

Лекция 4. БПФ и практический спектральный анализ

Курс: Обработка сигналов · 4 курс · 01.03.02 «Прикладная математика и информатика»
Направленность: искусственный интеллект и системная аналитика **Длительность:** 2 ак. ч лекции + 4 ак. ч практики

На лекции 3 мы получили **ДПФ** — точный спектр из N отсчётов, но за $O(N^2)$ операций. Сегодня закрываем два практических вопроса: **как считать быстро** — алгоритм **БПФ** — и **как считать правильно** — окна, разрешение, утечка, усреднение (метод Уэлча) и спектрограмма. На выходе — рабочий рецепт: как из сырых отсчётов получить спектр, которому можно верить.

К концу лекции вы сможете: объяснить идею БПФ «разделяй и властвуй» и «бабочку» radix-2; понять роль бит-реверса и откуда $O(N \log N)$; выбирать окно под задачу; объяснить $\Delta f = 1/T$ и почему zero-padding не улучшает разрешение; усреднять периодограммы и читать PSD; строить спектрограмму и считать всё это в Python.

1. Зачем БПФ: ещё раз о цене

Прямое ДПФ — умножение матрицы $N \times N$ на вектор, то есть $O(N^2)$. Для $N = 10^6$ это 10^{12} операций — неприемлемо. **БПФ** (Cooley–Tukey, 1965) считает **тот же** результат за $O(N \log_2 N)$: для 10^6 около $2 \cdot 10^7$, то есть в десятки тысяч раз быстрее.

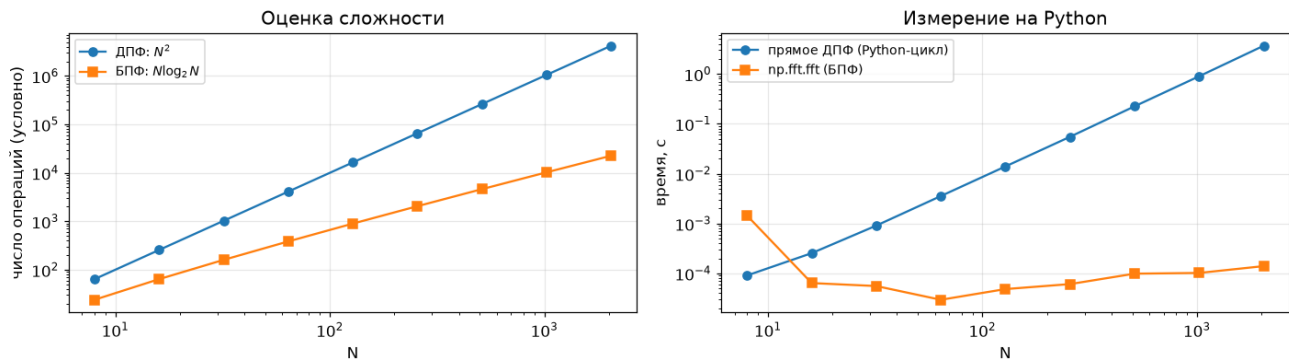


Figure 1: Стоимость ДПФ и БПФ

2. Идея БПФ: разделяй и властвуй

Пусть N чётно. Разделим сумму на **чётные** и **нечётные** отсчёты и учтём $W_N^{2kn} = W_{N/2}^{kn}$:

$$X[k] = E[k] + W_N^k \cdot O[k], \quad \begin{aligned} E[k] &= \sum_r x[2r] \cdot W_{N/2}^{kr}, \\ O[k] &= \sum_r x[2r+1] \cdot W_{N/2}^{kr}. \end{aligned}$$

ДПФ длины N свелось к **двум** ДПФ длины $N/2$, а рекурсия даёт $\log_2 N$ уровней — это **прореживание по времени** (decimation in time, DIT).

Бабочка radix-2

Два ДПФ половинной длины объединяются одной операцией — **бабочкой**:

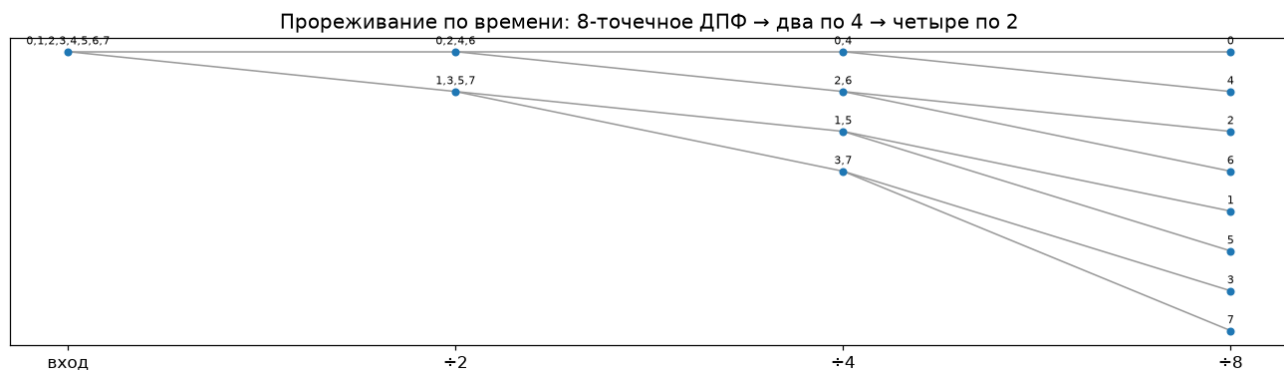


Figure 2: Дерево прореживания по времени

$$X[k] = E[k] + W_N^k \cdot O[k], \quad X[k + N/2] = E[k] - W_N^k \cdot O[k].$$

Одно комплексное умножение и два сложения на пару. Повторённые $\log_2 N$ раз, эти бабочки и дают выигрыш.

Бабочка radix-2 (ДПФ по основанию 2)

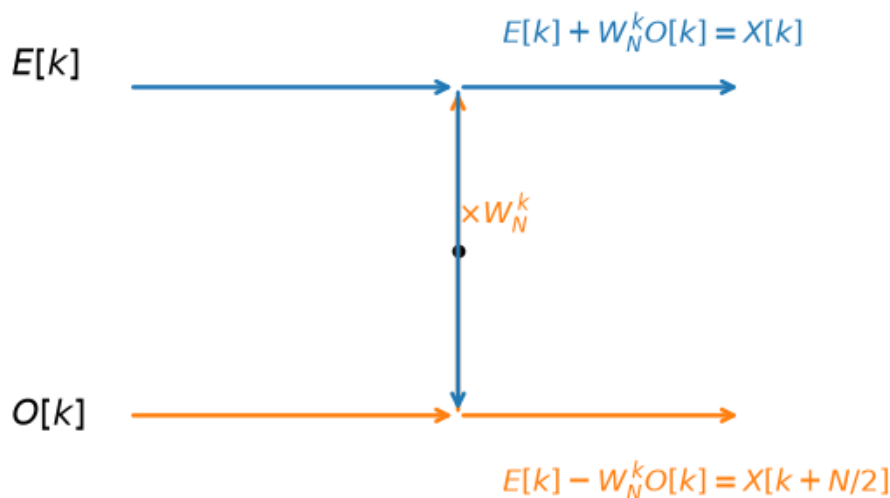


Figure 3: Бабочка radix-2

Граф и бит-реверс

Полный граф для $N = 8$ — три стадии по четыре бабочки. На **входе** отсчёты стоят в **бит-реверсивном** порядке (индекс читается справа налево: $1 = 001_2 \rightarrow 100_2 = 4$), на выходе — обычный порядок. Для $N = 8$: 0, 4, 2, 6, 1, 5, 3, 7.

Варианты и практика

radix-2 (проще всего, $N = 2^m$), radix-4, split-radix, mixed-radix — меньше операций. Простое большое N требует смешанных алгоритмов или округления до степени двойки. Для

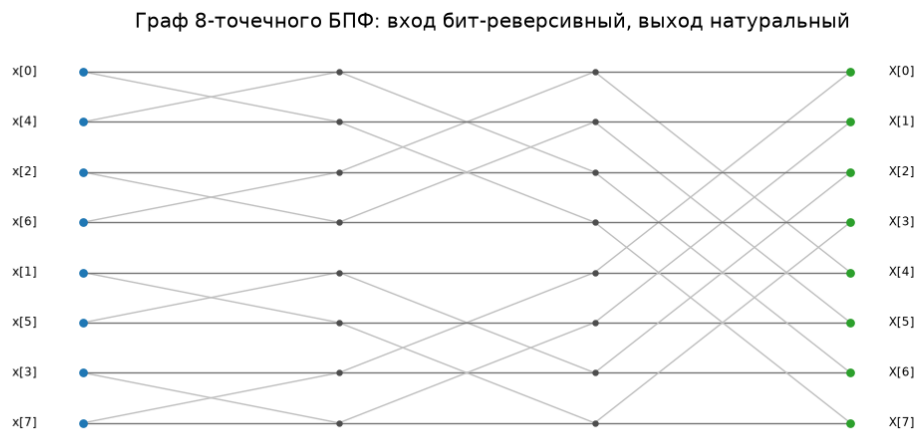


Figure 4: Граф 8-точечного БПФ

Бит-реверс индексов для N=8: 0,4,2,6,1,5,3,7

n	двоичный	реверс	новый индекс
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

Figure 5: Бит-реверс индексов

вещественного сигнала достаточно $N/2+1$ точек (rfft) — вдвое дешевле. Зрелые библиотеки: FFTW, pocketfft (NumPy/SciPy), Intel MKL, cuFFT (GPU).

3. Конечная запись: откуда утечка

Мы считаем ДПФ **конечного** куска: формально умножаем сигнал на **прямоугольное окно** шириной T . Умножение во времени — это **свёртка в спектре** с функцией окна (для прямоугольного — sinc). Отсюда **утечка**: даже чистый тон превращается в «холмик» с боковыми лепестками. **Окно** — плавное обрезание записи: оно снижает боковые лепестки, но **расширяет главный**.

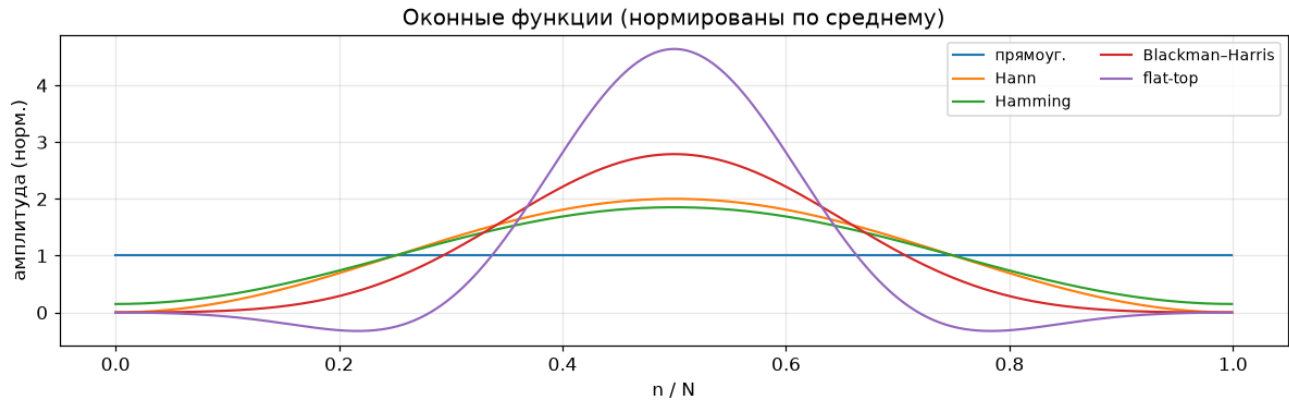


Figure 6: Оконные функции

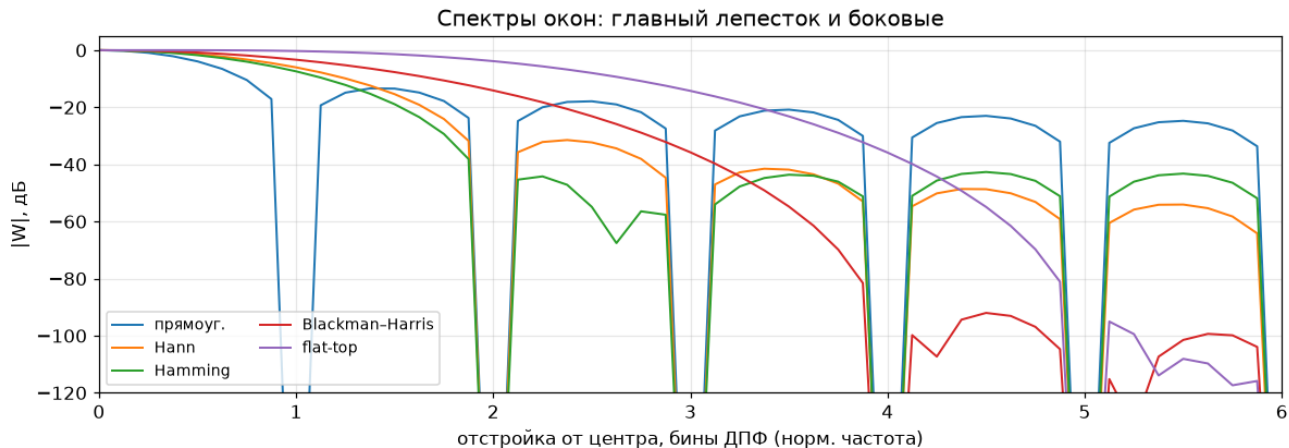


Figure 7: Спектры окон: главный лепесток и боковые

Компромисс. Узкий главный лепесток и низкие боковые одновременно невозможны. Прямоугольное: 2 бина, -13 дБ; Hann: 4 бина, -31 дБ; Hamming: 4 бина, -43 дБ; Blackman-Harris: 8 бинов, -92 дБ; flat-top: 10 бинов, зато точная амплитуда.

Паспорт окна. Кроме формы важны коэффициенты: **coherent gain** = $\bar{w}$ (нормировка амплитуды: Hann 0.5, Hamming 0.54, flat-top 0.22) и **ENBW** = $N \cdot \Sigma w^2 / (\Sigma w)^2$ (нормировка мощности/PSD: прямоуг. 1.0, Hann 1.5, flat-top 3.77). Амплитуду линии делят на coherent gain, шумовую полку — на ENBW. В `scipy.signal.welch` это учитывается автоматически.



Figure 8: Компромисс окна

4. Утечка и её подавление

Возьмём тон **ровно между бинами** и сравним прямоугольное окно и Hann: прямоугольное даёт медленно падающие боковые (≈ -13 дБ), Hann — быстро падающие, поэтому слабые линии рядом с сильной видны.

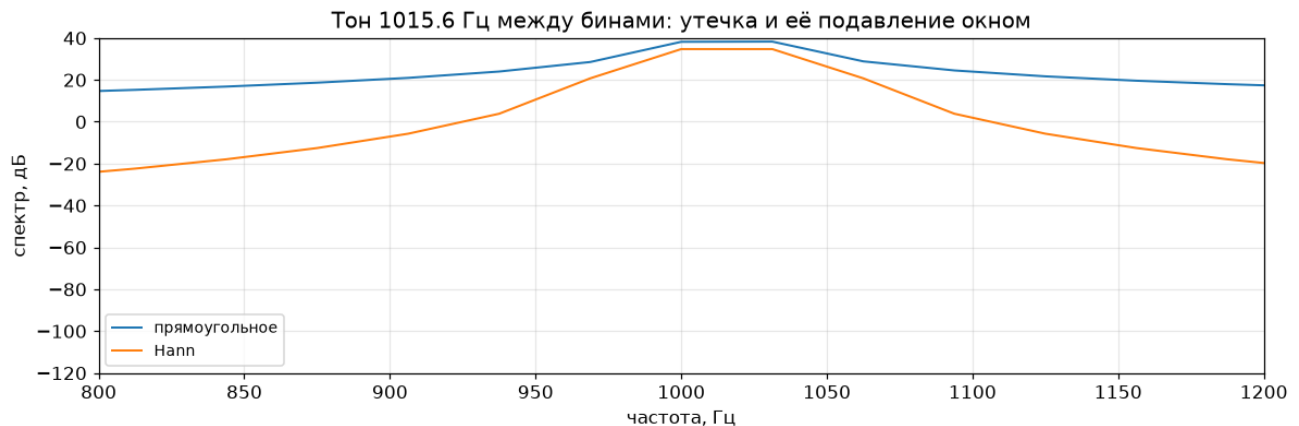


Figure 9: Утечка и подавление окном

5. Разрешение и zero-padding

Разрешение задаёт длительность записи, а не частота дискретизации: $\Delta f = f_d/N = 1/T$. Два тона различимы (критерий Рэлея), если расстояние $> \Delta f$; для Hann практического разрешения нужно $\approx 2\Delta f$.

Zero-padding даёт плотную сетку частот — это **интерполяция** той же спектральной функции

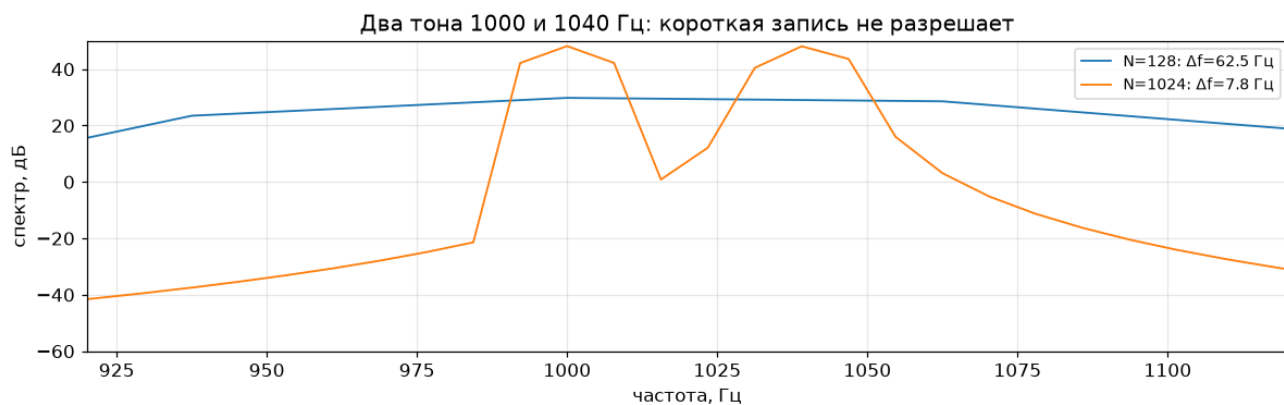


Figure 10: Разрешение: короткая запись не разрешает

между бинами. Разрешить близкие частоты это не помогает.

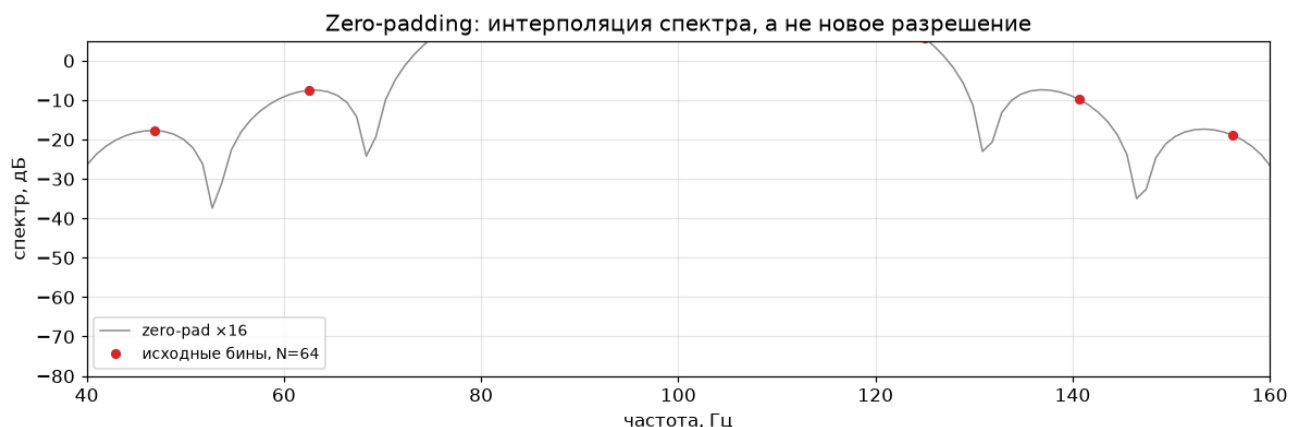


Figure 11: Zero-padding интерполирует, а не разрешает

6. Усреднение: метод Уэлча (Welch)

Одна периодограмма $P(f) = |X(f)|^2$ сильно **шумит**: её дисперсия не падает с ростом N . Идея Уэлча: разрезать запись на **сегменты** (можно с перекрытием), к каждому применить **окно**, посчитать $|БПФ|^2$ и **усреднить** по сегментам. Это несмещённая оценка с меньшей дисперсией: шум сглаживается, реальные линии остаются. Цена — разрешение по длине сегмента.

Что такое PSD

Спектральная плотность мощности — мощность на **герц**. `scaling='density'` — $\text{ед.}^2/\text{Гц}$ (для шумов), `scaling='spectrum'` — ед.^2 в бине (для линий). Площадь под PSD равна полной мощности. Окно нормирует PSD через ENBW: $\text{PSD} = |W|^2 / (f_d \cdot \text{ENBW})$ — в `welch` это automatic.

7. Время—частота: спектрограмма (STFT)

Если сигнал меняется во времени, одного спектра мало. **Короткое преобразование Фурье** считает спектр в скользящем окне; результат — **спектрограмма**: по X время, по Y частота, цвет —

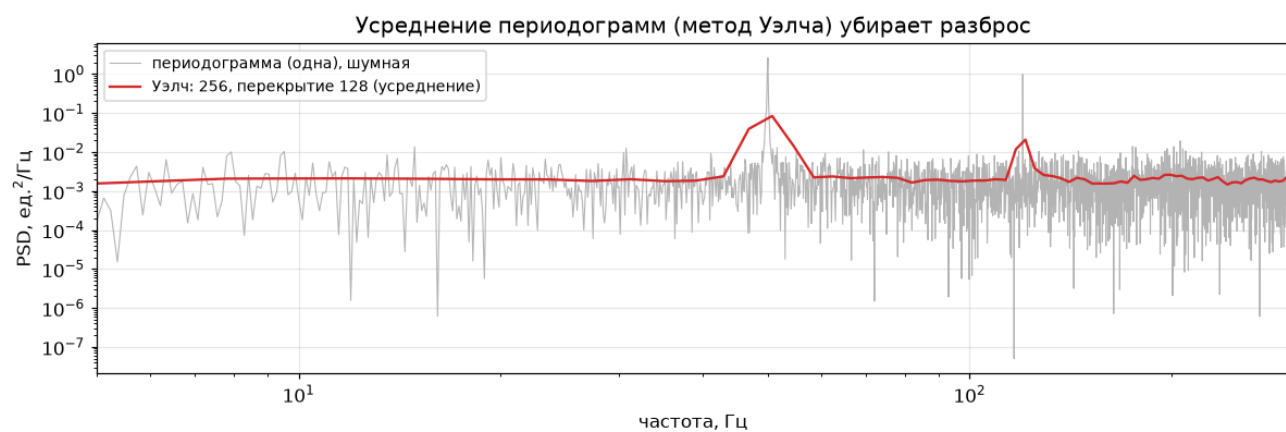


Figure 12: Метод Уэлча убирает разброс

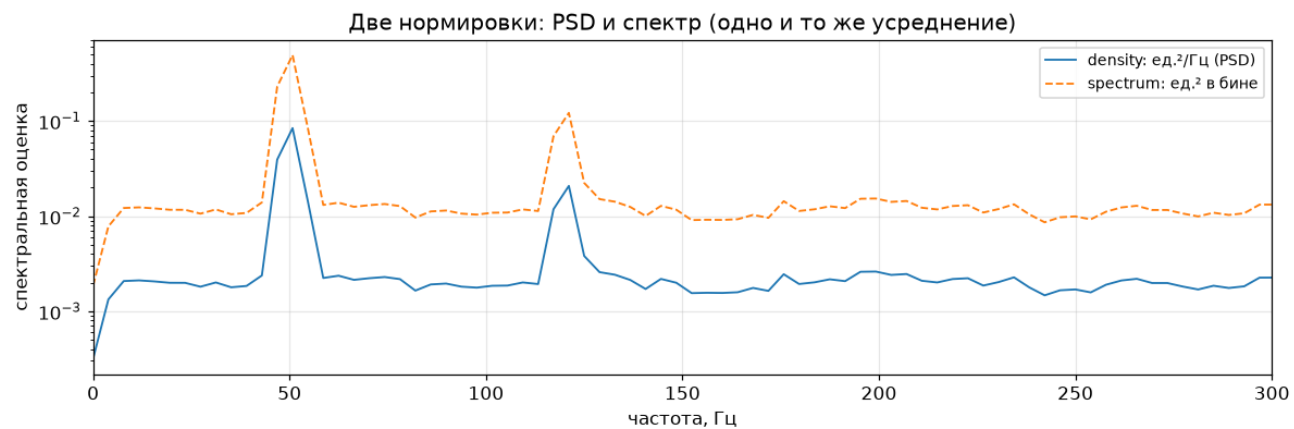


Figure 13: Нормировки PSD и спектра

мощность. Работает **принцип неопределённости**: короткое окно даёт хорошее **время** и плохую **частоту**, длинное — наоборот.

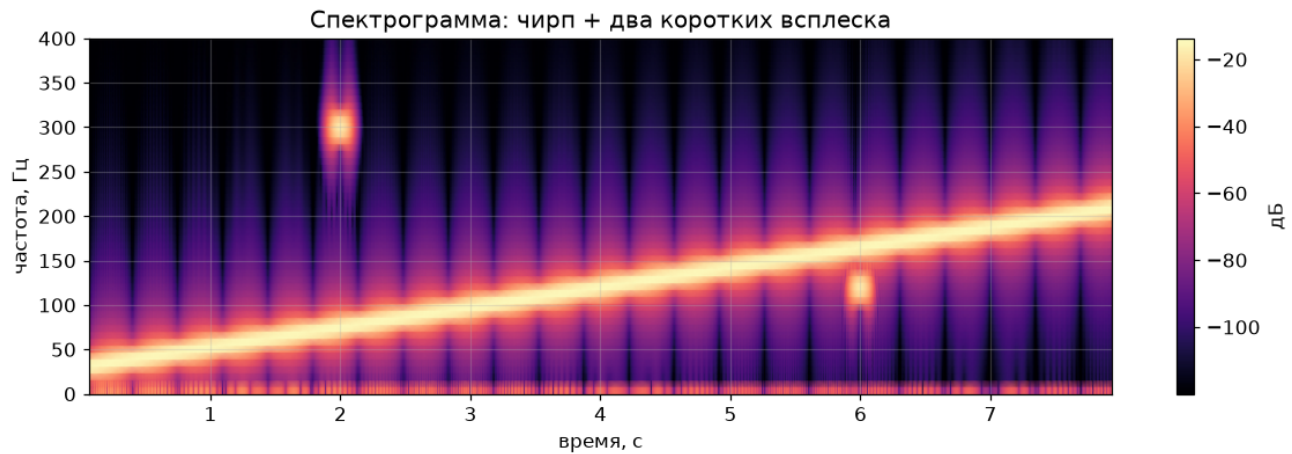


Figure 14: Спектрограмма: чирп и всплески

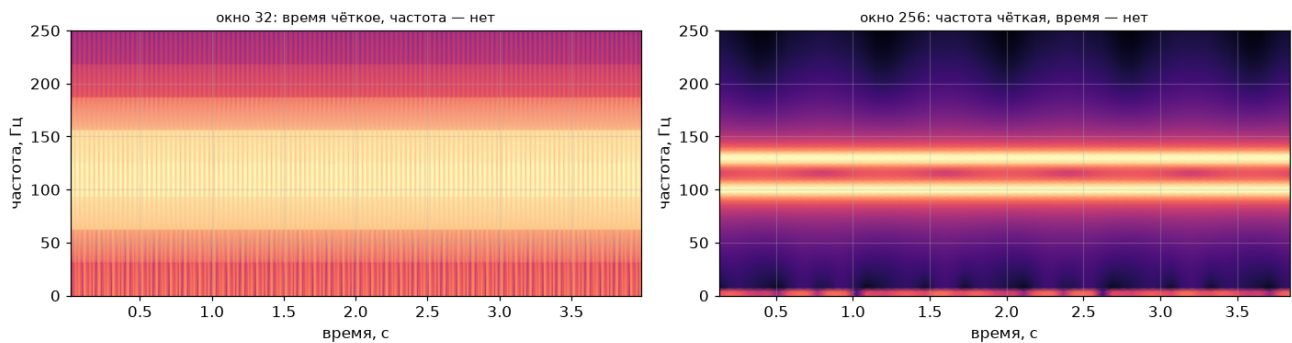


Figure 15: Компромисс время—частота

8. Всё вместе: спектр вибрации

База 50 Гц, дефект 120 Гц, слабая вторая гармоника 240 Гц и шум: одна периодограмма «утопает» в шуме, а **Уэлч** вытаскивает слабую линию 240 Гц.

Точная оценка частоты

Параболическая интерполяция $\log|X|$ по трём точкам вокруг пика: $\delta = \frac{1}{2} \cdot (y_{-1} - y_{+1}) / (y_{-1} - 2y_0 + y_{+1})$, $f_0 \approx (k + \delta) \cdot \Delta f$. Для вибрации и радиосигналов этого обычно достаточно.

9. Рецепт на Python (`scipy.signal`)

```
from scipy import signal
```

```
w = signal.get_window("hann", N)
X = np.fft.rfft(x * w)
```

```
# 'hamming', 'blackmanharris', ...
```

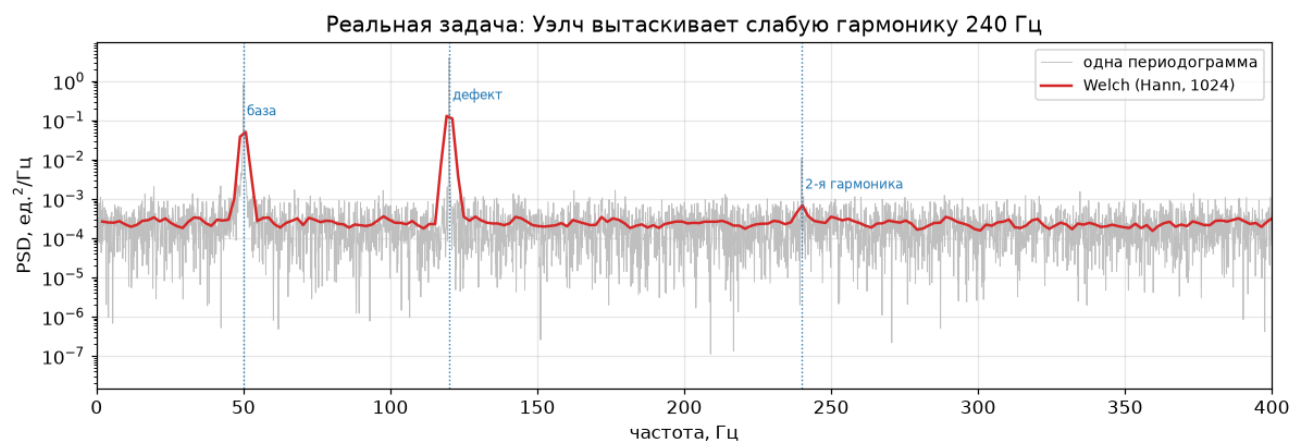


Figure 16: Реальная задача: Уэлч вытаскивает слабую гармонику

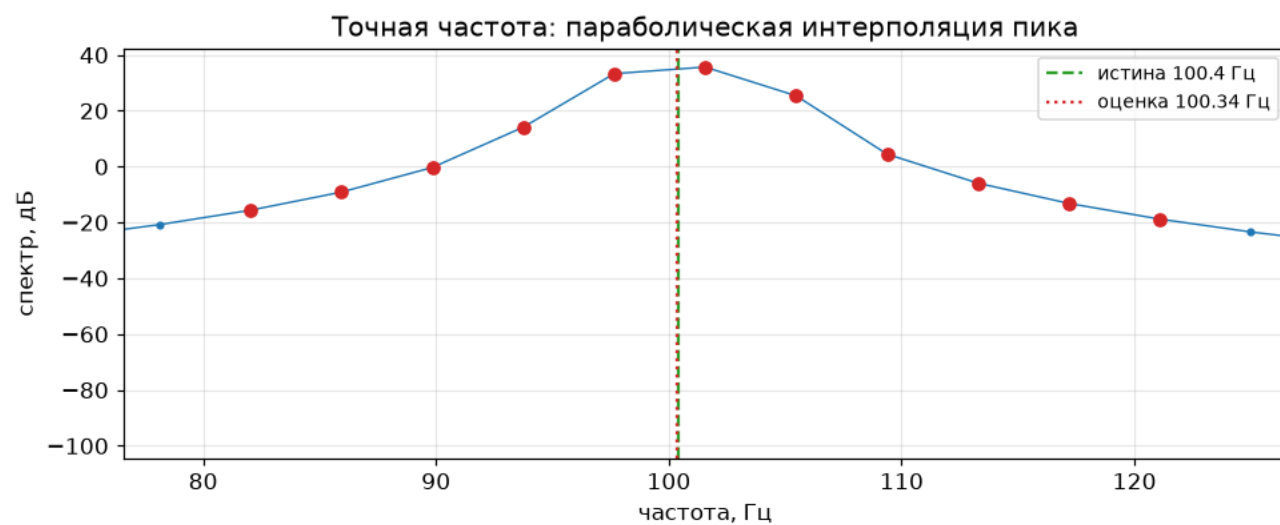


Figure 17: Точная оценка частоты

```
f = np.fft.rfftfreq(N, 1 / fs)
```

```
f, Pxx = signal.welch(x, fs, window="hann", nperseg=1024,  
                      noverlap=512, scaling="density")  
f, t, S = signal.spectrogram(x, fs, window="hann", nperseg=128,  
                              noverlap=96, scaling="density")
```

Порядок действий: выбрать T под нужное $\Delta f \rightarrow$ выбрать окно $\rightarrow$ взять N как степень двойки (быстрее) $\rightarrow$ посчитать rfft/welch $\rightarrow$ перевести в дБ для глаз.

Сквозной проект: задание №4

1. Реализовать **свою** функцию БПФ radix-2 (рекурсивную) и сверить с `numpy.fft`.
2. Показать **утечку** и её подавление окном (прямоуг. против Hann/Hamming).
3. Сравнить **разрешение** при двух длительностях; проверить, что zero-padding разрешение не улучшает.
4. Оценить **PSD** методом **Уэлча** и объяснить выбор `nperseg/noverlap`.
5. Построить **спектрограмму** нестационарного участка своего сигнала.

Сдавать: ноутбук с графиками + **один** слайд-вывод (какое окно выбрали и почему).

Итоги

- **БПФ** — не новое преобразование, а быстрый алгоритм ДПФ: $O(N^2) \rightarrow O(N \log N)$.
- Идея: прореживание по времени, **бабочки** radix-2, **бит-реверс** на входе.
- Конечная запись = прямоугольное окно $\rightarrow$ **утечка**; окна лечат, но расширяют главный лепесток.
- Разрешение $\Delta f = 1/T$; **zero-padding** только интерполирует.
- **Уэлч**: сегменты + окно + усреднение $| \text{БПФ} |^2 \rightarrow$ устойчивая **PSD**.
- **Спектрограмма** показывает эволюцию спектра; компромисс время–частота.

Дальше (лекция 5): спектры радио и модуляция — АМ/ЧМ/ФМ, цифровые созвездия, комплексная огибающая (I/Q).

Литература

1. Оппенгейм А., Шафер Р. *Цифровая обработка сигналов*. — М.: Техносфера, 2012.
2. Сергиенко А. Б. *Цифровая обработка сигналов*. — СПб.: БХВ-Петербург, 2011.
3. Солонина А. И. и др. *Основы цифровой обработки сигналов*. — СПб.: БХВ-Петербург, 2005.
4. Умняшкин С. В. *Основы теории цифровой обработки сигналов*.
5. Harris F. J. *On the Use of Windows for Harmonic Analysis with the DFT*.
6. Welch P. D. *The Use of the Fast Fourier Transform for Estimation of Power Spectra*.
7. Материалы курса и RAG-ассистент по этим книгам — в папке курса.