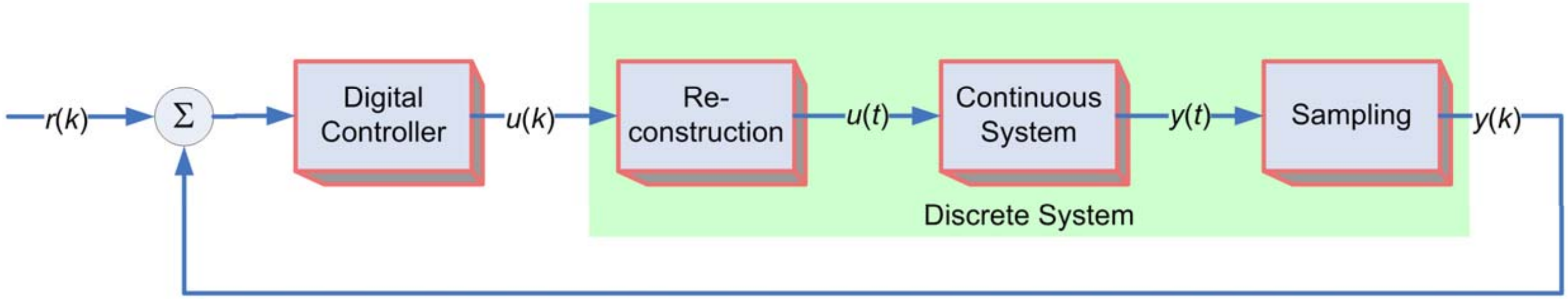
System Identification & Parameter Estimation

Wb2301: SIPE lecture 4

Perturbation signal design

Alfred C. Schouten, Dept. of Biomechanical Engineering (BMechE), Fac. 3mE
3/9/2010

Resume previous lectures: Discrete and Continuous Signals



- **Signal conversion: continue to discrete**
  AD conversion or sampling
  - Shannon (or Nyquist): $f < f_s/2$
  - Dirac's comb (& limited T): $\Delta f = 1/T$

- **Signal conversion: discrete to continuous**
  DA conversion or reconstruction
  - zero order hold (introduces higher frequencies, above f_s)

Contents Lecture 4

- Sources for error in (spectral) estimators:
  - Aliasing
  - Leakage
  - Low signal-to-noise ratio

- Optimal perturbation signals
  - Improve signal-to-noise ratio
  - Prevent aliasing
  - Prevent leakage

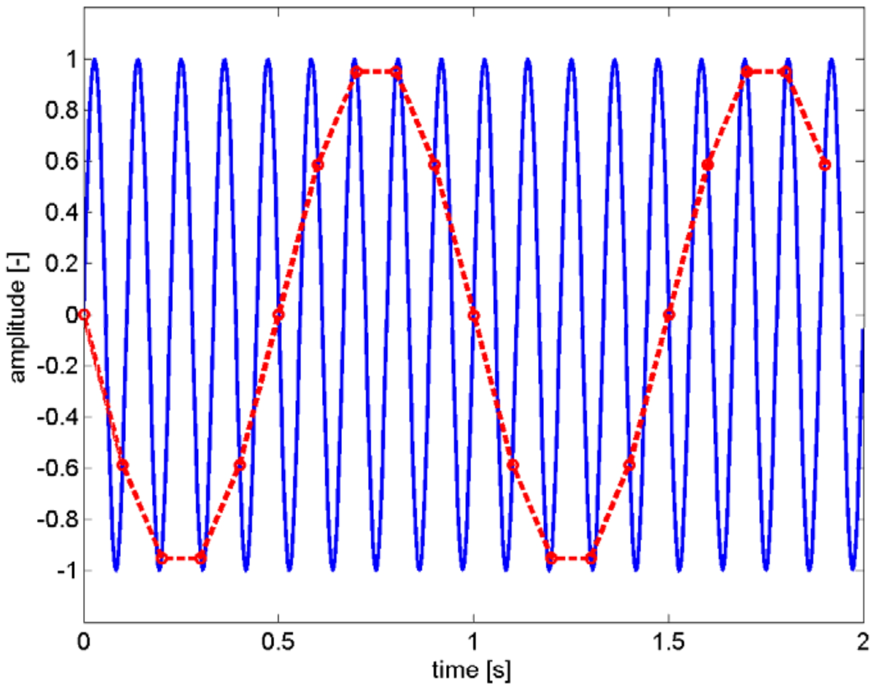
- Estimation with periodic signals
  - Spectral densities + variance
  - FRF + variance

Recording of signals

- In practical situations a **limited time** is recorded and digitized with a given **sample frequency**

- What errors can/do occur as a result
  - The sampling theorem (also Nyquist or Shannon)
    - Signal > $f_s/2$ => aliasing (frequency folding)
  - Resolution in frequency domain (Dirac's comb)
    - $\Delta f = T^{-1}$
    - What if frequency does not 'fit' on Δf

Example aliasing



Matlab: Lec4_AliasingDemo.m

- 9 Hz sine
- f_s = 10 Hz

- 9 Hz > $f_s/2$, and 1 Hz is 'seen'

- Frequencies above the Nyquist frequency are mirrored around $f_s/2$

How to prevent aliasing

- Sample high enough:
  - rule of thumb: sample frequency should be at least 2.5 times the cut-off frequency (preferably more)
- Apply anti-aliasing filter:
  - before digitizing!
  - analog filter!

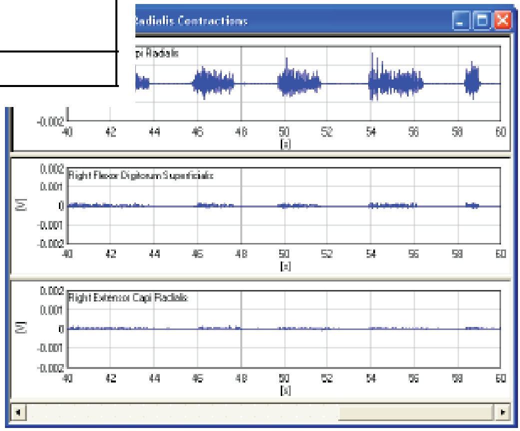
Example anti-aliasing; EMG system

Specifications

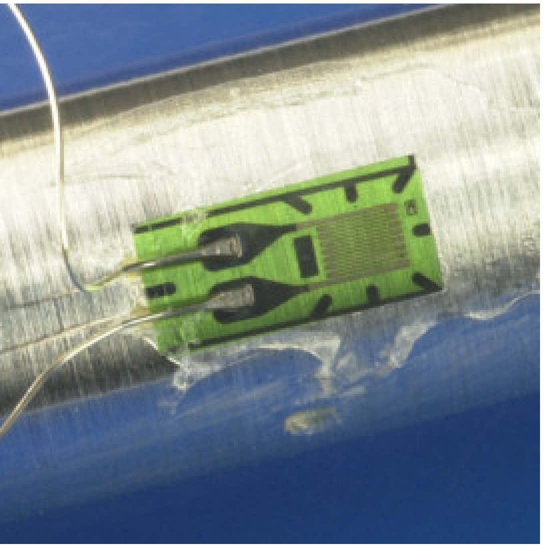
Main Amplifier Unit (Common)

Number of Channels	4, 8 or 16 analog
Overall Amplification per Channel	0, 100, 1000, 10000, Custom (AUX setting)
Max. Output Voltage Range	± 5 Volts
Channel Frequency Response	20±5 Hz to 450±50 Hz, 80 dB/decade
EMG Sensors	DE-2.1 (single differential) or DE-3.1 (double differential)
System Noise (R.T.I.)	<1.2 μV(rms) for the specified bandwidth
Channel Output Isolation	3750 V(rms) @ 60 Hz for 60 sec.
Output Signal Connectors	4, 8, or 16 BNC connectors 68-pin 0.05" pitch high-density D-Sub male
Signal Quality Check	Line Frequency Interference (50 or 60 Hz) Channel Saturation Check (± 4.8 V threshold)
Signal Quality Warnings	Yellow LED, Selectable Audio Buzzer
Operating Temperature	15°C to 40°C

f_s » 900 Hz



Example: strain gauge amplifier



Specifications

Voltage Specifications:
- Electrical input: ±10mV, ±20mV, ±30mV, ±50mV, ±100mV
- Excitation voltage: 1 ~10Vdc (20mA max.)

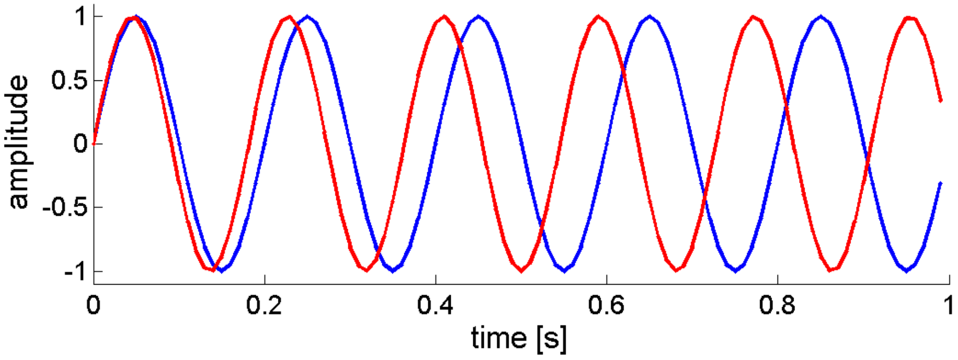
Voltage output:
- Bipolar: ±5V, ±10V
- Unipolar: 0~5V, 0~10V
- Output impedance: <50Ω

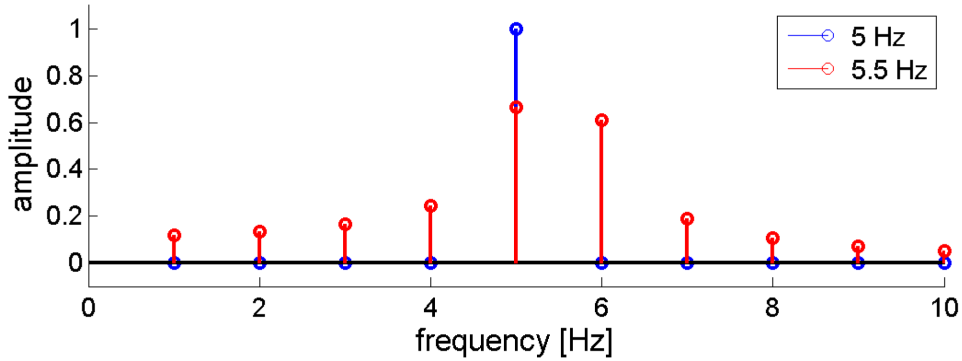
Current output:
- Current: 0~20mA
- Current load resistor:0~500Ω (Source)

General
- Three-way isolation: 1000 Vdc
- Accuracy: ±0.1% of full range
- Bandwidth: 600Hz (typical)@-3dB
- Operation temperature range:-25°C~75°C
- Storage temperature range:-30°C~85°C

Supply Voltage
- Input Range: 10~30Vdc
- Consumption: 1.44W (voltage output)
  1.74W (current output)

Example leakage



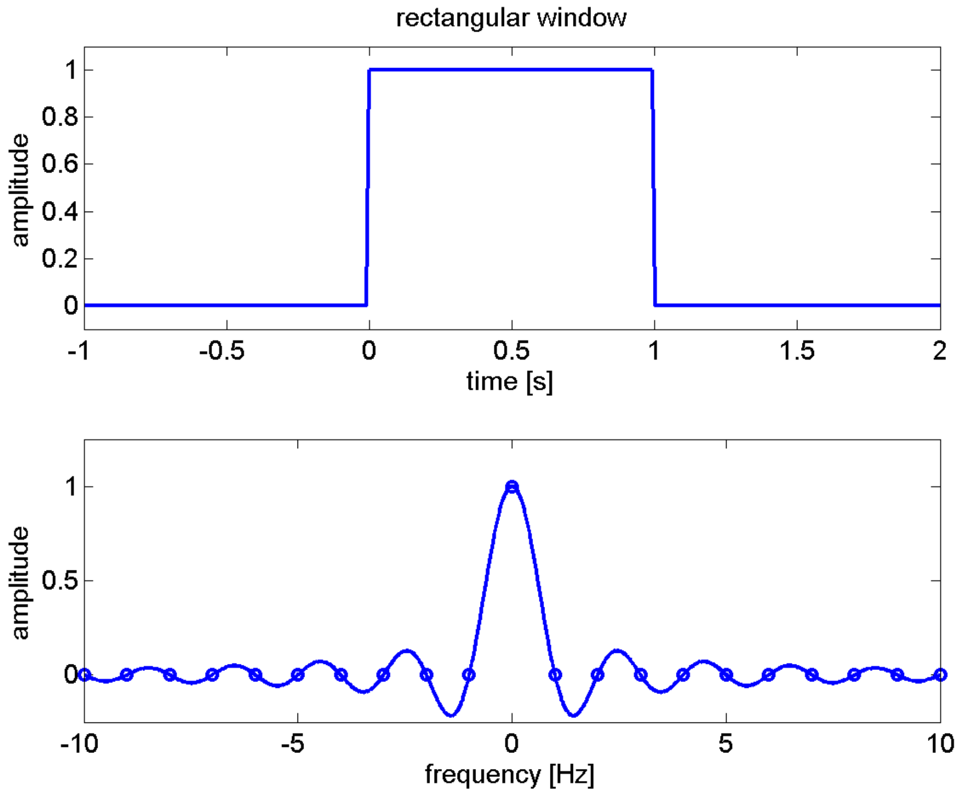


- Matlab: Lec4_LeakageDemo.m
- 1 second observation => resolution: 1Hz
- 5 Hz and 5.5 Hz sine
- Only frequencies with integer number of period are correctly observed!
- Other frequencies will 'leak' to neighboring frequencies

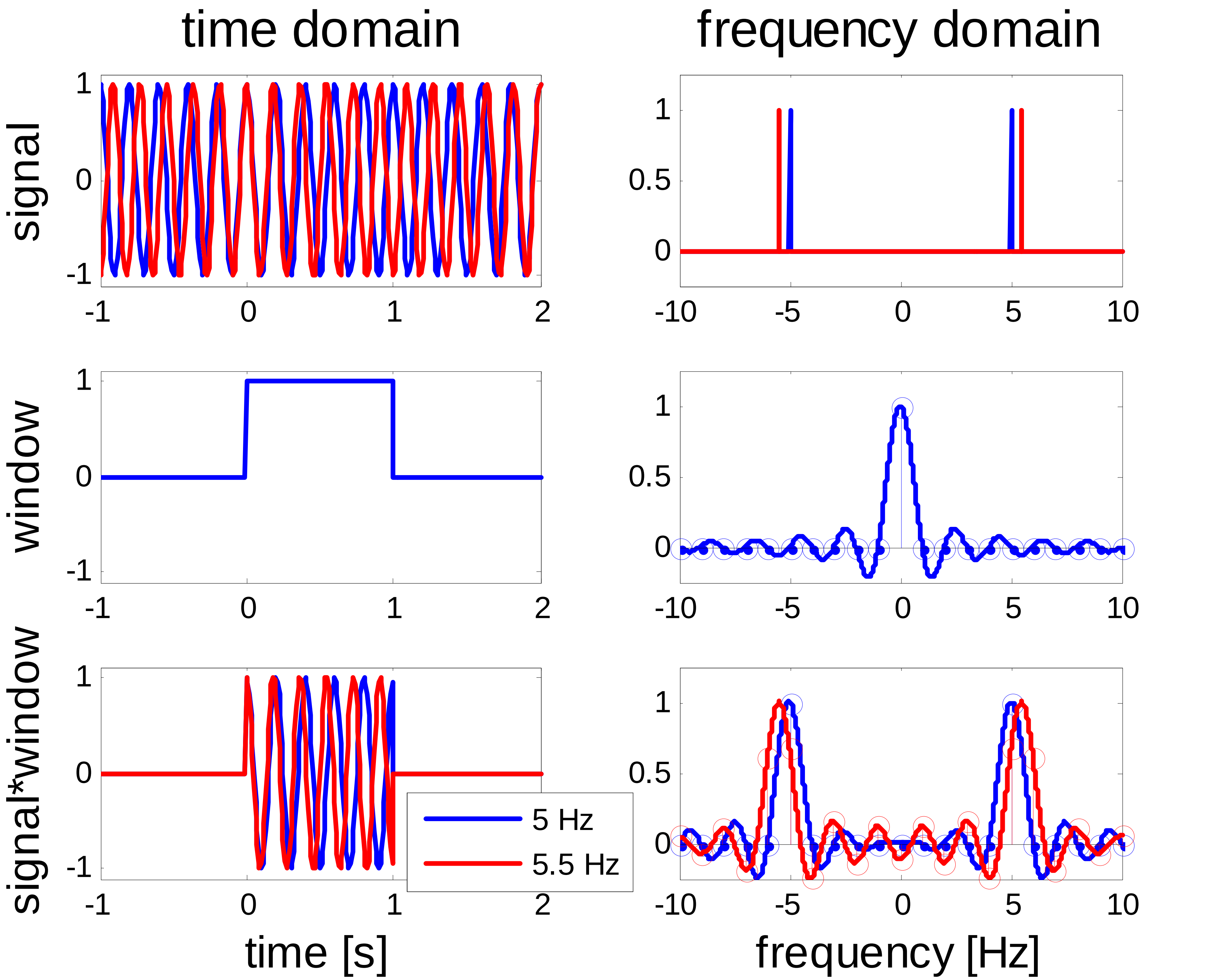
Effect of observation time

- Signal is infinite and a finite observation is made
  - Theoretical concept: signal is multiplied with a window
    - window(t) = 1 within observation time
    - window(t) = 0 elsewhere
  - What is effect of this window (‘windowing’)?

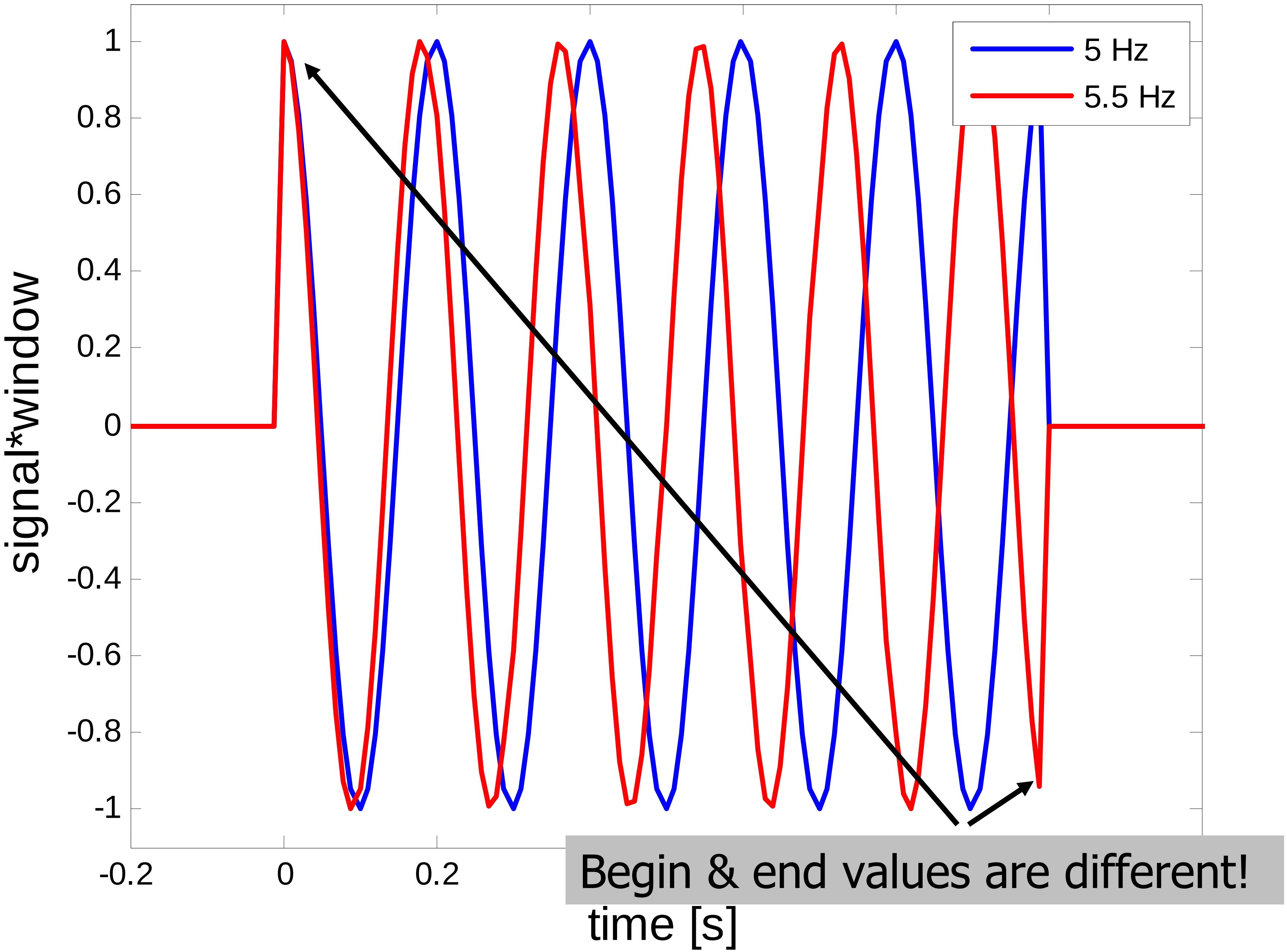
Window and it's Fourier transform

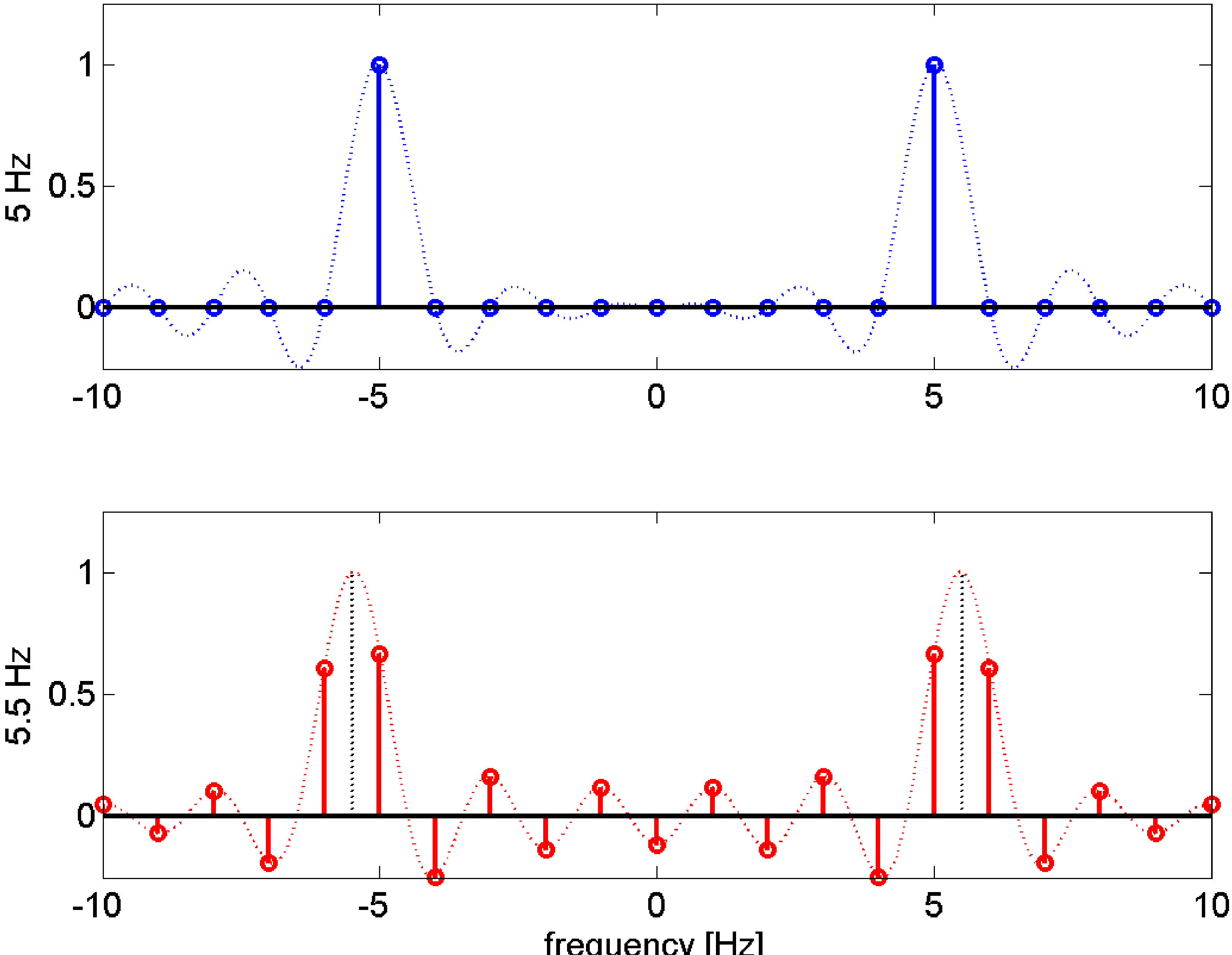


- Note: multiplication in time-domain is convolution in frequency-domain (and vice versa)
- The sharp transitions in the (time) window introduces considerable side-lobs in frequency domain
- Matlab: Lec4_TheorieLeakage.m Lec4_TheorieLeakageHanning.m



time domain
frequency domain
signal
window
signal*window
5 Hz
5.5 Hz
time [s]
frequency [Hz]


5 Hz
5.5 Hz
signal*window
time [s]
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-0.2
0
0.2
Begin & end values are different!



5 Hz
1
0.5
0
-10
-5
0
5
10
5.5 Hz
1
0.5
0
-10
-5
0
5
10
frequency [Hz]

How to reduce leakage

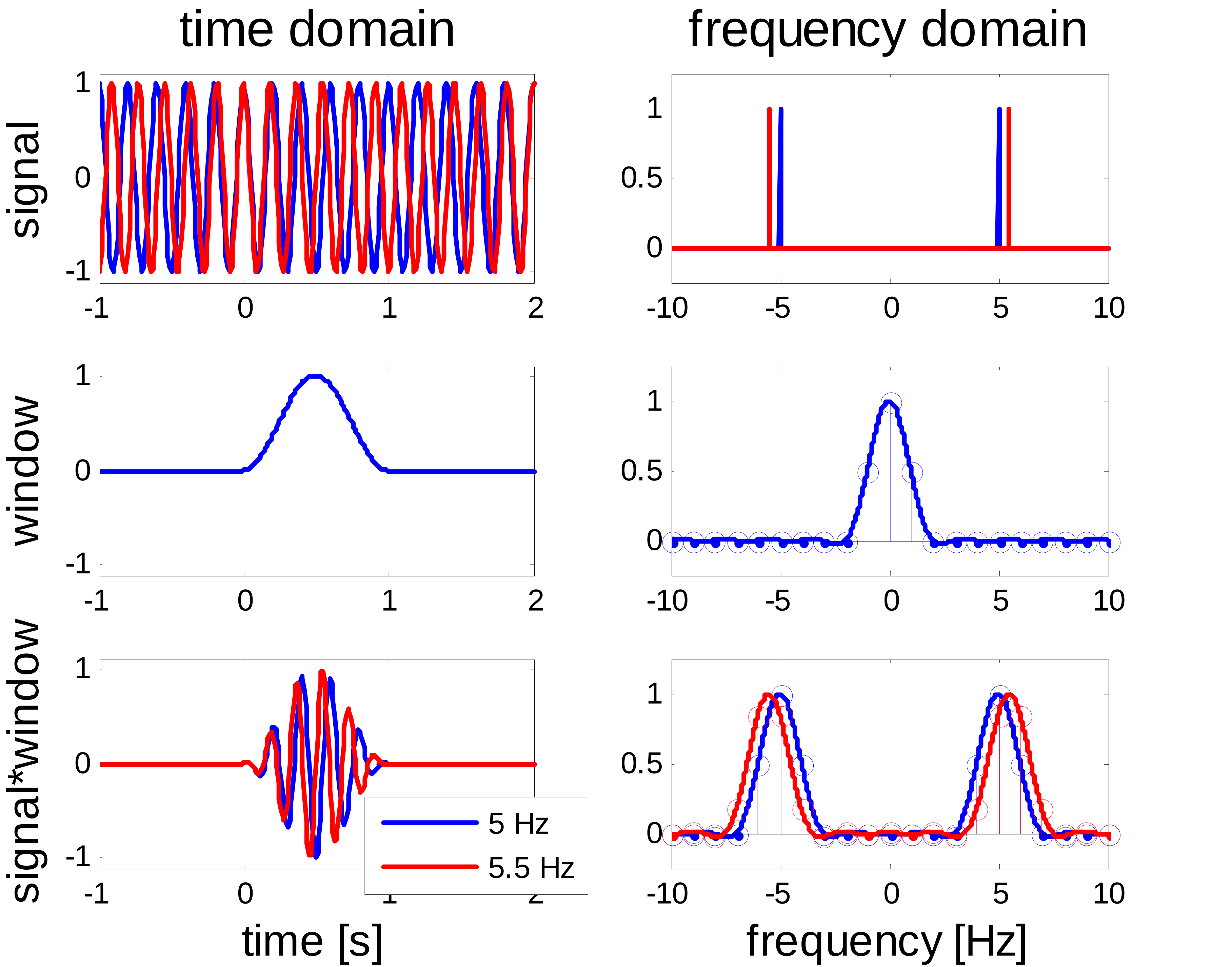
- No window = rectangular window
- A rectangular window has substantial side-lobs

=> premultiply signals with a window before FFT

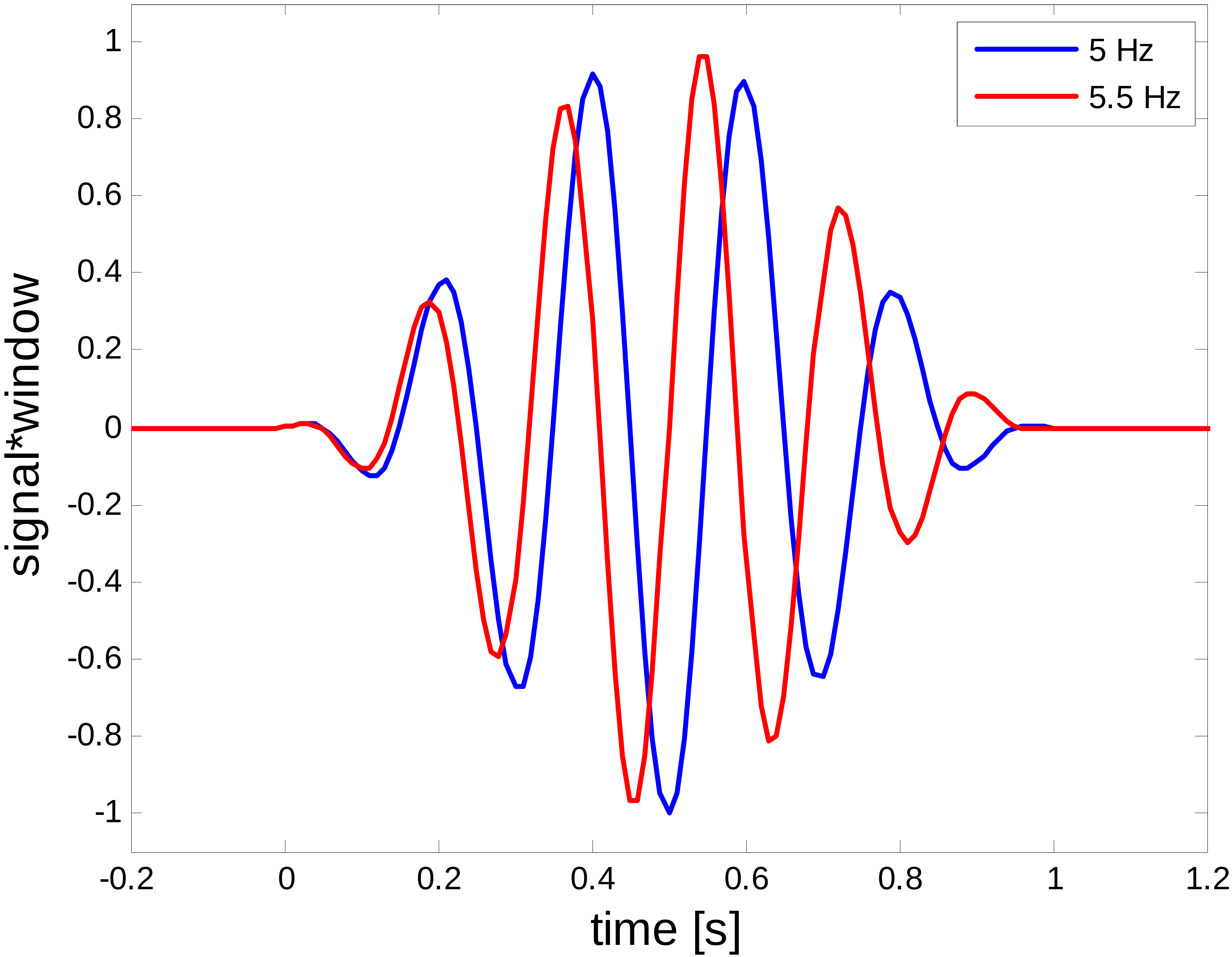
- Hanning window (cosine window: $w=0.5\ (1+\cos\pi t/T)$

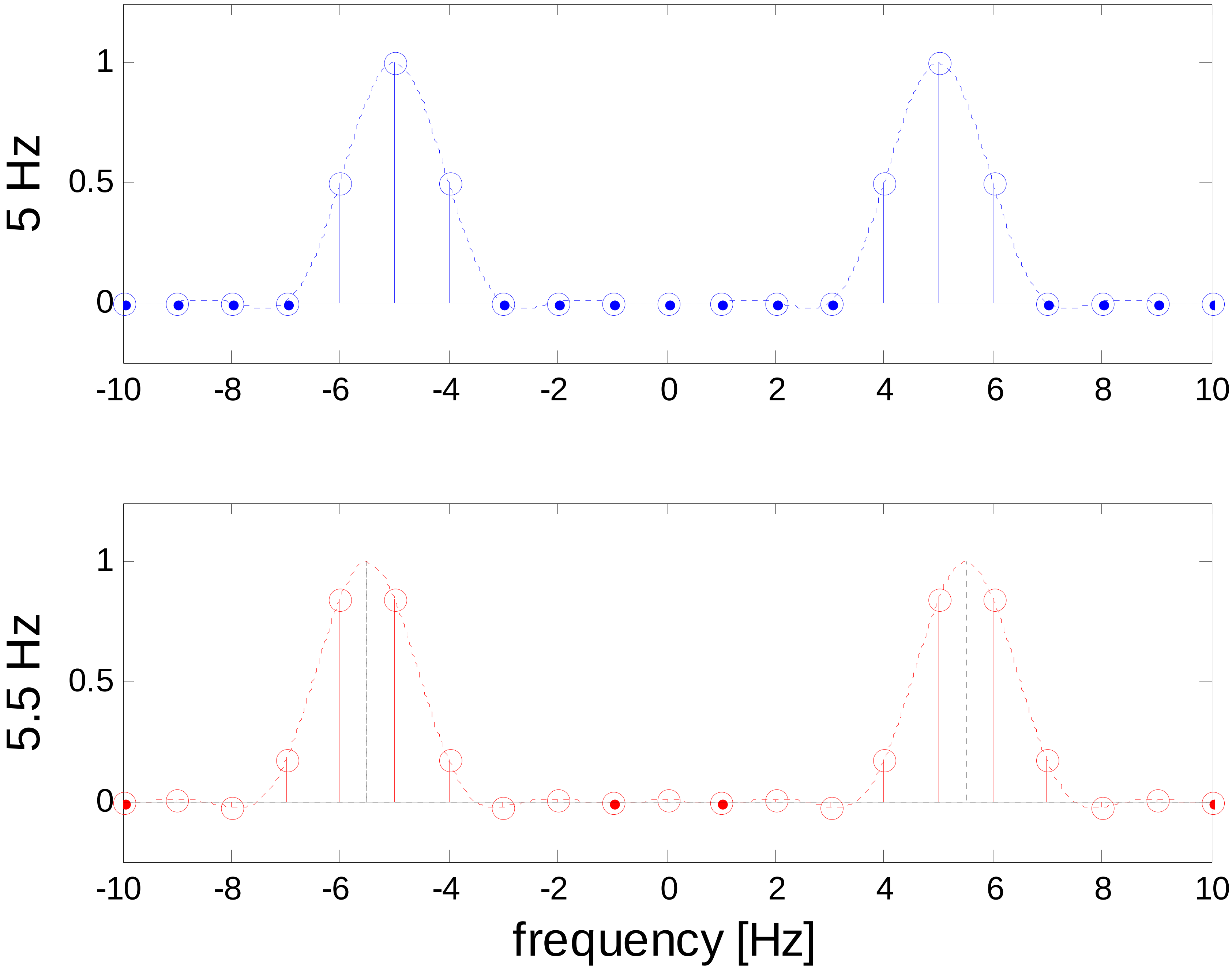
Note:

- 1. This method is also called tapering
- 2. Do not to confuse with windowing of the covariance (which is used to reduce the variance of the spectral estimator based on FFT of covariance, see Lecture 3)



time domain
frequency domain
signal
window
signal*window
5 Hz
5.5 Hz
time [s]
frequency [Hz]


5 Hz
5.5 Hz
signal*window
time [s]
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-0.2
0
0.2
0.4
0.6
0.8
1
1.2


5 Hz
5.5 Hz
1
0.5
0
-10
-8
-6
-4
-2
0
2
4
6
8
10
frequency [Hz]

Remarks windowing

- Shown examples are extremes: methods are developed for ‘noise’ signals, and not a single sine
- Numerous windows exist
  With windowing the begin & end effects are tempered, Hanning window is the most used window
- Premultiplying with window reduces side-lobs but introduces other artifact
  => there is always a trade-off between advantages and disadvantages!

How to prevent leakage

- Can we make ‘leakage’ free signals?
  (and consequently do not have to apply windowing to reduce leakage!)

- White noise contains all frequencies, so leakage will always occur (at least in theory)

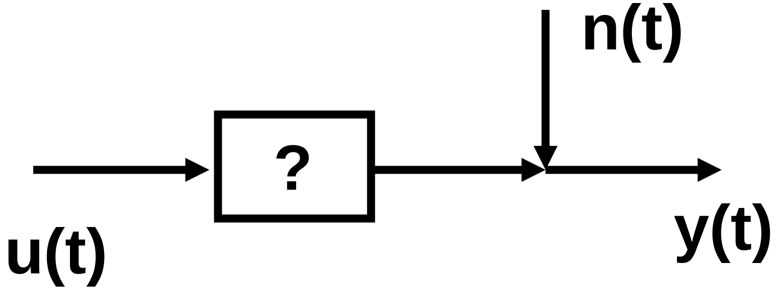
Signal-to-Noise ratio

- SNR: ratio between signal power and noise power
- Definition:

$$SNR = \frac{P_{signal}}{P_{noise}} = \left(\frac{A_{signal}}{A_{noise}} \right)^2$$

$$SNR(dB) = 10\,{}^{10}\log\left(\frac{P_{signal}}{P_{noise}} \right) = 20\,{}^{10}\log\left(\frac{A_{signal}}{A_{noise}} \right)$$

Signal-to-Noise Ratio (SNR)



- Improve SNR by increasing signal (u), or decreasing noise (n)

- Increase signal: turn up the volume!
- Decrease noise: average (in time or frequency)

Ideal perturbation

Properties

- Persistently exciting
- Introducing no bias and variance

Additional

- Long enough to gain sufficient frequency resolution
- Short enough to limit measurement time (Humans: fatigue, attention, etc)
- In case of human subjects: unpredictable

White noise vs. colored noise

- White noise
  - Leakage and aliasing, bad SNR

- Colored noise (=filtered white noise)
  - Improved SNR, no aliasing, leakage not solved.
  - Note that discrete noise is always ‘colored’ as result from sample frequency

Example colored noise: boost the signal

- Matlab: Lec4_ColoredNoise.m

- For most systems the input amplitude is restricted!

- If white and colored noise have the same variance (or the same maximum amplitude)
- The power in colored noise is concentrated in a limited number of frequencies
- => the power per frequency will be higher (Parseval!)
- => better SNR (within the bandwidth)

Improving the estimate 1

Bias (structural errors):

- Main causes
  - finite observation of stochastic input (leakage!)
  - frequency averaging
- Cure
  - application of 'leakage free' deterministic perturbation signals
  - moderate frequency averaging
  - apply method that do not need frequency averaging

Improving the estimate 2

Variance (random errors):

- Main causes
  - noise
- Cure
  - improve SNR
  - averaging (time or frequency domain)

Multisine signals

- With a limited observation time only the frequencies with an integer number of period can be ‘seen’ by spectral estimators.
- Idea: construct signal with all frequencies with integer number op period
- => Multisine signals
- Advantages:
  - no leakage, no aliasing
  - better SNR compared to white noise

Example multisine signal

- Matlab: Lec4_crest_example.m

- If white noise and multisine have same variance

- The power in multisine is concentrated in a limited number of frequencies
- => the power per frequency is higher in multisine (Parseval!)
- => better SNR
- => no leakage, no aliasing

Cresting

- An normal uncrested multisine signal (as white noise) has many outliers
- Probability density function is Gaussian (matlab)

- ‘Trick’ of cresting: reduces the variance of the multisine signal, by removing outliers: cresting
- Probability density function is altered!

- Crest factor = $\max|x(t)| / \sigma_x$

- Even more improved SNR

More advanced multisine tricks

- Basically two possibilities to improve multisine

- Reduce number of frequencies
  - Power per frequency can be increased, better SNR
  - Example: linear frequency spacing
  - Example: (quasi-)logarithmic frequency spacing
    - => Bode diagram has logarithmic frequency axis!

- Shape of the gain per frequency
  - shape input signal such that output signal becomes flat
    - => ideal case for white noise disturbance on the output
  - shape input signal such that output signal has same shape as output noise

Matlab demo

- Lec4_PeriodicDemo.m

Reducing variance

- Average in time
  - Losing frequency resolution
  - Improves SNR
- Average adjacent frequencies
  - Can introduce bias
- Average in frequency domain
  - Multiple realizations or chop in multiple segments. Calculate spectral densities for each segment and average spectral densities over the segments ('Welch' method)
  - Losing frequency resolution
  - Improves SNR

Summary

- Sources for error in (spectral) estimators:
  - Aliasing ($f > f_s/2$)
  - Leakage ($f \neq k\Delta f$ with integer k)
  - Low signal-to-noise ratio (SNR)

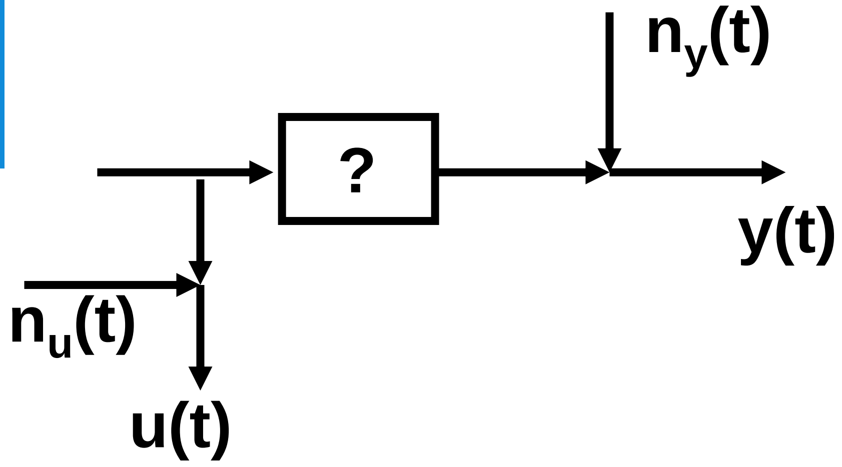
- Multisine signals
  - No aliasing, no leakage
  - Signal can be ‘shaped’ in frequency domain
  - Cresting further improves the SNR

Continuous and transient perturbations

- Transient perturbations
  - Impulses
  - Steps
  - Ramps

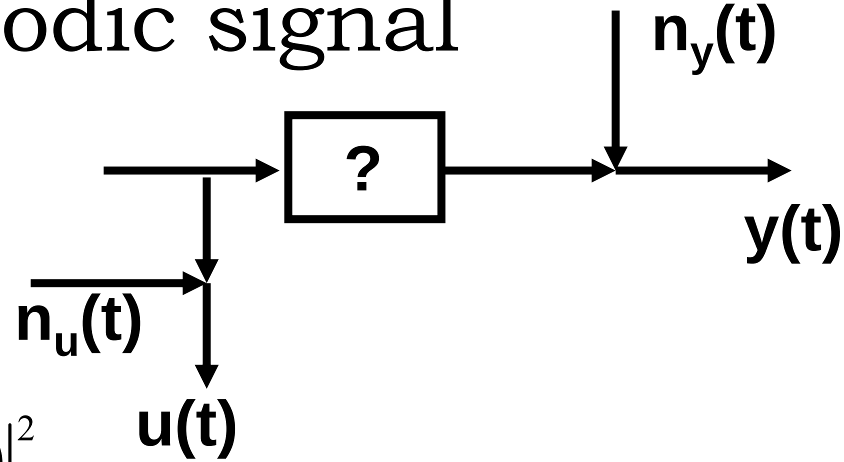
- Continuous perturbations
  - Random
    - White noise
    - Colored noise
  - Periodic
    - Sinusoids
    - Multisines
    - Binary noise (not discussed: switches randomly between two values)

FRF measurements with multiple periods of a periodic signal



- Periodic signal:
  - N samples per period
  - M periods
- Noise (**n_u, n_y**) on measured input **u(t)** and output **y(t)**

FRF measurements with multiple periods of a periodic signal



- Sample mean (in freq domain)

$$U(f) = \frac{1}{M}\sum_{l=1}^{M} U^{[l]}(f)$$

$$S_{UU}(f) = \frac{1}{N} U^*(f) U(f) = \frac{1}{N}\left|U(f)\right|^2$$

- Sample (co-)variance (in freq domain)

$$\sigma_U{}^2(f) = \frac{1}{N(M-1)}\sum_{l=1}^{M}\left|U^{[l]}(f) - U(f)\right|^2$$

$$\sigma_{UY}{}^2(f) = \frac{1}{N(M-1)}\sum_{l=1}^{M}\left(U^{[l]}(f) - U(f)\right)^* \left(Y^{[l]}(f) - Y(f)\right)$$

FRF measurements with multiple periods of a periodic signal

- Periodic signal
  - N samples per period
  - M periods

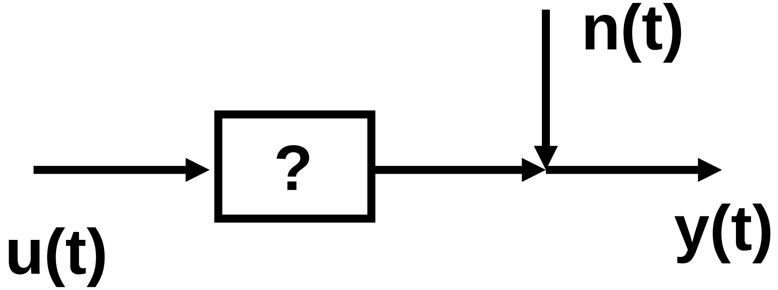
- Frequency response function (FRF)

$$H(f) = \frac{Y(f)}{U(f)}$$

- Variance FRF

$$\sigma_H{}^2(f) = \frac{1}{M}\left|H(f)\right|^2 \left(\frac{\sigma_Y{}^2(f)}{S_{YY}(f)} + \frac{\sigma_U{}^2(f)}{S_{UU}(f)} - 2re\left(\frac{\sigma_{UY}{}^2(f)}{S_{UY}(f)} \right) \right)$$

FRF measurements with multiple periods of a periodic signal



- No noise on input:
- Variance FRF

$$\sigma_H^{\ 2}(f) = \frac{1}{M} \left| H(f) \right|^2 \left(\frac{\sigma_Y^{\ 2}(f)}{S_{YY}(f)} \right)$$

Example multiple periods

- Lec4_AveragePeriods.m

- Assume signal *u* which is contaminated with noise *n*

$$u(t) = x(t) + n(t)$$

- With multiple periods of periodic signal:

$$X(f) \approx U(f) = \frac{1}{M}\sum_{l=1}^{M} U^{[l]}(f)$$

$$S_{nn}(f) \approx \sigma_U{}^2(f) = \frac{1}{N(M-1)}\sum_{l=1}^{M} \left| U^{[l]}(f) - U(f) \right|^2$$

  - Sample variance (in freq domain) is (approximately) equal to the auto-spectral density of the noise!

Relevant Book Chapters

- Pintelon and Schoukens, System identification, a frequency domain approach

- Lecture 4:
  - Leakage and windowing/tapering: SP 2.2.2 & 2.2.3
  - Multisine signals: SP 2.3-2.6, 4.1-4.3

- Note that S&P use different scaling for the DFT (and order for S_{uu})
  - This course (and most used, o.a. W&K, Matlab):

$$U(f) = \sum_{t=1}^{N} u(t) e^{-j2\pi \frac{ft}{N}}; \quad S_{uu} = \frac{1}{N} U^*(f) U(f)$$

  - S&P:

$$U(f) = \frac{1}{\sqrt{N}} \sum_{t=1}^{N} u(t) e^{-j2\pi \frac{ft}{N}}; \quad S_{uu} = U(f) U^*(f)$$

Readings

- Book Westwick & Kearney
  - Chapter 1, all (lecture 1)
  - Chapter 2, sec. 2.1 – 2.3.4 (lecture 1+2)
  - Chapter 3, sec. 3.1 – 3.2 (lecture 2)
  - Chapter 5, sec. 5.1 – 5.3 (lecture 3)

- Book Pintelon & Schoukens
  - Chapter 1, sec 1.1 – 1.4 (optional, lecture 1)
  - Chapter 2, all (lecture 4)
  - Chapter 4, all (lecture 4)

- Articles
  - de Vlugt et al. (lecture 5)

Next week: lecture 5

- Up to now:
  - Single input – single output (SISO) systems
    - Frequency response function (FRF)
    - Coherence

- Next week:
  - Open-loop and closed-loop
  - Multi input – multi output (MIMO) systems
    - MIMO frequency response function
    - Multiple coherence: is an output linearly related with the inputs
    - Partial coherence: is one output linearly related with one input
    - Open-loop and closed-loop