MEEN 459/659 Notes 6

A Brief Introduction to the Discrete Fourier Transform and the Evaluation of System Transfer Functions

Original from Dr. Joe-Yong Kim (ME 459/659), modified by Dr. Luis San Andrés (MEEN 617, Jan 2013, 2019).

Consult free resources from commercial vendors of precision instruments

The Discrete Fourier Transform

The **Fourier Transform (FT)** and its **inverse FT** are (continuous functions) defined as

$$F_{(\omega)} = \int_{-\infty}^{\infty} f_{(t)} e^{-i2\pi\omega t} dt, \tag{1}$$

$$f_{(t)} = \frac{1}{2\pi} \int_{-\infty}^{\infty} F_{(\omega)} e^{i2\pi\omega t} d\omega \tag{2}$$

Above note the integrals are evaluated over infinite long time (intervals?).

Consider the set $\{x_n\}_{n=0,1,\dots,N-1}$ **recorded** at discrete times $\{t_n\} = t_0, t_1 = t_0 + \Delta t, t_2 = t_0 + 2\Delta t, \dots, t_{N-1} = t_0 + \Delta t(N-1)$, where N is the number of samples acquired the elapsed time for recording is $T=(N-1)\Delta t$.

The **Discrete Fourier Transform (DFT)** of a spatially or **time** **sampled** series x_n is

$$X_m = \sum_{n=0}^{N-1} x_n \, e^{\left(-i2\pi \frac{mn}{N}\right)}, \quad m = 0, \dots, N-1. \tag{3}$$

and the **inverse DFT** is

$$x_n = \frac{1}{N} \sum_{m=0}^{N-1} X_m \, e^{\left(i2\pi \frac{mn}{N}\right)}, \quad n = 0, \dots, N-1. \tag{4}$$

The vector $\{X_m\}_{m=0,\dots,N-1} = (a_m + i\, b_m)$ is **complex**.

Note the DFT and its inverse are the **discrete form of a truncated FT.**

Presently, the DFT and inverse DFT can be calculated fast and efficiently by using various Fast Fourier Transform (FFT) algorithms. (e.g., the “fft” command in Matlab® or MATCAD®)

For real x, the **DFT** shows that, $X_0 = X^*_{N-1},\ X_1 = X^*_{N-2}, \ldots, X_2 = X^*_{N-3}, \ldots$ where (*) denotes the complex conjugate, $(a_m - i\,b_m)$.

In practice, software usually delivers a vector of ½ N values (shifted), i.e.,

$$\widehat{X}_0 = X_{(N-1)-k},\ \widehat{X}_1 = X_{(N-1)-k+1}; \ldots; \ \widehat{X}_{k-1} = X_{N-2};\ \widehat{X}_k = X_{N-1}; \quad k = \frac{N}{2} \tag{5}$$

The **maximum frequency** (f_{max}) of the DFT of a time series $\{x_n\}_{n=0,\,N-1}$ sampled at Δt satisfies the Nyquist Sampling Theorem, i.e.,

$$f_{\max} \le \frac{f_{\text{sample}}}{2} = \frac{1}{2\,\Delta t}. \tag{6}$$

There are k=½N data points in the frequency spectrum (complex numbers). Since the maximum frequency is $f_{\max} = f_{\text{sample}}/2$, the frequency resolution (Δf) equals

$$\Delta f = \frac{f_{\text{sample}}}{N} = \frac{1}{N\,\Delta t} = \frac{1}{T} = \frac{1}{\text{time record length}}. \tag{7}$$

Hence, the longer T is (the more samples N), the smaller Δf is; while the maximum frequency is set by the sampling rate.

Example 1

Figure 1(a) below shows $x_{(t)}=1\ \sin(\omega t)$, with $\omega=2\pi f$, f=22 Hz, sampled at 100 Hz (samples/s) or Δt=0.01 s, and the number of points is N=256 (T_{max}=2.55 s). Note that Δt << 0.045 s, the period of the f=22 Hz wave.

Figure 1(b) shows the amplitude of the DFT, $\left|X_m\right|_{m=0,\ldots,N/2-1}$ versus frequency. The maximum frequency in the DFT is f_{max}=50 Hz with a step of $\Delta f = \dfrac{1}{\Delta t\, N}$ =0.391 Hz. The number of frequencies in the DFT is 128. Note the amplitude of the DFT $|X_m|$ shows components at other frequencies than 22 Hz.

The DFT is a collection of k= ½ N complex numbers, i.e., it is a **discrete set (**not continuous). Figure 1(c) graphs the real and imaginary parts of the DFT X_m .

$f_{req} =$ (Hz)

	1
1	0
2	0.391
3	0.781
4	1.172
5	1.563
6	1.953
7	2.344
8	2.734
9	3.125
10	3.516
11	3.906
12	4.297
13	4.688
14	5.078
15	5.469
16	5.859

*

$X =$

	1
1	$4.921\cdot10^{-3}$
2	$9.843\cdot10^{-3}+1.683i\cdot10^{-4}$
3	$9.848\cdot10^{-3}+3.368i\cdot10^{-4}$
4	$9.856\cdot10^{-3}+5.059i\cdot10^{-4}$
5	$9.867\cdot10^{-3}+6.758i\cdot10^{-4}$
6	$9.882\cdot10^{-3}+8.468i\cdot10^{-4}$
7	$9.9\cdot10^{-3}+1.019i\cdot10^{-3}$
8	$9.921\cdot10^{-3}+1.193i\cdot10^{-3}$
9	$9.945\cdot10^{-3}+1.369i\cdot10^{-3}$
10	$9.974\cdot10^{-3}+1.548i\cdot10^{-3}$
11	$0.01+1.729i\cdot10^{-3}$
12	$0.01+1.913i\cdot10^{-3}$
13	$0.01+2.101i\cdot10^{-3}$
14	$0.01+2.292i\cdot10^{-3}$
15	$0.01+2.488i\cdot10^{-3}$
16	$0.01+2.688i\cdot10^{-3}$

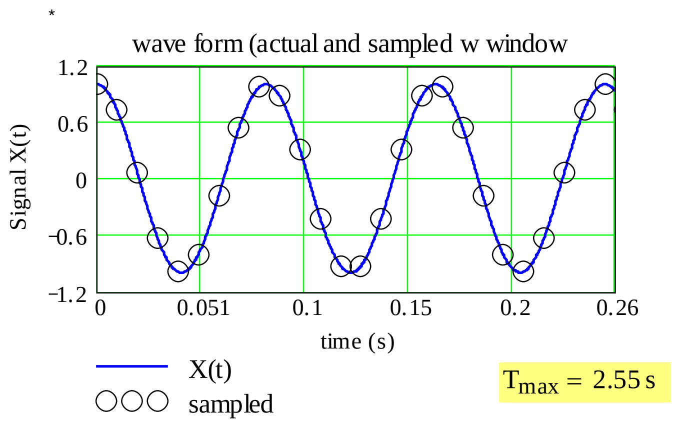
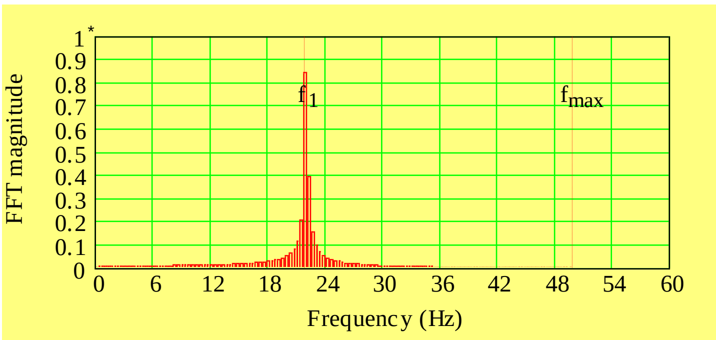


Fig. 1(a): 22 Hz signal sampled at 100 samples/s.



$N_P = 256$

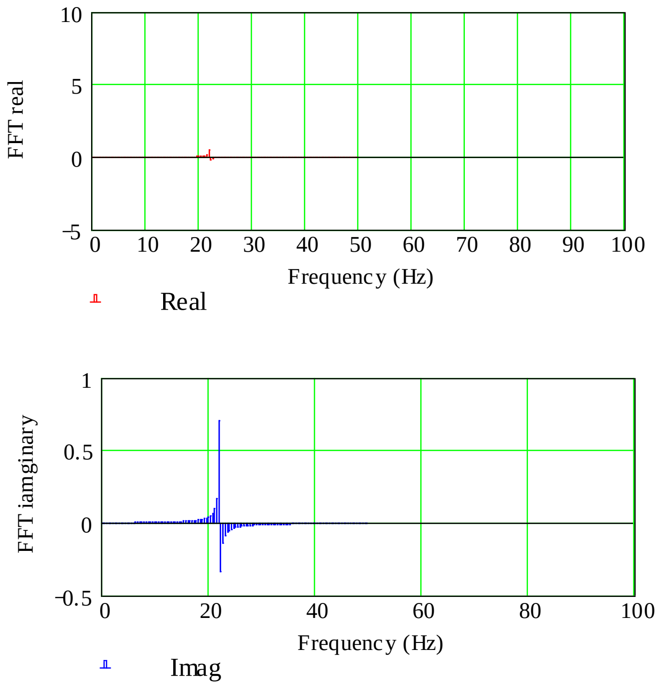
$\Delta f = 0.391$ Hz

$f_{max} = 50$ Hz

$f_1 = 22$ Hz

$max(A) = 0.843$

Fig. 1(b): amplitude of DFT for 22 Hz signal sampled at 100 samples/s.



$\max(Re\_X) = 0.456$

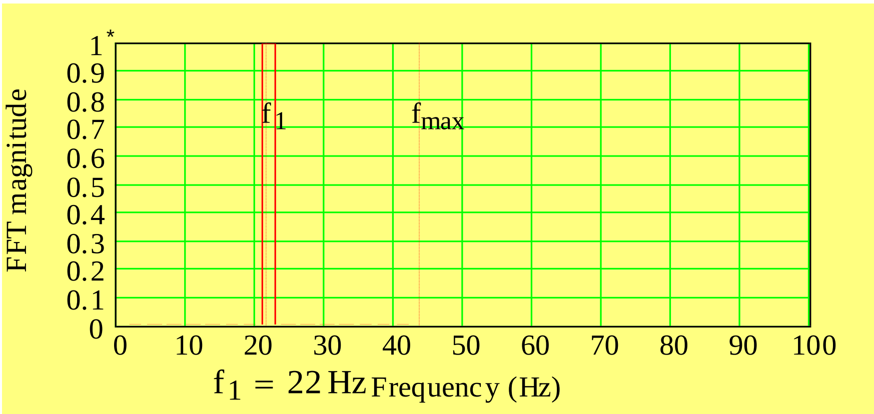
$\min(Re\_X) = -0.206$

$\max(Im\_X) = 0.709$

$\min(Im\_X) = -0.334$

Fig. 1(c): Real and imaginary parts of DFT for 22 Hz signal sampled at 100 samples/s.

The **ideal FFT output** would be a single amplitude *X*=1 at 22 Hz and 0's at all other frequencies. This ideal representation only occurs when sampling at a frequency that is a multiple of the signal frequency, as shown in Fig 1(d) for sampling at 88 Hz.



$\Delta rate = 88\ Hz$

$N_P = 32$

$\Delta f = 2.75\ Hz$

$\max(A) = 1$

Fig. 1(d): amplitude of DFT for 22 Hz signal sampled at 88 samples/s.

$freq^T =$ (Hz)

	1	2	3	4	5	6	7	8	9	10
1	0	2.75	5.5	8.25	11	13.75	16.5	19.25	22	24.75

$X^T =$

	1	2	3	4	5	6	7	8	9	0
1	0	0	0	0	0	0	0	0	1	0

Notes

1) increasing the number of recorded data points N, while keeping the same sampling rate, increases the total time (T) for sampling, but has no impact on the span of the frequency range (f_{max} is the same). Increasing T (recording time) makes Δf to decrease (the frequency resolution increases).

2) increasing the sampling rate (f_{sample}) while keeping N extends the span of the frequency range ($f_{max} = ½\ f_{sample}$), and also increases the frequency step Δf (decreases resolution as it makes Δf larger). Increasing f_{sample}, decreases the total elapsed time for measurement, $T=(N\text{-}1)\Delta t$

The table below verifies the relationships $f_{max} = ½\ f_{sample}$ and $(f_{max} /\Delta f\ ) = k= ½\ N$, where N and f_{sample} are specified (input).

N	f_{sample} (Hz)	f_{max} (Hz)	Δf (Hz)	T (s)
2^5=32	40	20	1.250	0.775
2^6=64	40	20	0.625	1.575
2^7=128	40	20	0.313	3.175
2^6=64	40	20	0.625	1.575
2^6=64	80	40	1.250	0.788
2^6=64	160	80	2.500	0.394

ALIASING

Figure 2(a) shows the same function $x_{(t)}$=1 sin(ωt), with $\omega=2\pi f$, f=22 Hz, sampled at 30 Hz (samples/s) or Δt=0.033 s, and the number of points is $N=2^8$=256 (T_{max}=8.5 s). Note that Δt ~ 0.045s, the period of the 22 Hz wave, while the time step for sampling is 1/30=0.033 s.

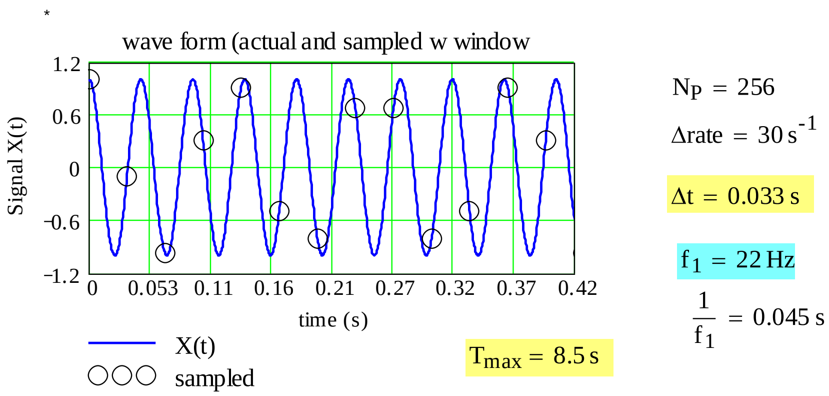


Fig. 2(a): 22 Hz wave sampled at 30 samples/s.

As shown in Fig. 2(b) depicting the amplitude of the DFT, when a 22 Hz sinusoidal signal is sampled at 30 Hz, the sampled data can be misinterpreted as an 8 Hz sinusoidal signal. This is referred to as **aliasing**. Thus, the sampling frequency should be at least **44 samples/s (22 Hz Nyquist)** in order to avoid this problem.

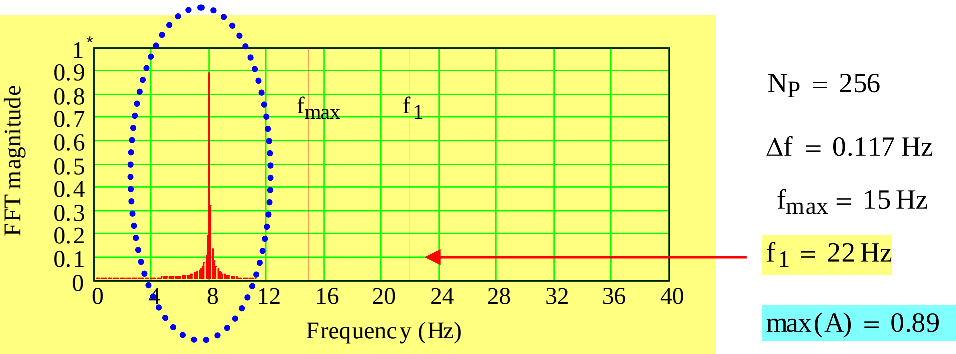


Fig. 2(a): DFT of 22 Hz wave sampled at 30 samples/s.

Leakage

Consider a case where a continuous signal with main frequency f=12 Hz, $f_{(t)}$= 1 cos(2π f t), is sampled at a frequency of f_{sample}=100 samples/s (ΔT=10 ms), and the number of the total sampled data is N = 32, as shown in Fig. 3(a).

Note in Fig. 3(b) the amplitude of the DFT with components at other frequencies than 12 Hz, including 0 frequency.

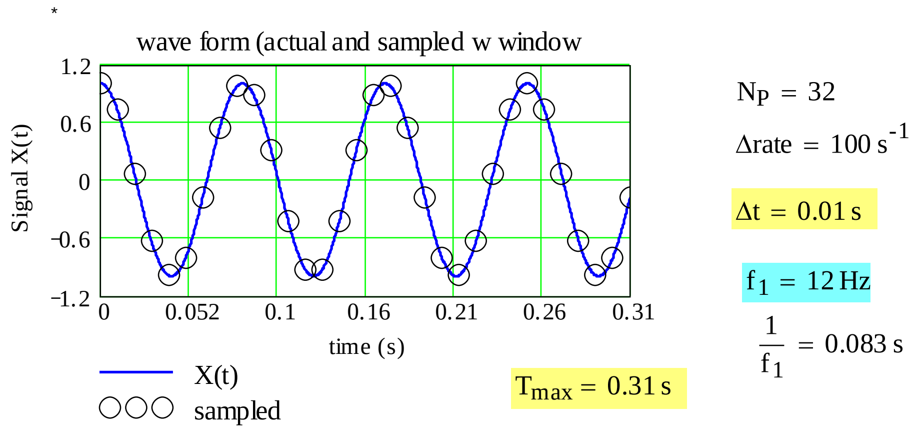


Fig. 3(a): 12 Hz wave sampled at 100 samples/s.

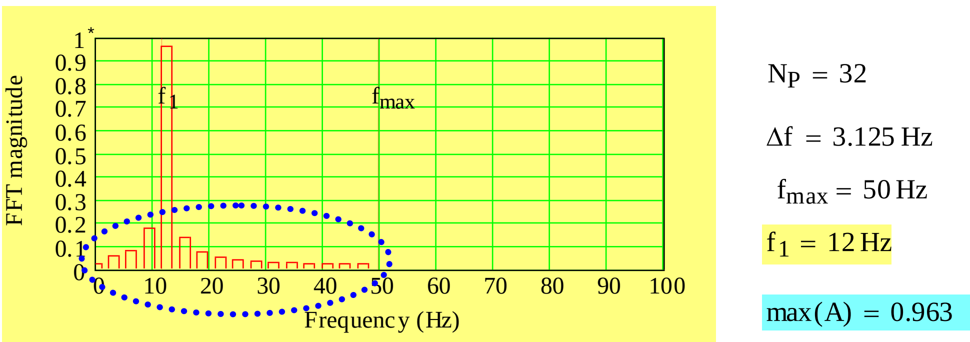


Fig. 3(b): Amplitude of DFT for 12 Hz wave sampled at 100 Hz.

The amplitudes at near zero-frequencies (i.e., the first data points in Fig. 3(b) show **leakage** and is caused by the truncation of the time data. That is, the time data at t = 0 and t = T have non-zero amplitudes, see Fig. 3(a). The graph immediately tells you that the mean value of the function shown is NOT zero.

To reduce the truncation error and leakage effect, a **Hanning window**[1] is introduced. The window is defined as

$$H_m = \frac{1}{2}\left[1 - \cos\left(\frac{2\pi m}{N}\right)\right]. \tag{8}$$

and displayed below in Fig. 4 as

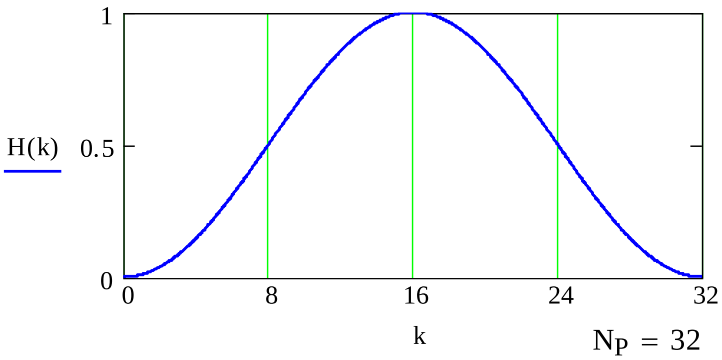


Fig. 4. Hanning window with 32 data points.

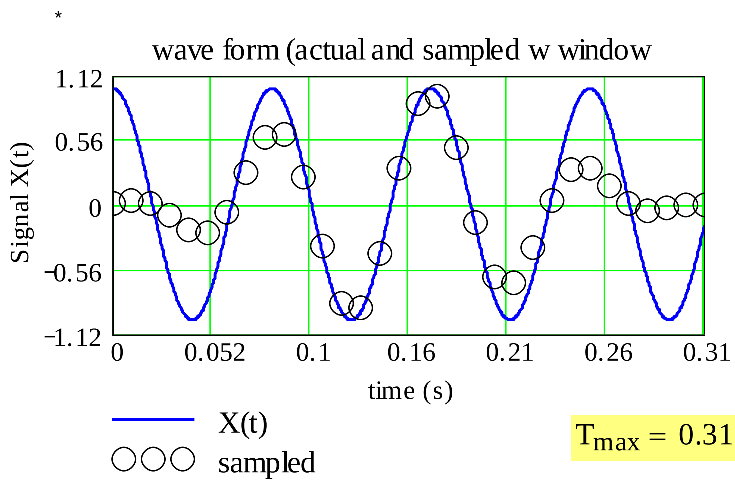
Figure 5 shows the signal data set x_n weighted with the Hanning window. The DFT of a windowed time data is

$$X_m = \sum_{n=0}^{N-1} w_n\, x_n\, e^{\left(-i2\pi\frac{mn}{N}\right)}, \tag{9}$$

where w_n represents the window function. Based on the window function, two constants are defined as

$$\alpha_1 = \sum_{n=1}^{N-1} w_n \text{ and } \alpha_2 = \sum_{n=1}^{N-1} w_n^2 \tag{10}$$

[1] There are many different types of windows or windowing procedures. Refer to a more advanced resource for details on their implementation and accuracy.



$N_P = 32$

$\Delta rate = 100\ s^{-1}$

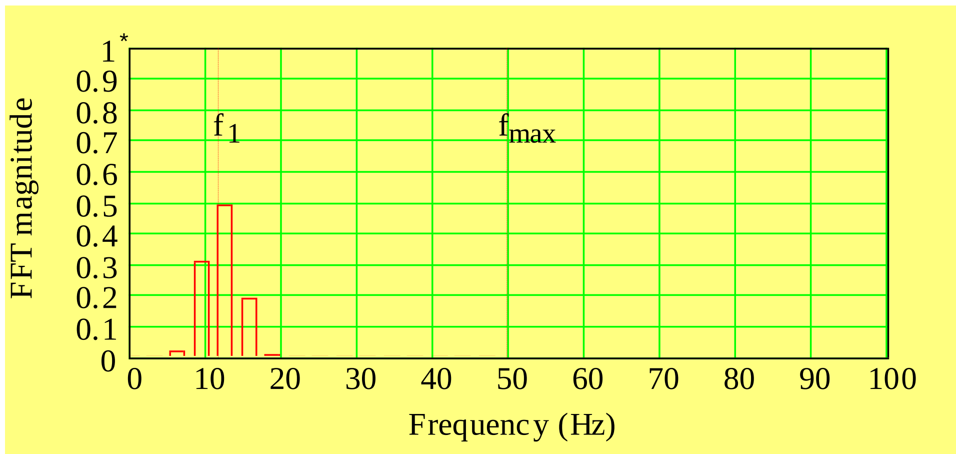
$\Delta t = 0.01\ s$

$f_1 = 12\ Hz$

$T_{max} = 0.31\ s$

Fig. 5: Sampled 12 Hz wave (100 samples/s) with Hanning window.

At $t = 0$ and $t = T$, the amplitude of the signal =0. In the frequency domain, as shown in Fig. 6, the **leakage** of the windowed data is smaller than that for the original data, see Fig. 3(b), although the frequency resolution of the windowed data is lower than the original data (i.e., the peaks of the windowed data become broader than the original data). [Certainly the amplitude at 12 Hz is much smaller than 1]



$N_P = 32$

$\Delta f = 3.125\ Hz$

$f_{max} = 50\ Hz$

$f_1 = 12\ Hz$

$max(A) = 0.492$

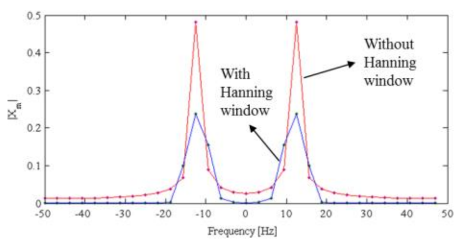


Fig. 6: Amplitude of DFT with Hanning window for 12 Hz wave sampled at 100 Hz.

Spectrum and Spectral Density

All experimental (recorded) data contains noise! **Spectral averaging** is applied to reduce the effects of noise.

The **cross-spectrum** of two signals X and Y is (think of a dot product or projection of one signal onto the other)

$$S_{xy_m} = \frac{2X_m^* Y_m}{\alpha_1^2}, \quad m = 0,1,...,k = \frac{N}{2} \tag{11}$$

where X_m^* is the complex conjugate of X_m and α_1 is a scaling factor

The **auto-spectrum** is also defined as

$$S_{xx_m} = \frac{2X_m^* X_m}{\alpha_1^2} \cdot m = 0,1,...,k = \frac{N}{2} \tag{12}$$

The **cross-spectral density** is defined as

$$\text{CSD}_{xy_m} = \frac{2X_m^* Y_m}{f_{\text{sample}}\alpha_2} \cdot m = 0,1,...,k = \frac{N}{2} \tag{13}$$

The **cross spectral density** is the **cross-spectrum** per unit frequency interval.

Spectral Estimation

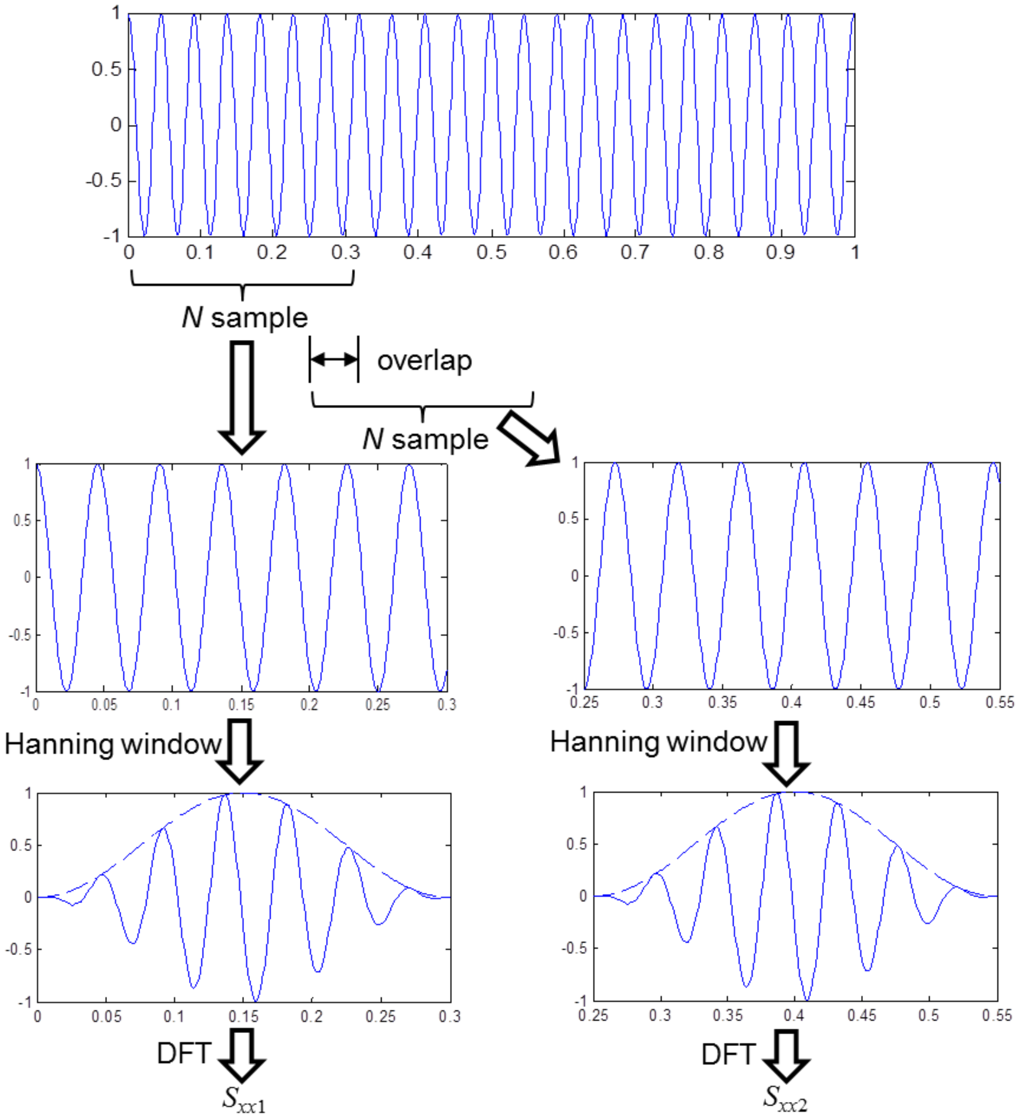


Fig. 7: Averaging process of time data.

Based on the procedure shown in Fig. 7, when the maximum number of averaging is N_a, the spectral averaging process is represented as

$$\bar{S}_{xx} = \frac{1}{N_a}\sum_{m=1}^{N_a} S_{xx(m)} . \tag{14}$$

When the **statistical properties** of a signal do NOT change with respect to time, the signal is referred to as a **"stationary"** signal. Thus, (random) noise effects can be reduced by using a time averaging process, as shown in Fig. 7 and Eq. (14) for any stationary signals.

A useful operation to check when performing multiple (time) averages leads to expected (credible) results is the **coherence function**. (See later these notes).

Transfer Function Estimation

Figure 8 shows a single input and single output (SISO) system with transfer function ***H***.

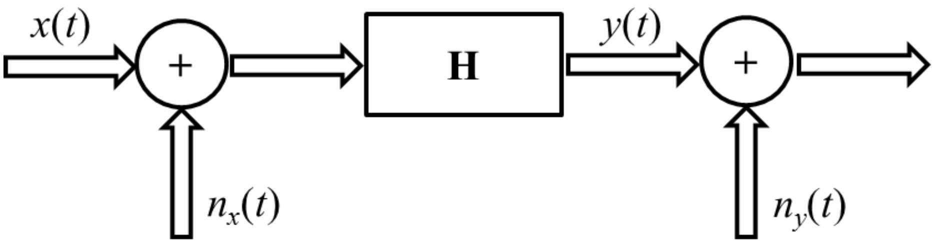


Fig. 8: Depiction of SISO system with transfer function *H*.
x: input, *y*: output, and *n*: noise

In an **ideal case** **without measurement noise**, the transfer function[2] is

$$H = \frac{Y_{(\omega)}}{X_{(\omega)}}. \tag{15}$$

where $X_{(\omega)} = DFT(x_{(t)})$ and $Y_{(\omega)} = DFT(y_{(t)})$.However, when noise[3] components $\boldsymbol{n_x}$ and $\boldsymbol{n_y}$ are present at the input and output of the system, one records the input and output signals as $\bar{x}_{(t)} = x_{(t)} + n_{x_{(t)}}$, $\bar{y}_{(t)} = y_{(t)} + n_{y_{(t)}}$, respectively. Hence, the transfer function becomes

$$H_{(\omega)} = \frac{\bar{Y}_{(\omega)}}{\bar{X}_{(\omega)}} = \frac{Y_{(\omega)} + N_{y(\omega)}}{X_{(\omega)} + N_{x(\omega)}}. \tag{16}$$

Here, the **estimated transfer function *H* is biased due to the noise.** Note that once noise is present in a signal, one cannot know with certainty the actual (true) value of a function, *Y* or *X*, and worse yet *H*.

To **estimate** an **accurate** transfer function, the noise components must be suppressed (or filtered).

Two types of transfer function estimators are introduced. The **first type of estimator** uses **a cross-spectral correlation with respect to the input.**

[2] By function here I mean a discrete function of frequency. That is, both *Y* and *X* (and *H*) have values at specific frequencies, ω_k. A more proper notation should be $X_{(\omega_k)} \rightarrow X_k$, $H_{(\omega_k)} \rightarrow H_k$, etc.

[3] Here ***noise*** is a broad band frequency signal with zero mean (aleatory in character).

$$H_{m1}=\frac{\left(\bar{X}^*\bar{Y}\right)}{\left(\bar{X}^*\bar{X}\right)}=\frac{\left(X+N_x\right)^*\left(Y+N_y\right)}{\left(X+N_x\right)^*\left(X+N_x\right)}=\frac{S_{xy}+S_{xn_y}+S_{n_xy}+S_{n_xn_y}}{S_{xx}+S_{xn_x}+S_{n_xx}+S_{n_xn_x}}. \quad (17)$$

When the input $x_{(t)}$ and output $y_{(t)}$ are not correlated with **either** noise (input) n_x and (output) n_y, that is $\left(S_{xn_y}=0, S_{n_xy}=0, S_{xn_x}=0, S_{n_xx}=0\right)$, and further the noises (n_x , n_y) are not correlated to each other $\left(S_{n_xn_y}=0\right)$, the estimator of the transfer function can be simplified, after **taking the time average,** as

$$H_{m1}=\frac{S_{xy}}{S_{xx}+S_{n_xn_x}}.=\frac{\left(\bar{X}^*\bar{Y}\right)}{\left(\bar{X}^*\bar{X}\right)}=\frac{\left(X+N_x\right)^*\left(Y+N_y\right)}{\left(X+N_x\right)^*\left(X+N_x\right)}\sim\frac{S_{xy}}{S_{xx}+S_{n_xn_x}} \quad (18)$$

This first kind of estimator has no bias error when the **uncorrelated** noise is present only in the **output signal (*y*),** i.e., $S_{n_xn_x}\approx 0$. Then, the first type estimator becomes

$$H_{m1}=\frac{S_{xy}}{S_{xx}}. \quad (19)$$

This estimator is good at **anti-resonance frequencies** of a system **where the input signal (*X*) has a large signal to noise ratio (SNR).**

The **second type of estimator** uses the **cross-spectral correlation with respect to the output**

$$H_{m2}=\frac{\left(\bar{Y}^*\bar{Y}\right)}{\left(\bar{Y}^*\bar{X}\right)}=\frac{\left(Y+N_y\right)^*\left(Y+N_y\right)}{\left(Y+N_y\right)^*\left(X+N_x\right)}=\frac{S_{yy}+S_{yn_y}+S_{n_yy}+S_{n_yn_y}}{S_{xy}^*+S_{yn_x}+S_{n_yx}+S_{n_yn_x}}. \quad (20)$$

With uncorrelated $\left(S_{yn_x}=0, S_{n_yy}=0, S_{yn_y}=0, S_{n_yx}=0\right)$ and noises $S_{n_xn_y}=0$, then the 2nd estimator simplifies to

$$H_{m2}=\frac{S_{yy}+S_{n_yn_y}}{S_{xy}^*}. \quad (21)$$

This **estimator has no bias error** if the noise is present only in the **input signal *(x)*; but not the output,** i.e., $S_{n_yn_y}\approx 0$. Thus, the second type estimator becomes

$$H_{m2}=\frac{S_{yy}}{S_{xy}^*}. \quad (22)$$

This estimator is good at resonance frequencies of a system where (in general) the output signal (*Y*) has a large signal to noise ratio (SNR).

About the coherence function

https://en.wikipedia.org/wiki/Coherence_(signal_processing)

The coherence is a statistic function that examines the relation between two signals, $x_{(t)}$: input and $y_{(t)}$: output. The coherence estimates the power transfer between input and output of a linear system. If the signals are **ergodic (random)**, and the system function linear, the **coherence can be used to estimate the causality between the input and output.**

T

he coherence between two signals $x_{(t)}$ and $y_{(t)}$ is a real-valued function

$$C_{xy_m} = \frac{\left|\bar{S}_{xy_m}\right|^2}{\bar{S}_{xx_m}\bar{S}_{yy_m}} \tag{23}$$

where S_{xy} is the (averaged) cross-spectral density between x and y, and S_{xx} and S_{yy} are the (averaged) auto-spectral density of x and y, respectively (see Eqs. 11-14). The magnitude of the spectral density is denoted as $|S|$.

The **coherence** always satisfies $0 \le C_{xy_m} \le 1$ and estimates the extent to which $y_{(t)}$ may be predicted from $x_{(t)}$ by an optimum linear least squares function.

If the *coherence* is less than one but greater than zero it is an indication that either noise is entering the measurements, that the assumed function relating $x_{(t)}$ and $y_{(t)}$ is not linear, or that **$y_{(t)}$ is producing output due to input $x_{(t)}$ as well as other inputs (including noise).**

If the *coherence* = zero → $x_{(t)}$ and $y_{(t)}$ are completely unrelated.

If the *coherence* = 1 → $x_{(t)}$ and $y_{(t)}$ are completely correlated, **the output *y* is due to the input *x***.

In **vibration measurements,** the larger the number of independent tests conducted $(N_a \to \infty)$ (and averaged) will produce **better** coherence values as the averaging process reduces (filters) noise, for example.

Do NOT use or interepret transfer function estimations in frequency ranges with low values of coherence $C_{xy_m} <<< 1$.

More on estimations of transfer functions for actual physical systems (experimental data) will follow as the class progresses.

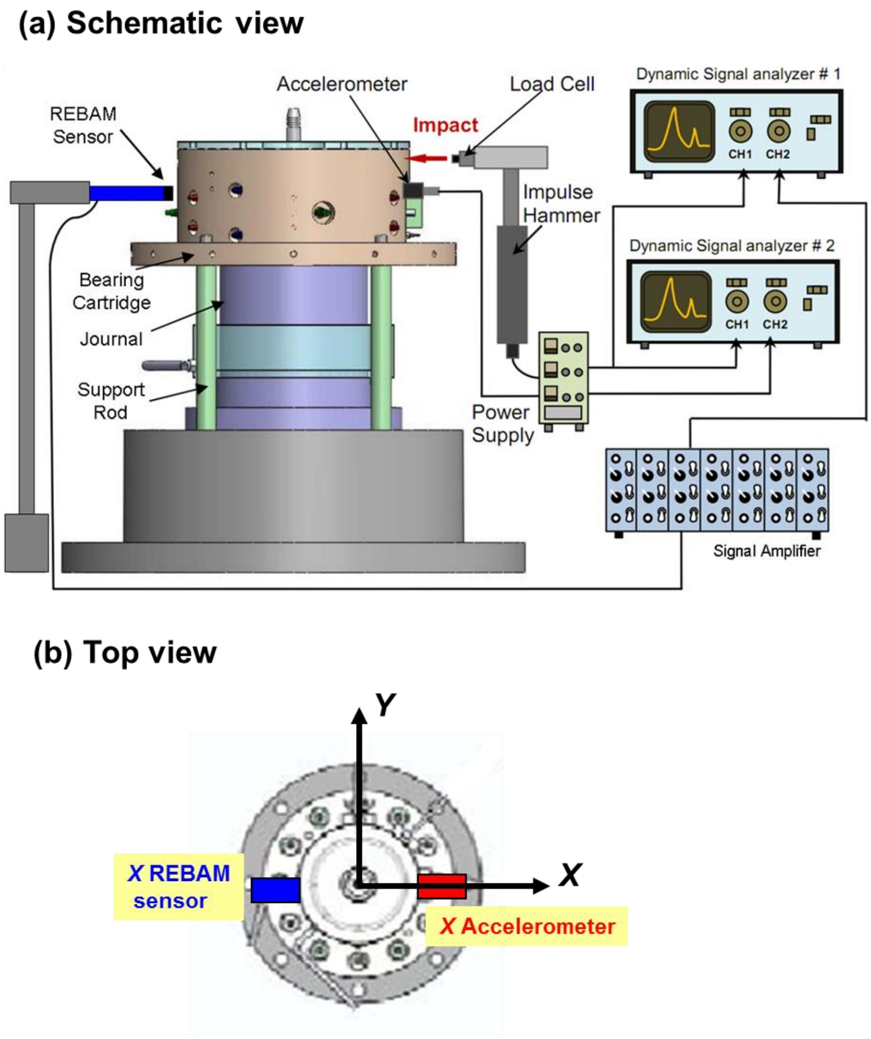
Final notes:

A word of wisdom/caution

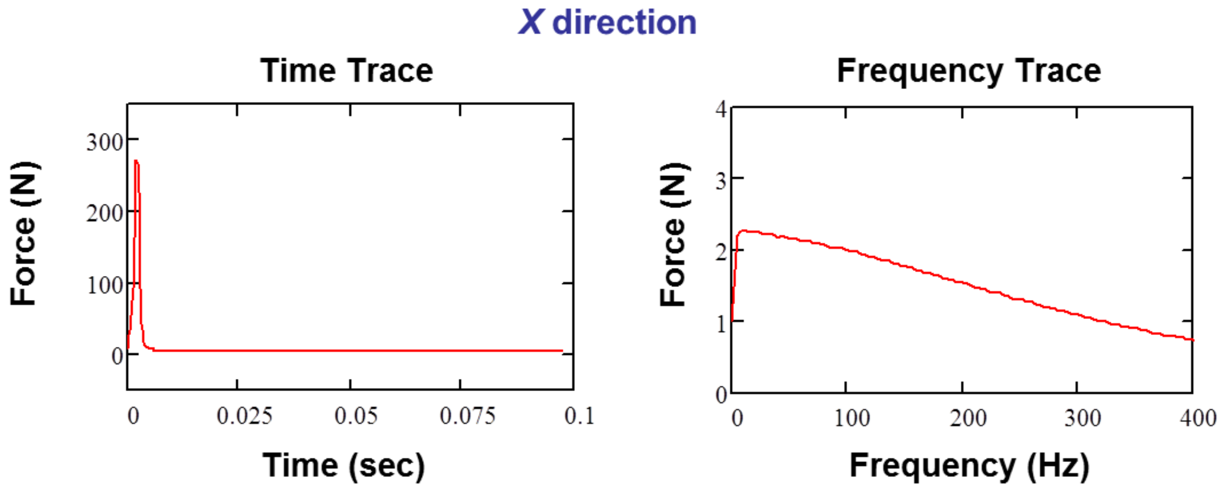
Please practice this knowledge (and learn more) by building your own canned routines (MATLAB) to produce the estimators as shown above.

Most computational software produce both spectra and cross-spectra correlation operators at the click of a mouse.

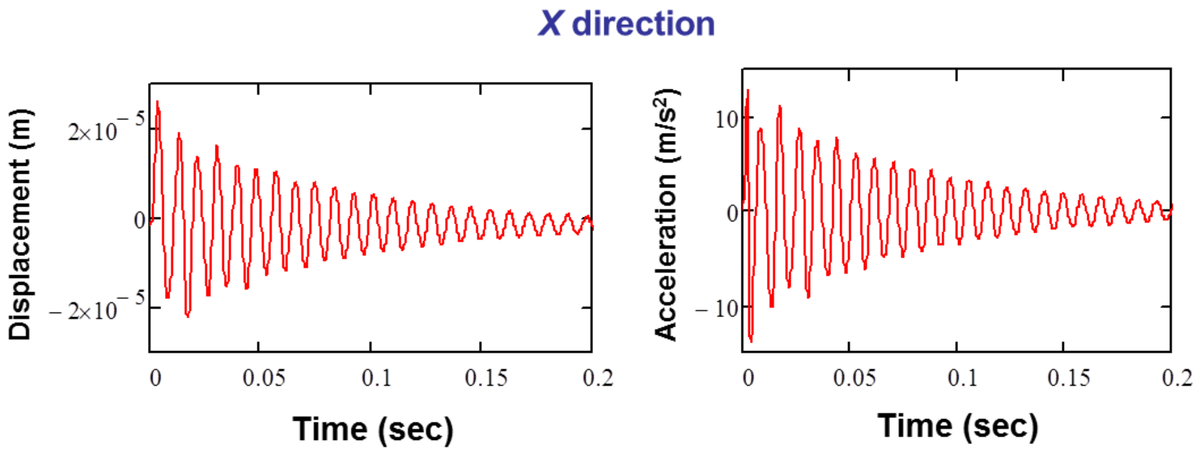
EXAMPLE of Time Response Signals → DFTs → Transfer functions → Coherence



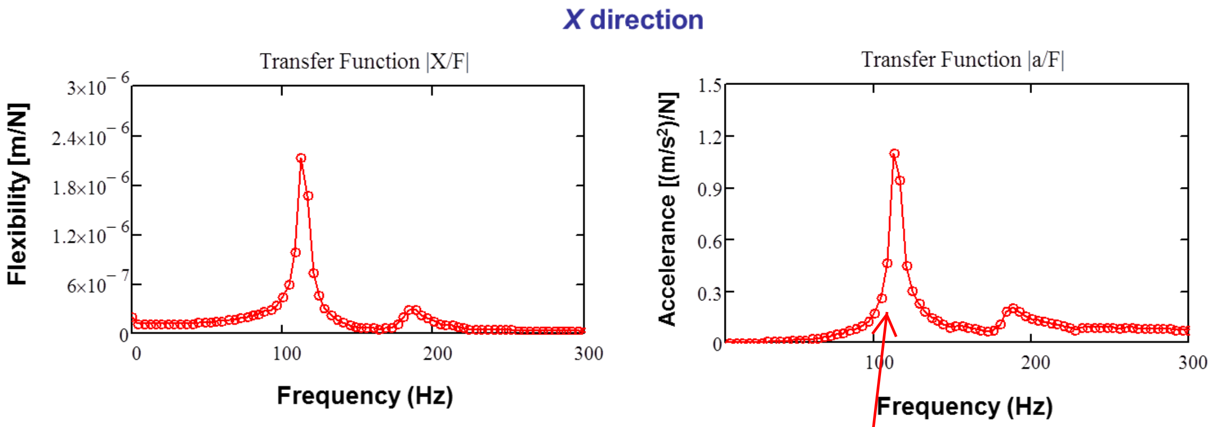
Schematic and top view of test rig and instrumentation for an impact load test



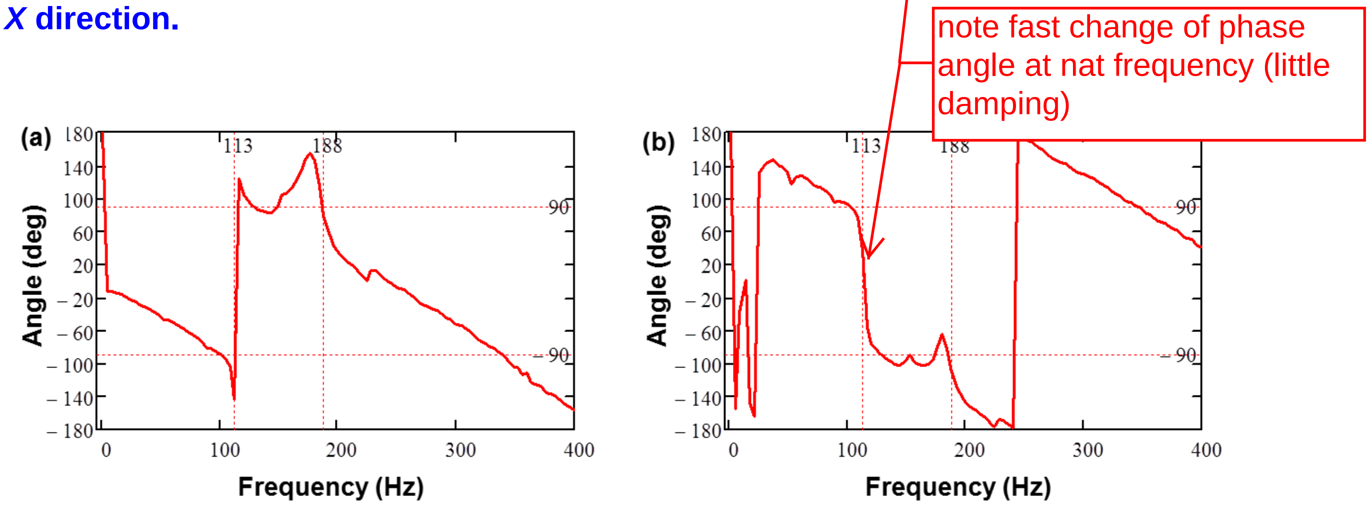
Example. Typical impact loads: time and frequency domains along *X* direction.



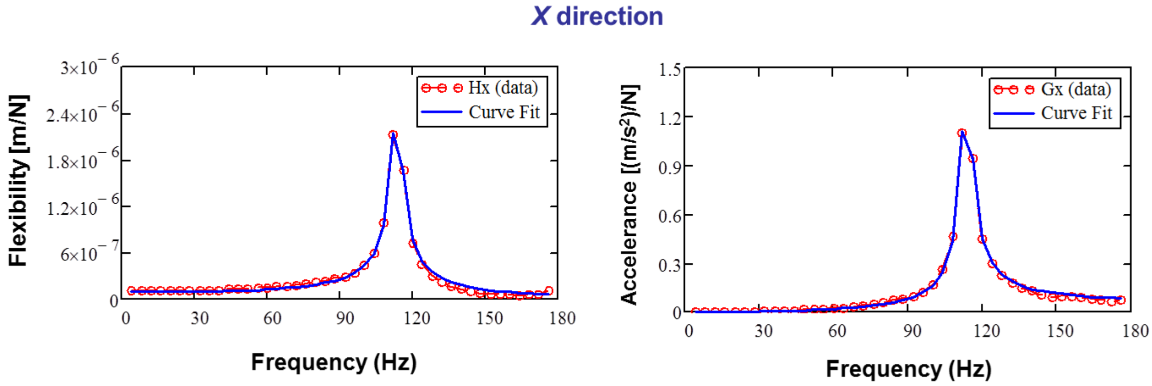
Example. Typical displacement (left) and acceleration (right) time responses to impact loads along *X* direction.



Example. Amplitude of transfer functions : flexibility function H = |X/F| and accelerance function G = |A/F| versus frequency. Response to impact load test along *X* direction.

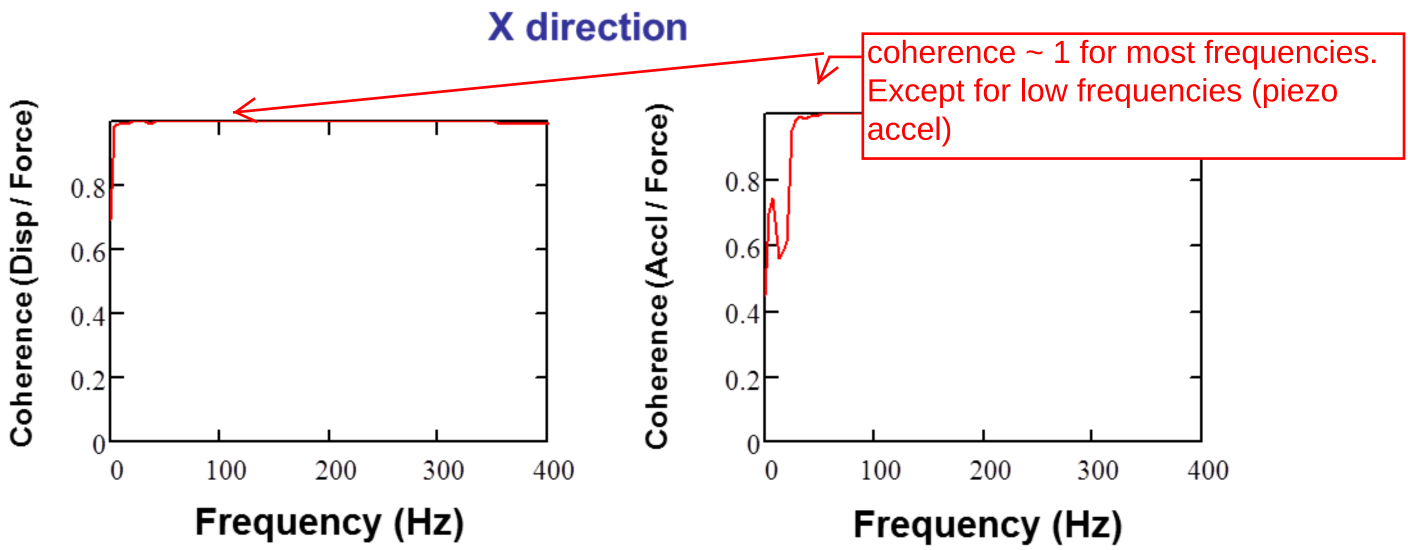


Example. Phase angle of recorded impact response versus frequency: Phase angle of (a) displacement and (b) acceleration.



Example: Amplitude of flexibility function H = |X/F| and accelerance function G = |A/F| versus frequency. Test data and model curve fit. Response to impact load test along *X* direction.

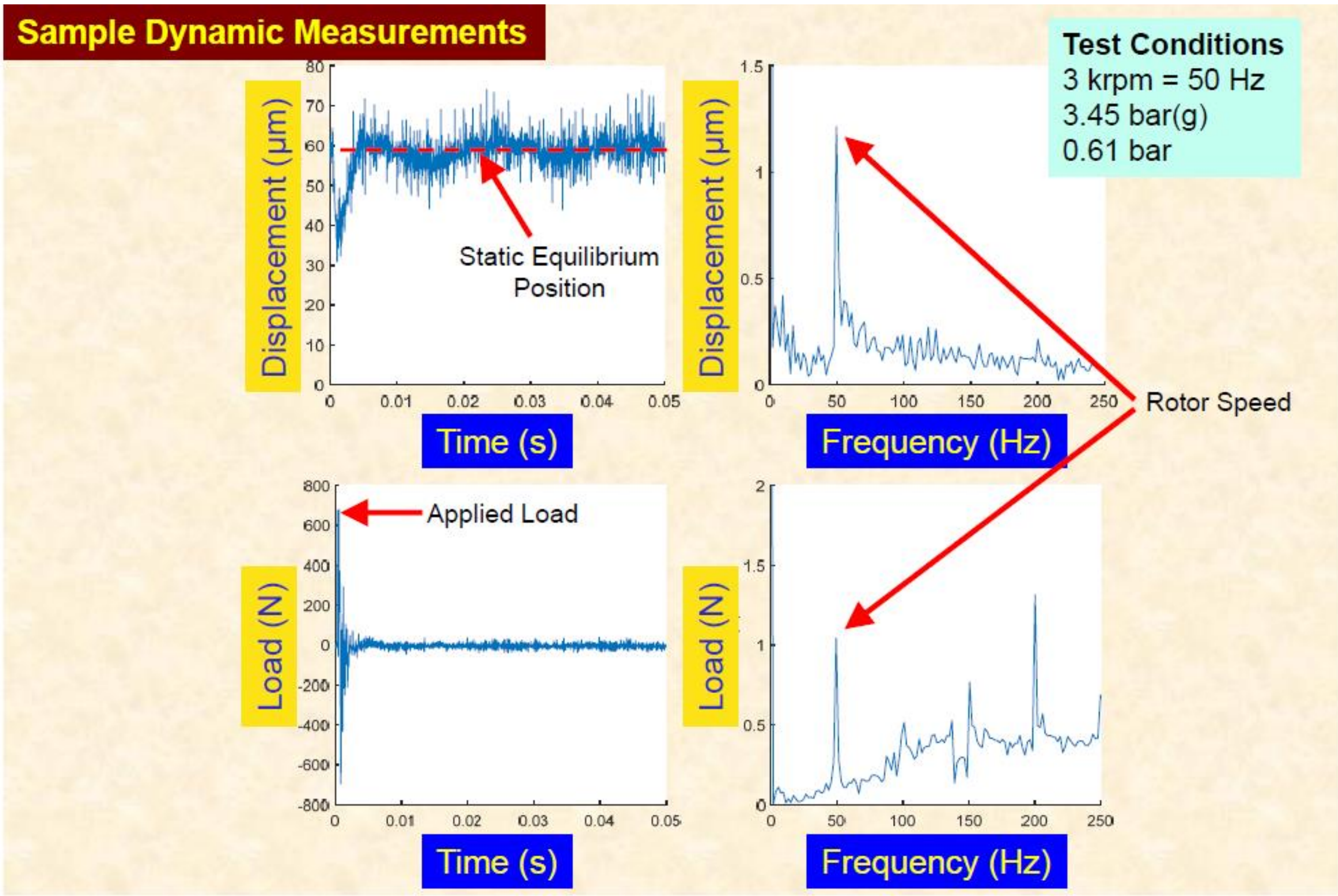
will lead to good estimation of K,C,M



Example: Coherence of flexibility (left) and accelerance (right) functions obtained from 10 impact loads on the BC along X direction.

C~ 1: Output (X) is due to input (Force)

An example with noise and shaft speed (rotor run out):



Noisy I and O! More for displacement (output) due to shaft rotation (50 Hz) and flow turbulence in bearing