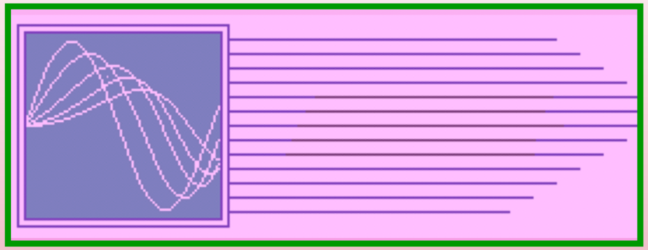
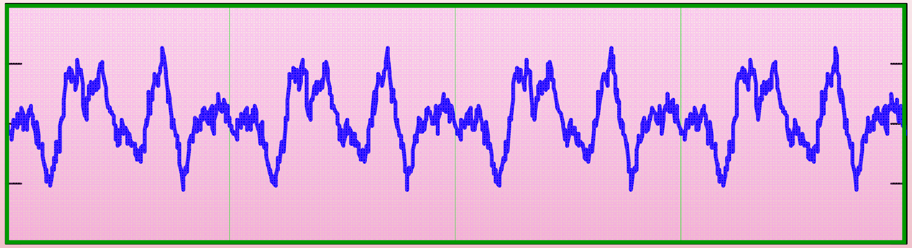
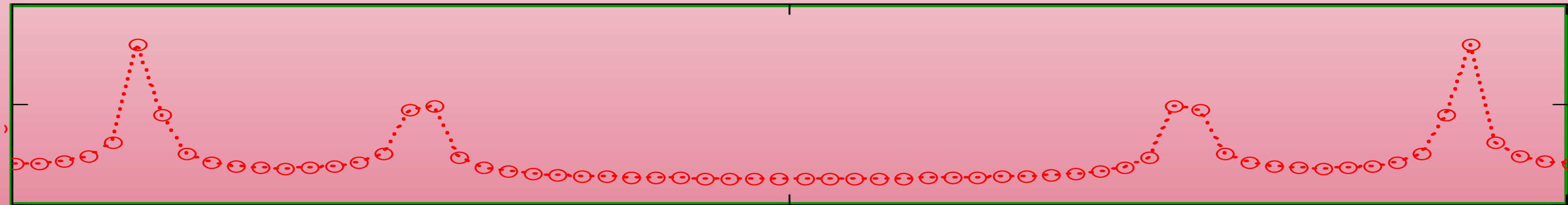
ANALOG AND DIGITAL SIGNAL PROCESSING

ADSP – Chapter 8







$$Fm = \sum_{n=0}^{N-1} f(n \cdot T) \cdot e^{\frac{-j \cdot 2 \cdot \pi \cdot m \cdot n}{N}}$$

$$\begin{pmatrix} X(0) \\ X(2) \\ X(1) \\ X(3) \end{pmatrix} = \begin{pmatrix} 1 & W^0 & 0 & 0 \\ 1 & W^2 & 0 & 0 \\ 0 & 0 & 1 & W^1 \\ 0 & 0 & 1 & W^3 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & W^0 & 0 \\ 0 & 1 & 0 & W^0 \\ 1 & 0 & W^2 & 0 \\ 0 & 1 & 0 & W^2 \end{pmatrix} \cdot \begin{pmatrix} x_0(0) \\ x_0(1) \\ x_0(2) \\ x_0(3) \end{pmatrix}$$

Chapter 8 Discrete Fourier Transform (DFT), FFT

What is the biggest challenge?

Introduction to discrete Fourier transform

- Time-limited signal made periodic
- Negative aspect and solution of the periodicity

Mathematical approach to the DFT

- Basic concept
- DFT application example

Fast Fourier transform (FFT)

- Matrix formulation - intuitive development

Applications

- Choosing the sampling frequency
- Windowing - Blind FFT output interpretation

Problems

Analog and Digital Signal Processing

INTRODUCTION TO DISCRETE FOURIER TRANSFORM:

Basic concept:

In previous chapters, we dealt with the Fourier Series and the Fourier Transforms of continuous waveforms. When it comes to computing one of them from a real signal (continuous or discretized), we face a practical problem:

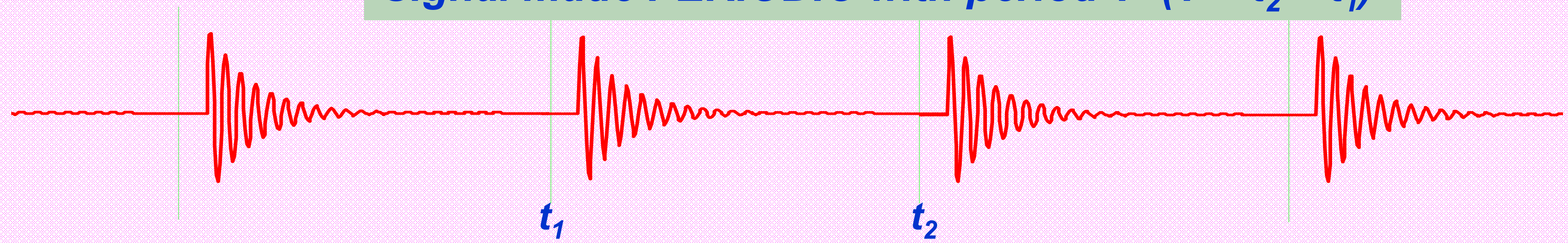
Time limitation!

➔In most situations we exactly know our signal within a time-window starting at t_1 and ending at t_2; however , we totally ignore what happened before t_1 and what will happen after t_2.

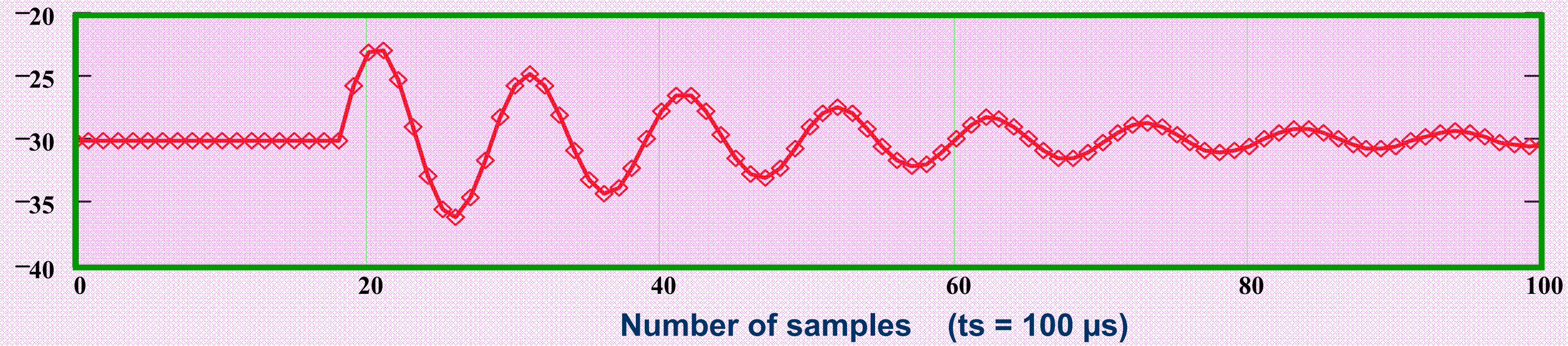
SOLUTION: To make PERIODIC the "Time limited" signal

Example 1:

Signal made PERIODIC with period T ($T = t_2 - t_1$)



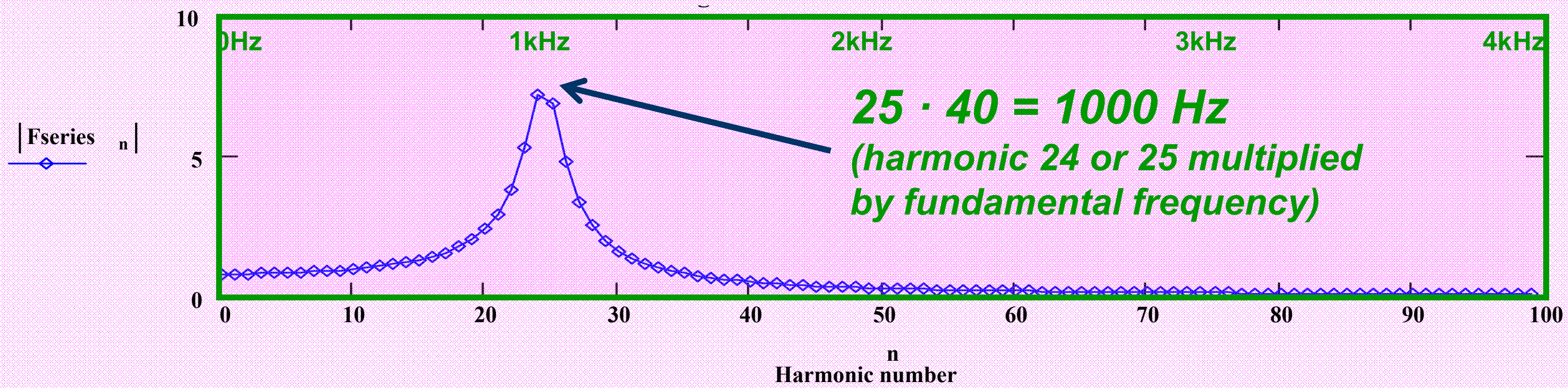
Example 1 cont'

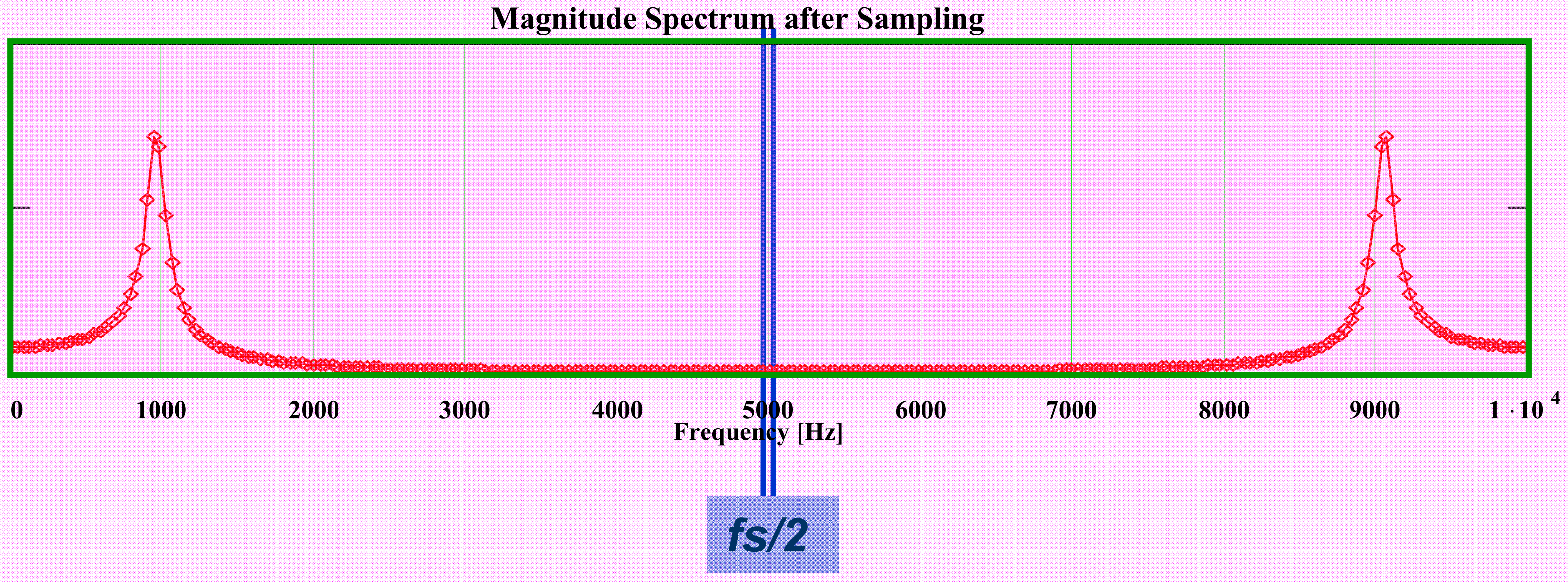


Decaying sine-wave approximate frequency: ***1000 Hz (10 samples/cycle)***

256 samples for 1 period of the periodic signal ➔ ***T = 100µs·256 = 25.6ms (≈ 40Hz)***

Fourier Series coefficient magnitudes:

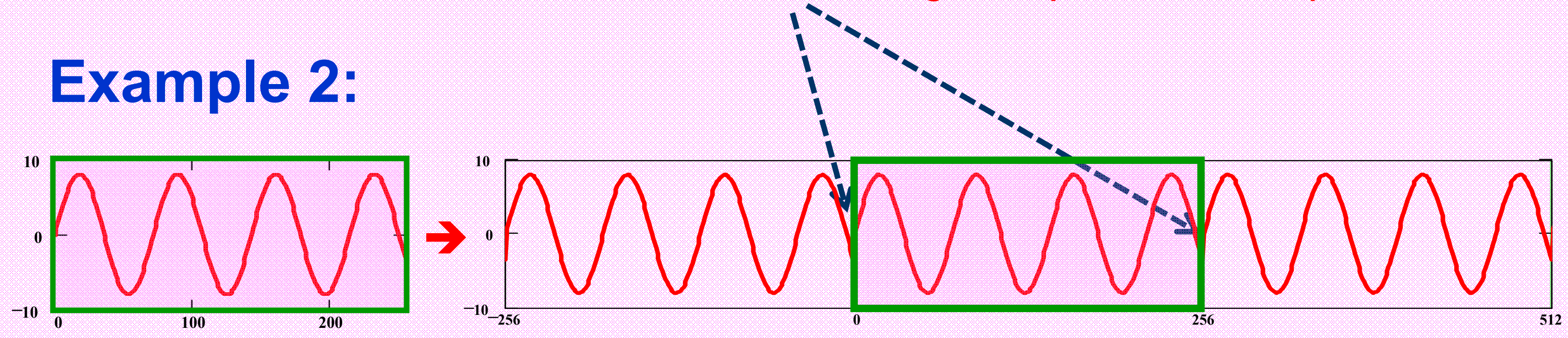




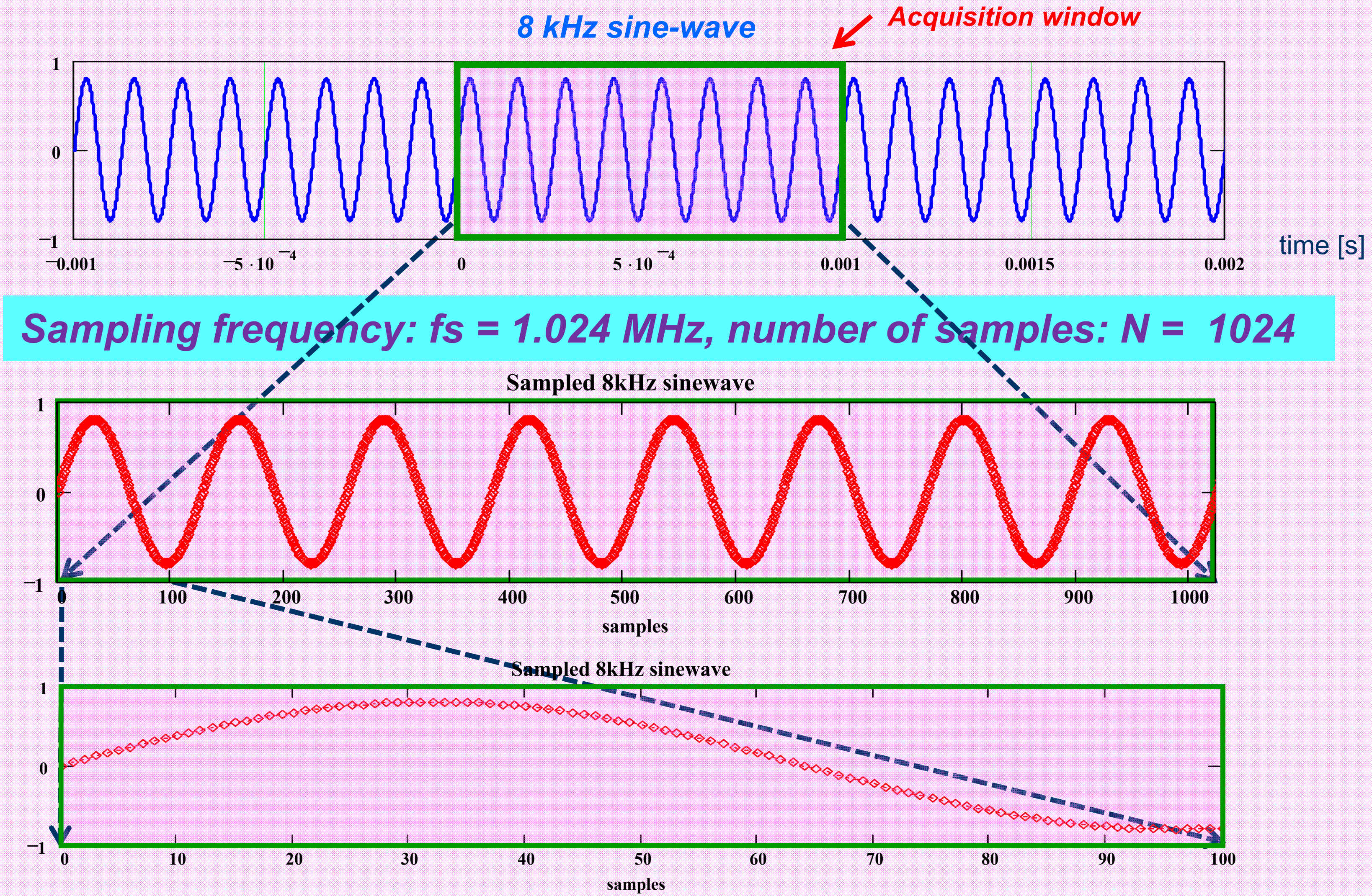
Potential problems due to the time-limited signal made periodic:

Discontinuities ➔ *Generate unwanted high frequencies components*

Example 2:

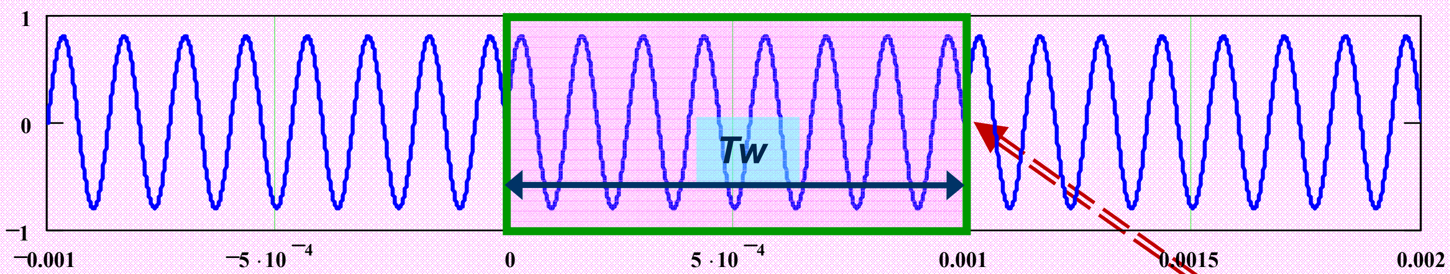


Example 3: *Pure sine-wave a) Integer number of periods*



Example 3: *Pure sine-wave b) Non-integer number of periods*

7.5 kHz sine-wave

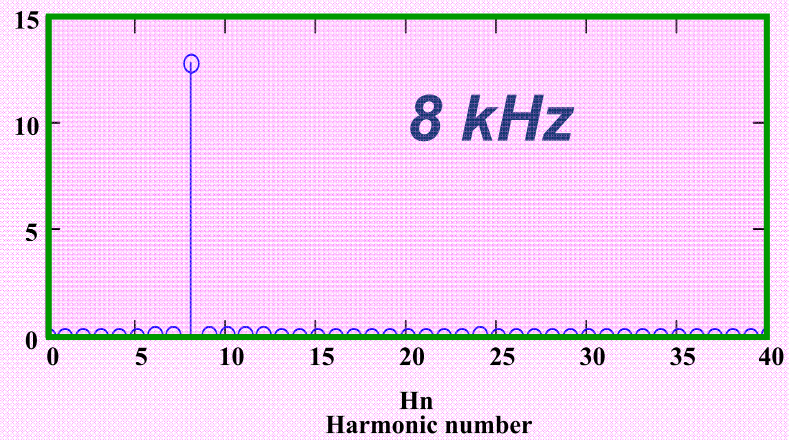


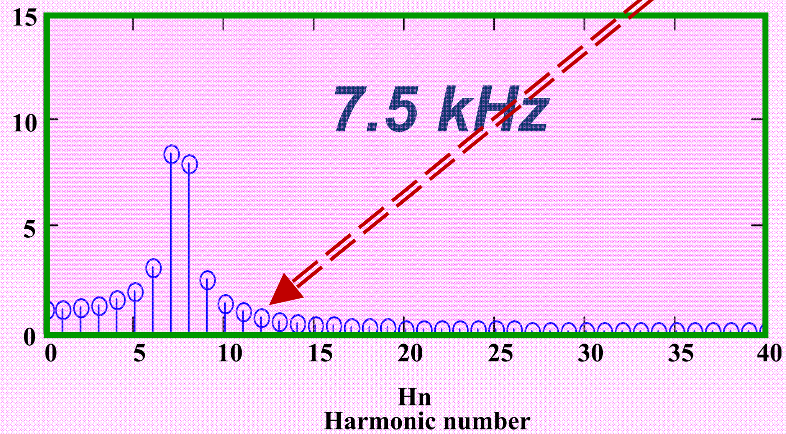
Tw: acquisition time window, ts: sampling interval ➔ **$Tw = N \cdot ts = N / fs$**

➔ Fundamental frequency: **$\Delta f = 1 / Tw = fs / N = 1$ kHz**

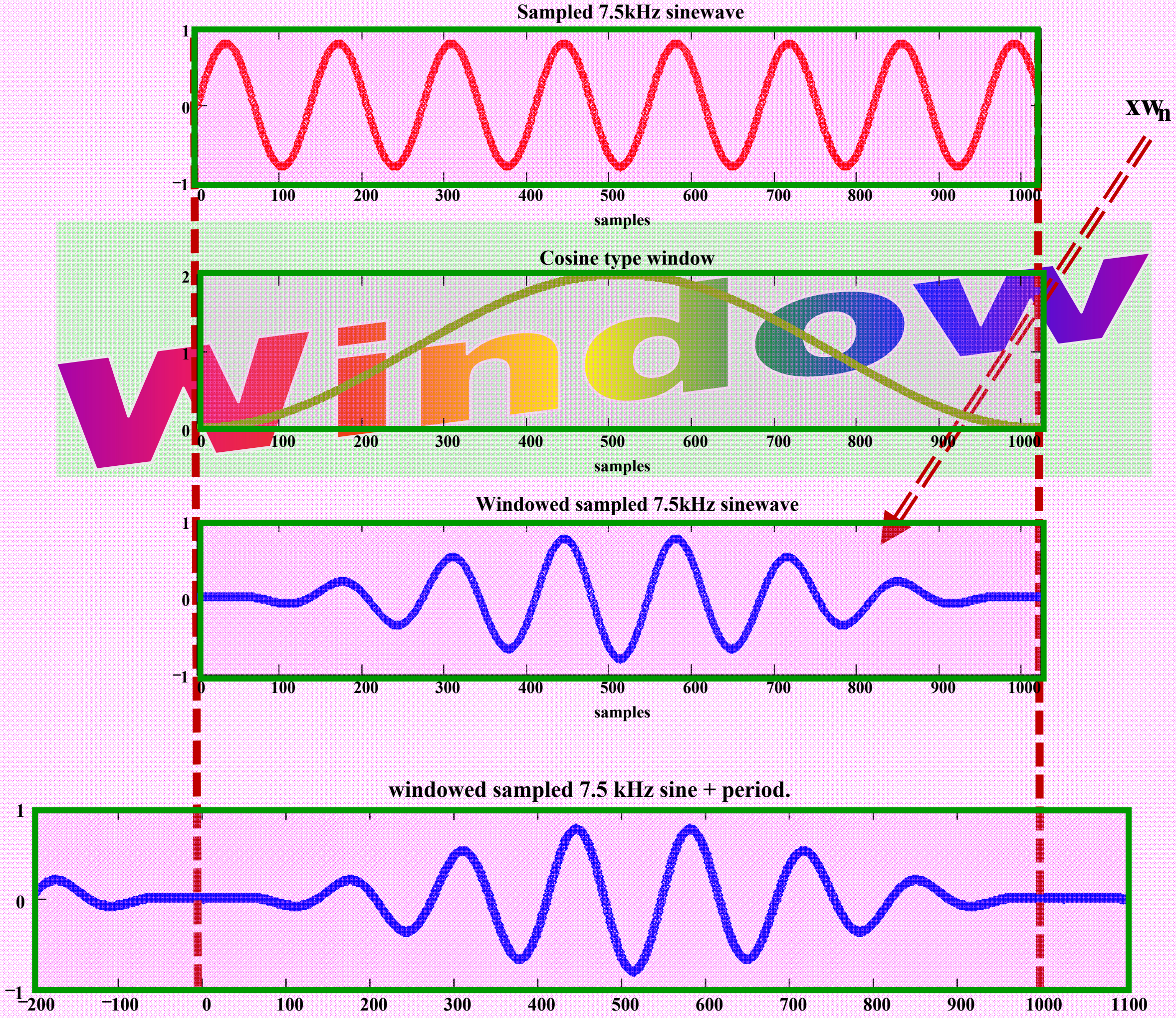
Discontinuities in the time-domain ➔ many spectral components

Magnitude of Fourier Coefficients



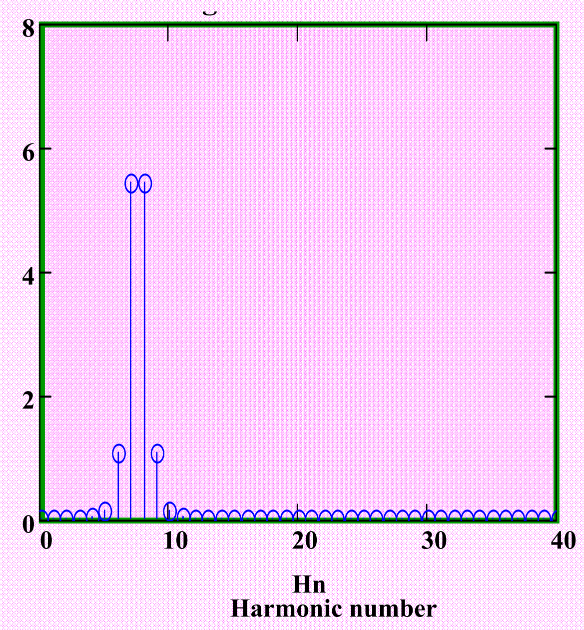


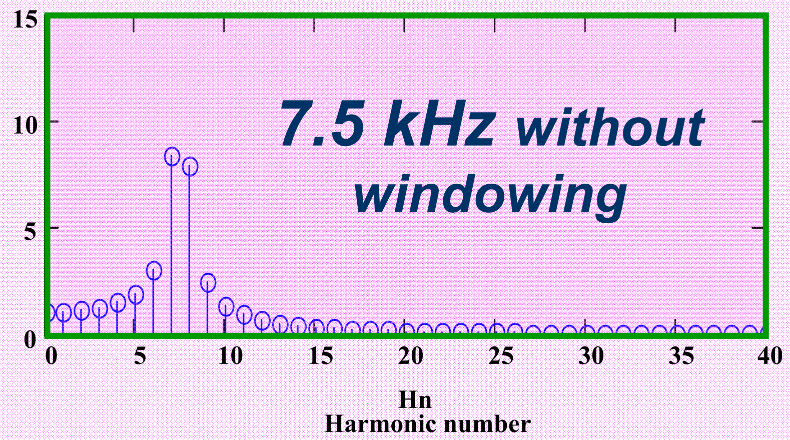
SOLUTION: WINDOWING

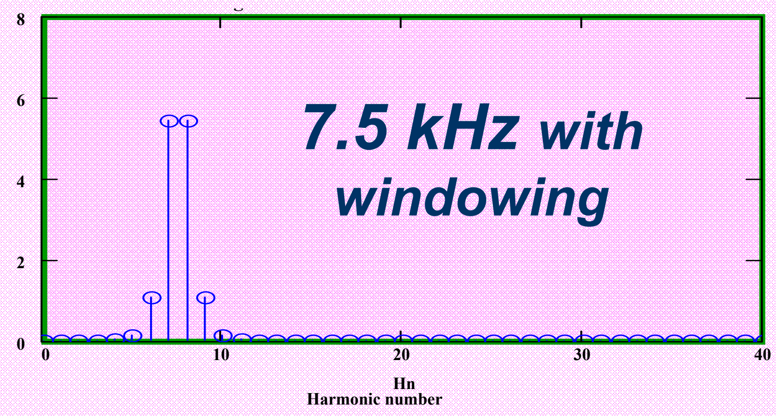


$$xw_n := A \cdot \sin\left(\frac{2 \cdot \pi}{128} \cdot n \cdot \frac{7500}{8000}\right) 0.5 \cdot \left(1 + \cos\left(\frac{2 \cdot \pi \cdot n}{1024} + \pi\right)\right)$$

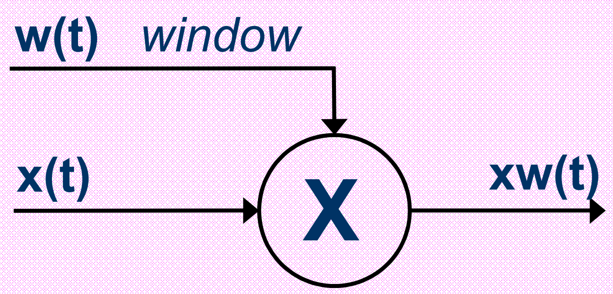
Magnitude of Fourier Coefficients







WINDOWING DOWNSIDE:



$x(t) \cdot w(t) = xw(t)$

➔ $X(j\omega) * W(j\omega) = XW(j\omega)$

Frequency domain convolution of the "window spectrum" with the signal spectrum!

➔ ***Broadening of the signal spectrum***

MATHEMATICAL APPROACH OF THE DFT

Basic concept:

If f(t) is our continuous waveform, fd(t) is its discretized version:

$$f(t) \rightarrow \quad fd(t) = \sum_{n=-\infty}^{\infty} f(n \cdot ts) \cdot \delta(t - n \cdot ts)$$

Where ***ts*** is the time between consecutive samples. By definition, the Fourier Transform of fd(t) is:

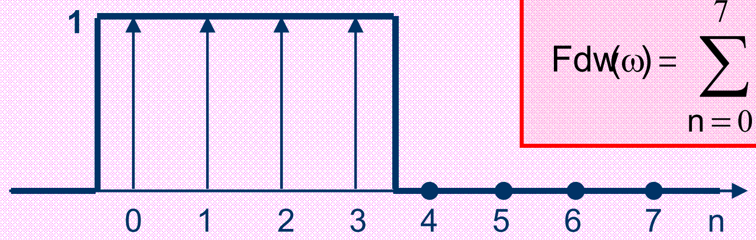
$$F(fd(t)) = Fd(\omega) = \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(n \cdot ts) \cdot \delta(n - n \cdot ts) \cdot e^{-j \cdot \omega \cdot t} \, dt$$

$$Fd(\omega) = \sum_{n=-\infty}^{\infty} f(n \cdot ts) \cdot e^{-j \cdot \omega \cdot n \cdot ts}$$

Previously, we introduced a time-limitation. The simplest way to do this is to set the lower bound of the summation at 0 and the upper bound at N-1, thus limiting the total number of samples considered to N. Rewriting $Fd(\omega)$ gives:

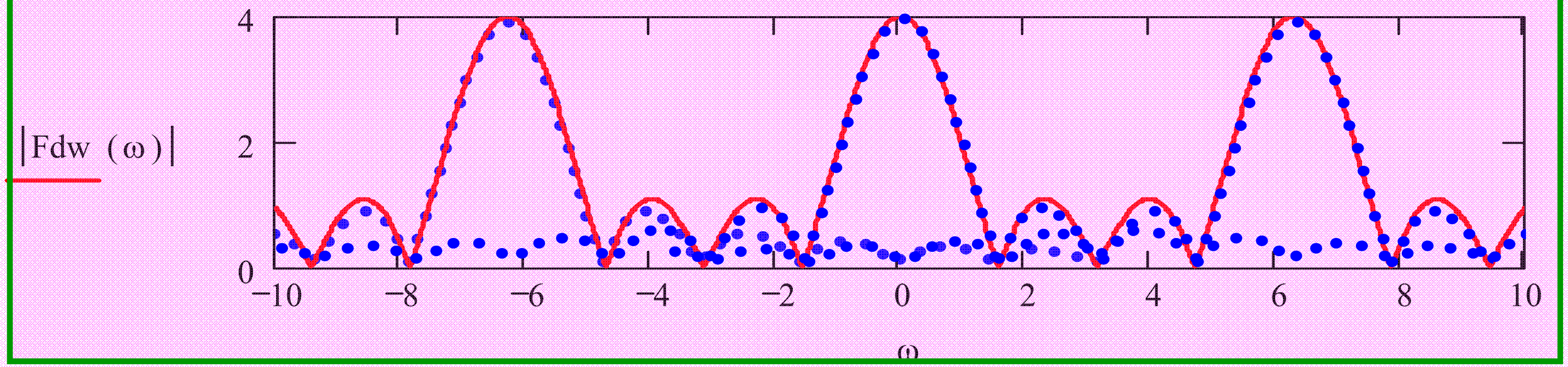
$$Fdw(\omega) = \sum_{n=0}^{N-1} f(n \cdot ts) \cdot e^{-j \cdot \omega \cdot n \cdot ts} \qquad \textbf{WDFT}$$

The letter W used to characterize this new definition refers to *'window'* since our discretized waveform f(n·ts) is effectively *windowed*, that is, multiplied by a time-window function. In order to illustrate the implication of the WDFT, consider a pulse represented by 8 equally spaced samples (N=8):



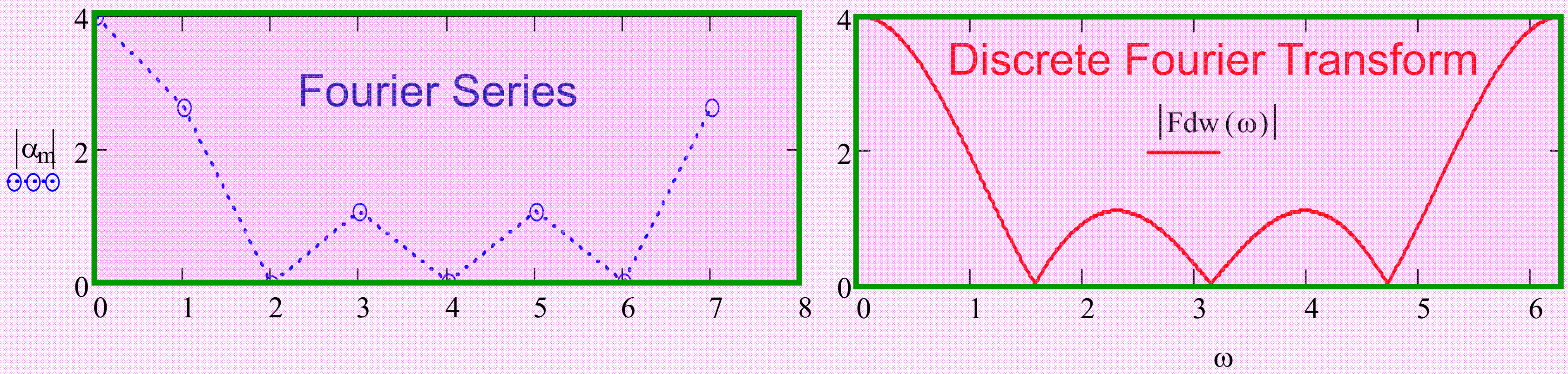
$$Fdw(\omega) = \sum_{n=0}^{7} f(n \cdot ts) \cdot e^{-j\omega n ts} = 1 \cdot e^{-j\omega 0 ts} + 1 \cdot e^{-j\omega 1 ts} + 1 \cdot e^{-j\omega 2 ts} + 1 \cdot e^{-j\omega 3 ts}$$

From this set of samples, we get the following spectrum (with ***ts = 1***) :



The periodicity in ω comes from the sampling theorem (period: 2π/ts = ω_s). Since the theoretical Power Spectrum (from its Fourier transform) of a pulse expend from -∞ to + ∞, that produces a noticeable aliasing **(blue)** !

If the initial waveform is artificially made periodic with a period of 8ts, then we can compute the Fourier coefficients by using the definition of the Fourier series. The next figures compare the spectrum obtained by the two approaches: *Fourier Series Coefficients* and *Discrete Fourier Transform*.



It can be easily shown that the Fourier Series Coefficients can be obtained by simply sampling the Discrete Fourier Transform.

What is commonly referred to as the Discrete Fourier Transform (DFT) is, in reality, a ***Discrete-Time Fourier Series (DTFS)*** derived after transforming a limited number of samples (8 in our example) into a discrete periodic waveform. The fundamental frequency of this new signal is **1/(N·ts)**. Thus, the spectral lines will appear at **m/(N·ts)** Hz where m is an integer.

Analog and Digital Signal Processing

So finally, we can write the DTFS or DFT in the most common form as:

$$F\left(\frac{2\cdot\pi\cdot m}{N\cdot ts}\right) = F(m) = \sum_{n=0}^{N-1} f(n\cdot ts)\cdot e^{\frac{-j\cdot 2\cdot\pi\cdot m\cdot n}{N}}$$

DFT

The inverse DFT is:

$$f(n\cdot ts) = \frac{1}{N}\cdot\sum_{m=0}^{N-1} F(m)\cdot e^{\frac{j\cdot 2\cdot\pi\cdot m\cdot n}{N}}$$

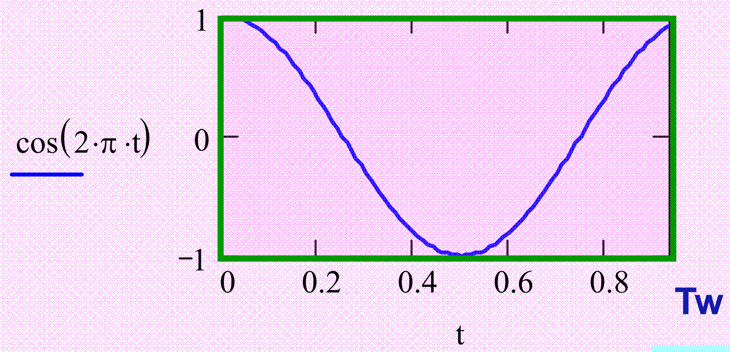
InvDFT

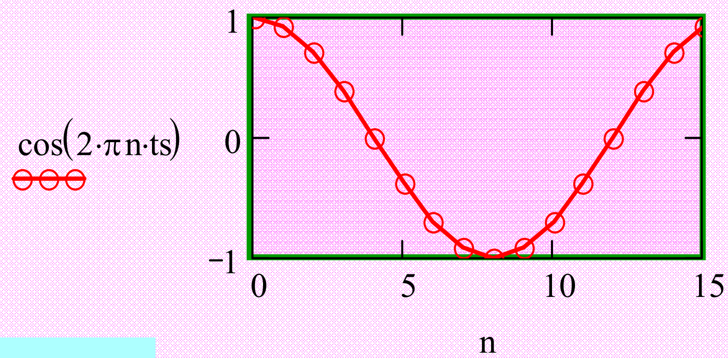
As already mentioned in the beginning of this chapter, due to the limited number of samples used in the computation of the WDFT, numerous problems can appear that one has to be aware of.

SIMPLE DFT APPLICATIONS #1 (integer number of cycle)

f(t) = cos(2πt) ; a cosine function with a period of 1s, sampled every ***ts*** second, ***Tw = ts· (N-1)***

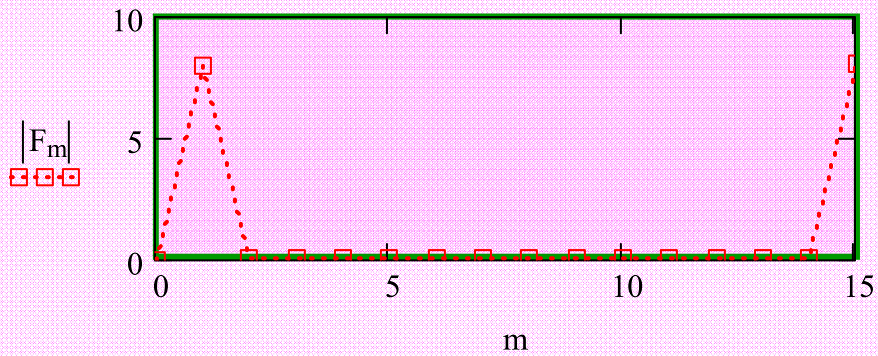
N = 16, ts = 1/16 ➔ T = 1s





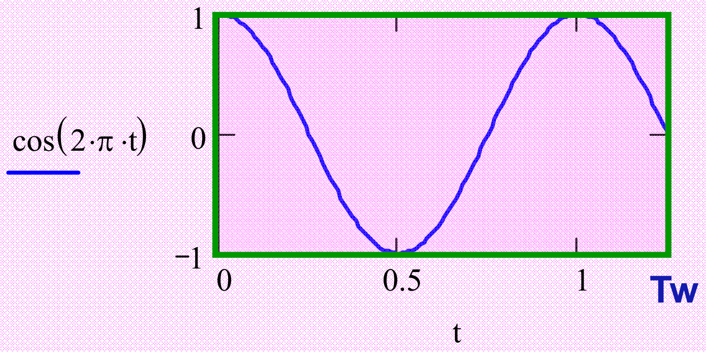
t → n · ts

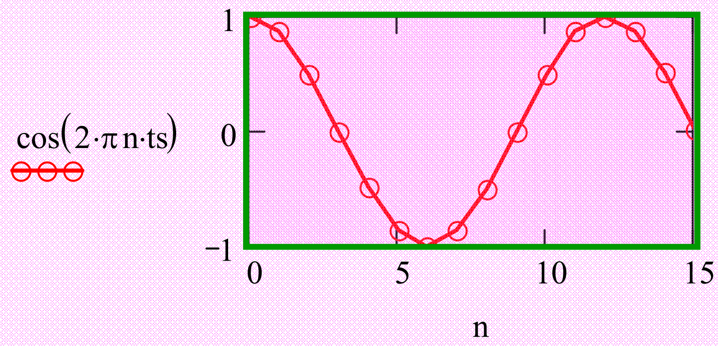
$$F_m := \sum_{n=0}^{N-1} \cos(2\cdot\pi\cdot n\cdot ts)\cdot e^{\frac{-j\cdot 2\cdot\pi\cdot m\cdot n}{N}}$$



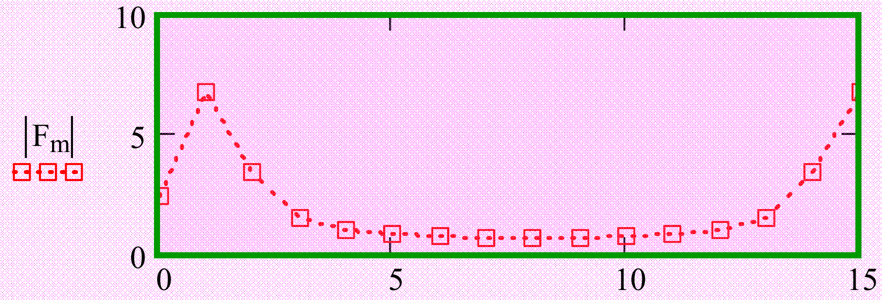
SIMPLE DFT APPLICATIONS #2 (non-integer number of cycle)

$f(t) = \cos(2\pi t)$; **N = 16, ts = 1/12 ➔ Tw = 1.25s**





$$F_m := \sum_{n=0}^{N-1} \cos(2\cdot\pi\cdot n\cdot ts)\cdot e^{\frac{-j\cdot 2\cdot\pi\cdot m\cdot n}{N}}$$



Since the number of periods used to compute the WDFT is ***not an integer number***, a discontinuity in the time domain is created, thus generating ***artifacts*** (unwanted high frequencies spectral components) in the power spectrum.

In order to solve this problem, various "window" types have been proposed, each one of them with particular benefits and draw-backs!! The optimum choice will depend upon the features one wishes to emphasize.

FAST FOURIER TRANSFORM (FFT)

The FFT is simply an algorithm that can compute the discrete Fourier transform much more rapidly than other available algorithms. In the case of the DFT, the approximate number of multiplication grows with the ***square of N***. If N can be written as 2^K (where K is an integer), a substantial saving can be achieved on the number of multiplication which becomes in the order of : ***2 N log₂ N*** → $2N\log_2 N$. Many algorithms have been designed and the interested reader is encouraged to refer to one of the classical text books covering this topic. A simple matrix factoring example is used to intuitively justify the FFT algorithm.

Matrix formulation

Consider the discrete Fourier transform: **8-1**

$$F(m) = \sum_{n=0}^{N-1} f_0(n)\cdot e^{\frac{-j\cdot 2\cdot\pi\cdot m\cdot n}{N}} \qquad m := 0..N-1$$

(8-1) describes the computation of N equations. Let N = 4 and **8-2**

$$W = e^{\frac{-j\cdot 2\cdot\pi\cdot m\cdot n}{N}}$$

➔ **8-3**

$$F(0) = f_0(0)\cdot W^0 + f_0(1)\cdot W^0 + f_0(2)\cdot W^0 + f_0(3)\cdot W^0$$
$$F(1) = f_0(0)\cdot W^0 + f_0(1)\cdot W^1 + f_0(2)\cdot W^2 + f_0(3)\cdot W^3$$
$$F(2) = f_0(0)\cdot W^0 + f_0(1)\cdot W^2 + f_0(2)\cdot W^4 + f_0(3)\cdot W^6$$
$$F(3) = f_0(0)\cdot W^0 + f_0(1)\cdot W^3 + f_0(2)\cdot W^6 + f_0(3)\cdot W^9$$

➔ **8-4**

$$\begin{pmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \end{pmatrix} = \begin{pmatrix} W^0 & W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 \\ W^0 & W^2 & W^4 & W^6 \\ W^0 & W^3 & W^6 & W^9 \end{pmatrix} \cdot \begin{pmatrix} f_0(0) \\ f_0(1) \\ f_0(2) \\ f_0(3) \end{pmatrix}$$

(8-4) reveals that since **W** and $\mathbf{f_o(n)}$ are complex matrices, thus N^2 complex multiplication and N (N-1) complex additions are necessary to perform the required matrix computation.

INTUITIVE DEVELOPMENT

To illustrate the FFT algorithm, it is convenient to choose the number of sample points of $f_o(n)$ according to the relation **$N = 2^\gamma$** , where γ is an integer. The first step in developing the FFT algorithm for this example is to rewrite (8-4) as:

8-5

$$\begin{pmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & W^1 & W^2 & W^3 \\ 1 & W^2 & W^0 & W^2 \\ 1 & W^3 & W^2 & W^1 \end{pmatrix} \cdot \begin{pmatrix} f_0(0) \\ f_0(1) \\ f_0(2) \\ f_0(3) \end{pmatrix}$$

Matrix Eq. (8-5) was derived from (8-4) by using the relationship $\mathbf{W^{m \cdot n} = W^{m \cdot n \bmod (N)}}$. Than: ***8-6*** $\mathbf{W^6 = W^2}$

The next step is to factor the square matrix in (8-5) as follows:

8-7

$$\begin{pmatrix} F(0) \\ F(2) \\ F(1) \\ F(3) \end{pmatrix} = \begin{pmatrix} 1 & W^0 & 0 & 0 \\ 1 & W^2 & 0 & 0 \\ 0 & 0 & 1 & W^1 \\ 0 & 0 & 1 & W^3 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & W^0 & 0 \\ 0 & 1 & 0 & W^0 \\ 1 & 0 & W^2 & 0 \\ 0 & 1 & 0 & W^2 \end{pmatrix} \cdot \begin{pmatrix} f_0(0) \\ f_0(1) \\ f_0(2) \\ f_0(3) \end{pmatrix}$$

The method of factorization is based on the theory of the FFT algorithm. It is easily shown that multiplication of the two matrices of (8-7) yields the square matrix of (8-5) with the exception that rows 1 and 2 have been interchanged (the rows are numbered 0, 1, 2 and 3).

Having accepted the fact that (8-7) is correct, although the result is *scrambled*, one should then examine the number of multiplication required to compute the equation. First let:

8-8

$$\begin{pmatrix} f_1(0) \\ f_1(1) \\ f_1(2) \\ f_1(3) \end{pmatrix} = \begin{pmatrix} 1 & 0 & W^0 & 0 \\ 0 & 1 & 0 & W^0 \\ 1 & 0 & W^2 & 0 \\ 0 & 1 & 0 & W^2 \end{pmatrix} \cdot \begin{pmatrix} f_0(0) \\ f_0(1) \\ f_0(2) \\ f_0(3) \end{pmatrix}$$

Element $f_1(0)$ is computed by one complex multiplication and one complex addition.

8-9

$$f_1(0) = f_0(0) + W^0 \cdot f_0(2)$$

Element $f_1(1)$ is also determined by one complex multiplication and one complex addition. Only one complex addition is required to compute $f_1(2)$. This from the fact that $\mathbf{W^0 = -W^2}$; hence:

8-10

$$f_1(2) = f_0(0) + W^2 \cdot f_0(2) = f_0(0) - W^0 \cdot f_0(2)$$

where the complex multiplication $W^0\, f_0(2)$ has already been computed in the determination of $f_1(0)$. By the same reasoning, $f_1(3)$ is computed by only one complex addition and no multiplications. The intermediate vector $\mathbf{f_1(n)}$ is then determined by four complex additions and two complex multiplications. Let us continue by completing the computation of (8-7):

8-11

$$\begin{pmatrix} F(0) \\ F(2) \\ F(1) \\ F(3) \end{pmatrix} = \begin{pmatrix} f_2(0) \\ f_2(1) \\ f_2(2) \\ f_2(3) \end{pmatrix} = \begin{pmatrix} 1 & W^0 & 0 & 0 \\ 1 & W^2 & 0 & 0 \\ 0 & 0 & 1 & W^1 \\ 0 & 0 & 1 & W^3 \end{pmatrix} \cdot \begin{pmatrix} f_1(0) \\ f_1(1) \\ f_1(2) \\ f_1(3) \end{pmatrix}$$

Term $f_2(0)$ is computed by one complex multiplication and addition!

8-12

$$f_2(0) = f_1(0) + W^0 \cdot f_1(1)$$

Element $f_2(1)$ is computed by one addition because $W^0 = -W^2$. By similar reasoning, $f_2(2)$ is determined by one complex multiplication and addition, and $f_2(3)$ by only one addition. Thus:

- Computation of **F(m)** by means of Eq. (8-7) requires a total of:

four complex multiplication and eight complex additions

- Computation of **F(m)** by (8-4) requires:

sixteen complex multiplication and twelve complex additions

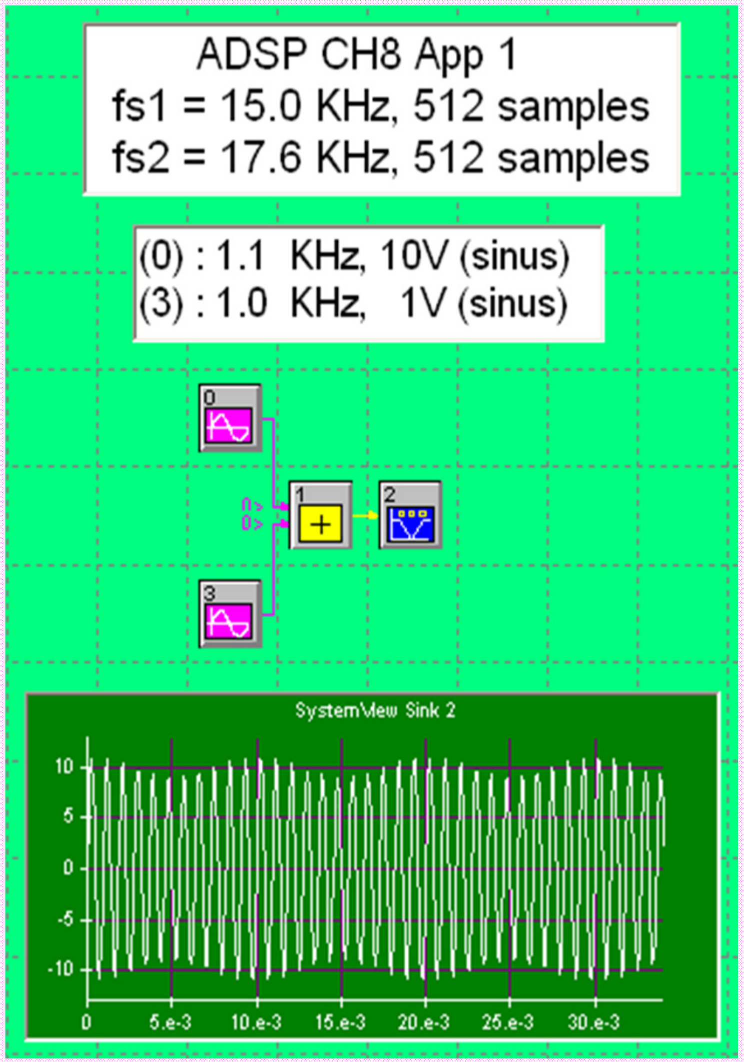
However, the matrix factoring procedure does introduce one discrepancy. That is:

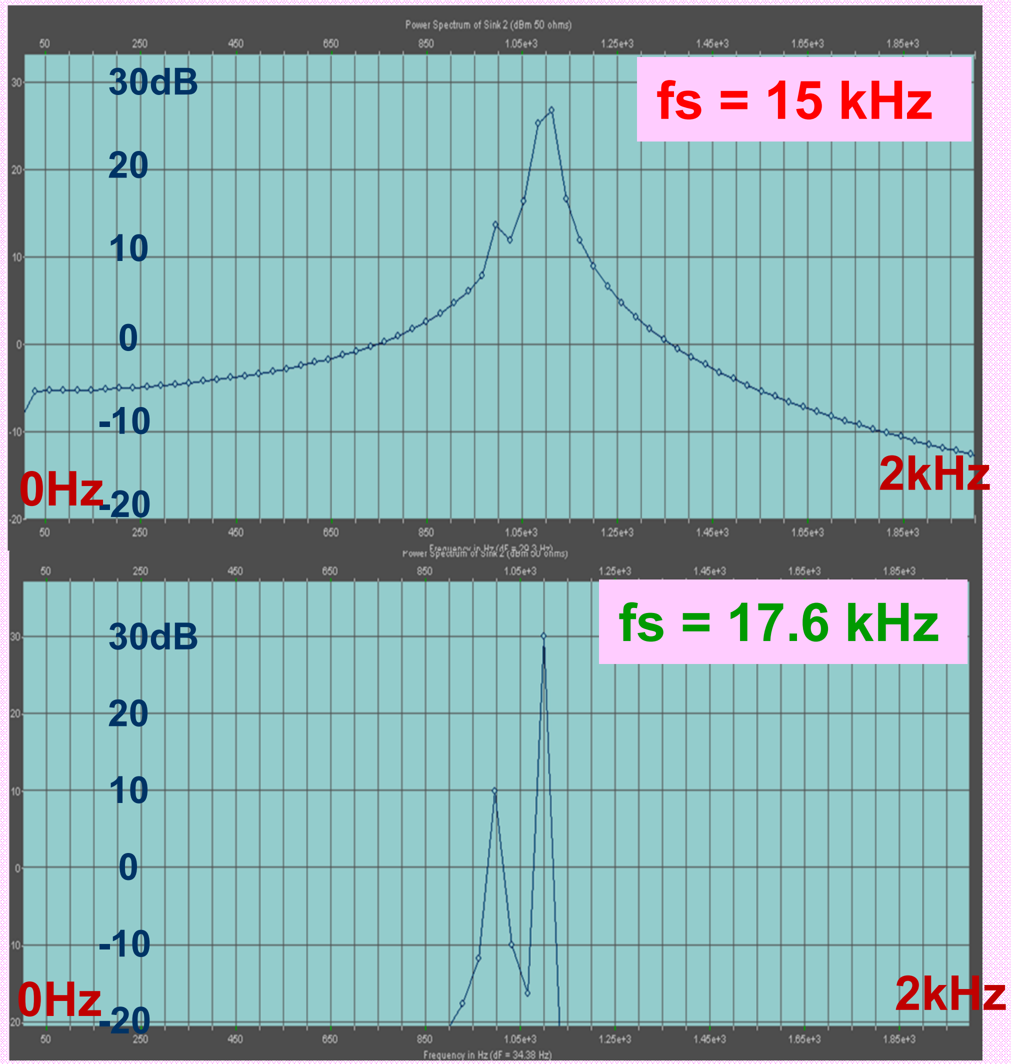
8-13 $$\mathrm{Fscrab(m)} = \begin{pmatrix} F(0) \\ F(2) \\ F(1) \\ F(3) \end{pmatrix} \quad \text{instead of} \quad F(m) = \begin{pmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \end{pmatrix}$$

This rearrangement is inherent in the matrix factoring process and is a minor problem because it is straightforward to generalize a technique for unscrambling Fscrab(m) and obtain F(m).

Thus, if N can be written as 2^K (where K is an integer), a substantial saving can be achieved on the number of multiplication which becomes in the order of : *$2\ N \log_2 N$.*

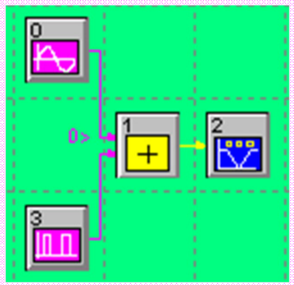
APPLICATIONS #1 CHOOSING THE SAMPLING FREQUENCY



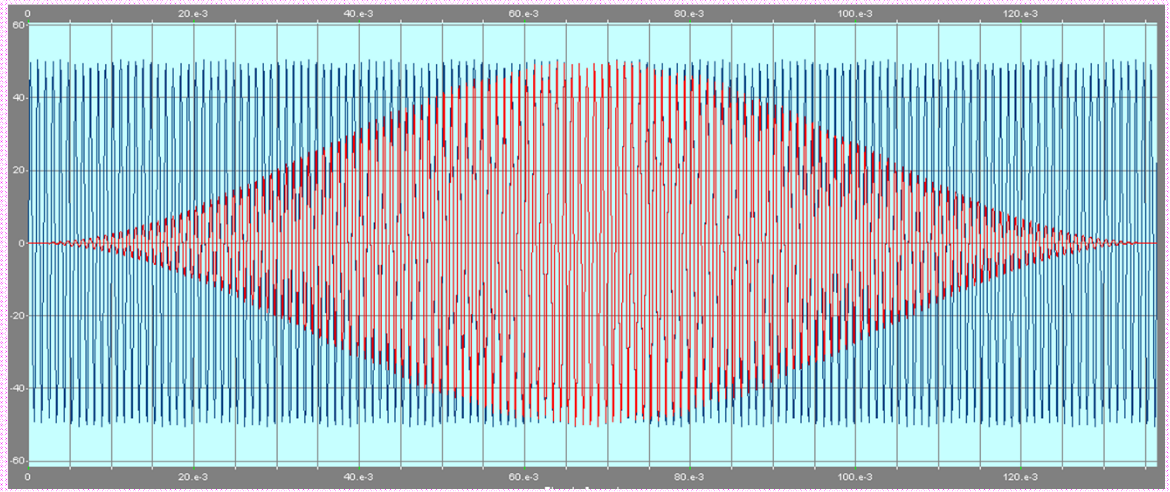


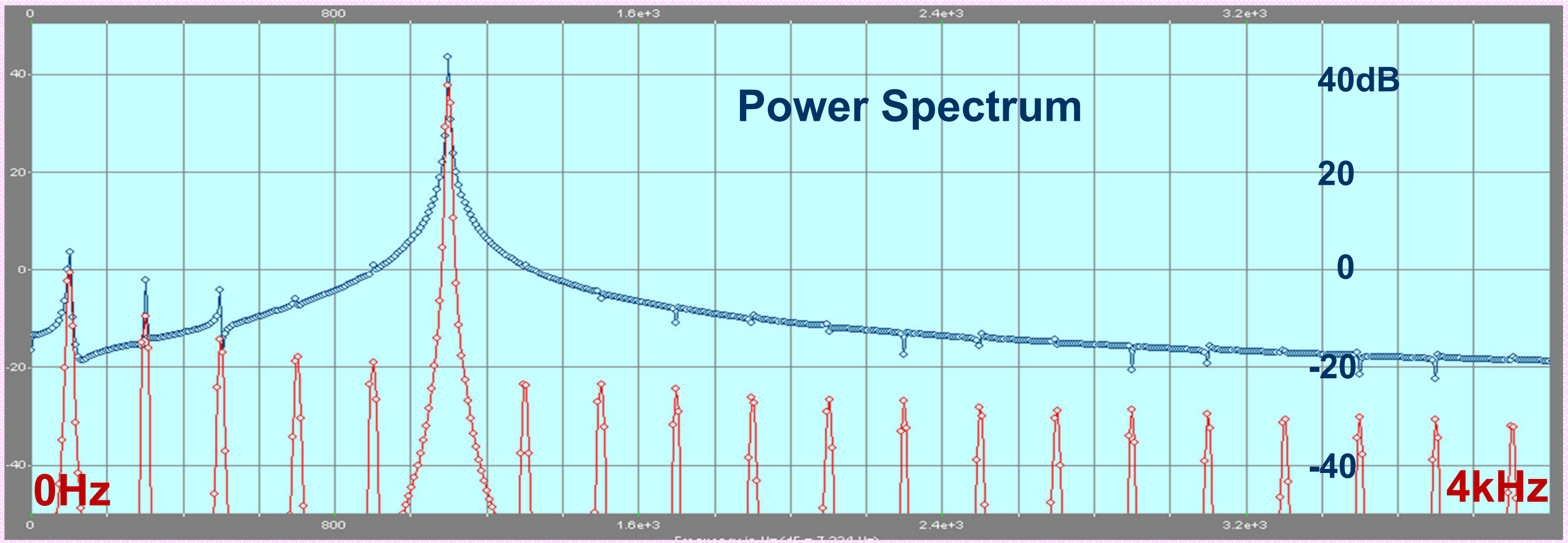
APPLICATIONS #2 WINDOWING

fs = 15 kHz, 2048 samples
(0) 1.1 kHz, 50V (sinus)
(3) 100Hz, 1V (square-wave)



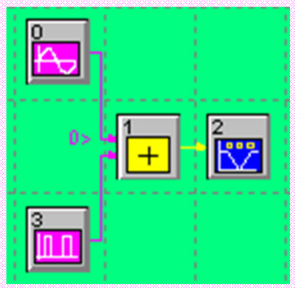
No-window after windowing (Hanning)

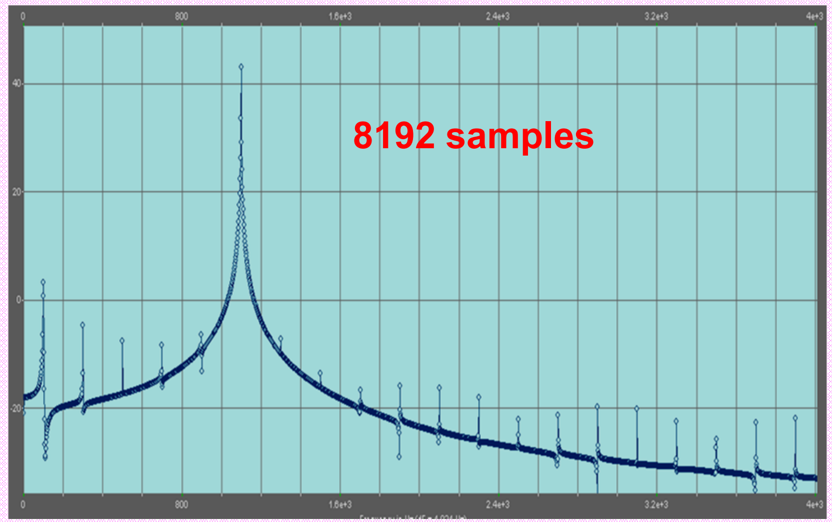


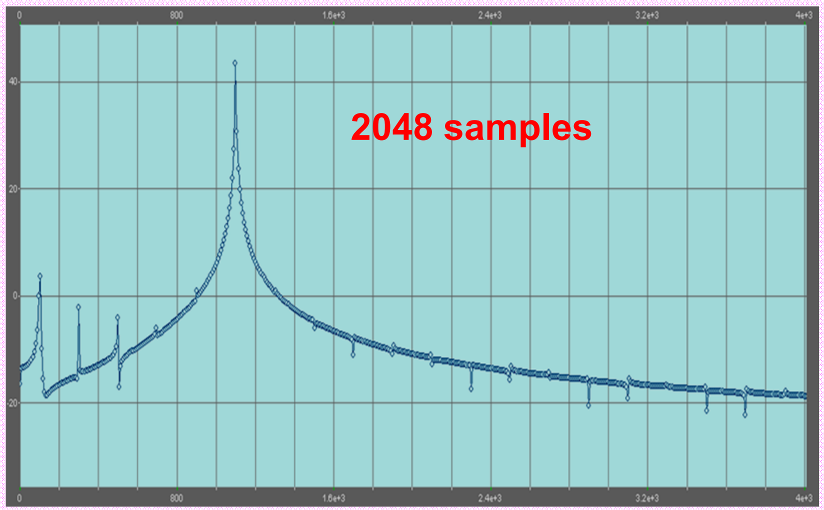


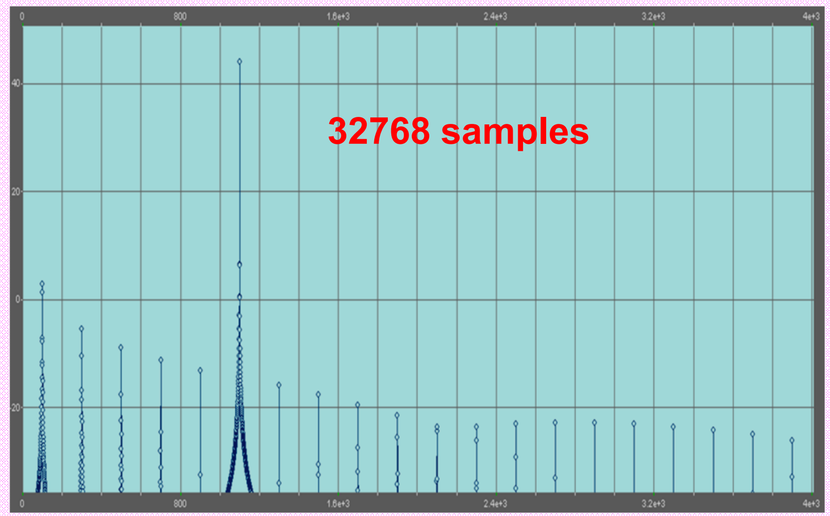
APPLICATIONS #3 ACQUISITION WINDOW (Tw) LENGTH

fs = 15 kHz, **X samples**
(0) 1.1 kHz, 50V (sinus)
(3) 100Hz, 1V (square-wave)



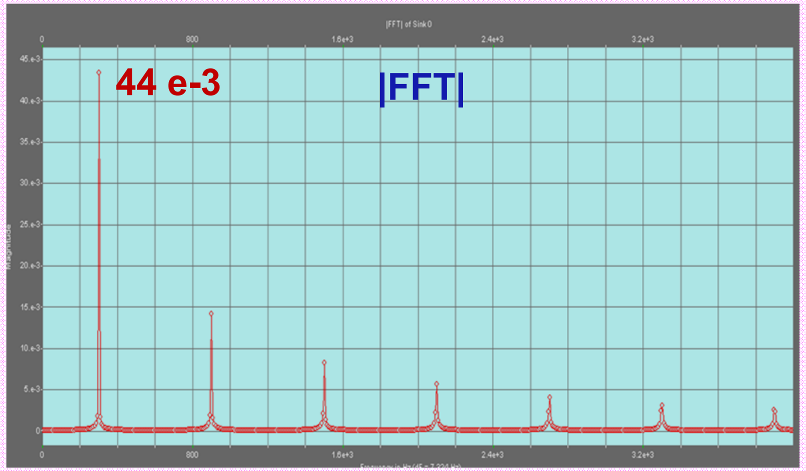


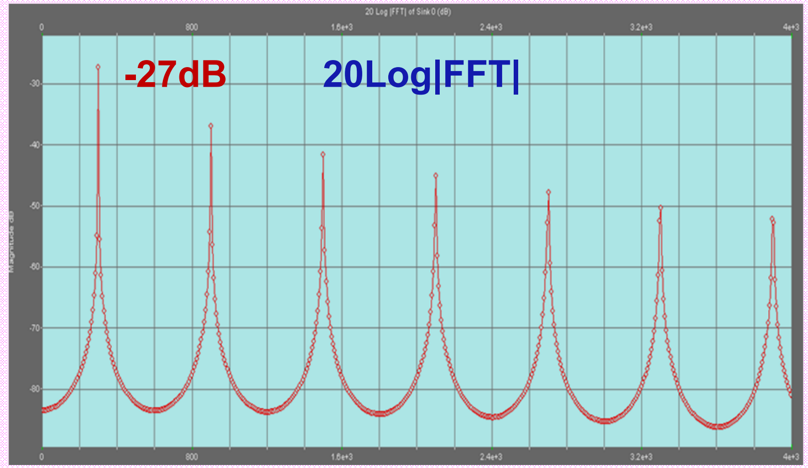


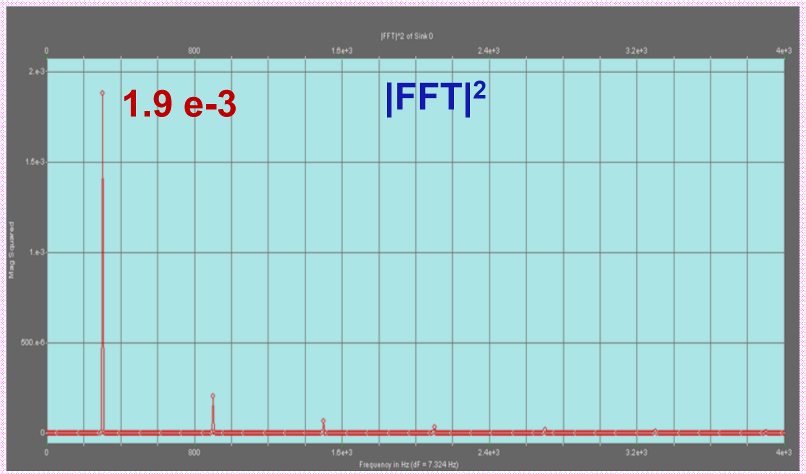


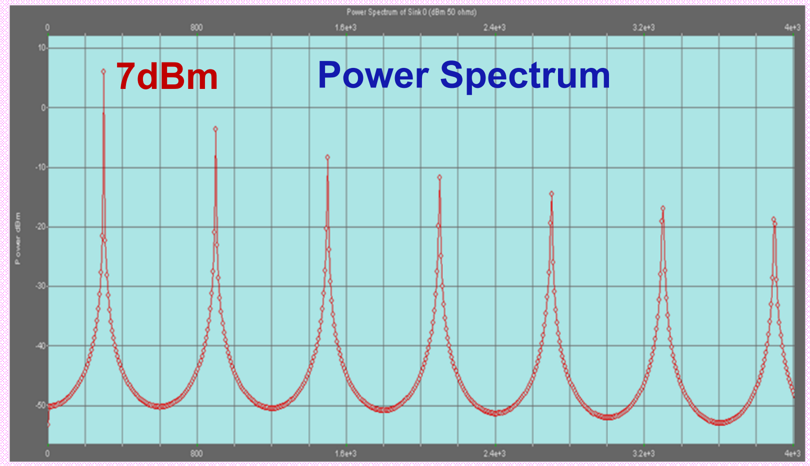
APPLICATIONS #4 |FFT| → |FFT|^2 → Power Spectrum

300 Hz square-wave, 1Vpp - sampling frequency : 15 kHz



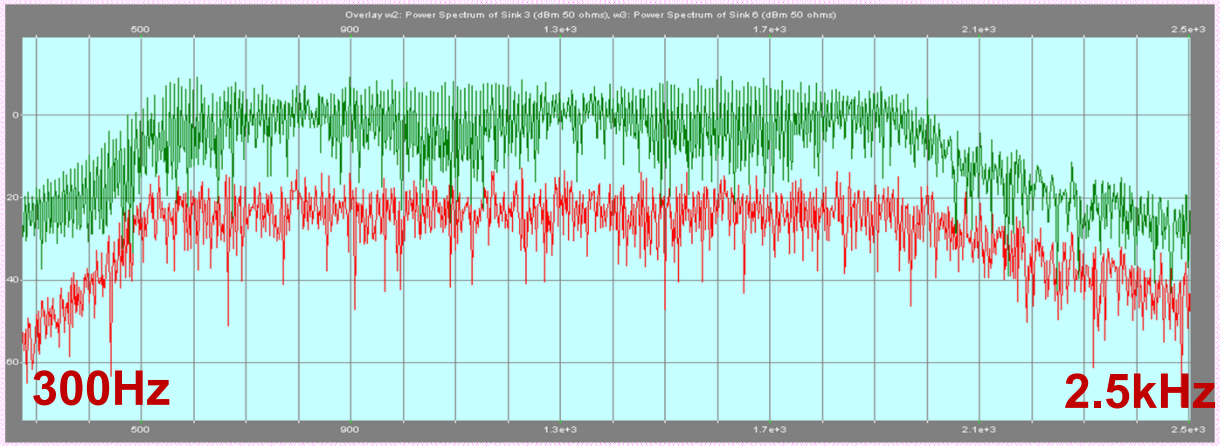




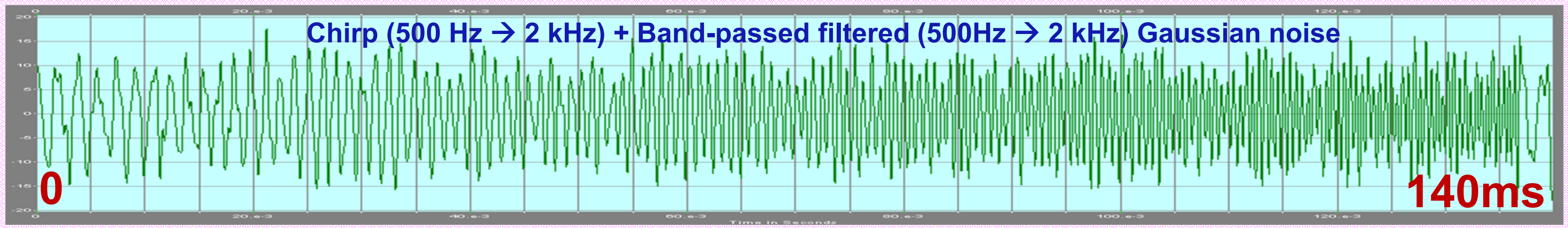


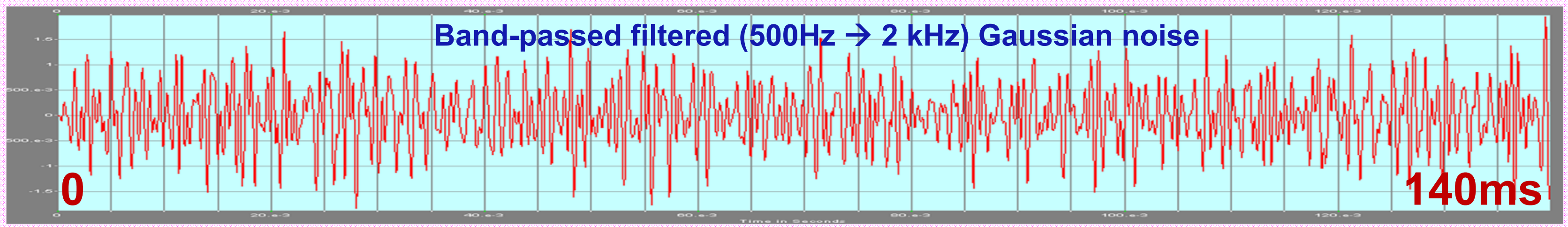
APPLICATIONS #5 BLIND FFT INTERPRETATION

Consider the following Power Spectrum (red and green):



What signals produced these spectrums?





PROBLEMS

Problem 8.1

In p.8-14 , prove that $|F_1| = 8$

Problem 8.2

With $f(nT) = \delta(t - 5T)$ and $N = 16$, determine $|F_m|$ for $0 \leq m < 8$

Problem 8.3

With $N = 8$, compute the module, the real and imaginary part of $F(m)$ for $0 \leq m < 4$

$$f(0) = f(2) = f(4) = f(6) = 1 \quad \text{and} \quad f(1) = f(3) = f(5) = f(7) = -1$$

Problem 8.4 (SystemView)

The input signal x(t) of an acquisition system has the following form :

$x(t) = A \sin(\omega_0 t) + B\ Sqw(t)$, $Sqw(t)$: symmetrical square-wave of 1 ms period

The sampling rate (fs) can be chosen between 50 kHz and 100 kHz and f_0 is smaller then 10 kHz.

Draw several possible FFT plots (magnitude) you may obtain when varying the following parameters:

fs, N, with and without windowing, f_0, A/B

PROBLEMS

Problem 8.5 (SystemView)

The input signal x(t) of an acquisition system has the following form :

$x(t) = A \sin(\omega_0 t) + B \sin(\omega_1 t) + C \sin(\omega_2 t) + D \sin(\omega_3 t)$

with $f_0 = 1$ kHz, $f_1 = f_0 + \Delta f$, $f_2 = f_1 + \Delta f$, $f_3 = f_2 + \Delta f$, $\Delta f = 50$ Hz

A, B C and D can take the following values: +1 or 0 (constant over the sampling window)

Assume that $fs = 1/T = 12$ kHz, how do you choose N?

Problem 8.6

Problem 8.7