Компьютерное моделирование артикуляторных и акустических процессов в естественных языках

Занятие 5

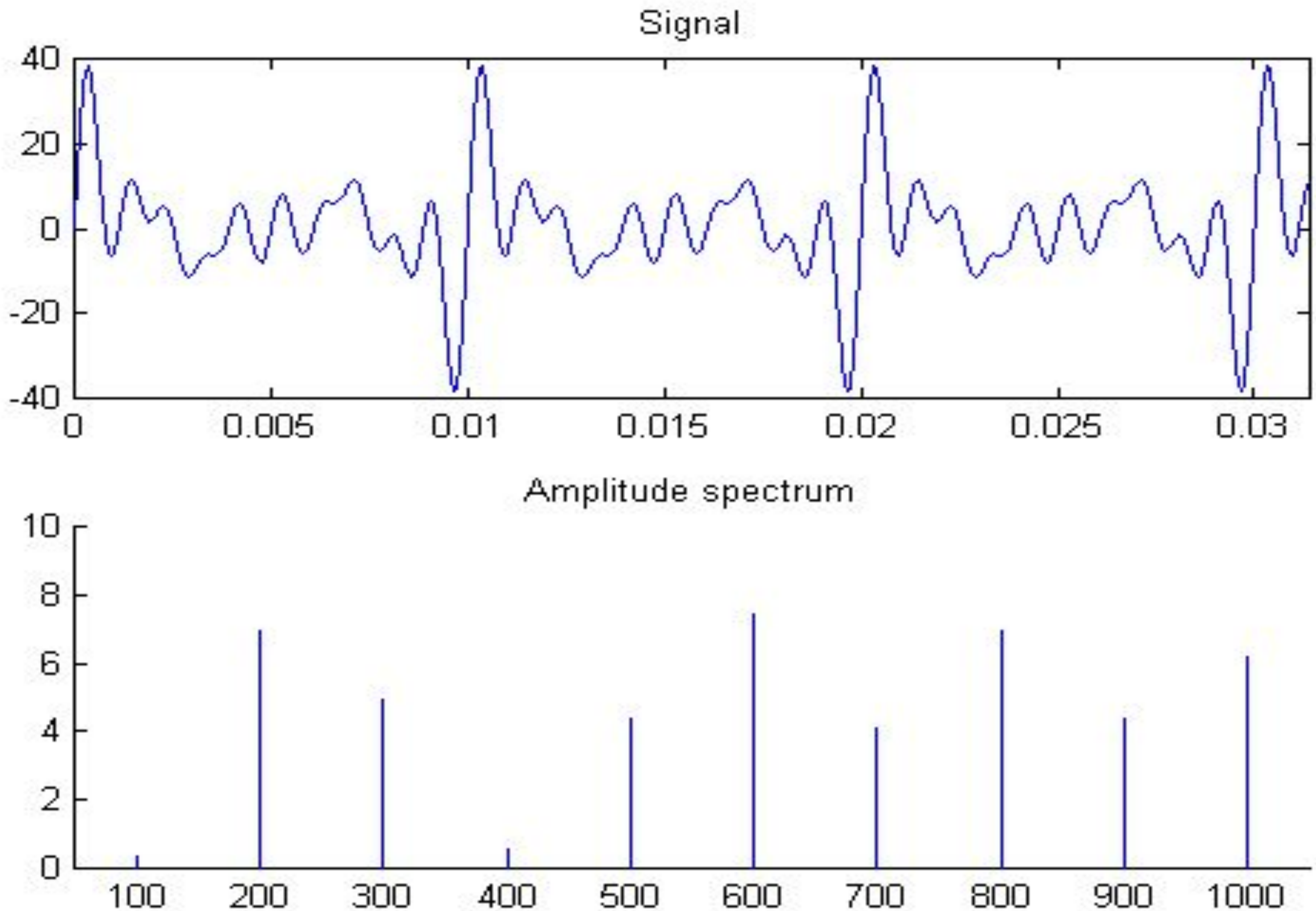
Теорема Фурье

Всякое периодическое колебание частоты *F* можно получить в результате суммирования бесконечного числа гармоник с частотами *F, 2F, 3F, 4F, ...,* и **специально подобранными** амплитудами и фазами

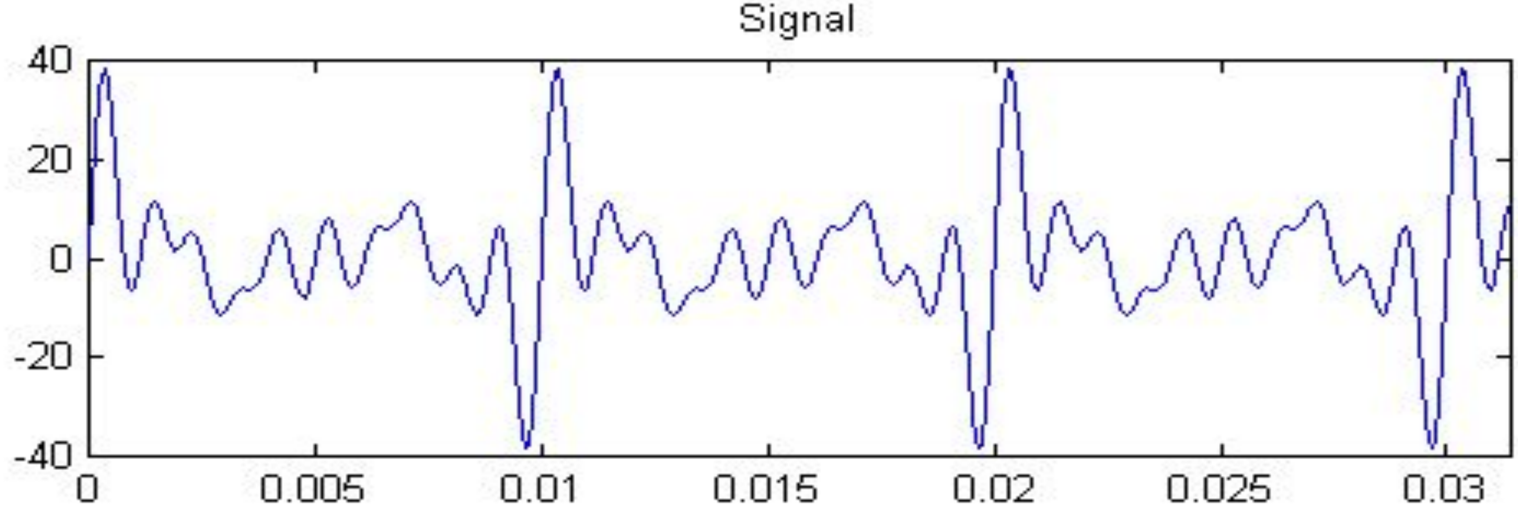
$x(t) = A_0 + A_1 sin(2\pi Ft + \phi_1) + A_2 sin(2\pi 2Ft + \phi_2) + A_3 sin(2\pi 3Ft + \phi_3) + ...$ *(и т.д.) ИЛИ*

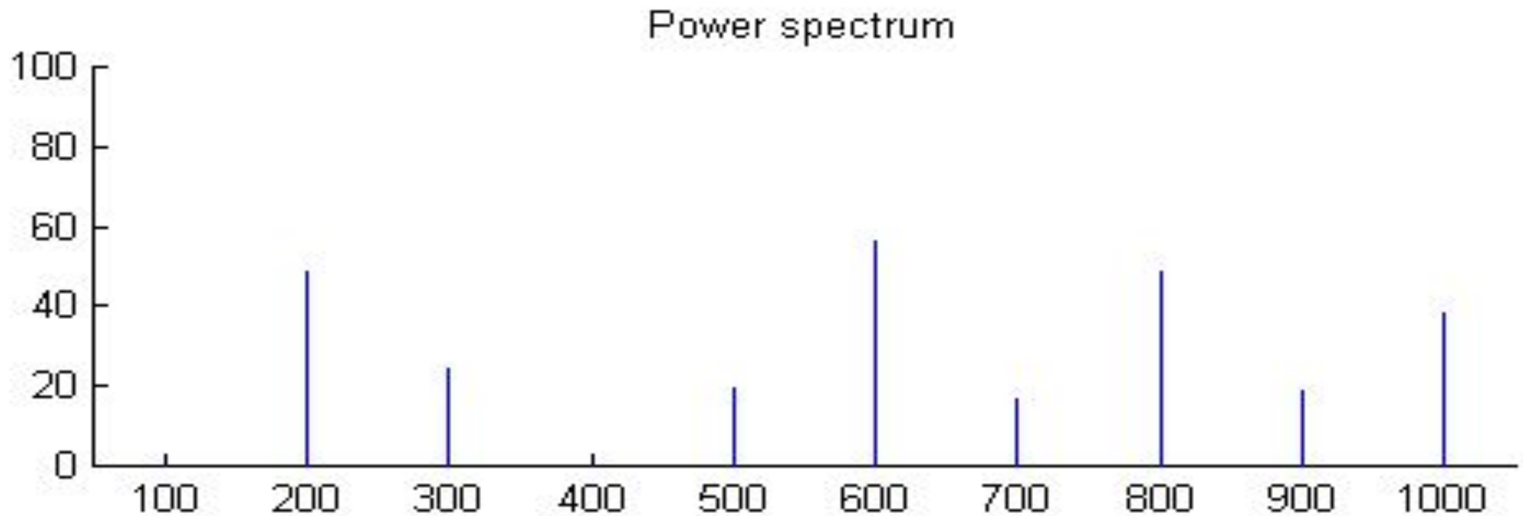
$$x(t) = A_0 + \sum_{k=1}^{\infty} A_k \sin(2\pi kFt + \varphi_k)$$

Амплитудно-частотный спектр

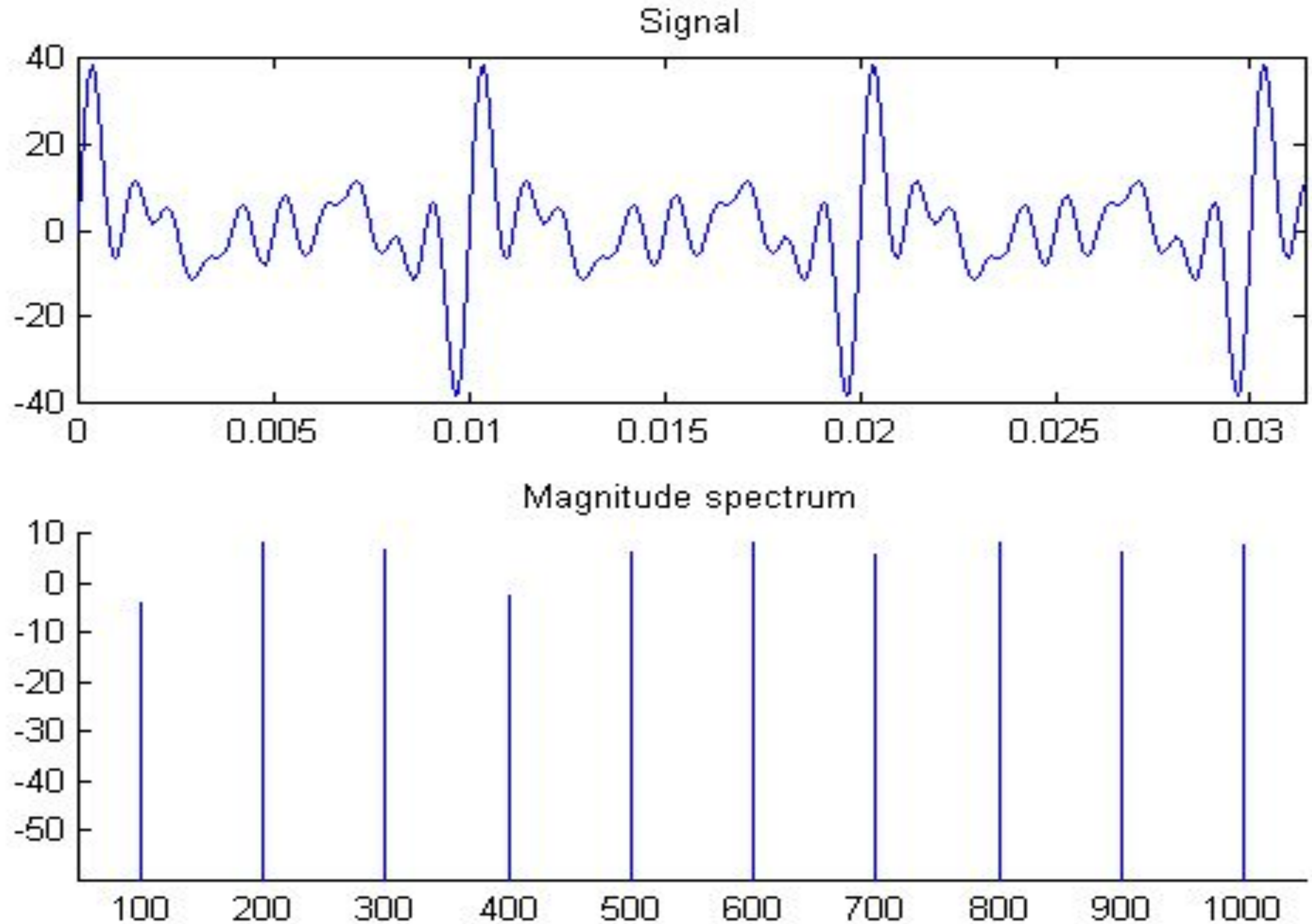


Спектр мощности





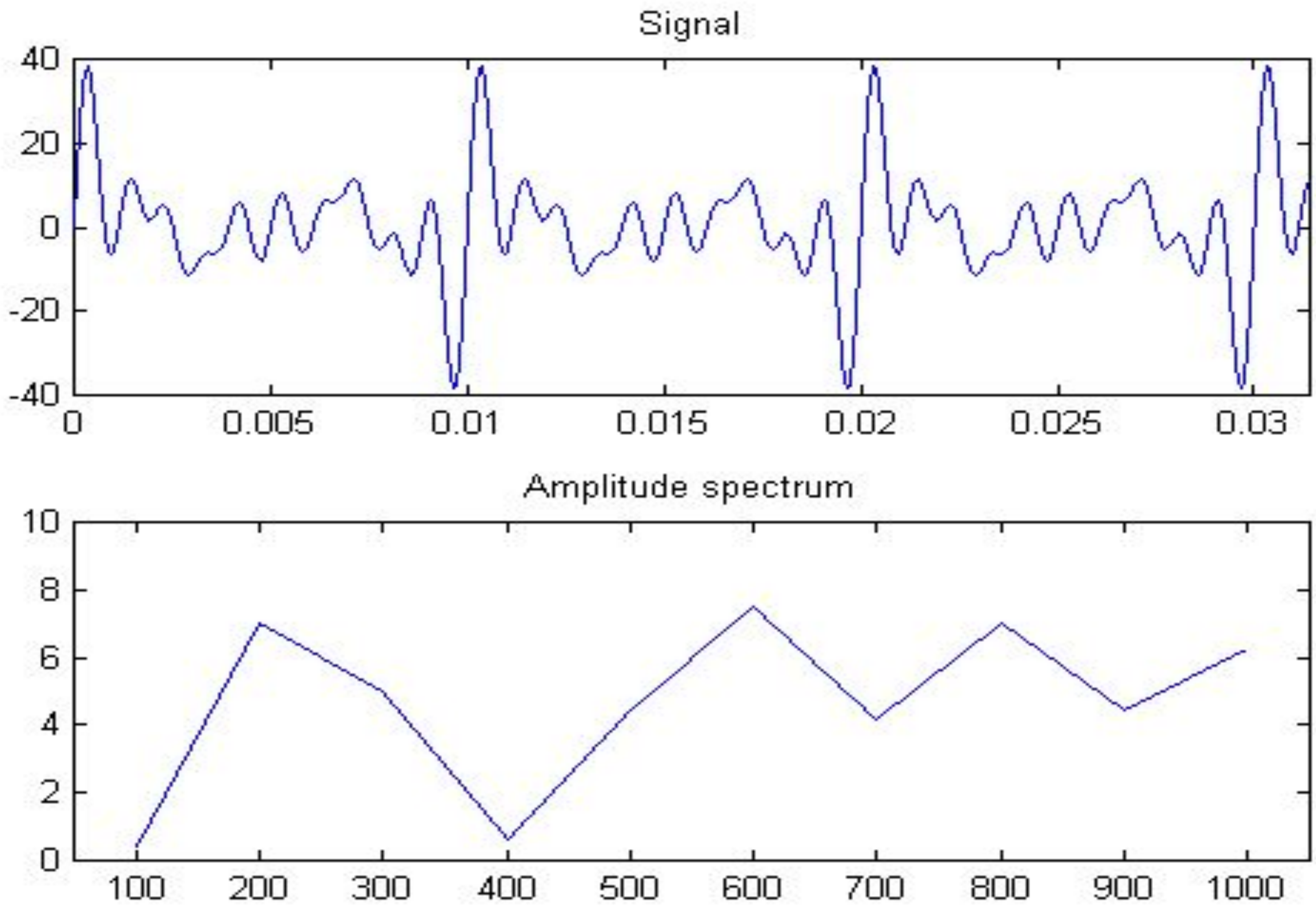
Логарифмический спектр



Перевод в децибеллы

- Имеем дискретный набор гармоник
- Для каждой гармоники считаем десятичный логарифм от амплитуды данной гармоники
- Умножаем результат на 10
- Получаем логарифмический спектр в децибеллах (дБ)

Огибающая спектра (spectral envelope)

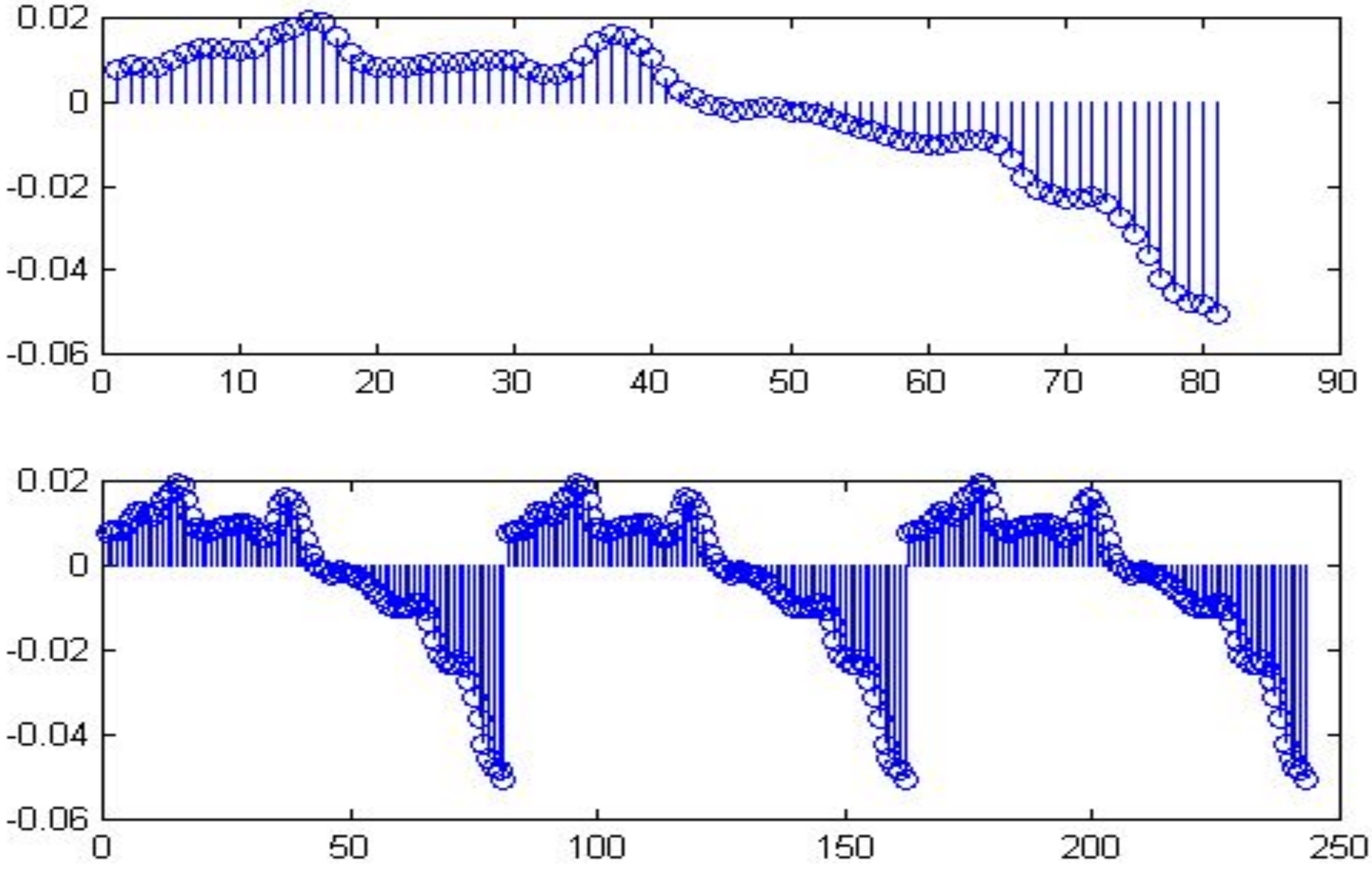


Как быть с фазой?

Периодическое продолжение

С точки зрения спектрального анализа дискретных сигналов, **ЛЮБОЙ** дискретный сигнал считается **периодически продолженным**

Пример – исходный и периодически продолженный сигналы



Периодическое продолжение

- Любой сигнал (вне зависимости от того, является ли он физически периодически или нет) рассматривается как **периодически продолженный (= периодический)**
- Для БПФ и участок гласного, и участок фрикативного будут равно периодическими

Теорема Фурье

- Раз любой дискретный сигнал рассматривается как периодический (с периодом ***T***, равным длительности сигнала), то к нему можно применить теорему Фурье
- Следовательно, любой дискретный сигнал может быть представлен как сумма гармоник с частотами ***(1/T), (2/T), (3/T), (4/T)*** и т.д.

Пример

- Пусть длительность ***T*** анализируемого сигнала = 20 миллисекунд (0.02 секунд). Тогда сигнал может быть представлен в виде суммы гармоник с частотами 50 Гц (1 / 0.02), 100 Гц (2 / 0.02), и т.д.

- Для данного сигнала частота 50 Гц **никакого отношения** не имеет к частоте колебаний голосовых складок.

Дискретное преобразование Фурье

- **Дискретное преобразование Фурье (ДПФ) (Discrete Fourier Transform, DFT)** – результат применения теоремы Фурье к дискретному сигналу
- ДПФ позволяет вычислить спектр сигнала по самому сигналу
- **Обратное дискретное преобразование Фурье (ОДПФ) (Inverse Discrete Fourier Transform, IDFT)** позволяет вычислить сигнал по его спектру

Свойства ДПФ

Свойство 1

- Если длина сигнала в отсчетах = ***N***, то количество гармоник в Фурье-разложении также будет ***N*** (а не бесконечное число, как для непрерывных сигналов)

- Соответствующий спектр Фурье также будет иметь ***N*** спектральных линий

Пример

- Пусть частота дискретизации сигнала 16 кГц, длительность сигнала в отсчетах = 160 отсчетов (10 миллисекунд). Тогда общее количество гармоник ДПФ-разложения = 160
- Частота самой нижней гармоники будет равна 1 / 0.01 = 100 Гц
- Частота самой высокой гармоники будет равна 160 / 0.01 = 16 кГц
- Разрешение между соседними гармониками по частоте = разности между частотами соседних гармоник = 100 Гц

Свойство 2

- Если частота дискретизации сигнала = ***Fs***, то частота самой высокой гармоники в ДПФ-разложении равна частоте дискретизации ***Fs***

- Если длительность сигнала (в секундах) = ***T*** , то разрешение по частоте равно 1/***T***

Скорость вычисления спектра

- Если длина сигнала в отсчетах = $\boldsymbol{N}$, то общее количество операций, необходимых для вычисления спектра, примерно равно N^2
- Например, если длина сигнала = 256 отсчетов, для вычисления спектра необходимо совершить 65536 операций
- Нельзя ли сократить число операций?

Быстрое преобразование Фурье

- **Быстрое преобразование Фурье (БПФ) (Fast Fourier Transform, FFT)** – способ «быстрого» вычисления ДПФ за счет одного математического трюка
- **Обратное быстрое преобразование Фурье (ОБПФ) (Inverse Fast Fourier Transform, IFFT) -** способ «быстрого» вычисления ОДПФ за счет одного математического трюка
- Общее количество операций в БПФ – примерно $N \log_2 N$
- Например, для 256 отсчетов имеем количество операций 2048 операций (вместо 65536 для ДПФ)

В чем трюк?

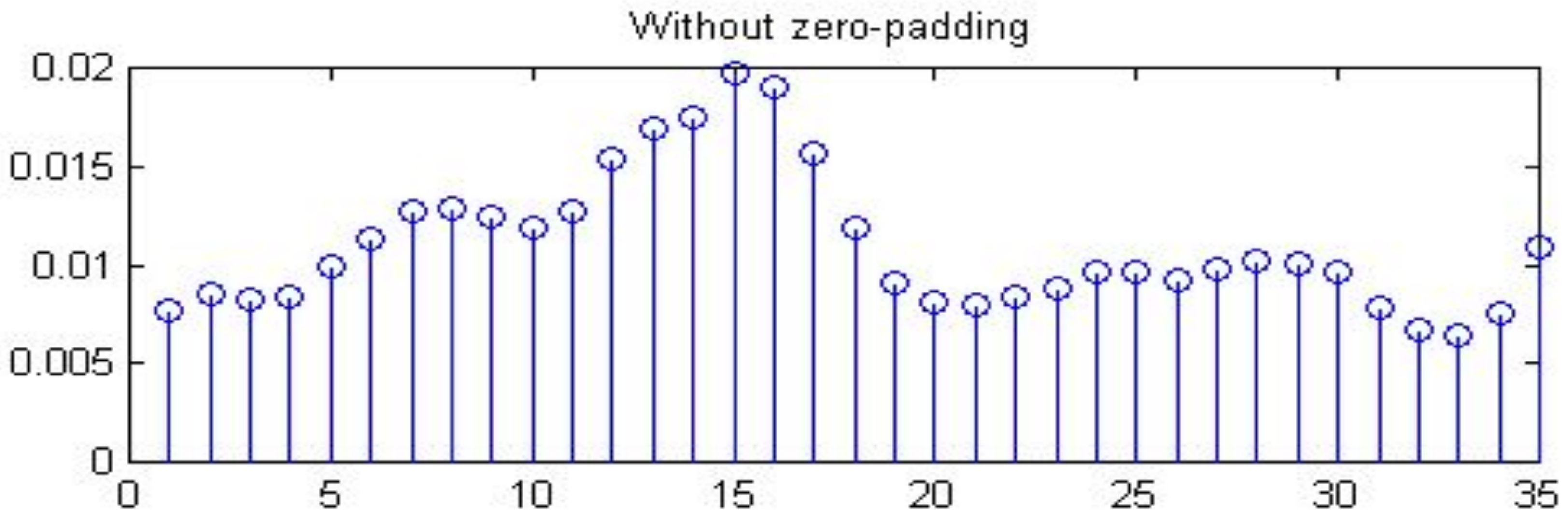
Если длина сигнала в отсчетах есть степень двойки (например, 256 отсчетов = 2^8, 512 отсчетов = 2^9), то количество операций можно существенно сократить

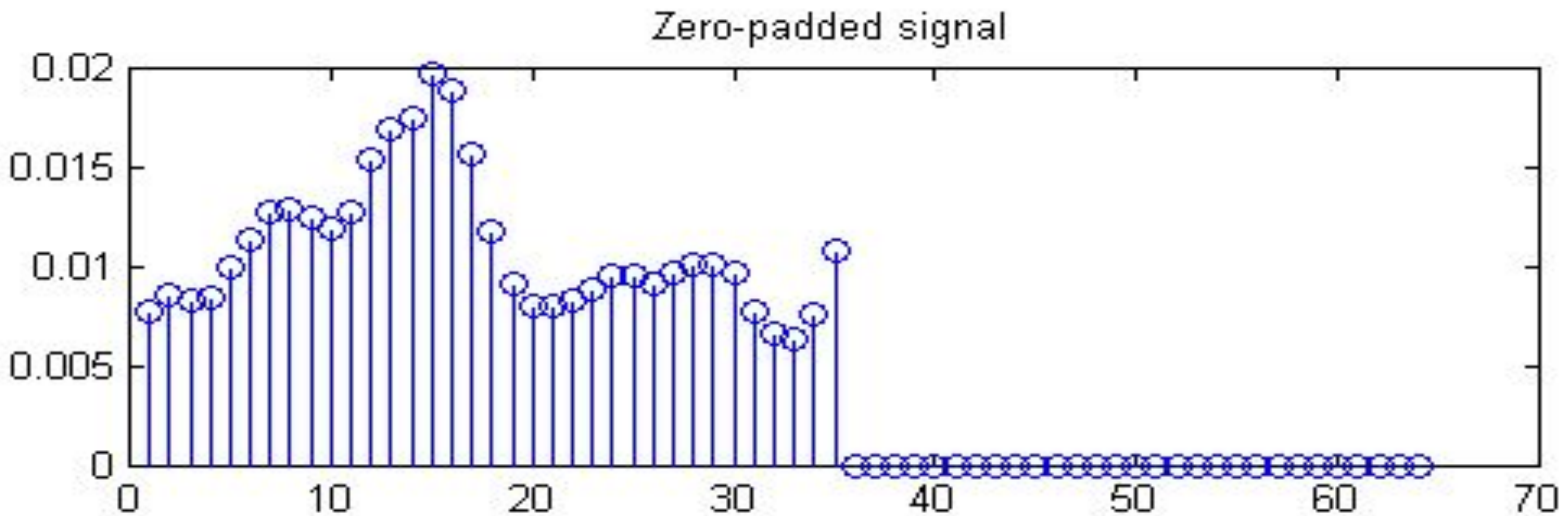
БПФ

- Таким образом, для эффективного использования БПФ длина сигнала в отсчетах должна быть **64** или **128** или **256** или **512** или **1024** или **2048** и т.д.

- Как этого добиться в действительности?

Дополнение нулями (zero-padding)

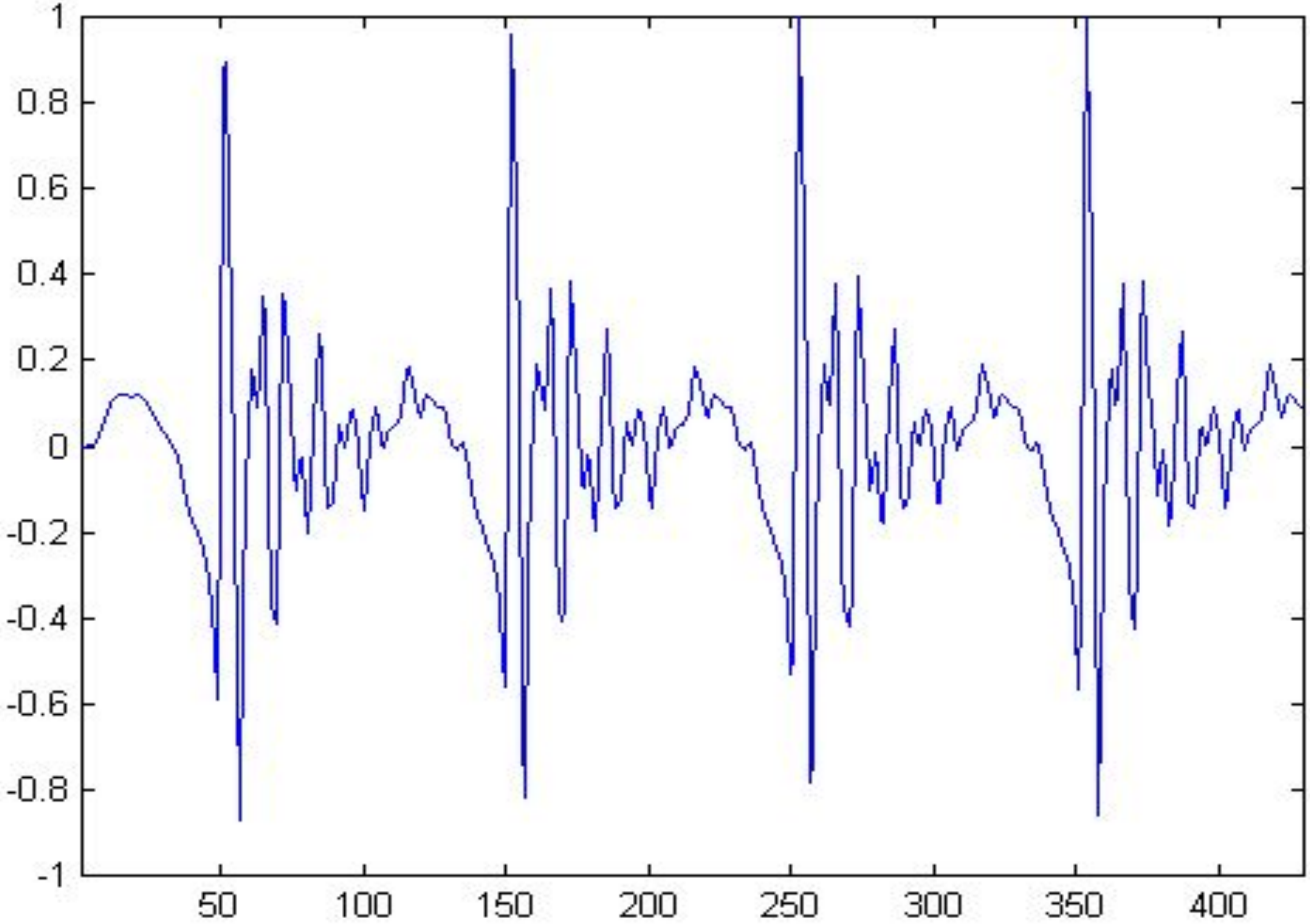




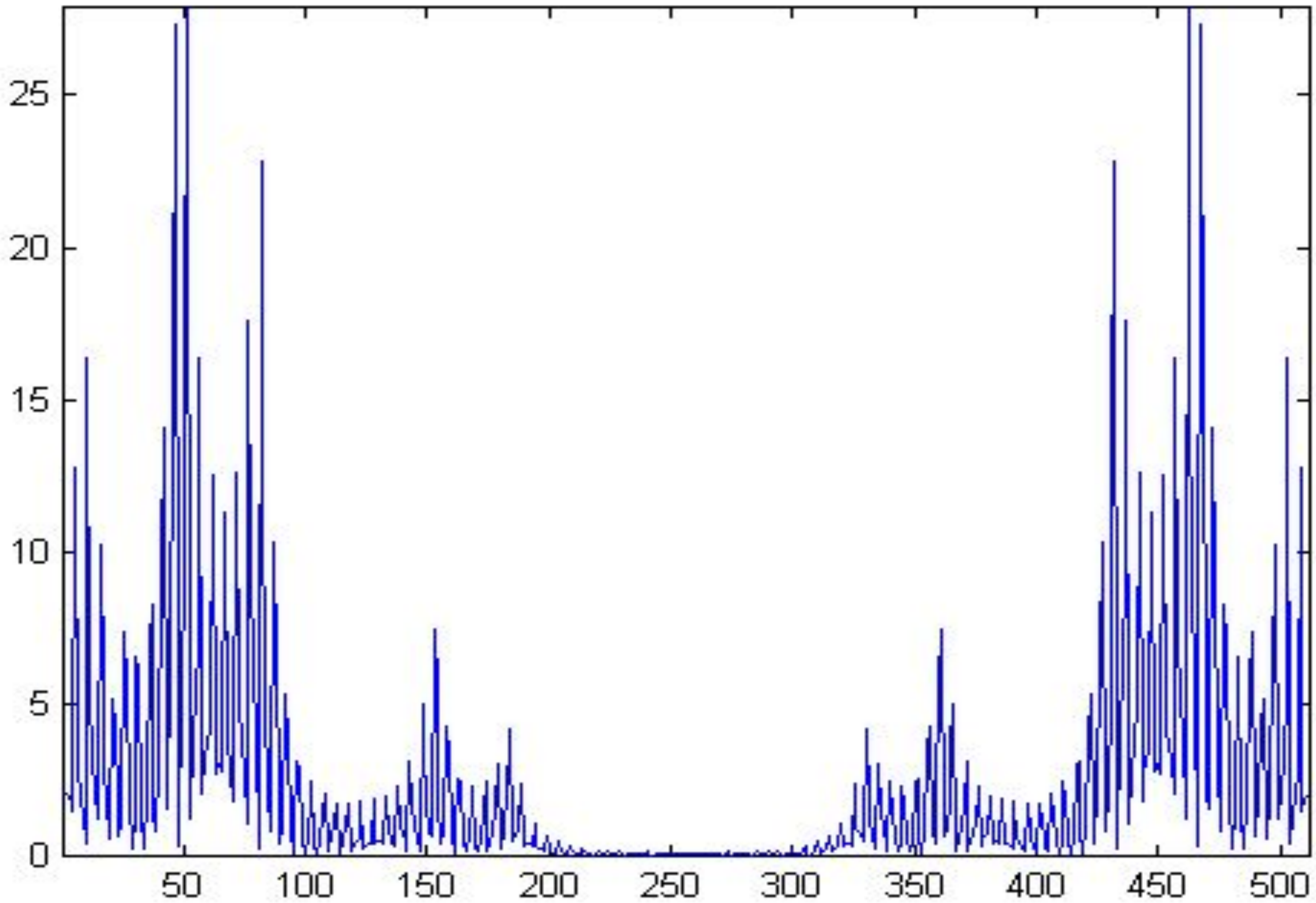
MATLAB

- ***Y = fft(x)*** - без дополнения нулями (может вычислять ОЧЕНЬ медленно, если длина сигнала ***x*** в отсчетах не равна степени двойки)
- ***Y = fft(x, N)*** – с дополнением нулями до ***N*** (где ***N*** – число, равное степени двойки, и большее, чем исходная длина сигнала ***x*** в отсчетах)
- ***X = ifft(Y)*** – ОБПФ

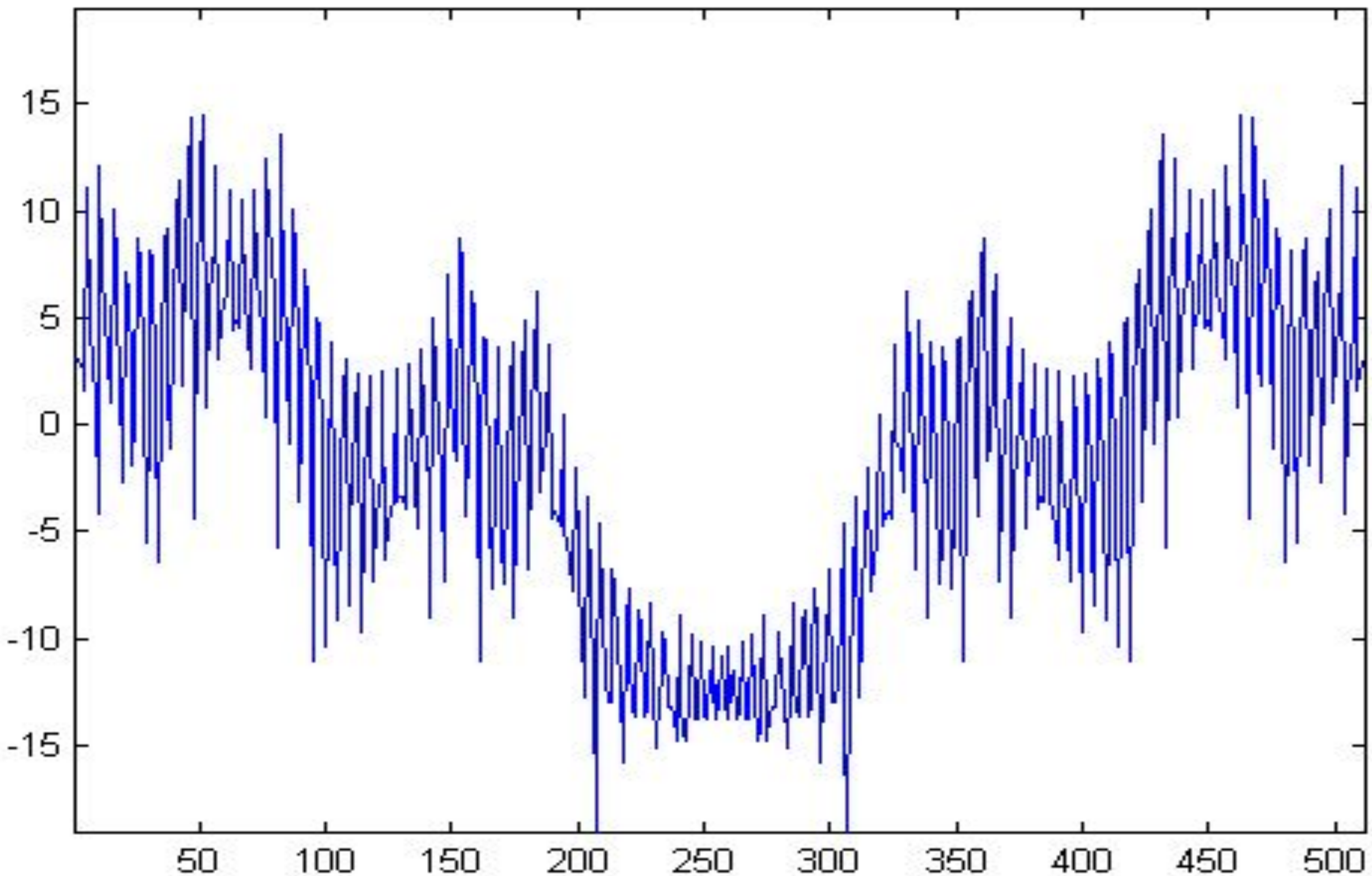
Пример



512-БПФ (амплитудный спектр)



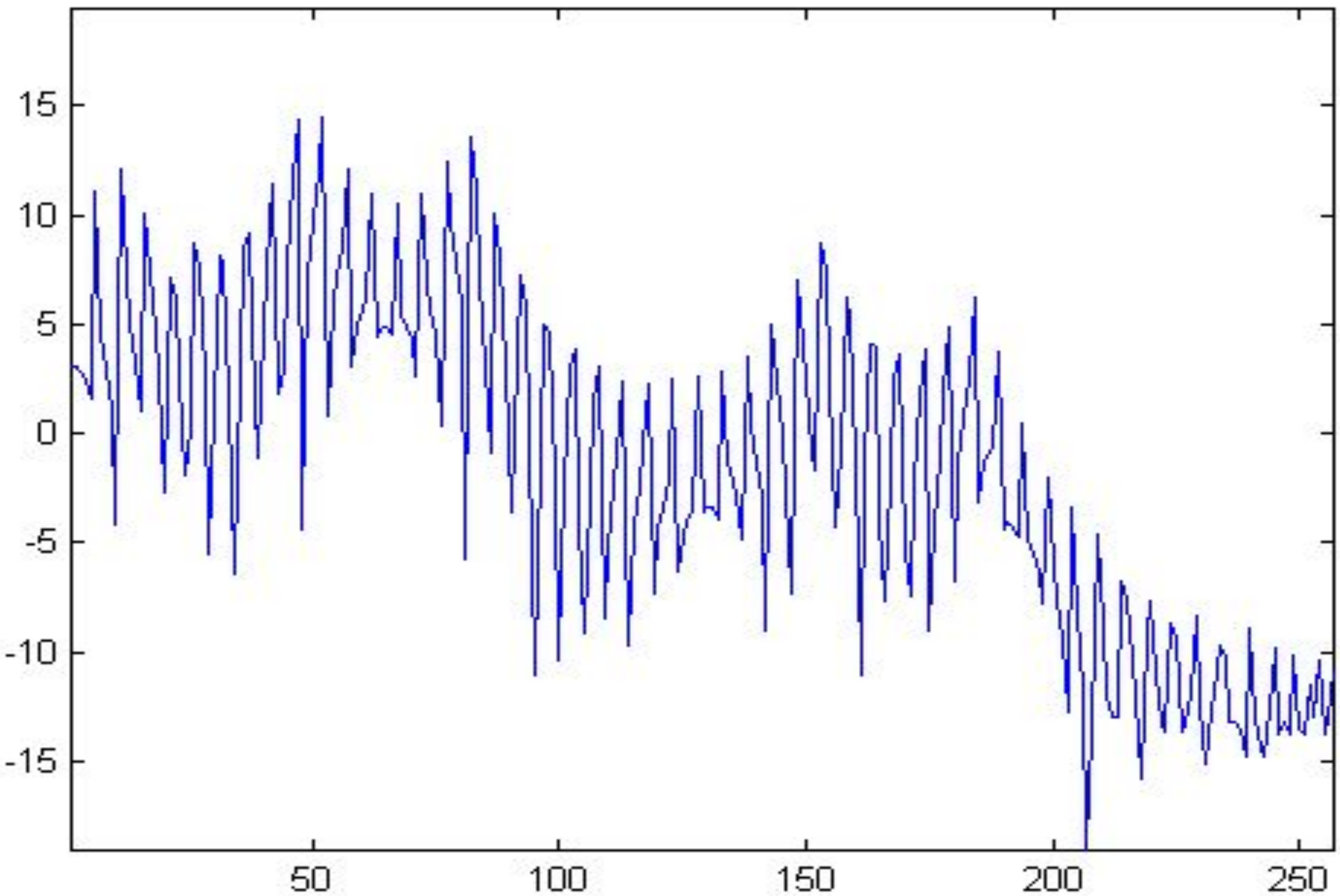
512-БПФ (логарифмический спектр)



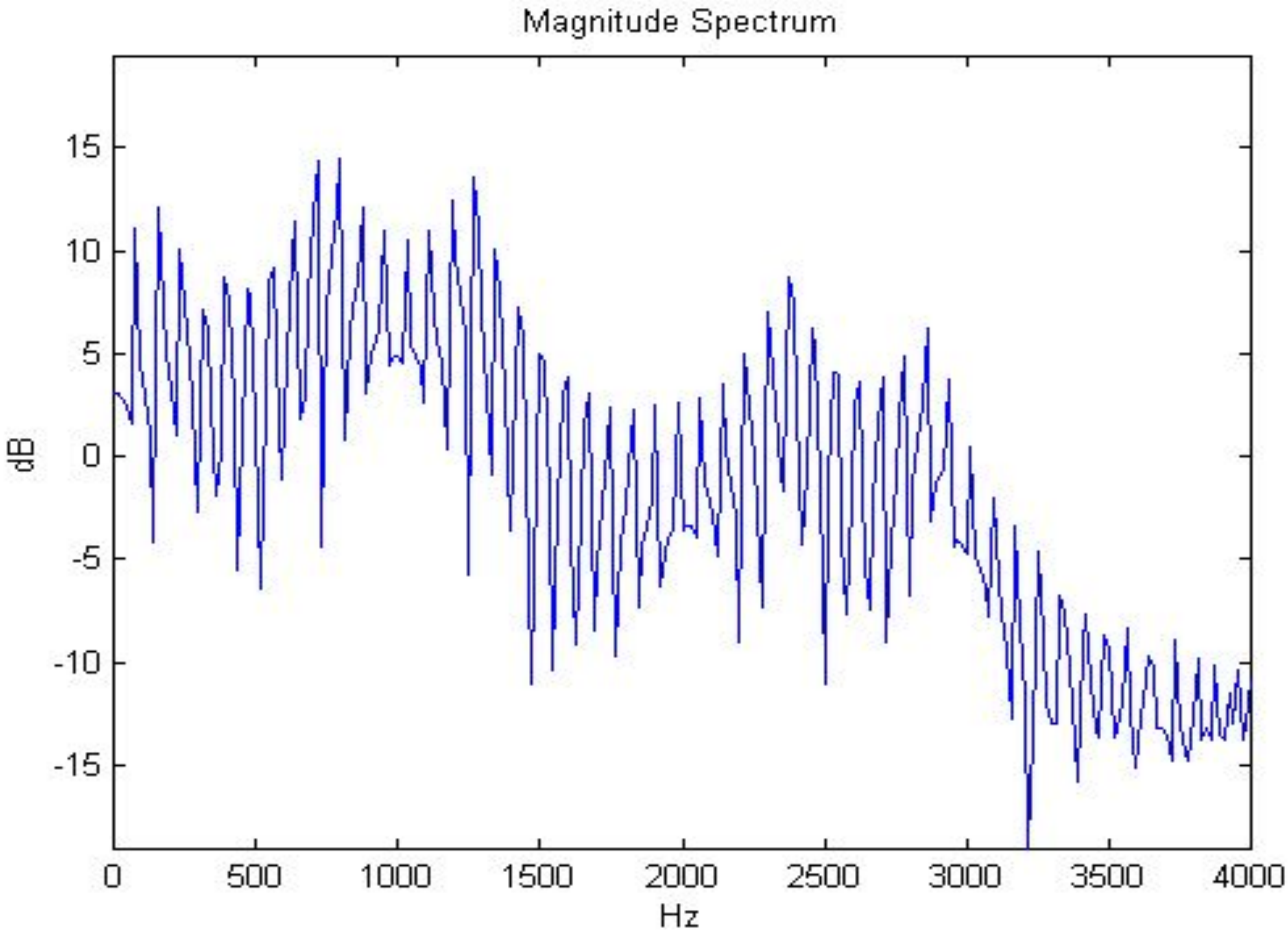
Свойство 3

- БПФ-спектр симметричен относительно срединной гармоники (например, 256-й гармоники для 512-точечного БПФ)
- Соответствующая частота = половине частоты дискретизации
- Например, для частоты дискретизации 16 кГц БПФ-спектр симметричен относительно частоты 8 кГц
- Необходимо вычислять спектр только до половины частоты дискретизации

512-БПФ, физический спектр

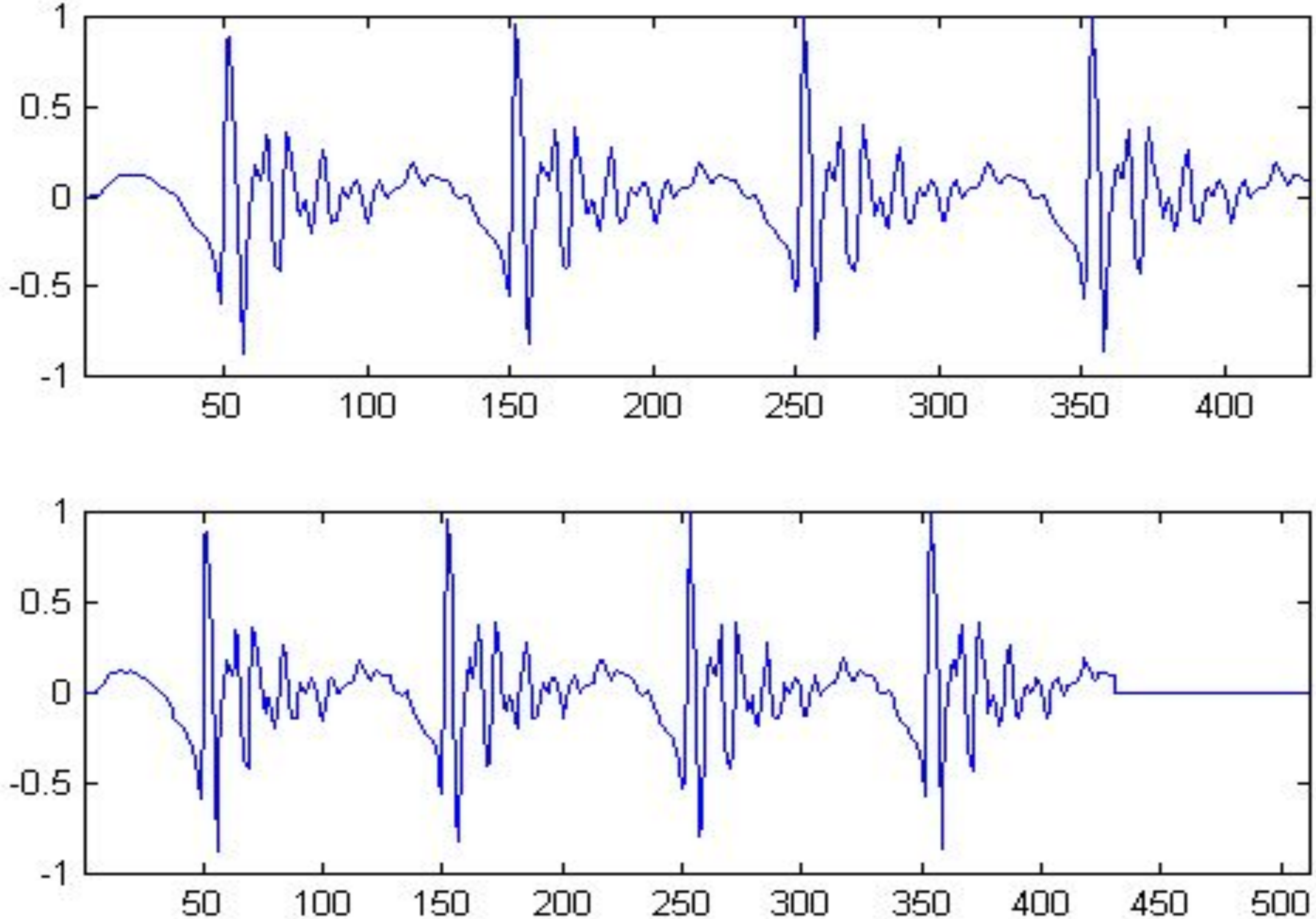


512-БПФ



Magnitude Spectrum
15
10
5
0
-5
-10
-15
dB
0
500
1000
1500
2000
2500
3000
3500
4000
Hz

ОБПФ


1
0.5
0
-0.5
-1
50
100
150
200
250
300
350
400
1
0.5
0
-0.5
-1
50
100
150
200
250
300
350
400
450
500

Что нужно помнить

- Если длина сигнала в отсчетах = ***N***, в секундах = ***T***, то сигнал можно представить суммой из ***N*** гармоник с частотами ***1/T***, ***2/T***, ***3/T***, …, ***N/T***
- БПФ-спектр нужно вычислять до гармоники с частотой ***N/(2T)***
- Если частота дискретизации сигнала = ***Fs***, то БПФ-спектр вычисляется до частоты ***Fs/2***
- Если *N* – не степень двойки, то необходимо дополнить нулями сигнал до ближайшего числа, являющегося степенью двойки (в **MATLAB** это делается автоматически)