1 Описание задачи

Требуется написать алгоритм, распознающий звук хлопка в произвольном звуковом потоке.

2 О данных...

Чтобы проверять корректность работы и сравнивать алгоритмы между собой, я записала 8 тестовых аудиозаписей длительностью по 3 секунды. Частота аудиозаписи 44100 Гц.

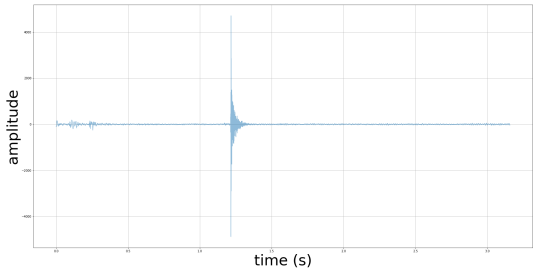


Рис. 1: Хлопок

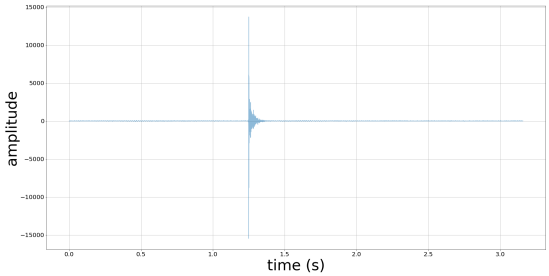


Рис. 2: Хлопок

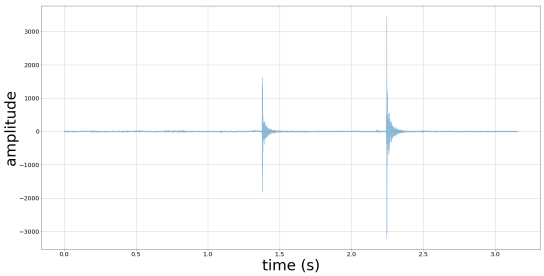


Рис. 3: 2 хлопка

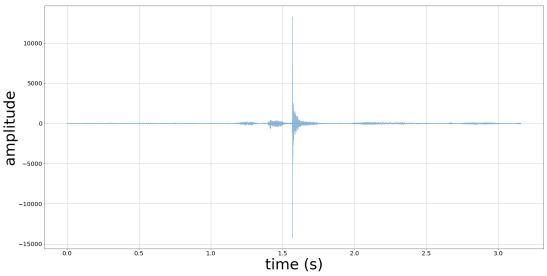


Рис. 4: Хлопок на фоне речи

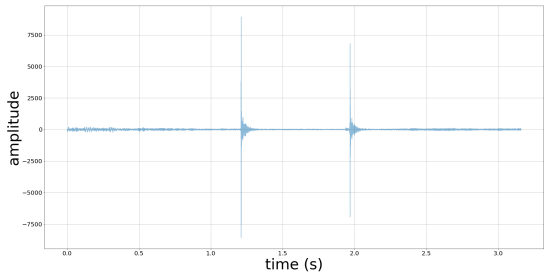


Рис. 5: 2 хлопка на фоне музыки

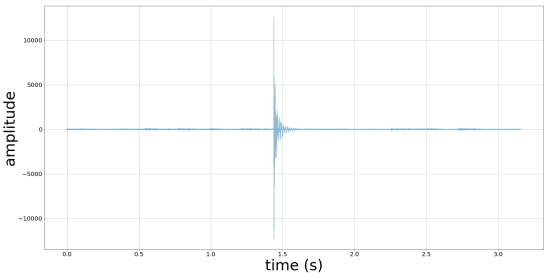


Рис. 6: Удар на фоне музыки

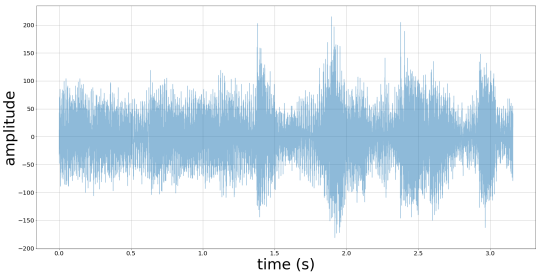


Рис. 7: Музыка

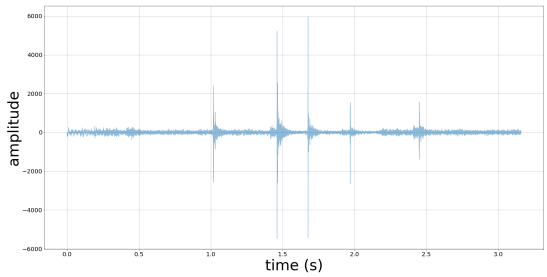


Рис. 8: Несколько хлопков на фоне музыки

Для шаблона я взяла фрагмент аудиозаписи с хлопком, записанной также с частотой 44100 Гц, состоящий из 229 отсчетов (примерно 0.005 секунды). Длительность шаблона выбиралась таким образом, чтобы захватить первые несколько звуковых колебаний в момент хлопка.

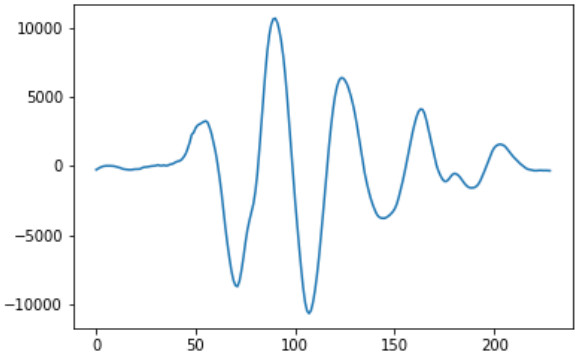
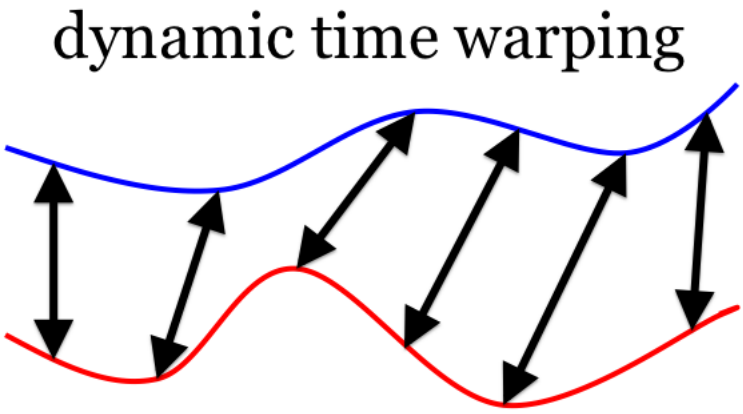


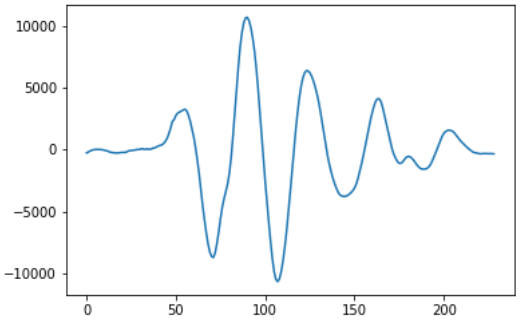
Рис. 9: Шаблон

3 Описание реализованного алгоритма

Я воспользовалась алгоритмом динамической трансформации временной шкалы(пакет dtw).Он позволяет измерять расстояние между временными рядами, не обращая внимание на локальные и глобальные сдвиги по временной шкале.



Последовательно применяем алгоритм к шаблону и части временного ряда, попадающей в скользящее окно ширины 2×229. Скользящее окно движется по ряду с шагом 229. Таким образом, для подотрезков ряда получаем расстояния до шаблона. Подотрезки, для которых расстояние достаточно меньше среднего по врем подотрезкам, принимаем за интервалы с хлопком.


10000
5000
0
−5000
−10000
0
50
100
150
200

3.1 Результат работы алгоритма

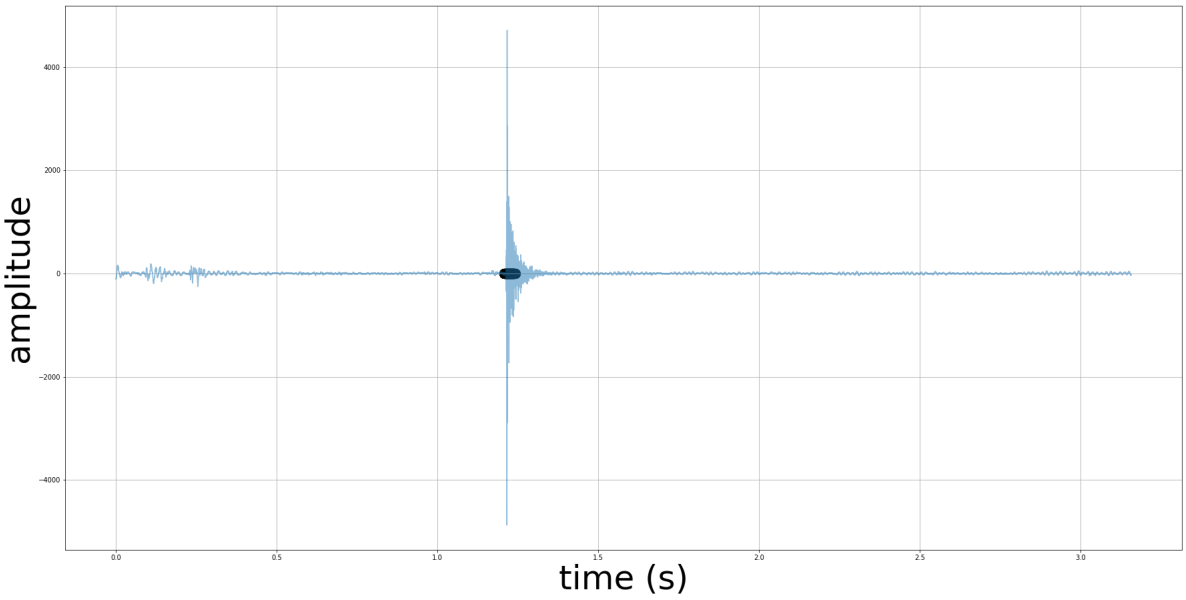


Рис. 10: Хлопок

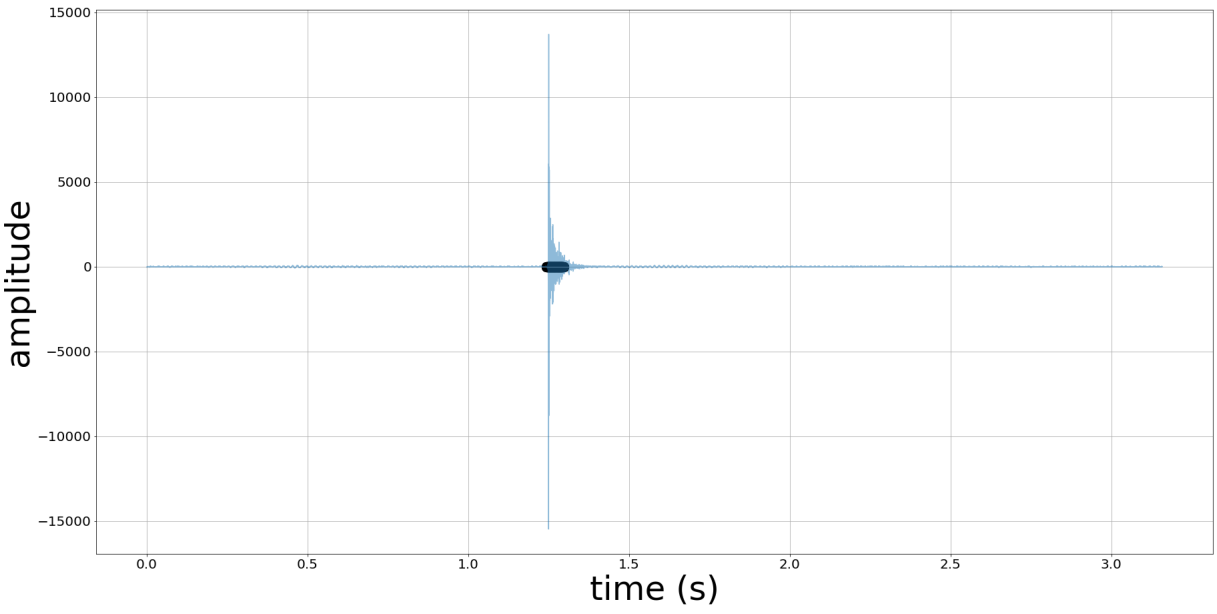


Рис. 11: Хлопок

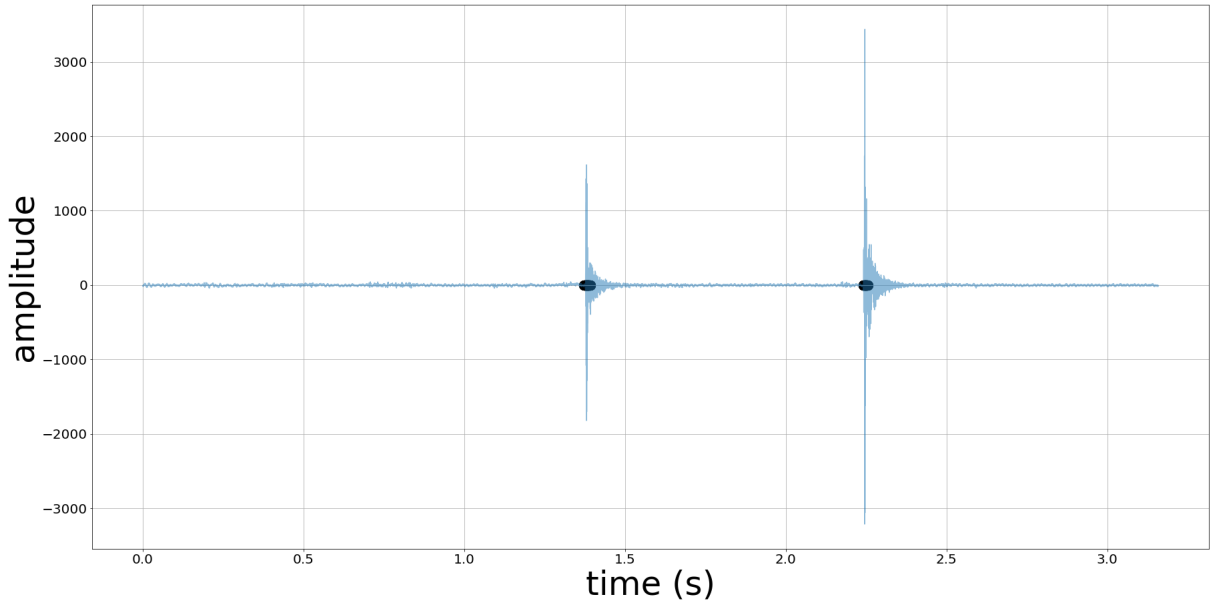


Рис. 12: 2 хлопка

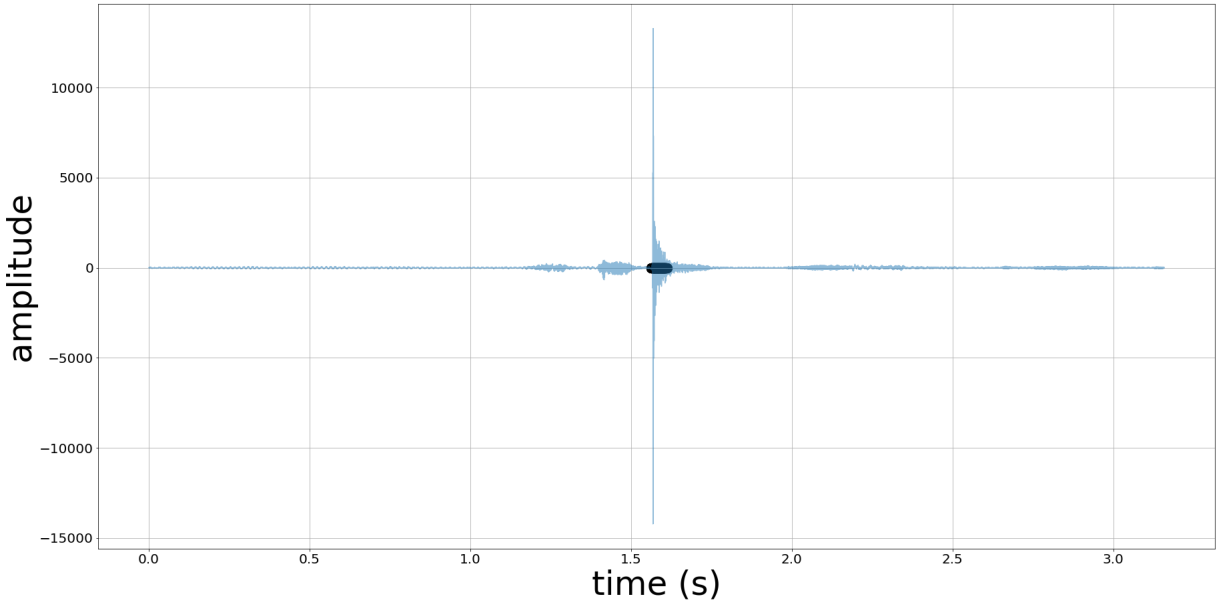


Рис. 13: Хлопок на фоне речи

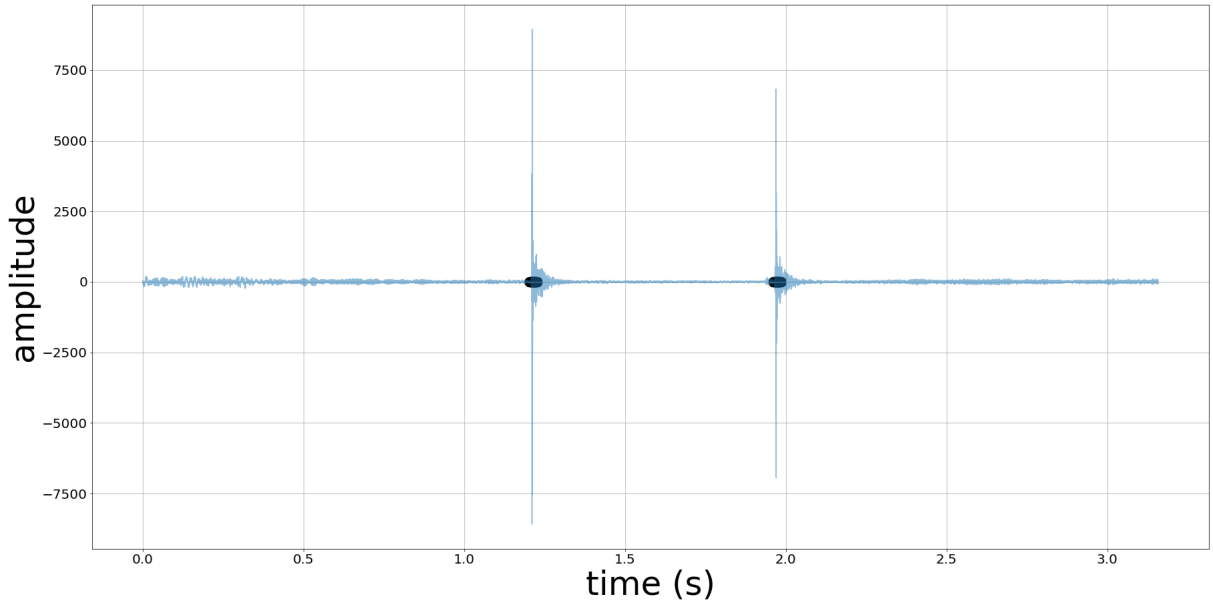


Рис. 14: 2 хлопка на фоне музыки

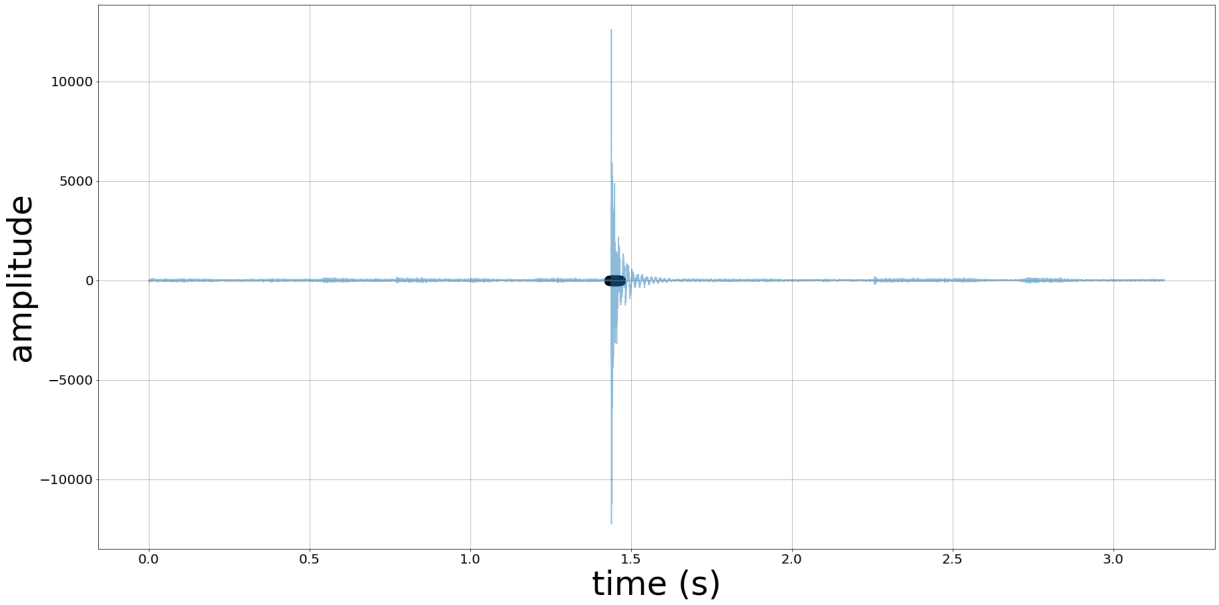


Рис. 15: Удар на фоне музыки

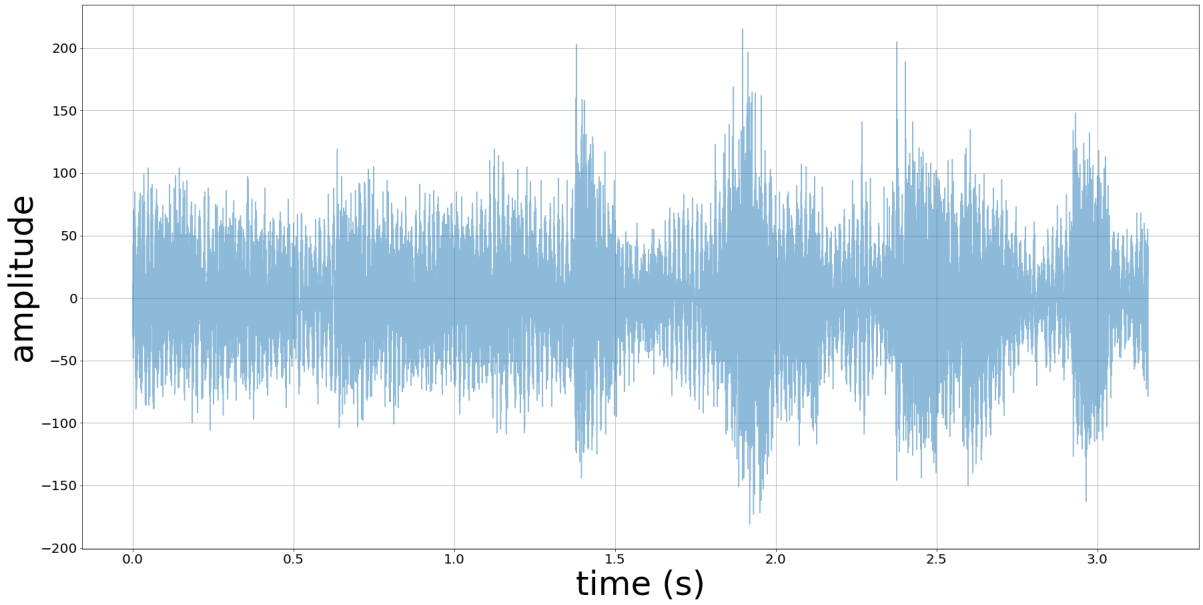


Рис. 16: Музыка

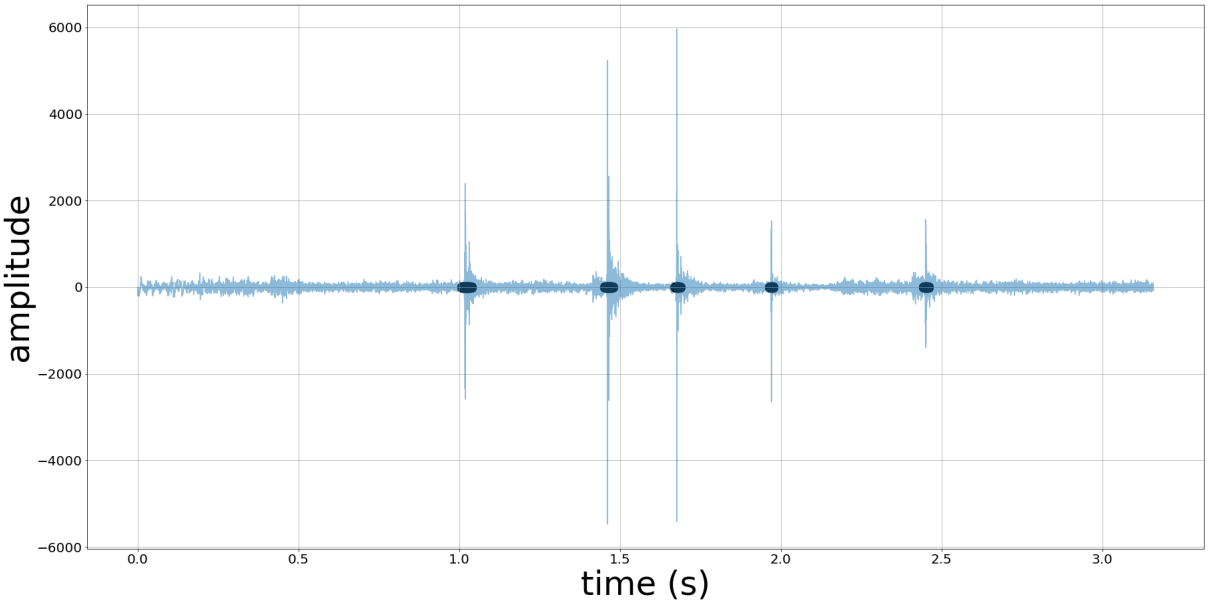


Рис. 17: Несколько хлопков на фоне музыки

3.2 Время работы алгоритма

При вычислении расстояния между двумя временными рядами длины n и m работает за $O(nm)$. Если имеется запись из N отсчетов, то время составит $O(\frac{N}{229} \times 229 \times 229 \times 2) = O(N)$

3.3 Недостатки алгоритма

Расстояние от шаблона до фрагмента с громким резким ударом оказывается тоже меньше среднего, что ведет к ошибочным положительным ответам алгоритма.

4 Другие рассмотренные алгоритмы

4.1 Кросс-корреляция

(функция numpy.correlate)

Алгоритм кросс-корреляции последовательно применят свертку шаблона с отрезком ряда. Отрезки, где корреляция с шаблоном больше, чем порог(`THRESHOLD`), считаем отрезками с хлопком. Для того, чтобы использовать фиксированный порог, перед применением алгоритма выполняем нормировку шаблона и временного ряда - вычитаем среднее и делим на стандартное отклонение.

Алгоритм кросс-корреляции дает верные положительные ответы для громких хлопков.

Достоинство алгоритма - скорость(линейное время работы + быстрая реализация вычислений в библиотеке numpy), хорошая точность - все хлопки из теста были обнаружены.

Недостатки такие же, как и у предыдущего алгоритма: ложный положительный результат для удара. Также, необходимо экспериментально подбирать оптимальное значение порога.

4.1.1 Результат работы алгоритма на основе кросс-корреляции

Ниже представлены результаты работы для THRESHOLD = 400.

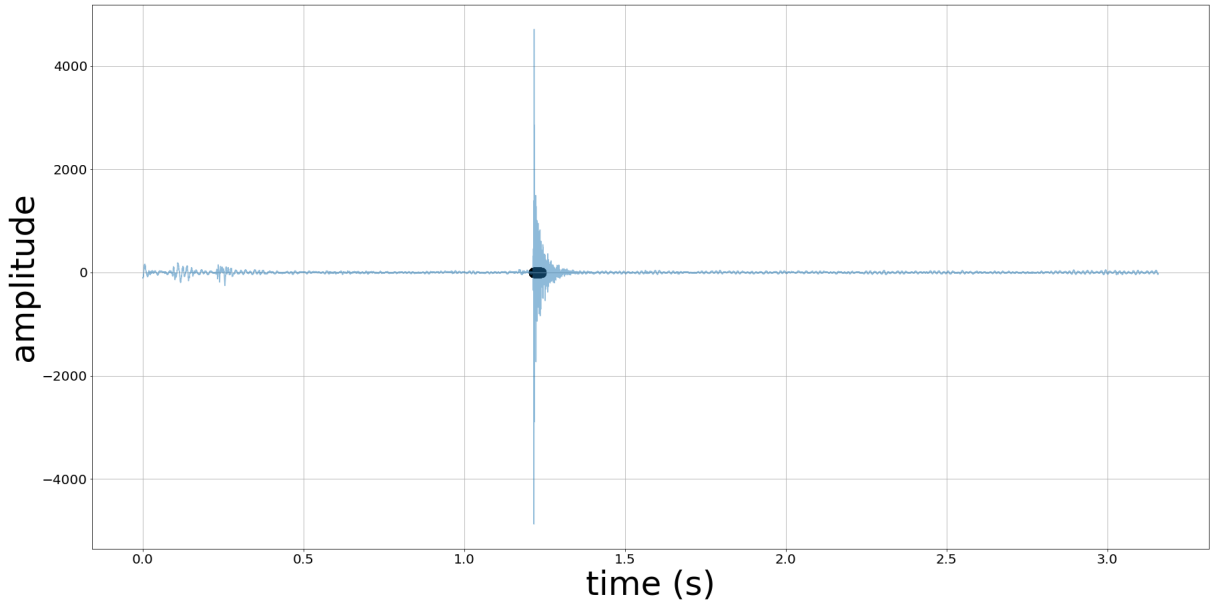


Рис. 18: Хлопок

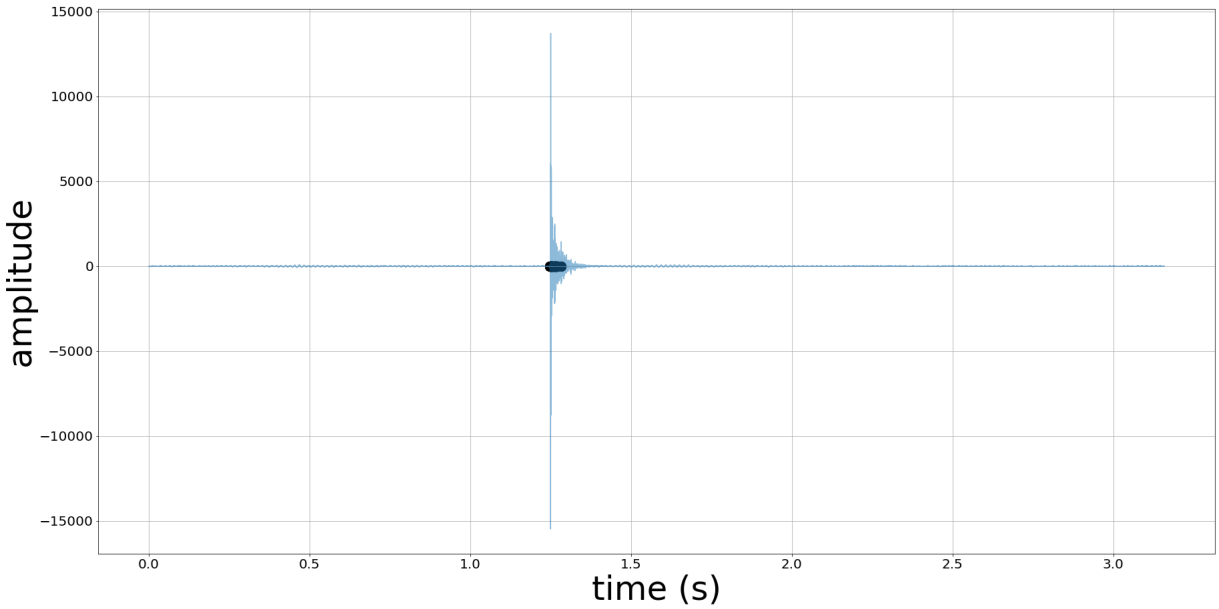


Рис. 19: Хлопок

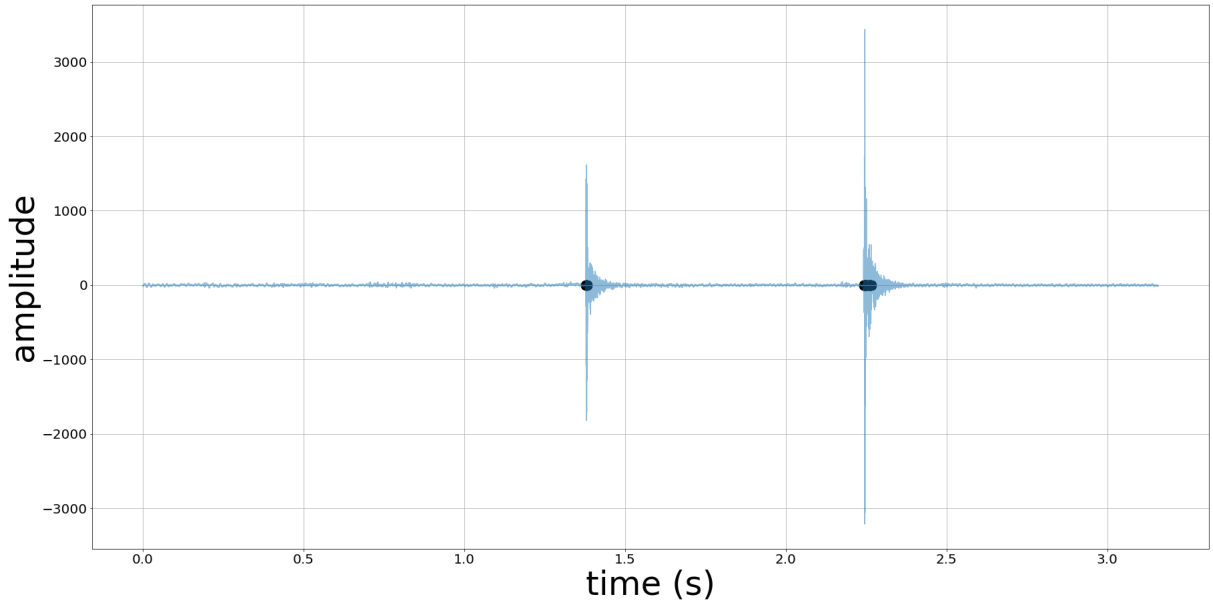


Рис. 20: 2 хлопка

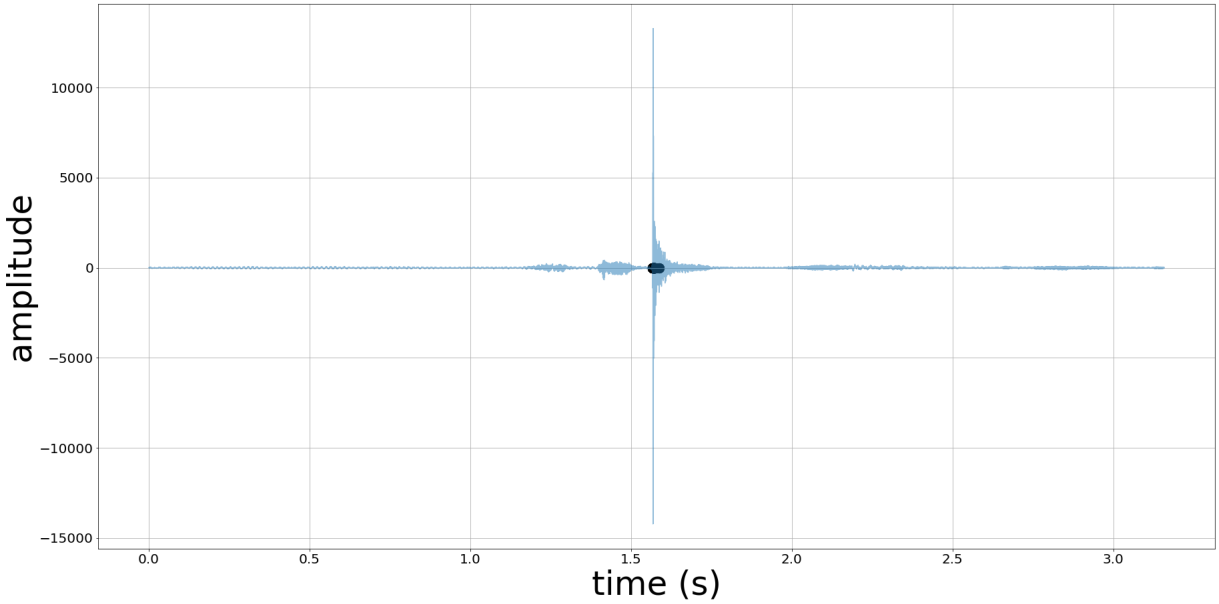


Рис. 21: Хлопок на фоне речи

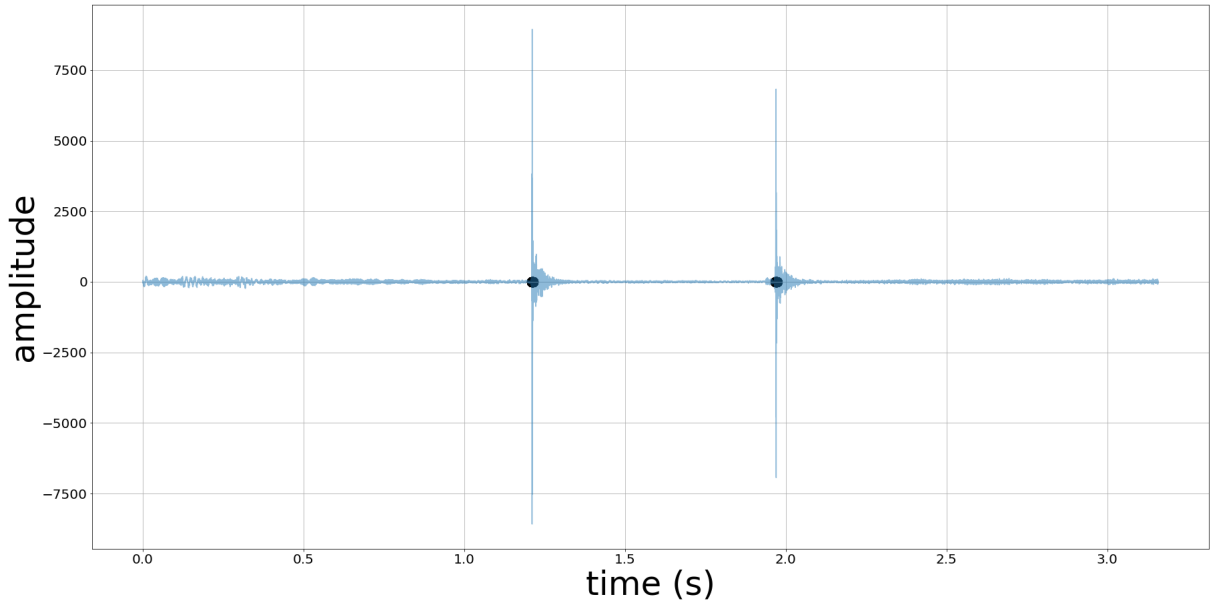


Рис. 22: 2 хлопка на фоне музыки

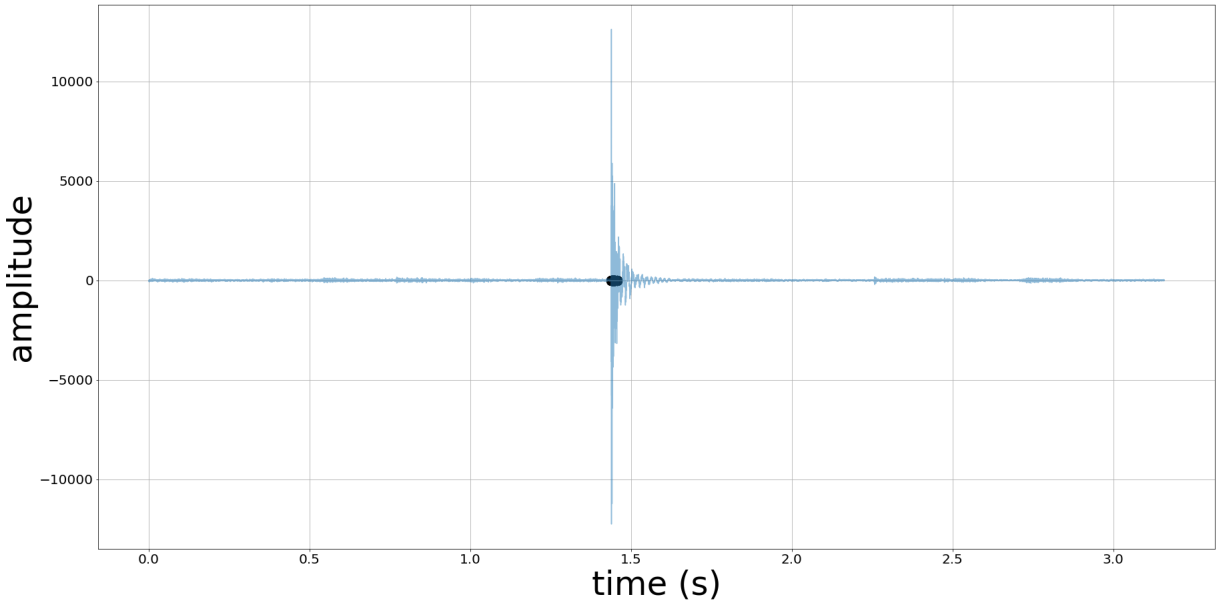


Рис. 23: Удар на фоне музыки

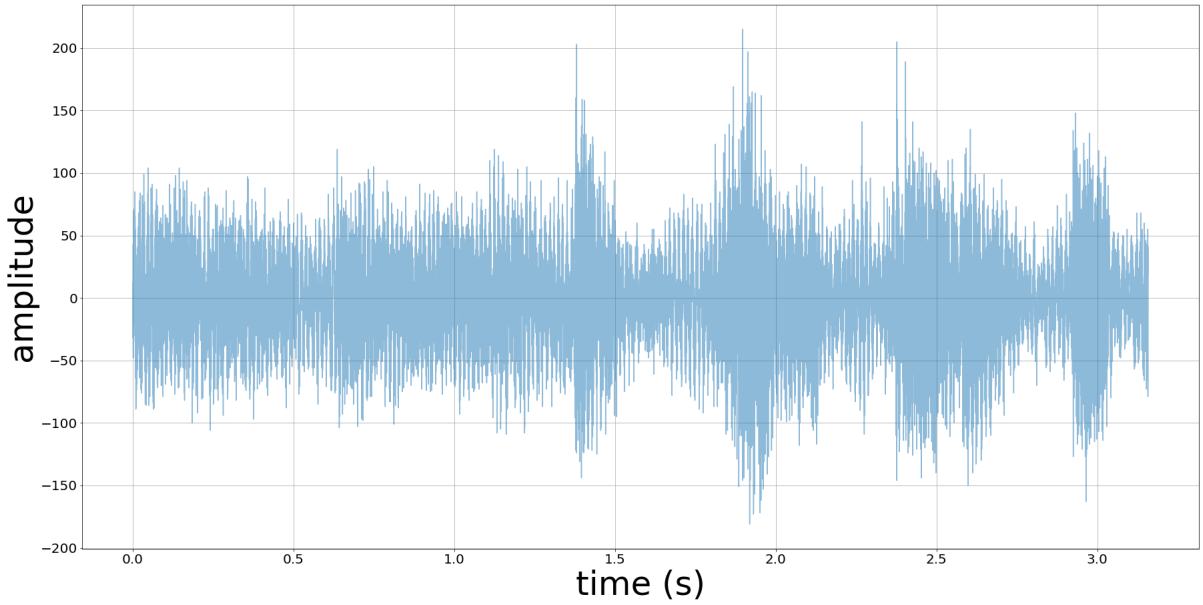


Рис. 24: Музыка

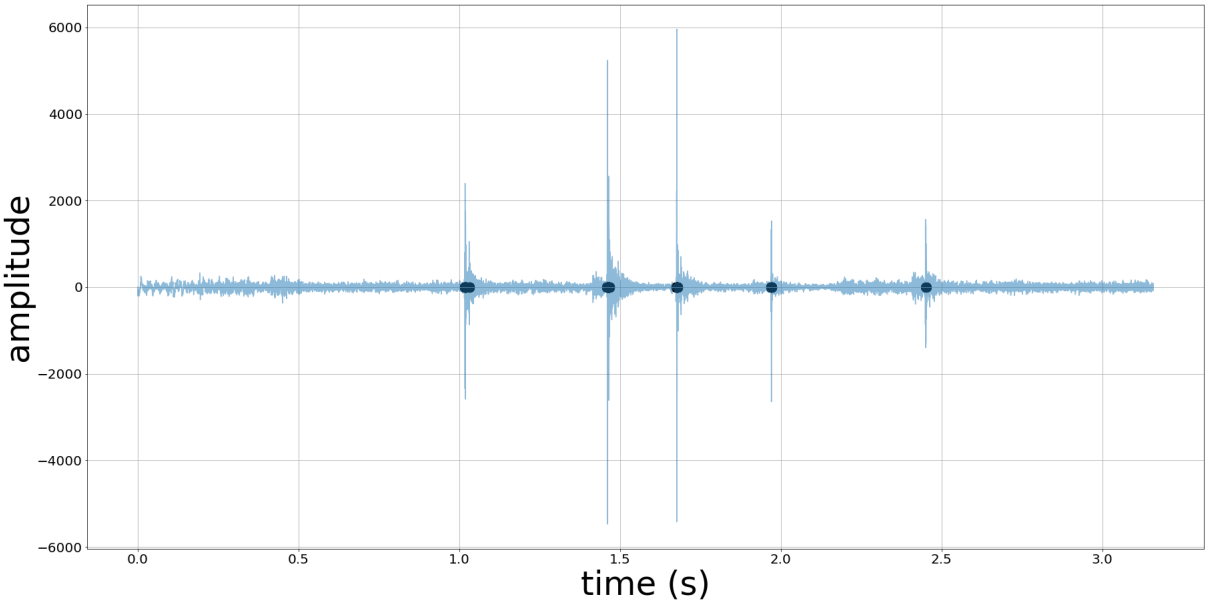


Рис. 25: Несколько хлопков на фоне музыки

4.2 Преобразование Фурье

(Пакет scipy.fftpack)

Алгоритм заключается в последовательном применении преобразования Фурье к подотрезкам длины, равной длине шаблона(229). Для каждого подотрезка вычисляем норму его коэффициентов Фурье. Для момента хлопка характерен резкий скачек этой нормы.

Для определения момента хлопка фиксируем порог. Если норма на данном отрезке больше, чем произведение порога на среднее арифметическое норм предыдущих десяти отрезков, фиксируем хлопок.

Достоинство этого алгоритма в том, что он не использует шаблон, быстро работает, определяет только начальный момент хлопка (интервалы - ответы алгоритма уже чем при использовании dtw). Недостатки - не определяет тихие хлопки, определяет хлопок во время удара.

4.2.1 Результат работы алгоритма с использованием преобразования Фурье

Ниже представлены результаты работы для THRESHOLD = 5.

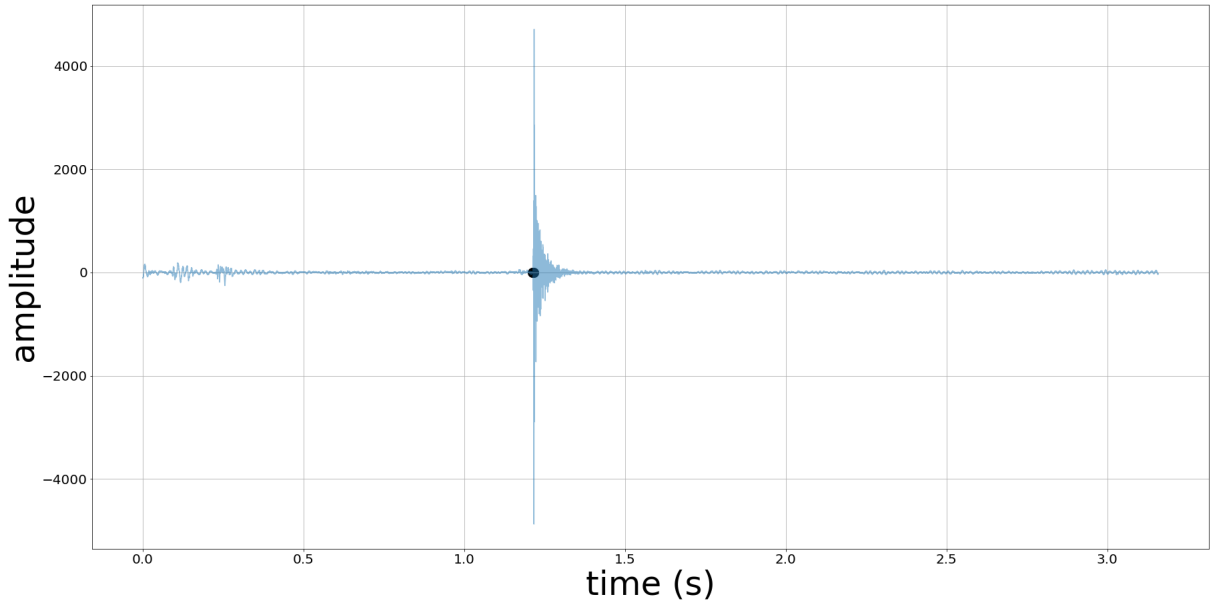


Рис. 26: Хлопок

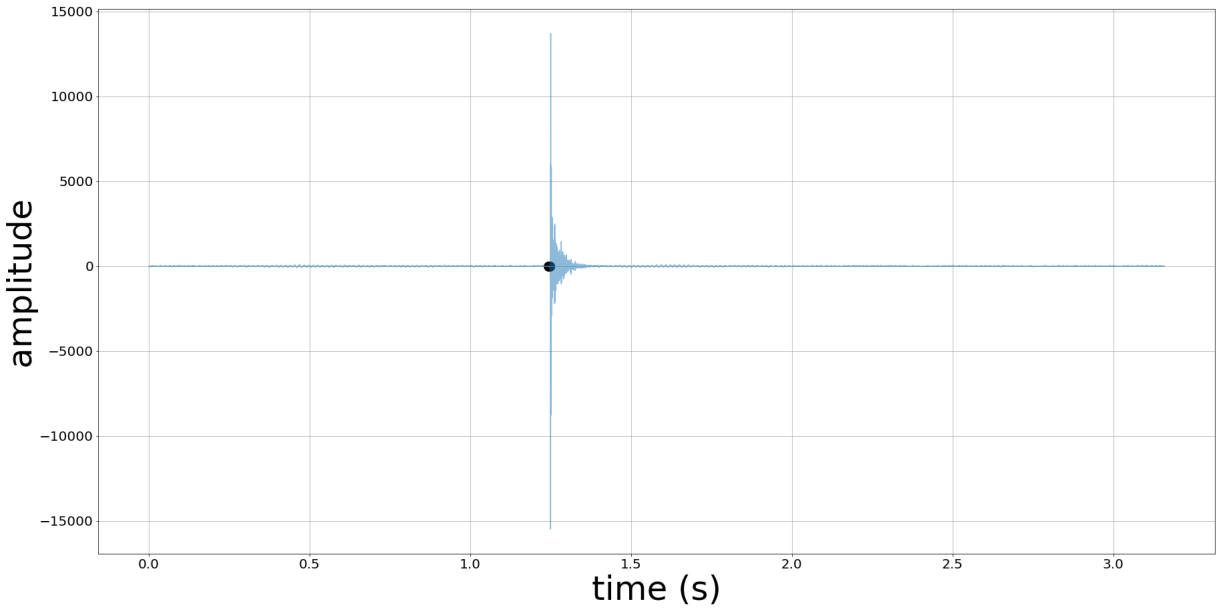


Рис. 27: Хлопок

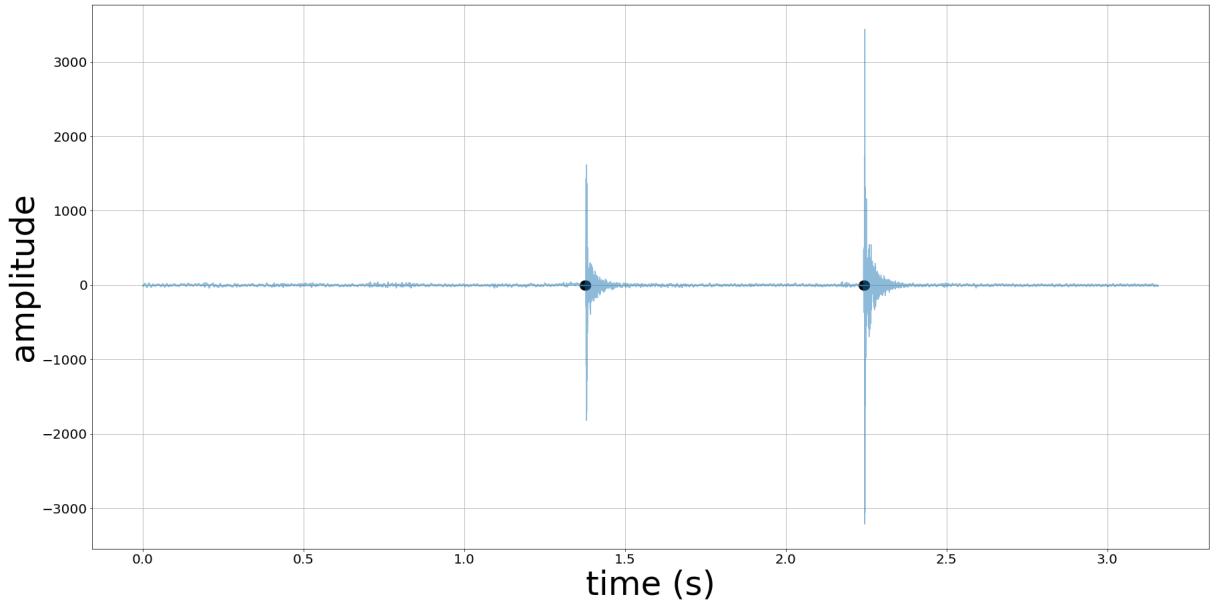


Рис. 28: 2 хлопка

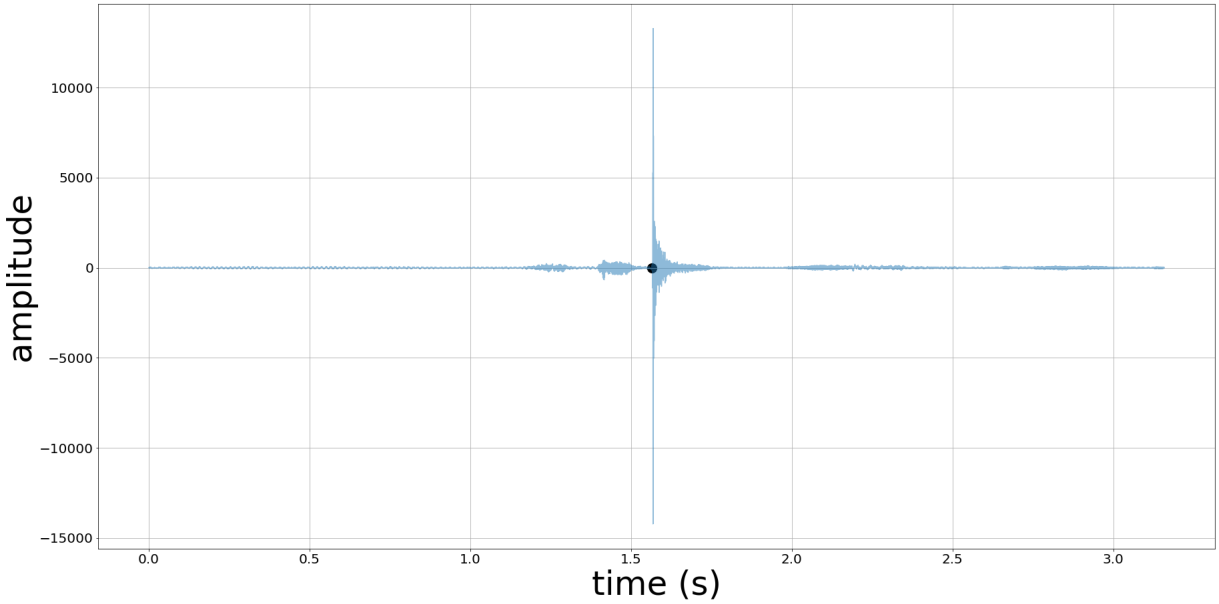


Рис. 29: Хлопок на фоне речи

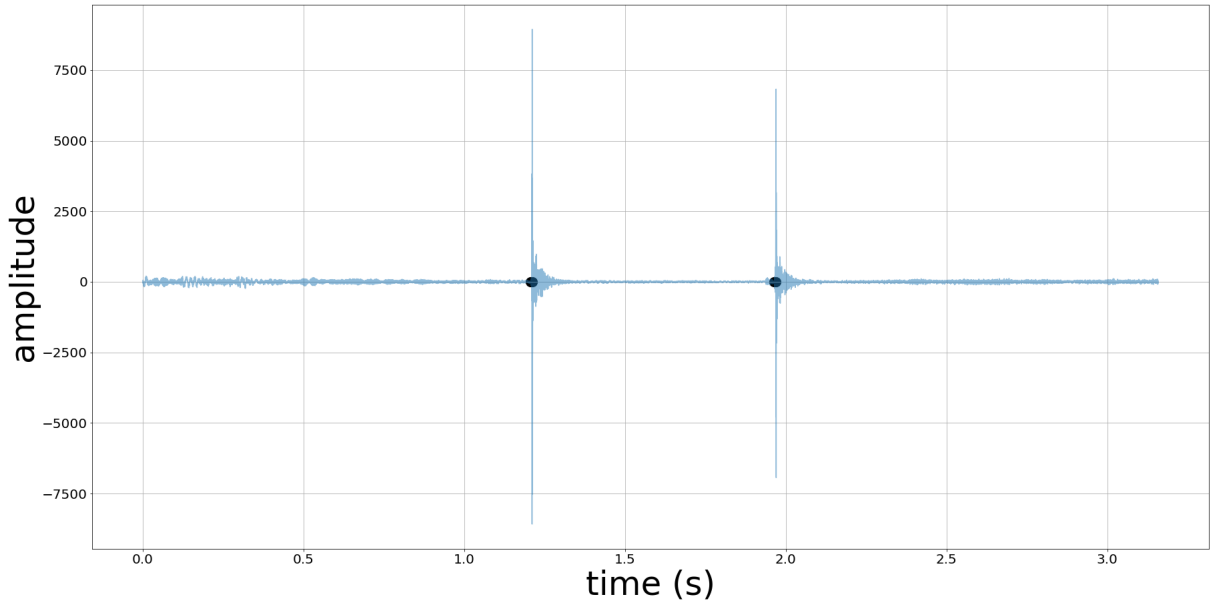


Рис. 30: 2 хлопка на фоне музыки

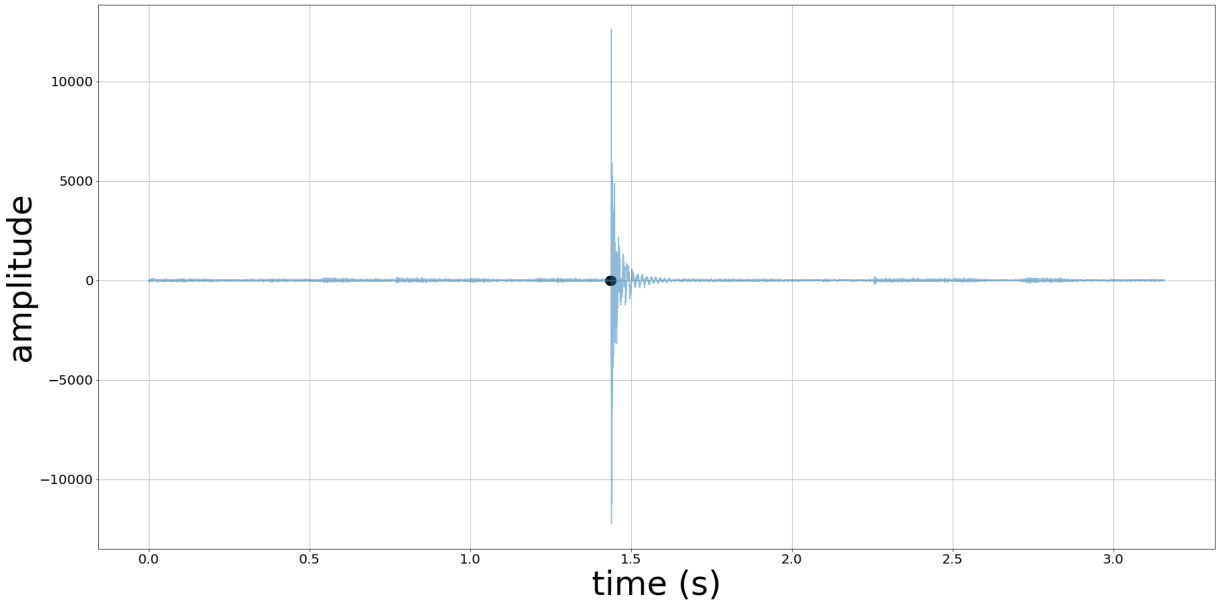


Рис. 31: Удар на фоне музыки

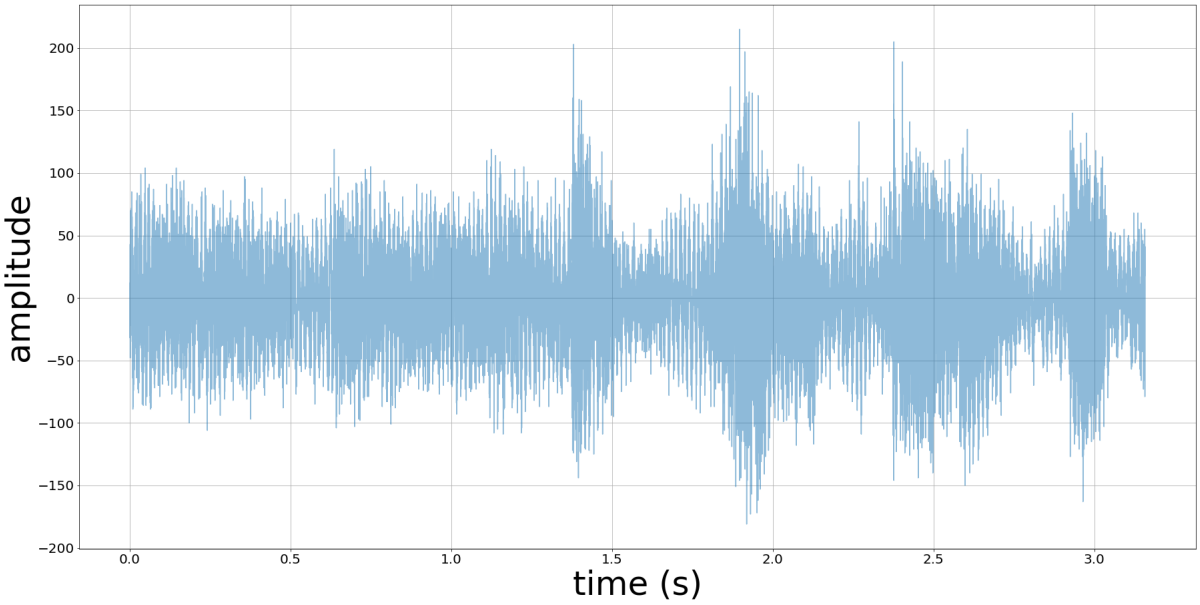


Рис. 32: Музыка

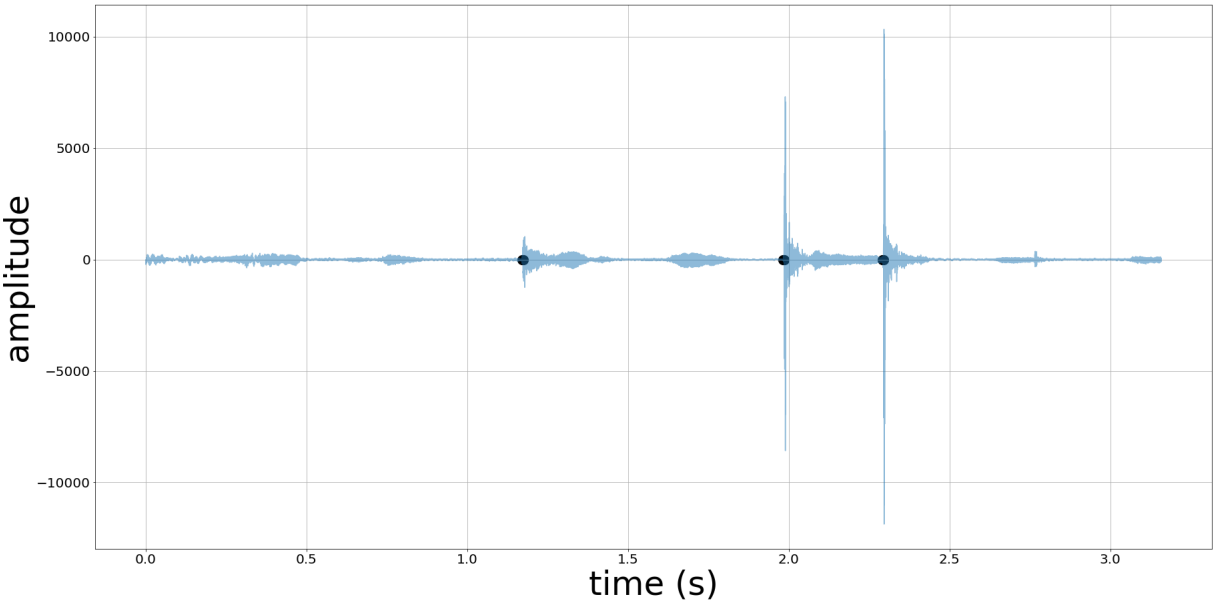


Рис. 33: Несколько хлопков на фоне музыки

4.3 Time series fingerprinting

(Пакет saxpy)

Не удалось подобрать параметры алгоритма (число букв алфавита, длина слова), обеспечивающие стабильность работы.

5 Архитектура приложения

Программа состоит из 3 частей: запись аудио, вычисления, вывод результатов в виде графика.

5.1 Запись аудио

Эта часть программы расположена в файле `record.py`. Запись аудиопотока осуществляется с помощью модуля `alsaaudio`. Среди глобальных переменных - частота (`RATE=44100`) и длительность записи в секундах(`RECORD_SECONDS=3`). В момент начала и окончания записи в консоль выводятся соответственно строки `"***recording***"` и `"written"`. Функция `recordAudio` сохраняет записанное аудио в файл `1.wav`.

5.2 Вычисления

Эта часть программы расположена в файле `util.py`. Состоит из двух функций.

Первая - `algorithm()` - вычисляет расстояния(`distances`) до шаблона для подотрезков скользящего окна. Выход функции - массив расстояний.

Вторая - `compare_with_median` - принимает на вход массив расстояний. Пусть `median` - медиана этого массива. Функция находит те подотрезки, расстояние до которых меньше, чем `THRESHOLD*median`.

`THRESHOLD` - порог, позволяющий регулировать количество положительных ответов алгоритма. При увеличении порога увеличивается количество верных положительных ответов(детектируются более тихие хлопки). Если в аудиозаписи нет посторонних звуков, то количество ложных положительных ответов при этом не увеличивается, а если есть - увеличивается. Выход функции - массив отсчетов, во время которых был зафиксирован шаблон.

5.3 Вывод результатов

Для вывода результатов используем модуль `matplotlib.pyplot`. График сохраняется в файл `1.png` или выводится на экран.