sum_{l=1}^{L} \Big[a_l \cos(l\omega_0 n) - b_l \sin(l\omega_0 n) \Big] + e(n) \tag{16}$$

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

$$\boldsymbol{x} = \boldsymbol{Z}_L(\omega_0)\boldsymbol{\alpha}_L + \boldsymbol{e} \tag{17}$$

where

$$\boldsymbol{Z}_L(\omega) = \begin{bmatrix} \boldsymbol{z}_\mathrm{c}(\omega) & \boldsymbol{z}_\mathrm{c}(2\omega) & \cdots & \boldsymbol{z}_\mathrm{c}(L\omega) & \boldsymbol{z}_\mathrm{s}(\omega) & \boldsymbol{z}_\mathrm{s}(2\omega) & \cdots & \boldsymbol{z}_\mathrm{s}(L\omega) \end{bmatrix}$$

$$\boldsymbol{z}_\mathrm{c}(\omega) = \begin{bmatrix} \cos(\omega 0) & \cdots & \cos(\omega(N-1)) \end{bmatrix}^T$$

$$\boldsymbol{z}_\mathrm{s}(\omega) = \begin{bmatrix} \sin(\omega 0) & \cdots & \sin(\omega(N-1)) \end{bmatrix}^T$$

$$\boldsymbol{\alpha}_L = \begin{bmatrix} \boldsymbol{a}_L^T & -\boldsymbol{b}_L^T \end{bmatrix}^T, \; \boldsymbol{a}_L = \begin{bmatrix} a_1 & \cdots & a_L \end{bmatrix}^T, \; \boldsymbol{b}_L = \begin{bmatrix} b_1 & \cdots & b_L \end{bmatrix}^T$$

Fundamental frequency estimation II

The least squares error is

$$\sum_{n=0}^{N-1} e^2(n) = \boldsymbol{e}^T\boldsymbol{e} = [\boldsymbol{x} - \boldsymbol{Z}_L(\omega_0)\boldsymbol{\alpha}_L]^T [\boldsymbol{x} - \boldsymbol{Z}_L(\omega_0)\boldsymbol{\alpha}_L] \qquad (18)$$

Fundamental frequency estimation II

The least squares error is

$$\sum_{n=0}^{N-1} e^2(n) = \boldsymbol{e}^T \boldsymbol{e} = [\boldsymbol{x} - \boldsymbol{Z}_L(\omega_0)\boldsymbol{\alpha}_L]^T [\boldsymbol{x} - \boldsymbol{Z}_L(\omega_0)\boldsymbol{\alpha}_L] \tag{18}$$

Conditioned on ω_0, the estimate of $\boldsymbol{\alpha}_L$ is the linear LS estimate, i.e.,

$$\hat{\boldsymbol{\alpha}}_L(\omega_0) = \left[\boldsymbol{Z}_L^T(\omega_0)\boldsymbol{Z}_L(\omega_0)\right]^{-1} \boldsymbol{Z}_L^T(\omega_0)\boldsymbol{x} \,. \tag{19}$$

Fundamental frequency estimation II

The least squares error is

$$\sum_{n=0}^{N-1} e^2(n) = \boldsymbol{e}^T\boldsymbol{e} = [\boldsymbol{x} - \boldsymbol{Z}_L(\omega_0)\boldsymbol{\alpha}_L]^T [\boldsymbol{x} - \boldsymbol{Z}_L(\omega_0)\boldsymbol{\alpha}_L] \tag{18}$$

Conditioned on ω_0, the estimate of $\boldsymbol{\alpha}_L$ is the linear LS estimate, i.e.,

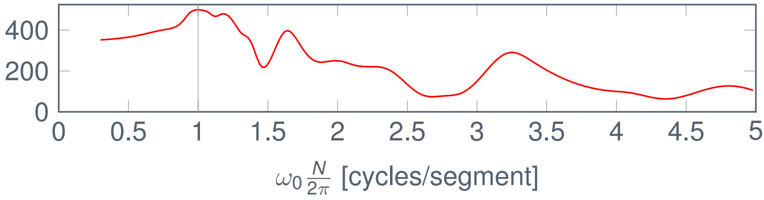
$$\hat{\boldsymbol{\alpha}}_L(\omega_0) = \left[\boldsymbol{Z}_L^T(\omega_0)\boldsymbol{Z}_L(\omega_0)\right]^{-1} \boldsymbol{Z}_L^T(\omega_0)\boldsymbol{x}\ . \tag{19}$$

Inserting this back into the objective yields the NLS estimator

$$\hat{\omega}_{0,L} = \underset{\omega_0 \in [\omega_{\text{MIN}}, \omega_{\text{MAX}}]}{\operatorname{argmax}}\ \boldsymbol{x}^T\boldsymbol{Z}_L(\omega_0)\left[\boldsymbol{Z}_L^T(\omega_0)\boldsymbol{Z}_L(\omega_0)\right]^{-1} \boldsymbol{Z}_L^T(\omega_0)\boldsymbol{x} \tag{20}$$

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

Fundamental frequency estimation II

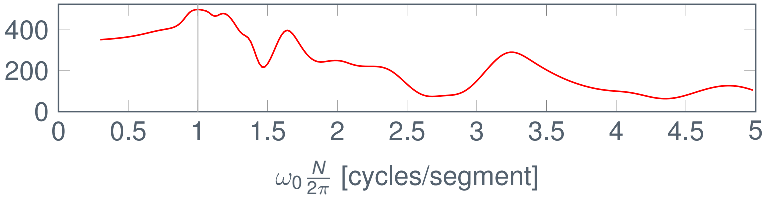


1. Compute NLS cost function

$$\hat{\omega}_{0,L} = \underset{\omega_0 \in [\omega_{\mathrm{MIN}}, \omega_{\mathrm{MAX}}]}{\operatorname{argmax}} \; \boldsymbol{x}^T \boldsymbol{Z}_L(\omega_0) \left[\boldsymbol{Z}_L^T(\omega_0) \boldsymbol{Z}_L(\omega_0) \right]^{-1} \boldsymbol{Z}_L^T(\omega_0) \boldsymbol{x} \quad (21)$$

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

Fundamental frequency estimation II

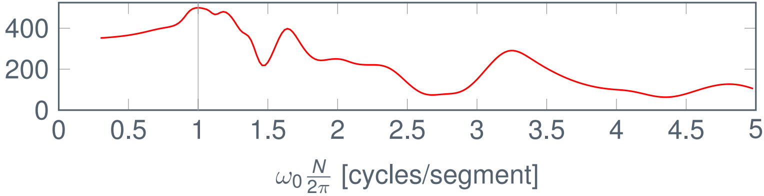


1. Compute NLS cost function

$$\hat{\omega}_{0,L} = \underset{\omega_0 \in [\omega_{\mathrm{MIN}}, \omega_{\mathrm{MAX}}]}{\operatorname{argmax}} \boldsymbol{x}^T \boldsymbol{Z}_L(\omega_0) \left[\boldsymbol{Z}_L^T(\omega_0) \boldsymbol{Z}_L(\omega_0) \right]^{-1} \boldsymbol{Z}_L^T(\omega_0) \boldsymbol{x} \quad (21)$$

   on an F/L-point uniform grid for all model orders $L \in \{1, \dots, L_{\mathrm{MAX}}\}$.
2. Optionally refine the L_{MAX} grid estimates.

Fundamental frequency estimation II



1. Compute NLS cost function

$$\hat{\omega}_{0,L} = \underset{\omega_0 \in [\omega_{\mathrm{MIN}}, \omega_{\mathrm{MAX}}]}{\operatorname{argmax}} \; \boldsymbol{x}^T \boldsymbol{Z}_L(\omega_0) \left[\boldsymbol{Z}_L^T(\omega_0) \boldsymbol{Z}_L(\omega_0) \right]^{-1} \boldsymbol{Z}_L^T(\omega_0) \boldsymbol{x} \quad (21)$$

   on an F/L-point uniform grid for all model orders $L \in \{1, \dots, L_{\mathrm{MAX}}\}$.
2. Optionally refine the L_{MAX} grid estimates.
3. Do model comparison.

Fundamental frequency estimation II

Fast NLS Algorithm

The most costly step is the first one (on the previous slide). Specifically, evaluating the NLS estimator

$$\hat{\omega}_{0,L} = \underset{\omega_0 \in [\omega_{\text{MIN}}, \omega_{\text{MAX}}]}{\operatorname{argmax}} \boldsymbol{x}^T \boldsymbol{Z}_L(\omega_0) \left[\boldsymbol{Z}_L^T(\omega_0) \boldsymbol{Z}_L(\omega_0) \right]^{-1} \boldsymbol{Z}_L^T(\omega_0) \boldsymbol{x} \tag{22}$$

over a F/L-point uniform grid for all model orders $L \in \{1, \dots, L_{\text{MAX}}\}$ will cost you $\mathcal{O}(F \log F) + \mathcal{O}(FL_{\text{MAX}}^3)$ floating point operations (flops).

Fundamental frequency estimation II
Fast NLS Algorithm

The most costly step is the first one (on the previous slide). Specifically, evaluating the NLS estimator

$$\hat{\omega}_{0,L} = \underset{\omega_0 \in [\omega_{\text{MIN}}, \omega_{\text{MAX}}]}{\operatorname{argmax}} \boldsymbol{x}^T \boldsymbol{Z}_L(\omega_0) \left[\boldsymbol{Z}_L^T(\omega_0)\boldsymbol{Z}_L(\omega_0)\right]^{-1} \boldsymbol{Z}_L^T(\omega_0)\boldsymbol{x} \tag{22}$$

over a F/L-point uniform grid for all model orders $L \in \{1, \ldots, L_{\text{MAX}}\}$ will cost you $\mathcal{O}(F \log F) + \mathcal{O}(FL_{\text{MAX}}^3)$ floating point operations (flops).

Fast NLS

Recently in (Nielsen et al., 2017), we have reduced the complexity to just $\mathcal{O}(F \log F) + \mathcal{O}(FL_{\text{MAX}})$. The fast algorithm is divided in two:

Data-independent step is run once when the estimator is initialised

Data-dependent step is run for every new segment of data

Fundamental frequency estimation II
Fast NLS Algorithm

A MATLAB implementation of the NLS estimator

```
% create an estimator object (the data independent step is computed)
f0Estimator = fastF0Nls(nData, maxNoHarmonics, f0Bounds);
% analyse a segment of data
[f0Estimate, estimatedNoHarmonics, estimatedLinParam] = ...
    f0Estimator.estimate(data);
```

- The algorithm also includes model comparison.
- The algorithm can also be set-up to work for a model with a non-zero DC-value.
- A C++-implementation is also available (although not as refined as the MATLAB implementation).
- Can be downloaded from `https://github.com/jkjaer/fastF0Nls`.

Fundamental frequency estimation II
Fast NLS Algorithm

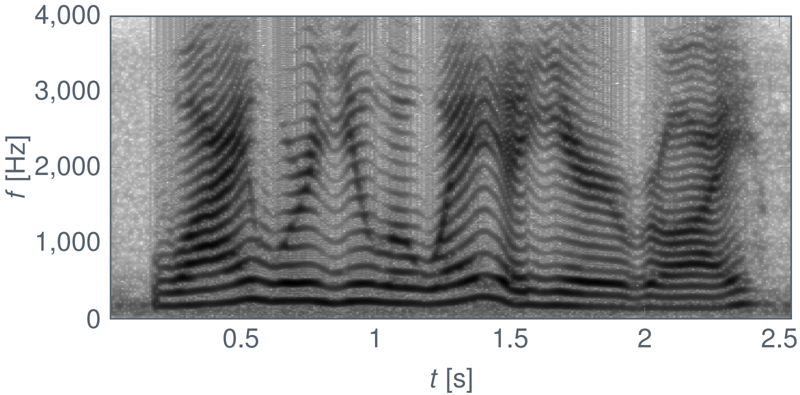
A MATLAB implementation (example)

```
% load the mono speech signal
[speechSignal, samplingFreq] = audioread('roy.wav');
nData = length(speechSignal);

% set up
segmentTime = 0.025; % seconds
segmentLength = round(segmentTime*samplingFreq); % samples
nSegments = floor(nData/segmentLength);
f0Bounds = [80, 400]/samplingFreq; % cycles/sample
maxNoHarmonics = 15;
f0Estimator = fastF0Nls(segmentLength, maxNoHarmonics, f0Bounds);

% do the analysis
idx = 1:segmentLength;
f0Estimates = nan(1,nSegments); % cycles/sample
for ii = 1:nSegments
    speechSegment = speechSignal(idx);
    f0Estimates(ii) = f0Estimator.estimate(speechSegment);
    idx = idx + segmentLength;
end
```

Fundamental frequency estimation II

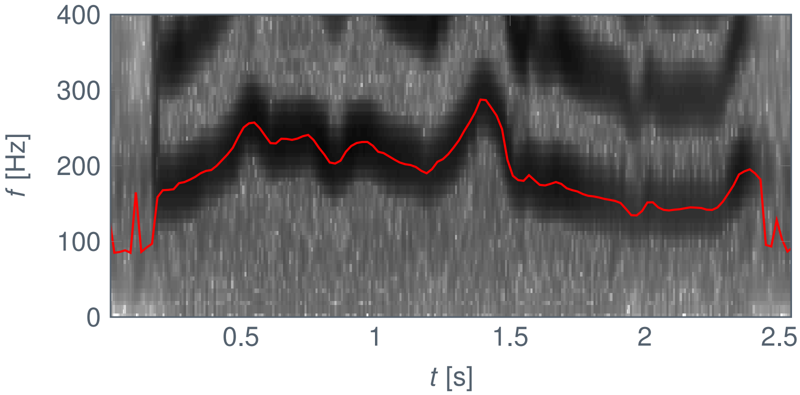
Fast NLS Algorithm

Example (Speech signal)



Fundamental frequency estimation II
Fast NLS Algorithm

Example (Speech signal)



Fundamental frequency estimation II

Summary

- Nonlinear least-squares (NLS) estimator is given by

$$\hat{\omega}_{0,L} = \underset{\omega_0 \in [\omega_{\text{MIN}}, \omega_{\text{MAX}}]}{\operatorname{argmax}} \boldsymbol{x}^T \boldsymbol{Z}_L(\omega_0) \left[\boldsymbol{Z}_L^T(\omega_0) \boldsymbol{Z}_L(\omega_0) \right]^{-1} \boldsymbol{Z}_L^T(\omega_0) \boldsymbol{x} \,. \quad (23)$$

- Until recently, the NLS estimator was extremely slow, which made it impractical.
- The NLS estimator is one of the best pitch estimators if the noise is (approximately) white (you will see some evidence later).

Agenda

Fundamental frequency estimation III

Motivation

In about 20 minutes, you will know

- what the harmonic summation (HS) is
- why HS is an approximate NLS estimator
- when the approximation is good and when it is not

Fundamental frequency estimation III

Recall that the least squares error is

$$e(n) = x(n) - s(n, \theta) , \qquad n = 0, 1, \ldots , N - 1 \tag{24}$$

where $s(n, \theta)$ is a periodic signal model given by

$$s(n, \theta) = \sum_{l=1}^{L} A_l \cos(l\omega_0 n + \phi_l) . \tag{25}$$

We wish to find the parameter vector θ which minimises

$$J(\theta) = \sum_{n=0}^{N-1} |e(n)|^2 = \sum_{n=0}^{N-1} |x(n) - s(n, \theta)|^2 . \tag{26}$$

Fundamental frequency estimation III

From Parseval's theorem, we have that

$$\lim_{N\to\infty} \sum_{n=0}^{N-1} |e(n)|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |E(\omega)|^2 d\omega \tag{27}$$

where

$$E(\omega) = X(\omega) - S(\omega) \tag{28}$$

$$S(\omega) = 2\pi \sum_{l=1}^{L} [c_l \delta(\omega - \omega_0 l) + c_l^* \delta(\omega + \omega_0 l)] \tag{29}$$

$$c_l = A_l \exp(j\phi_l)/2\ . \tag{30}$$

Note that $S(\omega)$ is simply the DTFT of a periodic signal.

Fundamental frequency estimation III

Let us now minimise

$$\lim_{N\to\infty} \sum_{n=0}^{N-1} |e(n)|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |E(\omega)|^2 d\omega \tag{31}$$

instead of

$$J(\theta) = \sum_{n=0}^{N-1} |e(n)|^2 \, . \tag{32}$$

Although not as obvious, this can also be performed using least squares.

Fundamental frequency estimation III

Given ω_0, the optimal value for c_l is

$$\hat{c}_l = \frac{1}{2\pi} X(\omega_0 l) \tag{33}$$

Inserting this into the error $E(\omega) = X(\omega) - S(\omega)$ yields the objective

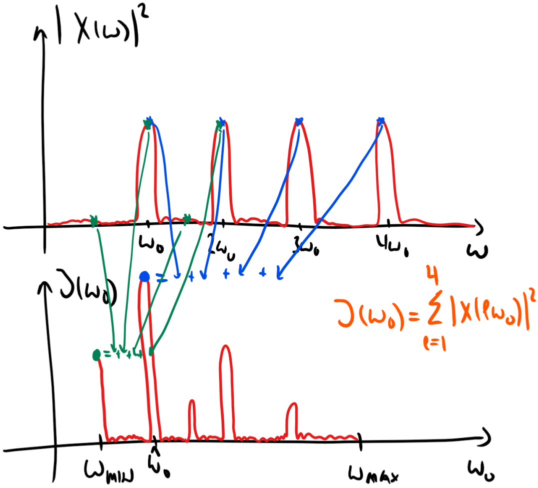
$$G(\omega_0) = \frac{1}{2\pi} \int_{-\pi}^{\pi} |X(\omega)|^2 d\omega - \frac{2}{2\pi} \sum_{l=1}^{L} |X(\omega_0 l)|^2 \, . \tag{34}$$

The harmonic summation (HS) estimator is

$$\hat{\omega}_0 = \underset{\omega_0 \in [\omega_{\text{MIN}}, \omega_{\text{MAX}}]}{\operatorname{argmax}} \sum_{l=1}^{L} |X(\omega_0 l)|^2 \tag{35}$$

Thus, the objective function is simply a sum of evenly spaced values from the squared amplitude spectrum of the observed signal!

Fundamental frequency estimation III



Fundamental frequency estimation III

HS algorithm for a model order L:

1. Compute the DFT of observed signal $x(n)$ (rule-of-thumb: zero-pad to a length of $5NL$)
2. For a candidate pitch ω_0 in $[\omega_{\text{MIN}}, \omega_{\text{MAX}}]$, extract the corresponding spectral values from the DFT of $x(n)$, i.e., $X(\omega_0 l)$ for $l = 1, \ldots, L$
3. Square and sum the L extracted spectral values.
4. Go to 2., until the objective has been evaluated for all candidate pitches.
5. Find the pitch which maximises the objective.

Fundamental frequency estimation III

Alternative derivation of the HS estimator

Asymptotically,

$$\lim_{N\to\infty} \frac{2}{N} \boldsymbol{Z}_L^T(\omega_0)\boldsymbol{Z}_L(\omega_0) = \boldsymbol{I}_L \, . \tag{36}$$

Using this limit as an approximation gives the harmonic summation estimator (Noll, 1969)

$$\hat{\omega}_{0,L} = \underset{\omega_0\in[\omega_{\text{MIN}},\omega_{\text{MAX}}]}{\operatorname{argmax}} \; \boldsymbol{x}^T\boldsymbol{Z}_L(\omega_0)\boldsymbol{Z}_L^T(\omega_0)\boldsymbol{x} = \underset{\omega_0\in[\omega_{\text{MIN}},\omega_{\text{MAX}}]}{\operatorname{argmax}} \sum_{l=1}^{L} |X(\omega_0 l)|^2$$

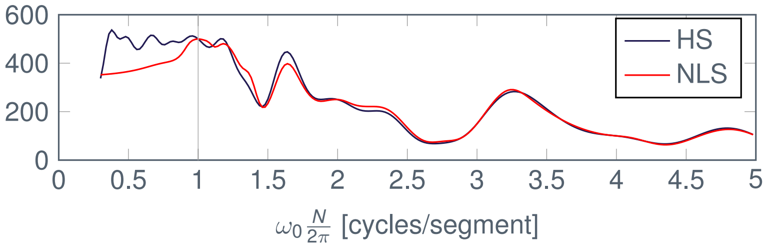
The HS estimator is also referred to as approximate NLS (aNLS).

Fundamental frequency estimation III

NLS vs. HS

Some remarks:

- The HS method works very well, unless the fundamental frequency is low or the maximum harmonic component is close to the Nyquist frequency.
- The HS method can be implemented very efficiently using a single FFT.
- The order of complexity of HS and fast NLS are the same, but HS has a smaller scale factor (typically 6-8 times faster).



Fundamental frequency estimation III

Summary

- The harmonic summation (HS) estimator is given by

$$\hat{\omega}_0 = \underset{\omega_0 \in [\omega_{\text{MIN}}, \omega_{\text{MAX}}]}{\text{argmax}} \sum_{l=1}^{L} |X(\omega_0 l)|^2 \,. \tag{37}$$

- The HS estimator is also referred to as approximate NLS (aNLS).
- The HS estimator has the same estimation accuracy as the exact NLS, unless the number of pitch periods in the observed signal is low (less that 2 pitch periods). Thus exact NLS has a better time-frequency resolution.
- The HS estimator is around 6-8 times faster than fast NLS.

Fundamental frequency estimation III

5 minutes active break

The harmonic summation estimator is given by

$$\hat{\omega}_0 = \underset{\omega_0 \in [\omega_{\text{MIN}}, \omega_{\text{MAX}}]}{\operatorname{argmax}} \sum_{l=1}^{L} |X(\omega_0 l)|^2 \,. \tag{38}$$

A related method is the harmonic product spectrum (HPS) method given by

$$\hat{\omega}_0 = \underset{\omega_0 \in [\omega_{\text{MIN}}, \omega_{\text{MAX}}]}{\operatorname{argmax}} \prod_{l=1}^{L} |X(\omega_0 l)|^2 \,. \tag{39}$$

Think about/discuss with your neighbour

- pros and cons of HS and HPS
- what are good properties of a pitch estimator?

Outline

Comparison of Methods

Motivation

In about 20 minutes, you will know

- what properties are (normally) important for a pitch estimator
- how some pitch estimators perform on data in terms of
  1. estimation accuracy
  2. robustness to noise
  3. time-frequency resolution
- pros and cons of various pitch estimators

Comparison of Methods

What could be evaluated?

1. Estimation accuracy
2. Robustness to noise
3. Time-frequency resolution
4. Computational complexity

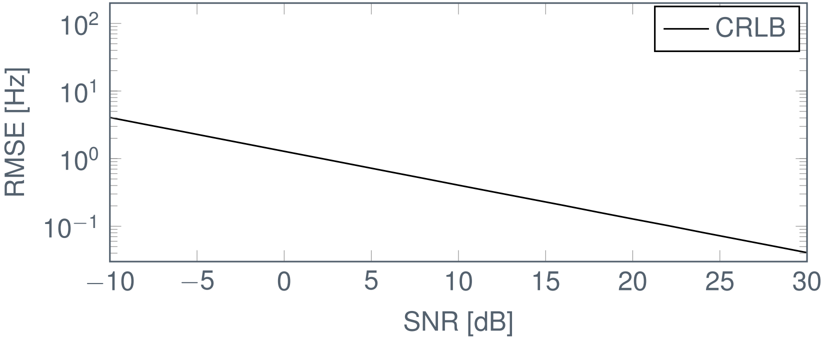
Comparison of Methods
Robustness to noise

Simulation setup

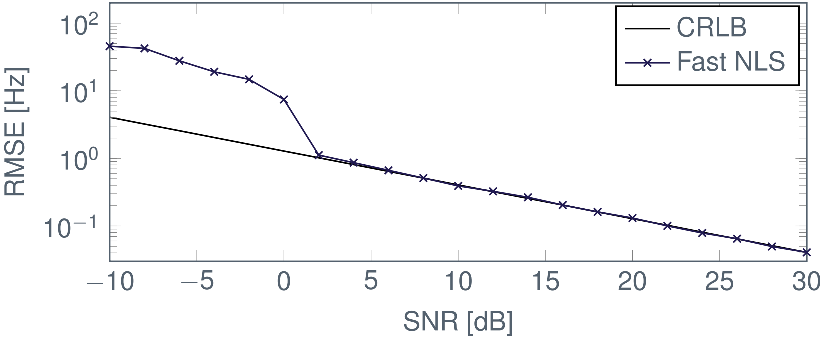
- Segment size of 25 ms at a sampling frequency of 8000 Hz.
- Estimate the pitch from 1000 Monte Carlo runs for every SNR.
- In each run, the true pitch is randomly selected from $[90, 380]$ Hz and the true phases are also generated at random.
- The true amplitudes are exponentially decreasing.
- The true model order is 7.
- Each method searches for a pitch in the range $[80, 400]$ Hz.
- The maximum model order in NLS is set to 15.
- The noise is white and Gaussian.
- No pitch tracking used in any method.

Comparison of Methods

Robustness to noise

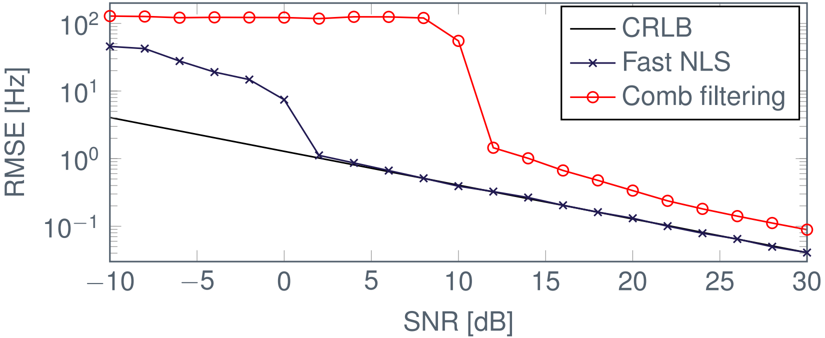


Comparison of Methods
Robustness to noise



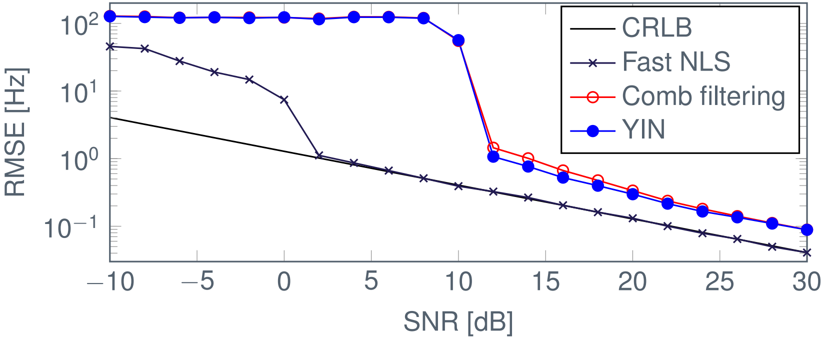
Comparison of Methods

Robustness to noise

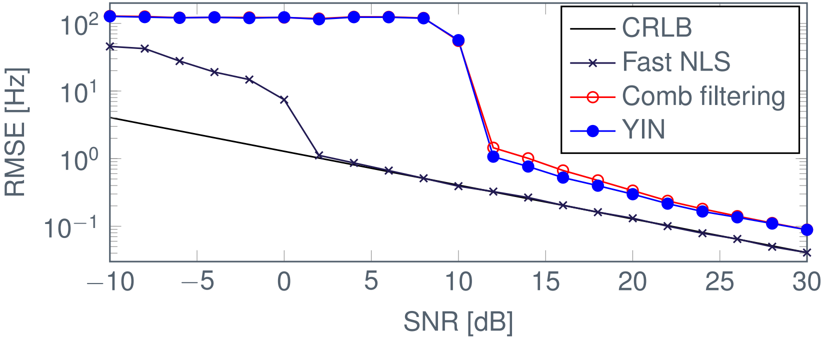


Comparison of Methods

Robustness to noise



Comparison of Methods
Robustness to noise

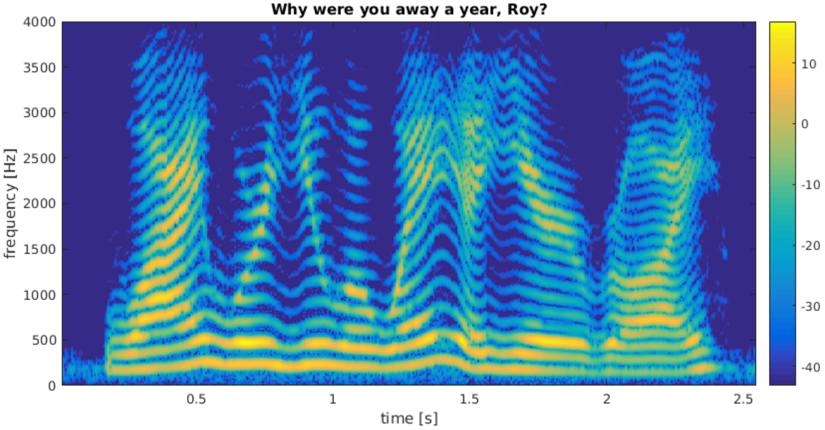


Average computation times in MATLAB

Fast NLS: 7.6 ms, Comb filter: 2.4 ms, YIN: 0.7 ms

Comparison of Methods

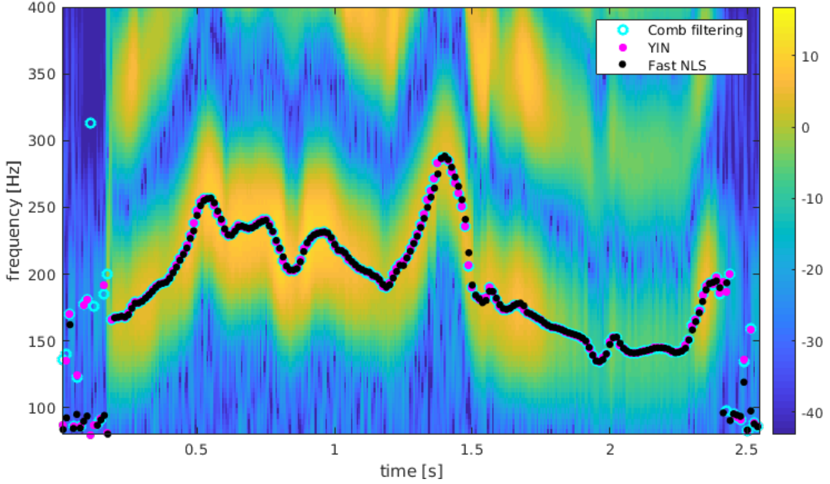
Robustness to noise



Comparison of Methods

Robustness to noise

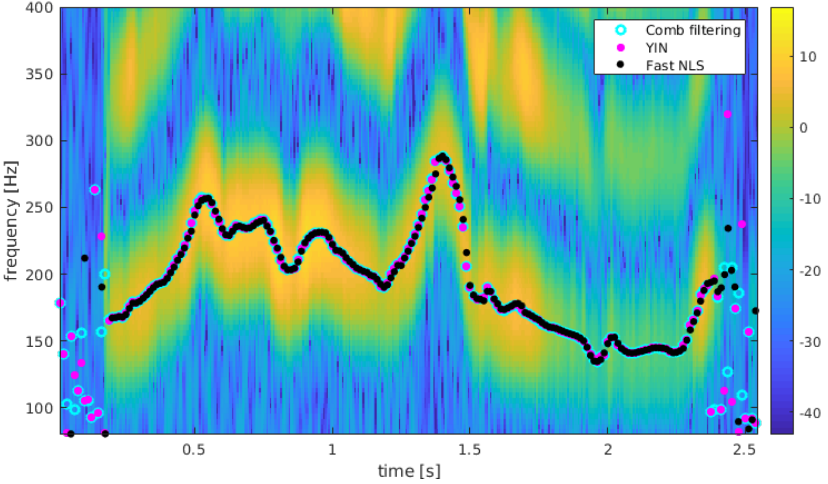
No noise and window size of 25 ms.



Comparison of Methods

Robustness to noise

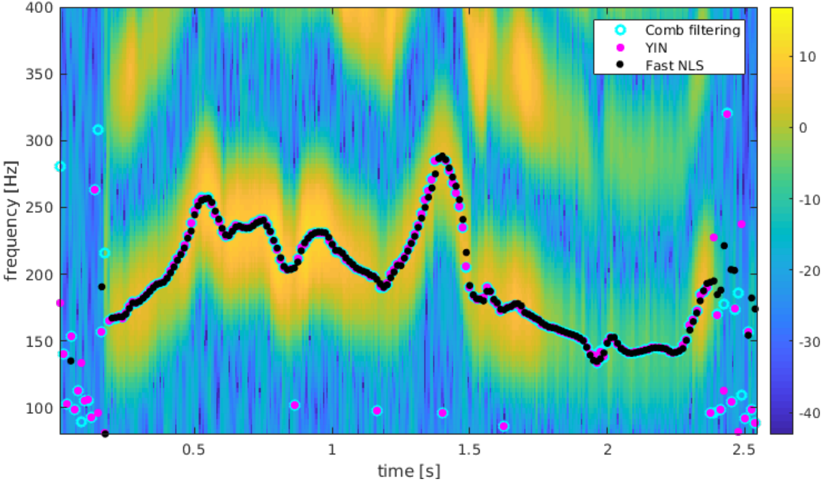
20 dB SNR and window size of 25 ms.



Comparison of Methods

Robustness to noise

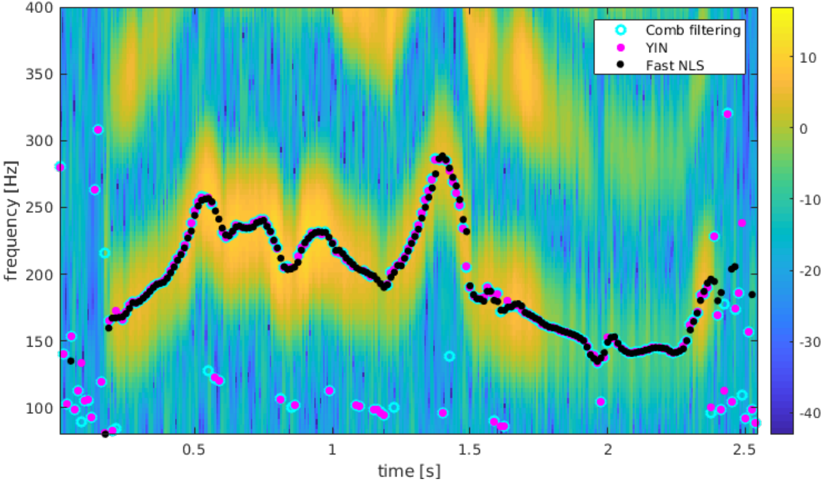
15 dB SNR and window size of 25 ms.



Comparison of Methods

Robustness to noise

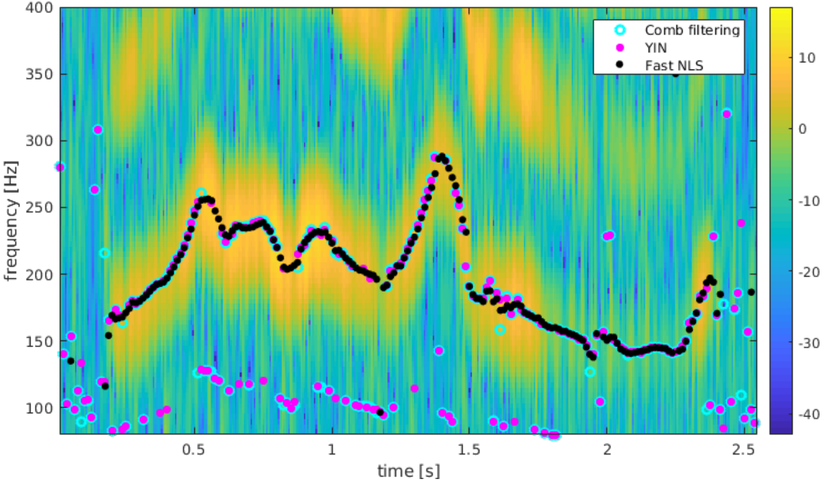
10 dB SNR and window size of 25 ms.



Comparison of Methods

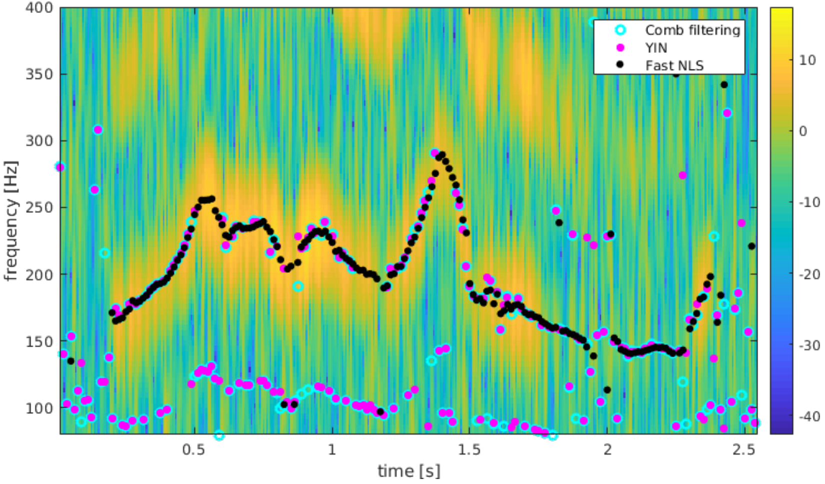
Robustness to noise

5 dB SNR and window size of 25 ms.



Comparison of Methods
Robustness to noise

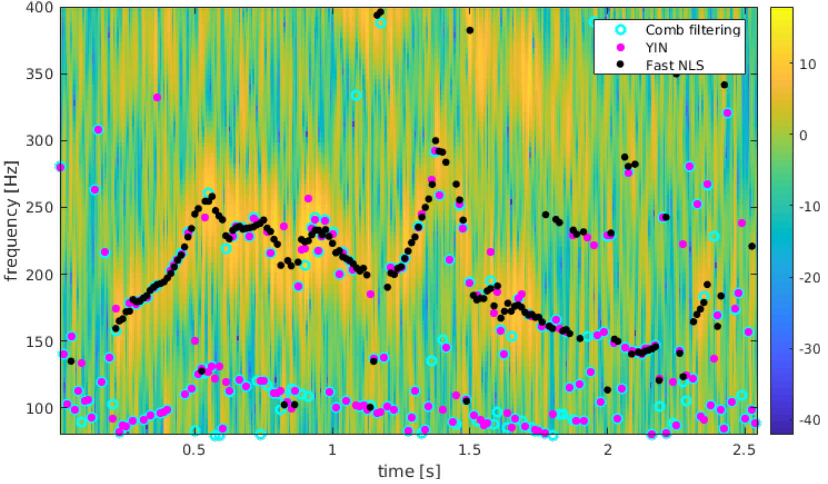
0 dB SNR and window size of 25 ms.



Comparison of Methods

Robustness to noise

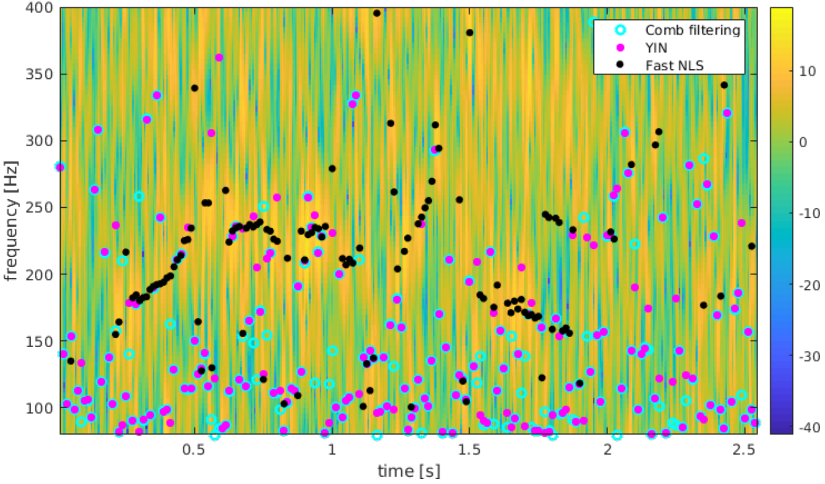
-5 dB SNR and window size of 25 ms.



Comparison of Methods

Robustness to noise

-10 dB SNR and window size of 25 ms.



Comparison of Methods

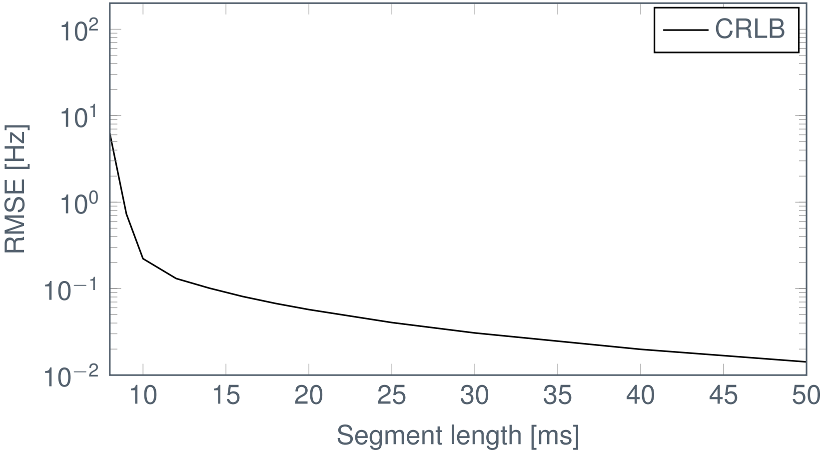
Time-frequency resolution

Simulation setup

- SNR of 30 dB at a sampling frequency of 8000 Hz.
- Estimate the pitch from 1000 Monte Carlo runs for every segment time.
- In each run, the true pitch is randomly selected from $[90, 380]$ Hz and the true phases are also generated at random.
- The true amplitudes are exponentially decreasing.
- The true model order is 7.
- Each method searches for a pitch in the range $[80, 400]$ Hz.
- The maximum model order in NLS is set to 15.
- The noise is white and Gaussian.
- No pitch tracking used in any method.

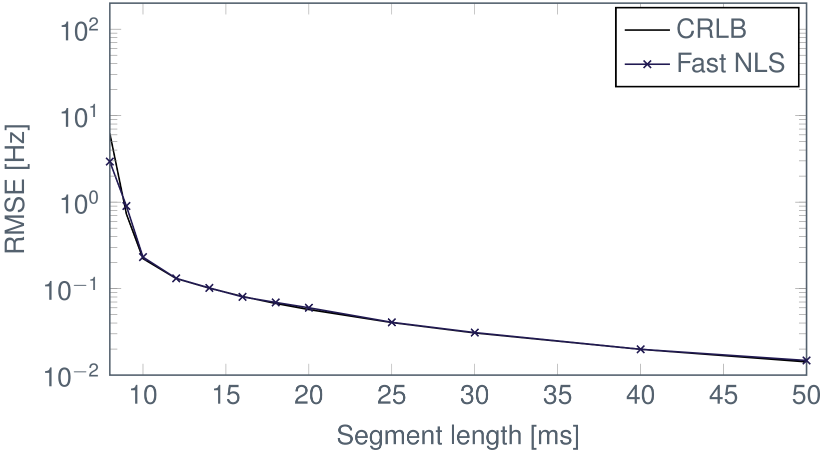
Comparison of Methods

Time-frequency resolution

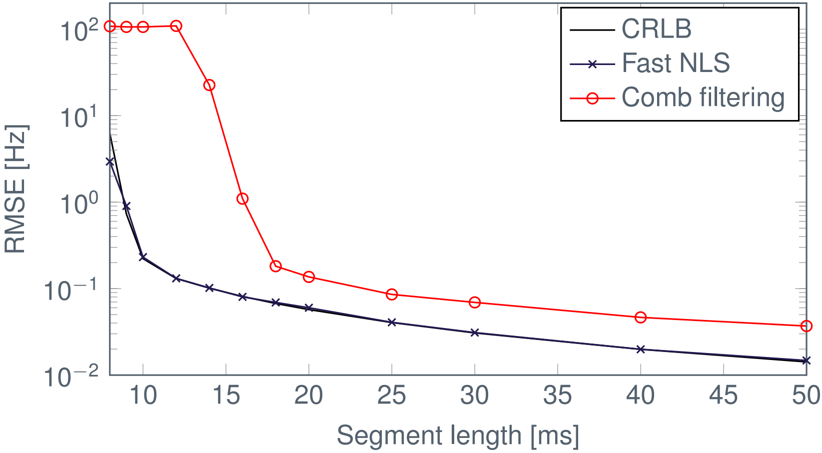


Comparison of Methods

Time-frequency resolution

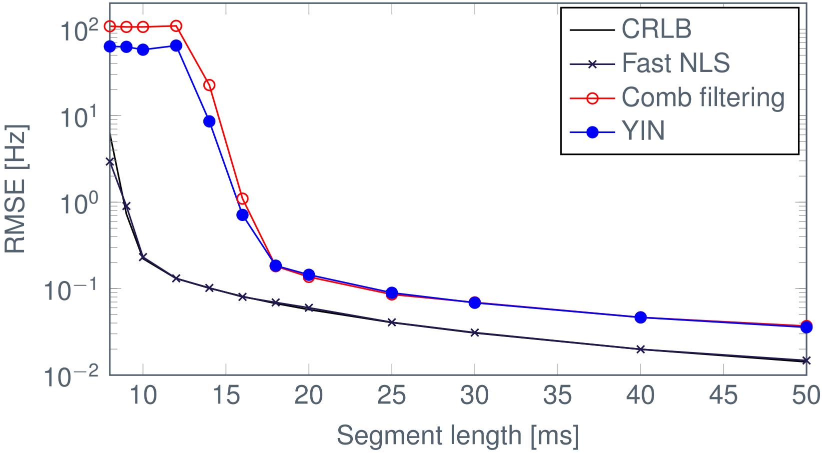


Comparison of Methods
Time-frequency resolution



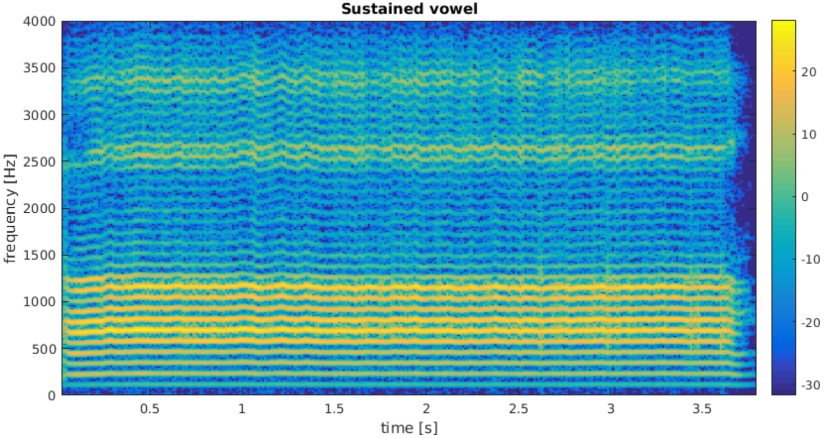
Comparison of Methods

Time-frequency resolution



Comparison of Methods

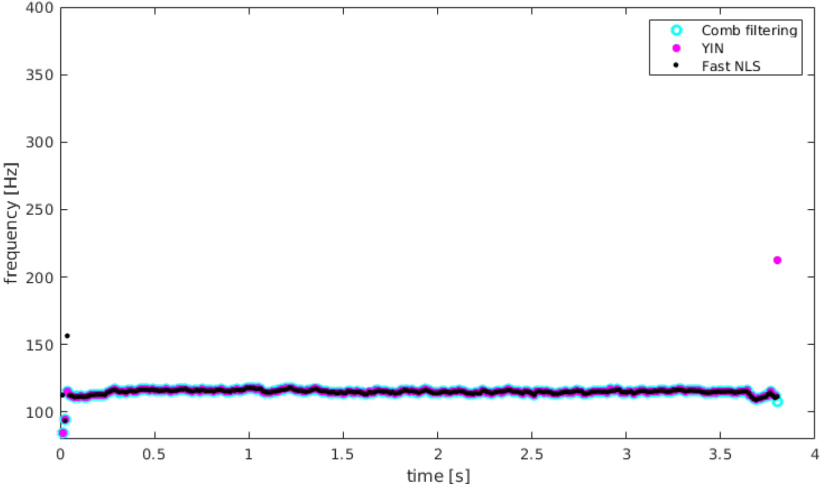
Time-frequency resolution



Comparison of Methods

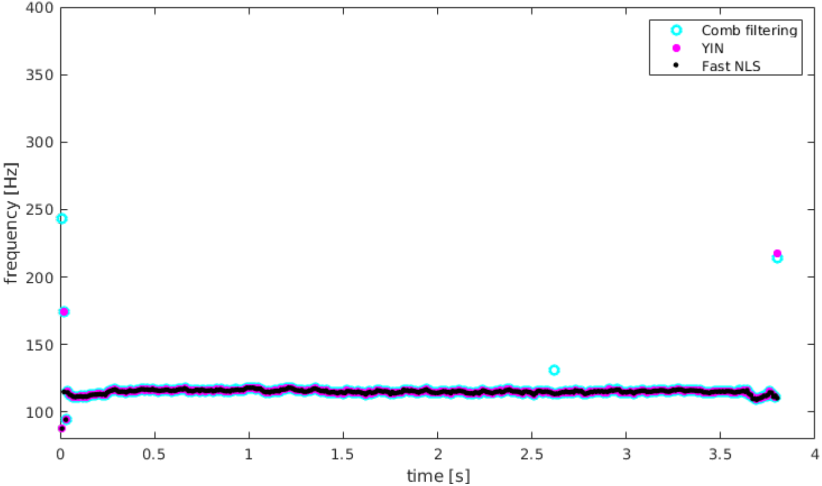
Time-frequency resolution

Window size of 25 ms and no noise.



Comparison of Methods
Time-frequency resolution

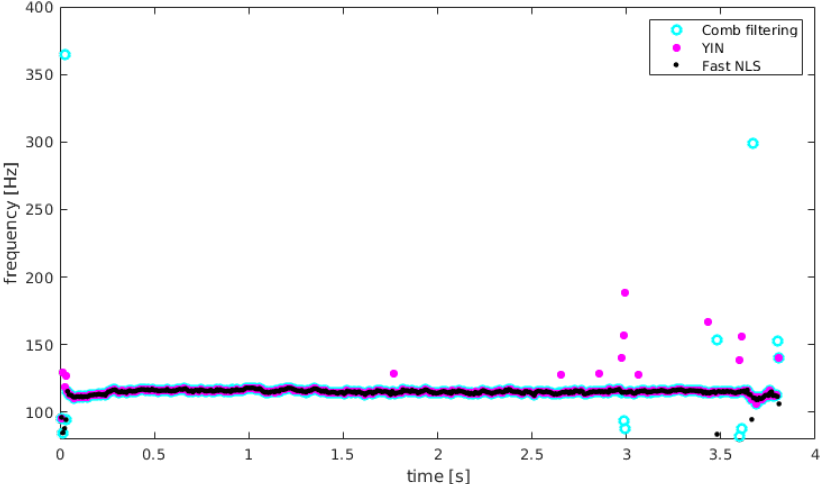
Window size of 20 ms and no noise.



Comparison of Methods

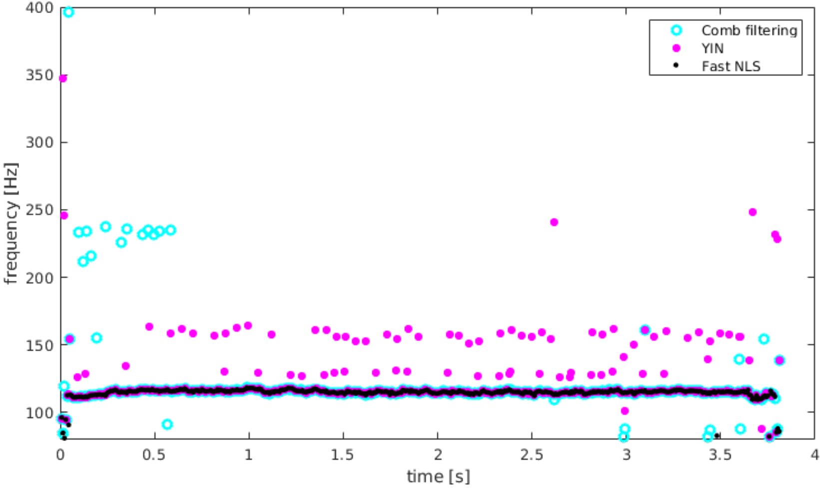
Time-frequency resolution

Window size of 16 ms and no noise.



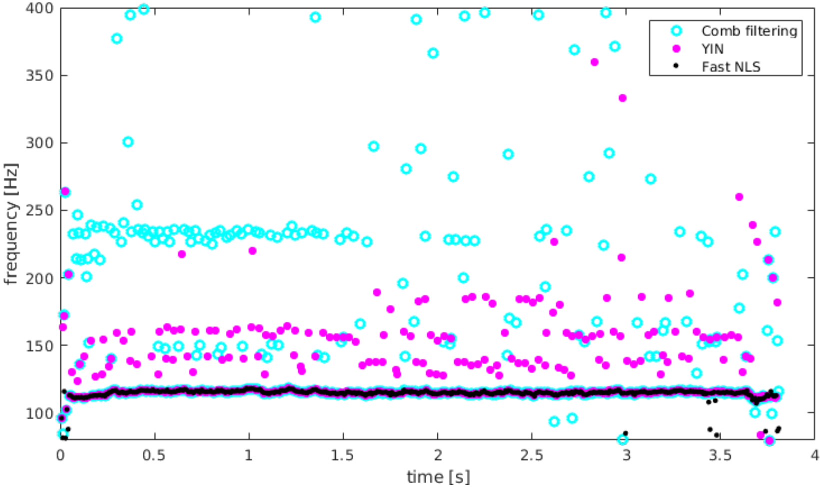
Comparison of Methods
Time-frequency resolution

Window size of 15 ms and no noise.



Comparison of Methods
Time-frequency resolution

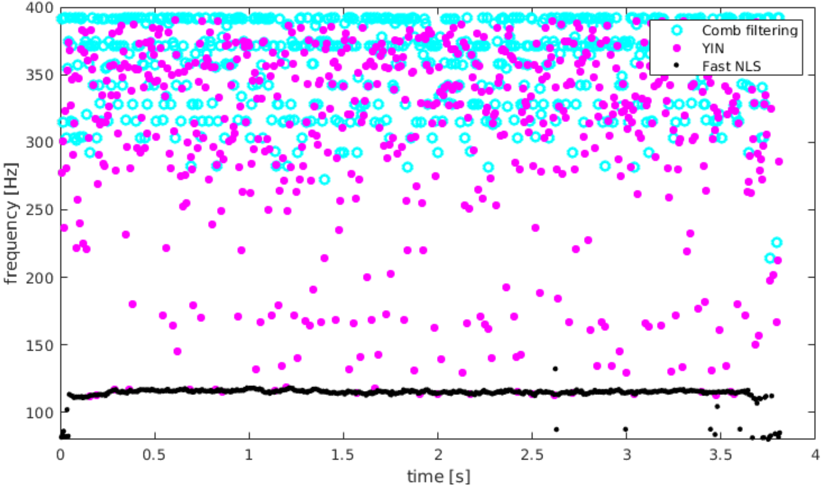
Window size of 14 ms and no noise.



Comparison of Methods

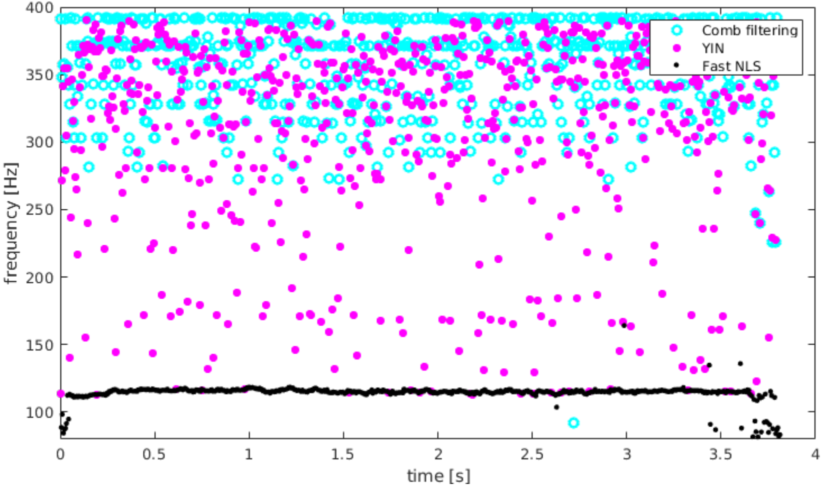
Time-frequency resolution

Window size of 12 ms and no noise.



Comparison of Methods
Time-frequency resolution

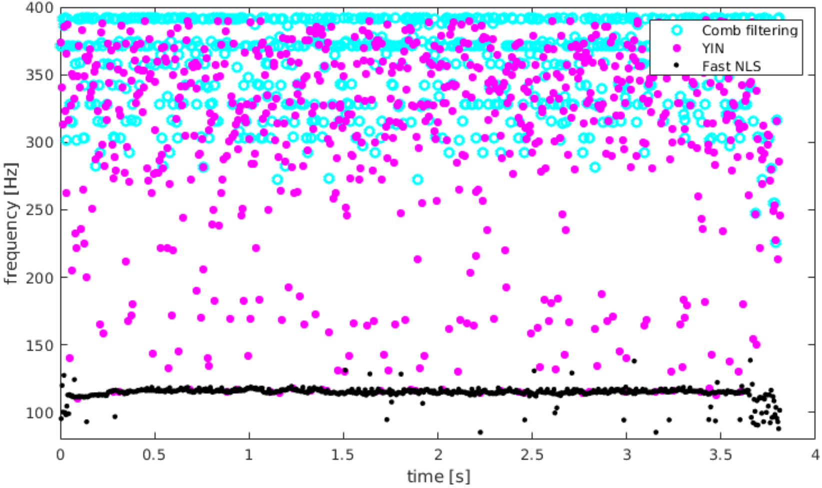
Window size of 11 ms and no noise.



Comparison of Methods

Time-frequency resolution

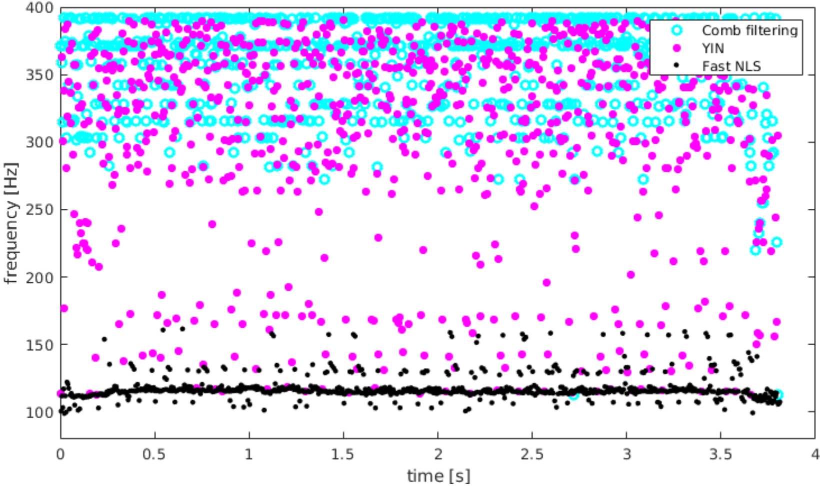
Window size of 10 ms and no noise.



Comparison of Methods

Time-frequency resolution

Window size of 9 ms and no noise.



Comparison of Methods

Summary - part I

Correlation-based methods are based on

$$x(n) = x(n - \tau) \tag{40}$$

where $\tau = 2\pi/\omega_0$ is the period.

- \+ Intuitive and simple
- \+ Low computational complexity
- \+ Mature and refined set of methods
- +/- No need to estimate the model order
- \- Interpolation needed for fractional delay estimation
- \- Poor time-frequency resolution
- \- Are sensitive to noise

Comparison of Methods

Summary - part II

Model-based methods (such as NLS and HS) are based on

$$x(n) = \sum_{l=1}^{L} A_l \cos(l\omega_0 n + \phi_l) + e(n) \tag{41}$$

- \+ High estimation accuracy
- \+ Work very well in even noisy conditions
- \+ Good time-frequency resolution
- +/- The model order has to be estimated
- \- Higher computational complexity
- \- Early stage methods without fine tuning (yet)

Questions?

AALBORG UNIVERSITY
DENMARK