

Министерство науки и высшего образования РФ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Космических и информационных технологий
институт

Вычислительная техника
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

О. В. Непомнящий

подпись

инициалы, фамилия

« ____ » _____ 20 ____ г.

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы**

Студенту Айзенберг Наталье Михайловне

фамилия, имя, отчество

Группа КИ15-06Б Направление (специальность) 09.03.01

номер

код

Информатика и вычислительная техника

наименование

Тема выпускной квалификационной работы Программная часть аппаратно-программного комплекса для распознавания жестов

Утверждена приказом по университету № _____ от _____

Руководитель ВКР О. В. Непомнящий, профессор, заведующий кафедрой

Вычислительная техника, канд.тех.наук

инициалы, фамилия, должность, учёное звание и место работы

Исходные данные для ВКР: Провести анализ проблем коммуникации людей с нарушением слуха, исследовать известные решения, полученные при создании программного обеспечения устройства для распознавания жестов. Разработать программное обеспечение для устройства распознавания жестов из тактильной азбуки русского жестового языка. Провести лабораторные испытания разработанного алгоритма и анализ полученных результатов.

Перечень разделов ВКР: Анализ существующих систем распознавания жестов, анализ алгоритмов обработки данных, разработка программного обеспечения аппаратно-программного комплекса для распознавания жестов, лабораторные испытания алгоритма и анализ полученных результатов.

Перечень графического материала: презентация в формате Power Point, раздаточный материал.

Руководитель ВКР

подпись

О. В. Непомнящий

инициалы, фамилия

Задание принял к исполнению

подпись

Н.М. Айзенберг

инициалы, фамилия

«___» _____ 2019

РЕФЕРАТ

Выпускная квалификационная работа по теме «Программная часть аппаратно-программного комплекса для распознавания жестов» содержит 52 страницы текстового документа, 1 таблицу, 22 иллюстрации, 2 формулы, 27 использованных источников, 2 приложения.

РАСПОЗНАВАНИЕ ЖЕСТОВ, ФИЛЬТРАЦИЯ ЦИФРОВОГО СИГНАЛА, МЕДИАННАЯ ФИЛЬТРАЦИЯ, САМОАДАПТИВНЫЕ АЛГОРИТМЫ, КЛАСТЕРИЗАЦИЯ, K-MEDOIDS, МОДЕЛИРОВАНИЕ, MATLAB, НЕЙРОННАЯ СЕТЬ

Проведен анализ систем распознавания жестов, рассмотрены существующие системы для распознавания жестов. На основе анализа были выделены основные требования к программному обеспечению аппаратно-программного комплекса для распознавания жестов. Проведено моделирование и анализ алгоритмов обработки данных – методов фильтрации и алгоритмов машинного обучения. По результатам проведенного анализа было принято решение использовать в аппаратно-программном комплексе медианный фильтр и алгоритм кластеризации методом k-медиан. Разработано программное обеспечение аппаратно-программного комплекса для распознавания жестов с использованием нейронной сети, способное распознавать жесты из дактильной азбуки русского жестового языка.

СОДЕРЖАНИЕ

Введение	4
1 Анализ систем распознавания жестов	5
1.1 Общая информация о жестовом языке	5
1.2 Структура жестового языка	6
1.3 Структура систем распознавания жестов	6
1.4 Обзор существующих систем для распознавания жестов	8
1.5 Выводы	14
2 Анализ алгоритмов обработки данных	16
2.1 Методы фильтрации	16
2.1.1 Медианная фильтрация	19
2.1.2 Алгоритм скользящего среднего	21
2.1.3 Фильтр Савицкого-Голея	22
2.1.4 Обоснование выбранного метода	24
2.2 Алгоритмы машинного обучения в обработке данных	25
2.2.1 Кластеризация данных методом k-средних	27
2.2.2 Кластеризация данных методом k-medoids	29
2.2.3 Обоснование выбранного метода	30
2.3 Выводы	31
3 Разработка программного обеспечения аппаратно-программного комплекса для распознавания жестов	32
3.1 Разработка алгоритма обработки поступающих данных	32
3.2 Разработка нейросетевого алгоритма	35
3.3 Разработка алгоритма определения жеста	38

3.4 Выводы.....	41
4 Лабораторные испытания и анализ полученных результатов	42
4.1 Результаты лабораторных испытаний.....	42
4.2 Выводы.....	44
Заключение	45
Список сокращений.....	46
Список использованных источников	47
Приложение А	51
Приложение Б.....	52

ВВЕДЕНИЕ

В настоящее время проблема коммуникации людей с ограниченными возможностями остается весьма актуальной. По данным Всемирной организации здравоохранения более 360 миллионов людей страдают глухотой или нарушением слуха. Согласно статистике за 2016 год [1], в Российской Федерации 14 миллионов (около 10%) человек обладают различными нарушениями слуха. Из них около 400 тысяч (около 3%) страдают тяжкими нарушениями. По данным Всероссийской переписи населения [2] лишь около 120 тысяч человек владеют русским жестовым языком.

Согласно информации о русском жестовом языке (РЖЯ) [3], для РЖЯ существует небольшое количество словарей. Стоит отметить, что ни один из них не является полным и лингвистически осмысленным. У людей, использующих язык жестов на территории России, коммуникация с обычными людьми вызывает ряд затруднений.

Целью данной работы является разработка программного обеспечения для устройства распознавания жестов русского жестового языка.

Сформулированы основные задачи для выпускной квалификационной работы:

- Выполнить анализ существующих систем для распознавания жестов;
- Выполнить анализ алгоритмов обработки данных;
- Разработать программное обеспечение, выполняющее функции распознавания жестов из дактильной азбуки русского жестового языка;
- Произвести тестирование программного обеспечения и провести анализ полученных результатов.

1 Анализ систем распознавания жестов

1.1 Общая информация о жестовом языке

На сегодняшний момент в мире насчитывается порядка 135 жестовых языков [4]. О развитии структуры жестового языка лучше всего известно на основе никарагуанского языка глухих. Он появился в восьмидесятые годы XX века. Прежде в Никарагуа не было общего жестового языка, но в 1977 году, когда создали первую школу для глухих, ученые сразу принялись фиксировать, как на протяжении двух-трех поколений развивалась грамматика и устанавливались правила. Дети общались со старшим поколением, перенимали и изменяли язык взрослых, создали правила, которых не было раньше. И такой естественный путь создания языка прошло большинство жестовых языков. На рисунке 1 для наглядности представлена дактильная азбука русского жестового языка.

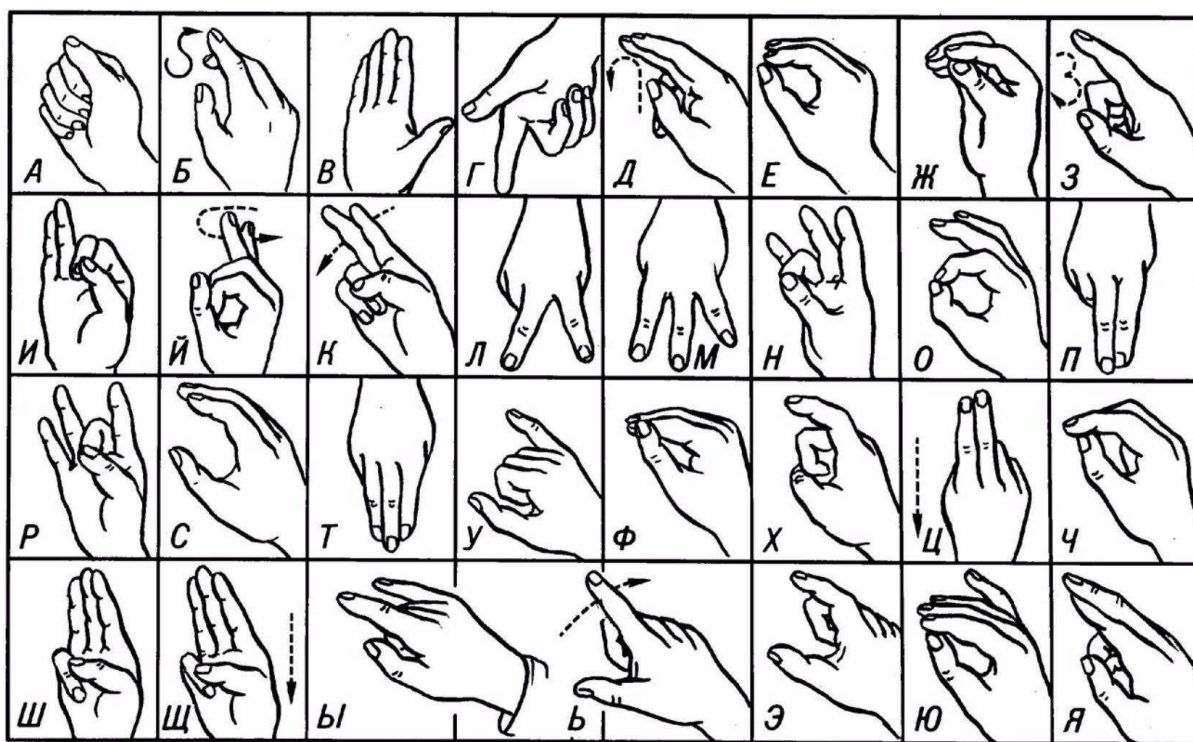


Рисунок 1 – Алфавит русского жестового языка

Систематично изучать русский жестовый язык начали только около десяти лет назад, при этом грамматика данного языка не описана [5], а словарей существует всего лишь несколько, ни один из которых не является полным и лингвистически осмысленным.

1.2 Структура жестового языка

Жест можно разделить на четыре компоненты: конфигурацию руки, движение, локализацию (место исполнения) и ориентацию руки. Плюс, учитывается мимика, а именно: выражение лица и движение губ.

Случается так, что понятие существует, а жест для него еще не придуман. Тогда используется дактильная азбука («дактиль» с греч. — «палец») — то есть жесты, соответствующие буквам словесного языка. Иногда слышащие люди принимают пальцевый алфавит за жестовый язык. Однако это разные вещи: в то время как пальцевым алфавитом передают звуки речи (буква за буквой), жестовые знаки представляют целые слова [6].

Многие жесты, используемые в русском жестовом языке, были заимствованы из французского, немецкого и австрийского жестовых языков, благодаря чему его лексика близка международной. С другой стороны, как и другие жестовые языки, в русский жестовый язык взято много слов из русского языка или трансформировавшихся под его влиянием. К примеру, жесты дней недели основаны на дактилировании первых букв, с которых начинаются эти слова в русском языке.

1.3 Структура систем распознавания жестов

Жест, в понимании для реализации различных интерфейсов управления, — это форма невербального общения, при которой некоторое действие, движение человеческого тела или его части, передает информацию. Жест

может содержать движение как одной, так и нескольких частей тела и нести эмоциональную нагрузку, что делает его информативным средством общения и взаимодействия.

Жесты классифицируются по следующим категориям:

- а) По типу:
 - 1) Простые жесты;
 - 2) Комплексные жесты;
- б) По способу обработки:
 - 1) Обработка после ввода;
 - 2) Обработка в процессе ввода;
- в) По длительности:
 - 1) Статические жесты;
 - 2) Динамические жесты.

Так, простыми жестами являются пальцевые жесты (для распознавания таких жестов используются, например, устройства с сенсорными экранами), жесты руками (Kinect, MYO) [7]. К комплексным жестам относятся движения, которые задействуют несколько частей тела или состоят из нескольких простых (Wii, Kinect) [8].

Классификация жестов по способу обработки разделяет их на 2 категории: жесты, обрабатываемые после ввода и жесты, обрабатываемые в процессе ввода. Пример жеста второй категории – увеличение/уменьшение фотографии в сенсорных устройствах фирмы «Apple» [9].

Длительность жеста показывает, изменяется ли он со временем. Следовательно, жесты могут быть разделены на статические и динамические. Статические жесты представляются в виде одного изображения, в то время как динамические жесты – это последовательность изображений.

Упрощенная структура системы распознавания жестов состоит из 3 блоков (рисунок 2):

- Блок с устройством ввода;
- Блок с методом распознавания;
- Реагирующий блок.

Устройство ввода является точкой входа в систему. Оно преобразовывает жест в цифровую форму и передает его в блок распознавания, который сравнивает поступившую информацию с шаблонами, хранящимися в базе данных, и в качестве результата возвращает распознанный жест. В итоге реагирующий блок выполняет некоторые функции в соответствии с распознанными жестами.

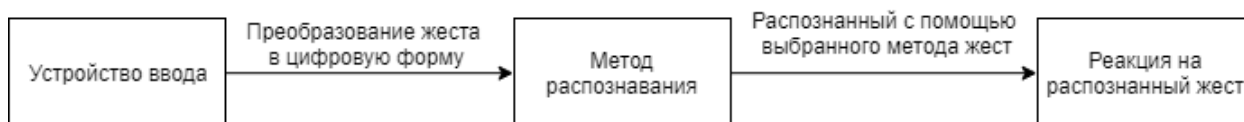


Рисунок 2 – Упрощенная структура системы распознавания жестов

1.4 Обзор существующих систем для распознавания жестов

В настоящее время существует множество систем для распознавания жестов, но не многие из них адаптированы под задачи распознавания алфавита русского жестового языка [10].

Устройства, используемые для захвата и распознавания жестов условно можно разделить на две группы: устройства на основе портативных датчиков и устройства на основе оптических камер.

К первой группе относятся, например, такие устройства, как проект, разработанный российскими энтузиастами для ролевых игр по вселенной всеми известной саги о ведьмаке Анджея Сапковского (перчатка «Ведьмака») [11] и перчатка «Language of Glove» [12].

Вторая группа устройств – это оптические камеры, которые записывают набор изображений для захвата движений жестов на расстоянии. Методы распознавания жестов, основанные на этих устройствах, распознают жесты, анализируя визуальную информацию, извлеченную из захваченных изображений. К подобным устройствам относятся такие системы, как мультисенсорная система распознавания жестов рук водителя [13] и Deep Sign [14] и Kinect [15].

Сравнительная характеристика рассмотренных аналогов приведена в таблице 1.

Таблица 1 – Сравнительная характеристика аналогов

Аналог	Мобильность устройства	Серийный выпуск	Переносимость ПО	Возможность модификации ПО	Наличие службы поддержки
Kinect	-	+	-	+	+
Мультисенсорная система для распознавания жестов рук водителя [13]	-	-	+	+	-
Deep sign [14]	-	-	-	+	-
Language of Glove	+	-	+	+	-
Перчатка «Ведьмака» [11]	+	-	-	-	-

Из таблицы 1 видно, что устройства на основе оптических камер (Kinect, Мультисенсорная система для распознавания жестов рук водителя, Deep Sign) не обладают таким качеством, как мобильность устройства, в отличие от устройств на основе портативных датчиков. Переносимостью программного обеспечения обладают только Мультисенсорная система для распознавания жестов рук водителя и «Language Of Glove», так как только эти системы представляют свои библиотеки в открытый доступ, и следовательно, их программное обеспечение может быть использовано на другой аппаратной платформе. Возможностью модификации программного обеспечения обладают все перечисленные аналоги, кроме «Перчатки Ведьмака», поскольку этот проект не имеет качественной документации в открытом доступе. Серийный выпуск и служба поддержки присутствуют только в Kinect – на официальном сайте постоянной основе появляются пакеты с обновлениями, а также различные поддерживаемые средства и расширения системы.

По результатам сравнительного анализа было принято решение подробно рассмотреть такие устройства для распознавания жестов, как Kinect и «The Language of Glove».

«Language of Glove» – это проект калифорнийских студентов, представленный в виде автономной перчатки для сурдоперевода.

Устройство основано на тензодатчиках, которые изготовлены из пьезорезистивного композиционного материала, содержащего частицы углерода, встроенные в гибкий материал. Маломощная печатная плата на перчатке преобразует девятизначный ключ в букву, а затем передает сигналы через Bluetooth на смартфон или экран компьютера. Датчики меняют свое электрическое сопротивление при растяжении или изгибе. Это позволяет им кодировать разные буквы алфавита английского языка, основываясь на положениях всех девяти суставов. Прямой или расслабленный сустав закодирован как «0», а согнутый – закодирован как «1». При подписании

буквы перчатка создает девятизначный двоичный ключ, который переводится в эту букву. Например, код для буквы «А» (большой палец, все остальные пальцы скручены) равен «01111111», а код «В» (большой палец, все остальные прямые) – «10000000». На рисунке 3 представлено устройство и принцип работы «Language of Glove».

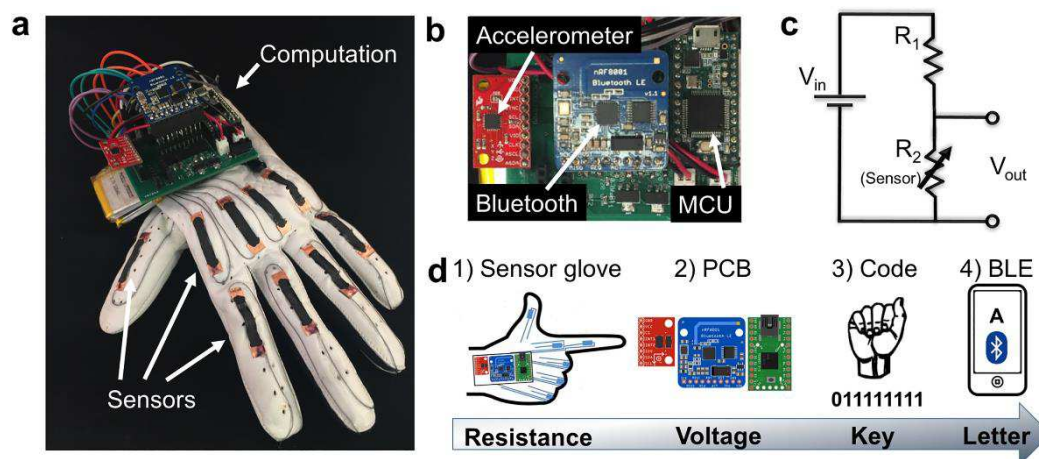


Рисунок 3 – Демонстрация устройства «The Language of Glove»

Программное обеспечение данного устройства основано на создании 3D-модели руки (или «виртуальной руки»), в которой, с помощью скрипта на объектно-ориентированном языке Python, каждый сустав руки был соединен с тремя точками на виртуальном объекте. Каждый виртуальный объект имел связанное с ним состояние изгиба с использованием стандартного геометрического преобразования. Был разработан алгоритм, принимающий сигналы напряжения от Arduino (через последовательный порт или текстовый файл) и сопоставляющий состояние изгиба каждого сустава на виртуальной руке с состоянием реальной руки. Программа принимает девятизначный ключ, соответствующий девяти согнутым или несогнутым суставам, который сопоставляется с внесенными в базу ключами 26 букв американского жестового языка. Траектория кисти сохраняется в файл для последующей визуализации с помощью открытого инструмента визуализации OVITO [16].

Таким образом, перчатка способна преобразовать 26 букв, найденных на американском языке жестов, в текст с помощью смартфона или компьютера. Перчатка была создана с целью улучшения взаимодействия в виртуальной реальности, поскольку его функции распознавания жестов «более интуитивно понятны», чем использование джойстика и других существующих контроллеров [17].

Kinect – бесконтактный сенсорный контроллер, оборудованный двумя сенсорами, цветной видеокамерой и микрофонной решеткой [15]. В данном устройстве первый инфракрасный сенсор служит для распознавания пространства вокруг пользователя, второй монохромный CMOS-сенсор предназначен для перенесения полученных данных в программную среду для дальнейшей их обработки и преобразования. Также устройство оснащено механической подставкой, при необходимости механический привод автоматически наклоняет сенсор вверх и вниз. Дальность распознавания 3-5 метров.

Kinect интерпретирует информацию 3D сцены с помощью инфракрасного проецируемого структурированного света (рисунок 4). Эта система трехмерного сканирования, называемая Light Coding [18], использует вариант трехмерного восстановления рисунка на основе полученных изображений. Датчик Kinect представляет собой горизонтальную панель, соединенную с небольшим основанием с помощью моторизованного шарнира и предназначенную для размещения выше или ниже монитора. В Kinect также есть камера RGB.



Рисунок 4 – Бесконтактный сенсорный контроллер Kinect

Разработанное для Kinect программное обеспечение использует библиотеку OpenCV [19]. Данная библиотека представляет собой набор алгоритмов из области компьютерного зрения. Она кроссплатформенная, и поддерживает языки программирования C/C++ и Python. С помощью неё осуществляется подготовка изображения для анализа, она содержит всевозможные фильтры, распознавание лиц и жестов, отслеживание объектов на изображении.

Данное программное обеспечение выполняет определения положений основных суставов тела человека. Однако в нем отсутствует механизм распознавания динамических жестов, более того данное программное обеспечение является закрытым, и для воспроизведения лежащего в его основе метода требуется большая база данных всевозможных поз человека для обучения классификатора. Качество снятых изображений чувствительно к освещению и помехам на заднем плане, поэтому очень трудно надежно находить и отслеживать изображение руки, что в значительной степени влияет на качество распознавания жестов. Также одним из главных недостатков устройства стоит отметить отсутствие мобильности и невозможности использовать его повсеместно.

1.5 Выводы

Рассмотрены основные аспекты структуры русского жестового языка, а также системы распознавания жестов. Произведен сравнительный анализ существующих систем для распознавания жестов. По результатам сравнительного анализа было принято решение подробно рассмотреть системы Kinect и «The Language of Glove». Библиотека OpenCV, используемая в Kinect, позволяет выполнить распознавание жестов и отслеживание объектов на изображении, однако в виду того, что устройство основано на использовании датчиков глубины и камер, оно ненадежно в

измерениях при применении в различных условиях, например, в темноте, а также не обладает таким качеством, как портативность.

Устройство «The Language of Glove», напротив, может быть использовано при любых заданных условиях освещенности и видимости в виду того, что перчатка никак не связана с камерами и не использует датчики захвата движения, опирается лишь на датчики сгиба и подобные ему приборы. Программное обеспечение данного устройства использует специально разработанный алгоритм, который обрабатывает поступающий с датчиков сигнал, определяет, был ли согнут палец, и сопоставляет девятизначный ключ, соответствующий девяти согнутым или несогнутым суставам, с внесенным в базу ключами 26 букв американского жестового языка. Проанализировав алгоритм работы перчатки, было выяснено, что точность распознавания зависит от показаний, полученных с датчиков. Система не учитывает особенности рук каждого человека (например, размер руки, хорошая или плохая гибкость пальцев), что может стать проблемой при определении положения сустава (согнут или не согнут).

Таким образом, проанализировав несколько существующих устройств для распознавания жестов, были выделены основные требования к программной части для аппаратно-программного комплекса: создание программного обеспечения для устройства в виде библиотек (по принципу Kinect), а также использование алгоритмов обработки данных, а именно фильтрацию цифрового сигнала и самоадаптивные алгоритмы для улучшения точности распознавания и получения возможности использовать перчатку в различных условиях.

Для обеспечения высокой точности распознавания жеста, необходимо производить первоначальную обработку данных, включающую в себя фильтрацию данных, поступающих с датчиков изгиба. Использование алгоритмов машинного обучения, на пример на основе нейронной сети также позволит увеличить точность распознавания.

2 Анализ алгоритмов обработки данных

2.1 Методы фильтрации

Разработанное устройство имеет в своем составе датчики сгиба функционирующие на основе принципа ослабления светового пучка (рисунок 5).

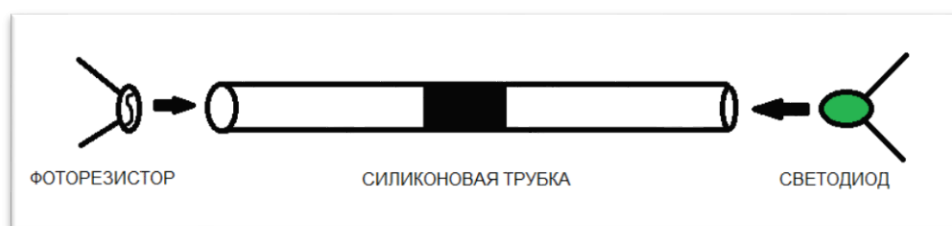


Рисунок 5 – Датчик сгиба

Используемые датчики обладают определенной погрешностью, кроме того следует учитывать ряд внешних и внутренних факторов, которые приводит к формированию шума в выходном сигнале датчика [20]. Экспериментально определено, что для данного датчика наиболее часто появляющимися шумами являются импульсные, случайные помехи.

Поэтому предлагается программная реализация уменьшения случайных импульсных помех в сигнале, то есть организация программного цифрового фильтра.

В данном случае цифровой фильтр рассматривается как реализация математического алгоритма, входом которого является цифровой сигнал, а выходом – другой цифровой сигнал с определенным образом модифицированной формой.

Это позволит обеспечить наибольшую точность диапазонов значений напряжения при распознавании состояния пальца и наиболее правильное определение жеста.

Поскольку сигнал, поступающий с перчатки – это медленно изменяющийся низкочастотный сигнал, то главная задача фильтра – убрать пики с поступающего сигнала. Таким образом, для фильтрации может подойти любой сглаживающий фильтр.

На основании результатов поиска алгоритмов фильтрации был выделен ряд фильтров: медианный фильтр, фильтр с использованием алгоритма скользящего среднего, а также фильтр Савицкого-Голея.

Для данных фильтров в среде моделирования MatLab разработаны проекты и выполнено моделирование (п.п. 2.1.1-2.1.3). На рисунке 6 изображены данные, полученные с датчика, когда палец находился в 4 заданных состояниях:

- 1) палец выпрямлен;
- 2) средняя фаланга пальца опущена (Состояние пальца: полусогнут);
- 3) нижняя фаланга пальца опущена (Состояние пальца: перпендикулярно к ладони);
- 4) средняя и нижняя фаланги пальца опущены (Состояние пальца: согнут).

На представленном графике (рисунок 6) цифрами 1, 2, 3 и 4 обозначены состояния фаланги пальца, $t_1 \dots t_n$ – это переходный процесс из одного состояния в другое, U_k – это диапазон напряжений, соответствующий определенному состоянию пальца, а I – случайная импульсная помеха.

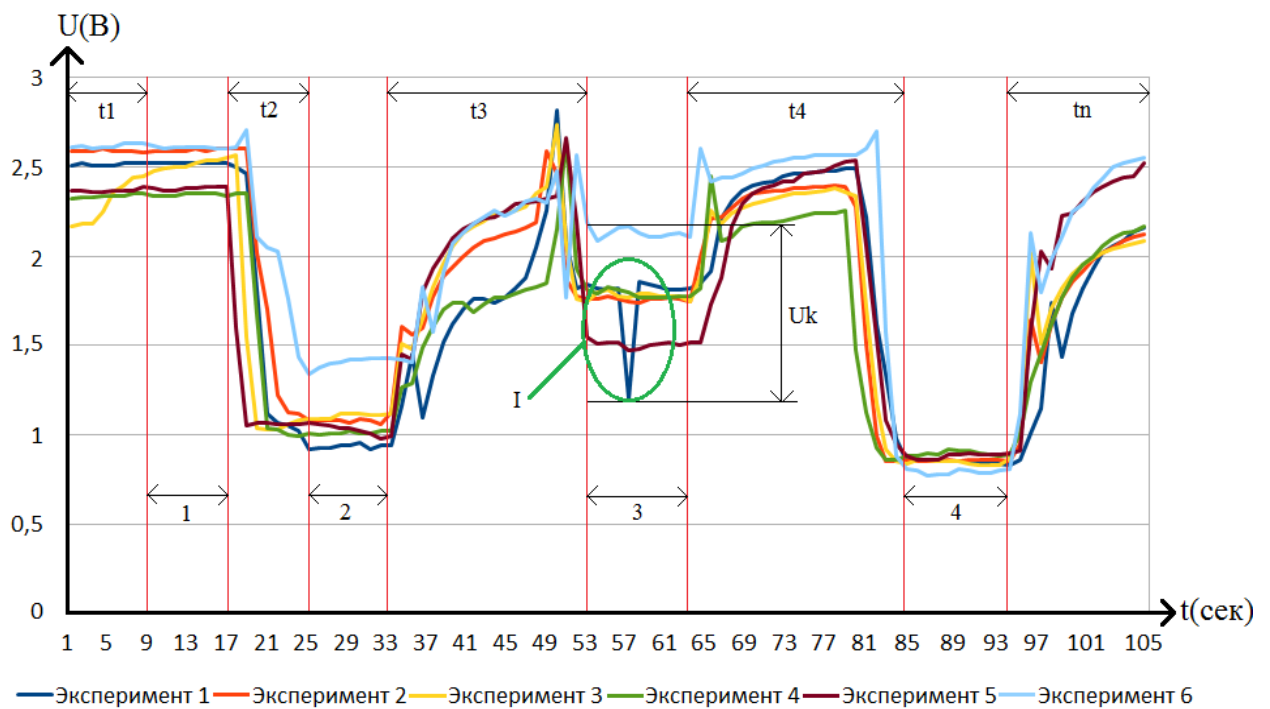


Рисунок 6 – Данные, полученные с датчика, в четырех заданных состояниях фаланги пальца

Далее были рассмотрены различные методы фильтрации. Входные данные для фильтрации изображены на рисунке 7.

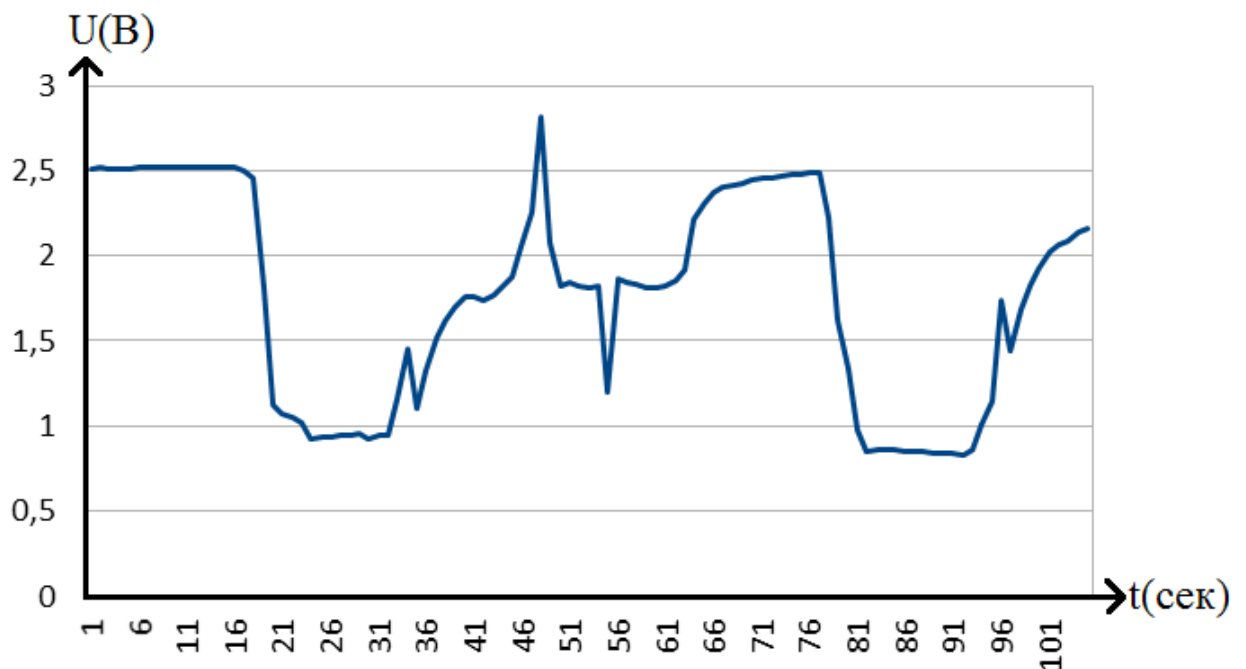


Рисунок 7 – Входные данные для фильтрации

2.1.1 Медианная фильтрация

Медианная фильтрация относится к нелинейным алгоритмам фильтрации. Суть данной фильтрации состоит в том, что к входному сигналу применяется скользящее окно [21] заданной длины, отсчеты в пределах окна упорядочиваются и в качестве выходного отсчета возвращается значение из середины упорядоченного окна (или полусумма двух ближайших к середине элементов, если окно имеет четную длину). Медианная фильтрация применяется, например, для устранения импульсных помех (щелчков) при обработке звуковых сигналов [22].

Специфической особенностью таких фильтров является слабая реакция на отсчеты, резко выделяющиеся на фоне соседних. Благодаря этой особенности, медианные фильтры при оптимально выбранной апертуре могут, например, сохранять без искажений резкие границы объектов, эффективно подавляя некоррелированные или слабо коррелированные помехи и малоразмерные детали. Это свойство позволяет применять медианную фильтрацию для устранения аномальных значений в массивах данных, уменьшения выбросов и импульсных помех.

Значения отсчетов внутри окна фильтра сортируются в порядке возрастания (убывания); и значение, находящееся в середине упорядоченного списка, поступает на выход фильтра. В случае четного числа отсчетов в окне выходное значение фильтра равно среднему значению двух отсчетов в середине упорядоченного списка. Окно перемещается вдоль фильтруемого сигнала и вычисления повторяются.

Таким образом, медианная фильтрация осуществляет замену значений отсчетов в центре апертуры медианным значением исходных отсчетов внутри апертуры фильтра. На практике апертура фильтра для упрощения алгоритмов обработки данных, как правило, устанавливается с нечетным числом отсчетов.

Для иллюстрации данного метода фильтрации была написана программа в среде Matlab с использованием встроенной функции *medfilt1*, реализующей одномерную медианную фильтрацию. Функция принимает два параметра: x – входные данные и n – порядок фильтрации.

Результаты фильтрации медианным фильтром 5-го порядка можно увидеть на рисунке 8.

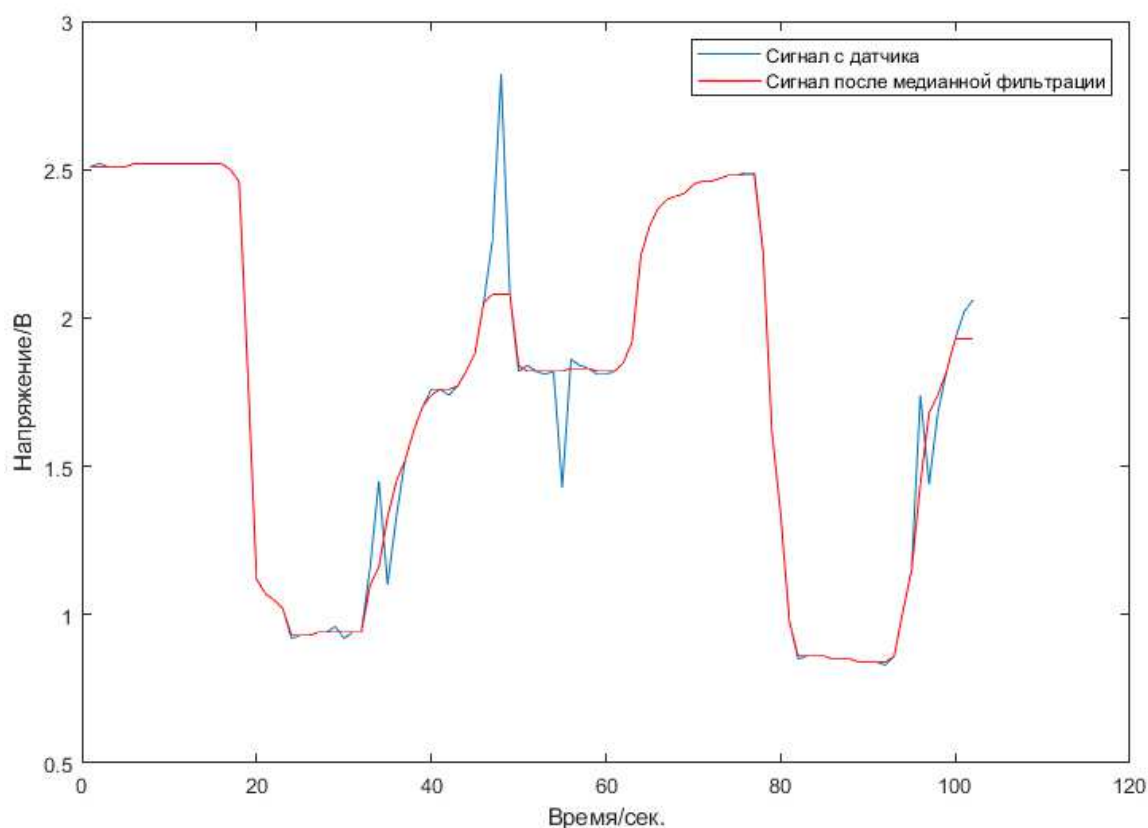


Рисунок 8 – Использование медианного фильтра в Matlab

По данному графику (рисунок 8) можно сделать вывод, что медианный фильтр действительно подавляет случайные импульсные помехи, когда палец находится в определенном положении, и при этом сохраняет без искажений броски напряжения во время изменения положения пальца.

2.1.2 Алгоритм скользящего среднего

В методе скользящего среднего исходные данные y_i сглаживаются по следующему правилу:

$$y_{s_i} = \frac{1}{2N + 1} \sum_{k=-N}^N y(i + k), \quad (1)$$

где $2N + 1$ – число точек, выбираемых для сглаживания [23].

Суть алгоритма заключается в том, что слева и справа от текущей точки выбирается по N точек. Данные, расположенные в точках, близких к границам отрезка, не сглаживаются. Метод заключается в замене фактических значений членов ряда средним арифметическим значений нескольких ближайших к нему членов. Набор усредняемых значений образует так называемое окно скольжения. Член, значение которого заменяется на среднее по окну, занимает в окне срединное положение.

Результаты фильтрации методом скользящего среднего с шириной окна усреднения, равной 5, можно увидеть на рисунке 9.

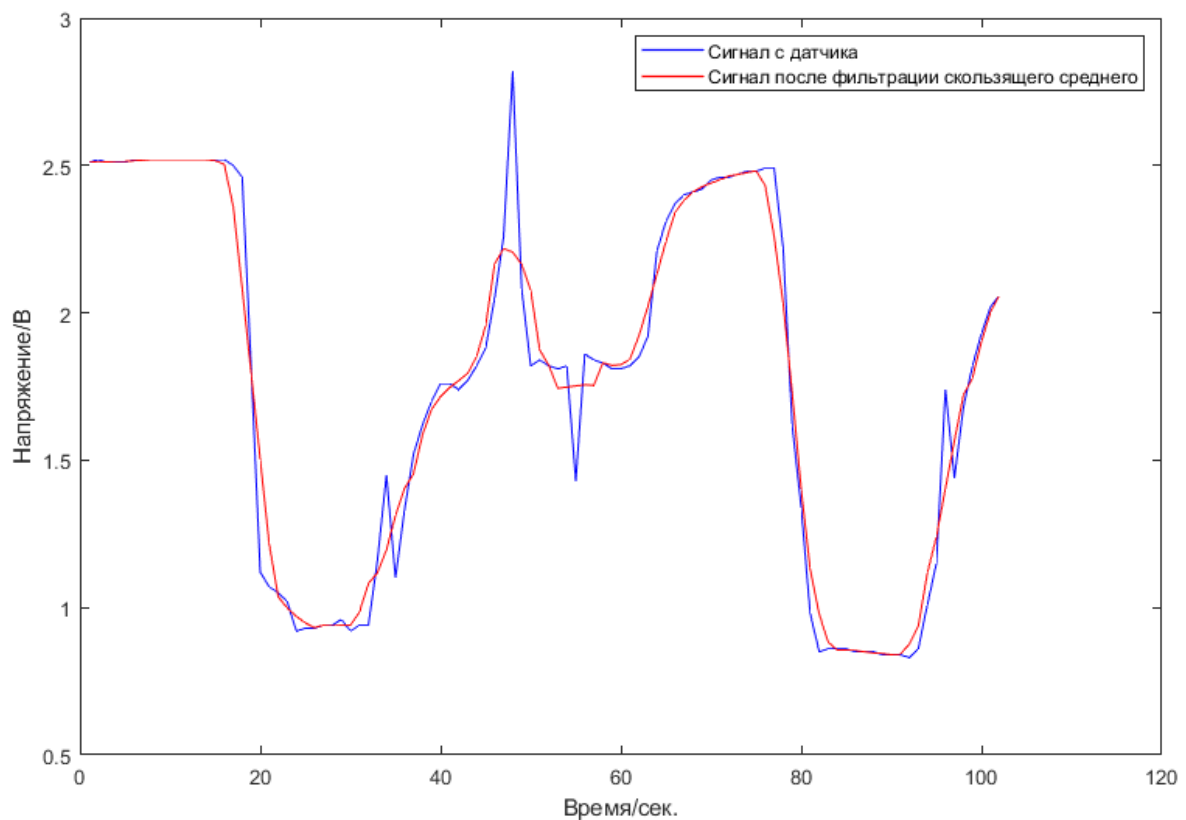


Рисунок 9 – Использование фильтрации методом скользящего среднего в Matlab

На данном графике (рисунок 9) можно увидеть, что фильтр эффективно сглаживает помехи во время перехода из одного состояния в другое, однако не полностью убирает случайные импульсные помехи, когда палец находится в определенном положении фаланги пальца.

2.1.3 Фильтр Савицкого-Голея

Фильтр Савицкого-Голея осуществляет полиномиальную аппроксимацию отдельных кадров входного сигнала [24]. В целом, фильтр Савицкого-Голея является развитием метода скользящего среднего. В нем вместо линейной аппроксимации в окрестности каждого измерения $x_i = x(t_i)$ по методу наименьших квадратов строится аппроксимирующий

полином k -го порядка. Выборочное значение x_i заменяется значением полинома y_i в этой точке:

$$y_i = a_0 + a_1 t_i + a_2 t_i^2 + \dots + a_k t_i^k. \quad (2)$$

При этом на усредняющем интервале (окно сглаживания) используется по m соседних точек слева и справа от точки t_i .

Для иллюстрации данного метода фильтрации была написана программа в среде Matlab с использованием встроенной функции *sgoleyfilt*, реализующей дискретную фильтрацию с помощью фильтра Савицкого-Голея. Данная функция имеет три входных параметра: x – входные данные; k – степень сглаживающих полиномов, она должна быть меньше размера кадра m . Размер кадра должен быть нечетным. Если $k = m-1$, фильтр не вносит изменений во входной сигнал.

Результаты фильтрации с помощью фильтра Савицкого-Голея 3-го порядка при длине кадра, равной 5, можно увидеть на рисунке 10.

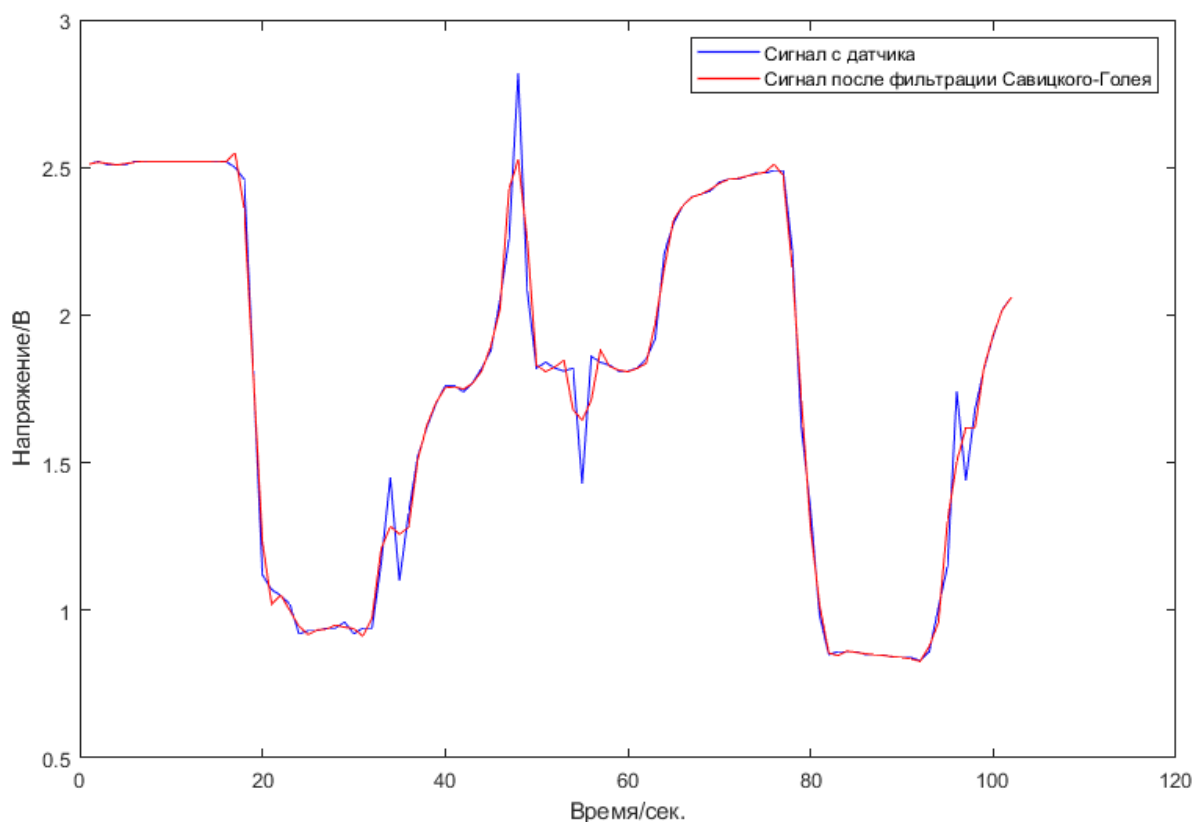


Рисунок 10 – Использование фильтрации Савицкого-Голея в Matlab

На представленном графике (рисунок 10) видно, что фильтр сглаживает слабые импульсные помехи во время перехода из одного состояния в другое, при этом сохраняя резкие изменения напряжения.

2.1.4 Обоснование выбранного метода

Проведя моделирование выбранных методов фильтрации, необходимо дать оценку в данном случае не по общему принципу работы, а по работе конкретно для представленной выборки данных. Фильтр Савицкого-Голея по сравнению с медианным фильтром лучше сглаживает слабые помехи при смене положений пальца, однако не убирает помехи, когда палец находится в заданном положении, то есть не выполняет поставленной задачи, так как необходимо фильтровать помехи именно в момент, когда палец находится в определенном состоянии. Фильтр с использованием метода среднего

скользящего в целом сильнее сглаживает входной сигнал, чем два других рассмотренных фильтра, однако для поставленной нами задачи требуется сохранить резкие спады напряжений. В виду этого, было принято решение использовать медианный фильтр в аппаратно-программном комплексе для распознавания жестов.

2.2 Алгоритмы машинного обучения в обработке данных

Каждый человек обладает индивидуальными характеристиками руки – размер руки, длина пальцев, хорошая или плохая гибкость пальцев. Индивидуальные особенности могут сказываться на точности распознавания. Задача использования самообучающегося алгоритма в аппаратно-программном комплексе для распознавания жестов возникает в виду необходимости устройству в дальнейшем подстраиваться под морфологию рук человека. За счет разных морфологических особенностей показания с датчиков будут подвергаться корреляционным изменениям, и, учитывая это, необходимо уметь точно определять показанный человеком жест.

В связи с этим возникает задача провести исследование доступных решений машинного обучения и проверить их пригодность для использования в аппаратно-программном комплексе для распознавания жестов, основанном на получении информации о жесте за счет обработки поступающих значений напряжений с тензодатчиков.

В первую очередь стоит отметить главную составляющую практически всех алгоритмов машинного обучения – наличие у объекта признаков, выделяющих его среди других объектов. В случае с диапазонами напряжений таковыми будут являться пределы значений, выделяющие положения фаланг пальцев. Необходимо также учитывать наличие двух основных способов обучения – с учителем и без него. В первом случае, для каждого прецедента задается пара «ситуация, требуемое решение», а в случае с обучением без

учителя, для прецедента задается только «ситуация», при этом требуется сгруппировать объекты в кластеры, используя данные о попарном сходстве объектов, или понизить размерность данных [25].

Применение алгоритмов машинного обучения обуславливается тем, что необходимо произвести классификацию получаемых данных по признакам. Это требует высокой точности распознавания. В ходе проведенных экспериментов были обнаружены конфликтующие области, содержащие данные, уменьшающие точность определения жеста. А потому использование простого метода установления границ напряжений не эффективно для данной задачи.

Так, на рисунке 11 выделены основные конфликтующие области: при ограничении напряжений, возникают конфликты между областями 2 и 4, а также 1 и 3. При попадании в данные области, устройство может неправильно распознать жест и, например, вместо того, чтобы зафиксировать состояние «полусогнут» (2 область), устройство выдаст информацию о том, что палец полностью согнут (4 область), и распознавание будет выполнено неверно.

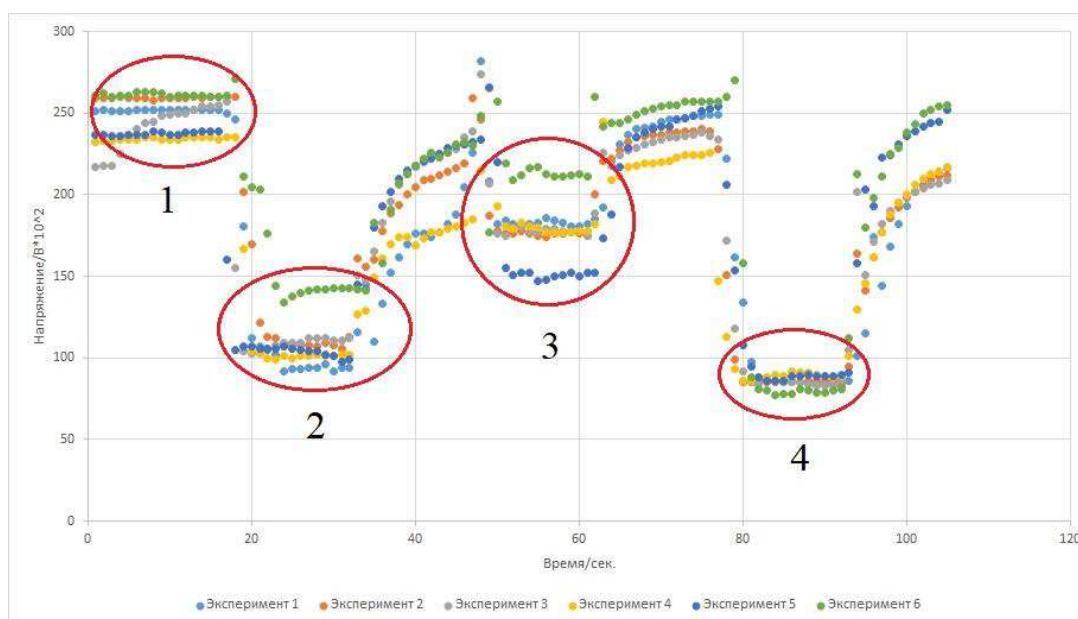


Рисунок 11 – Выборка данных с иллюстрацией конфликтующих областей

Поэтому было принято решение рассмотреть алгоритмы машинного обучения, способные предотвратить данную ситуацию и решить проблему с пересечением диапазонов данных того или иного жеста. Кроме того, алгоритм машинного обучения, распознающий жесты, также должен соответствовать следующим требованиям: обладать высокой точностью распознавания и использовать оптимальное количество вычислительных ресурсов.

Для проверки работы алгоритмов была взята обработанная медианным фильтром выборка данных с прототипа устройства распознавания жестов, получающего данные с одного пальца.

Для объединения в группы по признакам используются алгоритмы кластеризации – семейство наиболее популярных методов машинного обучения. Согласно анализу доступных источников, было выделено два метода кластеризации: кластеризация методом k -средних и методом k -медиан. В среде Matlab были созданы проекты и выполнено моделирование.

2.2.1 Кластеризация данных методом k -средних

В области анализа данных широко распространена задача разделения множества объектов на подмножества таким образом, чтобы все объекты каждого подмножества имели больше сходства друг с другом, чем с объектами других подмножеств.

Метод k -средних – это один из методов кластерного анализа, применяемый для решения ряда подобных задач. Основная идея алгоритма k -средних заключается в том, что данные произвольно разбиваются на кластеры, после чего итеративно перевычисляется центр масс для каждого кластера, полученного на предыдущем шаге, затем векторы разбиваются на кластеры вновь в соответствии с тем, какой из новых центров оказался ближе по выбранной метрике. Цель алгоритма заключается в разделении n

наблюдений на k кластеров таким образом, чтобы каждое наблюдение принадлежало ровно одному кластеру, расположенному на наименьшем расстоянии от наблюдения.

Преимущество в использовании для перчатки – известны примерные образцовые значения для того или иного положения фаланги, которые, по сути, будут являться значениями центров масс четырех кластеров, связанных с распознаванием соответственно четырех жестов.

Метод k -means случайным образом выбирает центр масс для каждого из k кластеров и присваивает каждому объекту метку того кластера, расстояние до центра масс которого от него меньше. А затем, на каждой итерации, алгоритм вычисляет центры масс кластеров и переписывает их метки объектам до сходимости, то есть неизменности меток всех объектов. На рисунке 12 представлены результаты моделирования алгоритма k -means для выборки данных, полученной с датчика изгиба.

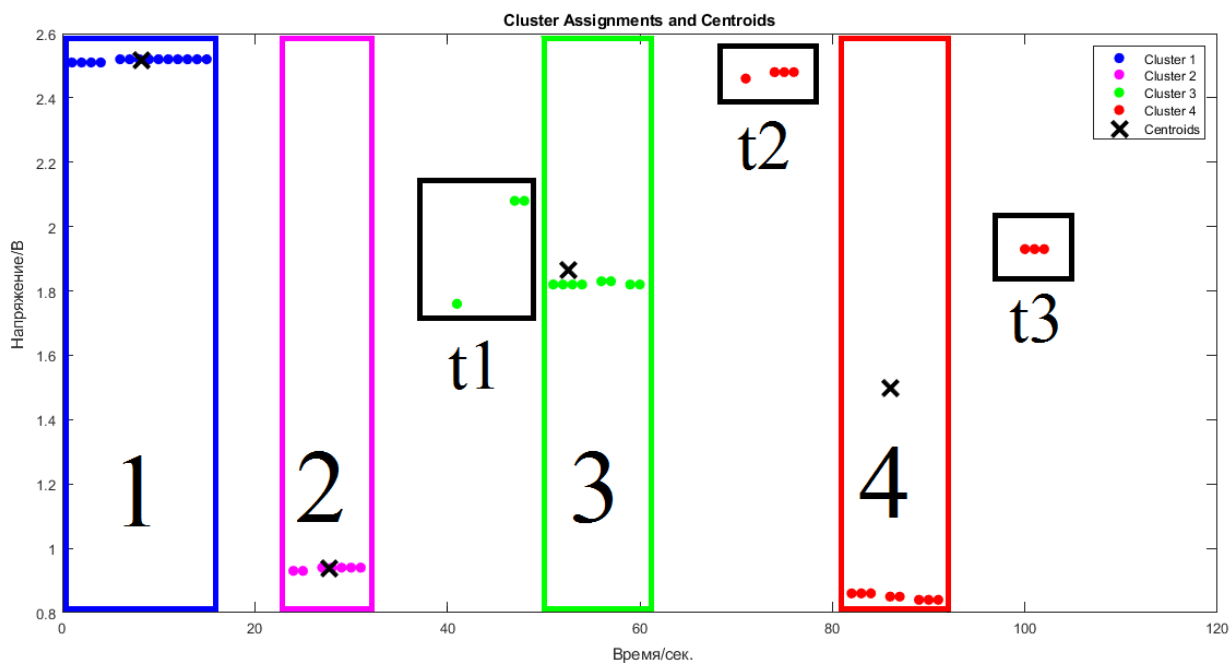


Рисунок 12 – Моделирование использования алгоритма k -средних

Алгоритм определил 4 кластера в соответствии с 4 заданными положениями пальца: выпрямлен (1), полусогнут (2), нижняя фаланга опущена (3) и согнут (4). На графике (рисунок 11) видно, что в виду того, что в 4 кластер попали значения переходных состояний t_2 и t_3 , центроид четвертого кластера был определен неверно.

2.2.2 Кластеризация данных методом k-medoids

Алгоритм k-medoids является модификацией классического алгоритма кластеризации k-means и предназначен для решения задач выделения групп объектов (кластеров) в случаях, когда проводится кластеризация объектов без использования свойств линейного пространства [26].

Функция kmedoids () в matlab разбивает наблюдения матрицы на k кластеров и возвращает вектор, содержащий индексы кластеров каждого наблюдения. Подобно функции kmeans, kmedoids по умолчанию использует квадрат евклидовых расстояний и модификация алгоритма k-means – алгоритм k-means ++ для выбора начальных положений медоида кластера.

В отличие от k-means, k-medoids выбирает в качестве центра масс медианный объект из кластера, таким образом, решая проблему устойчивости к выбросам. Произведено моделирование алгоритма k-medoids в среде Matlab, результаты которого изображены на рисунке 13.

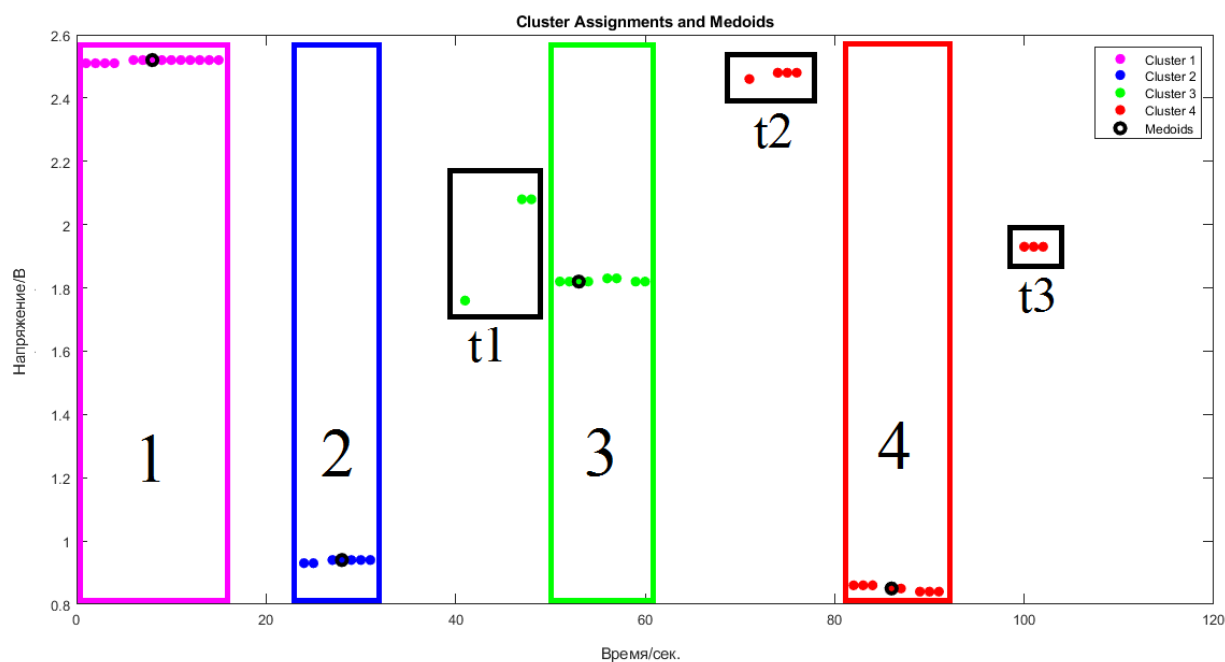


Рисунок 13 – Моделирование использования алгоритма k-medoids

На представленном графике (рисунок 13) видно, что центральные значения кластеров определены с более высокой точностью, чем при использовании алгоритма k-means. К тому же, несмотря на то, что значения переходных состояний t1 были определены в кластер 3, а значения в t2 и t3 были определены в кластер 4, центральные значения (медоиды) этих кластеров были вычислены верно.

2.2.3 Обоснование выбранного метода

При анализе результатов моделирования выбранных методов кластеризации, необходимо дать оценку в данном случае не по общему принципу работы, а по работе конкретно для представленной выборки данных. Алгоритм k-средних не точно определяет центроиды, так как при расчете среднего значения учитываются значения во время переходов из одного состояния пальца в другое. Алгоритм k-medoids берет медианное значение в кластере, значения из переходных состояний не влияют на определение медоидов. Следовательно, было принято решение использовать

кластеризацию методом k-medoids в аппаратно-программном комплексе для распознавания жестов.

2.3 Выводы

Произведено исследование сглаживающих фильтров цифрового сигнала. На основе анализа доступных источников [22] были выделены три сглаживающих фильтра: медианный фильтр, фильтр скользящего среднего и фильтр Савицкого-Голея. Произведено моделирование выбранных фильтров в среде Matlab на основе выборки, полученной с датчика изгиба. Произведен анализ полученных результатов. Выбран медианный фильтр как наиболее подходящий фильтр для нормализации поступающих с датчика данных.

При анализе экспериментальных данных была найдена ситуация, в ходе которой происходит неверное определение положений фаланг пальцев за счет возникновения конфликтующих областей на значениях, полученных с датчика изгиба. Для решения данной проблемы был рассмотрен вариант использования алгоритмов кластеризации. Произведен анализ алгоритмов кластеризации, в ходе которого были выбраны два алгоритма, наиболее подходящие для решения данной задачи – k-means и k-medoids. Произведено моделирование данных алгоритмов на выборке данных с конфликтующими областями. На основе результатов анализа был выбран алгоритм k-medoids для решения данной проблемы. Алгоритм точно определяет положение фаланги пальца без возникновения ситуации с пересечением диапазонов напряжений.

3 Разработка программного обеспечения аппаратно-программного комплекса для распознавания жестов

Разработка алгоритма работы программы аппаратно-программного комплекса для распознавания жестов было произведено с помощью инструментов MATLAB.

3.1 Разработка алгоритма обработки поступающих данных

Алгоритм поделен на два этапа: определение значений каждого пальца в тот момент времени, когда показывается жест, и определение буквы алфавита русского жестового языка, соответствующей положениям всех пяти пальцев.

Для начала для каждого файла, содержащего значения напряжений, полученные с датчика, вызывается функция **DataProcessing**. Входным параметром этой функции является файл напряжениями **dataSet**, а на выход подается двумерный массив **X**, в который записываются напряжения из данного файла и соответствующие им отсчеты времени. Функция записывает в массив **finger** напряжения, затем вызывается функция медианного фильтра **MedianFilt**, которая фильтрует поступающий на вход массив **finger** и выдает отфильтрованные значения напряжений. Затем цикл проходит по строкам отфильтрованного массива и проверяет, отличаются ли соседние значения более, чем на 0,02. Если условие принимает значение «истина» (результатом сравнения является единица), то текущее значение удаляется из массива, путем присвоения ему значения NaN (Not a Number). Затем значения измененного одномерного массива **finger** записываются в двумерный массив **X**, содержащий два столбца – количество отсчетов времени **observations** и соответствующие им значения напряжений **finger**.

Затем для двумерного массива **X** вызывается функция **Clustering**, на вход которой подается массив **X** и количество кластеров для кластеризации. Функция выполняет кластеризацию методом k-медиан и возвращает столбец-вектор, состоящий из такого же количества строк, как и **X**, в каждой строке которого находится номер кластера, к которому относится соответствующее значение в **X**, а также возвращает двумерный массив **C**, в первой строке которого находится медианное значение напряжения в первом кластере и соответствующий ему отсчет времени, во второй строке медианное значение второго кластера, и так далее.

После этого данные массива **C** сортируются по времени с помощью функции **SortByTime**. По окончании цикла формируется двумерный массив **Gestures**, в каждую строку которого записываются медианные значения массивов **C** с каждого пальца. Затем для массива символов **Symbols** вызывается функция **DetectLetter**, которая возвращает массив символов, заполненный буквами алфавита в соответствии с положениями пальцев в каждом столбце массива **Gestures**. Функция **DetectLetter** использует обученную нейронную сеть для определения буквы алфавита РЖЯ в соответствии с положениями пальцев. После этого на экран выводится массив распознанных букв.

Блок-схема алгоритма обработки поступающих данных представлена на рисунке 14.

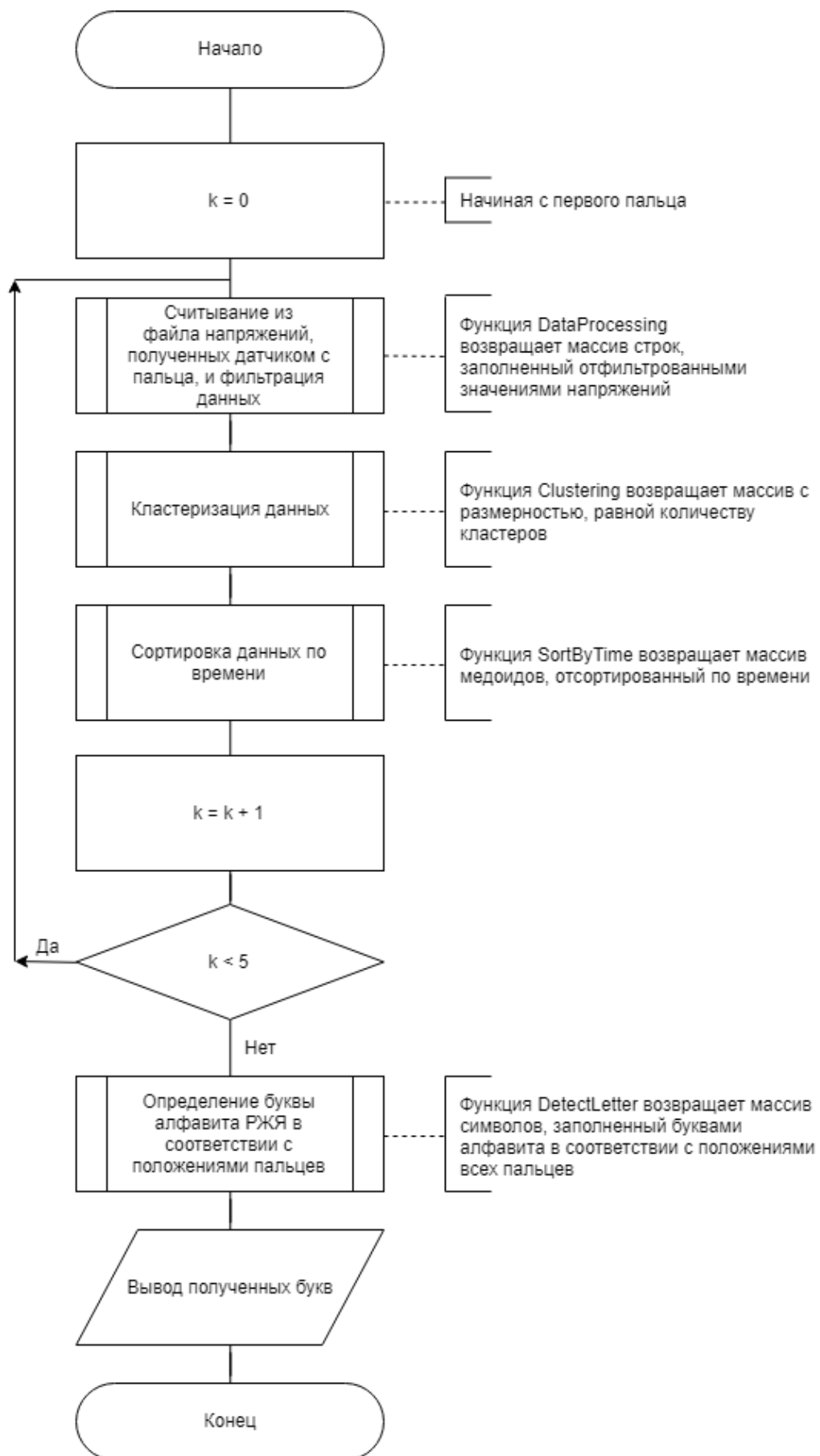


Рисунок 14 – Блок-схема алгоритма распознавателя жестов

3.2 Разработка нейросетевого алгоритма

Для улучшения точности распознавания с помощью средств среды Matlab была разработана нейронная сеть. Ее обучение было проведено на образцовых данных расположения кластеров, полученных с помощью алгоритма k-medoids, приведенного в п.п. 2.2.2.

Таким образом получается строка с пятью образцовыми значениями напряжений с каждого пальца. На одну такую строку приходится 100 строк тестовых значений напряжений (рисунок 15).

	Тестовые значения					Образцовые значения					
	A	B	C	D	E	F	G	H	I	J	
Для буквы А	91	1,814	0,813	0,622	0,607	0,714	1,7	0,7	0,7	0,7	0,7
	92	1,46	0,514	0,47	0,414	0,657	1,7	0,7	0,7	0,7	0,7
	93	1,742	0,543	0,761	0,668	0,493	1,7	0,7	0,7	0,7	0,7
	94	1,537	0,546	0,49	0,825	0,551	1,7	0,7	0,7	0,7	0,7
	95	1,618	0,493	0,564	0,801	0,763	1,7	0,7	0,7	0,7	0,7
	96	1,402	0,58	0,777	0,753	0,763	1,7	0,7	0,7	0,7	0,7
	97	1,846	0,85	0,778	0,775	0,841	1,7	0,7	0,7	0,7	0,7
	98	1,589	0,719	0,785	0,554	0,551	1,7	0,7	0,7	0,7	0,7
	99	1,716	0,795	0,699	0,787	0,664	1,7	0,7	0,7	0,7	0,7
	100	1,654	0,75	0,648	0,483	0,502	1,7	0,7	0,7	0,7	0,7
Для буквы В	101	1,96	1,96	1,949	1,955	2,116	2,5	2,5	2,5	2,5	2,5
	102	2,252	2,488	1,971	2,161	2,498	2,5	2,5	2,5	2,5	2,5
	103	2,533	2,421	2,323	2,5	2,062	2,5	2,5	2,5	2,5	2,5
	104	2,323	2,07	2,306	2,055	2,453	2,5	2,5	2,5	2,5	2,5
	105	2,593	2,283	2,41	2,422	2,075	2,5	2,5	2,5	2,5	2,5
	106	2,595	2,047	2,113	1,959	2,006	2,5	2,5	2,5	2,5	2,5
	107	2,212	1,932	2,259	2,577	2,335	2,5	2,5	2,5	2,5	2,5
	108	2,468	1,957	2,246	2,366	2,009	2,5	2,5	2,5	2,5	2,5
	109	2,087	2,385	2,153	2,393	2,017	2,5	2,5	2,5	2,5	2,5
	110	2,53	2,316	2,566	2,084	2,544	2,5	2,5	2,5	2,5	2,5
	111	2,492	2,423	1,901	2,229	2,425	2,5	2,5	2,5	2,5	2,5
	112	2,15	2,376	2,386	2,36	1,964	2,5	2,5	2,5	2,5	2,5
	113	2,248	2,354	2,444	2,532	1,915	2,5	2,5	2,5	2,5	2,5

Рисунок 15 – Формат данных значений напряжений для жеста

Всего было взято 20 различных образцовых значений, так как дактильная азбука русского жестового языка содержит только 20 статичных жестов, без использования данных о положении руки и пальцев в пространстве. Каждое из 20 образцовых значений по 100 раз было записано в массив данных **SampleDataSet** размерностью 2000 на 5. На каждое

образцовое значение в массив данных **TrainingDataSet** размерностью 2000 на 5 было записано 100 различных строк тестовых значений напряжений. Эти данные были взяты для обучения нейронной сети. Количество нейронов в скрытом слое нейронной сети равно 30.

Структура нейронной сети изображена на рисунке 16.

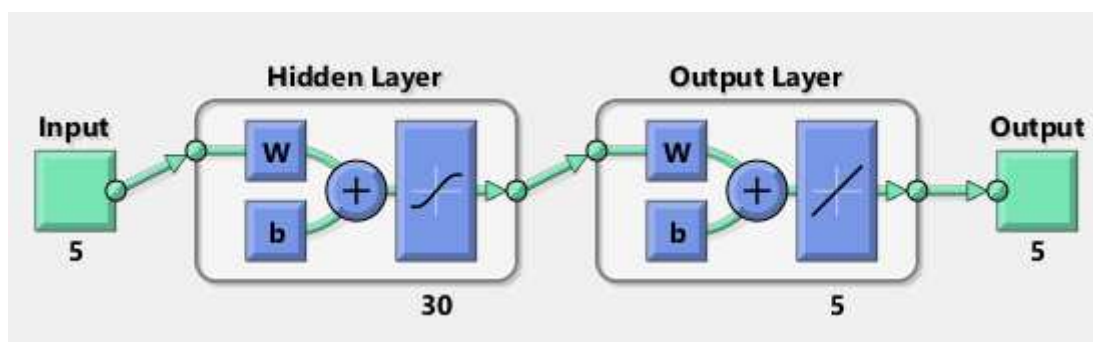


Рисунок 16 – Структура нейронной сети

Для обучения был выбран алгоритм Левенберга-Марквардта [27]. Алгоритм обучения заключается в последовательном приближении заданных начальных значений параметров к искомому локальному оптимуму (значению из образцовой выборки). В данном алгоритме в качестве критерия оптимизации используется среднеквадратичная ошибка модели на обучающей выборке. На рисунке 17 видно, что наилучшее значение среднеквадратичной ошибки модели было получено на 83 итерации.

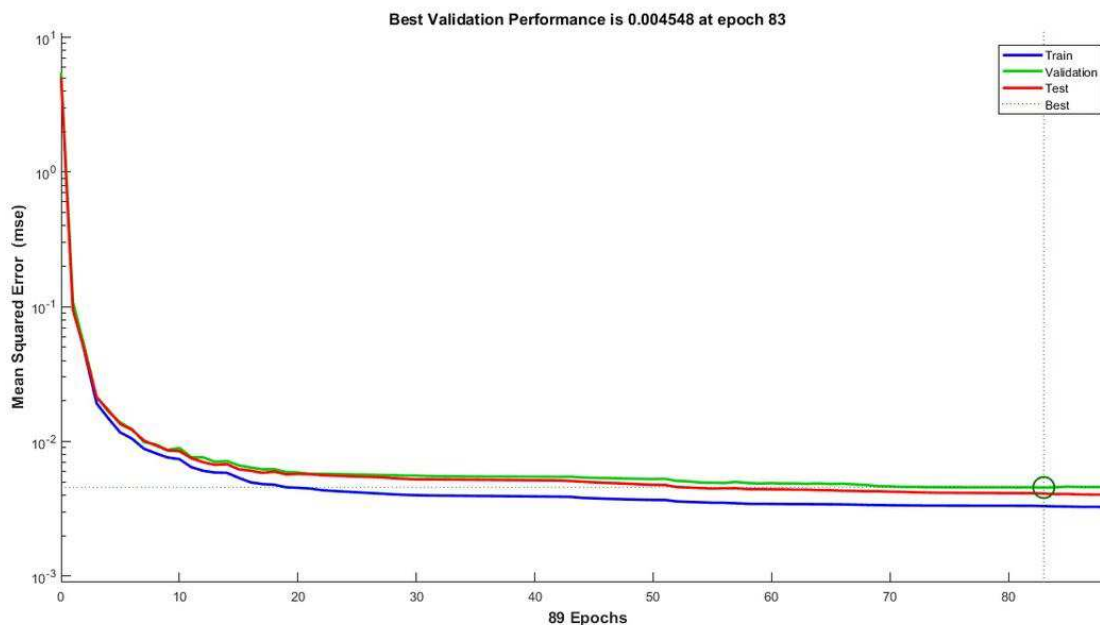


Рисунок 17 – График изменения валидации данных в течении 89 итераций

Прогоняя 100 значений для одной образцовой строки, с помощью данного алгоритма нейронная сеть высчитывает коэффициент максимального приближения для каждой образцовой строки.

Затем, когда начинается использование нейронной сети, ей передаются реальные значения напряжений и она рассчитывает выходные значения с помощью высчитанных при обучении коэффициентов максимального приближения к подходящему образцовому значению. Таким образом, на выход нейронной сети подаются 5 напряжений, максимально приближенных к одной из строк в образцовой выборке данных.

Для проверки эффективности работы нейронной сети был проведен сравнительный анализ работы алгоритма с использованием нейронной сети и без использования нейронной сети. На вход программы в обоих случаях было подано 70 строк с пятью значениями напряжений с каждого пальца, то есть по 10 различных значений напряжений для 7 различных букв.

На рисунке 18 изображен график точности распознаваний алгоритма с использованием нейросети и алгоритма без нейросети.

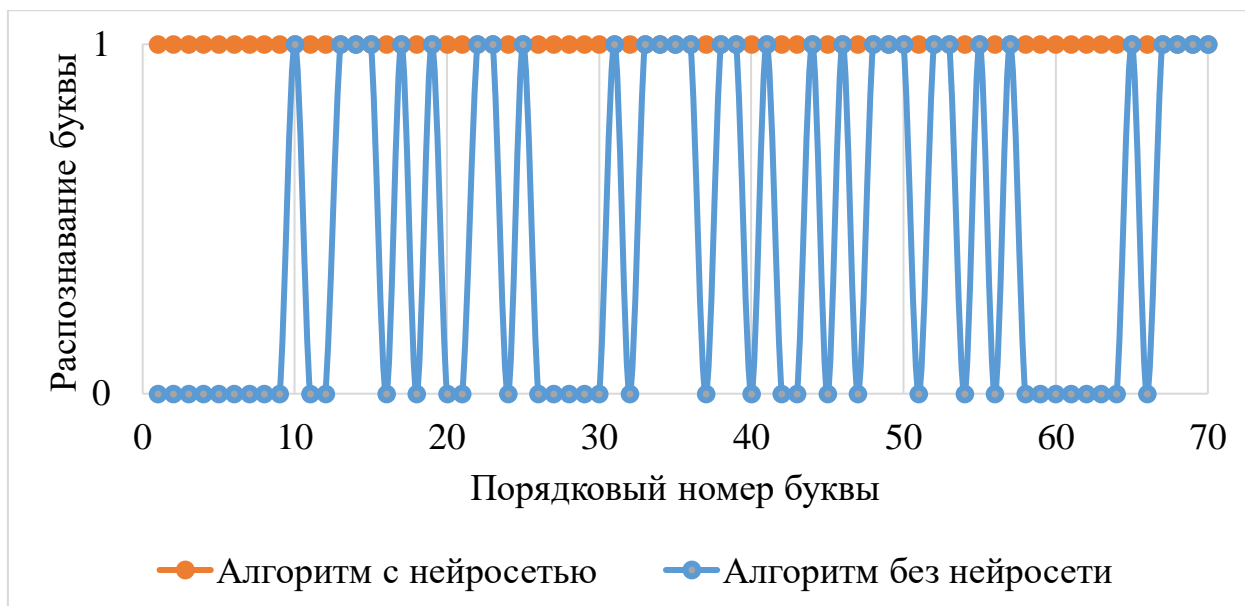


Рисунок 18 – График точности распознаваний алгоритма с использованием и без использования нейронной сети

На приведенном графике (рисунок 18) по оси абсцисс представлены номера входящих строк со значениями напряжений для одной из 7 различных букв, а на оси ординат показано, правильно была распознана предполагаемая буква (если буква была распознана верно, значит ее значение на оси ординат равно 1). На графике видно, что алгоритм без нейросети верно распознал только 31 букву из 70 предложенных. Алгоритм с использованием нейросети верно распознал все 70 предполагаемых букв. Таким образом, на основе анализа полученных результатов, можно сделать вывод, что точность распознавания увеличивается на 56% при использовании в алгоритме для распознавания нейронной сети.

3.3 Разработка алгоритма определения жеста

Функция определения жеста **DetectLetter** принимает на вход вектор-столбец **code**, который содержит распознанные значения напряжений на 5 пальцах, матрицу **LettersCode** размерностью 31 x 5, содержащую образцовую выборку значений напряжений для каждого жеста из дактильной

азбуки русского жестового языка, а также вектор-столбец **Letter**, в каждой строке которого находятся буквы, соответствующие значениям напряжений в каждой строке матрицы **LettersCode**. В функции запускается цикл, который проходит по всем строкам матрицы **LettersCode** и в котором проверяется, равны ли распознанные значения напряжений значениям, находящимся в строках образцовой выборки, с погрешностью $\pm 0,1$. Если строка в матрице, соответствующая распознанным значениям, была найдена, в переменную **symbol** заносится значение буквы на соответствующей строке в **Letter**. В противном случае в **symbol** запишется символ «-», что означает, что программе не удалось распознать жест.

Блок-схема алгоритма определения жеста представлена на рисунке 19.

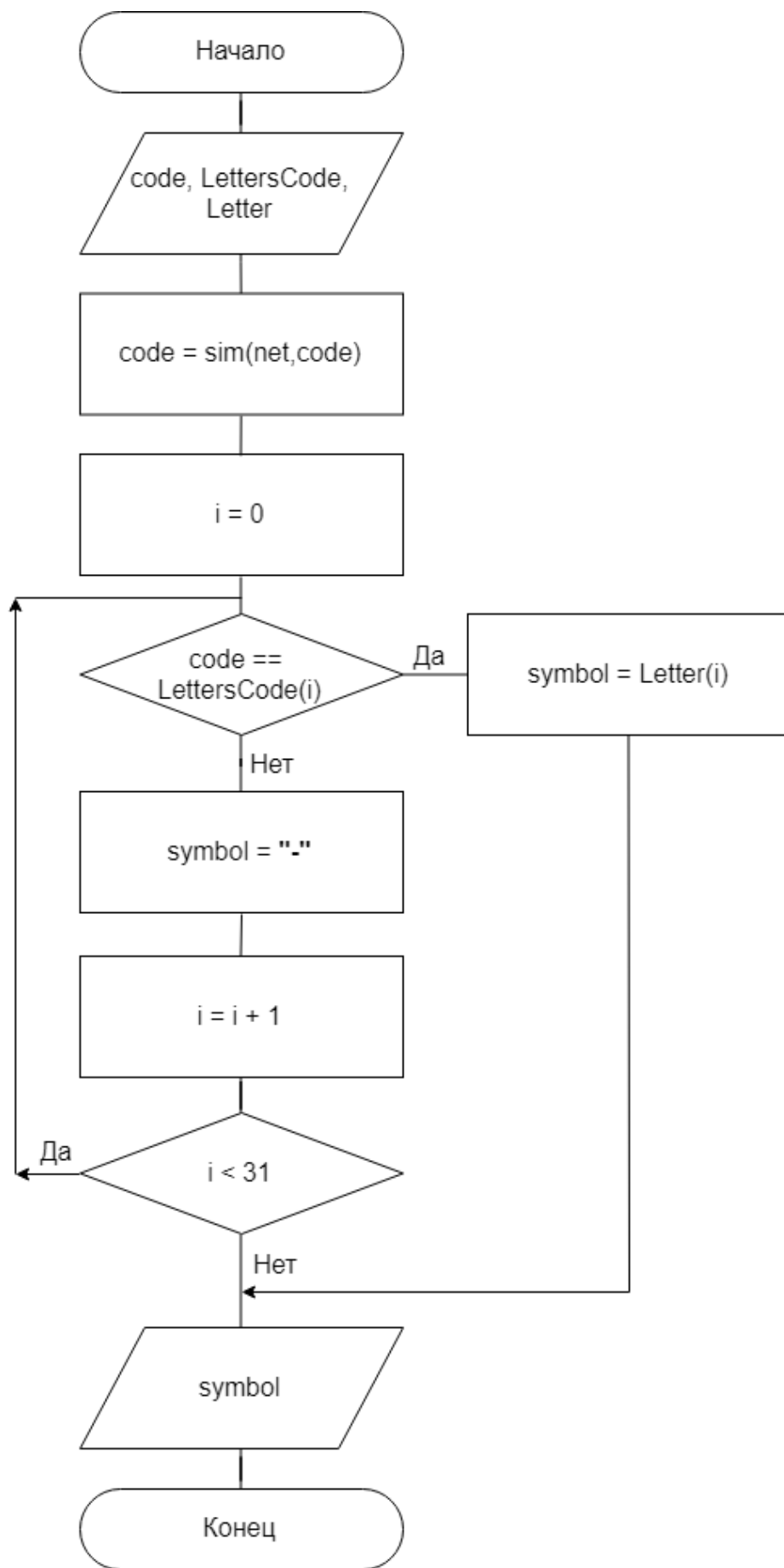


Рисунок 19 – Блок-схема алгоритма определения жеста

3.4 Выводы

Разработаны алгоритмы обработки поступающих данных, описаны основные этапы функционирования алгоритмов, включающие ранее описанные методы кластеризации и фильтрации цифрового сигнала. Выполнено обучение нейронной сети. Определено, что система «умеет» распознавать 20 жестов из русского жестового языка. Проведен сравнительный анализ работы алгоритма с использованием нейронной сети и без использования нейронной сети. По результатам анализа был сделан вывод, что использование нейронной сети увеличивает точность распознавания на 56%.

4 Лабораторные испытания и анализ полученных результатов

Для подтверждения работоспособности алгоритма, были проведены испытания на выборке данных, полученной в реальном времени с прототипа, реализующего модуль управления аппаратно-программного комплекса.

Прототип состоит из макетной платы, к которой подсоединена платформа Arduino Uno, а также 5 датчиков изгиба, расположенных на перчатке и измеряющих изменение напряжений (рисунок 20).

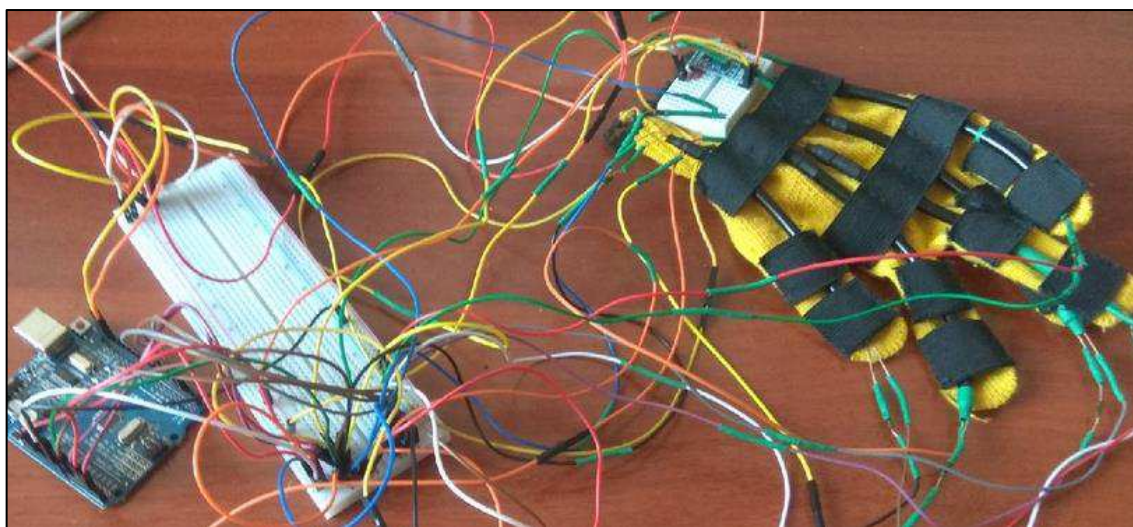


Рисунок 20 – Лабораторный стенд

4.1 Результаты лабораторных испытаний

Для распознавания были показаны 4 жеста из дактильной азбуки русского жестового языка: «М», «Е», «Н» и «Ю» (рисунок 21).

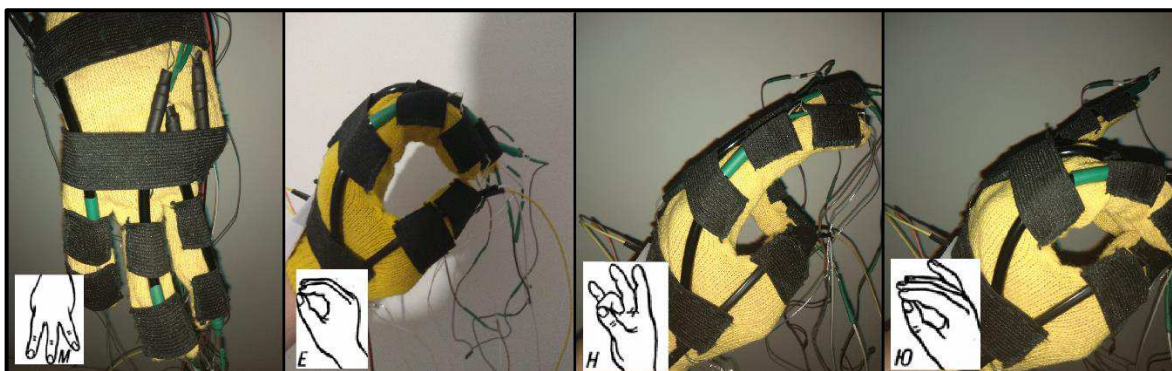


Рисунок 21 – Изображение букв «М», «Е», «Н» и «Ю» с помощью устройства

Для изображения буквы «М» необходимо согнуть большой палец и мизинец, а указательный, средний и безымянный пальцы остаются выпрямленными. Для изображения буквы «Е» все пальцы должны быть полусогнутыми. Для буквы «Н» полусогнуты должны быть большой и безымянный пальцы, для «Ю» – полусогнуты все, кроме мизинца.

Данные о напряжениях, полученные с датчиков, были переданы в программу, после чего произошел процесс обработки по алгоритму, описанному в главе 3.1, и распознанные буквы были выведены на экран (рисунок 22).

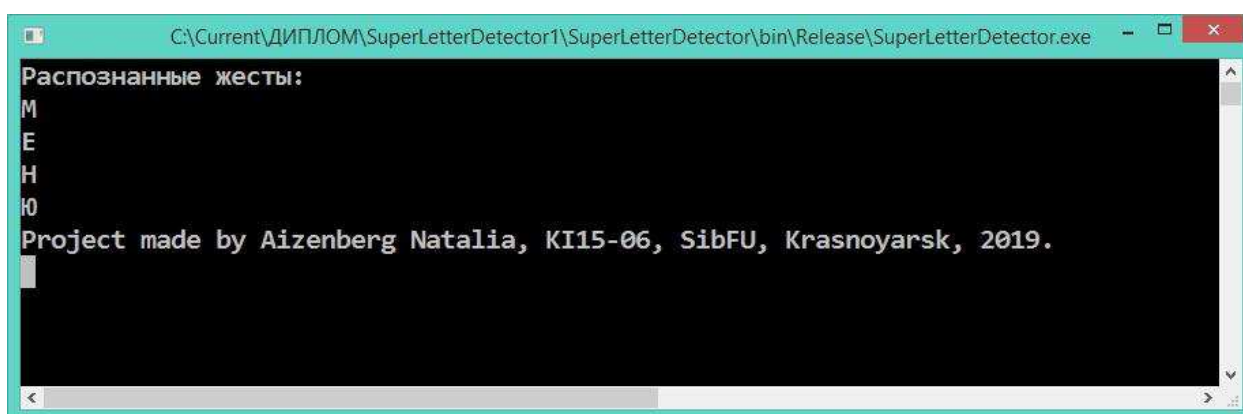


Рисунок 22 – Вывод результатов распознавания на экран

По результатам тестирования можно сделать вывод, что программа производит распознавание жестов с высокой точностью.

4.2 Выводы

В ходе тестирования на вход программы были поданы данные о напряжениях, полученных датчиками с 5 пальцев. Программа произвела обработку этих данных и вывела результат распознавания на экран.

В результате лабораторных испытаний при распознавании буква «Е» распознавалась не всегда точно. На первые 10 попыток один раз вместо «Е» на экран выводился знак «-», что означает, что жест не был распознан. Тем не менее, по мере обучения нейронной сети ошибка распознавания была устранена. По окончательным результатам тестирования на реальной выборке данных алгоритм успешно выполнил распознавание жестов из тактильной азбуки русского жестового языка.

ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы было разработано программное обеспечение аппаратно-программного комплекса для распознавания жестов, включающее в себя фильтрацию поступающих данных, а также использование самоадаптивных алгоритмов для повышения точности распознавания.

Так, в ходе проведенной работы, был написан алгоритм для распознавания жестов из русского жестового языка, включающий в себя использование медианного фильтра для нормализации поступающих данных, и применение методов машинного обучения, а именно алгоритм кластеризации методом k-медиан для повышения точности при определении диапазонов напряжений. Для реализации распознавания была написана обучающаяся нейронная сеть, принимающая на вход значения напряжений, которые в дальнейшем сравнивались с эталонной выборкой для установления соответствий с выбранными жестами дактильной азбуки.

Разработанное программное обеспечение удовлетворяет поставленным требованиям: состоит из библиотек и использует алгоритмы обработки данных.

По результатам лабораторных испытаний было установлено, что разработанный алгоритм распознает жесты из дактильной азбуки русского жестового языка с высокой точностью.

Результатами работы также является 2 публикации, 1 призовой диплом 2 степени за доклад на студенческой конференции «Перспектив Свободный – 2018» (приложение А), призовой сертификат за участие на студенческой конференции «Перспектив Свободный – 2019» (приложение Б).

Перспективами данной работы является использование данных с акселерометра для распознавания динамических жестов из дактильной азбуки русского жестового языка.

СПИСОК СОКРАЩЕНИЙ

- РЖЯ – Русский Жестовый Язык
- CMOS – complementary metal-oxide-semiconductor
- RGB – red green blue
- OVITO – Open Visualization Tool
- OpenCV – Open Computer Vision

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Тем, кто не слышит [Электронный ресурс]: Коммерсантъ. – Режим доступа: <https://www.kommersant.ru/doc/3171206>
2. Лингвисты увлеклись жестовым языком [Электронный ресурс]: Информационный сайт глухих, слабослышащих и всех. – Режим доступа: <http://www.deafnet.ru/dn/new.phtml?c=70&id=10568&page=212>
3. Лингвистика жестового языка [Электронный ресурс]: N+1: научные статьи, новости, открытия. – Режим доступа: <https://nplus1.ru/material/2016/04/13/jesture>
4. Жестовые языки мира [Электронный ресурс]: Лаборатория лингвистики жестового языка. – Режим доступа: <https://signlang.ru/science/lang/>
5. Давиденко, Т.П. Краткий очерк по лингвистике РЖЯ / Т.П. Давиденко, А.А. Комарова // Современные аспекты жестового языка. – 2006. – №1 – С.146 –161.
6. Из чего состоит жест [Электронный ресурс]: Лаборатория лингвистики жестового языка. – Режим доступа: <https://signlang.ru/studyrsl/sign-components/>
7. ОСНОВНЫЕ ЖЕСТЫ ПРИ РАБОТЕ С СЕНСОРНЫМИ ЭКРАНАМИ [Электронный ресурс]: Бесплатные онлайн-курсы от компании QATestLab. – Режим доступа: <https://training.qatestlab.com/blog/technical-articles/basic-touch-gestures/>
8. Распознавание жестов в Kinect [Электронный ресурс]: Microsoft – официальная страница. – Режим доступа: <https://support.xbox.com/ru-RU/xbox-360/accessories/body-controller>

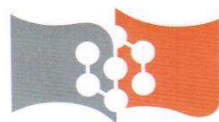
9. Visual interpretation of hand gestures for human-computer interaction: a review [Электронный ресурс]: IEEE Xplore Digital Library. – Режим доступа:<https://ieeexplore.ieee.org/document/598226>
10. Распознавание жестов без камеры и аккумуляторов [Электронный ресурс]: ПОПУЛЯРНАЯ МЕХАНИКА. – Режим доступа:<https://www.popmech.ru/technologies/15601-raspoznavanie-zhestov-bez-kamery-i-akkumulyatorov/>
11. Магические жесты как вызов для электронщика [Электронный ресурс] : Хабр. – Режим доступа: <https://geektimes.ru/post/289005/>
12. The Language of Glove: Wireless gesture decoder with low-power and stretchable hybrid electronics [Электронный ресурс]: PLOS ONE: accelerating the publication of peer-reviewed science. – Режим доступа: <http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0179766>
13. Multi-sensor System for Driver's Hand-Gesture Recognition[Электронный ресурс]: CSAIL. - Режим доступа: <https://people.csail.mit.edu/kapu/papers/DriverHandGestureFG2015.pdf>
14. Deep sign — универсальная технология распознавания жестов [Электронный ресурс]: vc.ru. – Режим доступа:<https://vc.ru/tribuna/38876-deep-sign-universalnaya-tehnologiya-raspoznavaniya-zhestov>
15. Kinect – как устроен и работает [Электронный ресурс] : aver.ru – всё о новинках техники. – Режим доступа: <http://aver.ru/interesno/kinect-kinect-kak-ustroen-i-rabotaet/>
16. The Open Visualization Tool [Электронный ресурс]: OVITO - The Open Visualization Tool. – Режим доступа:<https://ovito.org/>
17. Стодолларовая перчатка переведет язык жестов в текст [Электронный ресурс]: N+1: научные статьи, новости, открытия. – Режим доступа: <https://nplus1.ru/news/2017/07/13/ASL-glove>

18. PrimeSense: Motion Control Beyond the Kinect [Электронный ресурс]: PCMag.com – Technology Product Reviews, News, Prices & Tips. – Режим доступа: <http://forwardthinking.pcmag.com/gadgets/282321-primesense-motion-control-beyond-the-kinect>
19. Справочник по OpenCV [Электронный ресурс]: OpenCV-Tutorial.ru – Справочник по OpenCV на русском с примерами кода. – Режим доступа: <https://opencv-tutorial.ru/>
20. Шумы и помехи [Электронный ресурс]: Bourabai Research Institution. – Режим доступа: <http://bourabai.ru/signals/ts0102.htm>
21. Оконное сглаживание [Электронный ресурс]: Displib.ru – Теория и практика цифровой обработки сигнала. – Режим доступа: <http://www.dsplib.ru/content/win/win.html>
22. А.Б.Сергиенко. Signal Processing Toolbox – обзор [Электронный ресурс]: Matlab.Exponenta.ru – Matlab и Simulink – сообщество пользователей, материалы, книги, форум. – Режим доступа: <http://matlab.exponenta.ru/signalprocess/book2/index.php>
23. Алгоритм скользящего среднего (Simple Moving Average) [Электронный ресурс]: Habr.com. – Режим доступа: <https://habr.com/ru/post/134375/>
24. Основы работы в Curve Fitting Tools [Электронный ресурс]: Matlab.Exponenta.ru – Matlab и Simulink – сообщество пользователей, материалы, книги, форум. – Режим доступа: http://matlab.exponenta.ru/curvefitting/3_10.php#02
25. Машинное обучение [Электронный ресурс]: MachineLearning.ru - Профессиональный информационно-аналитический ресурс, посвященный машинному обучению, распознаванию образов и интеллектуальному анализу данных. – Режим доступа: http://www.machinelearning.ru/wiki/index.php?title=Машинное_обучение

26. Быстрый алгоритм кластерного анализа k-medoids [Электронный ресурс]: НАУЧНАЯ ЭЛЕКТРОННАЯ БИБЛИОТЕКА «КИБЕРЛЕНИНКА». – Режим доступа:<https://cyberleninka.ru/article/v/bystryy-algoritm-klasternogo-analiza-k-medoids>
27. Алгоритм Левенберга-Марквардта [Электронный ресурс]: MachineLearning.ru – Профессиональный информационно-аналитический ресурс, посвященный машинному обучению, распознаванию образов и интеллектуальному анализу данных. – Режим доступа:http://www.machinelearning.ru/wiki/index.php?title=Алгоритм_Левенберга-Марквардта

ПРИЛОЖЕНИЕ А

Диплом 2 степени за доклад на студенческой конференции «Перспект
Свободный – 2018»



SIBERIAN FEDERAL UNIVERSITY СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ SIBERIAN FEDERAL UNIVERSITY

ДИПЛОМ

II СТЕПЕНИ

НАГРАЖДАЕТСЯ

Айзенберг Наталья Михайловна,

Бывшев Егор Сергеевич,

Мальгин Георгий Юрьевич

*студенты Института космических и информационных технологий
Сибирского федерального университета*

за доклад, успешно представленный

**на Международной конференции
студентов, аспирантов и молодых учёных**

«Перспект Свободный-2018»

(научный руководитель – ст. преподаватель А.С. Даниленко)

И.о. ректора



В.И. Колмаков

В.И. Колмаков

Красноярск 2018 г.

ПРИЛОЖЕНИЕ Б

Сертификат за участие на студенческой конференции «Перспектив Свободный – 2019»

ПРОСПЕКТ СВОБОДНЫЙ
МЕЖДУНАРОДНАЯ КОНФЕРЕНЦИЯ МОЛОДЫХ УЧЕНЫХ

Международная студенческая конференция «ПРОСПЕКТ СВОБОДНЫЙ – 2019»,
посвященная Международному году Периодической таблицы химических элементов Д. Менделеева

Секция Трехлетняя лингвистика

Институт Космических и информационных технологий

СЕРТИФИКАТ очного участия
в конференции

Выдан Лизенберг Наталья Михайловна
(фамилия, имя, отчество полностью)

участнику Международной конференции студентов, аспирантов и молодых ученых
«Перспектив Свободный – 2019», состоявшейся 22–26 апреля 2019 года, г. Красноярск

Председатель жюри К.А. С.И. Кузнецов

Секретарь А.В. Романовская

23.04.2019г.
(дата)

ПРОСПЕКТ СВОБОДНЫЙ
2019

Министерство науки и высшего образования РФ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Космических и информационных технологий
институт

Вычислительная техника
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

О. В. Непомнящий

подпись

инициалы, фамилия

«27» 06 20 19 г.

БАКАЛАВАРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника
код и наименование направления

Программная часть аппаратно-программного комплекса
для распознавания жестов

тема

Пояснительная записка

Руководитель



проф., зав. каф. ВТ,
канд.тех.наук


О. В. Непомнящий

подпись, дата

должность, ученая степень

инициалы, фамилия

Выпускник


 24.06.2019г.

подпись, дата

Н.М. Айзенберг

инициалы, фамилия

Нормоконтролер


26.06.19

доцент,
канд.тех.наук

подпись, дата

должность, ученая степень

В. И. Иванов

инициалы, фамилия

Красноярск 2019