

IRELE

**Measurement and Data Driven Modelling**

---

# Lab report

---

*Authors :*

Arico Amaury

Colot Emmeran

*Professor :*

J. Lataire

**Academic year :**

2024 - 2025

## Contents

<b>1 Lab 1</b>	<b>1</b>
1.1 DFT . . . . .	1
1.2 DFT of a (co)sine . . . . .	2
1.3 Time domain construction of a multisine . . . . .	5
1.4 Frequency domain construction of a multisine . . . . .	8
1.5 Influence of the phase of the multisine . . . . .	10
1.6 Random noise signals . . . . .	11
1.7 Set the Root-Mean-Square of the signal . . . . .	14
<b>2 Lab 2</b>	<b>16</b>
<b>3 Lab 3</b>	<b>17</b>
<b>4 Lab 4</b>	<b>23</b>

## 1.1 DFT

### Task 1.1.1.

**Prove that**

$$\omega_k = \frac{2\pi}{T} k$$

**where  $T = nT_s$  is the window length.**

Starting from the general definition of the iDFT and the one in case of a discrete signal:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{\frac{j2\pi kn}{N}}$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j\omega_k n T_s}$$

By simply comparing both equations,  $\omega_k = \frac{2\pi}{nT_s} k$ . Replacing the sampling period  $T_s$  multiplied with the number of samples  $n$  by the total sampling time, the result is obtained.

$$\omega_k = \frac{2\pi}{T} k$$

### Task 1.1.2.

**Prove that**

$$\omega_1 = \frac{2\pi}{T} = 2\pi \frac{f_s}{N}$$

The first equality is proven using the result of the previous task. By replacing  $T$  the window length by  $nT_s$  and then defining the sampling frequency  $f_s = \frac{1}{T_s}$ , the result is obtained.

**Task 1.1.3.****Prove that**

$$X(N - k) = X(-k) = X^*(k)$$

**Hint:** Use  $e^{j2\pi n} = 1$  for  $n \in \mathbb{N}$  and  $x(n) \in \mathbb{R}$ .

By using the definition of the DFT, the following equation is obtained:

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi kn}{N}} \\ X(N - k) &= \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi Nn}{N} - j\frac{2\pi kn}{N}} \\ X(N - k) &= \sum_{n=0}^{N-1} x(n) e^{j\frac{2\pi kn}{N}} \end{aligned}$$

Where the last equality is obtained by removing the term  $e^{-j2\pi n}$  which is equal to 1. By finally comparing the first and the third equation, one can see that only a minus sign is missing. Either  $k$  is replaced by  $-k$  giving:

$$X(N - k) = X(-k)$$

Either the conjugate of the whole equation is taken (needing the assumption that  $x(n) \in \mathbb{R}$ ):

$$X(N - k) = X^*(k)$$

## 1.2 DFT of a (co)sine

**Task 1.2.1. DFT of a 3 periods cosine**

**Generate a cosine sequence in Matlab with a randomly selected phase, and with a period that fits exactly 3 times in a data sequence of  $N = 1000$  samples. Make a plot of the DFT of this sequence (amplitude and phase).**

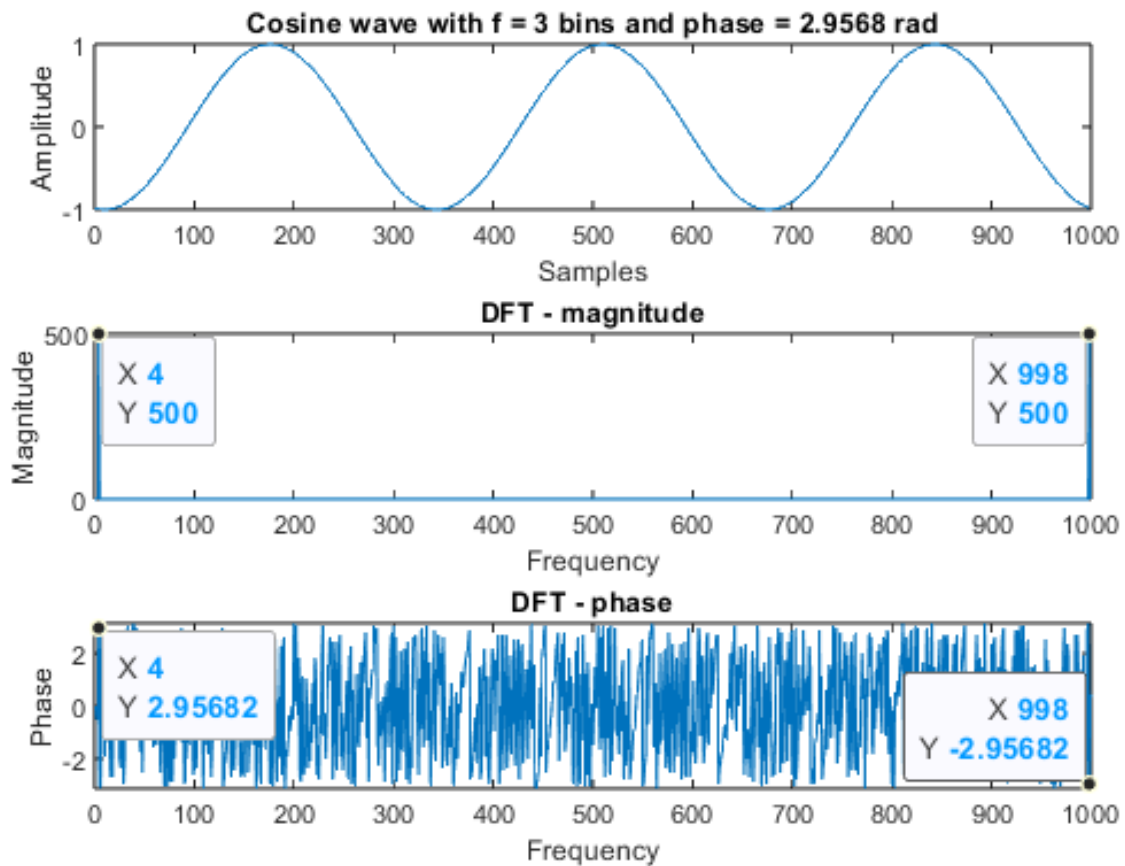


Figure 1.1: DFT of a 3 periods cosine

Remarks: from the previous section, it is known that the DFT of a cosine at a frequency of 3 bins would create a peak at the third bin and one at the  $N - 3$  bin as  $X(N - k) = X^*(k)$ . There is a shift of 1 bin as matlab indices starts at 1. concerning the phase plot, the phase at the third bin is indeed the one chosen randomly. The phases at the other bins are not relevant as the cosine is not present at these frequencies. The phase at the  $N - 3$  bin is the opposite of the one at the third bin as  $X(N - k) = X^*(k)$ .

#### Task 1.2.2. Perfect reconstruction

**From the DFT plot, check that the condition for perfect reconstruction is satisfied. Is there any leakage visible?**

*Hint: use a logarithmic amplitude axis to distinguish small (but non-zero) values.*

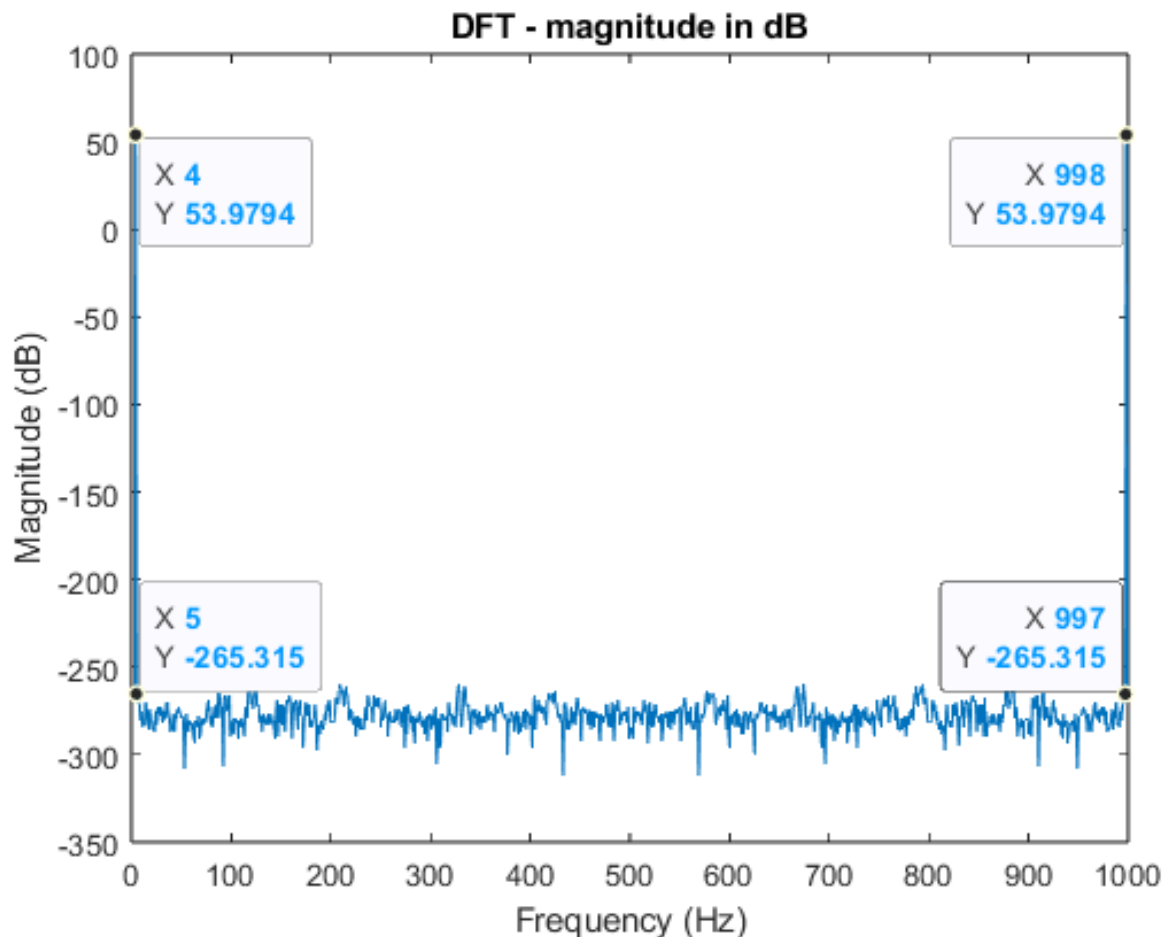


Figure 1.2: Perfect reconstruction

There is a difference in amplitude of more than 300 dB between the excited bin and its neighbors. This shows that the condition for perfect reconstruction is satisfied.

#### Task 1.2.3. Interpretation of the frequency axis

**At which indices of the DFT do you obtain non-zero values? Explain. (Keep in mind that Matlab indices start at 1, not at 0.)**

As already discussed in task 1.2.1, the DFT of a cosine at a frequency of 3 bins would create a peak at the third bin and one at the  $N - 3$  bin as  $X(N - k) = X^*(k)$ . There is a shift of 1 bin as matlab indices starts at 1. The other bins are (close to) zero as there is no excitations at these frequencies.

#### Task 1.2.4. Frequency axis in bins

**Construct the frequency axis for the plots, expressed in bins.**

It is already done in the matlab script as the x-axis of the plots is expressed in "DFT samples number", which are in fact bins.

#### Task 1.2.5. Frequency axis in Hz

Consider that the sample frequency is  $f_s = 100$  Hz. Construct the frequency axis for the plots, expressed in Hz.

(Hint: use the results from Task 1.1.1.)

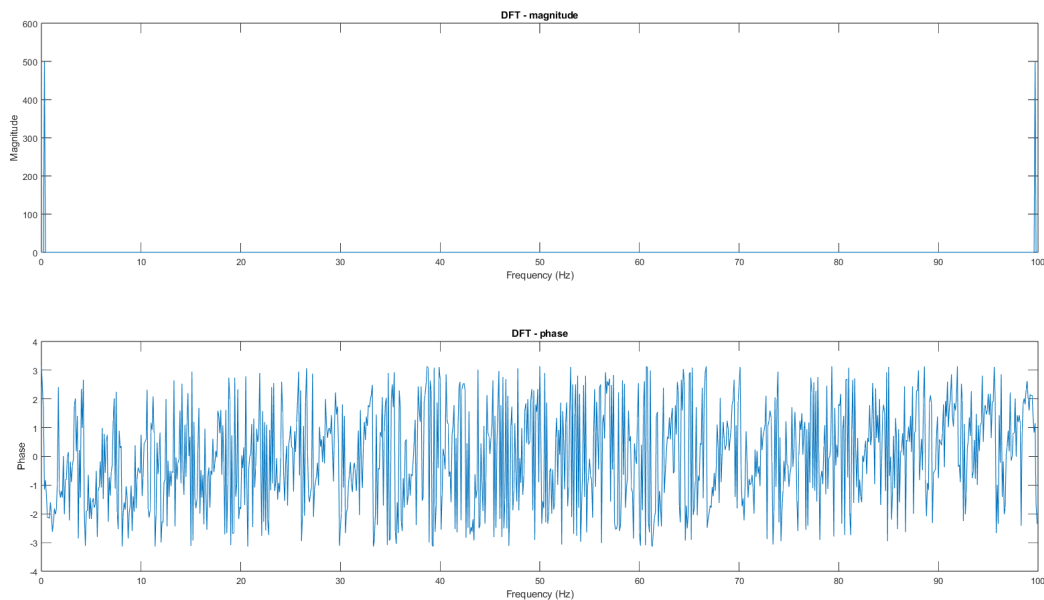


Figure 1.3: Frequency axis in Hz

To change the frequency axis from bins to Hz, it was simply multiplied by  $\omega_1 = \frac{f_s}{N}$ .

### 1.3 Time domain construction of a multisine

#### Task 1.3.1. Time domain random phase multisine

Generate a multisine in the time domain, by implementing (1.6), with  $N = 1000$  samples and  $K = 10$  excited frequencies. Set the amplitudes  $A_m = 1$ , and choose the phases  $\varphi_m$  randomly between  $0$  and  $2\pi$  (i.e. a random phase multisine). Check that this multisine satisfies the condition for perfect reconstruction by plotting its DFT. (Use a logarithmic amplitude axis). Include the frequency axis, expressed in bin.

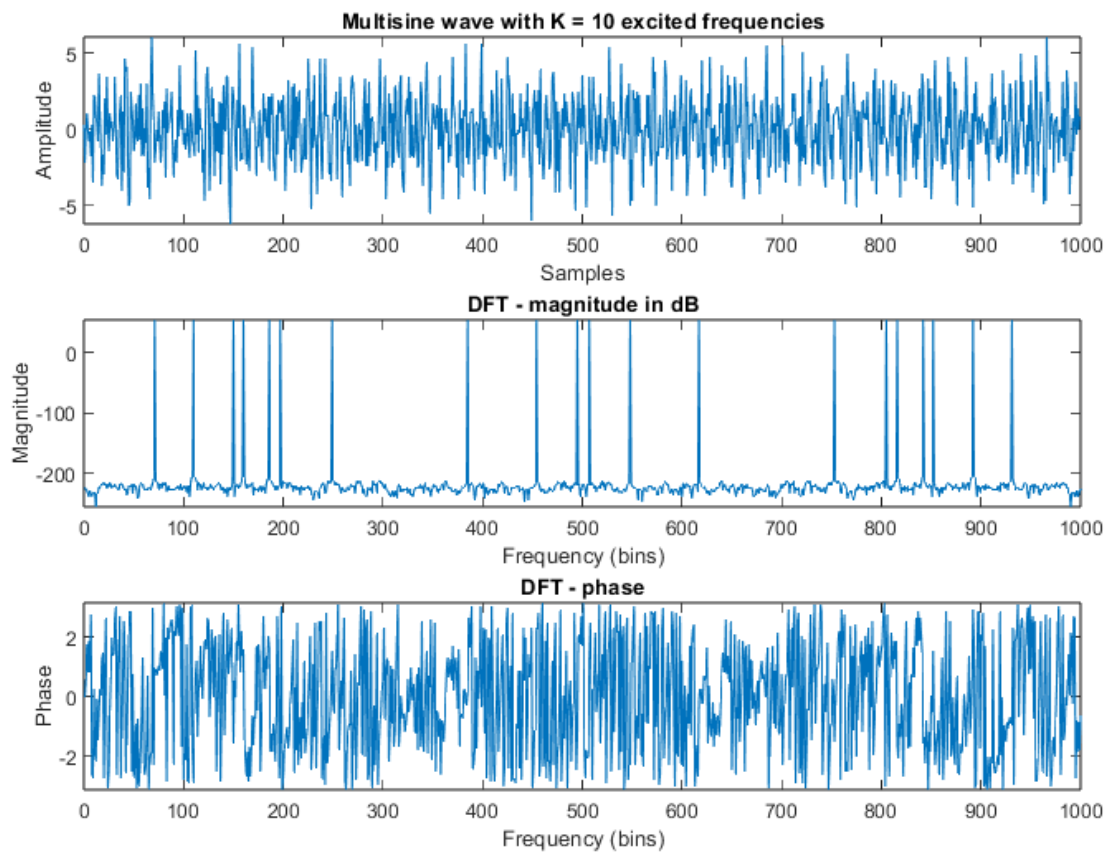


Figure 1.4: Time domain random phase multisine

The condition for perfect reconstruction is satisfied as the difference in amplitude between the excited bins and their neighbors is more than 250 dB.

**Task 1.3.2. Frequency axis in Hz**

**For the multisine generated in Task 1.3.1, consider that the sampling frequency is  $f_s = 100\text{Hz}$ . Include the frequency axis expressed in Hz in the DFT plot, and the time axis expressed in seconds for the time domain plot.**



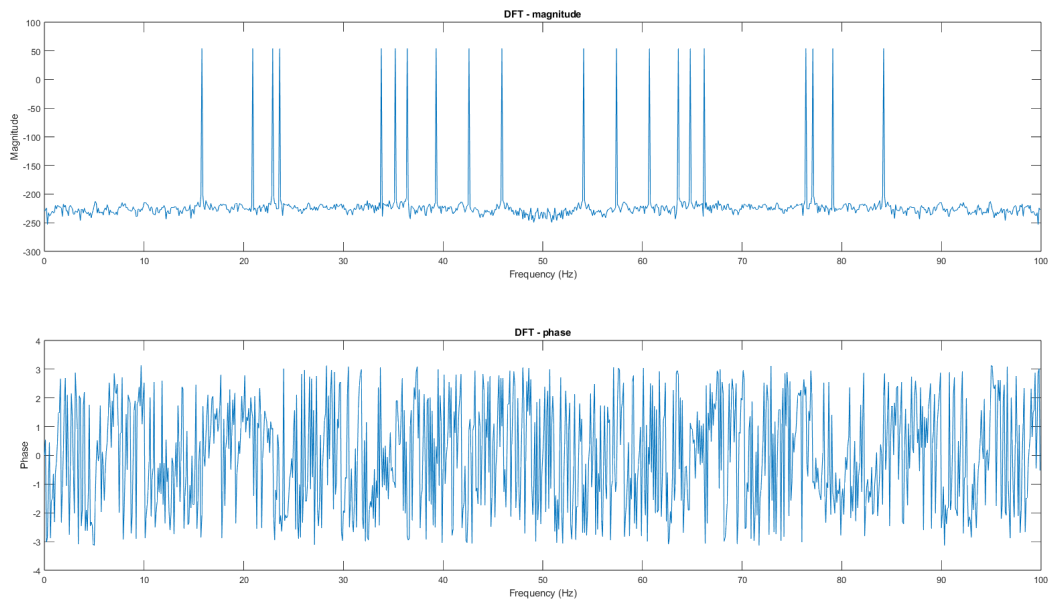


Figure 1.5: Frequency axis in Hz

As in task 1.2.5, the frequency axis was converted from bins to Hz by multiplying it by  $\omega_1 = \frac{f_s}{N}$ .

**Task 1.3.3. Excite specific frequencies**

**Generate a random phase multisine with a sampling frequency of  $200\text{ Hz}$ , with excited frequencies**

$[4, 8, 12, 16, 20, 24]\text{ Hz}$

**Plot the time and frequency domain results, with appropriate axes.**

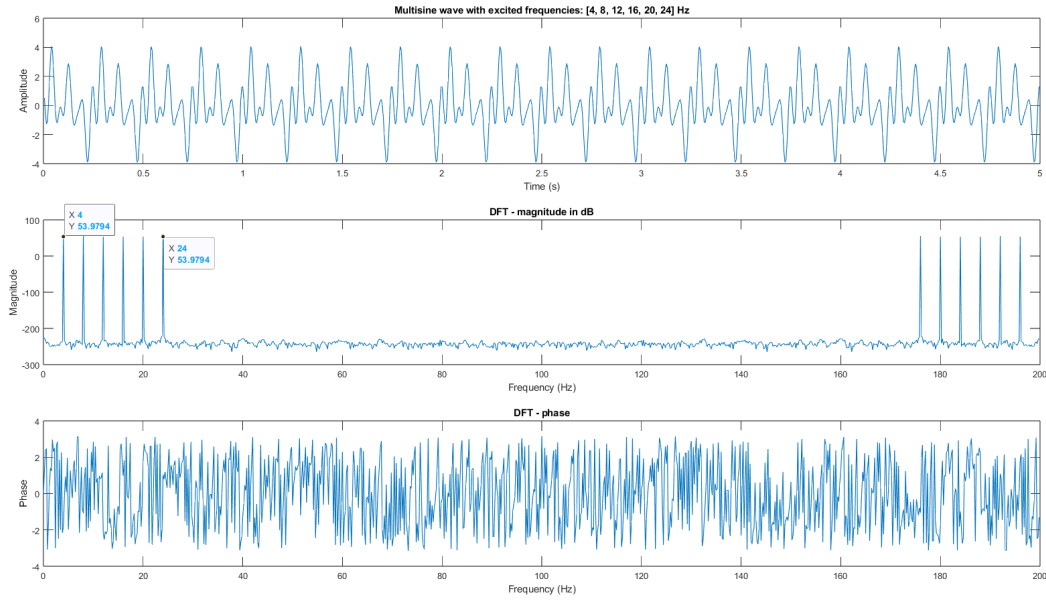


Figure 1.6: Excite specific frequencies

## 1.4 Frequency domain construction of a multisine

### Task 1.4.1. Trick for frequency domain multisine construction

Consider the vector  $\tilde{X}(k)$ , such that

$$\begin{aligned}\tilde{X}(k) &= A_k e^{j\varphi_k} && \text{for } 1 \leq k \leq K \\ \tilde{X}(k) &= 0 && \text{otherwise}\end{aligned}$$

Prove that

$$x(n) = N \Re \{ \text{iDFT}(\tilde{X}(k)) \} = \sum_{k=1}^K A_k \cos\left(\frac{2\pi k n}{N} + \varphi_k\right)$$

Where  $\Re$  denotes the real part.

*Hint: use the definition of the iDFT*

Starting from the definition of the iDFT:

$$\begin{aligned}x(n) &= \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{\frac{j2\pi kn}{N}} \\ &= \frac{1}{N} \sum_{k=1}^K A_k e^{j\varphi_k} e^{\frac{j2\pi kn}{N}} \\ &= \frac{1}{N} \sum_{k=1}^K A_k e^{\frac{2\pi k n}{N} + j\varphi_k}\end{aligned}$$

By then taking the real part of  $x(n)$  and multiplying it by  $N$ :

$$\begin{aligned} x(n) &= \frac{N}{N} \sum_{k=1}^K A_k \Re \left( e^{\frac{2\pi k}{N} + \varphi_k} \right) \\ &= \sum_{k=1}^K A_k \cos \left( \frac{2\pi k}{N} + \varphi_k \right) \end{aligned}$$

#### Task 1.4.2. Frequency domain multisine

Use the frequency domain approach to construct a random phase multisine, by using the trick from Task 1.4.1. Let  $N = 1000$ , and excite the first 30 bins. Make time and frequency domain plots (frequency axis expressed in bins).

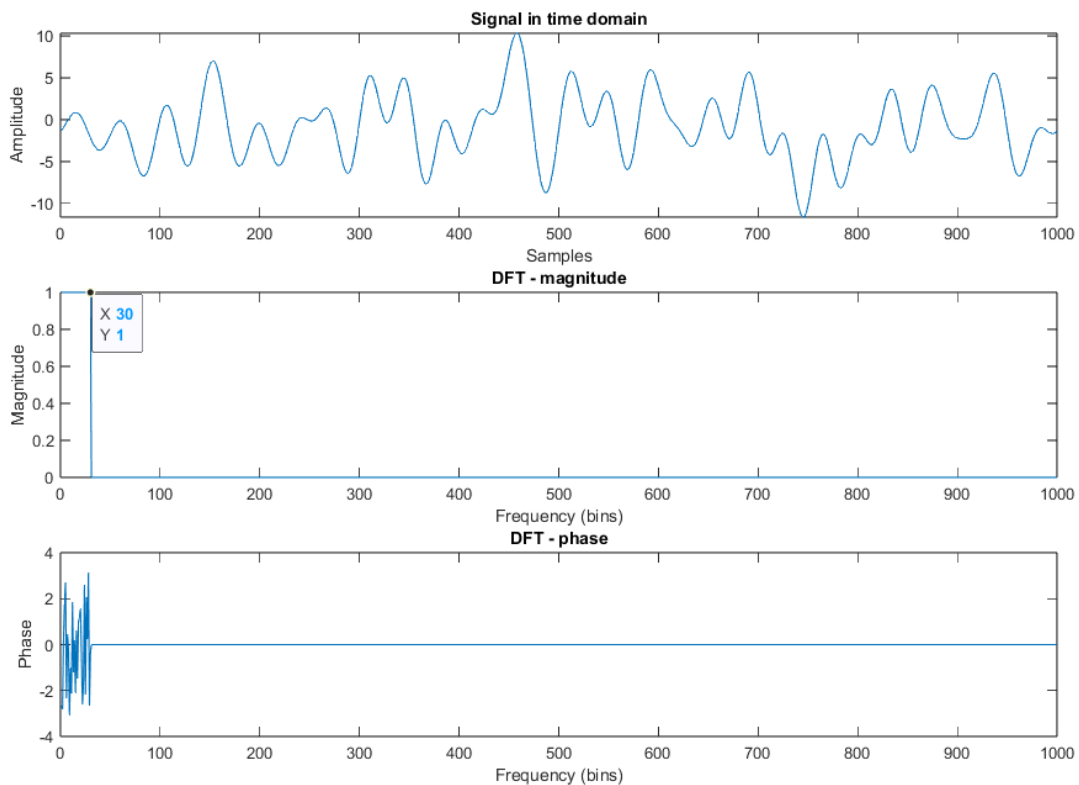


Figure 1.7: Frequency domain multisine

#### Task 1.4.3. Specified excited frequency band and frequency resolution

Construct a random phase multisine in the frequency domain, which excites the frequency band [5, 15] Hz at 30 equidistantly spaced frequencies. Choose an appropriate sampling frequency. Make time domain and frequency domain plots (time axis in seconds, frequency axis in Hz). How long is one period of this multisine (expressed in seconds)?

A frequency band of 10 Hz with 30 equidistantly spaced frequencies means that a bin must be equal to

$1/3\text{Hz}$  (or a divider of it). Using the formula of  $\Delta_f = \frac{f_s}{N}$  (where  $\Delta_f$  is the frequency resolution) and using a sampling frequency  $f_s$  larger than twice the maximum signal frequency:

$$f_s = 50\text{Hz} \quad N = 3f_s = 150$$

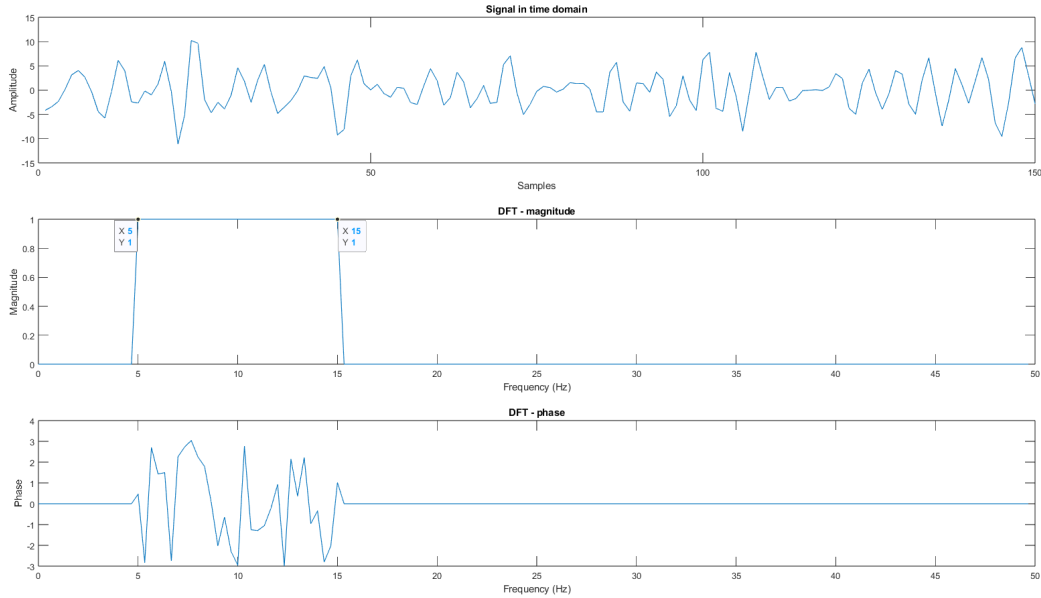


Figure 1.8: Specified excited frequency band and frequency resolution

Because it was built with the perfect reconstruction condition, the period of the multisine is equal to the period of a sine at the frequency resolution, which is  $1/3\text{Hz}$  so the period is 3 seconds.

## 1.5 Influence of the phase of the multisine

### Task 1.5.1. Crest Factor

Construct a multisine, with  $N = 500$ , with the first  $K = 60$  bins excited, and with the following phases:

- **random phase:** chosen randomly in  $[0, 2\pi]$  (uniform distribution)
- **Schroeder phase:**  $\varphi_k = \frac{m(m+1)\pi}{K}$
- **Linear phase:**  $\varphi_m = m\pi$

Make time and frequency domain plots (in samples and bins), and compute the Crest Factors. Describe, qualitatively, the relationship between the time domain plot and the crest factor. What is the advantage of a low/high crest factor?

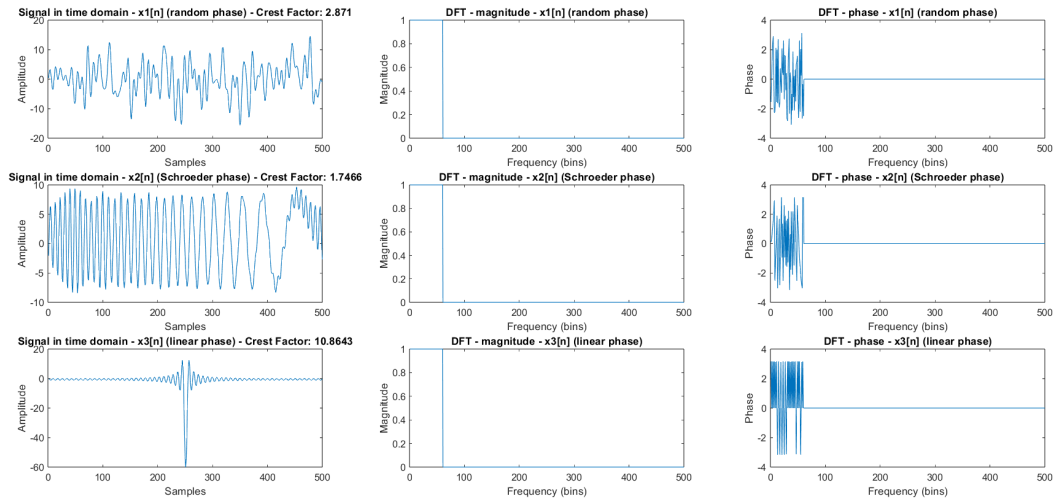


Figure 1.9: Crest Factor

Based on the figure, it is quite clear that a larger Crest Factor corresponds to a signal that has peaks of larger amplitude ( $\max|x_3| \approx 6 \times \max|x_2|$ ). This is easily understandable as the Crest Factor is defined as the ratio between the peak value and the RMS value of the signal. The advantage of a low Crest Factor is that the signal has no big peaks, which could damage an unprotected electronic system receiving it.

## 1.6 Random noise signals

### Task 1.6.1. White Gaussian random noise

Generate a normally distributed (Gaussian), random, white noise sequence of  $N = 1000$  samples, by using the Matlab function `randn`. Make time and frequency domain plots (axes in samples and bins). Observe that all the bins are excited, with random amplitudes and phases.

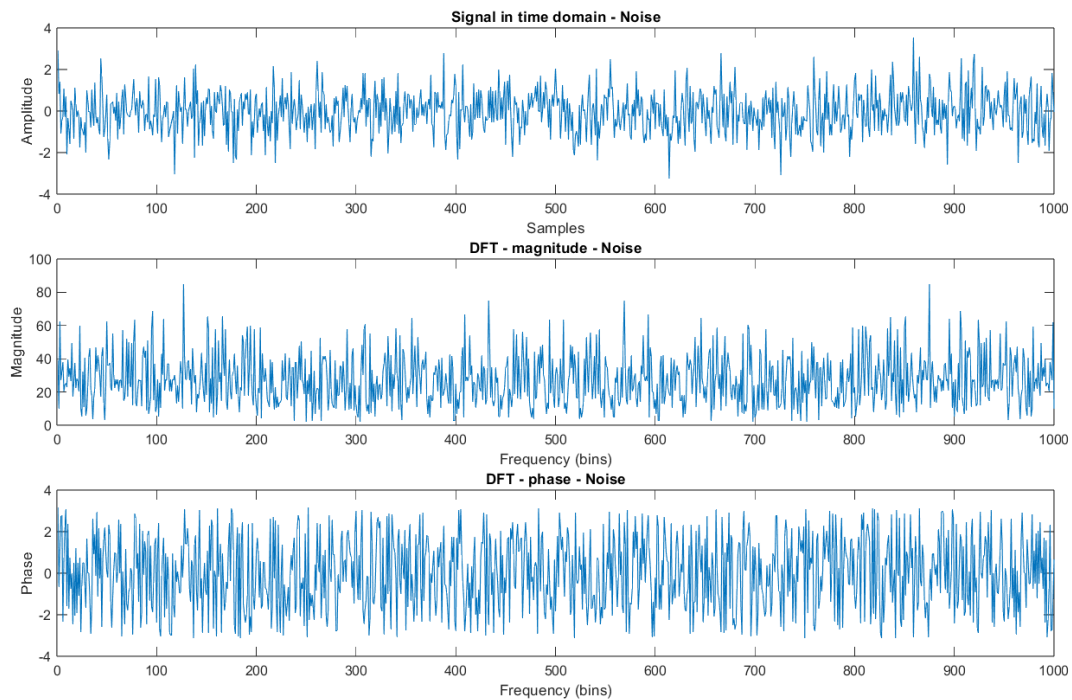


Figure 1.10: White Gaussian random noise

#### Task 1.6.2. Filtered random noise

Generate a filtered random noise sequence with  $N = 1000$ , sampling frequency 100 Hz, from a Gaussian white noise sequence (use `randn`). Do this by using the function `cheby1` to create a low-pass digital Chebyshev filter of order 5, ripple 2 dB, and such that the passband edge lies at 5 Hz. Filter the sequence by using the function `filter`. Make time and frequency domain plots (axes in seconds and Hz), and check that the excited frequency band is as expected. What do you observe in the stop-band of the filter? Is it equal to 0? Explain.

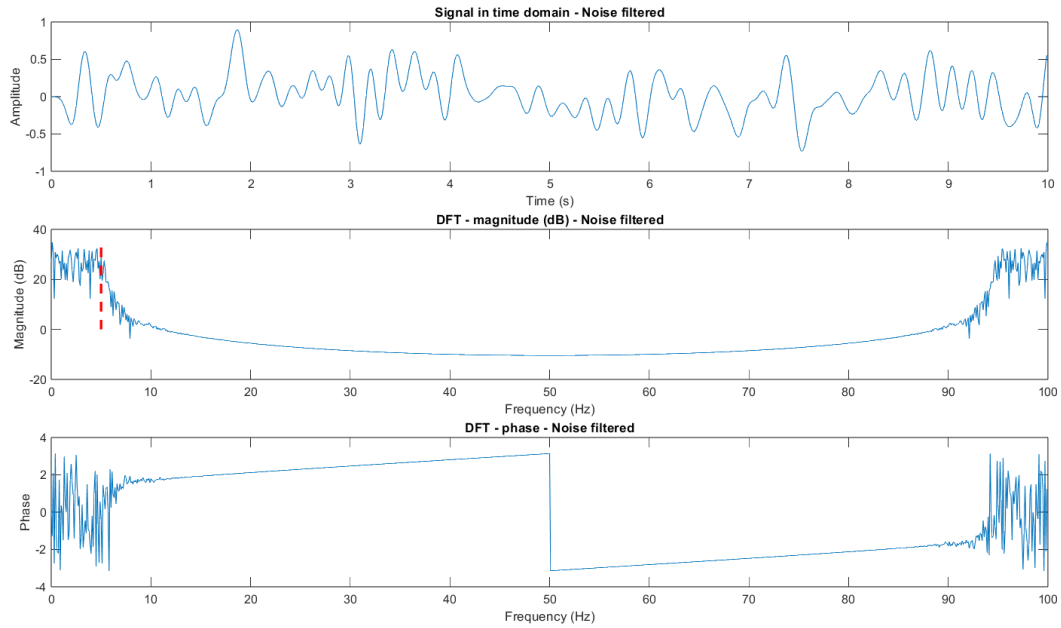


Figure 1.11: Filtered random noise

The excited frequency band is indeed reduced to 5 Hz as expected. In the stop-band of the filter, the amplitude is not equal to 0. This is due to the fact that the filter is not an ideal filter, and thus does not completely remove the frequencies above 5 Hz. The amplitude after 10 Hz is already reduced by 30 dB and the increase for higher bins is due to the symmetry of the DFT proved in task 1.1.3.

#### Task 1.6.3. Periodic band-limited random noise

**Generate a Gaussian random noise sequence (use `randn`), with  $N = 1000$  and sampling frequency 100 Hz. Compute the DFT, and set the DFT at all frequencies beyond 5 Hz to zero:**

$$\begin{aligned} \tilde{X}(k) &= 0 & \text{for } \omega_k \geq 10\pi \text{ rad/s} \\ \tilde{X}(k) &= X(k) & \text{otherwise} \end{aligned}$$

**and use the expression**

$$x(n) = 2\Re\{\mathbf{iDFT}(\tilde{X}(k))\}$$

**to obtain the time sequence. Make time and frequency domain plots (axes in seconds and Hz).**

**If you repeat this time domain sequence  $x(n)$  (by putting multiple copies of the sequence after each other), and computing the DFT of the result, no leakage should occur. Check this, and explain why this is the case.**

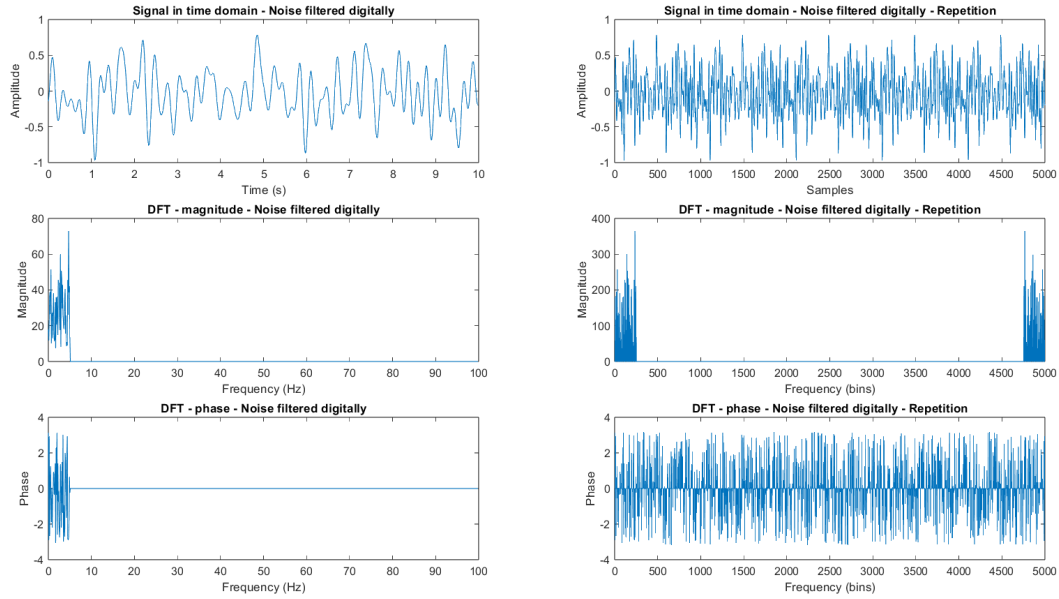


Figure 1.12: Periodic band-limited random noise

There is no leakage when repeating the **idft** of the filtered noise as the condition on perfect reconstruction is fulfilled. As the signal that is repeated is built from the frequential domain, it will have an integer number of periods in the time domain, and thus when replicated no leakage will occur as it will still have an integer number of periods inside the window of the **dft**.

## 1.7 Set the Root-Mean-Square of the signal

### Task 1.7.1. RMS value

Set the RMS value of your favourite signal from the previous tasks to  $RMS_{des} = 3$ , as follows:

$$x_{des}(n) = x(n) \frac{RMS_{des}}{RMS(x)}$$

Prove that the RMS of  $x_{des}(n)$  is indeed  $RMS_{des}$ .



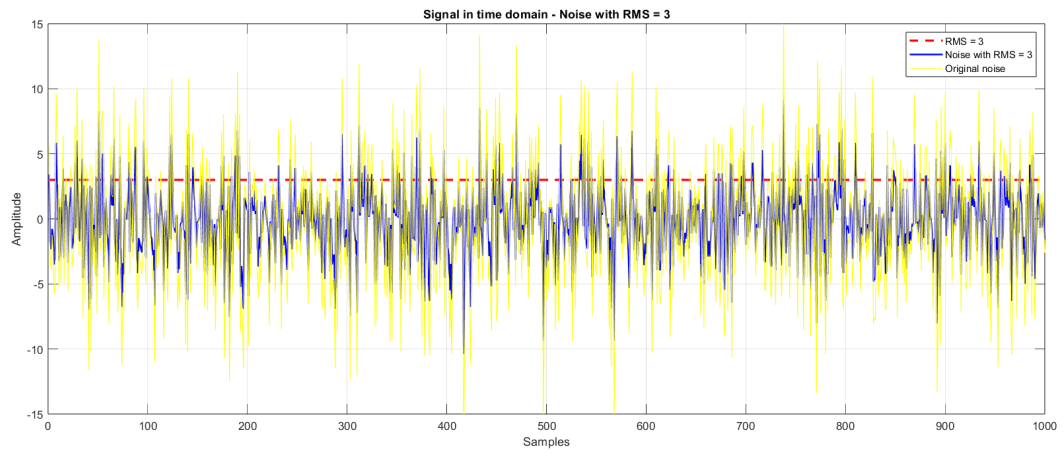


Figure 1.13: RMS value

Note that the title of the plot contains the RMS value of the modified signal and it indeed reaches 3.

# 2

## Lab 2

*Question 3.1. Length data records*

**The FFT algorithm works faster when the length of the data records  $N$  is a power of 2. Do you know why?**

To numerically implement a DFT, the DFT algorithm is used. It works by dividing the input signal into two parts, each of which is transformed separately, and then the results are combined and this is done recursively. It means that a number of samples that is a power of 2 will be perfectly divided at each step, leading to a faster computation.

*Question 3.2. Trigger signal*

**What is the purpose of a trigger signal? A number of processing methods were discussed in the theory to improve the SNR of the FRF by averaging of measurements. Which methods require a trigger signal to operate properly? Explain what happens if the trigger is absent while needed.**

For the averaging in the time domain and the averaging in the DFT spectra, the input and output signals need to be identical:

$$H_{\text{time}}(j\omega_k) = \frac{\text{DFT}\left(\frac{1}{M} \sum_{i=1}^M y_i(n)\right)}{\text{DFT}\left(\frac{1}{M} \sum_{i=1}^M u_i(n)\right)}$$

$$H_{\text{DFT}}(j\omega_k) = \frac{\frac{1}{M} \sum_{i=1}^M \text{DFT}(y_i(n))}{\frac{1}{M} \sum_{i=1}^M \text{DFT}(u_i(n))}$$

For the measurements to be identical between two repetitions (except for the noise of course), they should always start at the same point in the period of the excitation and the trigger signal make sure it is the case.

### Question 3.3. Frequency domain multisine

In your report, show the Matlab code you used to construct the multisine signal (with random or Schroeder phase) in the frequency domain. Make sure that the code is sufficiently commented to improve its readability.

```
fs = 16000; % Sampling frequency
excFreq = [4 1000]; % Excited frequencies in Hz
rmsVal = 0.5; % RMS value
N = 4000; % Number of samples

m = (excFreq(1)*N/fs:excFreq(2)*N/fs);
% Excited frequencies in bins
K = length(m); % Number of excited frequencies

shroedPhase = m.*(m+1) * pi / K;
signalFreq = zeros(1, N);

signalFreq(m + 1) = exp(1j*shroedPhase);
% m + 1 because the index starts from 1 in MATLAB

% Generate the signal
signalTime = 2*N*real(iff(signalFreq));
% Normalize the signal
signalTime = signalTime*rmsVal/rms(signalTime);
```

### Question 3.4. Averaging the measurements

Which averaging techniques are applicable to which excitation signals? Explain.

The time averaging and the frequency averaging can be applied with the periodic excitation only. The issue with a random input is that averaging it could still result in some bins with a close to zero amplitude, which would create spikes in the FRF. It is worth noting that these two methods are equivalent (at least theoretically, in practice there are some differences due to the way they are implemented).

Computing the FRF of each repetition separately and then averaging them could also lead to a division by zero, making it unusable for the random excitation.

Finally, using the average of the auto-power of the input/output is a method that can be used for any input signal as it divides by a power that is not zero. It is preferred to use the average of the auto-power of the input signal as it is less noisy than the output signal. This is because the DUT is a filter and the output signal is lower in amplitude than the input signal, leading to a smaller SNR. It must be noted that these methods

introduce a bias. This bias however is null if the additive noise is circular complex.

**Question 3.5. Plots**

**Provide relevant plots of the estimated FRF, obtained with the different methods, with the different excitation signals and with a different number of averaged records.**

On the 3 first figures, the aperiodic FRF is "too clean". It should have more spikes as explained in question 3.4.

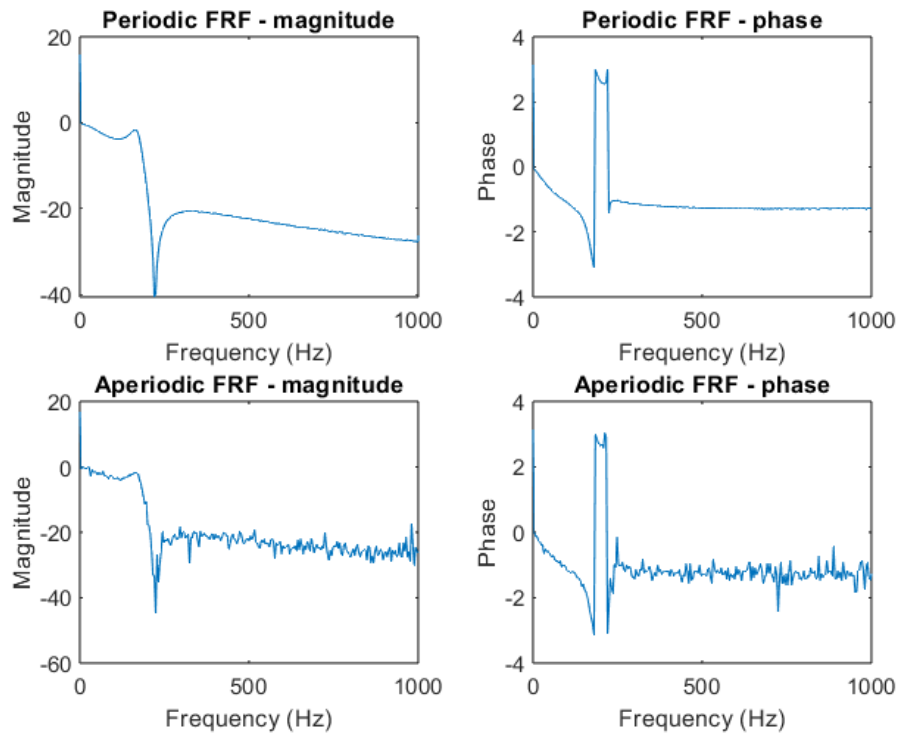


Figure 3.1: Estimated FRF with time averaging

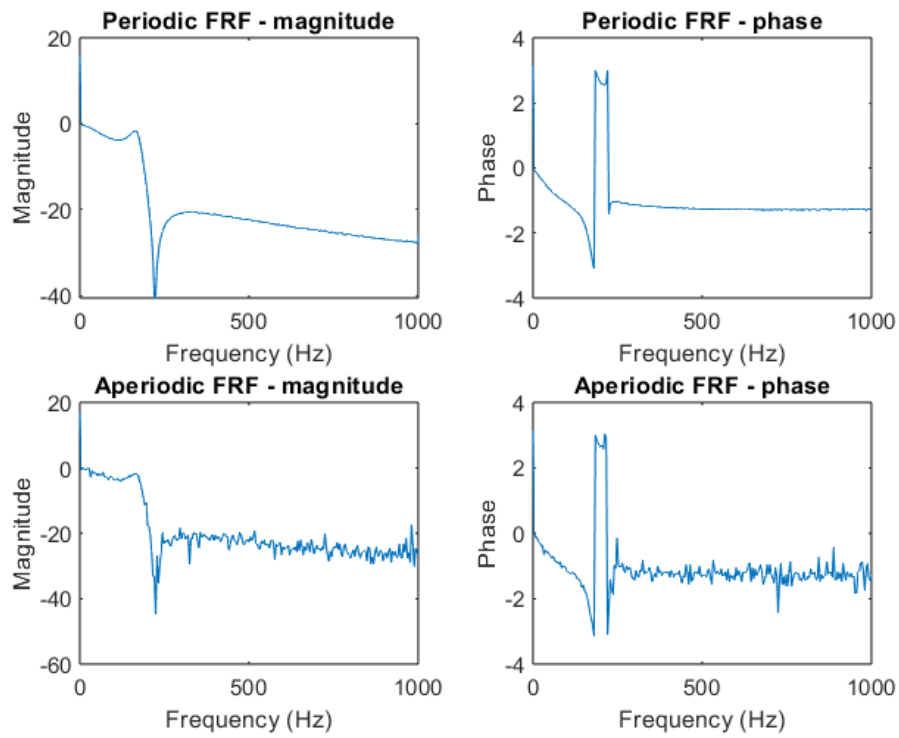


Figure 3.2: Estimated FRF with frequency averaging

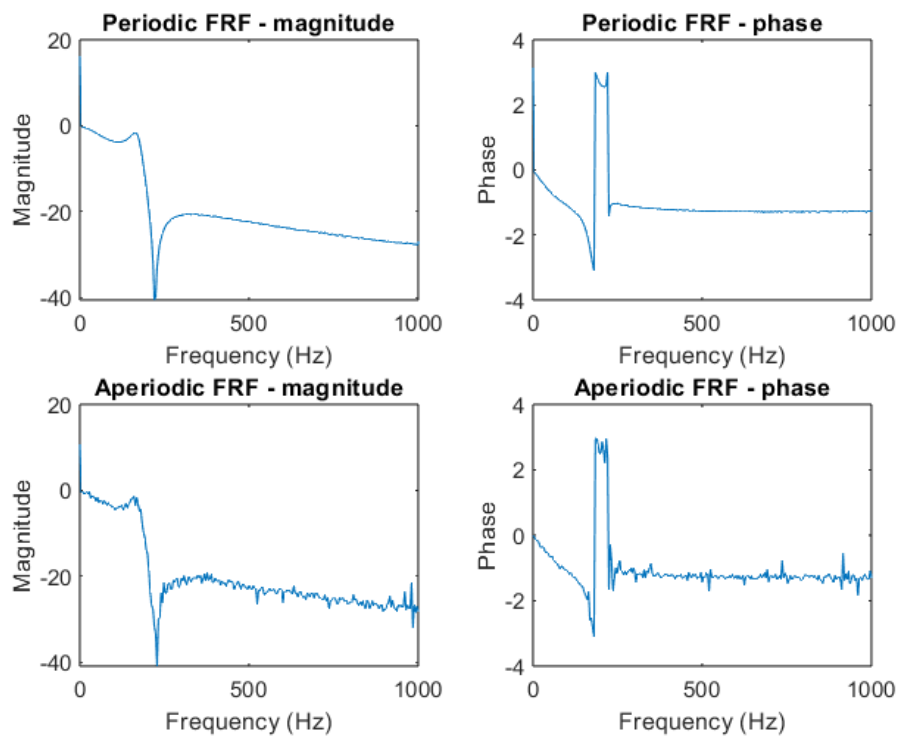


Figure 3.3: Estimated FRF by averaging the FRF

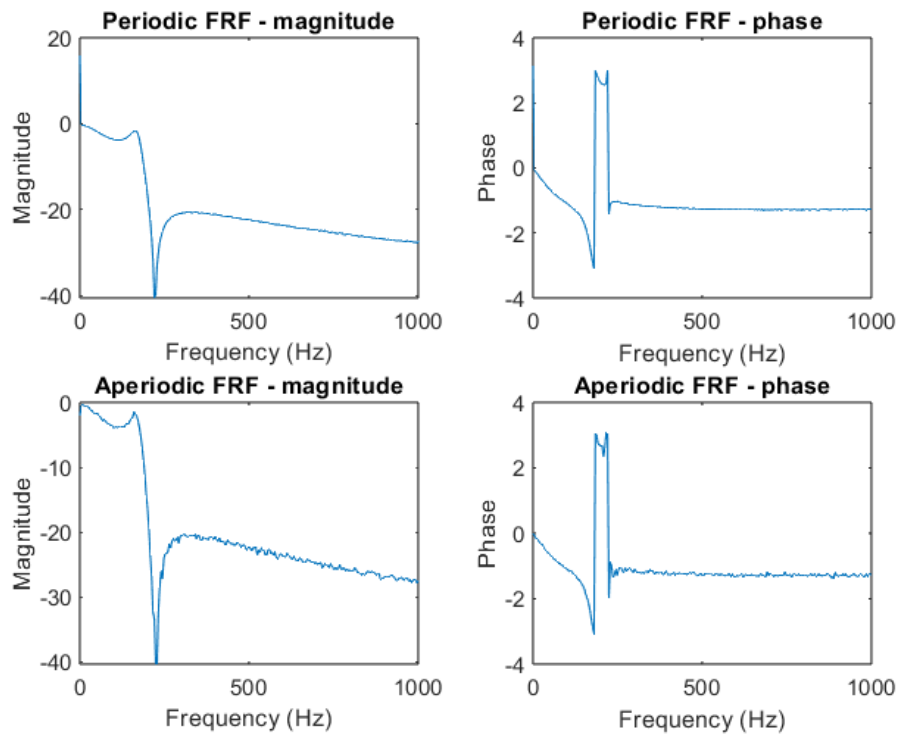


Figure 3.4: Estimated FRF by averaging the auto-power of the input signal

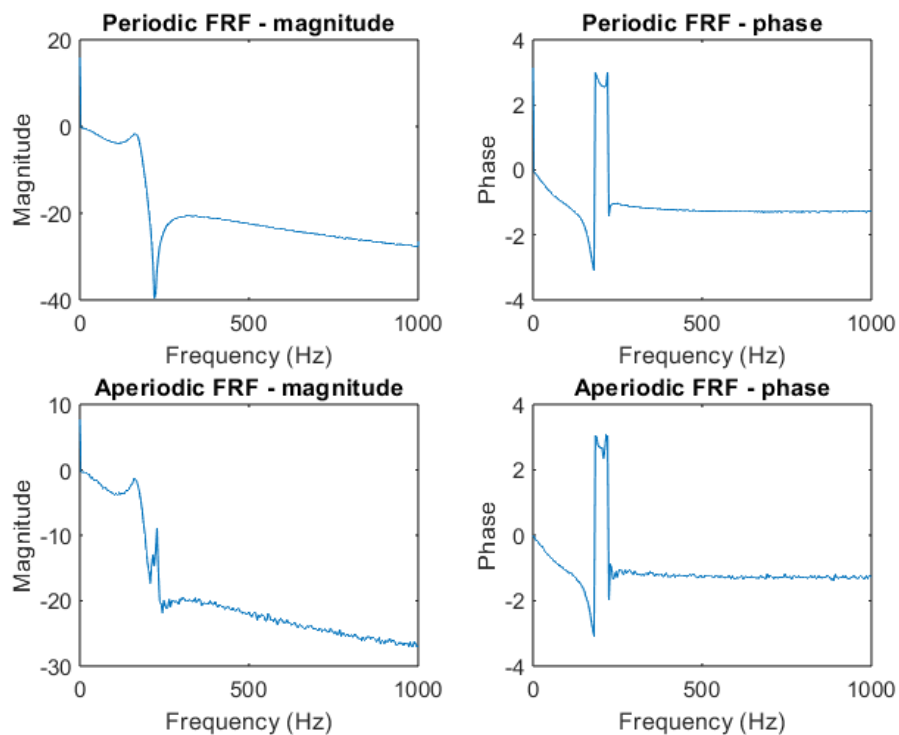


Figure 3.5: Estimated FRF by averaging the auto-power of the output signal

Here is the effect of the number of averaged records for the time averaging method:

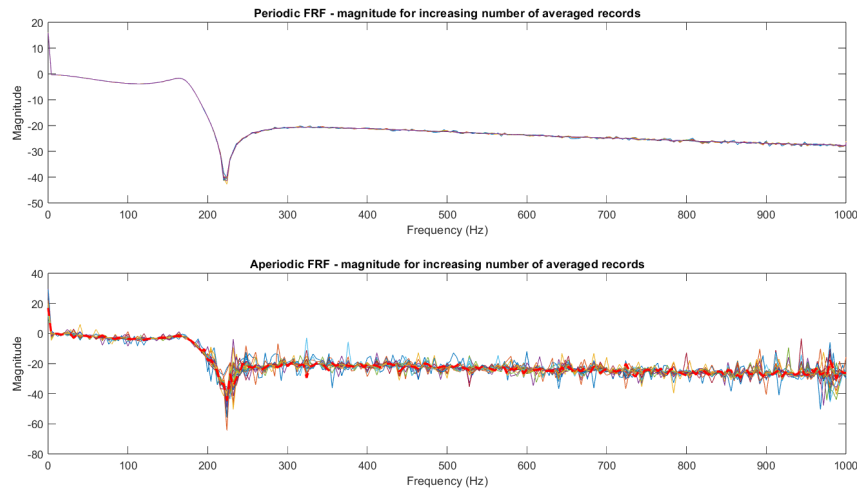


Figure 3.6: Estimated FRF for different number of averaged records

Because the transient has been removed by discarding the first 8 periods, the estimated FRF based on the periodic signal does not change. This indicates that the noise power was extremely small compared to the signal power (we might have accidentally saved the response of a noiseless test). The estimated FRF based on the random input is however more interesting: the last estimation (the red dotted line) has way less spikes than all the previous ones. It really shows the benefit of averaging the measurements.

*Question 3.6. Impact of repetition number*

**Determine the effect of the number of averaged records on the variability of the averaged result. Compare the standard deviation.**

TODO

*Question 3.7. Discussion*

**Discuss the differences and explain according to you, which will deliver the best result. Discuss the pros and cons of each excitation signal.**

TODO



# 4

## Lab 4