Лекция №3: Свёртка и корреляционный анализ сигналов

1. Введение: зачем нужна свёртка?

Свёртка – одна из фундаментальных операций в обработке сигналов. Она описывает, как линейная система преобразует входной сигнал.

Линейной называют систему, отклик которой на суммарное воздействие равняется сумме откликов на каждое воздействие в отдельности.

Любую линейную **стационарную** (то есть такую, свойства которой не меняются во времени) систему можно полностью охарактеризовать её **импульсной характеристикой** $h[n]$ (ответом на дельта-образное воздействие $\delta[n]$), и тогда выход системы на произвольный вход $x[n]$ вычисляется как свёртка $y = x * h$.

Почему это важно?

- Позволяет моделировать физические системы (фильтры, каналы связи, акустику).
- Лежит в основе цифровой фильтрации, свёрточных нейронных сетей, обработки изображений.
- Через теорему о свёртке связывает временную и частотную области, давая мощные вычислительные методы (быстрая свёртка через БПФ).

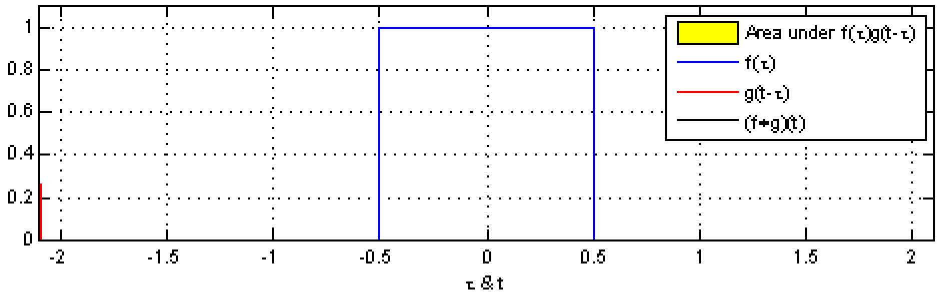
2. Определение свёртки и её свойства

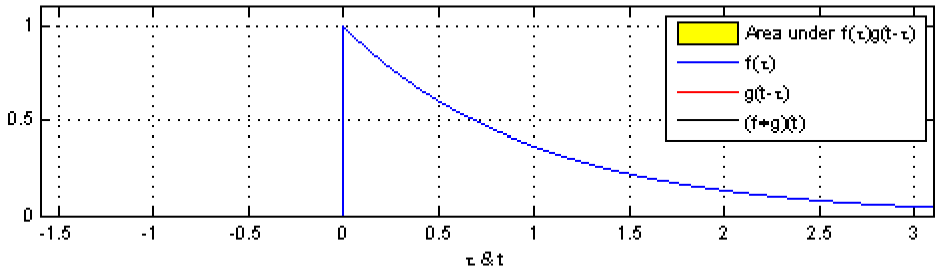
2.1. Формула дискретной свёртки

Для дискретных сигналов $x[n]$ и $h[n]$ линейная **свёртка** определяется как:

$$y[n] = (x * h)[n] = \sum_{k=-\infty}^{\infty} x[k]\, h[n-k]$$

Физический смысл: каждый отсчёт $y[n]$ – это сумма произведений всех отсчётов $x[k]$ на веса $h[n-k]$, то есть импульсная характеристика «скользит» вдоль входного сигнала.





2.2. Свойства свёртки

- **Коммутативность:**

$$x * h = h * x$$

- **Ассоциативность:**

$$x * (h_1 * h_2) = (x * h_1) * h_2$$

- **Дистрибутивность:**

$$x * (h_1 + h_2) = x * h_1 + x * h_2$$

- **Сдвиг:**

$$y[n] = x[n] * h[n] \to x[n - n_0] * h[n] = y[n - n_0]$$

2.3. Теорема о свёртке

Теорема: Дискретное преобразование Фурье (ДПФ) свёртки двух последовательностей равно поэлементному произведению их ДПФ.

$$\mathcal{F}\{x * h\}[k] = X[k] \cdot H[k]$$

Для линейной свёртки нужно дополнять последовательности нулями до длины $N \geq N_x + N_h - 1$, чтобы избежать эффекта циклического наложения.

3. Линейная и циклическая свёртка. Краевые эффекты

Линейная свёртка – результат для непереодических сигналов, длина $N_x + N_h - 1$.

Циклическая свёртка – используется для периодических сигналов или в алгоритмах БПФ. При длине N результат также имеет длину N. Вычисляется как

$$y[n] = \sum_{m=0}^{N-1} x[m]h[(n-m) \bmod N]$$

Краевые эффекты: если вычислять линейную свёртку через БПФ без дополнения нулями, получается циклическая свёртка, которая искажает края из-за замыкания последовательности в кольцо.

4. Корреляционный анализ

4.1. Определения

Кросс-корреляция:

$$R_{xy}[m] = \sum_{n=-\infty}^{\infty} x[n]\, y[n+m]$$

Автокорреляция:

$$R_{xx}[m] = \sum_{n=-\infty}^{\infty} x[n]\, x[n+m]$$

Связь со свёрткой: $R_{xy}[m] = x[-m] * y[m]$.

4.2. Свойства корреляционных функций

- **Чётность автокорреляции:** $R_{xx}[m] = R_{xx}[-m]$ (симметрия относительно нуля).
- **Максимум при нулевом сдвиге:** $|R_{xx}[m]| \leq R_{xx}[0]$
- Значение при нулевом сдвиге имеет смысл энергии сигнала (интеграла от его квадрата):

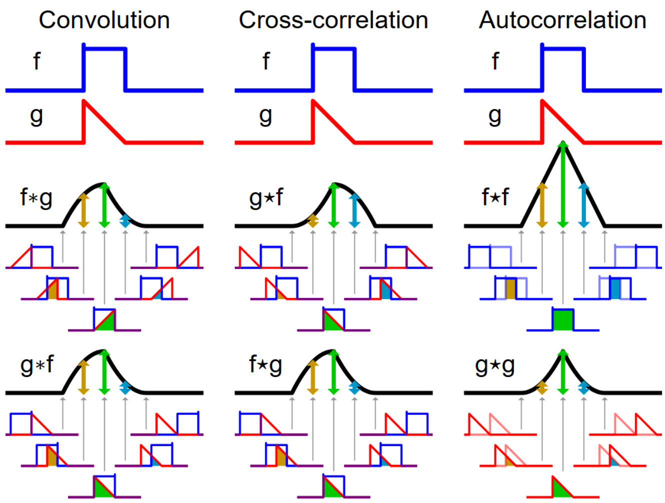
$$R_{xx}[0] = \sum_{n=-\infty}^{\infty} x^2[n]$$

- Для периодического сигнала автокорреляция имеет ту же периодичность:

$$x[n] = x[n+p] \to R_{xx}[m] = R_{xx}[m+p]$$

- **Кросс-корреляция не коммутативна:** $R_{xy}[m] \neq R_{yx}[m]$; вместо этого $R_{xy}[m] = R_{yx}[-m]$.

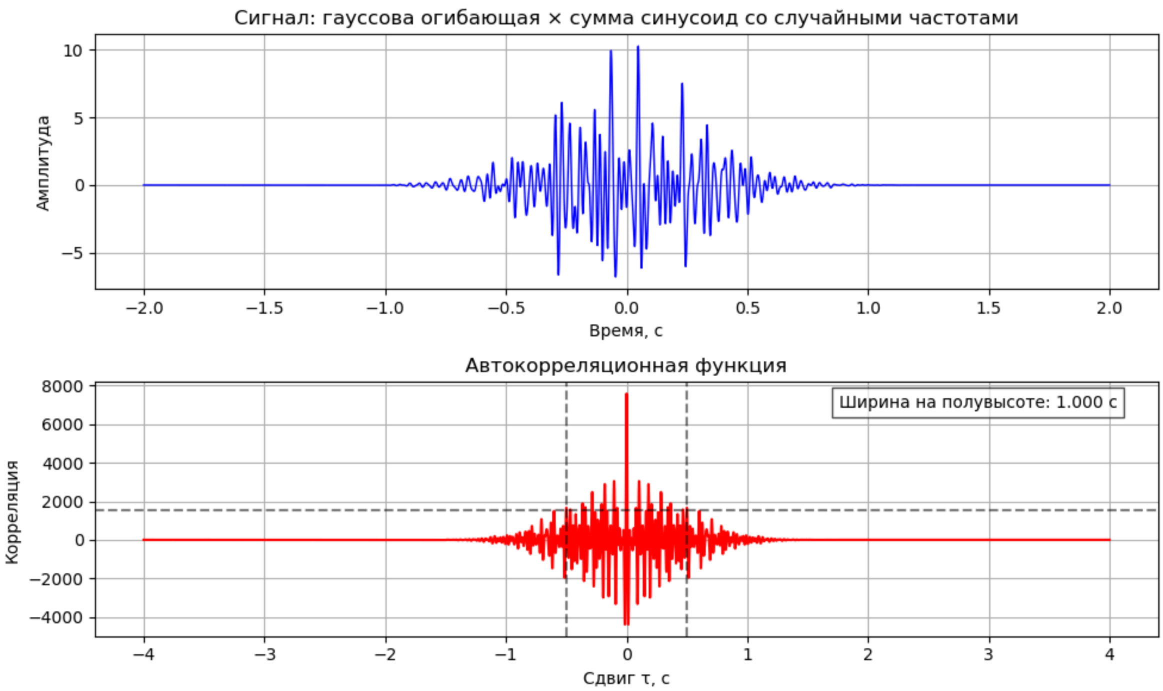
4.3. Некоммутативность



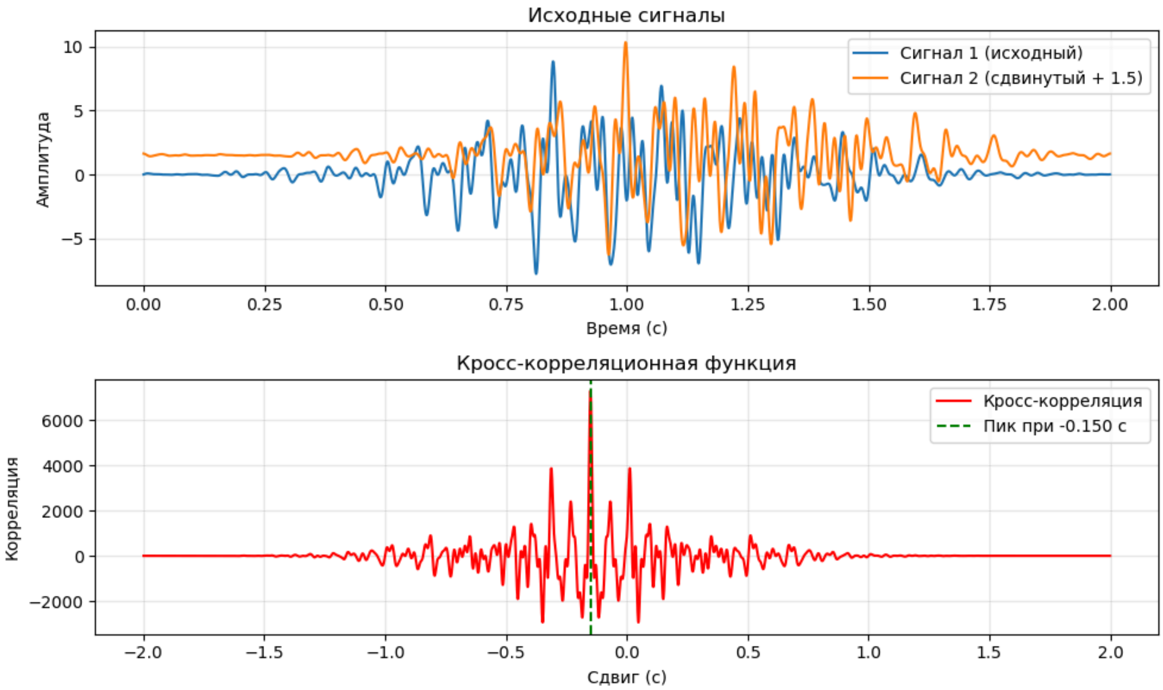
4.4. Краевые эффекты при вычислении корреляции

По умолчанию `np.correlate` вычисляет полную кросс-корреляцию (`mode='full'`), что даёт результаты для всех возможных сдвигов. При этом на краях, где перекрытие сигналов мало, значения корреляции могут быть занижены из-за меньшего числа перемножаемых отсчётов. Чтобы избежать этого, используют нормированную корреляцию или обрезают края.

4.5. Пример: определение длительности импульса через автокорреляцию



4.6. Пример: определение задержки между импульсами через кросс-корреляцию



6. Быстрая свёртка через БПФ

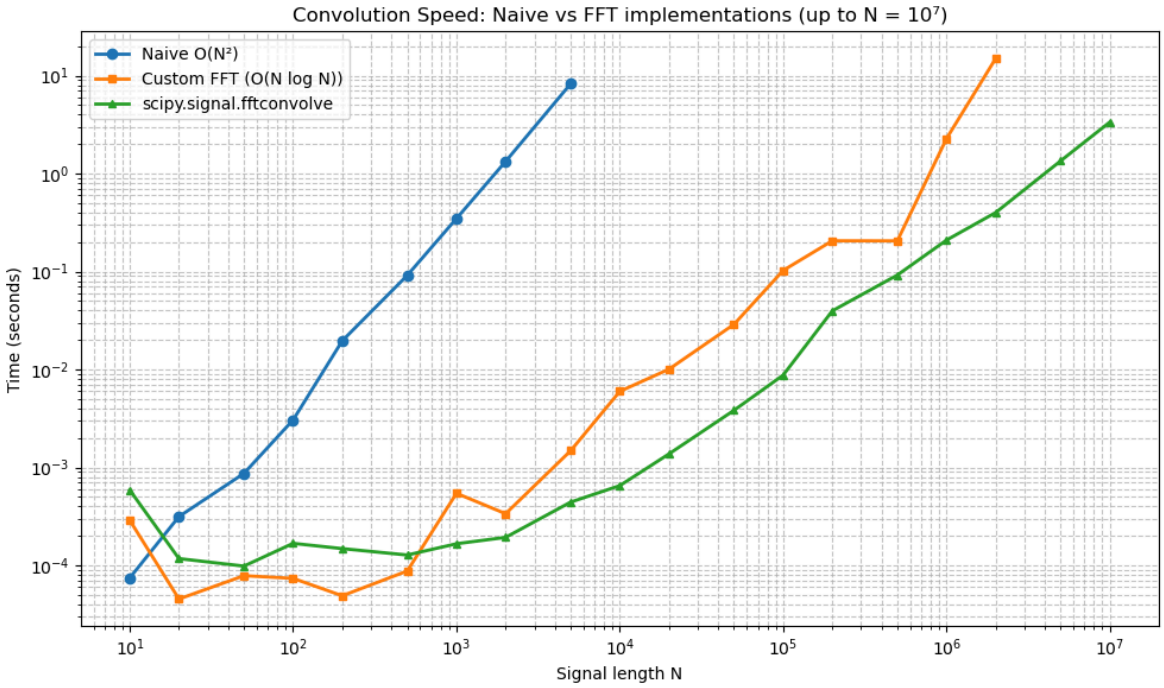
6.1. Идея и реализация

Благодаря теореме о свёртке, линейную свёртку можно вычислить как:

1. Дополнить x и h нулями до длины $L \geq N_x + N_h - 1$.
2. Вычислить БПФ обеих последовательностей.
3. Перемножить спектры поэлементно.
4. Выполнить обратное БПФ.

Сложность $O(L \log L)$ против $O(N_x N_h)$ для прямой свёртки.

6.2. Сравнение времени выполнения



Convolution Speed: Naive vs FFT implementations (up to N = 10^7)
Naive O(N²)
Custom FFT (O(N log N))
scipy.signal.fftconvolve
Time (seconds)
Signal length N
10^1
10^0
10^-1
10^-2
10^-3
10^-4
10^2
10^3
10^4
10^5
10^6
10^7