

Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование информационных систем
Системное программирование

Евсеев Олег Александрович

Разработка интерфейса управления
компьютером через нейроинтерфейс

Бакалаврская работа

Научный руководитель:
ст. пр. Ханов А.Р.

Рецензент:
к.т.н., доц. Воробьева А.А.

Санкт-Петербург
2019

SAINT-PETERSBURG STATE UNIVERSITY

Software and Administration of Information Systems
System Programming

Oleg Evseev

Designing an EEG-headset-assisted UI

Bachelor's Thesis

Scientific supervisor:
Senior Lecturer Artur Khanov

Reviewer:
Candidate of Engineering Sciences, Associate Professor Alisa Vorobyova

Saint-Petersburg
2019

Содержание

Введение	5
1. Постановка задачи	8
2. Обзор	9
2.1. Работа с электроэнцефалограммой	9
2.1.1. Потенциал P300 и его стимуляция	9
2.1.2. Организация рабочей сессии	11
2.1.3. Обнаружение P300 в электроэнцефалограмме	12
2.1.4. Улучшение качества распознавания P300	12
2.2. Существующие решения	13
2.3. ПО для работы с данными	13
2.3.1. Решения с открытым исходным кодом	14
2.3.2. OpenViBE	14
3. Анализ требований	16
3.1. Функциональные требования	16
3.2. Нефункциональные требования	17
4. Алгоритм распознавания P300	18
4.1. Описание данных	18
4.2. Предобработка сигнала	19
4.3. Сегментация сигнала	19
4.4. Опробованные классификаторы	20
5. Реализация прототипа системы	21
5.1. Архитектура	21
5.1.1. Общий вид	21
5.1.2. Модуль <code>realtime</code>	23
5.1.3. Модуль <code>app</code>	24
5.2. Пользовательский интерфейс	25
5.2.1. Ввод текста	25
5.2.2. Позиционирование курсора	26

5.3. Особенности реализации	27
6. Апробация	28
6.1. Методы оценки	28
6.2. Описание эксперимента	28
6.3. Сравнение опробованных методов	29
Заключение	31
Список литературы	32

Введение

Мозг человека — крайне сложная биологическая структура. Состоящий в среднем из 100 миллиардов нейронов, он является главным органом центральной нервной системы. Мозг отвечает за большинство процессов, происходящих в организме человека, а также выполняет когнитивную функцию, занимаясь восприятием информации, ее обработкой и принятием решений.

Понимание того, как работает мозг — это тема, представляющая интерес нейрофизиологов и психологов уже на протяжении многих столетий. Наибольшее применение находят результаты, достигнутые в области электроэнцефалографии: изменения электрических потенциалов, вызванные активностью нейронов мозга, способны дать огромное количество полезной информации о процессах, происходящих в мозге, и могут быть исследованы математическими средствами. Эти изменения можно зарегистрировать при помощи размещения электродов на поверхности кожи головы, а сформированный путем регистрации таких изменений временной ряд называется *электроэнцефалограммой*, или сокращенно *ЭЭГ*.

Наиболее часто извлекаемая из ЭЭГ информация — интенсивность волн определенной частоты в спектре сигнала, получаемая путем преобразования сигнала в спектр частот — так называемых ритмов головного мозга. Такая информация может быть использована для анализа состояний мозга (сон, покой, активная работа и т.п.), определения текущей испытываемой эмоции и других задач, касающихся исследования общего состояния мозга. Для некоторых прикладных задач более применимым оказывается изучение сигнала на наличие *потенциалов, связанных с событиями (event related potentials, ERP)* — резких изменений в ЭЭГ, возникающих в ответ на какой-либо аудиовизуальный либо физиологический стимул. В особенности полезным является потенциал *P300* — реакция мозга спустя примерно 300 мс на стимул, распознаваемый им как релевантный (например, на нахождение нужных слов в тексте при беглом его прочтении).

ЭЭГ, как и любой другой сигнал естественного происхождения, очень подвержен искажениям, возникающим в результате посторонних факто-

ров. В случае ЭЭГ такие помехи, например, вызываются электрическими импульсами, посылаемыми мышцам, а также самим движением мышц, если таковые находятся непосредственно вблизи электродов. Очень важную роль играет правильное позиционирование электродов и их хороший контакт с поверхностью головы. Дальнейшего улучшения качества можно добиться как аппаратными средствами, так и при помощи методов цифровой обработки сигналов.

С момента становления нейрофизиологии как области и до недавнего времени исследования в данной области были доступны лишь университетам и медучреждениям по причине дороговизны используемого оборудования. В настоящее же время в связи с появлением коммерчески доступных и портативных устройств — так называемых *нейрокомпьютерных интерфейсов (нейроинтерфейсов, НКИ)*, способных снимать ЭЭГ в домашних условиях, нейрофизиологические исследования стали доступны более широкому кругу энтузиастов. Такие нейроинтерфейсы обычно представляют собой набор из 5–15 электродов, закрепляемых на поверхности кожи головы и регистрирующих колебания напряжения. Портативность таких устройств позволяет их использовать в более широком ряде контекстов по сравнению с медицинским оборудованием — например, при исследовании активности мозга во время повседневной деятельности.

В рамках данной работы работа ведется с устройством Emotiv EPOC+. Данный НКИ был выбран из соображений того, что он является одним из наиболее популярных коммерчески доступных НКИ в настоящее время, а также обладает двумя референсными датчиками и аппаратным шумоподавлением.

Рост таких технологий, как виртуальная (VR) и дополненная (AR) реальность, а также распространенность ситуаций, требующих использования hands-free устройств: вождение, работа с людьми с ограниченными возможностями — привели к возросшему интересу к нейрофизиологии и нейроинтерфейсам. Так, сейчас существует достаточно большое количество работ, так или иначе касающихся регистрации вызванных потенциалов в ЭЭГ — в том числе и с применением современных методов машинного обучения, однако применение нейроинтерфейсов для управления компью-

тером пока ограничивается лишь рядом синтетических либо достаточно узких задач вроде управления конкретными приложениями [1; 2].

Учитывая возросшую распространенность нейроинтерфейсов и широкую применимость методов, предназначенных для распознавания ERP в сигнале, отсутствие открытого прикладного решения для управления компьютером, адаптированного под общие нужды и удобство конечного пользователя, кажется некоторым упущением. Реализация такого решения будет полезна хотя бы с той точки зрения, что для того, чтобы использовать какое-либо программное обеспечение с нейроинтерфейсом, его не нужно будет явно для этого адаптировать. При должном позиционировании такая система способна сделать нейроинтерфейсы еще более практически полезной технологией.

1 Постановка задачи

Целью данной работы является реализация прикладного решения, позволяющего осуществлять управление компьютером с использованием нейроинтерфейса Emotiv EPOC+.

Для достижения вышеуказанной цели были поставлены следующие задачи.

- Произвести анализ требований к решению.
- Реализовать алгоритм, позволяющий извлекать из ЭЭГ необходимую для работы с решением информацию.
- Спроектировать и реализовать прототип решения, способный осуществлять базовое взаимодействие с операционной системой в реальном времени с использованием НКИ.
- Произвести апробацию решения.

2 Обзор

2.1 Работа с электроэнцефалограммой

Мозг человека неоднороден по своей структуре — разные доли мозга отвечают за различные функции. Именно по этой причине для снятия ЭЭГ используется несколько датчиков, расположенных в местах, соответствующих разным областям мозга.

В некоторых долях можно обнаружить периодические низкочастотные колебания — так называемые ритмы головного мозга. Интенсивность таких колебаний напрямую зависит от того, чем занят мозг в текущий момент. В работе [3] приведен краткий обзор характеристик частотного спектра ЭЭГ человека и обсуждается их связь с различными состояниями человеческого тела, такими как сон, активная деятельность, концентрация, стресс и другими. В работе [4] приведен пример использования методов частотной (позволяет выделить конкретные ритмы) и пространственной (позволяет локализовать сигнал, специфичный для какой-либо из долей) фильтрации для построения вектора признаков в задаче классификации текущего психического состояния.

Исследование спектра сигнала — не единственный способ получить полезную информацию из ЭЭГ. Зачастую более полезным оказывается исследование сигнала на наличие потенциалов, связанных с событиями — реакцией мозга на какой-либо внешний стимул. Подробный обзор наиболее популярных из таких потенциалов приведен в работе [5].

2.1.1 Потенциал P300 и его стимуляция

Для нашей задачи наиболее полезным оказывается потенциал P300, получивший свое название благодаря тому, что возникает спустя 300 миллисекунд после редкого и релевантного стимула (например, при визуальном обнаружении нужного объекта среди множества нерелевантных). Выбор P300 обусловлен двумя причинами:

- Во-первых, благодаря простоте его стимуляции достаточно легко организовать сессию записи данных таким образом, чтобы в ответ на

некоторый стимул Р300 гарантированно вырабатывался, а при отсутствии — не вырабатывался [1]. Это обеспечивает его универсальность, делая пригодным для использования в широком ряде задач, которые можно свести к задаче выбора нужного класса из нескольких;

- Во-вторых, благодаря своей широкой применимости существует достаточно много работ, основанных на обнаружении Р300 в сигнале. Конкретные подходы разнятся в зависимости от используемого оборудования, поставленных задач и состояния испытуемого [6], но общие идеи оказываются применимы и в других работах.

На графике 1 приведен усредненный вид потенциала Р300 и произведено сравнение с обычным сигналом без каких-либо предшествующих внешних стимулов.

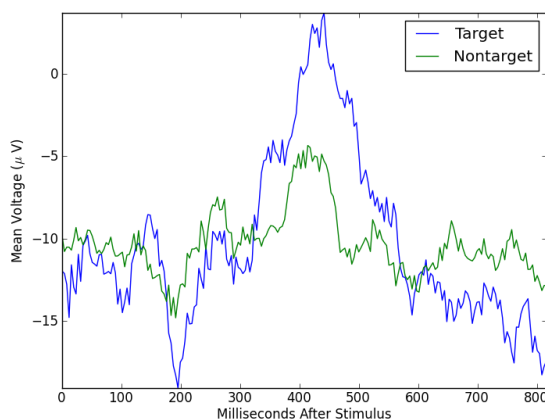


Рис. 1: Вид потенциала Р300 (источник — [7])

Стимуляция Р300 обычно осуществляется при помощи так называемой oddball-парадигмы [1] — испытуемому последовательно в случайном порядке показываются релевантные и нерелевантные стимулы, причем релевантные стимулы показываются реже. От испытуемого требуется мысленно отмечать появление релевантного стимула (например, подсчитать их), что и стимулирует возникновение данного потенциала. Oddball-парадигма легко обобщается на случай мультиклассовой классификации, когда от испытуемого требуется осуществить выбор между несколькими равнозначными классами.

Практически все решения, базирующиеся на распознавании Р300 в сигнале, состоят из четырех этапов.

1. Первичная предобработка сигнала (сглаживание, фильтрация, удаление артефактов).
2. Сегментация сигнала с учетом информации о стимулах.
3. Опционально — построение вектора признаков (например: усреднение измерений нескольких стимулов, результаты применения банка частотных фильтров, коэффициенты wavelet-преобразования [8], коэффициенты авторегрессии [9] и пр.).
4. Обучение и использование бинарного классификатора (SVM, LDA, нейронные сети и пр.) для определения наличия Р300 в сегменте.

2.1.2 Организация рабочей сессии

Среди параметров рабочей сессии, непосредственно влияющих на качество распознавания и время, затрачиваемое на одну итерацию распознавания, можно выделить следующие.

- *Количество повторений релевантного стимула* — если оно большое, это повышает вероятность корректного распознавания стимула, но требует большего количества времени на проведение одной итерации распознавания.
- *Расстояние между стимулами (target-to-target interval, TTI)* — чем оно меньше, тем менее выражены любые возникающие потенциалы, но сокращается время, затрачиваемое на распознавание.
- *Количество классов* — если классов мало, то стимул, соответствующий данному классу, возникает чаще, и может оказаться менее выражен.

Это — не единственные факторы, влияющие на качество процесса распознавания Р300, но они являются наиболее важны, потому что они, во-первых, не зависят от психического состояния испытуемого, а во-вторых —

могут быть при необходимости изменены. Более подробный обзор факторов, влияющих на качество данных ЭЭГ, приведен в статье [6].

2.1.3 Обнаружение P300 в электроэнцефалограмме

В работах [2; 10] приводится подробный обзор алгоритмов, применяемых на этапе классификации. Обсуждается применимость классических методов машинного обучения (LDA, SVM) к задаче классификации сегмента по признаку наличия P300, обсуждаются способы построения вектора признаков при помощи как простых средств обработки сигналов, так и специализированных алгоритмов (анализ главных компонент, генетические алгоритмы, адаптивная фильтрация и пр.)

Статья [11] предлагает использовать так называемый shape feature vector — вектор признаков на основе геометрической информации сегмента ЭЭГ.

2.1.4 Улучшение качества распознавания P300

В работе [12] обсуждается применение методов цифровой обработки сигналов (нормализация относительно среднего, фильтрация частот и пр.) для улучшения качества получаемого сигнала и устранения проблем вроде «плывущего среднего» (смещение значения среднего арифметического сигнала на временном окне относительно первоначального значения с течением времени).

В работе [13] обсуждается, как выглядят артефакты ЭЭГ, порожденные сердцебиением, сокращением мышц и движением глаз. Для нахождения аномалий предлагается использовать метод главных компонент (PCA) и метод независимых компонент (ICA) с последующей фильтрацией компонент с аномально высокой амплитудой.

Статья [14] демонстрирует, как для улучшения качества распознавания P300 применяется нормализация сигнала, его сглаживание при помощи фильтра и подавление немаксимумов (non-maximum suppression) для поиска пиков сигнала, являющимися потенциальными кандидатами на то, чтобы быть определены, как P300.

2.2 Существующие решения

В работе [1] предоставлен обзор наиболее популярных решений из области человеко-компьютерного взаимодействия, основной идеей которой является регистрация стимулированного искусственно потенциала P300. Большое внимание посвящено вопросу набора текста: исследуются способы конструирования интерфейса таким образом, чтобы максимизировать точность выбора нужной буквы и минимизировать время, потраченное на то, чтобы сделать выбор.

В статье [15] представлен простой пример использования P300 и классификатора на основе метода опорных векторов (SVM) для реализации простого детектора лжи.

Существует публичный набор данных [16], полученный с использованием медицинского оборудования и использованный для задачи построения системы ввода текста на базе потенциала P300. Есть ряд решений, посвященных вводу текста при помощи P300 и использующих данный набор данных для обучения и тестирования соответствующих классификаторов — [17] (на базе ансамбля SVM), [18] (на базе градиентного бустинга).

2.3 ПО для работы с данными

Компания Emotiv предоставляет два программных решения, позволяющих осуществлять работу с ЭЭГ.

- *EmotivPRO* [19] — проприетарная интегрированная среда для работы с данными ЭЭГ. Позволяет осуществлять запись, просмотр и сегментацию и разметку сигнала, а также осуществлять анализ ритмов головного мозга. Имеет возможность экспортирования данных в формате CSV.
- *Emotiv Community SDK* [20] — библиотека, позволяющая использовать возможности, аналогичные таковым в EmotivPRO, в собственных приложениях. Требуется наличие лицензии на EmotivPRO для работы с «сырыми» данными ЭЭГ. Поддержка прекращена с конца 2018

года, имеющимся пользователям предлагается пользоваться новым облачным сервисом *Emotiv Cortex*.

2.3.1 Решения с открытым исходным кодом

В силу лицензионных ограничений, налагаемых Emotiv на использование собственных решений для работы с Emotiv EPOC+ в других приложениях и сложности разработки альтернативного решения «с нуля», потребовалось произвести поиск альтернативных решений с открытым исходным кодом и более свободной лицензией.

В результате поиска имеющихся решений в открытых репозиториях (GitHub, GitLab, BitBucket) были найдены следующие библиотеки с заявленной поддержкой Emotiv EPOC+.

- *EmoKit* (<https://github.com/openyou/emokit>, 418 stars) — одно из первых решений. Обладает рядом минусов: не обновляется с 2017 года, нет поддержки устройств Emotiv EPOC+, выпущенных позже 2017 года.
- *CyKit* (<https://github.com/CymatiCorp/CyKit>, 38 stars) — идейный последователь EmoKit. Поддерживает работу со всеми устройствами Emotiv, существующими на сегодняшний день. По причине зависимости от `ruwinusb` данная библиотека работает только на ОС Windows, однако это компенсируется тем, что существует возможность использовать библиотеку как сервер, способный осуществлять передачу данных в реальном времени при помощи протокола WebSocket.

В конечном итоге для снятия данных была выбрана библиотека CyKit — как единственное найденное решение, способное работать с современными моделями Emotiv EPOC+.

2.3.2 OpenViBE

Также стоит упомянуть об OpenViBE [21] — инструментарии для работы с ЭЭГ, разработанным французским Национальным институтом исследований в информатике и автоматике (INRIA). OpenViBE предназначен для

работы с данными ЭЭГ в реальном времени и представляет набор инструментов для предобработки сигнала, его визуализации и дальнейшей работы с ним. Для разработки сценариев OpenViBE предоставляет специальный графический язык, что делает его крайне удобным для прототипирования.

Один из компонентов OpenViBE — Acquisition Server — предоставляет возможность получения данных ЭЭГ из различных источников: от конкретных моделей нейроинтерфейсов до приема произвольных числовых данных через Telnet. Получение данных от Emotiv EPOC+ не поддерживается OpenViBE напрямую, но благодаря поддержке WebSocket имеется возможность использовать OpenViBE совместно с CyKit.

3 Анализ требований

В данном разделе описываются требования, выдвинутые к прототипу. Они разделены на две группы — функциональные и нефункциональные.

Функциональные требования определяют минимальную функциональность, которую должен реализовывать прототип, чтобы он нес практическую значимость.

Нефункциональные требования призваны ограничить количество побочных технических задач, связанных с переносимостью и универсальностью решения, и позволяют сфокусироваться на основных задачах, поставленных для достижения цели настоящей работы.

3.1 Функциональные требования

Взаимодействие с ОС. Решение должно позволять осуществлять два вида взаимодействия с ОС: ввод текста и позиционирование курсора. Взаимодействие должно быть организовано в виде цикла последовательных действий пользовательского ввода.

Любые вспомогательные элементы пользовательского интерфейса, помогающие осуществлять взаимодействие (например, осуществляющий генерацию визуальных стимулов), не должны мешать восприятию текущего содержимого на экране и не должны быть доступны для взаимодействия напрямую.

Обработка сигнала в режиме реального времени. В силу того, что любые внешние стимулы (в том числе и визуальные) немедленно находят свое отражение в ЭЭГ, работа с сигналом должна быть организована в режиме реального времени.

Это значит, что в любой конкретный момент мы должны иметь возможность получить доступ к актуальным измерениям. Так как потенциалы, связанные с событиями, возникают спустя промежуток от нескольких десятков от нескольких сотен миллисекунд, задержка в 50 мс уже будет представлять проблемы для их корректного распознавания.

3.2 Нефункциональные требования

Сигнализация об ошибках. При возникновении каких-либо проблем с оборудованием, а также иных проблем, препятствующих дальнейшей работе, следует немедленно оповестить об этом пользователя и приостановить сессию до тех пор, пока проблема не будет решена.

Точность. В силу того, что ЭЭГ — сигнал естественного происхождения, его использование для управления компьютером сопряжено с определенными трудностями. Для того, чтобы решение было практически применимым, требуется минимизировать количество случаев, когда система неправильно распознает, какое из действий было выбрано пользователем.

Простота использования. Решение должно требовать от пользователя минимальных усилий по развертыванию и запуску. Единственное, что должно требоваться от пользователя для обеспечения корректного функционирования решения — указать адрес сервера СуKit и записать данные тренировочной сессии.

Целевая платформа. В качестве целевой платформы для разработки выбрана OS Windows по причине того, что она является наиболее популярной среди ОС потребительского сегмента [22].

Выбор конкретной платформы позволит сократить время, затрачиваемое на реализацию трансляции действий пользователя в конкретные системные вызовы и даст возможность сфокусироваться на более содержательных (и переносимых) задачах. По сравнению с другими ОС, выбор Windows как целевой платформы позволит увеличить охват потенциальной пользовательской базы решения.

Использование потенциала P300. В основе алгоритма, извлекающего из ЭЭГ информацию, необходимую для осуществления пользовательского ввода, должно лежать обнаружение потенциала P300. Причины данного выбора описаны в главе 2.1.1.

4 Алгоритм распознавания Р300

В данной главе описывается алгоритм распознавания Р300 в сигнале, состоящий из последовательного применения фаз первичной предобработки, сегментации и классификации.

4.1 Описание данных

Нейроинтерфейс Emotiv EPOC+ обладает 16 датчиками, из которых два (Р3 и Р4) являются референсными и предназначены для коррекции сигнала. Расположение датчиков приведено на рисунке 2.

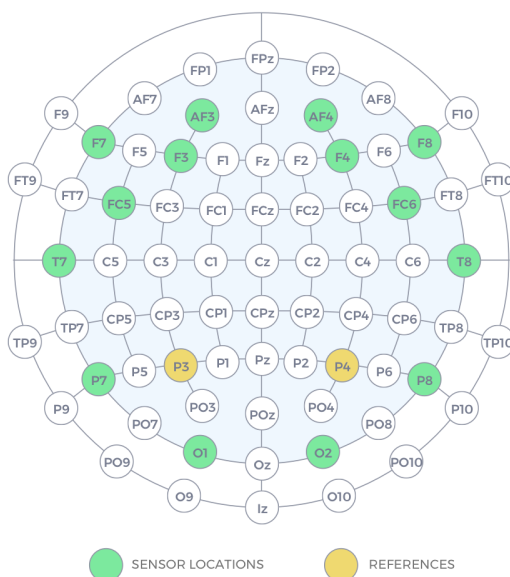


Рис. 2: Расположение датчиков Emotiv EPOC+ (из руководства [23])

Данные с электродов снимаются с внутренней частотой дискретизации 2048 Гц. Нейроинтерфейсом производится аппаратное удаление помех от сетей переменного тока при помощи двух режекторных фильтров с частотами 50 Гц и 60 Гц соответственно. Нормализация сигнала относительно значения референсных датчиков также выполняется аппаратно. После аппаратной фильтрации производится понижение частоты дискретизации до заданной частоты (128 Гц или 256 Гц в зависимости от выбранного режима работы устройства). На кодирование одного канала данных выделяется 16 бит, наименьшему значащему биту соответствует 0.13 микровольт.

Результирующие данные со всех 14 каналов дополняются счетчиком текущего измерения, обнуляемым каждые 128 (или 256, в зависимости от выбранного режима) измерений.

4.2 Предобработка сигнала

Работа над данными, снятыми с нейроинтерфейса, требует предварительной обработки. Для этого были выполнены следующие шаги:

- Применение к сигналу полосового фильтра с диапазоном частот 0.4–30 Гц;
- Подвыборка каждого второго измерения (64 Гц);
- Удаление артефактов при помощи ICA.

Полосовой фильтр применяется в связи с тем, что P300 — низкочастотный всплеск, а удаление частот меньше 0.4 Гц способствует исправлению проблемы *baseline drift* (плывущее среднее).

Подвыборка берется с целью удаления избыточной информации: по теореме Найквиста-Шеннона сигнал с частотой дискретизации 64 Гц может нести в себе информацию о частотах до 32 Гц. Так как до этого был применен полосовой фильтр и отсечены все частоты выше 30 Гц, можно закодировать полученные данные с частотой дискретизации 64 Гц без особых потерь качества.

Так как артефакты обычно не коррелируют с сигналом, при помощи ICA разделяем сигнал на независимые компоненты — артефакты и смысловую часть. Артефакты имеют большую амплитуду, чем сигнал, поэтому отсортируем компоненты по максимальной амплитуде и отбросим несколько первых компонент. Однако, есть риск случайно отсечь потенциал P300 — убирать слишком много компонент нельзя.

4.3 Сегментация сигнала

Процесс сегментации выглядит следующим образом:

- из ЭЭГ вырезается одна секунда данных после полученного стимула;
- отклики на стимулы одного класса объединяются в один вектор признаков путем усреднения сигнала (относительно каждого канала);
- данные всех каналов объединяются в один вектор признаков;
- классификатору передаются вектора признаков без каких-либо дополнительных изменений.

Такие параметры сессии, как ТП и количество повторений сегмента могут быть настроены перед проведением эксперимента.

4.4 Опробованные классификаторы

Для решения задачи классификации сигнала были опробованы популярные классификаторы, встречающиеся в решении подобных задач — SVM, LDA и полносвязные нейронные сети.

SVM. Классификатор SVM (Support Vector Machine, метод опорных векторов) был выбран из-за того, что он хорошо работает на данных с высокой размерностью. Однако для того, чтобы показать хорошую точность в задаче бинарной классификации, часто приходится подбирать параметры классификатора (в частности, тип ядра) либо заниматься задачей извлечения признаков из необработанного вектора.

LDA. LDA (Linear Discriminant Analysis, линейный дискриминантный анализ) опробован в связи с тем, что он с высокой точностью решает задачу поиска критерия, по которому два различных класса легко отличимы друг от друга. В данном случае это класс фрагментов сигнала, в которых присутствует P300, и класс фрагментов сигнала, в которых P300 не встречается. Зачастую такие классы можно отличить визуально.

Полносвязные нейронные сети. Также были использованы полносвязные нейронные сети, так как они хорошо справляются с задачей обучения на скрытых признаках.

5 Реализация прототипа системы

Несмотря на то, что средствами описанных в главе 2.3 решений возможно организовать запись данных ЭЭГ в файл для дальнейшей работы с ними в автономном режиме, для проведения полноценных экспериментов с пользовательским вводом в реальном времени перед реализацией алгоритма распознавания P300 требуется реализовать часть решения, ответственную за получение и обработку данных в реальном времени, а также пользовательский интерфейс.

5.1 Архитектура

С учетом требований, описанных в главе 3, была разработана следующая архитектура решения, позволяющего осуществлять ввод текста и позиционирование курсора путем работы с данными ЭЭГ в реальном времени.

5.1.1 Общий вид

Общая архитектура решения (в виде диаграммы компонентов UML) представлена на рисунке 3. На ней показано, что решение состоит из трех основных модулей.

- Модуль `realtime` отвечает за получение данных, их первичную обработку и сегментацию, а также оповещении пользовательского интерфейса о необходимости генерации стимула.
- Модуль `app` отвечает за основную логику приложения (подключение к серверу CyKit, запуск модуля `realtime`, реализация пользовательского интерфейса).
- Модуль `model` предоставляет алгоритм распознавания P300, описанный в главе 4.

Для получения данных ЭЭГ с нейроинтерфейса используется библиотека CyKit. Она же предоставляет возможность передачи данных через протокол TSP для дальнейшей обработки.

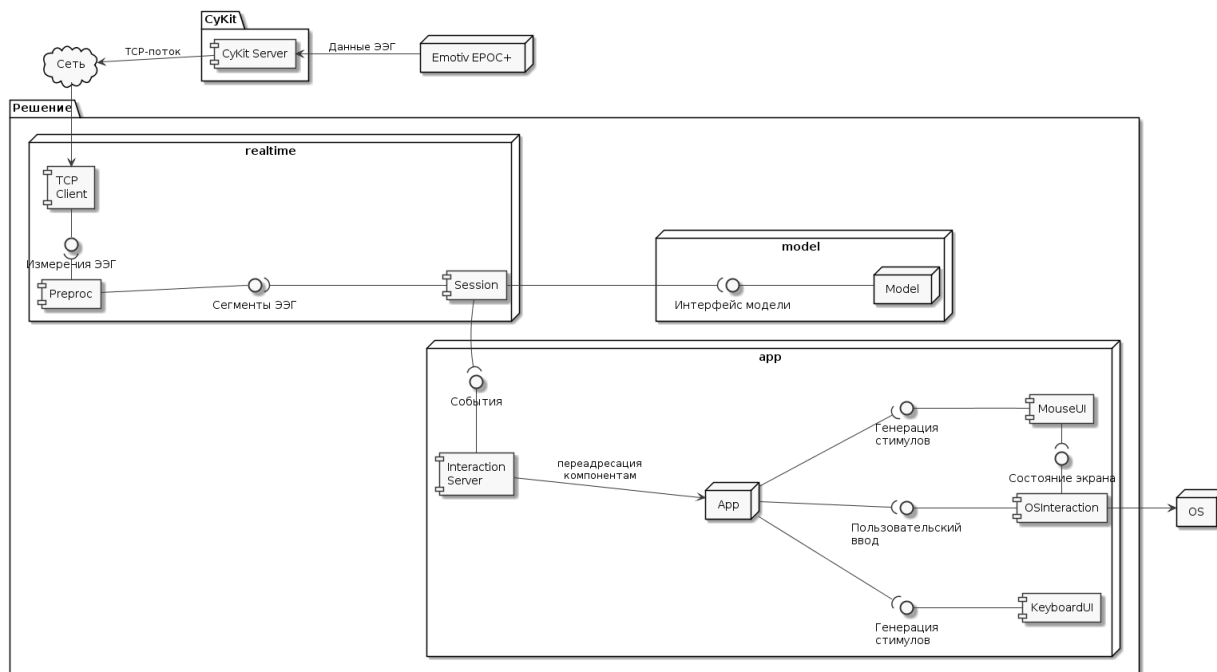


Рис. 3: Диаграмма компонентов решения

Разделение логики приложения на отдельные части обусловлено тем, что пользовательский ввод и генерация визуальных стимулов являются некоторыми вызываемыми событиями, что позволяет нам использовать для реализации пользовательского интерфейса и служебной логики любой событийно-ориентированный фреймворк (в нашем случае Qt). Напротив, обработка сигнала требует непрерывного взаимодействия с ним.

Изоляция обработки сигнала позволяет избежать ряда проблем: если бы все приложение целиком было реализовано в виде рабочего цикла, в реальном времени обрабатывающего данные ЭЭГ, это вызвало бы трудности при разработке пользовательского интерфейса. Наоборот, если бы мы использовали событийно-ориентированный фреймворк и реализовали получение измерений как отдельные события, то это бы, во-первых, вызвало бы значительную нагрузку на очередь событий (как минимум 128 событий в секунду при частоте дискретизации устройства 128 Гц), а во-вторых — увеличило бы задержку между регистрацией измерения и его обработкой.

Рассмотрим модули `app` и `realtime` более подробно.

5.1.2 Модуль `realtime`

Основным классом модуля `realtime` является класс `Session`, организующий высокоуровневую логику взаимодействия с компьютером посредством:

- получения сегментов ЭЭГ — путем отправки асинхронного вызова `get_segment` классу `Preproc` и ожидания результата по требованию;
- взаимодействия с моделью;
- оповещения приложения о необходимости генерации визуального стимула — путем отправки родительскому процессу сообщения;
- оповещения приложения о выбранном действии — путем отправки родительскому процессу сообщения.

Класс `Session` также предоставляет метод `run`, являющийся точкой входа дочернего процесса, порождаемого приложением при запуске сессии.

Класс `Preproc` инкапсулирует логику первичной предобработки данных и сегментации сигнала. Он предоставляет интерфейс на базе асинхронных вызовов, позволяющий обрабатывать получаемые от устройства измерения, выполнять их предобработку и дополнять запросы на сегменты новыми данными. Возвращаемые классом сегменты представляют собой `awaitable` значения, которые могут быть использованы, как только получено необходимое количество данных.

Класс `TCPClient` занимается получением и обработкой TCP-потока, получаемого от `CyKit`. Он выполняется в отдельном потоке и реализует паттерн «наблюдатель», реализуя функциональность оповещения подписанных на него классов о том, что доступны новые данные.

Типичный сценарий получения сегмента показан на рисунке 4. На диаграмме можно увидеть, как объект класса `Session` посылает TCP-клиенту сообщение о необходимости установки соединения с `CyKit`. Далее `Session` подписывает `Preproc` на обновления сервера — TCP-клиент при получении очередного сегмента оповещает об этом объект класса `Preproc`. Непосредственно получение сегмента от `Preproc` производится при помощи вызова

асинхронного метода `get_segment`, результат которого возвращается в тот момент, когда собрано необходимое количество данных.

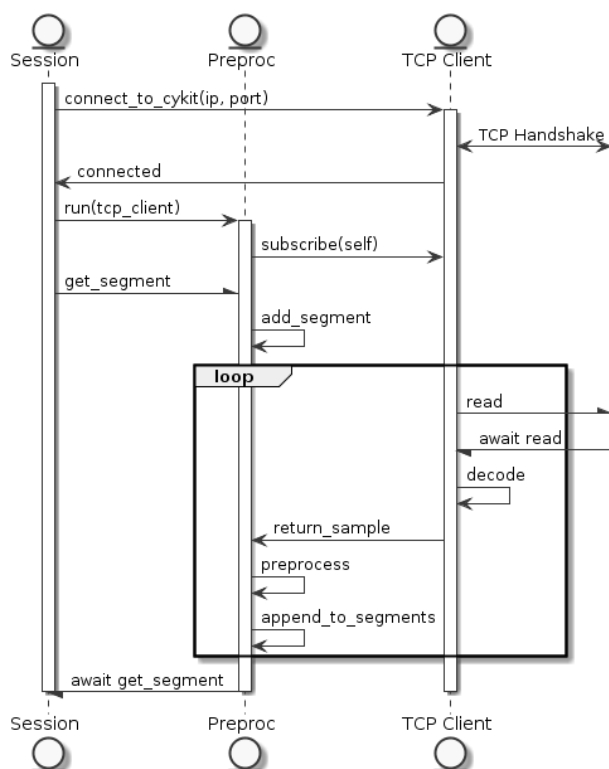


Рис. 4: Сценарий получения сегмента

5.1.3 Модуль `app`

Вся остальная логика приложения, не требующая взаимодействия с данными в реальном времени, помещена в модуль `App`:

- служебный пользовательский интерфейс (запустить/остановить сессию, настроить параметры подключения к серверу `CyKit`);
- элементы интерфейса, отвечающие за отображение визуальных стимулов (классы `MouseUI`, `KeyboardUI`);
- класс, инкапсулирующий логику взаимодействия с ОС (`OSInteraction`).

Реализация межпроцессного взаимодействия осуществлена посредством протокола TCP. Для получения сообщений от дочернего процесса реализо-

ван класс `InteractionServer`, способный принимать от дочернего процесса сообщения о необходимости генерации стимула и о выбранном действии.

5.2 Пользовательский интерфейс

Приложение способно работать в двух режимах.

- Рабочий — алгоритм распознавания Р300 находит, какие стимулы являются релевантными, и в зависимости от этого транслирует действия пользователя в системные вызовы.
- Тренировочный — от пользователя требуется выбрать заранее указанный класс. Соответствующие данные ЭЭГ собираются и используются для обучения классификатора.

Для контроля текущего выбранного режима, а также настройки параметров подключения к серверу `SuKit` используется меню приложения, доступное через значок в системном трее.

5.2.1 Ввод текста

В рамках данного решения для ввода текста используется следующий подход.

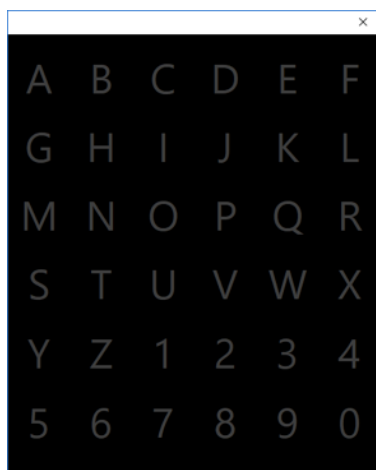


Рис. 5: Пользовательский интерфейс для ввода текста

- Символы, доступные для ввода, организованы в матрицу 6x6 (рисунок 5).
- В качестве стимулов выступает подсветка столбца или строки (всего 12 классов). От пользователя требуется сфокусироваться на нужном символе и мысленно отмечать, когда подсвечиваются содержащие его столбец или строка.
- Путем применения алгоритма распознавания Р300 находятся два релевантных класса, соответствующие релевантным столбцу и строке.

5.2.2 Позиционирование курсора

Изначально был разработан вариант позволяющий осуществлять перемещение курсора на фиксированное количество пикселей в одном из четырех направлений, а также осуществлять эмуляцию щелчка левой либо правой кнопкой мыши (рисунок 6).

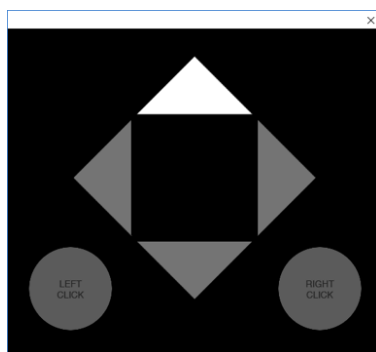


Рис. 6: UI для передвижения курсора, старый вариант

Однако, такой вариант имел очевидные недостатки — на перемещение курсора между отдаленными частями экрана приходилось затрачивать слишком много действий.

Более удобным оказался следующий подход, когда пространство делится на четверти и пользователю предлагается выбрать между ними. После выбора нужной части экрана процесс повторяется рекурсивно до тех пор, пока курсор не будет спозиционирован точно. Данный подход позволяет требует меньшее количество действий: например, на дисплеях с разрешением FullHD требуется осуществить всего 9 итераций.

Для того, чтобы гарантировать стимуляцию Р300, вместо того, чтобы подсвечивать нужную часть рабочего стола непосредственно, активная четверть отображается в увеличенном виде в отдельном окне (рисунок 7).

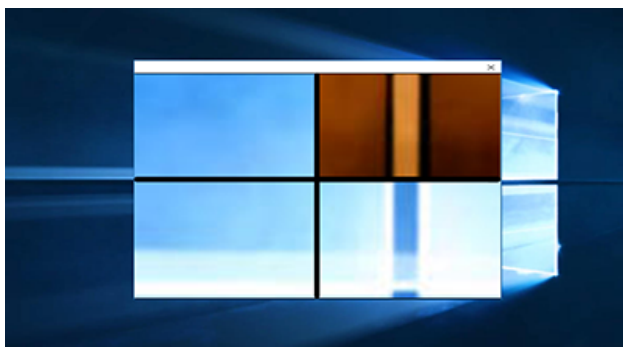


Рис. 7: Пользовательский интерфейс для передвижения курсора, новый вариант

5.3 Особенности реализации

Для разработки решения использовался язык Python версии 3.7. Выбор в сторону Python сделан благодаря простоте его применения для задач обработки данных при помощи математических средств.

Для реализации применялись следующие вспомогательные библиотеки:

- `scipy` (в частности `scipy.signal` и `scipy.fft`) — предобработка и сегментация сигнала;
- `scikit-learn` — реализации классификаторов, используемых для распознавания Р300;
- Qt for Python (PySide2) — пользовательский интерфейс;
- `pywin32` (обертка над WinAPI для Python) — взаимодействие с системой.

6 Апробация

Для того, чтобы проверить, насколько реализованное решение соответствует требованиям, а также для того, чтобы среди аналогичных вариантов реализации алгоритма распознавания Р300 выбрать тот, который дает наилучший результат на реальных данных, потребовалось произвести ряд экспериментов.

6.1 Методы оценки

Для оценки качества алгоритма рассмотрим сценарий ввода текста из главы 5.2.1 — перед нами стоит задача найти среди 12 классов два релевантных. Попробуем ввести вещественнозначную метрику, позволяющую нам сравнить варианты, полученные на этапе разработки алгоритма распознавания Р300.

Обозначим количество испытаний в эксперименте как `total`. Пусть среди них у нас имеется `correct` испытаний, в результате которых мы получили результат, совпадающий с загаданным. Тогда мы можем ввести метрику `accuracy` точности распознавания:

$$\text{accuracy} = \frac{\text{correct}}{\text{total}}.$$

Результаты экспериментов усредняются по всем испытуемым.

Так как разрабатываемое средство предполагает использование в реальном времени, важным параметром является время, которое требуется для одной итерации распознавания. Оно определяется однозначно параметрами сессии — для n классов и k повторений стимулов оно составляет:

$$T = TPI \cdot n \cdot k.$$

6.2 Описание эксперимента

Эксперимент по оценке качества распознавания Р300 состоит из двух фаз:

- Тренировочная сессия (100 испытаний) — от пользователя требуется выбрать конкретный указанный класс. Данные тренировочной сессии используются для обучения классификаторов (80 испытаний) и поиска оптимальных значений гиперпараметров (20 испытаний).
- Непосредственно эксперимент (100 испытаний) — от пользователя требуется ввести предложенный текст с использованием решения. Введенный текст сравнивается с модельным при помощи метрики accuracy.

Данные для экспериментов были записаны с привлечением двух испытуемых. Оба испытуемых были физически здоровы, не имели психических отклонений и на момент записи данных не находились в состояниях, отрицательно сказывающихся на концентрации (недосып, переутомление и другие подобные состояния).

6.3 Сравнение опробованных методов

Для качественного сравнения выбранных методов (SVM, LDA, нейронная сеть) варьировались следующие параметры.

- Метод удаления артефактов (PCA, ICA).
- Гиперпараметры — был произведен grid search (перебор) гиперпараметров. Выбирались количество отсекаемых компонент ICA, частота среза полосового фильтра, длина сегмента при сегментации и пр.
- Параметры сессии — их изменение предполагает получение новых данных. В отличие от изменения классификатора или способов предобработки, не требующих повторной записи данных, при изменении количества повторений стимула, количества классов и интервала между стимулами требуется провести несколько различных экспериментов. Однако, имеется возможность симитировать уменьшение количества повторений стимула, записав данные с большим количеством повторений и отбросив некоторое количество сегментов.

Изначально алгоритм был протестирован на данных [16] — удалось достичь ассурасу, равной 91.0%, при помощи следующей комбинации:

- классификатор — LDA;
- метод удаления артефактов — ICA;
- параметры сессии: ТТИ = 100 мс, 15 повторений стимула (18 с на одну итерацию ввода). Уменьшение количества повторений стимула до 5 уменьшает время одной итерации ввода до 6 с, но вызывает понижение ассурасу до значения 56.5%.

Аналогичная конфигурация, но с использованием нейронной сети вместо LDA дала результат 87.5% (для сравнения — при замене бинарного классификатора на случайный, возвращающий положительный ответ с вероятностью 0.5, математическое ожидание ассурасу для всего эксперимента получается равным 2.8%).

Также были опробованы подходы, использующие SVM, но они демонстрировали неприемлемые результаты на необработанных данных (менее 30%). Улучшение качества для них требует дополнительного шага извлечения признаков. Попытки использования PCA вместо ICA на этапе удаления артефактов также привели лишь к ухудшению качества.

На собственных же данных, записанных в результате эксперимента, аналогичная конфигурация дала лишь 57.5% ассурасу — для записи набора данных [16] применялось медицинское оборудование с большим количеством датчиков, что, вероятно, и является причиной того, что для Emotiv EROC+ были получены более плохие результаты.

Заключение

В процессе выполнения данной работы были получены следующие результаты.

- Осуществлен поиск решений, позволяющих осуществлять получение данных ЭЭГ с Emotiv EPOC+. Для работы с устройством была выбрана библиотека CyKit.
- Сформулирован ряд функциональных и нефункциональных требований к системе.
- Реализован алгоритм обнаружения P300 в ЭЭГ, состоящий из фаз предобработки, фильтрации и применения бинарного классификатора. Опробован ряд классификаторов (SVM, LDA, полносвязные нейронные сети) и методов предварительной обработки данных.
- С учетом требований разработан прототип решения, позволяющего осуществлять ввод текста и позиционирование курсора на компьютере под управлением ОС Windows. Для этого была разработана архитектура, в основу которой легло отделение логики реального времени от остальной логики. Был разработан пользовательский интерфейс для отображения визуальных стимулов. Реализация осуществлена на языке Python с использованием фреймворка Qt.
- Проведена серия экспериментов с привлечением реальных испытуемых. На полученных в результате экспериментов данных произведено сравнение опробованных методов — наилучший результат (ассигасу = 57.5%) показал алгоритм, использующий LDA в качестве классификатора и ICA в качестве метода удаления артефактов.

Список литературы

1. *Haider, Ali, Fazel-Rezai, Reza.* Application of P300 Event-Related Potential in Brain-Computer Interface // Event-Related Potentials and Evoked Potentials / под ред. Phakkharawat Sittiprapaporn. — Rijeka : IntechOpen, 2017. — Гл. 2. — DOI: 10.5772/intechopen.69309. — URL: <https://doi.org/10.5772/intechopen.69309>.
2. A survey of signal processing algorithms in brain-computer interfaces based on electrical brain signals / Ali Bashashati [и др.] // Journal of Neural engineering. — 2007. — Т. 4, № 2. — R32.
3. *Kumar, J. Satheesh, Bhuvaneswari, P.* Analysis of Electroencephalography (EEG) Signals and Its Categorization—A Study // Procedia Engineering. — 2012. — Т. 38. — С. 2525—2536. — ISSN 1877-7058. — DOI: <https://doi.org/10.1016/j.proeng.2012.06.298>. — URL: <http://www.sciencedirect.com/science/article/pii/S1877705812022114> ; INTERNATIONAL CONFERENCE ON MODELLING OPTIMIZATION AND COMPUTING.
4. *Lotte, Fabien.* A Tutorial on EEG Signal Processing Techniques for Mental State Recognition in Brain-Computer Interfaces // Guide to Brain-Computer Music Interfacing / под ред. Eduardo Reck Miranda, Julien Castet. — Springer, 2014. — URL: <https://hal.inria.fr/hal-01055103>.
5. *Sur, S., Sinha, V. K.* Event-related potential: An overview // Industrial psychiatry journal. — 2009. — 18(1). — С. 70—73.
6. *Polich, John, Kok, Albert.* Cognitive and biological determinants of P300: an integrative review // Biological psychology. — 1995. — Т. 41, № 2. — С. 103—146.
7. P300 Waves for Single Subjects (дата последнего обращения — 29 мая 2019 г.) — URL: http://www.cs.colostate.edu/eeg/data/json/doc/tutorial/_build/html/p300_single_subject.html.

8. Wavelet analysis of oddball P300 / Tamer Demiralp [и др.] // International journal of psychophysiology. — 2001. — Т. 39, № 2/3. — С. 221—227.
9. *Turnip, Arjon, Hong, Keum-Shik*. Classifying mental activities from EEG-P300 signals using adaptive neural network // Int. J. Innov. Comp. Inf. Control. — 2012. — Т. 8, № 7. — С. 5839—5850.
10. *Lakshmi, M. Rajya, Prasad, T. V., Prakash, V. Chandra*. Survey on EEG signal processing methods // International Journal of Advanced Research in Computer Science and Software Engineering. — 2014. — Т. 4, № 1.
11. P300 detection based on EEG shape features / Montserrat Alvarado-González [и др.] // Computational and mathematical methods in medicine. — 2016. — Т. 2016.
12. *Al-Ani, Tarik, Trad, Dalila*. Signal Processing and Classification Approaches for Brain-Computer Interface // / под ред. Vernon S. Somerset. — InTech, 01.2010. — ISBN 978-953-7619-58-9. — DOI: 10.5772/7032.
13. *Roy, Vandana, Shukla, Shailja*. A Survey on Artifacts Detection Techniques for Electro- Encephalography (EEG) Signals // International Journal of Multimedia and Ubiquitous Engineering. — 2015. — Т. 10, № 3. — С. 425—442.
14. *Tsuda, Mineyuki, Lang, Yankun, Wu, Haiyuan*. Analysis and Identification of the EEG Signals from Visual Stimulation // Procedia Computer Science. — 2014. — Т. 35. — С. 1292—1299. — ISSN 1877-0509. — DOI: <https://doi.org/10.1016/j.procs.2014.08.229>. — URL: <http://www.sciencedirect.com/science/article/pii/S1877050914011946> ; Knowledge-Based and Intelligent Information & Engineering Systems 18th Annual Conference, KES-2014 Gdynia, Poland, September 2014 Proceedings.
15. An experiment of lie detection based EEG-P300 classified by SVM algorithm / Artha Simbolon [и др.] //. — 10.2015. — DOI: 10.1109/ICACOMIT.2015.7440177.

16. BCI Competition III. — URL: <http://www.bbci.de/competition/iii/>.
17. *Rakotomamonjy, Alain, Guigue, Vincent*. BCI competition III: dataset II-ensemble of SVMs for BCI P300 speller // IEEE transactions on biomedical engineering. — 2008. — Т. 55, № 3. — С. 1147—1154.
18. A boosting approach to P300 detection with application to brain-computer interfaces / Ulrich Hoffmann [и др.] // Conference Proceedings. 2nd International IEEE EMBS Conference on Neural Engineering, 2005. — IEEE. 2005. — С. 97—100.
19. EmotivPRO (дата последнего обращения — 29 мая 2019 г.) — URL: <https://www.emotiv.com/emotivpro/>.
20. Emotiv Community SDK (дата последнего обращения — 29 мая 2019 г.) — URL: <https://github.com/Emotiv/community-sdk>.
21. OpenViBE (дата последнего обращения — 29 мая 2019 г.) — URL: <http://openvibe.inria.fr/>.
22. Market Share Statistics for Internet Technologies (дата последнего обращения — 29 мая 2019 г.) — URL: <https://netmarketshare.com/operating-system-market-share.aspx>.
23. EPOC+ User Manual (дата последнего обращения — 29 мая 2019 г.) — URL: <https://emotiv.gitbook.io>.