POLITECNICO DI MILANO





Frequency response function estimation

Marco Lovera

Dipartimento di Scienze e Tecnologie Aerospaziali, Politecnico di Milano

Outline

- Overview of the FRF estimation process
- Mean removal
- Estimation of correlation functions
- Estimation of auto- and cross-spectra
- Estimation of the FRF and of the coherence function
- Bias due to feedback
- Case study

Overview of the process

The FRF estimation process consists of the following steps:

1. Estimation and removal of mean values from input and output data.
2. Estimation of

$$R_{uy}(\tau), \quad R_{uu}(\tau).$$

   from time-domain data.

3. Computation of Fourier transforms, to get

$$S_{uy}(f), \quad S_{uu}(f).$$

4. Estimation of the FRF using $S_{uy}(f) = G(f)S_{uu}(f)$.

Mean values estimation and removal

- Experimental data are not zero-mean most of the time.
- Example: trim values of controls and velocities/attitude in aircraft data.
- Assuming that the measured input and output data are realisations of stationary, ergodic RPs we can use the sample mean to estimate input and output mean values.
- The mean values are then removed from the measurements and zero-mean data are then employed.

Estimation of the correlation functions

- The starting point is a collection of *N* samples of *u*(*t*) and *y*(*t*) collected with uniform sampling at times

$$t_n = t_0 + nT_s, \quad n = 1, \ldots, N.$$

- By
  - T_s we denote the sampling period
  - $f_s = \frac{1}{T_s}$ we denote the sampling frequency.

Estimation of the correlation functions

- Care must be taken in defining correlation functions, as we want to estimate *continuous* correlation functions using *discrete* data.

- Given the discrete nature of data, we can only time-shift by *multiples of the sampling period.*

- In particular, we can estimates samples of the correlation functions only at time-shifts given by

$$\tau = n_\tau T_s$$

where the index n_τ is an integer.

- In the following $R_u(\tau), \quad R_u(n_\tau T_s), \quad R_u(n_\tau)$ are used interchangeably.

Estimation of the correlation functions

- The exact expression for the correlation is

$$R_u(\tau) = E[u(t)u(t+\tau)] = \lim_{T \to \infty} \frac{1}{T} \int_0^T u(t)u(t+\tau)dt$$

- So in costructing estimators we have to make a number of approximations:
  - Finite duration of the data-set: the limit operation can be only approximated by taking long datasets.
  - Sampling: integrals must be approximated by summations.

Estimation of the correlation functions

- Based on these assumptions, an estimator can be defined as

$$\hat{R}_u^1(n_\tau) = \frac{1}{N-|n_\tau|}\sum_{n=0}^{N-|n_\tau|-1} u(n)u(n+n_\tau), \qquad |n_\tau| < N.$$

- It is possible to study unbiasdness, as follows:

$$\begin{aligned} E[\hat{R}_u^1(n_\tau)] &= E[\frac{1}{N-|n_\tau|}\sum_{n=0}^{N-|n_\tau|-1} u(n)u(n+n_\tau)] = \\ &= \frac{1}{N-|n_\tau|}\sum_{n=0}^{N-|n_\tau|-1} E[u(n)u(n+n_\tau)] = \\ &= R_u(n_\tau T_s). \end{aligned}$$

- We will sometimes use the alternative estimator

$$\hat{R}_u^2(n_\tau) = \frac{1}{N} \sum_{n=0}^{N-|n_\tau|-1} u(n)u(n+n_\tau), \quad |n_\tau| < N.$$

- This estimator has a simpler expression, but it is biased:

$$E[\hat{R}_u^2(n_\tau)] = \ldots = \frac{N-|n_\tau|}{N} R_u(n_\tau T_s).$$

Estimation of the correlation functions

- In terms of variance, it can be shown that

$$\mathrm{Var}[\hat{R}_u^1(n_\tau)] = \frac{N}{(N-|n_\tau|)^2} \sum_{r=-\infty}^{\infty} R_u^2(n_\tau) + R_u(r+n_\tau)R_u(r-n_\tau)$$

- From which we can see that

$$\mathrm{Var}[\hat{R}_u^1(n_\tau)] \underset{N\to\infty}{\to} 0.$$

- And same for the second estimator.

Estimation of the correlation functions

- For cross-correlations similar definitions can be used:

$$\hat{R}^1_{uy}(n_\tau) = \frac{1}{N - |n_\tau|} \sum_{n=0}^{N-|n_\tau|-1} u(n)y(n+n_\tau), \qquad |n_\tau| < N.$$

$$\hat{R}^2_{uy}(n_\tau) = \frac{1}{N} \sum_{n=0}^{N-|n_\tau|-1} u(n)y(n+n_\tau), \qquad |n_\tau| < N.$$

- Identical conclusions can be reached for bias and variance.

Fourier transforms of the correlations

- The exact definition of the autospectrum is given by

$$S_u(f) = \int_{-\infty}^{+\infty} R_u(\tau) e^{-j2\pi f\tau} d\tau$$

- This has, again, to be approximated to account for
  - Finite duration of the correlation interval
  - Finite sampling.

- This gives

$$S_u(f) \simeq T_s \sum_{n=-(N-1)}^{N-1} R_u(n) e^{-j2\pi f n T_s}.$$

Fourier transforms of the correlations

- It is convenient to write this expression in terms of the *normalised frequency*

$$\tilde{f} = \frac{f}{f_s}.$$

- Recall that due to sampling the frequency *f* is limited to the range

$$-\frac{f_s}{2} < f < \frac{f_s}{2}$$

- Therefore the normalised frequency is limited to the range

$$-\frac{1}{2} < \tilde{f} < \frac{1}{2}.$$

Fourier transforms of the correlations

- Starting from

$$S_u(f) \simeq T_s \sum_{n=-(N-1)}^{N-1} R_u(n) e^{-j2\pi f n T_s}$$

- And recalling that $f_s = \dfrac{1}{T_s}$

- We get

$$\begin{aligned} S_u(\tilde{f}) &= T_s \sum_{n=-(N-1)}^{N-1} R_u(n) e^{-j2\pi f n \frac{1}{f_s}} \\ &= T_s \sum_{n=-(N-1)}^{N-1} R_u(n) e^{-j2\pi \tilde{f} n}. \end{aligned}$$

Estimation of the auto- and cross-spectra

- This approximate expression for the autospetrum defines estimators as soon as we «plug» estimates of the correlation in it:

$$\widehat{S}_u^1(\tilde{f}) = T_s \sum_{n=-(N-1)}^{N-1} \widehat{R}_u^1(n) e^{-j2\pi \tilde{f} n}.$$

$$\widehat{S}_u^2(\tilde{f}) = T_s \sum_{n=-(N-1)}^{N-1} \widehat{R}_u^2(n) e^{-j2\pi \tilde{f} n}.$$

- These estimators are sometimes called *periodograms* or *rough* spectral estimators.

Estimation of the auto- and cross-spectra

- Are these estimators unbiased?

- For the first one, we get

$$E[\widehat{S}_u^1(\tilde{f})] = E[T_s \sum_{n=-(N-1)}^{N-1} \widehat{R}_u^1(n) e^{-j2\pi \tilde{f} n}] =$$

$$= T_s \sum_{n=-(N-1)}^{N-1} E[\widehat{R}_u^1(n)] e^{-j2\pi \tilde{f} n} =$$

$$= T_s \sum_{n=-(N-1)}^{N-1} R_u(n) e^{-j2\pi \tilde{f} n}$$

- Which converges to the true autospectrum for fast sampling and long datasets.
- For finite samples however the estimate is biased.

- Are these estimators unbiased?

- For the second one, on the other hand

$$
\begin{aligned}
E[\hat{S}_u^2(\tilde{f})] &= E[T_s \sum_{n=-(N-1)}^{N-1} \hat{R}_u^1(n) e^{-j2\pi \tilde{f} n}] = \\
&= T_s \sum_{n=-(N-1)}^{N-1} E[\hat{R}_u^2(n)] e^{-j2\pi \tilde{f} n} = \\
&= T_s \sum_{n=-(N-1)}^{N-1} \frac{N-|n|}{N} R_u(n) e^{-j2\pi \tilde{f} n}
\end{aligned}
$$

- Again, the estimate converges to the true autospectrum for fast sampling and long datasets, but is otherwise biased.

- A better understanding of the bias in periodograms can be gathered thinking in terms of *windows*, as follows.
- For the estimators

$$\hat{S}_u^1(\tilde{f}) = T_s \sum_{n=-(N-1)}^{N-1} \hat{R}_u^1(n) e^{-j2\pi\tilde{f}n}.$$

$$\hat{S}_u^2(\tilde{f}) = T_s \sum_{n=-(N-1)}^{N-1} \hat{R}_u^2(n) e^{-j2\pi\tilde{f}n}.$$

- We have proved

$$E[\hat{S}_u^1(\tilde{f})] = T_s \sum_{n=-(N-1)}^{N-1} R_u(n) e^{-j2\pi\tilde{f}n}$$

$$E[\hat{S}_u^2(\tilde{f})] = T_s \sum_{n=-(N-1)}^{N-1} \frac{N-|n|}{N} R_u(n) e^{-j2\pi\tilde{f}n}$$

- Focus now on

$$E[\widehat{S}_u^1(\tilde{f})] = T_s \sum_{n=-(N-1)}^{N-1} R_u(n) e^{-j2\pi \tilde{f} n}$$

- And note that it can be equivalently written as

$$E[\widehat{S}_u^1(\tilde{f})] = T_s \sum_{n=-\infty}^{+\infty} w_R(n) R_u(n) e^{-j2\pi \tilde{f} n}$$

where

$$w_R(n) = \begin{cases} 1 & -(N-1) \leq n \leq N-1 \\ 0 & |n| > N-1 \end{cases}$$

is the so-called *rectangular window* of width *N.*

- Similarly for the second estimator

$$E[\hat{S}_u^2(\tilde{f})] = T_s \sum_{n=-(N-1)}^{N-1} \frac{N-|n|}{N} R_u(n) e^{-j2\pi \tilde{f} n}$$

- We have that it can be equivalently written as

$$E[\hat{S}_u^2(\tilde{f})] = T_s \sum_{n=-\infty}^{+\infty} w_B(n) R_u(n) e^{-j2\pi \tilde{f} n}$$

where

$$w_B(n) = \begin{cases} \frac{N-|n|}{N} & -(N-1) \leq n \leq N-1 \\ 0 & |n| > N-1 \end{cases}$$

is the so-called *Bartlett* (*or triangular*) *window* of width *N.*

- Therefore, windows capture precisely the bias intrinsic in the use of periodograms.
- A better insight in the role of windows is obtained by looking at the estimators in the *frequency domain.*
- For this we need to define and use the Fourier transform for discrete signals.

Interlude: Fourier transform for discrete signals

- Consider a signal defined over discrete-time *n*, *v*(*n*).
- If the series

$$V(\tilde{\omega}) = \sum_{n=-\infty}^{+\infty} v(n) e^{-j\tilde{\omega}n}$$

exists at least for some values of $\tilde{\omega}$ then it defines the Fourier Transform of *v*(*n*).

- The discrete angular frequency is such that $-\pi \leq \tilde{\omega} \leq \pi$.
- Sometimes we will use frequency $\tilde{f}$ as independent variable:

$$V(\tilde{f}) = \sum_{n=-\infty}^{+\infty} v(n) e^{-j2\pi\tilde{f}n}.$$

- Existence of the FT implies that the signal in the time domain can be expressed as

$$v(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} V(\tilde{\omega})e^{j\tilde{\omega}n}d\tilde{\omega} = \int_{-\frac{1}{2}}^{\frac{1}{2}} V(\tilde{f})e^{j2\pi\tilde{f}n}d\tilde{f}.$$

- As in continuous-time, the IFT can be interpreted as a decomposition of the signal into an infinite number of harmonics, with amplitude and phase given at each frequency $\tilde{f}$ by the magnitude and phase of the complex number $V(\tilde{f})$.

Interlude: Fourier transform for discrete signals

- For a large class of signals the FT can be computed in closed form. Here are some notable signals we will use in the following.

- Impulse:

$$\delta(n) \quad \rightarrow \quad V(\tilde{\omega}) = \sum_{n=-\infty}^{+\infty} \delta(n) e^{-j\tilde{\omega}n} = 1.$$

- Delayed impulse:

$$\delta(n-M) \quad \rightarrow \quad V(\tilde{\omega}) = \sum_{n=-\infty}^{+\infty} \delta(n-M) e^{-j\tilde{\omega}n} = e^{-j\tilde{\omega}M}.$$

- Constant:

$$1 \quad \rightarrow \quad V(\tilde{\omega}) = 2\pi\delta(\tilde{\omega}).$$

- Finally, we need the discrete version of the complex *convolution theorem*.

- For a discrete signal *h*(*n*) given by $h(n) = w(n)g(n)$ letting

$$H(\tilde{\omega}) = \sum_{n=-\infty}^{+\infty} h(n)e^{-j\tilde{\omega}n} = \sum_{n=-\infty}^{+\infty} w(n)g(n)e^{-j\tilde{\omega}n}$$

$$W(\tilde{\omega}) = \sum_{n=-\infty}^{+\infty} w(n)e^{-j\tilde{\omega}n}, \quad G(\tilde{\omega}) = \sum_{n=-\infty}^{+\infty} g(n)e^{-j\tilde{\omega}n}$$

we have

$$H(\tilde{\omega}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} G(\tilde{\theta})W(\tilde{\omega}-\tilde{\theta})d\tilde{\theta}.$$

Estimation of the auto- and cross-spectra

- In the case of the spectral estimators we have

$$E[\hat{S}_u^{R/B}(\tilde{f})] = T_s \sum_{n=-\infty}^{+\infty} w^{R/B}(n) R_u(n) e^{-j2\pi \tilde{f} n}$$

therefore letting

$$W^{R/B}(\tilde{\omega}) = \sum_{n=-\infty}^{+\infty} w^{R/B}(n) e^{-j\tilde{\omega}n}, \quad \hat{R}_u(\tilde{\omega}) = \sum_{n=-\infty}^{+\infty} \hat{R}_u(n) e^{-j\tilde{\omega}n}$$

we have

$$E[\hat{S}_u^{R/B}(\tilde{\omega})] = \frac{T_s}{2\pi} \int_{-\pi}^{\pi} R_u(\tilde{\theta}) W^{R/B}(\tilde{\omega} - \tilde{\theta}) d\tilde{\theta}.$$

- Introducing the change of variable

$$\tilde{\eta} = \tilde{\omega} - \tilde{\theta} \quad \Rightarrow \quad d\tilde{\eta} = -d\tilde{\theta}$$

we have

$$E[\widehat{S}_u^{R/B}(\tilde{\omega})] = \frac{T_s}{2\pi} \int_{-\pi}^{\pi} R_u(\tilde{\omega} - \tilde{\eta}) W^{R/B}(\tilde{\eta}) d\tilde{\eta}.$$

from which we see that

- The autospectrum is no longer equal to the FT of the correlation…
- …but rather is a weighted average where weights are given by the FT of the window function.

Consider again the rectangular and Bartlett windows.

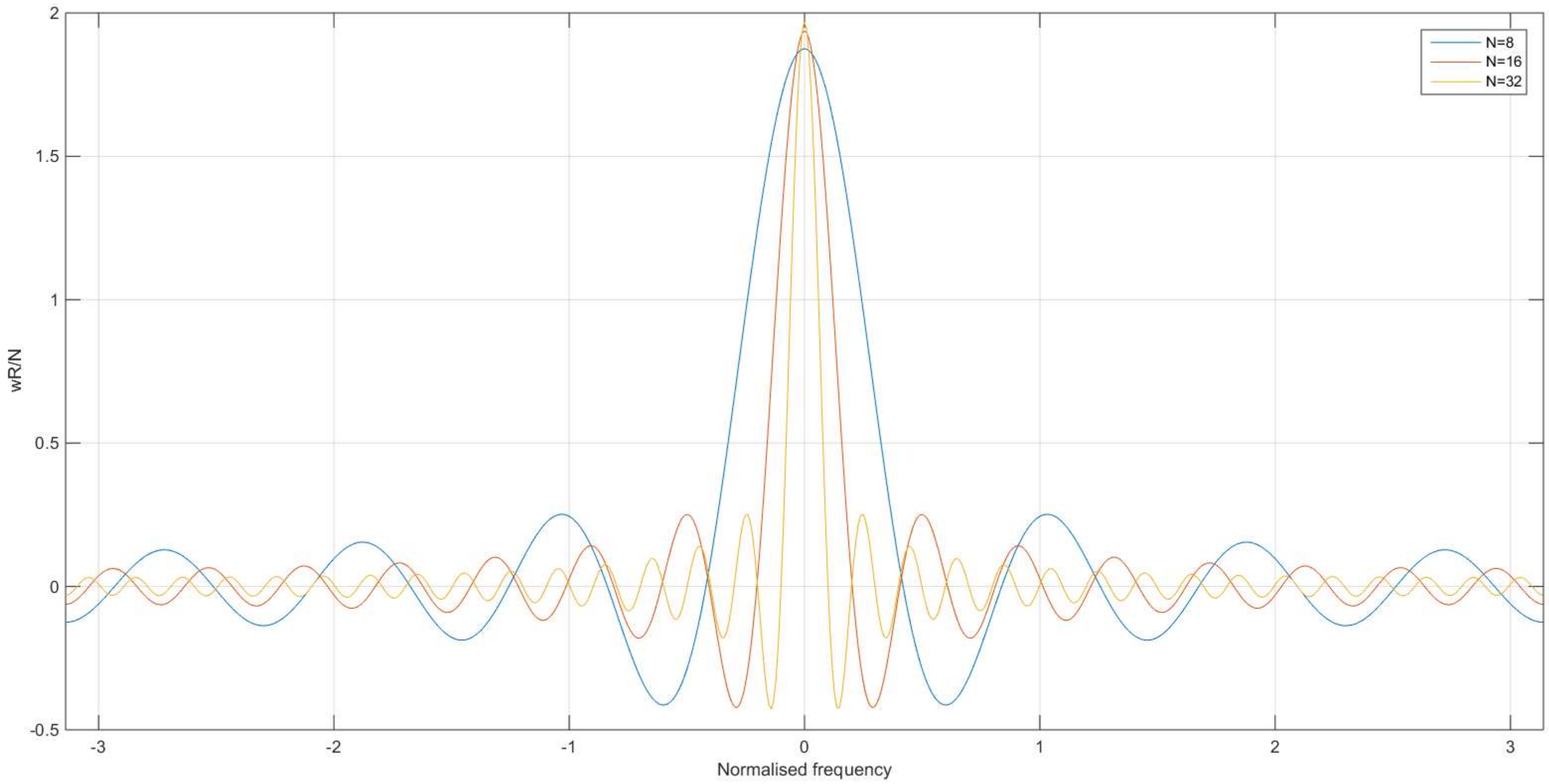
- We have for the Bartlett window:

$$w_B(n) = \begin{cases} \frac{N-|n|}{N} & -(N-1) \leq n \leq N-1 \\ 0 & |n| > N-1 \end{cases} \quad \rightarrow \quad W_B(\tilde{\omega}) = \frac{1}{N} \frac{\sin(\tilde{\omega}N/2)}{\sin(\tilde{\omega}/2)}$$

- And for the rectangular window:

$$w_R(n) = \begin{cases} 1 & -(N-1) \leq n \leq N-1 \\ 0 & |n| > N-1 \end{cases} \quad \rightarrow \quad W_R(\tilde{\omega}) = \frac{\sin(\tilde{\omega}(2N-1)/2)}{\sin(\tilde{\omega}/2)}$$

Rectangular window for increasing *N*:



- In the limit case, if we choose

$$W(\tilde{\eta}) = 2\pi\delta(\tilde{\eta}) \quad \Rightarrow \quad w(n) = 1$$

then we have

$$E[\widehat{S}_u(\tilde{\omega})] = T_s \int_{-\pi}^{\pi} R_u(\tilde{\omega} - \tilde{\eta})\delta(\tilde{\eta})d\tilde{\eta} = T_s R_u(\tilde{\omega}).$$

- Therefore a constant window leads to an estimate which is as accurate as the autocorrelation estimate.

Therefore:

- Bias can be interpreted in the frequency-domain as a smoothing effect introduced by the windows.
- The windows become narrower for increasing *N*.
- Asymptotically for large *N* the considered windows converge to impulses.

- Variance of the estimates: it can be proved that for both estimators

$$\text{Var}[\widehat{S}_u(\tilde{f})] \div \left(S_u(\tilde{f})\right)^2.$$

- Therefore the variance of these estimators is very large, which makes their application critical.

- Another issue is so-called *asymptotic incorrelation*, namely the fact that

$$E[\widehat{S}_u(\tilde{f}_1)\widehat{S}_u(\tilde{f}_2)] \underset{N\to\infty}{\to} 0$$

even for arbitrarily close pairs of frequencies.

- Therefore we must find a way to
  - Reduce the variance
  - Reduce the effect of asymptotic incorrelation.

- Two approaches have been developed to improve the performance of the periodogram:
  - Averaging
  - Windowing.

Averaging

The averaging, or Bartlett's, method proceeds as follows.

- The dataset of *N* samples is divided in *K* sequences of *M* samples each, so that *N*=*KM*.

- For each of the *K* sequences a periodogram is computed:

$$\widehat{S}_u^{(i)}(\tilde{f}) = T_s \sum_{n=-(M-1)}^{M-1} \widehat{R}_u^{(i)}(n) e^{-j2\pi \tilde{f} n}, \qquad i = 1, \dots, K.$$

- The averaged estimate is defined as

$$\widehat{S}_u(\tilde{f}) = \frac{1}{K} \sum_{i=1}^{K} \widehat{S}_u^{(i)}(\tilde{f}).$$

Averaging

- In terms of variance, assuming that the *K* estimates are independent we have that

$$\mathrm{Var}[\widehat{S}_u(\tilde{f})] \div \frac{1}{K}\left(S_u(\tilde{f})\right)^2 .$$

- So by averaging it is possible to reduce the variance, but it must be observed that each of the *K* estimators will have a larger bias
- Indeed each is based on 1/*K* fraction of the entire dataset so corresponds to the application of a wider window.
- Therefore, a bias/variance tradeoff is necessary.

Overlapped averaging

- The adverse effect of averaging on bias can be mitigated by means of *overlapped averaging*.
- The idea is to define subsequences with partial overlap among consecutive ones.
- Clearly the higher the percentage of overlap the longer will be each subsequence for given K.
- A typical choice is 50% overlap.

Windowing

- Consider again the estimators

$$\widehat{S}_u^1(\tilde{f}) = T_s \sum_{n=-(N-1)}^{N-1} \widehat{R}_u^1(n) e^{-j2\pi \tilde{f} n}.$$

$$\widehat{S}_u^2(\tilde{f}) = T_s \sum_{n=-(N-1)}^{N-1} \widehat{R}_u^2(n) e^{-j2\pi \tilde{f} n}.$$

- And recall we have proved that

$$E[\widehat{S}_u^1(\tilde{f})] = T_s \sum_{n=-(N-1)}^{N-1} R_u(n) e^{-j2\pi \tilde{f} n} = T_s \sum_{n=-\infty}^{+\infty} w_R(n) R_u(n) e^{-j2\pi \tilde{f} n}$$

$$E[\widehat{S}_u^2(\tilde{f})] = T_s \sum_{n=-(N-1)}^{N-1} \frac{N-|n|}{N} R_u(n) e^{-j2\pi \tilde{f} n} = T_s \sum_{n=-\infty}^{+\infty} w_B(n) R_u(n) e^{-j2\pi \tilde{f} n}$$

Windowing

- These observation lead to the definition of a more general estimator in the form

$$\hat{S}_u^W(\tilde{f}) = T_s \sum_{n=-\infty}^{+\infty} w(n)\hat{R}_u(n)e^{-j2\pi\tilde{f}n}$$

- In which the window function *w*(*n*) can be suitably designed to improve the quality of the estimate.

- The problem of designing the window function can be formalised using the Fourier transform for discrete signals.

- For this generic estimator we have

$$E[S_u^W(\tilde{\omega})] = T_s \int_{-\pi}^{\pi} \hat{R}_u(\tilde{\omega} - \tilde{\eta}) W(\tilde{\eta}) d\tilde{\eta}.$$

- Key idea: the *width* of the window does not have to coincide with the length of the dataset.

- For example we can consider the triangular window

$$w_B(n) = \begin{cases} \frac{M-|n|}{M} & -(M-1) \leq n \leq M-1 \\ 0 & |n| > M-1 \end{cases}$$

for $M < N$.

- In terms of variance it can be shown that

$$\mathrm{Var}[\widehat{S}_u^W(\tilde{f})] \div R\left(S_u(\tilde{f})\right)^2$$

where

$$R = \frac{1}{N}\sum_{m=-(M-1)}^{M-1} w^2(m).$$

- For example, for the triangular window

$$R = \frac{2}{3}\frac{M}{N}$$

so a reduction in variance can be obtained just by rescaling the window length.

Windowing

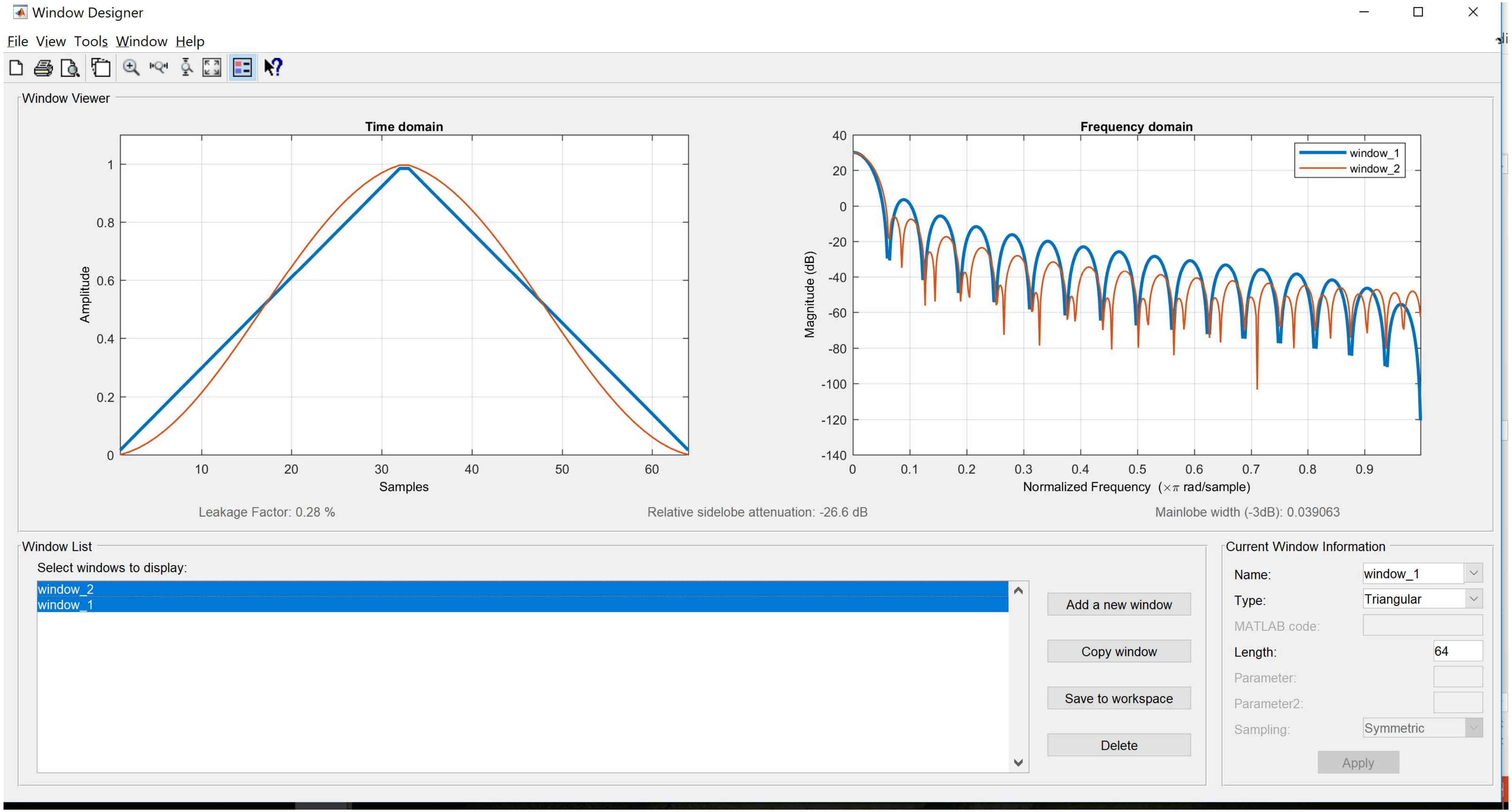
- Furthermore, the shape of the window can be modified, with respect to rectangular or triangular, to improve its performance.

- Many window designs have been proposed over the years, aimed at solving specific problems in spectral estimation.

- The most frequenly used in the Hanning window, which is a modification of the triangular one.

- Windows can be analysed using the Matlab *wintool* GUI.

Example: triangular vs Hanning window



The Welch spectral estimator

- The most popular approach to the problem is the so-called Welch method, which consists of the following steps:

1. The original dataset of length *N* is broken into *K* datasets of length *M* each, usually with 50% overlap.

2. Then *K* windowed estimates are computed, to get

$$S_u^{W,(k)}(\tilde{f}), \quad k = 1, \ldots, K.$$

3. Finally, the *K* estimates are averaged:

$$\widehat{S}_u(\tilde{f}) = \frac{1}{K} \sum_{k=1}^{K} \widehat{S}_u^{W,(k)}(\tilde{f}).$$

Resolution of spectral estimates

- By *resolution* we mean the smallest difference in frequencies which can be seen in the spectral estimate.

- For example, if a signal is given by

$$u(t) = \sin(2\pi f_1 t) + \sin(2\pi f_2 t)$$

  what is the smallest difference $f_2 - f_1$ which can be resolved in the spectral estimate?

- Accurate evaluation of the resolution is a non-trivial task.

Resolution of spectral estimates

- Roughly, for a sequence of length *N* with sampling period T_s, the resolution is equal to

$$\Delta f = \frac{1}{NT_s}$$

  i.e., the inverse of the length of the sequence in seconds.

- Clearly, if averaging is used and each subsequence is of length *M* then

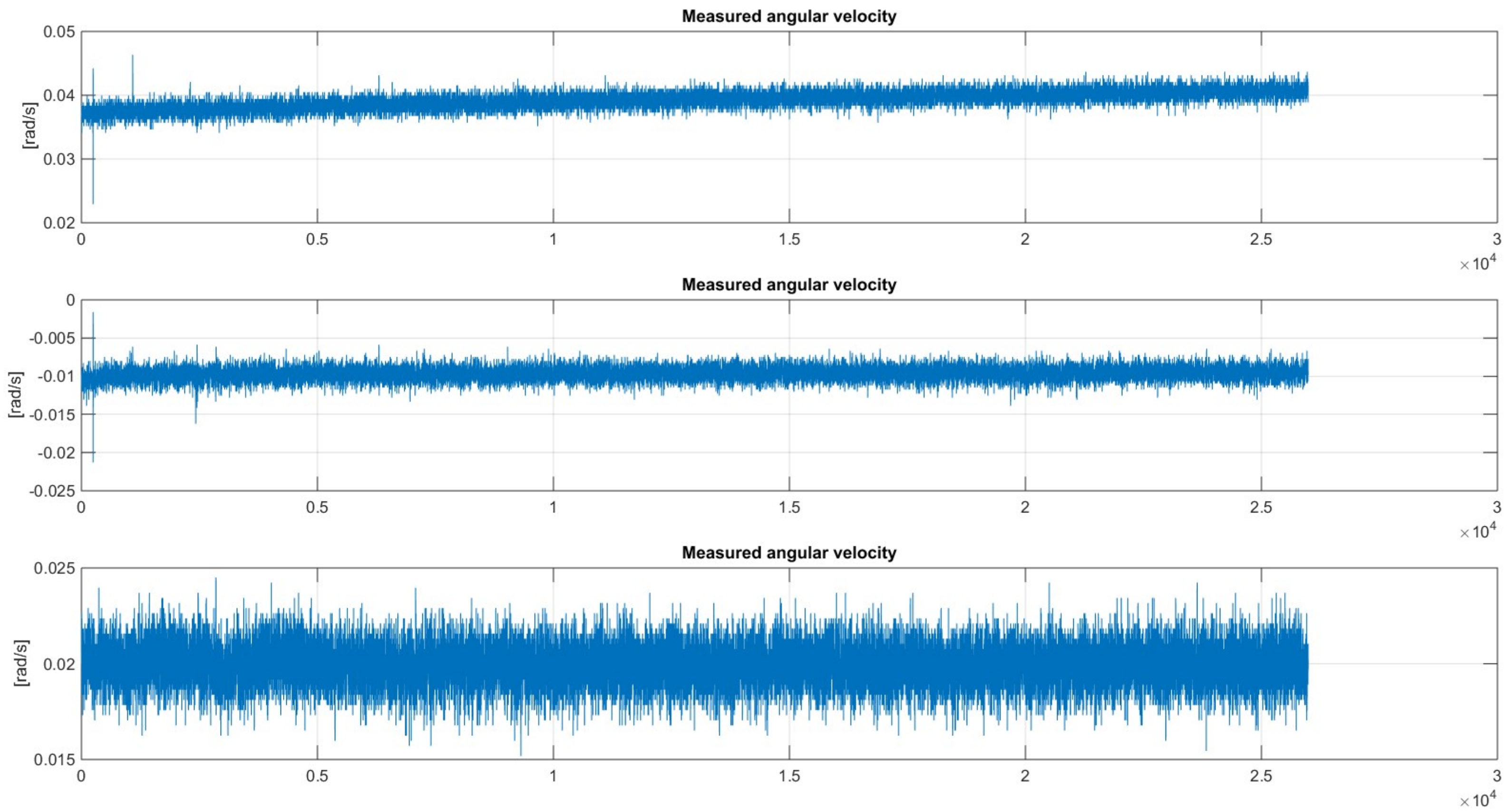
$$\Delta f = \frac{1}{MT_s}.$$

- Therefore averaging reduces the variance but leads to a loss of resolution.

- The effect of windowing is harder to assess, but it generally leads to a small improvement in resolution.

Example: gyro data

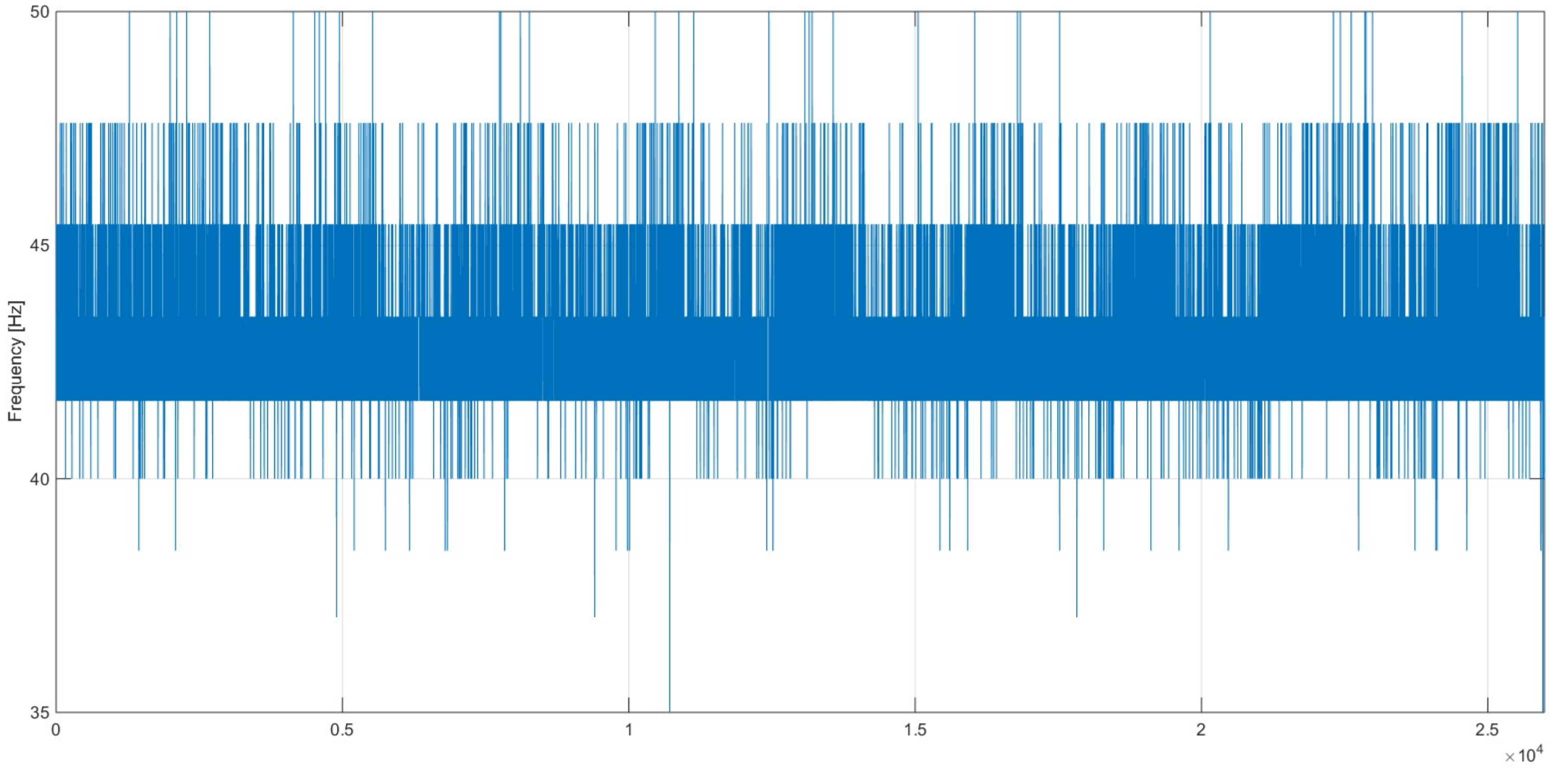
N=25991, sampling frequency approx 50 Hz, res. 0.002 Hz.



Example: gyro data

Nominal sampling frequency: 50 Hz. What we actually get is:

```
fsamp=1./diff(dati(:,2))*1e3;
fsampm=mean(fsamp(1:25500));
plot(fsamp),grid
```



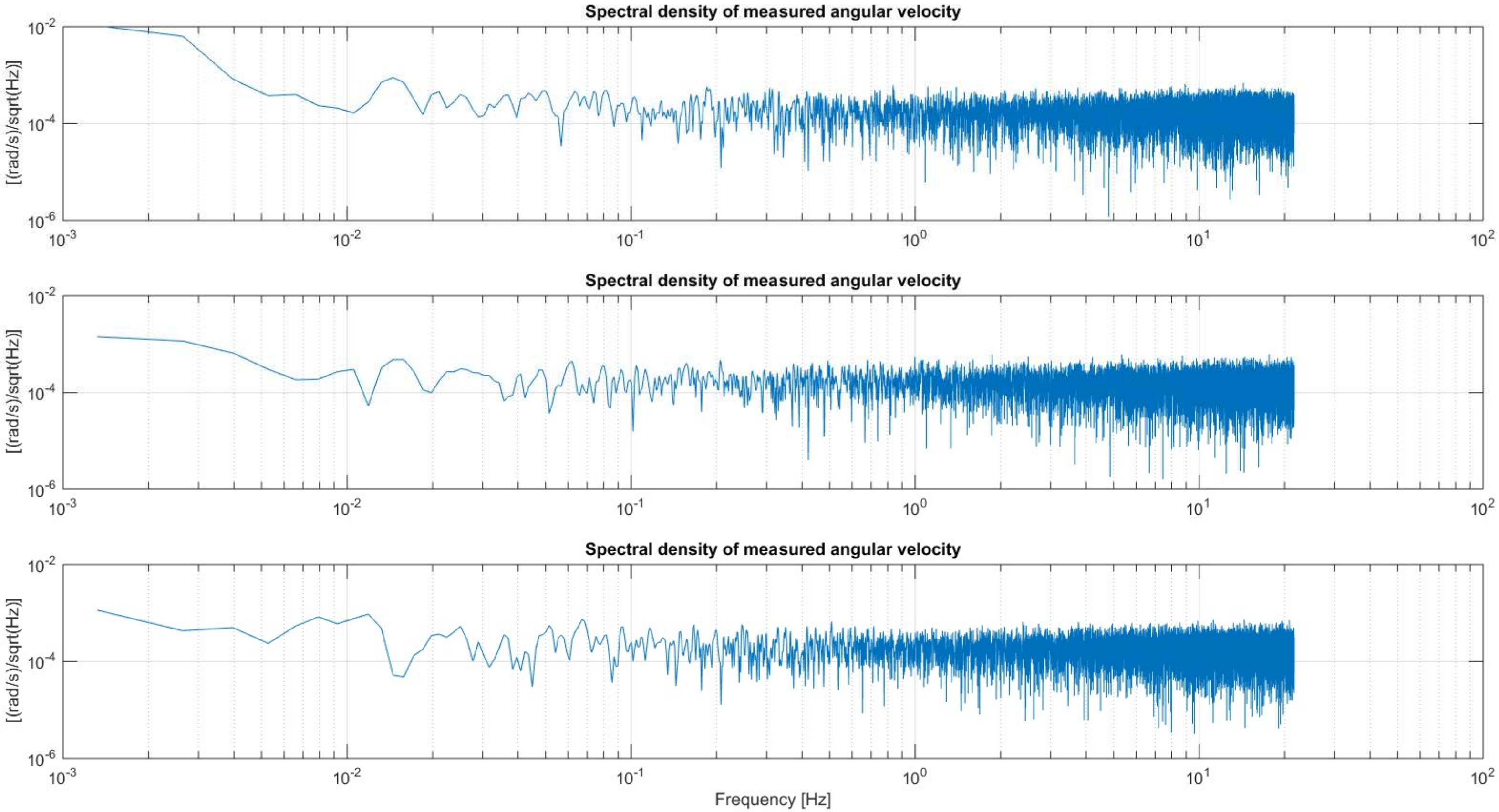
Example: gyro data


```
%
% PSD analysis
%

na=1;

[Pxx,f,fbin] = pwelchrun(dati(:,3)-mean(dati(:,3)),na,fsampm);
[Pyy,f,fbin] = pwelchrun(dati(:,4)-mean(dati(:,4)),na,fsampm);
[Pzz,f,fbin] = pwelchrun(dati(:,5)-mean(dati(:,5)),na,fsampm);


subplot(311)
loglog(f,sqrt(Pxx)),grid
ylim([1e-6,1e-2])
xlim([1e-3,1e2])
title('Spectral density of measured angular velocity')
ylabel('[(rad/s)/sqrt(Hz)]')
```

Example: gyro data

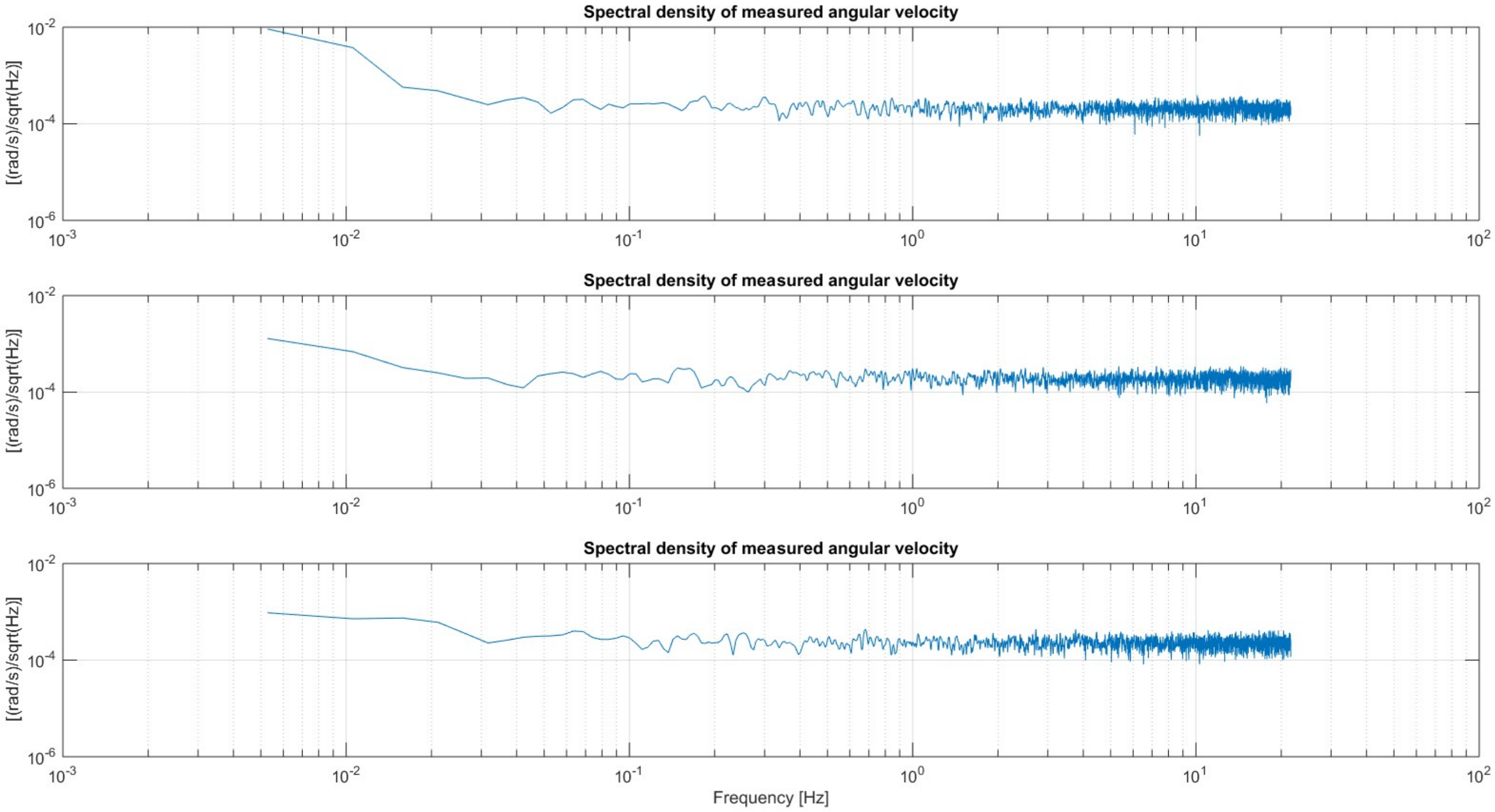

```
function [Pxx,f,fbin] = pwelchrun(x,na,fsamp)
%
% Calls pwelch to compute the one-sided PSD of signal x,
   with an averaging
% factor of na and a sampling frequency fsamp.
%

%Window
nx = max(size(x));
w = hanning(floor(nx/na));

[Pxx,f] = pwelch(x,w,0,[],fsamp,'onesided');

fbin = f(2) - f(1);
```

Without averaging:



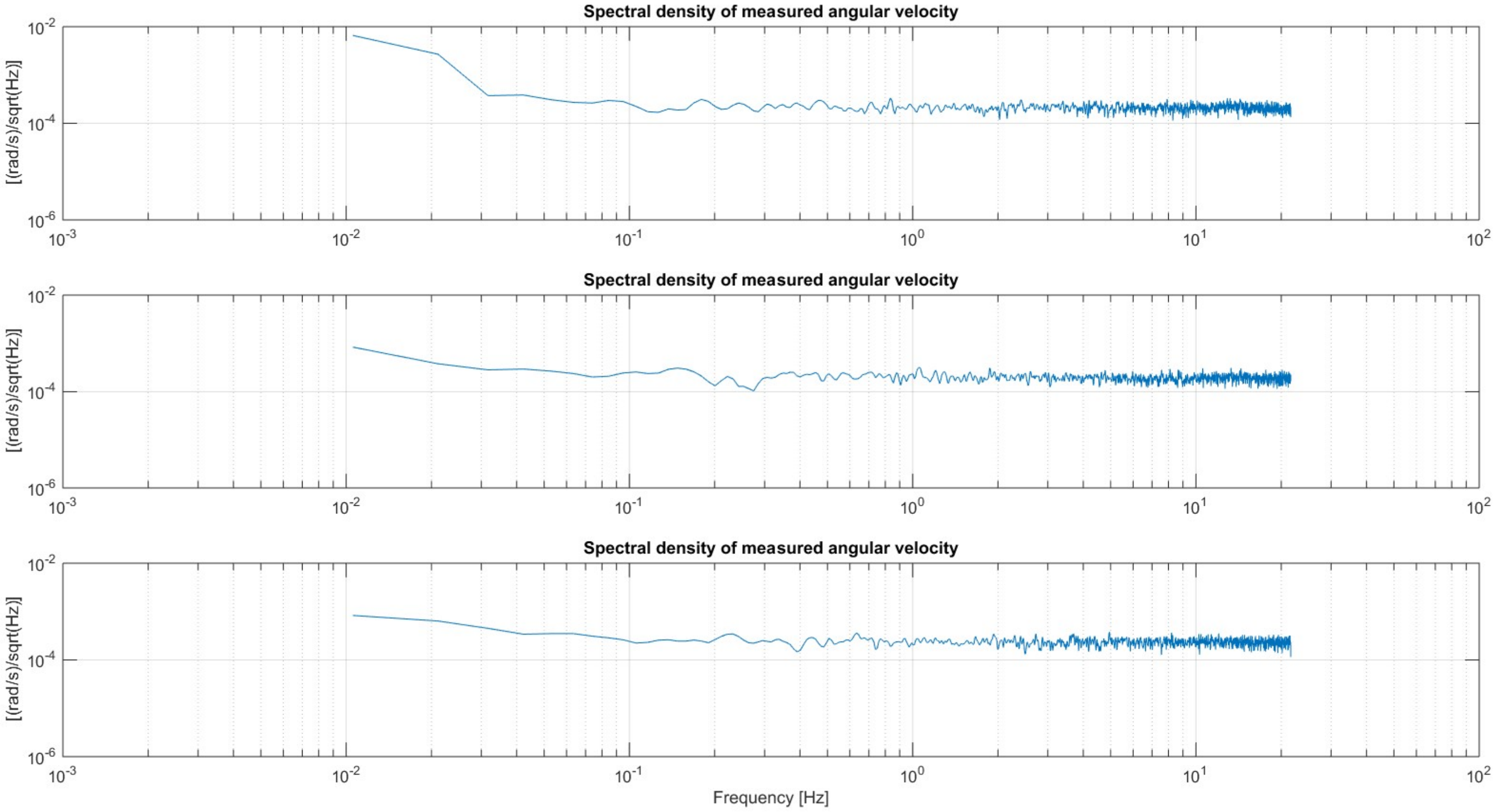
Example: gyro data

With *K*=5 averaging:



Example: gyro data

With K=10 averaging:



Comments:

- In the time-domain we see faster drift in the *x*-axis measurement, this is apparent also from the spectral density.
- All three axes seem to have the same ARW.
- The numerical value of ARW can be read directly from the plot (but recall this is a *one-sided* PSD).

- Finally, when the estimates of the input auto-spectrum and input-output cross-spectrum have been computed, the point estimate of the FRF can be obtained as

$$\widehat{G}(f) = \frac{\widehat{S}_{uy}(f)}{\widehat{S}_{uu}(f)}.$$

- Frequency by frequency the qualityof the estimate can be assessed using the coherence function:

$$\gamma^2_{uy}(f) = \frac{|S_{uy}(f)|^2}{S_{yy}(f)S_{uu}(f)}$$

which can be estimated using the estimates of the spectra:

$$\widehat{\gamma}^2_{uy}(f) = \frac{|\widehat{S}_{uy}(f)|^2}{\widehat{S}_{yy}(f)\widehat{S}_{uu}(f)}.$$

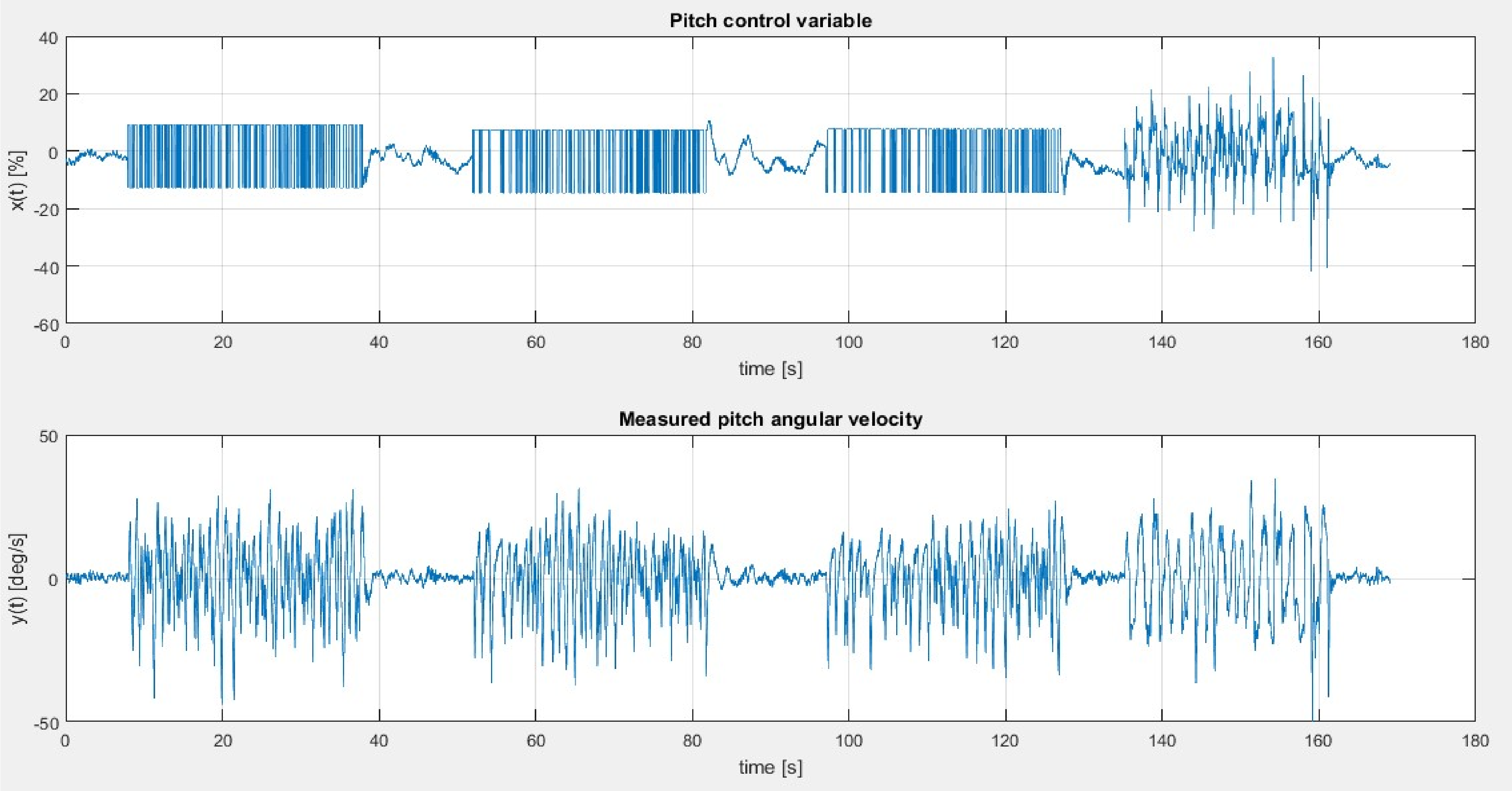
Case study:
FRF estimation for a variable pitch quadrotor

- MTOW = 5 kg
- Variable collective pitch (fixed RPM)
- Arms length = 0.415 m
- Rotors radius = 0.27 m

Case study

- Input signal: difference between collective pitch command % of back and front rotors $\rightarrow u(t)$ [%]
- Output signal: measured pitch angular velocity $\rightarrow y(t)$ $[deg/s]$
- PRBS (Pseudo Random Binary Sequences) excitation sequences
- Sampling frequency: $50\ Hz$
- Time of record: $168.993\ s$
- Number of samples: 8451
- Output delay: $0.06\ s$

Case study


Pitch control variable
x(t) [%]
40
20
0
-20
-40
-60
0
20
40
60
80
100
120
140
160
180
time [s]
Measured pitch angular velocity
y(t) [deg/s]
50
0
-50
0
20
40
60
80
100
120
140
160
180
time [s]

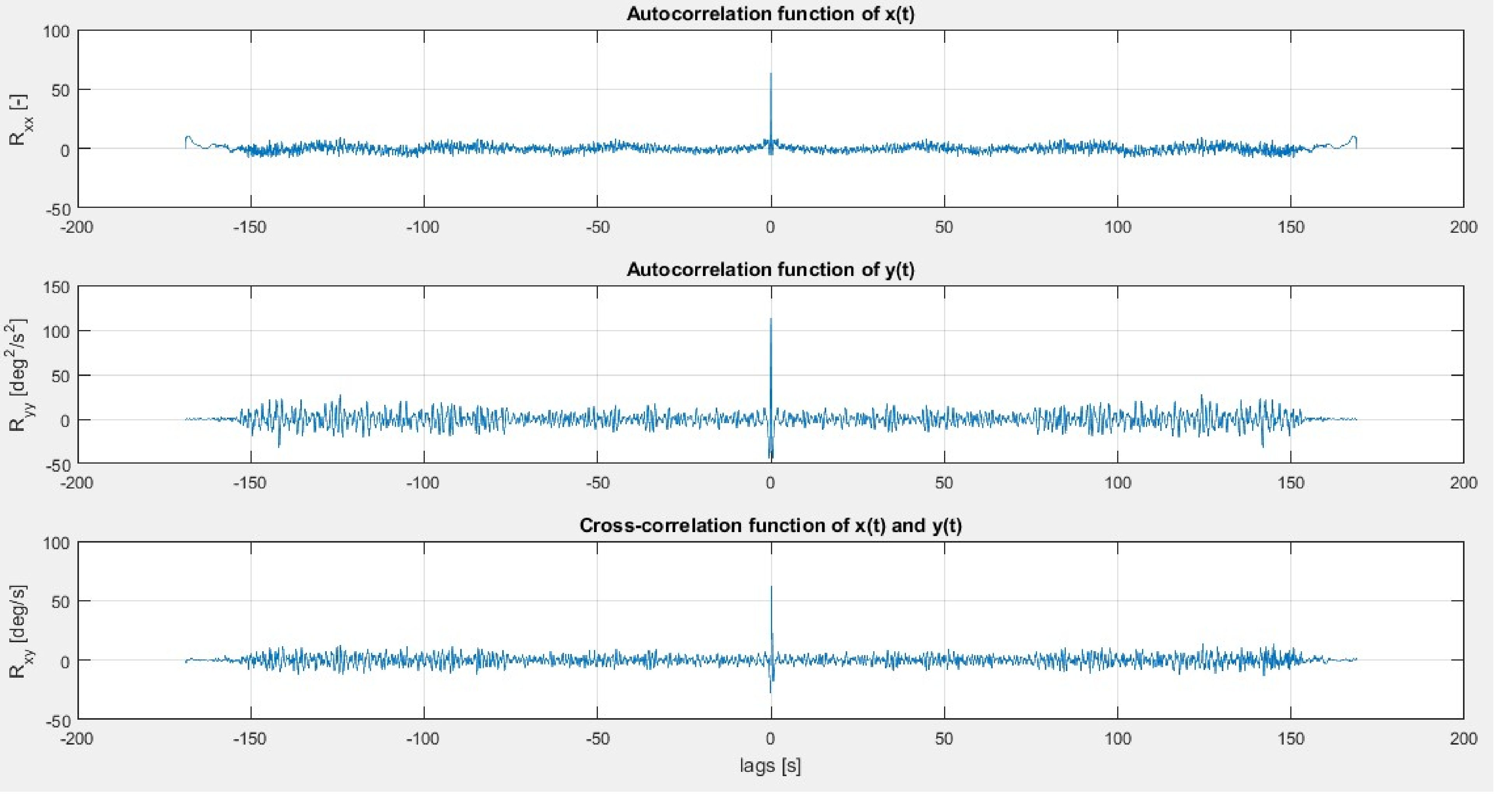
Case study

- Data of interest are measurements of two continuous random processes $\{u(t)\}$ and $\{y(t)\}$, which are assumed to be stationary
- Introducing an additional variable, *i.e.*, a time shift τ between $u(t)$ and $y(t)$, the correlation functions between $u(t)$ and $y(t)$ for any time delay τ are defined as follows.

Case study

```
%% Correlation function

N = length(t);                    % [-] number of samples


%% Subtracting means from original time data

x = u-mean(u);
y = q-mean(q);


%% Compute R_x, R_y, R_xy and R_yx

R_xx = zeros(N,1);
R_yy = zeros(N,1);
R_xy = zeros(N,1);
R_yx = zeros(N,1);

for n_tau = 1:N-1
    for n = 1:N-abs(n_tau)-1
        R_xx(n_tau) = R_xx(n_tau)+(sum(x(n)*x(n+n_tau)))/(N-abs(n_tau));
        R_yy(n_tau) = R_yy(n_tau)+(sum(y(n)*y(n+n_tau)))/(N-abs(n_tau));
        R_xy(n_tau) = R_xy(n_tau)+(sum(x(n)*y(n+n_tau)))/(N-abs(n_tau));
        R_yx(n_tau) = R_yx(n_tau)+(sum(y(n)*x(n+n_tau)))/(N-abs(n_tau));
    end
end
```

Case study



Autocorrelation function of x(t)
R_xx [-]
100
50
0
-50
-200
-150
-100
-50
0
50
100
150
200
Autocorrelation function of y(t)
R_yy [deg^2/s^2]
150
Cross-correlation function of x(t) and y(t)
R_xy [deg/s]
lags [s]

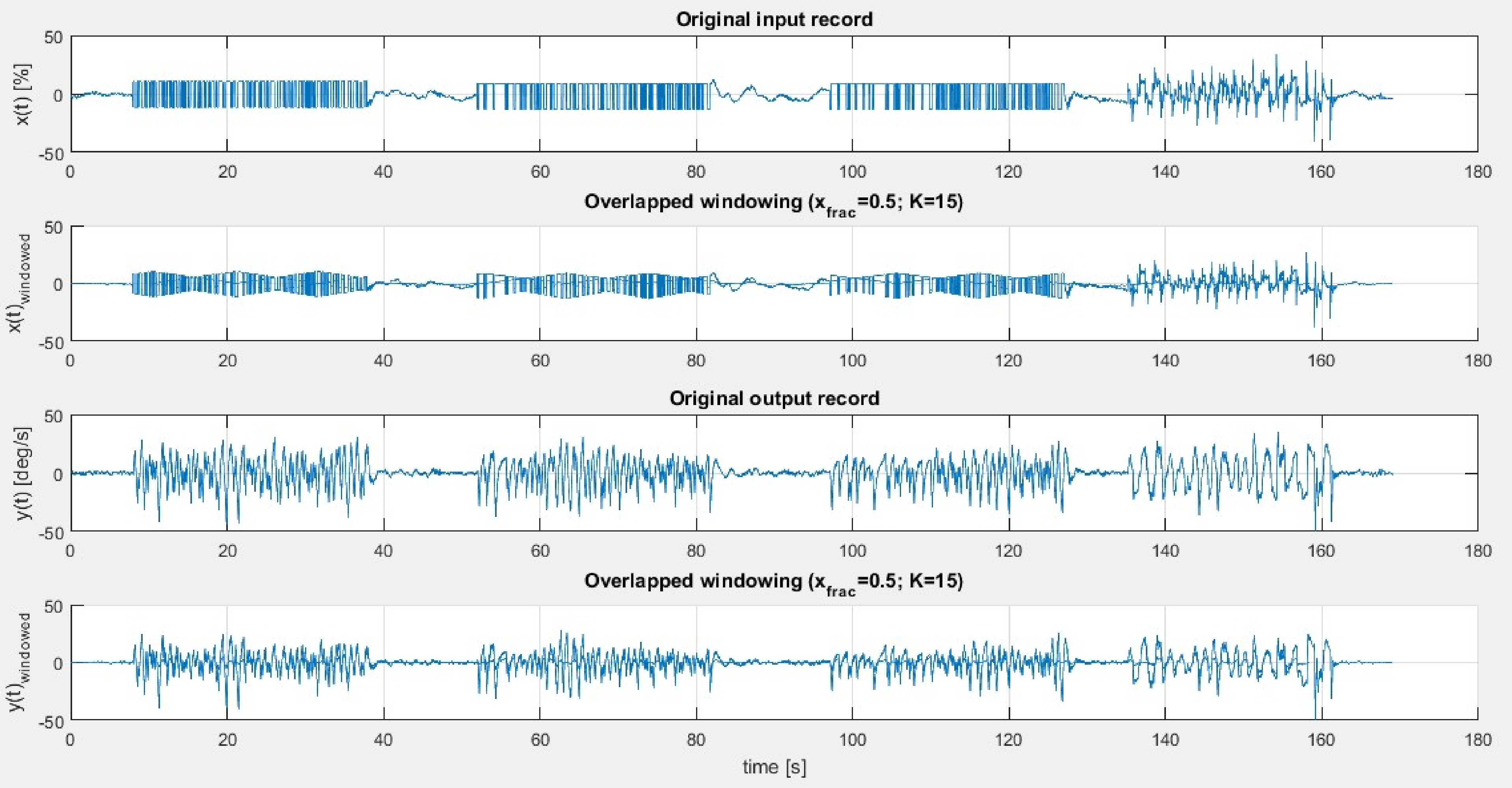
Overlapped windowing


```
N = length(t);                  % [-] number of samples
T_rec = t(end);                 % [s] records time
T_s = 0.02;                     % [s] sampling time
f_s = 1/T_s;                    % [Hz] sampling frequency
output_delay = 0.06;            % [s] output delay

% Subtracting means from original time history data

x_withoutmean = u-mean(u);
y_withoutmean = q-mean(q);

% Input variables

x_frac = 0.5;                   % overlap fraction
K = 111;                        % intervals in which dividing the records


% Window length

T_win = T_rec/((K-1)*(1-x_frac)+1);


% Number of samples in each window

N_win = round(N/((K-1)*(1-x_frac)+1));
```

Overlapped windowing


```
%% Subdivision of data into K records of individual length T_win

x_int = cell(1,K);              % preallocation
y_int = cell(1,K);              % preallocation
t_int = cell(1,K);              % preallocation

for k=2:K-1
  x_int{1} = x_withoutmean(1:N_win);
  x_int{k} = x_withoutmean((1-x_frac)*(k-1)*N_win:(1-x_frac)*(k-1)*N_win+N_win);
  x_int{K} = x_withoutmean(end-N_win:end);
  y_int{1} = y_withoutmean(1:N_win);
  y_int{k} = y_withoutmean((1-x_frac)*(k-1)*N_win:(1-x_frac)*(k-1)*N_win+N_win);
  y_int{K} = y_withoutmean(end-N_win:end);
  t_int{1} = t(1:N_win);
  t_int{k} = t((1-x_frac)*(k-1)*N_win:(1-x_frac)*(k-1)*N_win+N_win);
  t_int{K} = t(end-N_win:end);
end

%% Windowing
x_window = cell(1,K);           % preallocation
y_window = cell(1,K);           % preallocation

for k=1:K
    x_window{k} = x_int{k}.*bartlett(length(x_int{k}));
    y_window{k} = y_int{k}.*bartlett(length(y_int{k}));
end
```

Overlapped windowing



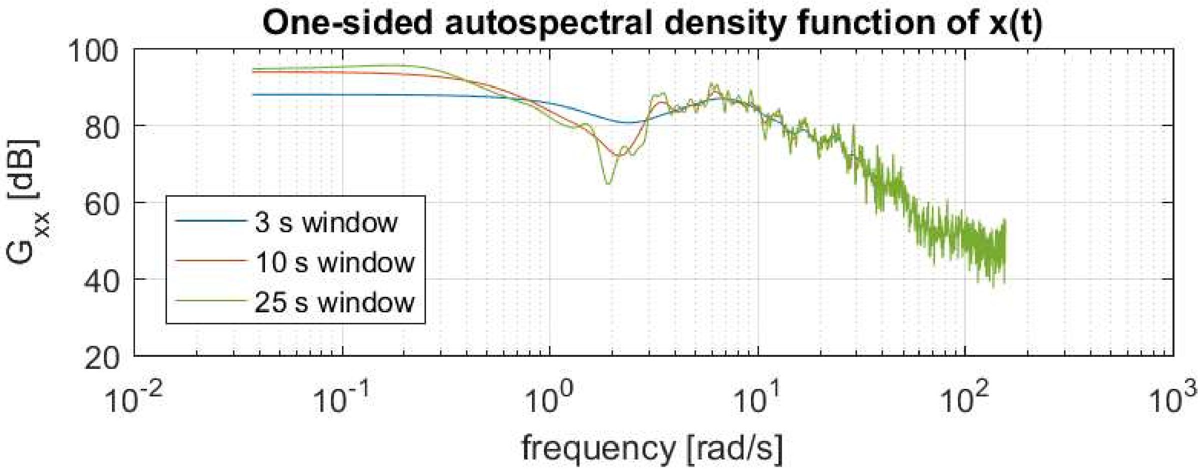
Fourier Transforms and rough estimates

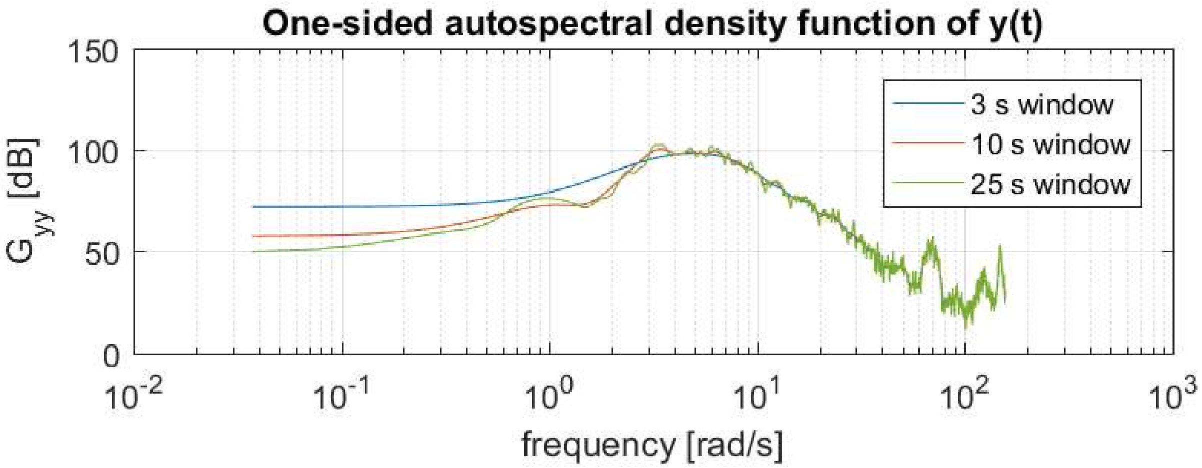

```
%% Discrete Fourier transform

X1 = cell(1,K);                 % preallocation
X = cell(1,K);                  % preallocation
Y1 = cell(1,K);                 % preallocation
Y = cell(1,K);                  % preallocation

for k=1:K
    X1{k} = fft (x_window{k},N);
    X{k} = X1{k}(1:(N+1)/2);
    Y1{k} = fft (y_window{k},N);
    Y{k} = Y1{k}(1:(N+1)/2);
end

%% Frequency

f = ((0:(N-1)/2)*f_s/N)';

%% Rough estimate

G_xx_rough = cell(1,K);         % preallocation
G_yy_rough = cell(1,K);         % preallocation
G_xy_rough = cell(1,K);         % preallocation

for k=1:K
    G_xx_rough{k} = abs(X{k}).^2*2/T_win;
    G_yy_rough{k} = abs(Y{k}).^2*2/T_win;
    G_xy_rough{k} = conj(X{k}).*Y{k}*2/T_win;
end
```

Smooth estimates

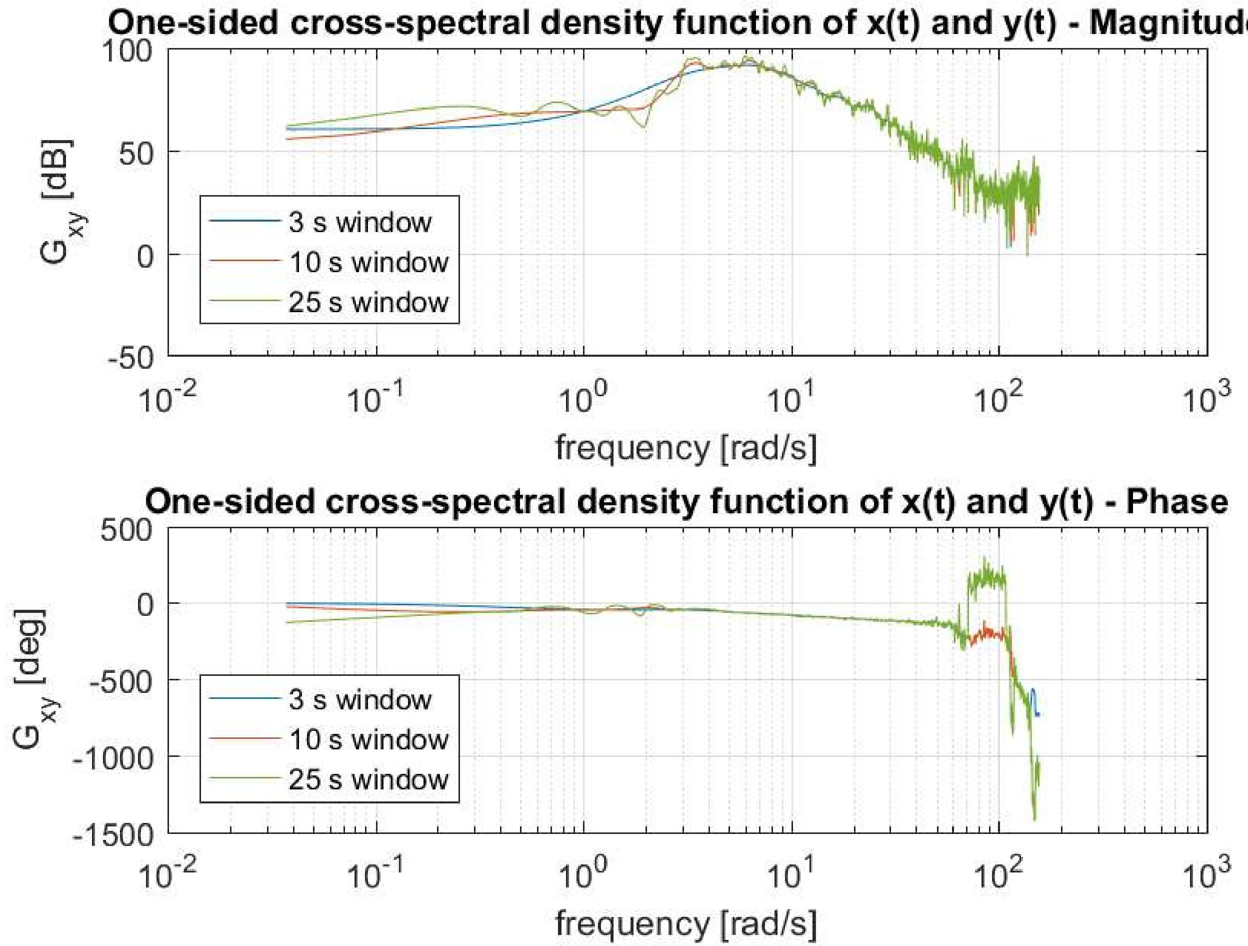

```
%% Smooth estimate

G_xx_mat = cell2mat(G_xx_rough); %converts a cell array into an ordinary array
G_xx = mean(G_xx_mat,2);         %computes mean

G_yy_mat = cell2mat(G_yy_rough); %converts a cell array into an ordinary array
G_yy = mean(G_yy_mat,2);         %computes mean

G_xy_mat = cell2mat(G_xy_rough); %converts a cell array into an ordinary array
G_xy = mean(G_xy_mat,2);         %computes mean


%% Smooth estimate-iterative procedure

G_xx = cell(1,K);               % preallocation
G_yy = cell(1,K);               % preallocation
G_xy = cell(1,K);               % preallocation

for k=2:K
    G_xx{1} = G_xx_rough{1};
    G_xx{k} = G_xx{k-1}+1/k*(G_xx_rough{k}-G_xx{k-1});
    G_yy{1} = G_yy_rough{1};
    G_yy{k} = G_yy{k-1}+1/k*(G_yy_rough{k}-G_yy{k-1});
    G_xy{1} = G_xy_rough{1};
    G_xy{k} = G_xy{k-1}+1/k*(G_xy_rough{k}-G_xy{k-1});
end
```

Smooth estimates

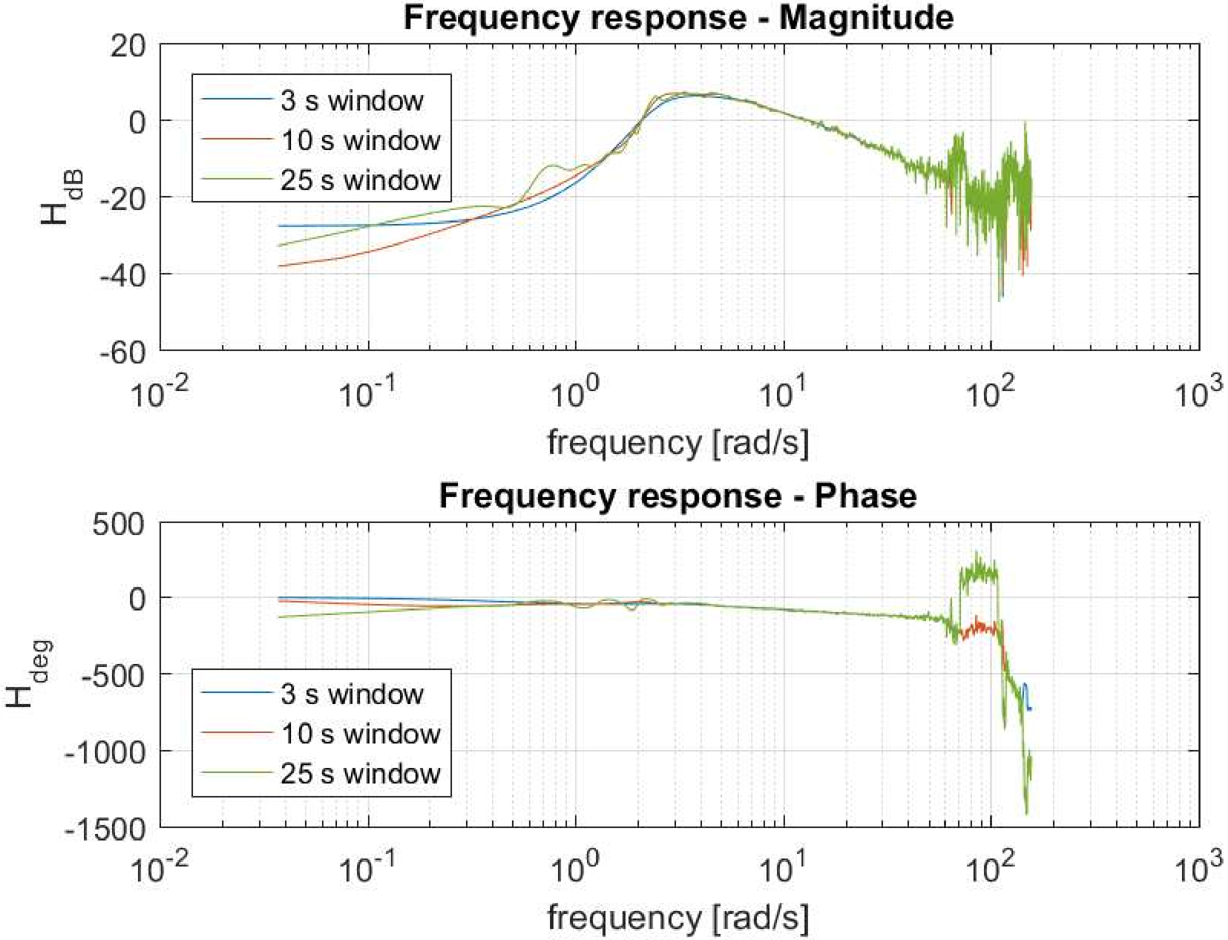




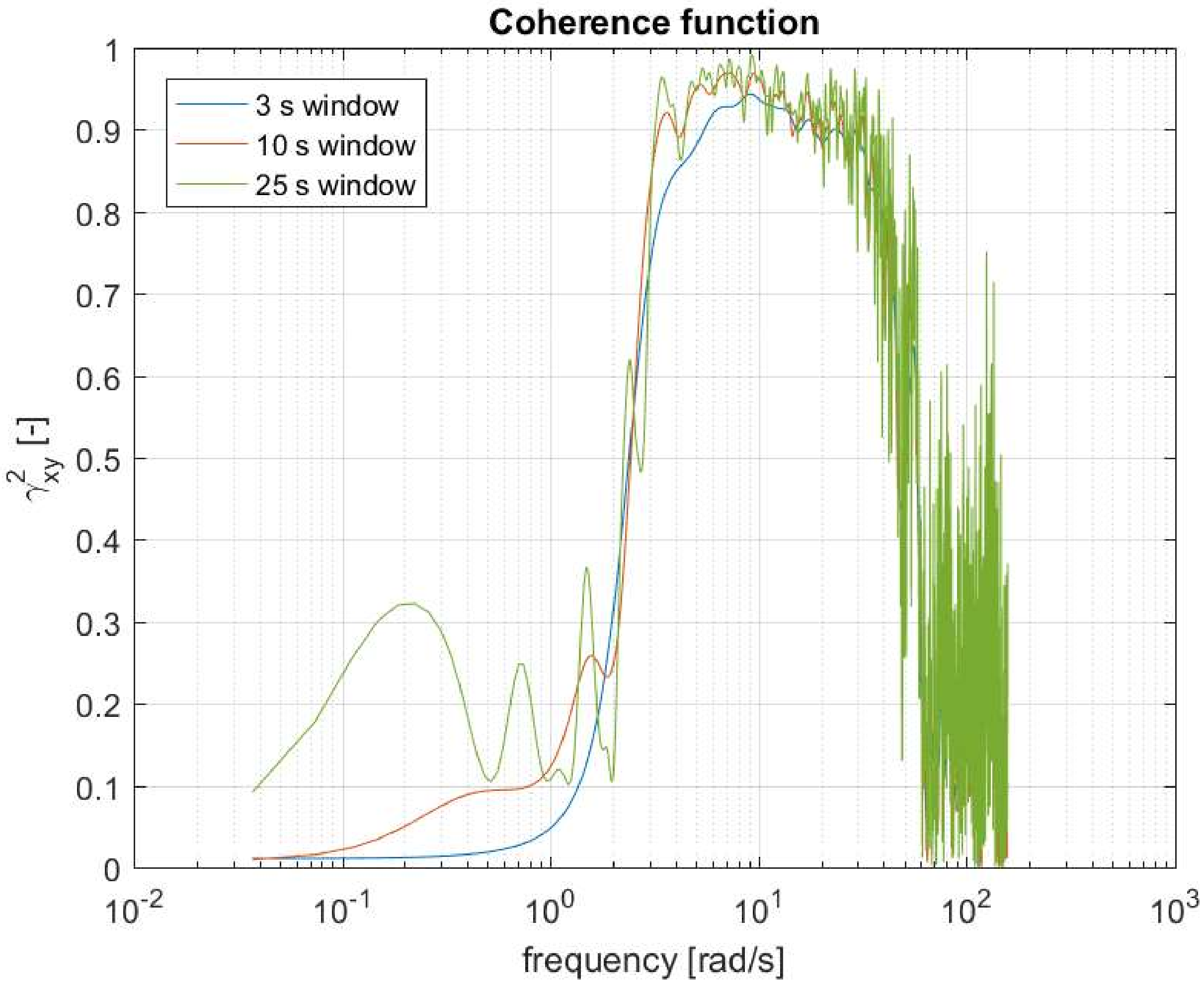
Smooth estimates



Non-parametric frequency response function estimation



Coherence function



Effect of output measurement noise