

# Программа для обработки электроэнцефалографических сигналов

Е. А. Соколова

Санкт-Петербургский государственный электротехнический университет  
«ЛЭТИ» им. В.И. Ульянова (Ленина)

evgeniia.ans@gmail.com

**Аннотация.** В статье приведено описание программного обеспечения для цифровой обработки и визуализации электроэнцефалографических сигналов. Программа позволяет отображать спектры сигналов, выполнять их вычитание и рассчитывать корреляцию. Реализована возможность одновременной работы с несколькими файлами, выделения временных интервалов и сохранения результатов работы. Применение программы позволяет сократить время на предварительную обработку сигналов и упрощает сравнение данных.

**Ключевые слова:** обработка ЭЭГ; спектр сигналов; корреляция; программное обеспечение

## I. ВВЕДЕНИЕ

Электроэнцефалография (ЭЭГ) остается одним из наиболее доступных способов исследования работы головного мозга. Существует большое количество различных электроэнцефалографов, в комплекте с которыми поставляется специальное программное обеспечение (ПО), позволяющее осуществлять обработку записанных сигналов. Однако, одним из недостатков таких решений является их коммерческая основа, что в большинстве случаев делает их недоступными для исследователей. В связи с этим разработка программы, направленной на обработку и визуализацию ЭЭГ сигналов, является актуальной задачей. Разрабатываемая программа позволяет выполнять предварительную обработку ЭЭГ сигналов, удаление выбросов с помощью порогового метода с медианной заменой, режекторную и полосовую фильтрацию заданного порядка, построение спектров сигналов, вычитание сигналов и спектров, а также расчет корреляции между сигналами или спектрами.

### A. Обзор существующих программных решений

Существует большое количество различного ПО, рассчитанного на специалистов разного уровня. Их можно разделить на коммерческие продукты, предназначенные для медицинских сотрудников, и научно-исследовательские, ориентированные на ученых и исследователей. Среди них можно выделить:

- Мицар (WinEEG, EEGStudio);
- Нейрософт (Нейрон-Спектр.NET);
- Matlab (EEGLAB, Brainstorm);
- Python (MNE-Python).

Пакеты программ WinEEG и EEGStudio от компании «Мицар» позволяют пользователям осуществлять регистрацию и обработку сигналов ЭЭГ (коррекцию

артефактов, визуальный, количественный анализ ЭЭГ сигналов и др.). Одной из особенностей EEGStudio является то, что он позволяет работать с файлами, сохраненными в международном формате EDF, записанными на электроэнцефалографах других производителей. Также EEGStudio включает в себя опцию поддержки биологической обратной связи (БОС) для тренажеров Mitsar-BFB [1]. Программа Нейрон-Спектр.NET от компании «Нейрософт» также предоставляет пользователям широкие возможности по обработке и анализу ЭЭГ сигналов: спектральный, амплитудный, корреляционный, когерентный и периодометрический анализ [2].

В то время как WinEEG, EEGStudio и Нейрон-Спектр.NET предназначены для специалистов медицинской сферы и применяются преимущественно в клинической практике, среда Matlab рассчитана в первую очередь на исследователей и инженеров.

Matlab позволяет пользователям создавать собственные приложения на его основе и поддерживает интеграцию с Python, Visual Studio и другими средствами разработки [3]. Среди наборов инструментов, предоставляемых Matlab, выделяется EEGLAB, который является одним из ведущих открытых ПО для проведения исследований в обработке ЭЭГ данных [4]. Также можно упомянуть такое приложение с открытым исходным кодом как Brainstorm [5], которое также предоставляет исследователям возможности по обработке ЭЭГ сигналов.

Для обработки ЭЭГ сигналов активно используется язык Python, предоставляющий разработчикам широкие возможности создания собственных приложений. Распространяясь бесплатно, он выгодно отличается от Matlab, а наличие MNE-Python (пакет Python с открытым исходным кодом) позволяет значительно упростить анализ и визуализацию нейрофизиологических данных [6].

Таким образом, рассмотрев некоторые существующие программы, было принято решение использовать язык Python как наиболее доступный. На текущем этапе разработки не используется библиотека MNE, но при расширении функционала планируется ее добавление.

### B. Интерфейс и структура программы

Графический интерфейс программы реализован в среде Spyder с помощью Tkinter, стандартной библиотеки для создания пользовательского интерфейса. Построение графиков выполнялось с помощью matplotlib.

Работа выполняется в рамках государственного задания №FSEE-2025-0015

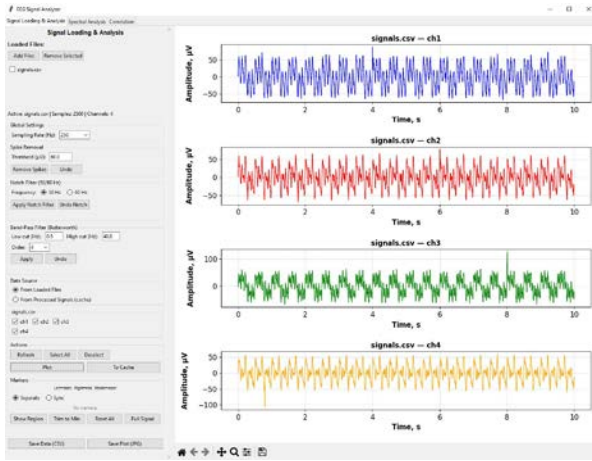


Рис. 1. Окно программы (Signal Loading & Analysis). Левая половина программы включает в себя функциональные элементы, а справа выводятся сигналы выбранных отведений. По оси X – отмеряется время в секундах, а по оси Y – Амплитуда в микровольтах.

Окно программы включает в себя такие возможности, как загрузка и удаление файлов («Add Files», «Remove Selected»), выбор частоты дискретизации («Sampling Rate (Hz)»), удаление выбросов («Spike Removal»), фильтрацию сетевых помех («Notch Filter (50/60 Hz)»), полосовую фильтрацию («Band-Pass Filter (Butterworth)») и другие. На рис. 1, приведен интерфейс первой страницы программы. На вкладке можно просмотреть загруженный файл, выбрать необходимые каналы из списков, расставить метки с помощью левой и правой кнопки мыши. Метки можно располагать как синхронно для каждого из выбранных сигналов («Sync»), так и раздельно («Separate»), с их помощью можно выделить исследуемую часть сигнала, например, для удаления выбросов («Remove Spikes»).

Программа получила название «EEG Signal Analyzer» и содержит в себе три вкладки.

- Signal Loading & Analysis (загрузка и анализ сигналов). Позволяет осуществить загрузку одного или нескольких файлов, выбрать частоту дискретизации (250, 500, 1000). Осуществляет построение выбранных каналов из загруженных файлов друг под другом. Добавлена возможность применение режекторной фильтрации (50 или 60 Гц), полосового фильтра Баттерворта с выбором порядка для выделения необходимой частоты в сигнале и адаптивной очистки выбросов. Кроме того, у пользователя есть возможность выбирать интересующие его участки сигнала с помощью специальных маркеров, которые также используются для выделения и устранения локальных выбросов.
- Spectral Analysis (Спектральный анализ). Расчет спектров сигналов, осуществление выбора оконной функции (прямоугольная, Ханна, Хэмминга), настройки ее параметров и применения метода Уэлча. Также можно осуществить вычитание выбранных сигналов или спектров.
- Correlation (Корреляция). Расчет коэффициентов корреляции Пирсона и Спирмена между выбранными сигналами или спектрами.

Построенные в программе графики можно сохранить как изображение в формате jpg («Save Plot (JPG)»), а также как файл csv («Save Data (CSV)»). Кроме того, реализована возможность сохранения (сигналов, спектров, результатов вычитания) в кэш программы («To Cache»). На рис. 2, представлена структурная схема разработанной программы.

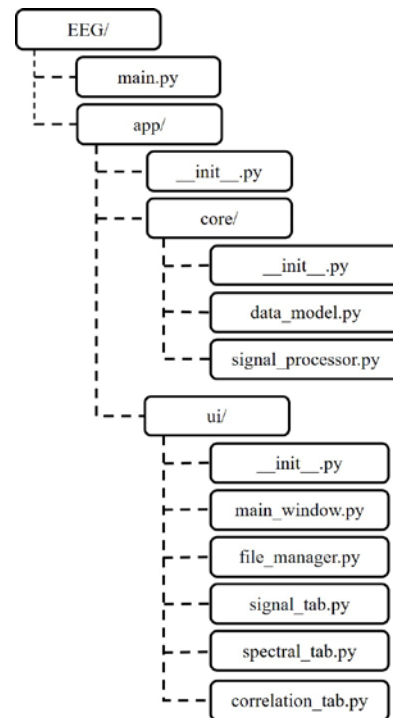


Рис. 2. Структурная схема разработанной программы

Согласно структурной схеме (рисунок 2), программа построена по модульному принципу и разделена на несколько блоков. В core осуществляется загрузка файлов с определением их формата (data\_model.py) и цифровая обработка сигналов (signal\_processor.py). В ui строится пользовательский интерфейс и включает в себя главное окно (main\_window.py), блок работы с файлами (file\_manager.py), вкладку загрузки и фильтрации сигналов (signal\_tab.py), вкладка спектрального анализа (spectral\_tab) и расчета корреляции (correlation\_tab.py).

## II. ФУНКЦИОНАЛ ПРОГРАММЫ

В основе функционала программы лежат методы цифровой обработки сигналов и статистики, включая быстрое преобразование Фурье, метод Уэлча, режекторную и полосовую фильтрацию, адаптивное удаление выбросов и корреляционный анализ. Реализация осуществлялась с помощью стандартных библиотек Numpy, Pandas, Scipy.

### A. Загрузка и обработка файлов

Программа позволяет загружать и обрабатывать данные из csv-файлов, при этом поддерживаются файлы разных кодировок (UTF-8, Windows-1251, cp1252). Формат данных, а файлах предполагает, что заголовки столбцов обозначают отведения, программа автоматически их выделяет и дает пользователю возможность выбрать необходимые для построения графиков и последующей обработки сигналов.

### В. Режекторная фильтрация

Режекторная фильтрация была реализована на основе фильтра с бесконечной импульсной характеристикой (БИХ) второго порядка (`scipy.signal.iirnotch` [7]), затем для получения нулевого фазового сдвига использовался `scipy.signal.filtfilt` [8]. Пользователь может выбрать частоту режекторного фильтра (50 или 60 Гц).

### С. Полосовая фильтрация

Полосовой фильтр Баттерворта с выбором настройки порядка (2, 4, 6), позволяет выделить необходимую полосу пропускания из сигнала. Таким образом, можно оставить только те частоты, которые соответствуют исследуемому ритму: дельта (0,5-4 Гц), тета (4-7 Гц), альфа (8-13 Гц), бета (13-30 Гц), гамма (30-100 Гц) [9]. Фильтр был реализован с помощью функций `butter`, `filtfilt` библиотеки `SciPy`.

### Д. Адаптивная очистка выбросов

В качестве метода удаления артефактов был выбран медианный фильтр с возможностью выбора порога. Принцип работы такого фильтра основан на том, что для выделенного маркерами участка сигнала находятся все отсчеты, абсолютное значение которых превышает заданный пользователем порог. Для каждого такого всплеска формируется окно, включающее в себя по два соседних отсчета слева и справа (до пяти значений). Из этого окна исключаются те значения, которые также превышают порог. Затем выброс заменяется на медиану оставшихся значений. Таким образом сглаживаются только те значения в сигнале, которые являются артефактами, не затрагивая полезный сигнал.

### Е. Получение спектров сигналов

Для получения амплитудного и фазового спектра сигнала использовался метод, основанный на дискретном преобразовании Фурье. Программа поддерживает три типа окон для построения спектров сигналов (прямоугольное, Ханна, Хэмминга), также пользователь по желанию может применить метод Уэлча. Для реализации оконных функций используются методы из библиотеки `NumPy` (`np.ones` [10], `np.hanning` и `np.hamming` [11]), для реализации метода Уэлча – функция `scipy.signal.welch` [12].

### Ф. Расчет коэффициентов корреляции

Программа позволяет вычислять корреляции Пирсона и Спирмена посредством встроенных средств библиотеки `scipy` (`scipy.stats.pearsonr`, `scipy.stats.spearmanr` [13]). При этом пользователь может выбрать, для каких именно спектров или сигналов ее рассчитывать.

## III. ТЕСТИРОВАНИЕ РАБОТЫ ПРОГРАММЫ

Для проверки функциональности программы был сгенерирован набор сигналов (`test_signals.csv`), соответствующих пяти каналам, с частотой дискретизации 250 Гц, длительностью 2 секунды и числом отсчетов 500.

- Канал 1 (Ch1): сумма синусоид для проверки полосовой фильтрации с добавлением сетевой наводки, а также выбросов.
- Каналы 2 и 3 (Ch2, Ch3): две синусоиды для проверки вычитания.

- Каналы 4 и 5 (Ch4, Ch5): две синусоиды с небольшим зашумлением для проверки корреляции.

### А. Создание тестовых сигналов

Первый канал (Ch1) состоит из синусоиды с частотой 5 Гц и амплитудой 30 мкВ, которая рассчитывалась по формуле  $x_1(t) = 30 \sin(2\pi \cdot 5t)$ , после чего к полученной синусоиде добавлялись выбросы на 0,5 и 1,5 секундах. Они создавались следующим образом:  $x_1(t)' = x_1(t) + 300\delta(t - 0.5) + 300\delta(t - 1.5)$ , где  $\delta$  – единичный импульс. Затем к нему были добавлены сигналы, имитирующие помехи от электросети и высокочастотный шум. Полученный сигнал создавался согласно формуле  $x_1(t)'' = x_1(t)' + 10 \sin(2\pi \cdot 50t) + 15 \sin(2\pi \cdot 45t)$ , где сетевая наводка имеет амплитуду 10 мкВ и частоту 50 Гц, а дополнительная высокочастотная помеха 15 мкВ и 45 Гц.

Для проверки вычитания двух сигналов, были созданы две синусоиды с частотой 10 Гц. Второй канал (Ch2)  $x_2(t) = 40 \sin(2\pi \cdot 10t)$ , с амплитудой сигнала 40 мкВ и канал 3 (Ch3)  $x_3(t) = 70 \sin(2\pi \cdot 10t)$ , с амплитудой 70 мкВ. При проведении вычитания должен быть получен сигнал  $x_3(t) - x_2(t) = 30 \sin(2\pi \cdot 10t)$ .

Четвертый и пятый канал (Ch4, Ch5) являются одинаковой синусоидой вида  $x_{4-5}(t) = 25 \sin(2\pi \cdot 10t)$ , к каждой из которых добавлялся случайный шум  $x_4(t) = x_{4-5}(t) + \varepsilon(t)$  и  $x_5(t) = x_{4-5}(t) + \varepsilon(t)$ , где  $\varepsilon(t)$  – помеха, генерируемая с помощью функции `np.random.randn` [14]. Благодаря минимальным различиям, полученные сигналы обладают высокой степенью корреляции.

### В. Алгоритм тестирования

Процесс тестирования работы программы включал в себя следующие шаги.

1) Загрузка файла `test_signals.csv` и построения графика сигналов для каждого из каналов.

Графики загруженных сигналов представлены на рис. 3.

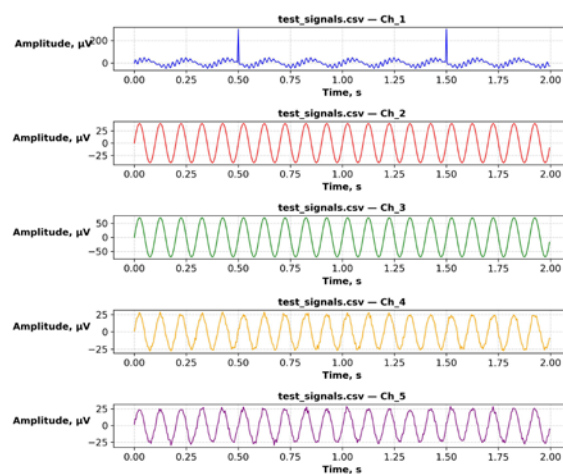


Рис. 3. Графики загруженных тестовых сигналов для отведений Ch1, Ch2, Ch3, Ch4, Ch5 файла «test\_signals.csv»; где Ch1 – сигнал с выбросами, сетевой наводкой и высокочастотным шумом; Ch2, Ch3 – две идеальные синусоиды; Ch4, Ch5 – две синусоиды с добавлением случайного шума

Согласно рисунку 3, на вход программы поданы 5 тестовых сигналов, нумерация идет сверху вниз. По оси X указано время, а по оси Y – амплитуда сигнала. Уравнения представленных сигналов приведены в разделе 3 пункт А.

### 2) Фильтрация сигнала Ch1.

Из сигнала Ch1 были удалены все выбросы с помощью адаптивного порогового метода (порог 60 мкВ), затем был применен режекторный фильтр 50 Гц, после чего была применена полосовая фильтрация четвертого порядка в диапазоне от 1 до 10 Гц. Полученный результат представлен на рис. 4.

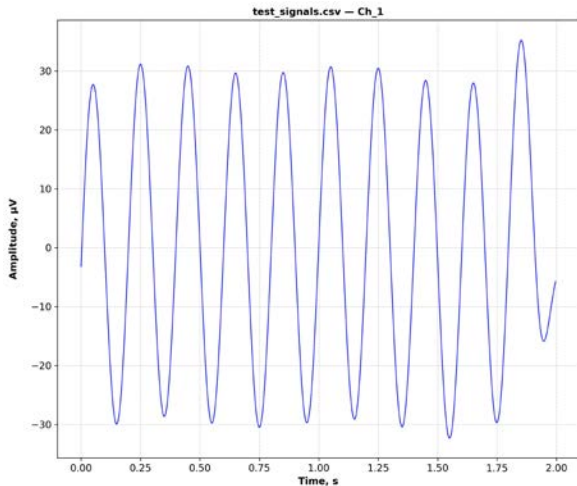


Рис. 4. Сигнал Ch1 после удаления выбросов, сетевой наводки (50 Гц) и полосовой фильтрации (1-10 Гц)

Согласно рис. 4, полученный сигнал стал наиболее близок к исходному сигналу (до зашумления). Исходная амплитуда составляла 30 мкВ, а частота 5 Гц, что можно увидеть на графике сигнала после фильтрации. Всплеск в конце сигнала может быть связан с краевым эффектом.

### 3) Вычитание спектров Ch3 – Ch2.

Поскольку сигналы Ch2, Ch3 представляют собой идеальные синусоиды, на графиках их амплитудный и фазовый спектры (прямоугольное окно) изображены в виде одиночных пиков на частоте 10 Гц. Результат вычитания спектров Ch3 – Ch2, приведен на рис. 5.

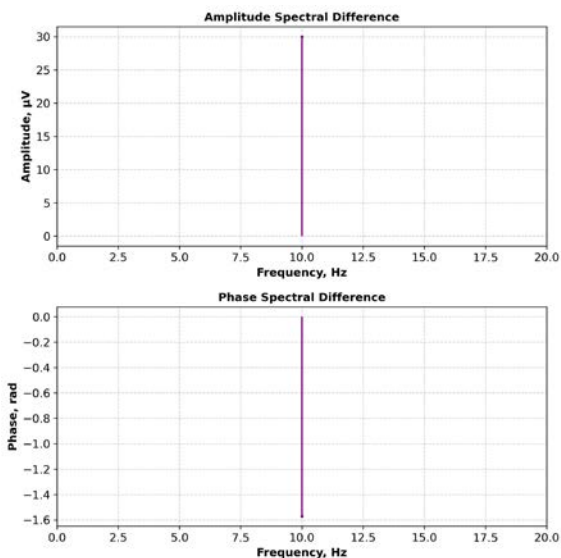


Рис. 5. График спектра, полученного после вычитания Ch3 – Ch2

Как видно из рис. 5, разность амплитуд между Ch3 (70 мкВ) и Ch2 (40 мкВ), равна 30 мкВ, что полностью соответствует ожидаемому результату.

### 4) Корреляция между сигналами Ch5 и Ch4.

Сигналы Ch5 и Ch4 практически равны между собой, они отличаются только наличием небольших шумов. Таким образом ожидаемая корреляция должна быть близка к 1. В результате расчетов программой, результат корреляции составил  $r=0,987$  (Пирсона) и  $\rho=0,977$  (Спирмена), что соответствует ожидаемому результату. Результат корреляции приведен на рис. 6.

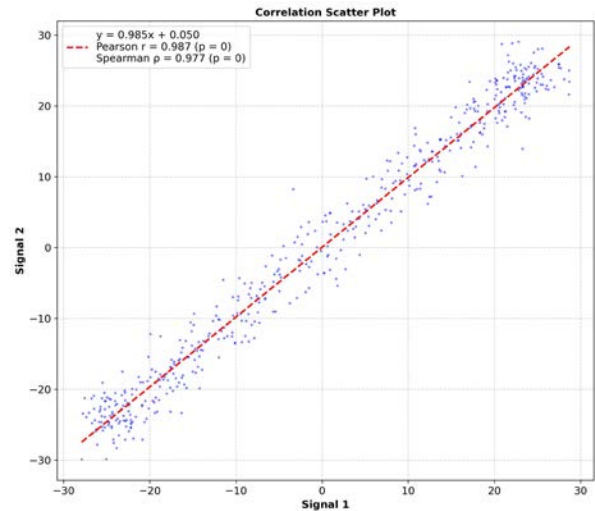


Рис. 6. Результат корреляции между сигналами, где signal 1 - Ch4, а signal 2 - Ch5

Благодаря проведенным тестам функциональность программы была проверена, каждая из команд выполняет поставленную перед ней задачу. Таким образом можно проводить исследования на записях ЭЭГ сигналов.

## IV. ПРОВЕРКА НА ЭЭГ СИГНАЛЕ

После проверки функциональности программы на тестовых сигналах, были загружены два файла EEG\_1.csv и EEG\_2.csv, первый соответствует записи ЭЭГ в покое при закрытых глазах, а второй ЭЭГ при закрытых глазах при поднятии руки. В качестве примера были выбраны сигналы с электродов Fp2. Частота дискретизации для обоих сигналов составила 500 Гц. Графики загруженных необработанных сигналов для отведения Fp2, представлены на рис. 7.

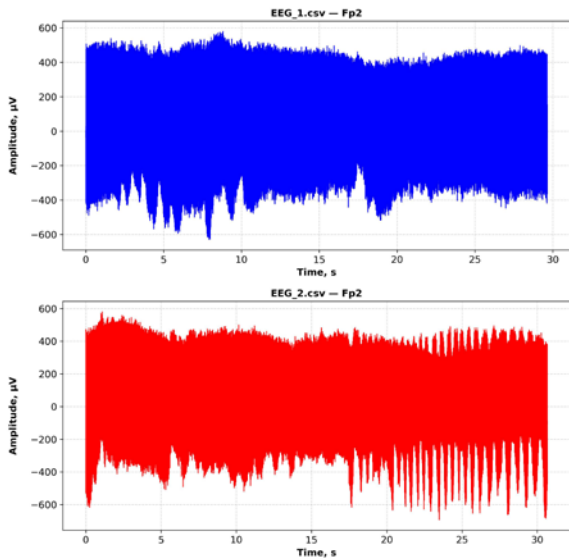


Рис. 7. Необработанные сигналы ЭЭГ из файлов EEG\_1, EEG\_2 для отведения Fp2

По графикам видно, что необработанный сигнал сильно зашумлен и требует предварительной фильтрации. После применения режекторной (50 Гц) и полосой фильтрации четвертого порядка в диапазоне от 8 до 13 Гц, были получены сигналы на рис. 8. Выброс вначале сигнала, вероятно, обусловлен краевым эффектом, возникающим после фильтрации.

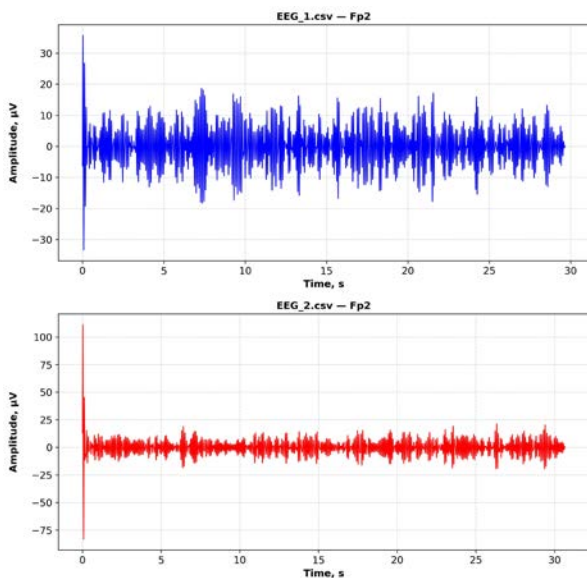


Рис. 8. Сигналы ЭЭГ из файлов EEG\_1, EEG\_2 для отведения Fp2 после проведения режекторной и полосовой фильтрации

Можно заметить, что применение режекторной и полосовой фильтрации позволило получить сигнал, соответствующий альфа-ритму. Амплитуда значительно снизилась с приблизительно 500 мкВ (необработанного сигнала), до примерно 20 мкВ.

После фильтрации сигнал был обрезан в диапазоне от 0,5 до 28,25 секунд. Для полученных сигналов были построены спектры (прямоугольное окно), графики которых представлены на рис. 9.

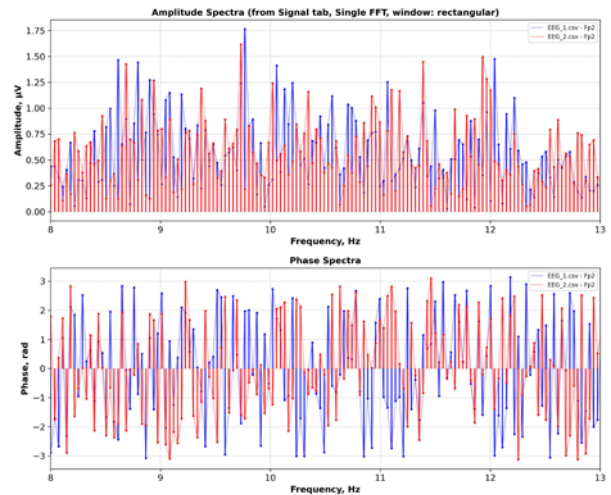


Рис. 9. Спектры сигналов EEG\_1 (Fp2) и EEG\_2 (Fp2)

Согласно рис. 9, спектры сигналов Fp2 в покое (EEG\_1) и при движении руки (EEG\_2) визуально схожи, что позволяет предположить высокую степень корреляции между ними. Рассчитанная корреляция для представленных спектров, приведена на рис. 10.

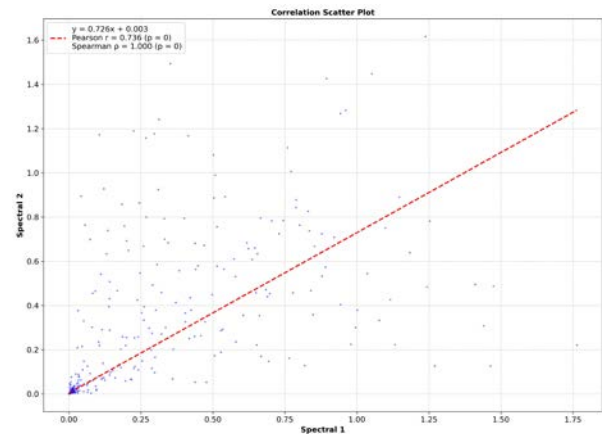


Рис. 10. Корреляция между спектрами сигналов, где spectral 1 и spectral 2 соответствуют EEG\_1, EEG\_2

Можно заметить, что значение корреляция между двумя спектрами  $r=0,736$  (Пирсона) и  $\rho=1,000$  (Спирмена), что соответствует предположению и говорит о сильной положительной корреляции.

Дополнительно было проведено вычитание спектров EEG\_2 (Fp2) – EEG\_1 (Fp2). Полученный результат приведен на рис. 11.

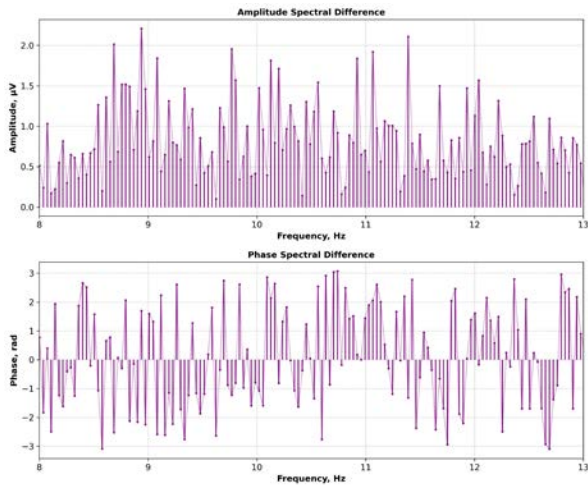


Рис. 11. Спектр полученный после вычитания EEG\_2 (Fp2) – EEG\_1 (Fp2)

Можно заметить, что согласно результату вычитания, значение амплитуды немного выше в сравнении с исходными спектрами сигналов. В дальнейшем можно провести более расширенное исследование по вычитанию спектров сигналов.

## V. ЗАКЛЮЧЕНИЕ

В ходе выполнения работы были рассмотрены существующие программные решения для обработки и анализа ЭЭГ сигналов. Разработана собственная программа, ее работоспособность проверена на тестовых данных. В результате исследования была проведена предварительная обработка ЭЭГ сигналов, получены их амплитудные и фазовые спектры, вычислена корреляция между сигналами и разность между спектрами.

Полученная программа и представленные результаты являются первым этапом подготовки к проведению углубленных исследований, направленных на декомпозицию сигналов и выделение из них определенных составляющих (например, двигательной активности на ЭЭГ сигналах). В дальнейшем планируется расширение функционала программы, а также разработка улучшенного метода декомпозиции сигналов.

## СПИСОК ЛИТЕРАТУРЫ

- [1] Программное обеспечение EEGStudio. Available at: <https://mitsar-eeeg.ru/page.php?id=eeegstudio> (accessed 30 April 2026)
- [2] Иванов А.А. Обзор возможностей современного программного обеспечения для регистрации и анализа ЭЭГ. Эпилепсия и пароксизмальные состояния. 2023. 15(1). С. 53-69. doi: 10.17749/2077-8333/epi.par.con.2023.144
- [3] MATLAB. Available at: <https://ch.mathworks.com/products/matlab.html> (accessed 30 April 2026)
- [4] R. Martínez-Cancino, A. Delorme, D. Truong, F. Artoni, K. Kreutz-Delgado, S. Sivagnanam, K. Yoshimoto, A. Majumdar, and S. Makeig, "The open EEGLAB portal interface: High-Performance computing with EEGLAB," *NeuroImage*. 2021 Jan 1, 224, 116778. doi: 10.1016/j.neuroimage.2020.116778
- [5] F. Tadel, S. Baillet, J.C. Mosher, D. Pantazis, R.M. Leahy, "Brainstorm: a user-friendly application for MEG/EEG analysis," *Comput Intell Neurosci*. 2011, 2011:879716. doi:10.1155/2011/879716
- [6] A. Gramfort, M. Luessi, E. Larson, D.A. Engemann, D. Strohmeier, C. Brodbeck, R. Goj, M. Jas, T. Brooks, L. Parkkonen, M. Hämäläinen, "MEG and EEG data analysis with MNE-Python," *Front Neurosci*. 2013 Dec 26, 7:267. doi: 10.3389/fnins.2013.00267
- [7] Iirnotch. Available at: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.iirnotch.html> (accessed 30 April 2026)
- [8] Fitfilt. Available at: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.filtfilt.html> (accessed 30 April 2026)
- [9] A. Suri, M. A. Wani, S. Jabin, M. Khan, K. I. K. Sherwani, M. Sardar, M. Salman, "BMT\_EEG: A Novel EEG Dataset for Evaluating the Effects of New Protocols on Biometric Authentication Systems," in *IEEE Access*, vol. 14, pp. 49535-49549, 2026, doi: 10.1109/ACCESS.2026.3678803.
- [10] Numpy.ones. Available at: <https://numpy.org/doc/stable/reference/generated/numpy.ones.html> (accessed 30 April 2026)
- [11] Window functions. Available at: <https://numpy.org/doc/stable/reference/routines.window.html> (accessed 30 April 2026)
- [12] Welch. Available at: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.welch.html> (accessed 30 April 2026)
- [13] Statistical functions (scipy.stats). Available at: <https://docs.scipy.org/doc/scipy/reference/stats.html> (accessed 30 April 2026)
- [14] Numpy.random.randn. Available at: <https://numpy.org/doc/stable/reference/random/generated/numpy.random.randn.html> (accessed 30 April 2026)