

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ

Ректор ПетрГУ,
д-р техн.наук, профессор

_____ А. В. Воронин
« ____ » _____ 2021 г.

ОТЧЕТ ОБ ИССЛЕДОВАНИЯХ

“Программно-аппаратные методы сенсорики и машинного восприятия для
робототехнических систем с автономным движением”

ЭТАП № 1 в рамках Соглашения 075-15-2021-1007

(заключительный)

Руководитель проекта,
вед. науч. сотрудник,
канд. физ.-мат. наук, доцент

_____ Д. Ж. Корзун
(подпись, дата)

СПИСОК ИСПОЛНИТЕЛЕЙ

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	7
1 Разработка концептуального прототипа модульной робототехнической системы	8
1.1 Программно-аппаратный прототип распознавателя движения физических объектов и человека с возможностями оценки качества движения	8
1.1.1 Введение	8
1.1.1.1 Постановка задачи	8
1.1.2 Архитектура программно-аппаратного прототипа	9
1.1.2.1 Описание архитектуры	9
1.1.2.2 Описание структурных элементов	10
1.1.3 Структура программного обеспечения	14
1.1.3.1 Цифровая модель - Репо Г.	15
1.1.3.2 Программа управления манипулятором	16
1.1.3.3 Синхронизатор положения цифровой модели и реального робота	17
1.1.3.4 Программа расчета точности	19
1.1.3.5 Документация	19
1.1.4 Испытания и эксперименты	20
1.1.4.1 Описание стенда для испытаний	20
1.1.4.2 Программа испытаний	21
1.1.4.3 Эксперименты	21
1.1.5 Выпускные квалификационные работы	24
1.1.6 Практика студентов	24
1.1.7 Рабочая программа практики	24
1.1.8 Публикации	24
1.1.9 Результаты	25
1.2 Программно-аппаратный прототип распознавателя действий человека или других объектов по отношению к человеку или друг к другу	26
1.2.1 Введение	26
1.2.1.1 Постановка задачи	26
1.2.2 Архитектура программно-аппаратного прототипа	27
1.2.2.1 Описание архитектуры	27
1.2.2.2 Конструкция зонда-манипулятора с использованием FSR датчиков	27
1.2.2.3 Конструкция зонда-манипулятора с использованием ёмкостных датчиков силы	30
1.2.3 Структура программного обеспечения	37
1.2.3.1 Документация	38
1.2.4 Испытания и эксперименты	38

1.2.4.1	Программа испытаний	38
1.2.5	Выпускные квалификационные работы	43
1.2.6	Практика студентов	43
1.2.7	Рабочая программа практики	43
1.2.8	Публикации	44
1.2.9	Результаты	44
2	Разработка концептуального прототипа для бионического костюма и системы обработки данных	45
2.1	Программно-аппаратный прототип распознавателя движения и когнитивного состояния человека с возможностями применения для цифрового производства и для умного дома	45
2.1.1	Современное состояние систем оценки вовлеченности человека	45
2.1.2	Архитектура программно-аппаратного прототипа	48
2.1.2.1	Аппаратная архитектура	49
2.1.2.2	Программная архитектура	50
2.1.3	Испытания программно-аппаратного прототипа	71
2.1.3.1	Построения модели оценки вовлеченности человека	71
2.1.3.2	Результаты испытаний программно-аппаратного прототипа	74
2.1.4	Альтернативные методы оценки вовлеченности человека	74
2.1.5	Перспективы развития программно-аппаратного прототипа	77
2.2	Программно-аппаратный прототип распознавателя приоритетных событий из множественных источников при движении робота на основе адаптивной стратегии оперативного отбора информации о текущей ситуации	77
2.2.1	Проблема распознавания приоритетных событий программно-аппаратных комплексов по данным из множественных источников.	77
2.2.2	Двух-этапный метода распознавания приоритетных событий	79
2.2.2.1	Нечеткий алгоритм интеллектуального отбора данных	80
2.2.2.2	Графовая модель как основа распознавания приоритетных состояний.	83
2.2.3	ПАК распознавания приоритетных событий	85
2.2.3.1	Концептуальная модель программно-аппаратного прототипа	85
2.2.3.2	Аппаратная часть	88
2.2.3.3	Алгоритм отбора данных	92
2.2.3.4	Алгоритм синхронизации данных	93
2.2.3.5	Алгоритм выявления приоритетных событий	93
2.2.4	Экспериментальный анализ	97
2.2.4.1	Описание экспериментов	97
2.2.4.2	Результаты анализа данных	98
2.2.5	Раздел про Лиончопинг	98

3	Разработка концептуального прототипа цифрового образа территории, создаваемого с помощью робототехнических сенсорных систем	99
3.1	Программно-аппаратный прототип распознавателя знаний о территории движения человека на основе технологий тактильного интернета и периферийных робототехнических IoT-систем на стороне мобильного пользователя	99
3.2	Программно-аппаратный прототип распознавателя текущих интересов мобильного пользователя на основе данных, получаемых от сопровождающих IoT-устройств	99
4	Интеграция образовательных программ с научной и производственной деятельностью	100
	ЗАКЛЮЧЕНИЕ	101
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	102

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

ПО — программное обеспечение.

ЦОД — центр обработки данных.

AR — дополненная реальность (от англ. Augmented Reality).

VR — виртуальная реальность (от англ. Virtual Reality).

ВВЕДЕНИЕ

1 Разработка концептуального прототипа модульной робототехнической системы

1.1 Программно-аппаратный прототип распознавателя движения физических объектов и человека с возможностями оценки качества движения

1.1.1 Введение

1.1.1.1 Постановка задачи

Целью работы является создание программно-аппаратного прототипа для распознавания движения физических объектов по данным, полученным от различных сенсорных устройств, с возможностью оценки качества движения.

В качестве физического объекта был выбран робот-манипулятор, действия которого оценивались с точки зрения точности выполненных движений.

Робот манипулятор представляет собой систему звеньев, соединенных приводами. Эта система сложна как в изготовлении, так и в эксплуатации. Последовательное расположение звеньев манипулятора приводит к тому, что каждое звено несет на себе еще и вес последующего сегмента. В результате, на каждое звено действуют большие изгибающие моменты. Для учета этих сил повышают требования к жесткости, что приводит к увеличению массы. Вторым элементом манипулятора являются сервоприводы. На работе сервопривода также сказывается вес конструкции, которую он перемещает. Таким образом, при перемещениях манипулятор испытывает влияние сил инерции, центробежных сил и сил Кориолиса. Все эти факторы усложняют управление роботом и влияют на точность его работы [1].

Точность работы робота-манипулятора определяется погрешностями позиционирования точки схвата и погрешностями угловой ориентации схвата. Если робот состоит из нескольких звеньев, то точность его работы определяется в том числе и точностью работы каждой пары звена робота и механизма привода.

Погрешности позиционирования определяются технологическими отклонениями размеров звеньев манипулятора, зазорами в кинематических парах манипулятора и механизмов приводов, деформациями (упругими и температурными) звеньев, а также погрешностями системы управления и датчиков обратной связи. Геометрическая погрешность возникает вследствие неточностей при сборке и обработке его основных узлов. Влияние геометрических ошибок сильно зависит от архитектуры, габаритов и рабочего пространства робота.

Определить общие закономерности влияния геометрических ошибок на точность работы робота-манипулятора не представляется возможным [1]. Поэтому для оценки

погрешности работы конкретного робота необходимо производить отдельное исследование [2, 3, 4, 5, 6].

Для оценки работы различных робототехнических устройств разработаны государственные стандарты. Например, ГОСТ 22267-76 "Станки металлорежущие. Схемы и способы измерений геометрических параметров" определяет следующие методы измерения геометрических параметров станков:

- а) с помощью поверочной линейки и прибора для измерения длин, отклонений от прямолинейности хода рабочего органа, несущего обрабатывающий инструмент
- б) с помощью консольной контрольной оправки и прибора для измерения длин;
- в) с помощью натянутой струны и микроскопа;
- г) с помощью гидростатического уровня;
- д) с помощью визирной трубы;
- е) с помощью оптического квантового генератора и фотоприемника;
- ж) с помощью автоколлиматора.

В работах Ивановой А.В. , Кольцова А.Г. [7] измерения точности работы робота проводятся с помощью датчика QC20-W ballbar. Современные методы обработки изображений также применяются для оценки точности работы роботов [8, 9].

Таким образом, оценка точности движений робота манипулятора является актуальной задачей. Для ее решения применяются различные подходы и инструменты.

В проекте для решения задачи разработан программно-аппаратный прототип для измерения точности работы робота-манипулятора. Распознавание выполняется по 4 типам данных наблюдения за движением: данные акселерометра, данные гироскопа, данные дальномера, видеоданные. Точность движения оценивается 3 числовыми показателями: расстояние между звеньями, угол между звеньями, расстояние от точки схвата до цели.

1.1.2 Архитектура программно-аппаратного прототипа

1.1.2.1 Описание архитектуры

Программно-аппаратный прототип распознавателя движения физических объектов состоит из следующих устройств:

- а) Ноутбук для управления манипулятором, обработки данных, поступающих с датчиков, и расчета точности движения;
- б) Система датчиков, устанавливаемых на манипулятор:
 - 1) Инерциальный датчик DFRobot 10DoF (5 штук);
 - 2) Лазерный дальномер на основе модуля VL53L0X (4 штуки);
- в) Одноплатный компьютер Raspberry Pi4;

- г) Контроллер управления сервоприводами PCA9685;
- д) Мультиплексор TCA9548A;
- е) Видеокамера.

Датчики (инерциальные датчики и дальномеры) соединены с одноплатным компьютером Raspberry Pi4 через интерфейс I2C. Контроллер ... Мультиплексор ... (Когочев А.Ю.)

Одноплатный компьютер соединен с ноутбуком посредством USB интерфейса. Видеокамера также соединена с ноутбуком через USB интерфейс.

Одноплатный компьютер соединен с сервоприводами манипулятора с помощью интерфейса ????. Система датчиков закреплена на звеньях и захвате манипулятора.

Схема взаимосвязей устройств приведена на рис. ??.

Рисунок 1 — Схема взаимосвязей элементов архитектуры

Общая схема работы ашартаной части включает следующие шаги. На ноутбуке установлено программное обеспечение, которое передает команды управления сервоприводами манипулятора на одноплатный компьютер. В процессе выполнения команд манипулятором одноплатный компьютер опрашивает инерциальные датчики, после выполнения команды одноплатный компьютер опрашивает датчики расстояния. Все собранные данные одноплатный компьютер передает на ноутбук. После выполнения команды видеокамера фиксирует расположение звеньев и захвата манипулятора. Видеоизображение также передается на ноутбук. На ноубке выполняются расчеты для определения координат звеньев манипулятора, углов между звеньями, расстояния между звеньями. На основе этих данных проводится расчет показателей точности выполненного манипулятором движения.

1.1.2.2 Описание структурных элементов

Ноутбук - Рего Г.Э.

Основные характеристики ноутбука:

- а) экран: 15.6"(1366x768)
- б) процессор: Intel Core i5 8265U (4x1.60 ГГц)
- в) память: RAM 8 ГБ, SSD 256 ГБ
- г) видеокарта: встроенная, Intel UHD Graphics 620
- д) разъемы: USB 2.0 Type A x 2, USB 3.1 Type A, выход HDMI, микрофон/наушники Combo
- е) беспроводная связь: Wi-Fi 802.11ac, Bluetooth 4.2
- ж) емкость аккумулятора: 36.7 Вт/ч
- з) время работы от аккумулятора: 9 ч

- и) операционная система: Endless OS
- к) размеры: 363.4x247.5x19.9 мм
- л) вес: 1.9 кг

Одноплатный компьютер Raspberry Pi4

Одноплатный компьютер Raspberry Pi4 построен на базе SoC Broadcom BCM2835 (рис. 2). Процессор — ARM11776JZ-F тактовой частотой 700МГц с возможностью разгона до 1 ГГц. Имеет 512Мб оперативной памяти, которая делится с графической подсистемой.

GPU компьютера аппаратно поддерживает OpenGL ES 2.0, OpenVG, MPEG-2, VC-1, кодирование и воспроизведение 1080p30 H.264/MPEG-4.

Основные характеристики компьютера:

- а) Процессор Broadcom BCM2835 (700 МГц);
- б) Оперативная память DIMM DDR3 512 Мб;
- в) Шина I2C/TWI;
- г) Шина SPI с селектором между двумя устройствами;
- д) Шины питания 3,3В, 5В;
- е) Модель видеочипа Videocore 4 видеопамять выделяется из оперативной;
- ж) 40 портов ввода-вывода общего назначения;
- з) Интерфейсы:
 - 1) UART (serial);
 - 2) HDMI;
 - 3) 3.5 мм jack (аудио);
 - 4) USB 2.0 x4;
 - 5) Ethernet (RJ-45);



Рисунок 2 — Внешний вид одноплатного компьютера Raspberry Pi4

Контроллер управления сервоприводами PCA9685

Контроллер управления сервоприводами PCA9685 – это 16-ти канальный 12-разрядный контроллер с настраиваемой частотой ШИМ-а в пределах от 24 до 1526 Гц. С помощью ШИМ контроллера можно управлять сервоприводами и другими устройствами, где в качестве управляющего сигналы применяется ШИМ сигнал (рис. 3).

Технические характеристики контроллера PCA9685:

- а) Питание модуля – 3.3 или 5 В.
- б) Питание чипа (VCC) – 2.3 ... 5.5 В.
- в) Сбой питания (VPOR) – до 2.0 В.
- г) Потребляемый ток в рабочем режиме – до 10 мА.
- д) Потребляемый ток режиме ожидания до 15.5 мкА
Ток нагрузки на выходах при VCC = 5В и выходом с открытым стоком – до 25 мА.
- е) Ток нагрузки на выходах при VCC = 5В и каскадным выходом чипа – до 10 мА.
- ж) Частота тактирования – 25 МГц.
- з) Внешний источник тактирования – от 0 до 50 МГц.
- и) Количество каналов ШИМ – 16 шт.
- к) Разрешение ШИМ – 12 бит 4096 тактов.
- л) Выходная частота ШИМ – от 24 до 1526 Гц для внутреннего генератора 25 МГц.
- м) Рабочая частота шины I2C – до 1 МГц с поддержкой 100кГц, 400кГц, 1МГц.
- н) Уровень «0» на шине I2C – от -0.5 до 0.3 VCC В.
- о) Уровень «1» на шине I2C – до 0.7 VCC до VCC+0.3 В.
- п) Рабочая температура – от -40 до 85°C.

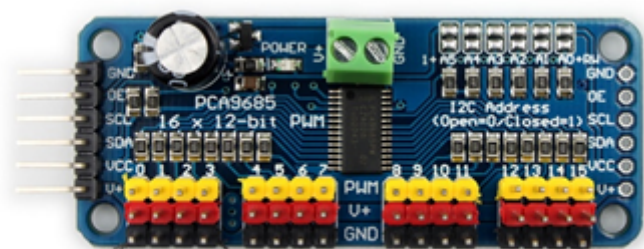


Рисунок 3 — Внешний вид контроллера управления сервоприводами PCA9685

Мультиплексор TCA9548A

TCA9548A — это переключатель I2C интерфейсов. На выходных линиях могут располагаться устройства с одинаковыми адресами, а их переключение производится управляющими командами по адресу самой микросхемы TCA9548A. Адрес TCA9548A для I2C интерфейса выбирается подачей на адресные контакты A0-A2 определённых уровней: 000,

это шестнадцатеричный адрес 0x70; 111 - 0x77. Внешний вид мультиплексора TCA9548A представлен на рис. 4.

Расширитель шины может работать как преобразователь логических уровней независимо по каждому каналу. К выводам SCL/SDA и выводам портов мультиплексора должны быть подключены подтягивающие резисторы. Например, общие выводы SCL/SDA можно подтягивать к линии питания +5 вольт, а выводы портов SC0/SD0...SC7/SD7 подтягивать к линии питания +3,3 вольт. Это необходимо при подключении трехвольтовых периферийных модулей (Raspberry Pi4) к пятивольтовой плате Arduino ?????? Причем не обязательно, чтобы рабочее напряжение всех портов расширителя было одинаковое.

Для подключения микросхемы TCA9548A на печатную плату модуля используется 24 контакта. Более подробное описание контактов микросхемы TCA9548A представлено в таблице 1.

Таблица 1 — Описание контактов мультиплексора TCA9548A

Номер вывода	Имя сигнала	Описание сигнала
1	Vin	Питание, подключенное к основному источнику питания, вывод питания, +5 В
2	GND	Земля, подключённым к основному заземлению
3	SCL	тактирующая линия шины I2C для связи мультиплексора с ведущим устройством
4	SDA	линия данных шины I2C для связи мультиплексора с ведущим устройством
5	RST	линия сброса, для сброса
6	A0	бит адреса мультиплексора; если соединить с Vin, то адрес увеличится на 1
7	A1	бит адреса мультиплексора; если соединить с Vin, то адрес увеличится на 2
8	A2	бит адреса мультиплексора; если соединить с Vin, то адрес увеличится на 4

Продолжение таблицы 1

Номер вывода	Имя сигнала	Описание сигнала
10, 12, 14, 16, 18, 20, 22, 24	SC0 ... SC7	линии SCL для периферийных устройств
9, 11, 13, 15, 17, 19, 21, 23	SD0 ... SD7	линии SDA для периферийных устройств

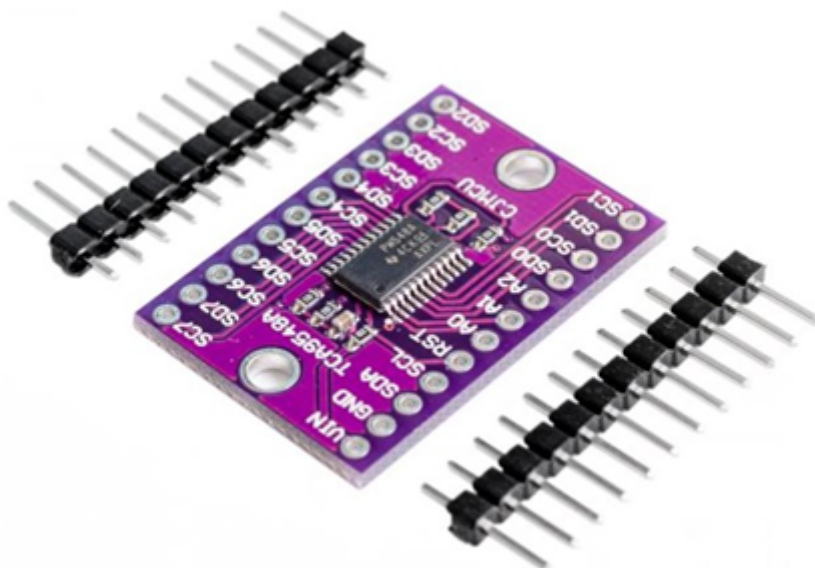


Рисунок 4 — Внешний вид мультиплексора TCA9548A

Датчики расстояния - Когочев А.Ю.

Инерциальные навигационные датчики DFRobot 10DoF

Инерциальный измерительный блок DFRobot 10DoF представляет собой электронное устройство, которое измеряет и сообщает о скорости объекта, ориентации и гравитационных силах, используя комбинацию акселерометров, гироскопов и магнитометров. Датчик объединяет акселерометр ADXL345, магнитометр HMC5883L, гироскоп ITG-3205 и датчик барометрического давления BMP280. Он подходит для большинства систем управления из-за небольшого размера. В него встроен малошумящий стабилизатор LDO для подачи входной мощности в широком диапазоне. Работает с входным напряжением 3-8 В. Датчик 10 DOF напрямую совместим с платами Arduino и Raspberry Pi. На рис. 5 представлен внешний вид инерциального измерительного блока.

Видеокамера - Баженов Н.

1.1.3 Структура программного обеспечения

Программное обеспечение программно-аппаратного прототипа распознавателя движения включает следующие элементы:

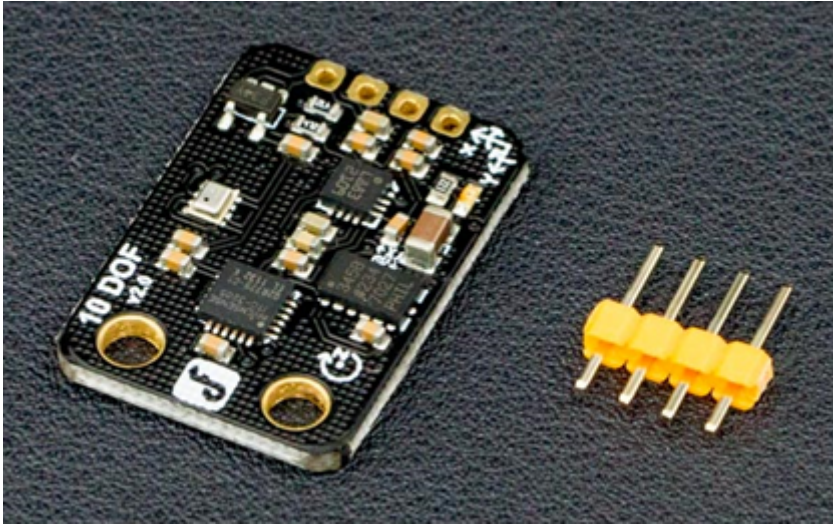


Рисунок 5 — Внешний вид датчика DFRobot 10DoF

- а) Программа управления манипулятором для Raspberry Pi
- б) Программа обработки данных
- в) Программа расчета точности
- г) Программа moovIt
- д) Программа Gazebo
- е) Программа для Raspberry Pi
- ж) Операционная система для роботов (ROS)
- з) Rviz — программа для визуализации модели и ее движений
- и) Описание цифровой модели

Схема взаимосвязей программного обеспечения приведена на рис. ??.

Рисунок 6 — Схема взаимосвязей программного обеспечения

Программное обеспечение

1.1.3.1 Цифровая модель - Рего Г.

Цифровая модель была разработана с помощью таких программных средств как Gazebo, Robot operating system (ROS) и MoveIt. Основная цель существования цифровой модели заключается в том, чтобы симулировать выполнение каких-то действий реальным роботом. Цифровая модель соответствует основным характеристикам реального прототипа: длины звеньев, их сочленения, возможные положения звеньев совпадают со схожими характеристиками реального робота. Gazebo используется как средство визуализации. Описание робота представлено на языке Unified Robot Description Format (URDF), который является XML-подобным языком. ROS используется для выполнения действий в

симуляции и передаче координат отдельных звеньев, захвата и других объектов. MoveIt используется как плагин для задания координат как отдельным звеньям, так и захвату.

1.1.3.2 Программа управления манипулятором

Назначение программы: управление положением звеньев робота и сбор данных с датчиков. Программа устанавливается на одноплатный компьютер Raspberry Pi 4. Программа получает данные от (синхронизатора робота и цифровой модели - может так называть?), затем отправляет команды сервоприводам, собирает данные с датчиков и отправляет синхронизатору.

Входные данные: Набор пар: Номер звена и угол поворота.

Выходные данные: Данные с датчиков; отчет об успешности/неуспешности выполнения движения.

Алгоритм работы программы:

После запуска программы происходит инициализация массивов: `servo_min` – хранит минимально допустимые углы поворота звеньев, `servo_max` – хранит максимально допустимые углы поворота звеньев, `servo_cur` – текущие значения углов поворота звеньев. Значения минимально и максимально допустимых углов были высчитаны эмпирически на основе экспериментов. Текущие значения при инициализации массива `servo_cur` устанавливаются равными 1000.

Инициализируются переменные `HOST` и `PORT` для установки клиент-серверного соединения через сокет `s`.

Инициализируются словари `data_from_accel`, `data_from_gyro`, `data_from_laser`, в которых хранятся данные, получаемые от датчиков.

Инициализируется переменная `start_time`, которая хранит начальное время, от которого будут отсчитываться временные отсчеты замеров данных с датчиков.

После подключения сокета `s`, программа переходит в режим прослушивания с помощью функции `s.listen()` и ждет подключения программы обработки данных.

После подключения Программы обработки данных звенья манипулятора переводятся в исходное положение.

После этого запускается бесконечный цикл, который прерывается только тогда, когда Программа обработки данных оборвет соединение.

В цикле:

Сначала происходит опрос всех датчиков и полученные данные накапливаются в словари `data_from_accel`, `data_from_gyro`, `data_from_laser`.

Далее программа принимает команды от программы обработки данных (пары номер звена и угол поворота). Выполняется проверка корректности переданного угла, а затем, при успешном завершении проверки, соответствующему сервоприводу передается команда о повороте звена на заданный угол

После успешного выполнения движения, происходит сбор данных с каждого датчика.

Начинается новая итерация цикла.

После того, как соединение было оборвано, программа отправляет запрос об остановке работы.

В случае получения положительного ответа, программа записывает все накопленные данные пересылаются в синхронизатор и записываются в лог-файл на ноутбуке.

На этом работа программы заканчивается.

Программа разработана на языке программирования Python. В программе использовались следующие модули:

- а) Модуль `socket` – предоставляет функции для настройки клиент-серверного соединения по типу сокета.
- б) Модуль `struct` – предоставляет функции для упаковки и распаковки данных в двоичном формате.
- в) Модуль `servoDriver`.
- г) Модуль `imuDriver_Multi`.
- д) Модуль `gyroDriver_Multi`.
- е) Модуль `laserDriver_Multi`.
- ж) Модуль `time` – предоставляет функции работы со временем.
- з) Модуль `logging` – модуль для сбора и сохранения данных в файл.
- и) Модуль `datetime` – предоставляет функции работы со временем и датами.

1.1.3.3 Синхронизатор положения цифровой модели и реального робота

Назначение программы: передача команд управления положением звеньев робота от цифровой модели Программе управления манипулятором, получение данных показаний датчиков от программы управления манипулятором. Программа устанавливается на компьютер с операционной системой Linux.

Основной задачей программы является получение углов звеньев робота из симуляции, их преобразование в значения углов для звеньев реального робота и передача преобразованных углов и номеров звеньев серверу.

Входные данные: Набор пар: Номер звена и угол поворота, получаемые от цифровой модели.

Выходные данные: Набор пар: Номер звена и угол поворота.

Алгоритм работы программы:

Инициализируется экземпляр класса `MoveGroupPythonInterfaceTutorial`. Основная задача класса заключается в том, чтобы осуществлять привязку своих экземпляров к существующим роботам, сцене и запущенным плагинам. Взаимодействие со звеньями осу-

ществляется с помощью обращения к группе, которой принадлежит конкретное звено. Есть две группы звеньев: `robot_arm`, которая отвечает за захват и имеет два звена (левая и правая клешня), и `robot_hand`, которая отвечает за все остальные звенья.

Инициализируются массивы значений, которые содержат обобщенные координаты каждого звена в симуляции (`degrees_from_simulation`) и реального робота (`degrees_real_robot`). При инициализации переменным в массиве присваивается значение 0.

Осуществляется подключение к Raspberry Pi с помощью сокета. В переменную `HOST` записывается ip-адрес Raspberry Pi, в переменную `PORT` номер порта, к которому подключается программа-клиент. Инициализировав переменные, программа создает объект класса сокет, с помощью стандартного модуля Python `socket`, используя встроенную функцию `socket`, и осуществляет подключение к Программе управления манипулятором. В случае, если осуществить подключение по каким-то причинам не удастся, программа прекращает свое выполнение.

После успешного подключения начинается основная стадия работы программы.

Запускается цикл `while`, условием которого является проверка значения флага остановки программы.

Если флаг остановки работы программы равен нулю, выполнение работы продолжается, иначе прекращается.

На каждой итерации цикла происходит считывание текущих целевых координат с помощью функции `???`. В результате массивы `result` (для группы `robot_hand`) и `result2` (для группы `robot_arm`) заполняются целевыми углами симуляции в радианах.

В массив `degrees_from_simulation` записываются углы положения звеньев, с помощью формулы: $\text{соответствующий_угол_в_радианах} * 180 / \text{число } \Pi$. Результат округляется до целого по правилам математического округления с помощью функции `round()`.

Далее для каждого звена проверяется условие равенства угла из симуляции углу реального робота. Если в конкретном движении звено не участвует, то на него команда не подается.

Если же угол из симуляции не совпадает с углом реального робота, то выполняется ряд команд. Переменная `dev_num`, которая отвечает за номер сервопривода, на который будет подана команда, устанавливается в соответствующее для данного звена значение. Далее переменная `data_to_serv` устанавливается в значение, которое будет передано на соответствующий сервопривод. Для каждого звена эмпирически выведено отклонение угла симуляции от реального угла. Например, для сервопривода, который отвечает за поворот основания, формула принимает следующий вид: $\text{data_to_serv} = 60 + \text{degrees_from_simulation}[0]$, где `data_to_serv` – угол, который будет передан как целевой соответствующему сервоприводу, 60 – отклонение угла в симуляции от реальных значений, `degrees_from_simulation[0]` – угол из симуляции.

После подготовки данных, значение переменной угла реального робота соответствующего звена становится равным значению угла симуляции.

Далее данные, предназначенные для отправки, структурируются с помощью функции `struct.pack()` и отправляются в Программу управления манипулятором с помощью функции `send()`.

После получения ответа с помощью функции `recv()`, в консоль выводится сообщение об ответе сервера.

После того как все звенья проверены и движения выполнены, пользователю предлагается установить флаг остановки программы с помощью ввода из консоли.

Программа разработана на языке программирования Python. В программе использовались следующие модули:

- а) Модуль `socket` – предоставляет функции для настройки клиент-серверного соединения по типу сокета.
- б) Модуль `struct` – предоставляет функции для упаковки и распаковки данных в двоичном формате.
- в) Модуль `time` – предоставляет функции работы со временем.
- г) Модуль `six` – помогает писать код, совместимый как с Python 2, так и с Python 3.
- д) Модуль `sys` – предоставляет набор функций, которые дают информацию о том, как интерпретатор Python взаимодействует с операционной системой.
- е) Модуль `rospy` – предоставляет API для работы с ROS с помощью Python.
- ж) Модуль `moveit_commander` – предоставляет оболочки для функций MoveIt, связанных с движением объектов.
- з) Модуль `moveit_msgs.msg` – предоставляет сообщения ROS, специфичные для MoveIt.
- и) Модуль `geometry_msgs.msg` – предоставляет функции сообщений для общих геометрических примитивов, таких как точки, векторы и позы.
- к) Модуль `math` – предоставляет математические функции и константы.
- л) Модуль `std_msgs.msg` – содержит общие типы сообщений, представляющие примитивные типы данных и другие базовые конструкции сообщений, такие как множественные массивы.
- м) Модуль `moveit_commander.conversions`.

1.1.3.4 Программа расчета точности

1.1.3.5 Документация

Для программного обеспечения разработан комплект документов в соответствии с ГОСТ 19.102-77, включающий:

- а) Описание программы на ??? листах;
- б) Спецификация на ??? листах;
- в) Текст программы на ??? листах;
- г) Программа и методика испытаний на ??? листах;

1.1.4 Испытания и эксперименты

1.1.4.1 Описание стенда для испытаний

Для проведения испытаний был использован манипулятор фирмы Abb Industrial Robot (Китай) с шестью степенями свободы (рис. 7).

Манипулятор изготовлен из алюминия, и представляется достаточно надёжной конструкцией. В таблице 2 представлены технические характеристики манипулятора.

Таблица 2 — Технические характеристики манипулятора Abb Industrial Robot

Характеристика	Значение
Количество степеней свободы	6 степеней
Количество сервоприводов	6 шт.
Максимальный вылет стрелы	485 мм
Масса конструкции	1.5 кг
Напряжение питания сервоприводов	4.8 .. 7.2 В

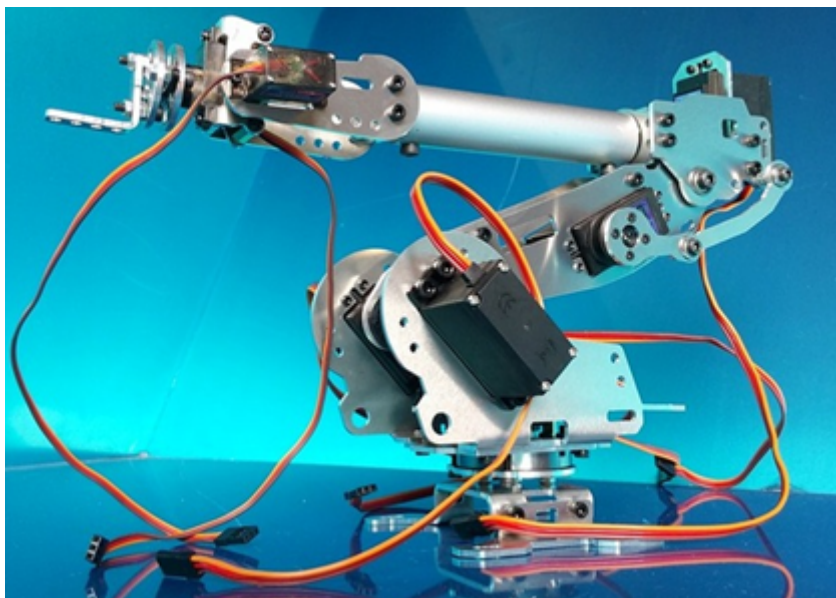


Рисунок 7 — Внешний вид манипулятора фирмы Abb Industrial Robot

На манипуляторе были установлены датчики. Схема расположения датчиков представлена на рис. ???. Инерциальные датчики расположены на звене 2, на ... Датчики рас-

стояния установлены на звене и захвате: один по направлению на цель, два по бокам захвата.

Рисунок 8 — Схема расположения датчиков на манипуляторе

На манипуляторе закреплен Одноплатный компьютер Raspberry Pi4. Видеокамера расположена на расстоянии

1.1.4.2 Программа испытаний

Для проведения испытаний была разработана Программа и методика испытаний. Целью испытания была проверка работоспособности программно-аппаратного прототипа распознавателя движения физических объектов. Задачи:

- а) Проверить выполнение команд, поступающих от цифровой модели;
- б) Проверить получение данных от Raspberry Pi;
- в) Проверить правильность расчетов оценок точности.

Программно-аппаратный прототип распознавателя движения физических объектов успешно прошел все испытания.

1.1.4.3 Эксперименты

С помощью программно-аппаратного прототипа распознавателя движения физических объектов были проведены исследования точности работы манипулятора фирмы ABB Industrial Robot. В каждом испытании робот манипулятор с помощью цифровой модели приводился в начальное состояние, затем с помощью команд переходил в конечное состояние, в котором фиксировались значения датчиков. Описание экспериментов представлено в таблице 3. Результаты экспериментов представлены в таблице 4.

Таблица 3 — Описание экспериментов

Номер испытания	Начальное состояние	Конечное состояние	Датчики
1	Звено2 = 0 (рис. 9)	Звено2 = -40	Акселерометр2 и все последующие
2	Звено3 = 0	Звено2 = 40	Акселерометр2 и все последующие
3	Звено4 = 0	Звено4 = 30	Акселерометр2 и все последующие

Продолжение таблицы 2

Номер испытания	Начальное состояние	Конечное состояние	Датчики
4	Звено4 = 30	Звено4 = -30	Акселерометр3 и все последующие
5	Звено4 = -30	Звено2 = 0	Акселерометр2
6	Звено5 = 0	Звено5 = 30	Акселерометр3
7	Звено5 = 30	Звено5 = -30	Акселерометр1
8	Звено5 = -30	Звено5 = 0	Акселерометр2
9	Звено1 = 0	Звено1 = -20	Все гироскопы
10	Звено1 = -20	Звено1 = 20	Все гироскопы
11	Звено1 = 20	Звено1 = 0	Все гироскопы
12	Захват = 0	Захват = 40	Все дальнометры
13	Захват = 40	Захват = 0	Все дальнометры
14	Одновременное движение Звено2 = -40, Звено3 = 40	Звено2 = -60, Звено3 = 60	Акселерометр2 и все последующие
15	Одновременное движение Звено2 = -60, Звено3 = 60	Звено2 = -40, Звено3 = 40	Акселерометр2 и все последующие

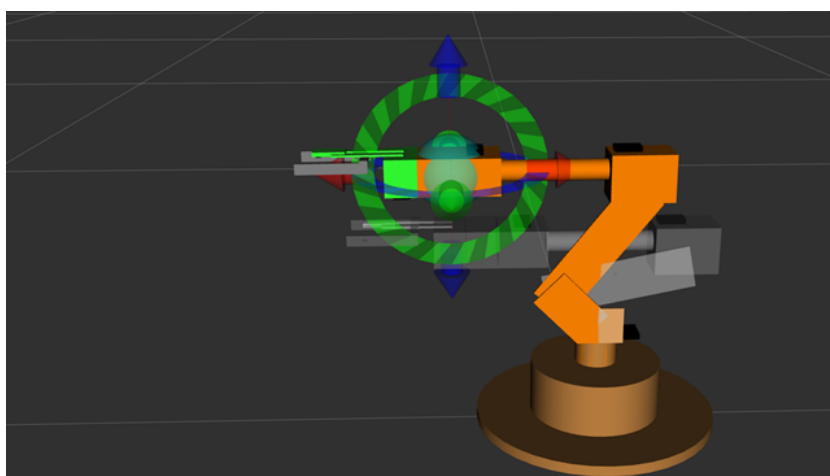


Рисунок 9 — Начальное положение манипулятора в испытании 1

Таблица 4 — Результаты экспериментов

Номер испытания	Планируемое значение датчика	Реальные показания датчика	Оценка точности движения (Accuracy)
1	Акселерометр1 = 0	Акселерометр1 = -1	0,9

Для проведения дальнейших экспериментов был приобретен робот-манипулятор Dobot Magician (рис. 10). Многофункциональный робот Dobot Magician представляет собой роботизированную "руку" которая может печатать 3D-фигуры, выполнять лазерную гравировку, воспроизводить на бумаге различные чертежи, схемы и рисунки, а также манипулировать предметами. Это эффективное учебное средство позволяет охватывать целый ряд учебных областей — информатику, технологию, черчение, ИЗО и др. Управлять многофункциональным роботом можно с помощью компьютера, смартфона или планшета по Bluetooth, WIFI или используя джойстик. Визуальный интерфейс программного обеспечения DobotStudio (Windows, Linux CNC, Mac), работающий по принципу plug-and-play (нажать и перетащить), настолько прост, что справиться с управлением робота может даже начинающий пользователь. Небольшие размеры основания (158x158 мм) позволяют разместить робота на любой рабочей поверхности и осуществить, например, 3D-печать на обычном рабочем столе. Сменные инструментальные блоки многофункционального робота (3D-принтер, лазер, держатель пера, вакуумная присоска, пневматический захват) меняются достаточно быстро, а точность позиционирования пера доходит до 0,2 мм. Dobot Magician может использоваться в качестве станка для работы прототипа при выполнении нескольких условий: (условия станут понятны, когда попробуем его в деле).

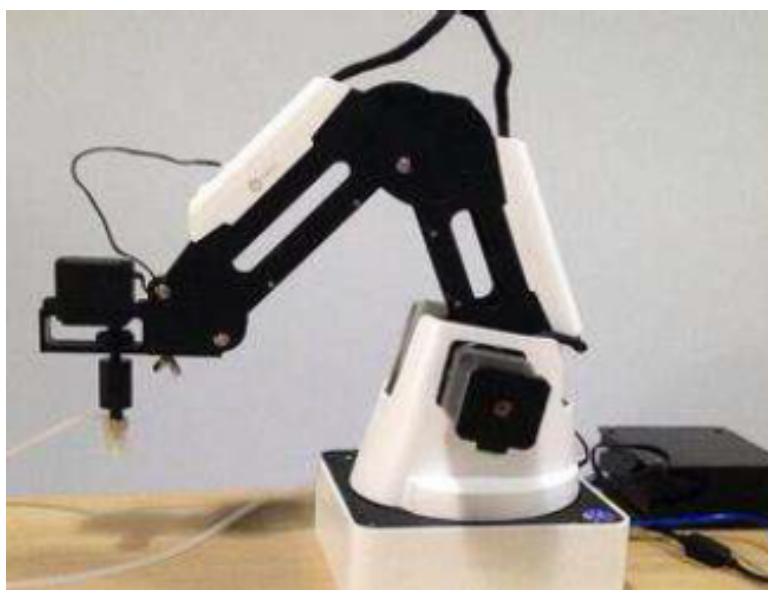


Рисунок 10 — Внешний вид манипулятора Dobot Magician

1.1.5 Выпускные квалификационные работы

В рамках разработки программно-аппаратного прототипа распознавателя движения физических объектов ведется подготовка выпускной квалификационной работы Тарицыной Анастасии Сергеевны по направлению подготовки 01.04.02 Прикладная математика и информатика, магистерская программа "Математическое моделирование и информационно-коммуникационные технологии" в Институте математики и информационных технологий.

В выпускной работе Тарицына А. С. разрабатывала цифровая модель робота-манипулятора и программное обеспечение для определения идеальных координат звеньев робота.

1.1.6 Практика студентов

В рамках разработки программно-аппаратного прототипа распознавателя движения физических объектов магистрант Максименко Леонид Михайлович проходил практику по научно-исследовательской работе по направлению подготовки 09.04.02 Информационные системы и технологии магистерская программа "Управление данными" в Институте математики и информационных технологий.

В рамках практики Максименко Л. М. разрабатывал программное обеспечение для автоматического построения цифровой модели по видеозаписи движения робота.

1.1.7 Рабочая программа практики

В рамках разработки программно-аппаратного прототипа распознавателя движения физических объектов была разработана рабочая программа практики "Учебная технологическая практика (программный проект)" по направлению подготовки 01.04.02 Прикладная математика и информатика, магистерская программа "Интеллектуальные интернет-технологии" в Институте математики и информационных технологий.

Рабочая программа предполагает прохождение практики на базе Центра искусственного интеллекта ПетрГУ. В рамках этой практики планируется групповая работа магистрантов над проектами по робототехнике. Один из проектов связан с задачей определения точности движения манипулятора на базе разработанного программно-аппаратного прототипа распознавателя движения физических объектов.

1.1.8 Публикации

В рамках проекта были подготовлены следующие публикации:

- а) Г.Э. Рего Программная архитектура для обеспечения взаимодействия робота-манипулятора и его цифрового двойника. // Цифровые технологии в образова-

нии, науке, обществе [Электронный ресурс] : материалы XV Всероссийской научно-практ. конф. 30 ноября – 03 декабря 2021 года, Петрозаводск, с. 101-102. с. 99-100.

- б) Г. Э. Рего, А. С. Тарицына, Л. В. Щеголева Цифровой двойник робота-манипулятора как инструмент измерения точности его работы // Цифровые технологии в образовании, науке, обществе [Электронный ресурс] : материалы XV Всероссийской научно-практ. конф. 30 ноября – 03 декабря 2021 года, Петрозаводск, с. 101-102.

На конференции The 30th Conference of Open Innovations Association FRUCT Oulu, Finland 27-29 October 2021 была представлена презентация Demo: The digital twin of a robotic arm, by Anastasia Taritsyna, Grigorij Rego and Lyudmila Schegoleva.

1.1.9 Результаты

Был разработан программно-аппаратный прототип распознавателя движения физических объектов с оценкой точности движения. Аппаратная часть прототипа включает ноутбук, систему датчиков, одноплатный компьютер Raspberry Pi4, контроллер управления сервоприводами, мультиплексор и видеокамеру. Программная часть включает сервер управления манипулятором, программу moovIt, программа Gazebo, программа для Raspberry Pi. Прототип предназначен для расчета точности движений робота манипулятора. Прототип использует 4 типа 4 типа данных наблюдения за движением. Точность движения определяется 3 числовыми показателями. На прототип разработана программная документация в соответствии с ГОСТ 19.102-77.

Проведены испытания прототипа согласно разработанной Программе и методике испытаний. Все испатния были пройдены успешно.

С использованием прототипа были проведены эксперименты по оценке точности манипуляторов Abb Industrial Robot и Dobot Magician.

В рамках разработки прототипа один магистрант проходил практику, одна магистрантка готовит выпускную квалификационную работу.

Задачи с использованием прототипа включены в практику "Учебная технологическая практика (программный проект)" по направлению подготовки 01.04.02 Прикладная математика и информатика.

Дальнейшее развитие программно-аппаратного прототипа планируется в направлении совершенствования возможностей по оценке качества движения, например, деликатности захвата, надежности захвата.

1.2 Программно-аппаратный прототип распознавателя действий человека или других объектов по отношению к человеку или друг к другу

1.2.1 Введение

1.2.1.1 Постановка задачи

Целью работы является создание программно-аппаратного прототипа для распознавания действий физического объекта по отношению к другому физическому объекту.

В качестве физического объекта был выбран робот-манипулятор, оснащенный зондом-манипулятором для неразрушающего анализа возможности перемещения препятствия.

Актуальной задачей является разработка устройств, позволяющих неразрушающим контактным способом анализировать возможность перемещения (сдвига, опрокидывания и пр.) препятствий (предметов) на пути следования автономного робототехнического комплекса. Такое устройство позволит расширить функциональные возможности автономного робототехнического комплекса и повысить его адаптируемость к окружающей среде, увеличить степень автономности комплекса. Органы технического зрения, часто применяемые в настоящее время, способны только фиксировать наличие препятствия на пути. Однако зачастую препятствия могут быть либо сдвинуты, либо не представляют сложности для преодоления комплексом – например, трава и иные плотно расположенные лёгкие предметы.

Для подобного устройства по аналогии с человеческим телом очевидным является использование тактильных датчиков и датчиков усилия сжатия для фиксации момента касания предмета и осуществления силового воздействия на предмет. Поэтому, в первую очередь был проведен анализ датчиков величины нормальной силы и давления, предлагаемых на рынке. В качестве наиболее подходящих сенсоров для задачи проекта (габариты, вес, простота съема сигнала, стоимость и т.п.) выбраны FSR датчики – force-sensing resistor [10]. Альтернативным решением, предложенным в работе, является использование ёмкостных датчиков силы оригинальной конструкции, разработанных в ПетрГУ.

В проекте для решения задачи разработан программно-аппаратный прототип для анализа возможности перемещения (сдвига, опрокидывания и пр.) препятствий (предметов) на пути следования автономного робототехнического комплекса. Распознавание выполняется по 3 типам данных наблюдения за движением: данные акселерометра, данные гироскопа, данные зонда. Распознается 5 видов действий: неподвижный объект, сдвигаемый объект с постоянным усилием, толкание зондом-манипулятором объекта с постоянным усилием и последующим опрокидыванием объекта, толкание зондом-манипулятором более массивного объекта с постоянным усилием и последующим опрокидыванием

объекта, упор зонда-манипулятора в неподвижный предмет с постоянным (пределным) усилием.

1.2.2 Архитектура программно-аппаратного прототипа

1.2.2.1 Описание архитектуры

Программно-аппаратный прототип распознавателя действий физического объекта состоит из следующих устройств:

- а) Зонд-манипулятор;

1.2.2.2 Конструкция зонда-манипулятора с использованием FSR датчиков

Резистивные FSR датчики состоят из двух слоев, которые разделены специальной прокладкой (spacer) (рис. 11). Чувствительная пленка состоит как из электропроводящих, так и из непроводящих субмикромикронных частиц, взвешенных в матрице. Приложение силы к поверхности чувствительной пленки заставляет частицы касаться проводящих электродов, изменяя сопротивление пленки между нанесенными электродами, выполненными, как правило, в виде встречно-штыревой структуры. Как и все резистивные датчики, FSR требуют относительно простой электрической схемы снятия измерительной информации. По сравнению с другими датчиками силы преимущества FSR заключаются в их размере (обычно толщина менее 0.5 мм), низкой стоимости и хорошей ударопрочности. Недостатком является их низкая точность: результаты измерений могут отличаться на 10% и более, что требует индивидуальной калибровки каждого экземпляра.

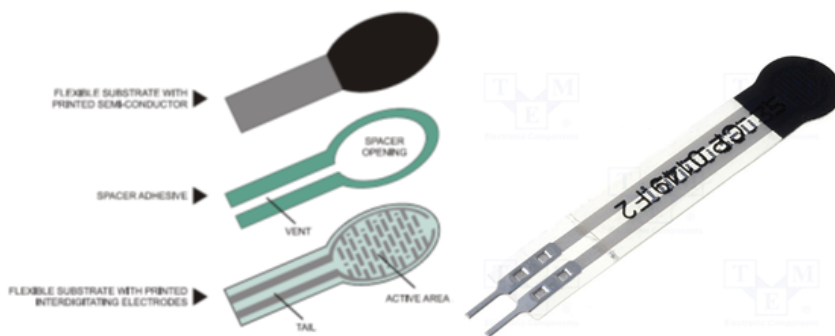


Рисунок 11 — Схема FSR датчика (слева) и фото датчика CP1-149NS (справа) компании IEE (International Electronics & Engineering) [11]

Существует два основных механизма изменения сопротивления FSR датчика: перколяция и квантовое туннелирование [10, 11, ?]. Хотя оба явления фактически происходят в проводящем полимере одновременно, одно явление преобладает над другим в зависимости от концентрации частиц. На сегодняшний день не существует всеобъемлющей модели, способной предсказать все нелинейные явления в FSR сенсорах. На рис. 12 приведена приблизительная зависимость сопротивления датчика от приложенной силы (массы груза).

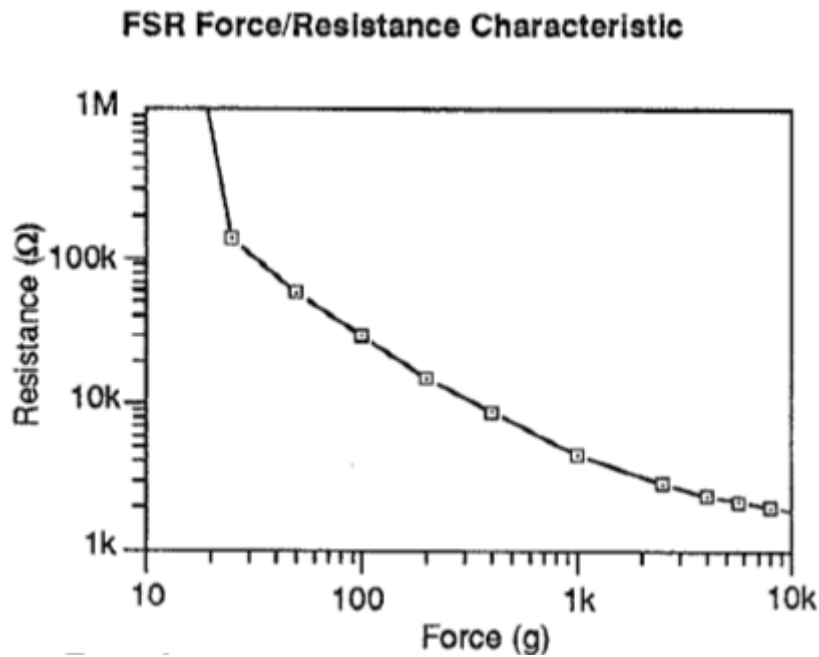


Рисунок 12 — Зависимость сопротивления датчика от приложенной силы [10]

В качестве используемых FSR датчиков были выбраны CP1-149NS компании IEE International Electronics & Engineering срабатывания с рабочим диапазоном $0.2 \div 100\text{H}$. Типовая схема измерений представлена на рис. 13. Датчик включается последовательно с постоянным резистором известного номинала, с которого снимается сигнал в виде напряжения (делитель напряжения). Постоянный резистор также выполняет роль токоограничивающего элемента, предохраняя датчик от повреждения вследствие джоулева нагрева при уменьшении сопротивления под действием приложенной силы. Существуют также иные способы снятия показаний: включение датчика в преобразователь ток-напряжение (эта схема даёт большую линейность), включение датчика в RC-цепочку, для которой измеряется постоянная времени и пр. [10, 11].

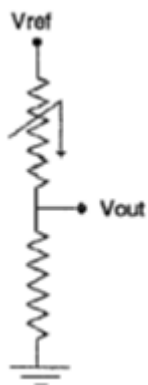


Рисунок 13 — Схема снятия показаний с FSR датчика [11]

На рис. 14 представлена схема разработанного устройства зонда-манипулятора.

Устройство состоит из следующих элементов, показанных на чертеже (рис. 14).

Зонда-манипулятора 1 цилиндрической формы, на торцевой и боковой поверхности ко-

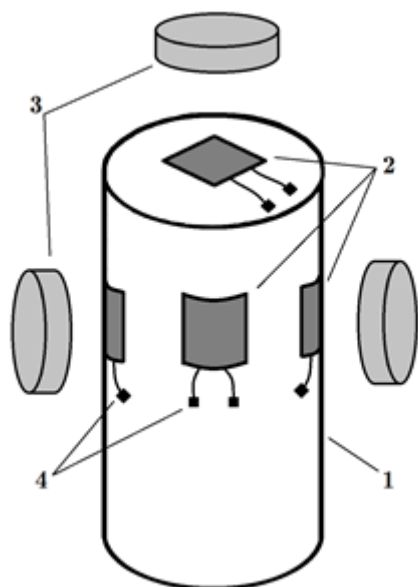


Рисунок 14 — Схема разработанного устройства (1 – тело зонда-манипулятора, 2 – датчики силы, 3 – мягкие накладки, 4 – выводы датчиков)

торого закреплены датчики нормальной составляющей силы 2 (силы нажима). Датчики накрываются мягкими накладками 3 из силикона, резины или иного материала для обеспечения контакта с препятствием в области расположения датчиков, а также для более равномерного распределения силы по площади датчика. Выводы датчиков 4 подключаются к измерительному интерфейсу системы автоматического управления контактным способом, либо к системе, способной передавать измерительную информацию с них беспроводным способом.

Устройство работает следующим образом. Роботизированный комплекс при обнаружении органами технического зрения препятствия приближается к нему, затем с помощью зонда-манипулятора 1, либо жёстко закрепленного на корпусе комплекса, либо установленного в специальном механическом манипуляторе ("механическая рука" и пр.), осуществляет попытку механического воздействия на препятствие. Воздействие может, например, заключаться в попытке отодвинуть препятствие с помощью органов передвижения комплекса (двигателя) или попытке сдвинуть препятствие толканием зонда-манипулятором и пр. При этом возникает механический контакт между препятствием и зондом-манипулятором, сила воздействия комплекса на препятствие порождает силу противодействия, которая воздействует на датчики нормальной составляющей силы 2. Следует отметить, что деликатное перемещение предметов может быть осуществлено с помощью механических манипуляторов типа "рука" и т.п., однако их использование невозможно в случае относительно легких, но габаритных предметов, плотно расположенных легких объектов и т.п.

В качестве датчиков силы могут быть использованы любые подходящие датчики, использующие различные физические принципы. Для увеличения вероятности воздействия силы именно на области расположения датчиков используются мягкие накладки 3

из силикона, резины или иного подобного материала. Накладки, деформируясь, передают силу на датчики, одновременно распределяя её на большую площадь. Это актуально в случае препятствий с выраженными резко выступающими элементами. При этом используются несколько датчиков – один расположенный на торцевой поверхности зонда-манипулятора, остальные – на его боковой поверхности. Использование датчиков на боковой поверхности позволяет осуществлять зондирование не только с помощью поступательного движения, но и с помощью вращательного, будь то поворот зонда с помощью механической руки или иного органа или поворот корпуса всего комплекса в целом. Величина нормальной силы, действующая на датчик, преобразуется в электрический сигнал и поступает на интерфейс системы автоматического управления комплексом и анализируется. По характеру изменения сигнала с активного датчика, который взаимодействует с препятствием, система управления определяет характер взаимодействия с предметом (статичный, сдвигаемый и т.п.). Дополнительно при этом могут использоваться данные с других инструментов комплекса – акселерометров, фиксирующих перемещение комплекса в целом или механического манипулятора, в котором закреплено устройство, органов технического зрения, отслеживающих изменение положение предмета и т.п. Образцы устройства были изготовлены следующим образом. В качестве зонда-манипулятора использовались цилиндрические стержни из диэлектрического материала (полиэтилен и т.п.), на торцевую и боковую поверхности которых наклеивались датчики силы нажатия. Один из образцов зонда был изготовлен на основе резистивных датчиков FSR (Force-sensing resistor). Для снятия сигнала с датчиков использовалась простейшая электрическая цепь, состоящая из последовательно включенного датчика и постоянного резистора номиналом 10 кОм. При увеличении силы нажатия сопротивление датчика уменьшается, напряжение на резисторе при этом увеличивается. Сигнал напряжения может быть оцифрован с помощью АЦП.

1.2.2.3 Конструкция зонда-манипулятора с использованием ёмкостных датчиков силы

Предложены и разработаны оригинальные ёмкостные датчики силы и давления на основе оксидной плёнки алюминия. Образцы датчиков были изготовлены из алюминиевой фольги с размерами порядка 7x15 мм² в лабораторных условиях. Предварительно обе полосы анодировались в 20 %-м растворе серной кислоты при плотности тока 10-15 мА/см². В качестве источника питания использовался Keithley Source-meter 2410, работающий в режиме стабилизации тока. На рис. 15 приведена общая схема процесса электролитического анодирования.

На рис. 16 приведена фото процесса анодирования.

Таким образом, на пластинках образовывался слой оксида алюминия, предположительно имеющий пористую структуру. Каждая обкладка образованного из двух пластинок конденсатора, имела электрический вывод, соединенный с небольшим участком чистого,

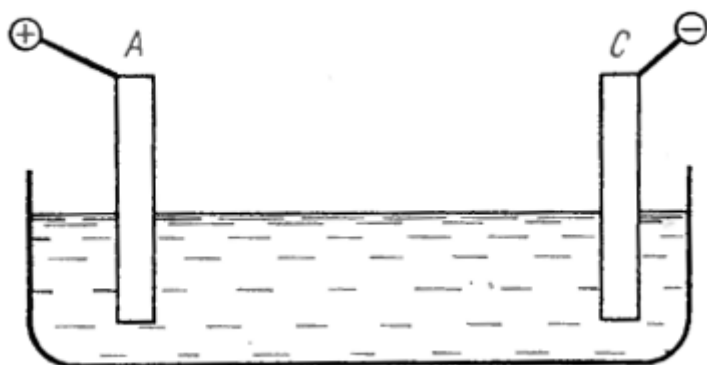


Рисунок 15 — Схема процесса анодирования: А – анод (алюминий), С – катод (свинец) [?].

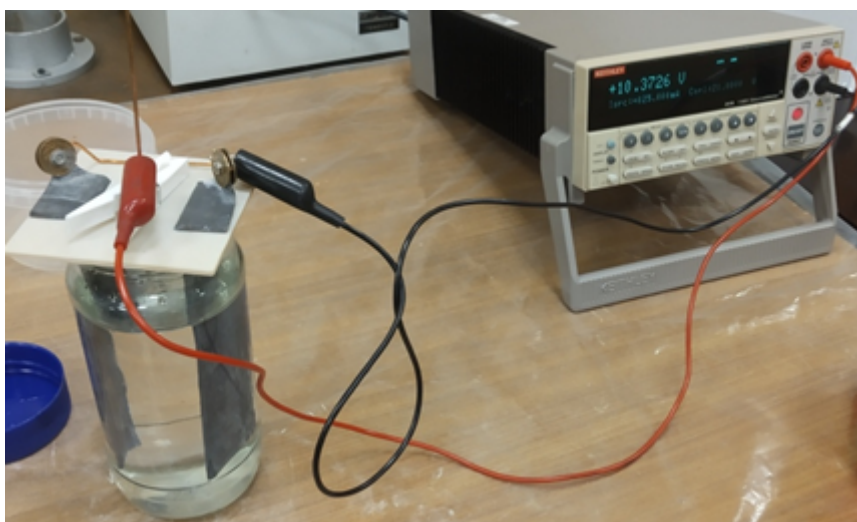


Рисунок 16 — Фото процесса анодирования

не покрытого оксидом, алюминия с помощью пайки. Полученный конденсатор запечатывается в пакет из полимерной плёнки (состав многослойной комбинации на основе PET, EVA, LDPE). При сжатии пакета нормальной силой электрическая ёмкость увеличивается. Получена относительно неплохая повторяемость параметров датчиков при выдерживании условий их изготовления. Схема датчика представлена на рис. 17, фото вариантов исполнения на рис. 18.

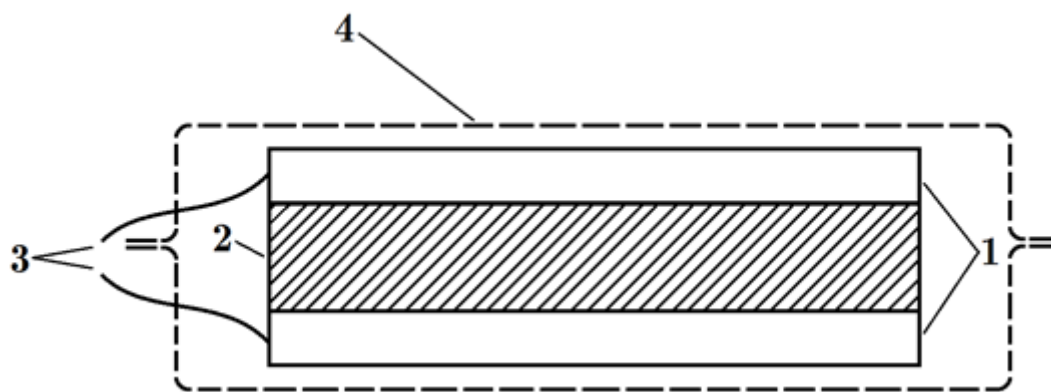


Рисунок 17 — Схема ёмкостного датчика силы: 1 – алюминиевые пластины, 2 – слой оксида, 3 – выводы, 4 – корпус из плотной полимерной плёнки

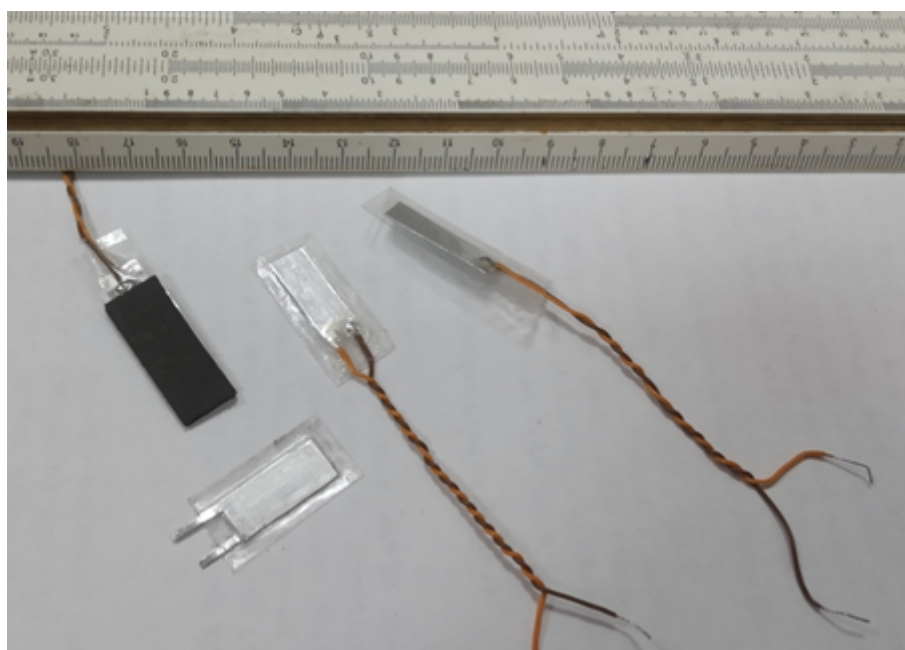


Рисунок 18 — Фото вариантов исполнения ёмкостных датчиков силы

На рис. 19 приведена зависимость величины ёмкости изготовленного датчика от приложенной силы F . Измерения ёмкости проводились LCR-метром на частоте 100 кГц с разрешением по электрической ёмкости 0.01 пФ. При начальной величине ёмкости около 135 пФ её значение достигает ~ 210 пФ при приложении силы $F=10$ Н. Чувствительность не является постоянной, для малых значений приложенной силы её величина больше, что

позволяет более точно измерять слабые силы до ~ 1 Н. После снятия нагрузки датчик восстанавливает начальную ёмкость с разбросом менее 1 %.

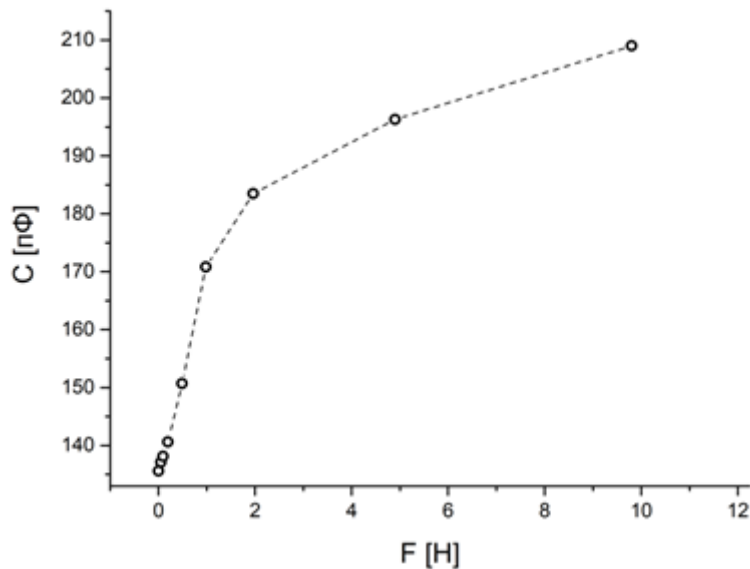


Рисунок 19 — зависимость величины ёмкости изготовленного ёмкостного датчика нажима с площадью 10×7 мм² от приложенной силы F

Разработаны два варианта измерения и оцифровки сигнала (величины электрической ёмкости) датчика с помощью микроконтроллера (плата Arduino). Первый способ основан на использовании активного интегратора на операционном усилителе (усилитель заряда), второй на использовании ёмкостного делителя, образованного измеряемой ёмкостью и опорной, подключенной к высокоомному аналоговому входу контроллера. Второй способ выбран как наиболее перспективный, проведены исследования и оптимизация измерений под условия данной задачи (выбор величины опорной ёмкости, усреднение сигнала).
 Схема измерения ёмкости на основе конвертера ёмкость-напряжение на ОУ

Широко известна схема преобразователя ёмкость-напряжение на основе усилителя заряда с переключаемым конденсатором [12], используемая для тактильных датчиков и датчиков силы/давления [13]. На рис. 20а показана упрощенная принципиальная схема преобразователя с переключаемым конденсатором [4] на операционном усилителе (ОУ), где переменная емкость C_x и опорная емкость C_r являются частями симметричного кремниевого датчика давления. Монолитные МОП-переключатели (1-4) управляются противоположнофазными тактовыми импульсами f_1 и f_2 . Когда часы переключаются, в узле общей емкости появляется заряд. Заряд обеспечивается источником постоянного напряжения, V_{PM} , и пропорционален $(C_x - C_r)$ и, следовательно, приложенному давлению к датчику. Этот заряд подается на Усилитель заряда [14] в напряжение, который включает в себя операционный усилитель, объединяющий конденсатор C_f и переключатель 5 разрядки (сброса) MOS. Выходной сигнал представляет собой импульсы переменной амплитуды (рис. 20b), которые могут передаваться по линии связи и либо демодулироваться для по-

лучения линейного сигнала, либо могут быть дополнительно преобразованы в цифровые данные. Пока коэффициент усиления разомкнутого контура интегрирующего операционного блока высок, выходное напряжение нечувствительно к паразитной входной емкости, напряжению смещения и температурному дрейфу.

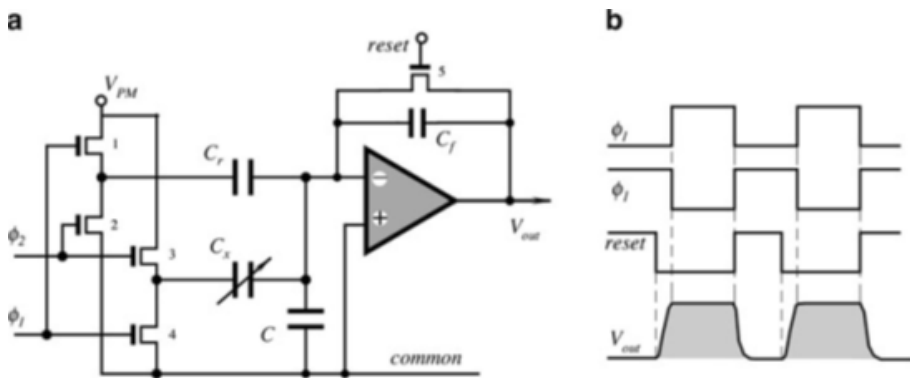


Рисунок 20 — Преобразователь ёмкость-напряжение на основе усилителя заряда на ОУ:
а – общая схема, б – эшюры сигналов

Минимальный обнаруживаемый сигнал (минимальный уровень шума) определяется составляющим шумом и температурными дрейфами компонентов. Анализ схемы показывает, что минимальная мощность шума возникает, когда интегральный конденсатор C_f приблизительно равен конденсатору компенсации частоты операционного усилителя. Когда переключатель сброса MOS переходит из состояния включения в состояние выключения, сигнал переключения вводит некоторый заряд от затвора транзистора сброса на входной суммирующий узел ОРАМ (инвертирующий вход). Этот заряд распространялся через емкость затвора-канала МОП-транзистора 5. Инжекционный заряд приводит к возникновению напряжения смещения на выходе. Эта ошибка может быть компенсирована устройством отмены заряда ****ЗДЕСЬ ДОЛЖНА БЫТЬ ССЫЛКА****, которое может улучшить отношение сигнал/шум на два порядка величины некомпенсированного заряда.

В работе была разработана схема (рис. 21)...

Схема измерения ёмкости на основе ёмкостного делителя напряжения

Для снятия сигнала с датчика использовалась электрическая цепь на основе ёмкостного делителя напряжения, образованного датчиком и опорным конденсатором. На делитель подавался импульс напряжения, сформированный на цифровом выходе контроллера. При этом измерялась величина напряжения на опорном конденсаторе, подключенном к высокоомному аналоговому входу микроконтроллера.

Подбирая ёмкость опорного конденсатора можно оптимизировать цепь для работы в диапазоне выходных значений датчика. Измеренная и оцифрованная АЦП величина напряжения может быть пересчитана в ёмкость датчика или может быть проведена прямая калибровка отчетов АЦП в зависимости от приложенной нагрузки. Величина опорной

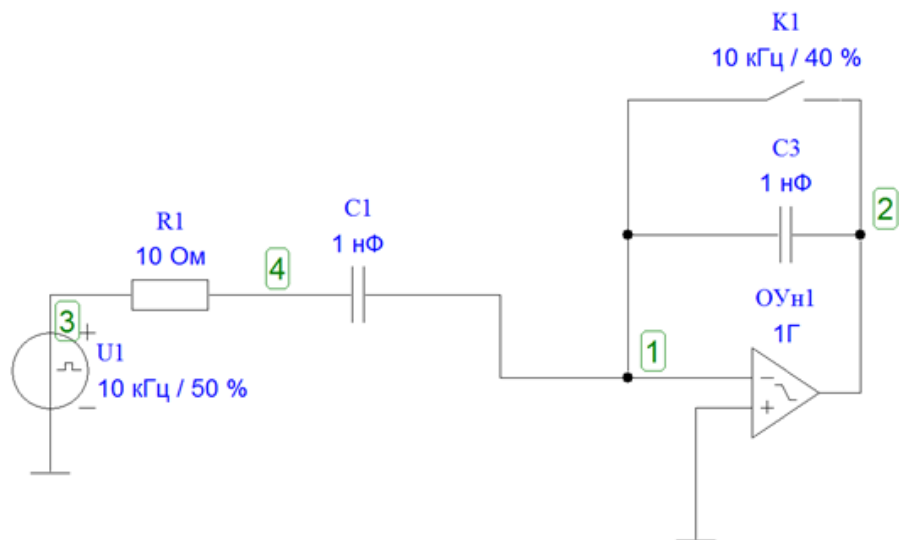


Рисунок 21 — Преобразователь ёмкость-напряжение на основе усилителя заряда на ОУ для ёмкостного датчика силы



Рисунок 22 — Осциллограммы сигналов преобразователя ёмкость-напряжение без приложенной нагрузки: верхняя эюра соответствует импульсу на фходе преобразователя, нижняя — на выходе. Осциллограмма слева иллюстрирует задержку между поданным на вход импульсом и размыканием ключа

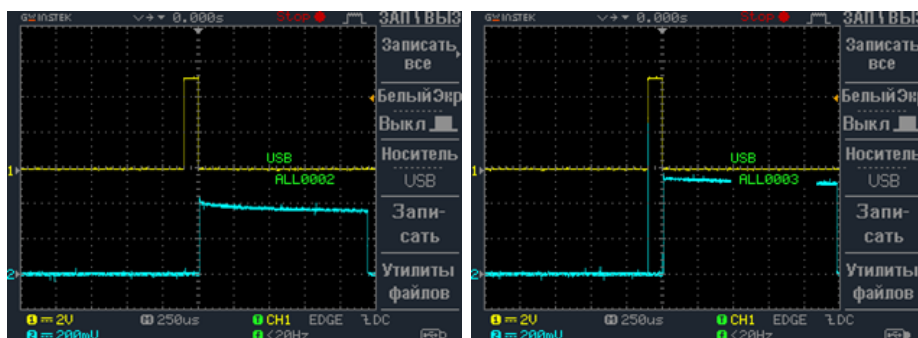


Рисунок 23 — Осциллограммы сигналов преобразователя ёмкость-напряжение с приложенной нагрузкой 1 кг (слева) и при насыщении выходного сигнала

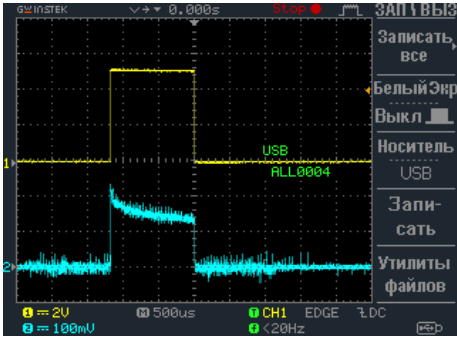


Рисунок 24 — Осциллограммы сигналов на выходе преобразователя ёмкость-напряжение и напряжения на ключе

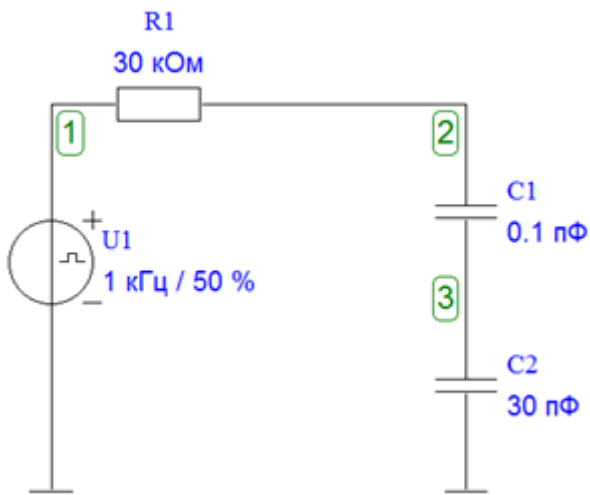


Рисунок 25 — Схема принципа измерения с помощью ёмкостного делителя напряжения

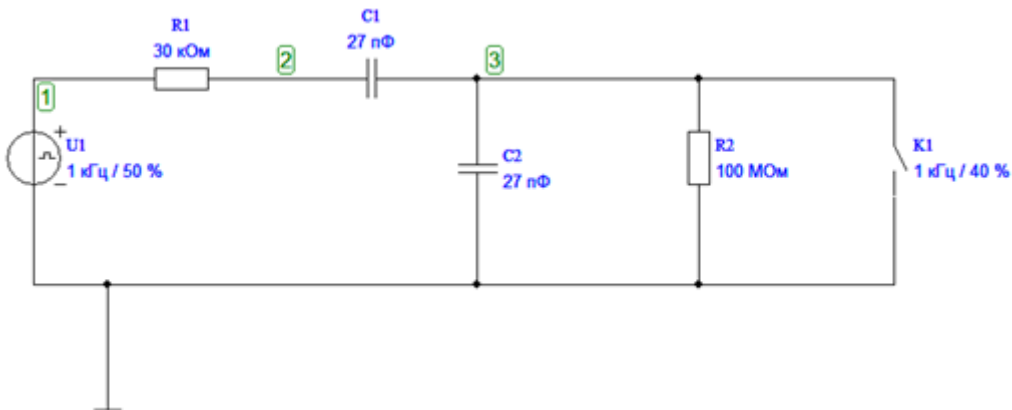


Рисунок 26 — Модельная схема измерения с помощью ёмкостного делителя напряжения на МК

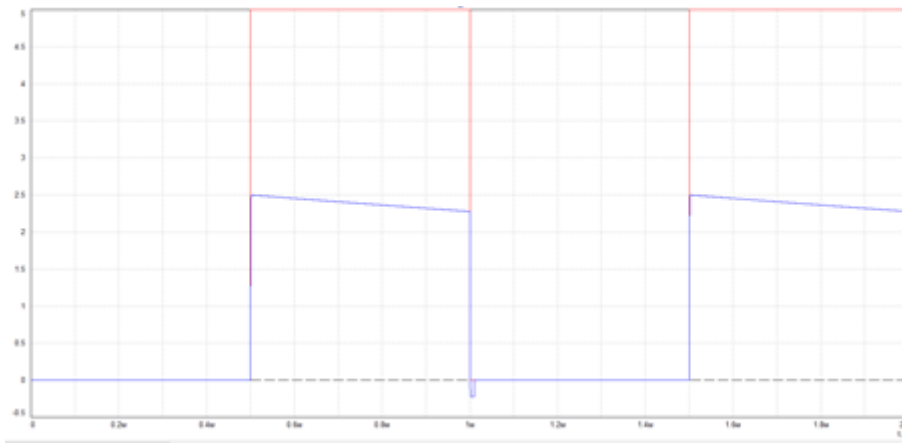


Рисунок 27 — Расчётные осциллограммы входного и выходного напряжения

ёмкости составляла 527 пФ, начальная ёмкость ненагруженного датчика составляла около 135 пФ.

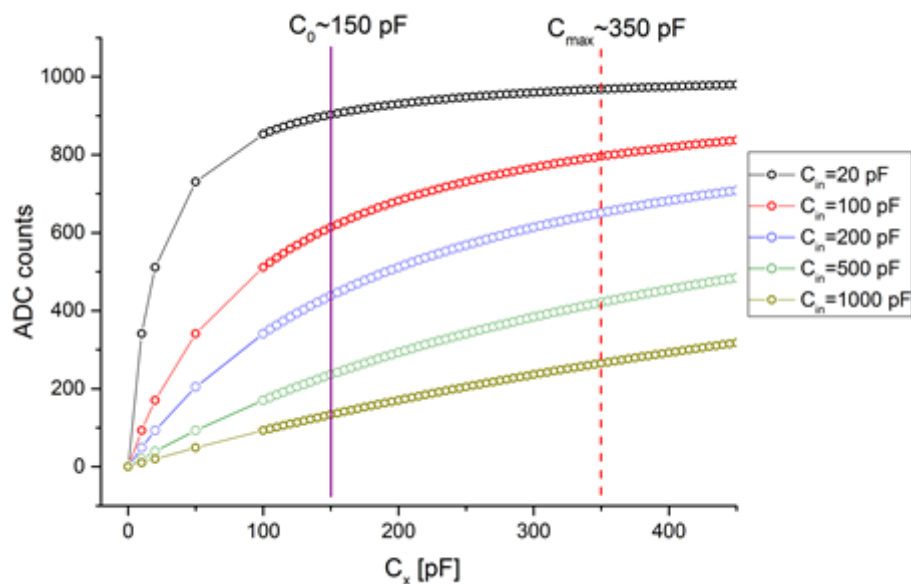


Рисунок 28 — Расчётные зависимости отчётов АЦП от величины измеряемой ёмкости

1.2.3 Структура программного обеспечения

Программное обеспечение программно-аппаратного прототипа распознавателя действий физического объекта включает следующие элементы:

а) Программа

Схема взаимосвязей программного обеспечения приведена на рис. ??.

Программное обеспечение

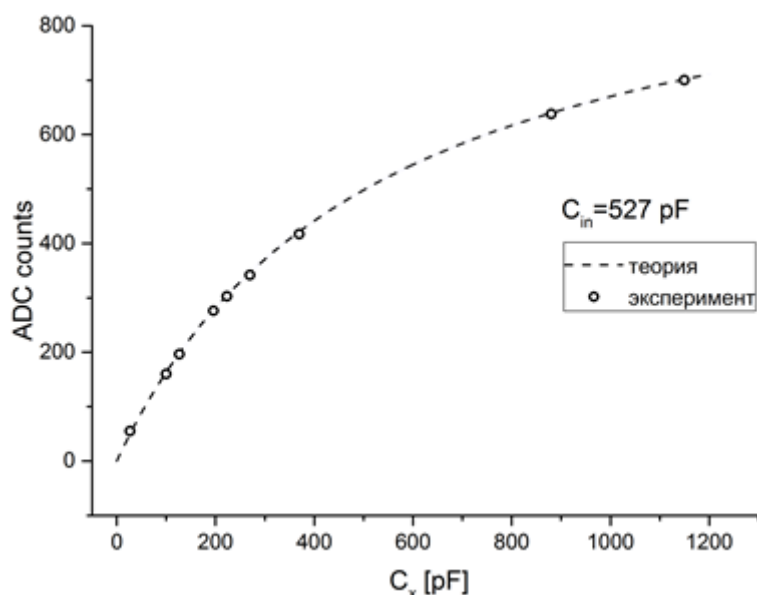


Рисунок 29 — Расчётная и экспериментальная зависимости отчётов АЦП от величины измеряемой ёмкости при фиксированной величине входной ёмкости $C_{in} = 527$ пФ

Рисунок 30 — Схема взаимосвязей программного обеспечения

1.2.3.1 Документация

Для программного обеспечения разработан комплект документов в соответствии с ГОСТ 19.102-77, включающий:

- а) Описание программы на ??? листах;
- б) Спецификация на ??? листах;
- в) Текст программы на ??? листах;
- г) Программа и методика испытаний на ??? листах;

1.2.4 Испытания и эксперименты

1.2.4.1 Программа испытаний

Для проведения испытаний была разработана Программа и методика испытаний. Целью испытания была проверка работоспособности программно-аппаратного прототипа распознавателя действий физического объекта. Задачи:

- а) Проверить

Программно-аппаратный прототип распознавателя действий физического объекта успешно прошел все испытания.

Эксперименты

На рис. 31 приведена зависимость напряжения на резисторе от времени в случае упора зонда-манипулятора в неподвижный предмет с постоянным (предельным) усилием. В начальный момент, когда сила нажима увеличивается, сигнал с резистора также растёт, достигая предельного значения ~ 4 В.

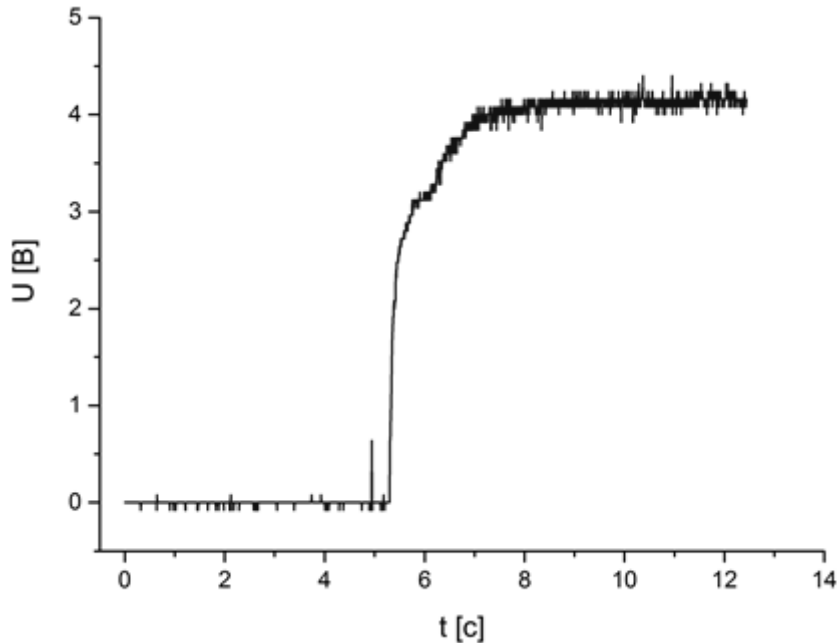


Рисунок 31 — зависимость напряжения на резисторе от времени в случае упора зонда-манипулятора в неподвижный предмет с постоянным (предельным) усилием

На рис. 32 приведена зависимость напряжения на резисторе от времени в случае толкания зондом-манипулятором сдвигаемого предмета с постоянным усилием. Аналогично случаю недвижимого предмета, в начальный момент, когда сила нажима увеличивается, сигнал с резистора также растёт, достигая значения ~ 3 В, меньшего, нежели чем в случае недвижимого предмета. Кроме того, при движении предмета возникают колебания силы нажатия, что приводит к появлению колебаний сигнала, существенно превышающих шум. Подобное может наблюдаться и в момент касания, когда сила нажатия увеличивается (рис. 31).

На рис. 33 приведена зависимость напряжения на резисторе от времени в случае толкания зондом-манипулятором предмета с постоянным усилием и последующим опрокидыванием предмета. Сначала сила нажатия увеличивается, затем, когда предмет начинает падать, сигнал уменьшается до нуля.

На рис. 34 приведена зависимость напряжения на резисторе от времени в случае толкания зондом-манипулятором более массивного предмета (по сравнению со случаем на рисунок 20) с постоянным усилием и последующим опрокидыванием предмета. Данная кривая отличается от случая (рисунок 20) наличием «плато», которое соответствует постоянной силе нажатия на уже наклоняющийся, но стремящийся вернуться в исходное положение, предмет.

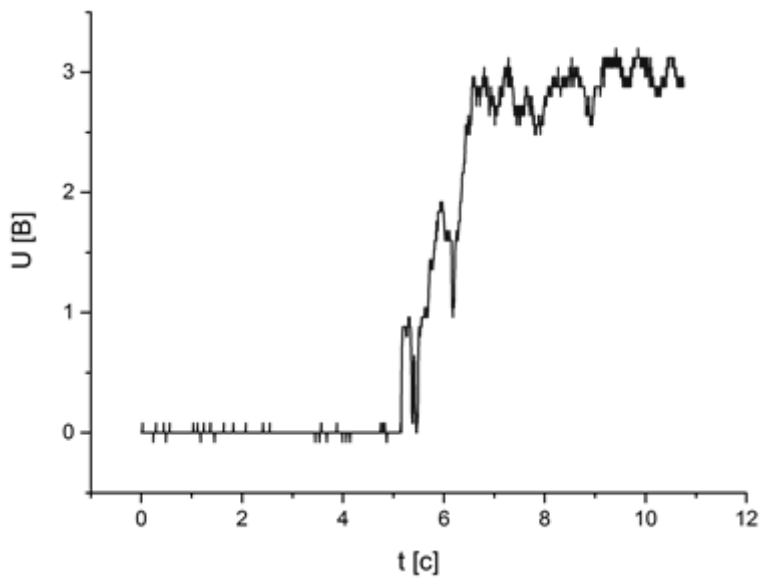


Рисунок 32 — зависимость напряжения на резисторе от времени в случае толкания зондом-манипулятором сдвигаемого предмета с постоянным усилием

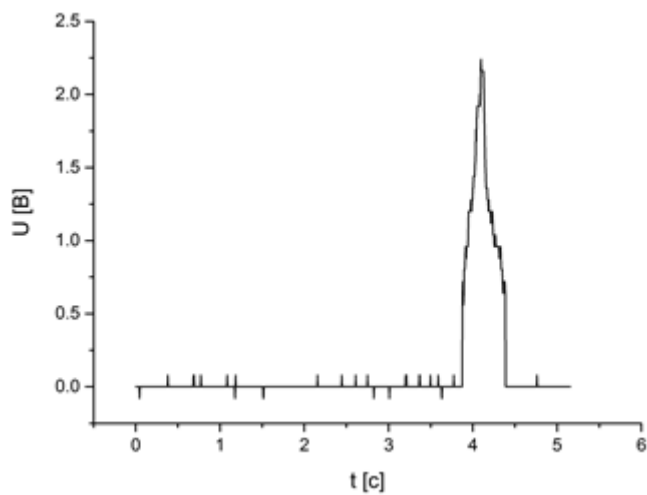


Рисунок 33 — зависимость напряжения на резисторе от времени в случае толкания зондом-манипулятором предмета с постоянным усилием и последующим опрокидыванием предмета

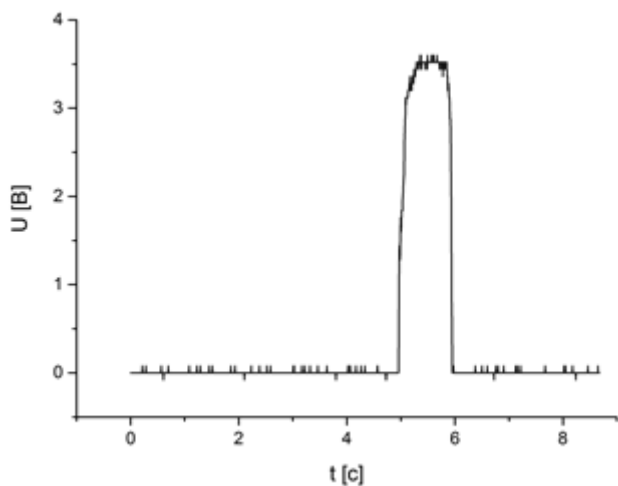


Рисунок 34 — зависимость напряжения на резисторе от времени в случае толкания зондом-манипулятором более массивного предмета с постоянным усилием и последующим опрокидыванием предмета

В случае крайне лёгких предметов (например, лист бумаги) сигнал остаётся близким к нулю.

Как видно, кривые отличаются друг от друга, что позволяет различать данные события. В случае перемещения предмета (толкания) дополнительно может быть использована информация от других датчиков роботизированной системы, отслеживающих перемещение (акселерометры и т.п.). Решение задач различения взаимодействия с объектами в случае конкретной роботизированной системы может быть проведено с помощью машинного обучения.

На рис. 35 приведена зависимость измеренной ёмкости датчика, установленного на зонд-манипулятор, от времени в случае упора зонда-манипулятора в неподвижный предмет с постоянным (предельным) усилием. В начальный момент, когда сила нажима увеличивается, величина ёмкости также растёт, достигая предельного значения 240 пФ.

На рис. 36 приведены зависимости измеренной ёмкости датчика, установленного на зонд-манипулятор, от времени в случае толкания зондом-манипулятором сдвигаемого предмета с постоянным усилием (А, Б – более тяжелый и более легкий предметы, соответственно). Аналогично случаю недвижимого предмета, в начальный момент, когда сила нажима увеличивается, ёмкость возрастает, достигая значения меньшего, нежели в случае неподвижного предмета. Кроме того, при движении предмета возникают колебания силы нажатия, что приводит к появлению колебаний сигнала, существенно превышающих шумы. Подобное может наблюдаться и при касании, когда сила нажатия увеличивается (Фиг.8А). Для более легкого предмета (Фиг.8Б) установившееся стационарное значение ёмкости меньше предельного, что объясняется большим усилием для начального сдвига предмета из состояния покоя.

На рис. 37 приведены зависимости измеренной ёмкости датчика, установленного на зонд-манипулятор, от времени в случае толкания зондом-манипулятором с постоян-

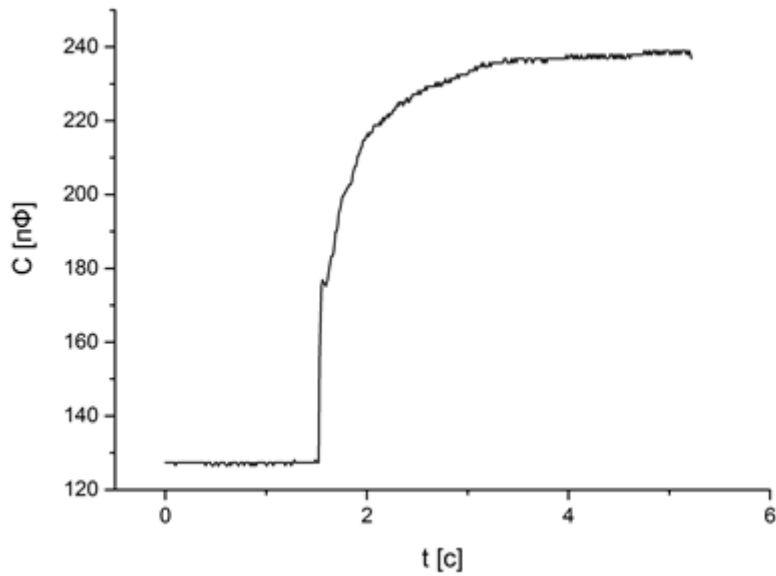


Рисунок 35 — зависимость измеренной ёмкости датчика с размерами 10x7 мм², установленного на зонд-манипулятор, от времени в случае упора зонда-манипулятора в неподвижный предмет с постоянным (предельным) усилием

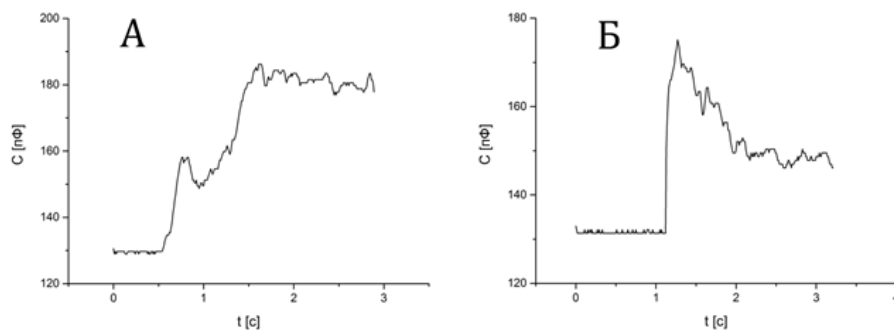


Рисунок 36 — зависимости измеренной ёмкости датчика размерами 10x7 мм², установленного на зонд-манипулятор, от времени в случае толкания зондом-манипулятором сдвигаемого предмета с постоянным усилием (А, Б — более тяжелый и более легкий предметы, соответственно)

ным усилием и последующим опрокидыванием предмета (Фиг.9А – более легкий предмет, Фиг.9Б – более массивный). Кривые содержат «плато», которое соответствует постоянной силе нажатия на уже наклоняющийся, но стремящийся вернуться в исходное положение, предмет. При опрокидывании предмета величина ёмкости уменьшается до исходного значения, соответствующему ненагруженному состоянию.

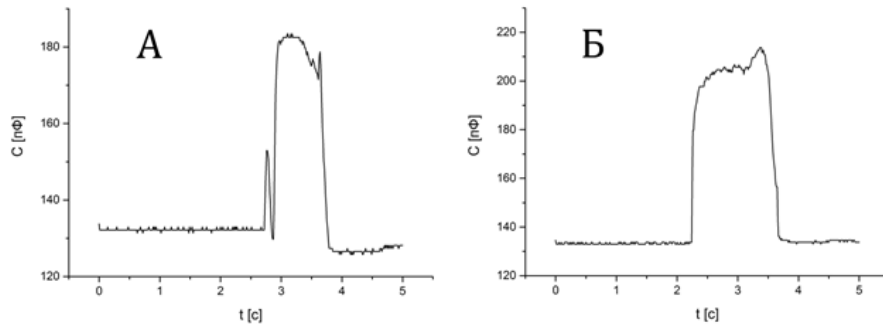


Рисунок 37 — зависимости измеренной ёмкости датчика, установленного на зонд-манипулятор, от времени в случае толкания зондом-манипулятором с постоянным усилием и последующим опрокидыванием предмета (А – более легкий предмет, Б – более массивный)

В случае крайне лёгких предметов (например, лист бумаги) сигнал остаётся близким к нулю. Как видно, кривые отличаются друг от друга, что позволяет различать данные события. В целом, характер кривых идентичен случаю использования FSR датчика. Решение задач различия взаимодействия с объектами также может быть проведено с помощью машинного обучения.

1.2.5 Выпускные квалификационные работы

В рамках разработки программно-аппаратного прототипа распознавателя действий физического объекта ведется подготовка выпускной квалификационной работы

В выпускной работе .

1.2.6 Практика студентов

В рамках разработки программно-аппаратного прототипа распознавателя действий физического объекта

В рамках практики .

1.2.7 Рабочая программа практики

В рамках разработки программно-аппаратного прототипа распознавателя действий физического объекта была разработана рабочая программа

Рабочая программа предполагает .

1.2.8 Публикации

В рамках проекта были подготовлены следующие публикации:

- а) Одна

1.2.9 Результаты

Был разработан программно-аппаратный прототип распознавателя действий физического объекта. Аппаратная часть прототипа включает . Программная часть включает . Прототип предназначен для . Прототип использует 3 типа данных наблюдения за действием. Прототип определяет 5 типов действий. На прототип разработана программная документация в соответствии с ГОСТ 19.102-77.

Проведены испытания прототипа согласно разработанной Программе и методике испытаний. Все испатния были пройдены успешно.

С использованием прототипа были проведены эксперименты.

В рамках разработки прототипа ??? проходил практику, ??? готовит выпускную квалификационную работу.

Дальнейшее развитие программно-аппаратного прототипа планируется в направлении ???.

2 Разработка концептуального прототипа для бионического костюма и системы обработки данных

2.1 Программно-аппаратный прототип распознавателя движения и когнитивного состояния человека с возможностями применения для цифрового производства и для умного дома

2.1.1 Современное состояние систем оценки вовлеченности человека

В настоящее время растет интерес к исследованиям в области компьютерных наук, сфокусированным на исследовании эмоционального воздействия на пользователя, возникающее в результате человеко-машинного взаимодействия. Чтобы предложить адаптивный опыт работы с эмоциями своего пользователя, интеллектуальные компьютерные системы должны иметь возможность определять эмоциональное (аффективное) состояние пользователя в соответствии некоторой моделью оценки эмоций, обрабатывать его и, по возможности, адаптировать процесс человеко-машинного взаимодействия в соответствии с чувствами пользователя в текущий момент [15]. Такие системы и технологии человеко-машинного взаимодействия на основе эмоционального состояния пользователя получили название “аффективных вычислений” (affective computing). Они становятся особенно актуальным с развитием дистанционных онлайн систем, адаптированных для обучения и работы, как в связи развитием широкополосного доступа в интернет, так и в связи с эпидемиологической обстановкой, связанной с распространением COVID-19. Исследования в данной области могут повысить эффективность работы и обучения, а также улучшить пользовательский опыт от взаимодействия с такими адаптивными системами. Это связано с тем, что на поведение людей часто влияет их эмоциональное состояние, а поведение, в свою очередь, влияет как на индивидуальную, так и на командную производительность [16]. Возможности таких систем могут быть использованы в интеллектуальном цифровом производстве, где они способны обеспечить гибкое, непрерывное взаимодействие с человеком, направленное на повышение безопасности, производительности, благополучия и удовлетворенности оператора оборудования и, как следствие, всего предприятия [17]. Существующие модели эмоционального состояния человека пытаются описывать эмоции человека в двухмерных системах координат вида валентность-стеничность, вовлеченность-удовлетворенность [18]. Среди них шкала вовлеченности наиболее точно коррелирует с показателями производительности работы и успехами в образовательном процессе. Исследования влияния вовлеченности учащихся на качество усвоения учебного материала является ключевой темой в учебной литературе начиная с 1980-х годов [19]. Исследования в области интеллектуального анализа образовательных данных, мультимодального обучения, когнитивных наук, психологии позволили значительно продвинуться в области анализа взаимосвязи продуктивности обучения и работы и вовлеченности, а также показа-

ли значительные перспективы в плане контроля за уровнем вовлеченности пользователей [20, 21]. При оценке уровня вовлеченности можно использовать различные параметры: аффективное состояние пользователя (такое как, скука, замешательство, разочарование и беспокойство [22]), уровень внимания (например, низкий, высокий, нормальный) [23] или количественные и качественные учебного или рабочего процесса [24].

В общем случае в системах, использующих «аффективные вычисления», можно выделить две подсистемы: измерительную и управляющую. И для эффективной работы управляющей подсистемы необходимо получать данные об эмоциональном состоянии пользователя от измерительной подсистемы. В предыдущие десятилетия для измерения уровня вовлеченности использовались опросники, которые заполняли пользователи по окончании экспериментов или данные контрольных наблюдений, которые заполняли сторонние наблюдатели, однако автоматизация измерений выдвинула на лидирующие позиции автоматизированные системы оценки [25]. Методы в автоматизированной оценке вовлеченности извлекают признаки из различных величин, измеряемыми внешними устройствами и датчиками. Это могут быть видеокамера (движения головы и глаз, мимика лица, жесты, позы и т. д.) [26], датчики измерения физиологических и неврологических параметров (электрокардиография, электроэнцефалография, измерение кровяного давления, частоты дыхания, электросопротивления кожи и т. д.) [27], микрофон [28], клавиатуры, компьютерная мышь [29, 30] и пр. Собираемые данные обрабатываются системой анализа, построенной на основе интеллектуальных методов обработки, таких как искусственные нейронные сети, регрессионный и кластерный анализы, нечёткая логика [26, 27, 28, 29, 30]. Например, в работе [26] для определения уровня вовлеченности онлайн-учащихся используется распознавание эмоций, осуществляемое с помощью анализа мимики лица искусственной нейронной сети с глубоким обучением. Изображение испытуемого захватывается с помощью видеокамеры. предлагается основанный на глубоком обучении. Для оценки уровня вовлеченности использовались классификаторы с двумя (не вовлечен и вовлечен) и трехуровневые (не вовлечен, вовлечен, сильно вовлечен) решения. Двухуровневые (незаятый и вовлеченный) и тремя уровнями. Для извлечения признаков, характеризующих различные мимические выражения и не зависящих от особенностей конкретного испытуемого использовался метод локальных направленных шаблонов (local directional pattern). Для выявления нелинейной корреляции между признаками использовался метод анализ основных компонентов. Предлагаемый метод обеспечивал высокую точность классификации различных уровней вовлеченности онлайн-учащихся во время различных вариантов активности (например, чтение, письмо, просмотр видеоуроков и участие в онлайн-встречах).

В работе [28] представлен автоматизированный способ оценки вовлеченности во время онлайн-встреч на основе данных видео- и аудиоанализа. Оценка вовлеченности состоит из 6 уровней и предполагает различную активность участников в соответствии с их вовлеченностью (не вовлечен, легкая вовлеченность, вовлеченное взаимодействие, намерение

действовать, активное действие, вовлеченное действие). Работа системы основана на характеристике позы и движения тела человека, голосе, и выражении лица с помощью анализа данных с 3D-камеры, 2D-камеры и микрофонов. На основе получаемых данных были выделены классификаторы, как определенные компоненты позы или движения тела, голоса, выражения лица, такие как громкость речи, улыбка, покачивание головой и т.д. Данные классификаторы использовались для построения модели на основе машины опорных векторов. Полученная модель позволяет оценить как вовлеченность отдельных участников онлайн-встречи, так и оценить общую групповую вовлеченность.

В работе [29] представлена система автоматизированной оценки эмоционального состояния с использованием неинвазивных методов диагностики на основе анализа особенностей использования клавиатуры и компьютерной мыши. Предложенный подход использует информацию о движении курсора мыши, нажатиях клавиш мыши, а также нажатия клавиш клавиатуры во время решения математических задач, включенных в состав онлайн-курса. Для оценки эмоционального состояния использовались 42 индикатора использования клавиатуры и 96 индикаторов использования компьютерной мыши. Для разметки полученного датасета и использования его в машинном обучении пользователей, участвовавших в эксперименте, просили описать свое эмоциональное состояние с помощью заранее подготовленного опросника. Полученные данные использовались для обучения различных классификаторов (наивный байесовский классификатор, случайный лес, AdaBoost), среди которых наибольшую точность показал случайный лес.

В работе [30] предложен автоматический многофакторный подход для измерения вовлеченности студентов. Для оценки уровня вовлеченности используются данные полученные от видеокамеры, клавиатуры и компьютерной мыши. Предложенная система работает в режиме реального времени и позволяет с достаточно высокой точностью определять вовлеченность студента, используя недорогое оборудование. Модель позволяет оценить три уровня вовлеченности (средняя, высокая, сильная) пользователя с использованием распознаваемых эмоций, количества нажатий клавиш клавиатуры и компьютерной мыши, средней скорости движения курсора, средней скорости печати на клавиатуре.

Исследование [27] описывает систему, использующую данные из „умной майки“ и трекера глазных яблок для сбора следующих видов данных: электрокардиограмм, электропроводимости кожи, частоты дыхания, движения тела и положения глазных яблок при обучении студентов управлению беспилотными летательными аппаратами. Во время проведения эксперимента испытуемые проходили три этапа: просматривали видео об управлении беспилотным летательным аппаратом, использовали программу-авиасимулятор, а также пытались с помощью радиоуправляемого квадрокоптера пролететь через специально подготовленную полосу препятствий. Для разметки датасета использовались заполняемые испытуемыми опросники. В результате было установлено, что уровни вовлеченности связаны с этапом эксперимента и наибольшую вовлеченность испытуемые приписывали управлению квадрокоптером в симуляторе и реальной жизни, а минимальную - просмотру видеоро-

ка с описанием способа управления квадрокоптером. Поэтому модель предполагает два уровня вовлеченности: низкую и высокую. Для выявления взаимосвязи физиологических и поведенческих особенностей и уровня вовлеченности использовалась логистическая регрессия. Было установлено, что пульс, параметры электрокардиограммы, электропроводность кожи и движение зрачков позволяют определять уровень вовлеченности с высокой точностью.

Таким образом, в настоящее время автоматизированные системы оценки вовлеченности пользователя обычно используются следующие виды аппаратного обеспечения: видеочамера (3D и 2D), микрофон, клавиатура, компьютерная мышь, трекер зрачков глаз и „умная майка“ (биокостюм) - система измерения физиологических параметров на основе множества датчиков, которые крепятся к поверхности текстильного изделия. Стоит отметить, что в рассмотренных исследованиях используется только 1-3 варианта оборудования. В предлагаемом прототипе по оценке степени вовлеченности человека предполагается использовать 4 вида оборудования (3D камера, микрофон, клавиатуры, мышь) и эмулировать 5 вид - «умную майку» (биокостюм) с помощью измерения пульса и частоты дыхания. Данное оборудование будет применяться для распознавания различных типов движения и когнитивного состояния человека, которые в свою очередь будут использованы для определения уровня его вовлеченности. Для построения моделей вовлеченности в описываемых исследованиях используется метод обучения с учителем, когда вовлеченность анализируется либо по результатам опросников, заполняемых испытуемыми, либо на основании анализа мимики человека с помощью видеочамеры. В предлагаемом нами прототипе будет использован более простой метод построения модели с использованием обучения без учителя на основе кластерного анализа получаемых данных.

Таким образом, целью данной работы будет являться разработка программно-аппаратного прототипа по распознаванию движения и когнитивного состояния человека, в том числе вовлеченности в рабочий процесс, с эмуляцией функций биокостюма, использующего данные от видеочамеры, микрофона, клавиатуры и компьютерной мыши.

2.1.2 Архитектура программно-аппаратного прототипа

Прототип представляет собой персональный компьютер, с двумя мониторами, клавиатурой и компьютерной мышью, к которому дополнительно подключаются измерительные устройства (3D камера и микрофон) через стандартизированные интерфейсы передачи данных. Пользователь использует персональный компьютер и один из мониторов для выполнения своих текущих задач (чтение информации с экрана монитора, набор текста, работа с графическим интерфейсом с помощью компьютерной мыши). Все первичные данные, полученные с устройств ввода (3D-камера, микрофон, клавиатура, компьютерная мышь), а также обработанные данные записываются в базу данных в фоновом режиме. Отображение данных в виде временных зависимостей первичных и обработанных дан-

ных производится с помощью второго монитора, который располагается вне поля зрения пользователя.

2.1.2.1 Аппаратная архитектура

Для работы прототипа необходимы следующие устройства:

- а) Рабочая станция для исполнения программных компонентов прототипа;
- б) Два монитора, один из которых используется пользователем в ходе работы прототипа, а второй - для визуализации работы прототипа через отображение пользовательского интерфейса;
- в) 3D-камера для анализа движения пользователя и измерения пульса;
- г) Микрофон для измерения частоты дыхания;
- д) Клавиатура для оценки активности пользователя по набору текста;
- е) Компьютерная мышь для оценки активности пользователя по взаимодействию с графическим интерфейсом.

В ходе работы прототипа используются методы интеллектуального анализа данных, такие как искусственные нейронные сети, поэтому рабочая станция должна быть оснащена видеокартой для ускорения тензорных вычислений. Кроме того, программная архитектура прототипа предполагает активное использование оперативной памяти и многочисленные операции записи в базу данных, поэтому рабочая станция должна обладать достаточным объемом оперативной памяти и твердотельным накопителем для быстрой записи в долговременное хранилище. Таким образом, конфигурация рабочей станции была выбрана следующим образом: **TODO: Указать конфигурацию используемого компьютера** К рабочей станции подключались подключалось два монитора **TODO: Указать модели и параметры мониторов и способ подключения** При выборе 3D-камеры мы руководствовались шириной угла обзора (камера должна быть достаточно широкоугольной, чтобы вмещать изображение пользователя, получаемое с небольшого расстояния), минимальным расстоянием построения карты глубины изображения, наличием распространенного стандартизированного интерфейса подключения, компактными размерами. По указанным параметрам была выбрана 3D-камера Intel RealSense d435 со следующими характеристиками:

- а) Интерфейс подключения USB 3.0;
- б) Разрешение RGB-сенсора и частота кадров: 1920x1080 с частотой дискретизации до 30 кадров в секунду;
- в) Угол обзора RGB-сенсора 69° по горизонтали и 42° по вертикали;
- г) Разрешение инфракрасного датчика глубины: 1280x720 с частотой дискретизации до 90 кадров в секунду;

- д) Минимальное фокусное расстояние датчика глубины: 11 сантиметров;
- е) Максимальное расстояние, определяемое с помощью датчика глубины: 10 метров;
- ж) Угол обзора RGB-сенсора 87° по горизонтали и 58° по вертикали;
- з) Размеры камеры: $90 \times 25 \times 25$ миллиметров.

Камера подключается к рабочей станции через кабель USB и крепится с помощью двухстороннего скотча на верхнюю часть пользовательского монитора таким образом, чтобы пользователь, сидящий в кресле перед экраном монитора, находился в центре кадра. При выборе микрофона первоначально рассматривалось настольное исполнение, однако предварительные эксперименты показали, что дополнительные вибрации, передающиеся через стол при печати, движениях рук пользователя, а также значительное расстояние между микрофоном и источником звука не позволяют надежно измерять частоту дыхания. Поэтому было решено использовать микрофон с оголовным креплением Defender Aura 104 со следующими характеристиками:

- а) Интерфейс подключения: проводной микрофонный вход, длина кабеля 1.8 метра;
- б) Импеданс микрофона - 2.2 кОм;
- в) Чувствительность микрофона - 54 дБ;
- г) Частотный диапазон микрофона 20-16000 Гц.

В качестве дополнительно модификации на заднюю (относительно лица пользователя) часть микрофона с помощью двухстороннего скотча приклеивался небольшой кусочек картона, который вибрировал при выдохе и позволял повысить точность определения момента выдоха. Клавиатура и компьютерная мышь, используемые в прототипе, подключались к рабочей станции через интерфейс USB 2.0. **TODO: Указать модели клавиатуры и компьютерной мыши и их параметры**

2.1.2.2 Программная архитектура

Прототип реализован по микросервисной архитектуре, в которой существует несколько независимых программных компонентов, каждый из которых выполняет отдельную функцию и слабо связано с остальными. Обмен данными между программными компонентами осуществляется через сервер системы управления базами данных (СУБД) MySQL. В состав программной архитектуры входят следующие программные компоненты (рисунок 38):

- а) Программный компонент видеоанализа;
- б) Программный компонент расчета статистических параметров на основе видеоданных;
- в) Программный компонент аудиоанализа;
- г) Программный компонент сбора данных с клавиатуры и компьютерной мыши;

- д) Программный компонент расчета статистических параметров на основе данных с клавиатуры и компьютерной мыши;
- е) Программный компонент оценки вовлеченности пользователя;
- ж) Программный компонента пользовательского интерфейса.

Программные компоненты видеоанализа, аудиоанализа и сбора данных с клавиатуры и компьютерной мыши занимаются сбором и предварительной обработкой данных получаемых с устройств (3D-камера, микрофон, клавиатура, компьютерная мышь). Полученные данные записываются в СУБД MySQL. Программные компоненты расчета статистических параметров на основе видеоданных, данных с клавиатуры и компьютерной мыши используют первичные и предобработанные данные, получаемые из СУБД MySQL для расчета и записи в базу данных статистических параметров, которые использует программный компонент оценки вовлеченности пользователя. Он, в свою очередь, использует эти данные для расчета уровня вовлеченности по трем видам активности (чтение текста, набор текста, взаимодействие с графическим интерфейсом с помощью компьютерной мыши) и вычисления суммарной оценки вовлеченности пользователя. Вычисления значения уровня вовлеченности пользователя записываются также записываются в базу данных. Программный компонент пользовательского интерфейса используется для визуализации работы прототипа и выводить собранные и вычисленные данные в виде временных зависимостей (каждая из таблиц имеет временные метки измерений/вычислений). Программные компоненты написаны на языке Python 3 и исполняются в операционной системе Linux Ubuntu 20.04. Версия сервера СУБД MySQL - 8.0.

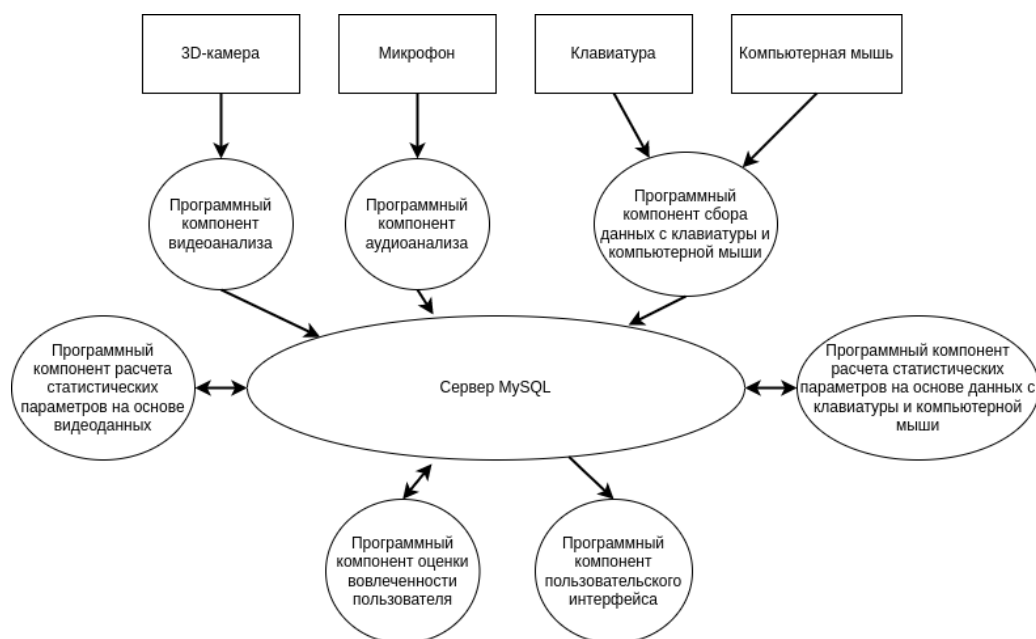


Рисунок 38 — Программная архитектура прототипа

План раздела:

- а) Получение и анализ видеоданных

- б) Получение и анализ аудиоданных
- в) Получение и анализ данных с клавиатуры и компьютерной мыши
- г) Эмуляция данных с бионического костюма
- д) Модель оценки вовлеченности человека
- е) Пользовательский интерфейс

Получение и анализ видеоданных

В рамках прототипа была поставлена задача распознавания следующих видов движения и когнитивного состояния человека по данным видео:

- а) движение и положение головы (повороты влево-вправо, наклоны вверх-вниз);
- б) движение и положение корпуса (наклоны корпуса вперед-назад, как расстояние от камеры до лица);
- в) движение зрачков глаз;
- г) движение век (моргание);
- д) эмоциональное состояние (по изменению мимики лица);
- е) пульс (на основе изменения интенсивности цвета лица).

Для этого был разработан программный модуль видеоанализа, в режиме реального времени извлекающий из видеопотока следующие параметры и сохраняющий их в базу данных:

- а) угол поворота головы по горизонтали (от 0 до 180 градусов, 0 – голова повернута влево, 90 – в камеру, 180 – вправо);
- б) угол наклона головы по вертикали (от 0 до 180 градусов, 0 – голова опущена вниз, 90 – направлена в камеру, 180 – поднята вверх);
- в) движение и положение корпуса (наклоны корпуса вперед-назад, как расстояние от камеры до лица);
- г) положение зрачка в глазнице по горизонтали;
- д) положение зрачка в глазнице по вертикали;
- е) положение век (отношение расстояния между веками к ширине глаза);
- ж) 7 типов эмоционального состояния (нейтральное, радость, удивление, грусть, гнев, отвращение, страх);
- з) усреднённый цвет кожи на щеках в формате RGB.

Функция вычисления пульса была вынесена в отдельный модуль, выполняющий вычисление статистик по сохранённым данным. Такое решение обосновано тем, что позволяет применять различные алгоритмы определения пульса с различными настройками постфактум, сравнивая их на одних и тех же данных. При этом объём сохраняемых дан-

ных увеличивается незначительно. Для извлечения параметров для каждого кадра модуль выполняет следующие действия.

- а) Поиск лица в кадре.
- б) Распознавание точек лица (построение сетки лица) в трёхмерных координатах.
- в) Извлечение из кадра изображения лица (исходя из полученных координат точек лица), масштабирование и нормировка, распознавание эмоционального состояния с помощью свёрточной нейронной сети (СНС).
- г) Вычисление угла поворота головы по горизонтали, как угла между вектором, соединяющим симметричные точки в височной области, и вектором, перпендикулярным плоскости кадра.
- д) Вычисление наклона головы по вертикали, как угла между вектором, соединяющим лоб и подбородок, и вектором, перпендикулярным плоскости кадра.
- е) Вычисление расстояния от камеры до лица, как расстояния до точки в центре лица (по данным карты глубин, получаемой с помощью стереокамеры).
- ж) Вычисление положения зрачка в глазнице по горизонтали, как координаты зрачка вдоль линии, соединяющей левый и правый уголки глаз, нормировка координаты на длину линии.
- з) Вычисление положения зрачка в глазнице по вертикали, как координаты зрачка вдоль линии, соединяющей верхнюю и нижнюю точки глазницы, нормировка координаты на длину линии.
- и) Вычисление положения век (степени открытия глаза), как отношения расстояния между веками к расстоянию между левым и правым уголками глаз.
- к) Выделение на изображении области щёк, вычисление средних значений составляющих цвета.
- л) Сохранение полученных параметров в базе данных.

Для поиска лица в кадре и построения сетки лица используется библиотека MediaPipe, предлагающая готовое решение для поиска лица и построения сетки с помощью нейронных сетей [31]. Использован программный интерфейс библиотеки для языка программирования Python. В качестве входных данных передаётся изображение с камеры. Далее выполняется детектирование лица с помощью нейронной сети BlazeFace [32]. По полученным координатам лица извлекается изображение лица и передаётся нейронной сети, выполняющей построение сетки лица [33]. Результатом обработки является набор координат точек лица в трёхмерном пространстве. Пример построения сетки лица с помощью библиотеки MediaPipe показан на рисунке 39.

Для распознавания эмоционального состояния разработана свёрточная нейронная сеть (СНС). Архитектура сети представлена в таблице 5. Входными данными нейронной сети является изображение разрешением 64x64 пиксела в градациях серого. Основу

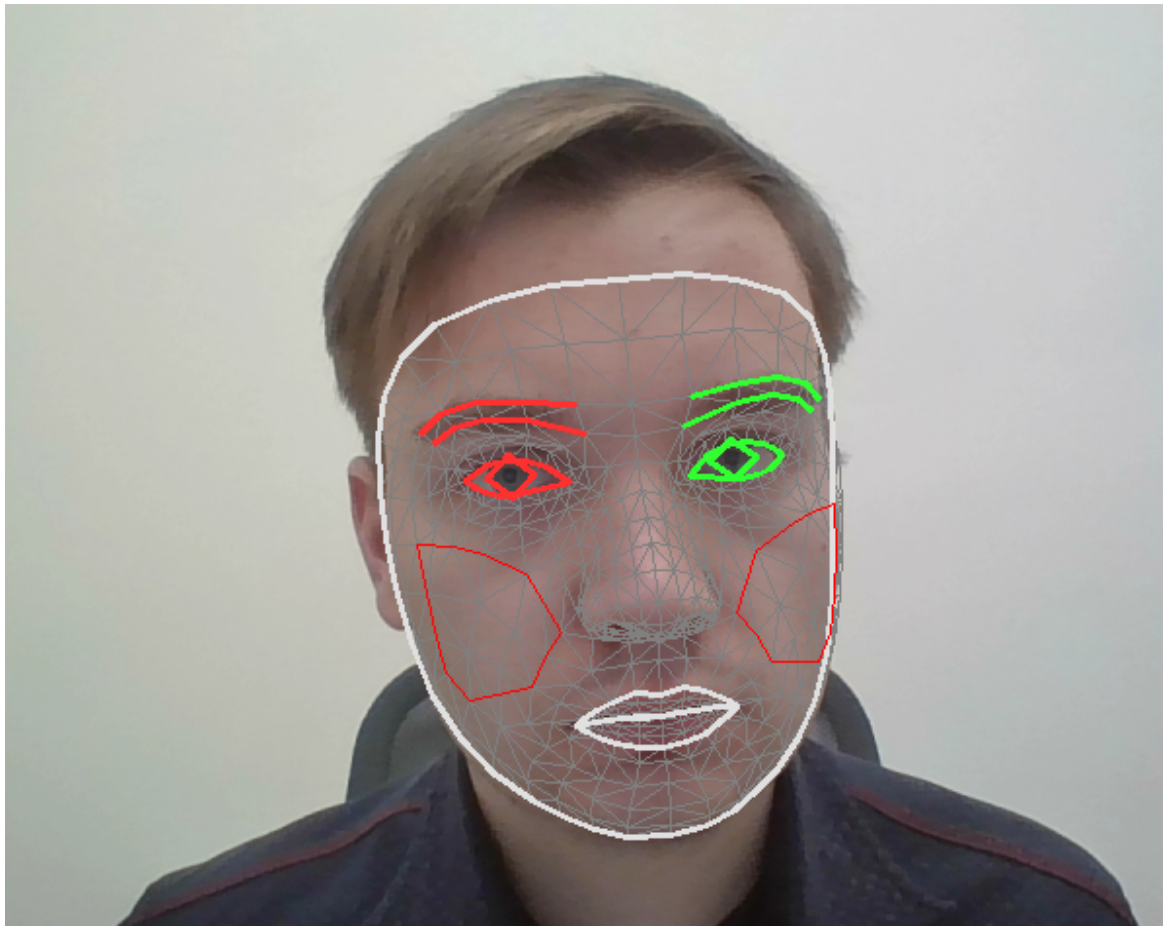


Рисунок 39 — Пример построения сетки лица с помощью библиотеки MediaPipe

нейронной сети составляет последовательность слоёв свёртки (англ. Convolution) и субдискретизации (англ. Pooling), выполняющих извлечение признаков из изображения. По полученным признакам с помощью полносвязного слоя выполняется определение степени выраженности каждой эмоции. Для всех свёрточных слоёв используется размер фильтров 3×3 , без смещения, шаг 1×1 , пакетная нормализация и функция активации ReLU6. Поскольку разные эмоции могут проявляться одновременно (например, удивление и страх), для полносвязного слоя применяется сигмоидальная функция активации, что позволяет независимо оценивать степень проявления каждой эмоции, которая выражается числом от 0 до 1. Для улучшения обобщающей способности были применены слои отключения (англ. Dropout).

Таблица 5 — Архитектура СНС для распознавания эмоций

Слой	Размер выхода	Параметры
Input	(64, 64, 1)	
Conv2D	(64, 64, 64)	Кол-во фильтров: 64
Conv2D	(64, 64, 64)	Кол-во фильтров: 64
MaxPooling2D	(32, 32, 64)	Размер подвыборки: 2×2

Продолжение таблицы 5

1	2	3
Dropout	(32, 32, 64)	Процент отключения: 5%
Conv2D	(32, 32, 128)	Кол-во фильтров: 128
Conv2D	(32, 32, 128)	Кол-во фильтров: 128
MaxPooling2D	(16, 16, 128)	Размер подвыборки: 2x2
Dropout	(16, 16, 128)	Процент отключения: 5%
Conv2D	(16, 16, 256)	Кол-во фильтров: 256
Conv2D	(16, 16, 256)	Кол-во фильтров: 256
MaxPooling2D	(8, 8, 256)	Размер подвыборки: 2x2
Dropout	(8, 8, 256)	Процент отключения: 5%
Conv2D	(8, 8, 512)	Кол-во фильтров: 512
Conv2D	(8, 8, 512)	Кол-во фильтров: 512
MaxPooling2D	(4, 4, 512)	Размер подвыборки: 2x2
Flatten	(8192)	
Dropout	(8192)	Процент отключения: 30%
Dense	(10)	Кол-во нейронов: 10

Обучение нейронной сети выполнялось на наборе данных FER+ [34]. Данный набор основан датасете FER-2013 [35] и отличается разметкой данных. В то время, как в оригинальном наборе данных для каждого изображения указан только один класс (эмоция), в наборе данных FER+ каждому изображению соответствует 10 чисел от 0 до 10, соответствующих 10 классам:

- а) нейтральное (neutral),
- б) радость (happiness),
- в) удивление (surprise),
- г) грусть (sadness),
- д) гнев (anger),
- е) отвращение (disgust),
- ж) страх (fear),
- з) презрение (contempt),
- и) неизвестно (unknown),
- к) лицо отсутствует (NF)

Число, присвоенное каждому классу, соответствует количеству экспертов, отметивших данное изображение, как относящееся к данному классу. Набор данных состоит из 28709 изображений в обучающей выборке, 3589 в валидационной и 3589 в тестовой. Каждое изображение имеет разрешение 48x48 пикселей и представлено в градациях серого.

Примеры изображений и число экспертов, проголосовавших за каждый класс приведены на рисунке 40.

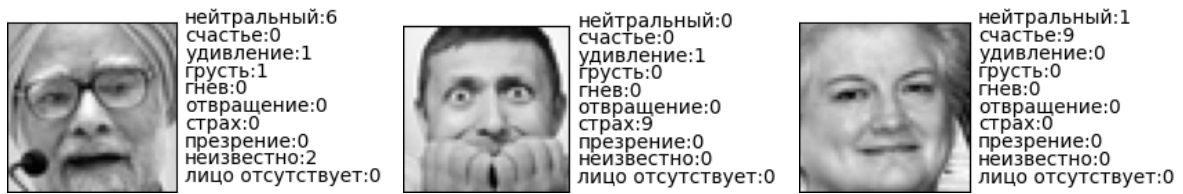


Рисунок 40 — Примеры изображений из набора данных FER+ [34]

Для формирования входных данных нейронной сети применялось масштабирование изображений к размеру 64x64 пиксела с помощью бикубической интерполяции. Эталонный вектор выходных данных формировался из 10 меток классов с применением нормировки для каждого изображения таким образом, чтобы сумма элементов вектора была равна 1. Обучение выполнялось в течении 100 эпох, с применением оптимизатора Adam [36] с размером мини-пакета равным 8 скоростью обучения равной 0.0001. В качестве функции потерь использовалась перекрёстная энтропия. На рисунке 41 представлены полученные кривые обучения, на рисунке 42 приведены матрицы ошибок на обучающей (train), валидационной (val) и тестовой (test) выборках, нормализованные по истанным классам. Результаты распознавания изображений из тестовой выборки с помощью полученной нейронной сети приведены на рисунке 43.

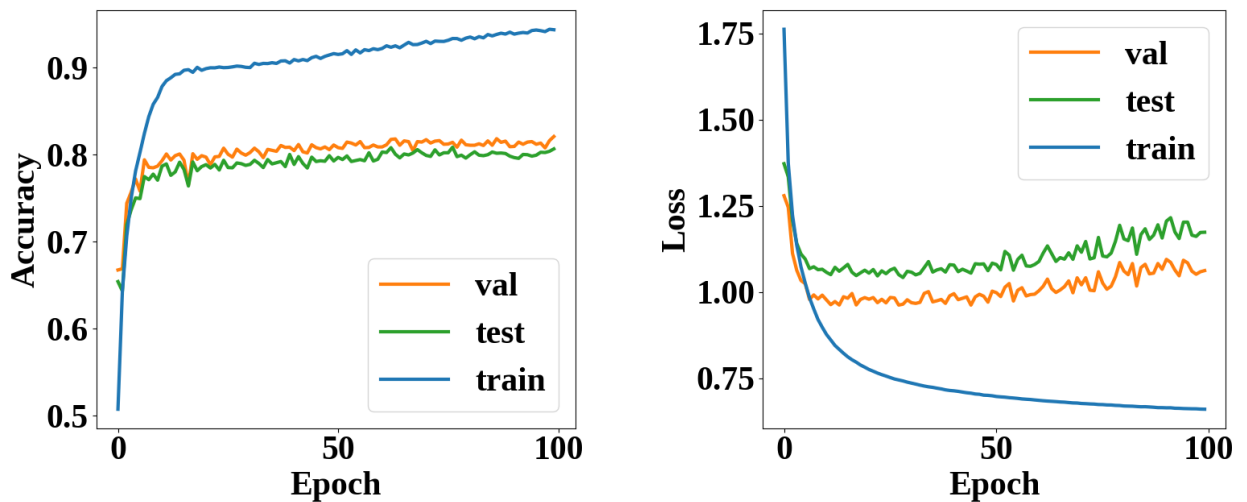


Рисунок 41 — Изменение точности и ошибки нейронной сети в процессе обучения

Из представленных графиков можно видеть, что модель достигает минимальной ошибки на валидационной и тестовой выборках примерно на 20-ой эпохе обучения. При этом точность классификации составляет около 80%. На приведённых матрицах ошибок можно видеть, что лучше всего распознаются нейтральное состояние, радость и удивление. Наихудшая точность классификации наблюдается для презрения (contempt) и изображений, для которых большинство экспертов затруднились дать оценку – «неизвестно»

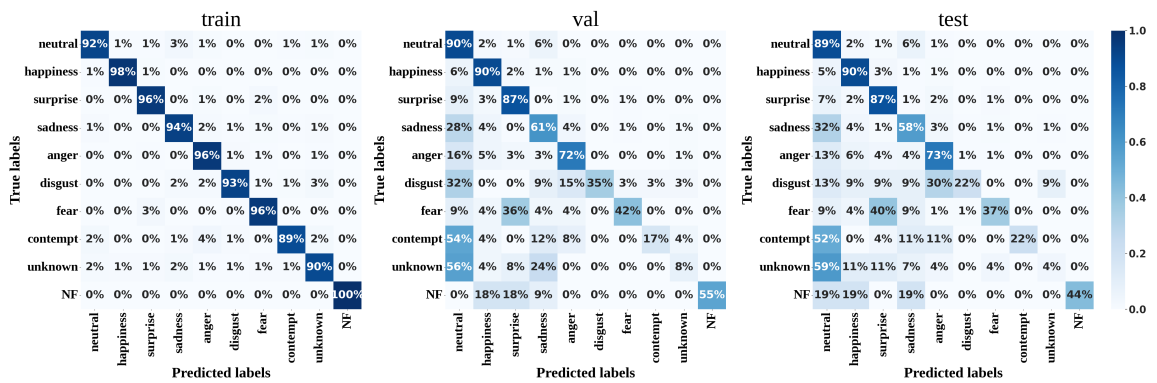


Рисунок 42 — Матрицы ошибок на обучающей (train), валидационной (val) и тестовой (test) выборках, нормализованные по истинным классам

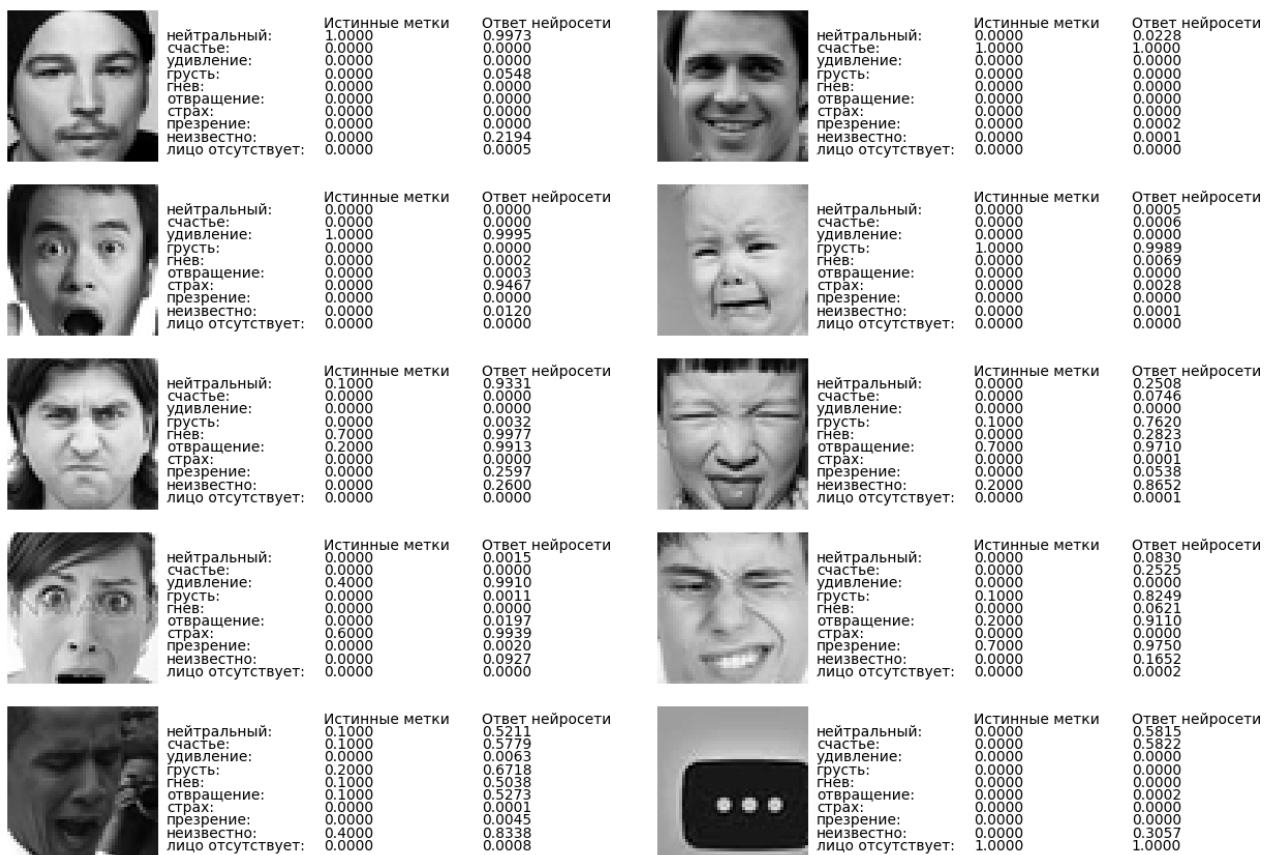


Рисунок 43 — Результаты распознавания изображений из тестовой выборки с помощью полученной нейронной сети

(unknown). Категории «презрение», «неизвестно» и «лицо отсутствует» были исключены из списка сохраняемых эмоций. Исходя из полученных результатов можно сделать вывод, что точности полученной СНС достаточно для разрабатываемого прототипа распознавателя движения и когнитивного состояния человека.

Для расчёта статистических параметров на основе параметров, извлечённых из видеопотока с помощью модуля видеоанализа, был разработан программный модуль расчета статистических параметров на основе видеоданных. Данный модуль выполняет обнаружение событий, вычисление статистических параметров и сохранение в базу данных:

- а) пульс (число ударов в минуту);
- б) частота моргания (мограний в секунду);
- в) событие фиксации глаза (координата, временная метка);
- г) событие движения глаза (конечная координата, временная метка);
- д) AVG_FIX – среднее время фиксации глаза;
- е) DEV_FIX – среднее квадратичное отклонение времени фиксации глаза;
- ж) AVG_DST – среднее расстояние между точками фиксации глаза;
- з) DEV_DST – среднее квадратичное отклонение расстояний между точками фиксации глаза;
- и) AVG_MOV – среднее время движения глаза.

В основе применённого метода определения пульса лежит принцип фотоплетизмографии, заключающийся в регистрации изменений, возникающих при наполнении мелких сосудов кровью в зависимости от фазы кардиоцикла. Существует ряд исследований, демонстрирующих возможность дистанционного измерения пульса человека на основе колебаний цвета кожи, регистрируемых видеокамерой при нормальных условиях освещения [37, 38, 39]. В разработанном прототипе определение пульса выполняется по изменению цвета кожи щёк. Для этого используется усреднённое по площади значение зелёного цветового канала. Применён следующий алгоритм.

- а) Выбор массива значений интенсивности зелёного канала и соответствующих им временных меток на указанном временном интервале.
- б) Интерполяция сигнала на постоянную частоту дискретизации, так как значения интенсивности измеряются в момент захвата кадра с камеры, а частота кадров может изменяться в зависимости от нагрузки на центральный процессор.
- в) Фильтрация резких перепадов по превышению порога среднее квадратичное отклонения.
- г) Удаление линейного тренда.
- д) Применение полосового КИХ-фильтра с полосой пропускания 0.7–4 Гц (42–240 уд/мин.).
- е) Применение окна Хэмминга.
- ж) Нормировка.
- з) Вычисление амплитудного спектра.
- и) Выбор диапазона частот 50–180 уд/мин., в котором идёт поиск пульса.
- к) Интерполяция спектра на фиксированный интервал частот 50–180 уд/мин. с шагом 0.5 уд/мин.
- л) Экспоненциальное усреднение спектра со спектрами, полученными на предыдущих временных интервалах.

м) Определение частоты сердцебиения как максимума в полученном спектре.

Метод фильтрации резких перепадов по превышению порога среднеквадратического отклонения был заимствован из работы [38], однако вместо фильтрации 5% сегментов сигнала с наибольшим среднеквадратическим отклонением, была применена фильтрация сегментов сигнала со значениями среднеквадратического отклонения, превышающими заданный порог. Метод заключается в разбиении сигнала на сегменты, в каждом из которых вычисляется среднеквадратическое отклонение. Далее из сигнала исключаются сегменты, среднеквадратическое отклонение для которых превышает заданный порог. Оставшиеся сегменты конкатенируются друг с другом таким образом, чтобы последний отсчёт i -го сегмента совпадал с первым отсчётом сегмента $i + 1$. Это позволяет исключить из сигнала резкие перепады, вызванные изменением освещённости либо поворотами головы. Фильтрация по превышению порога была выбрана, так как уровень полезного сигнала (пульсаций цвета кожи) является примерно постоянным, в то время как процент участков с резкими перепадами сигнала может быть разным.

Графики сигнала и спектров на основных этапах обработки приведены на рисунке 44. Полученный в результате спектр нормирован на единицу. На полученном в результате обработки спектре можно видеть хорошо выраженную гармонику на частоте около 70 ударов в минуту.

Частота морганий вычисляется раз в M секунд на временном интервале длиной N секунд как отношение количества событий моргания к длительности временного интервала. Событием моргания считается уменьшение расстояния между верхним и нижним веком меньше заданного порога.

Для обнаружения событий фиксации и движения глаза используются данные о положении зрачка по горизонтали и вертикали, полученные модулем видеоанализа. Обнаружение событий фиксации и движения глаза выполняется следующим образом. Движением считается изменение положения зрачка вдоль одной из осей (горизонтальное или вертикальное направление), превышающее заданное пороговое значение. Началом события фиксации глаза считается отсутствие движения между двумя последовательными кадрами. Окончанием события фиксации считается обнаружение движения, измеряемого относительно положения зрачка, зафиксированного в начале события фиксации. Соответственно, началом события движения считается окончание события фиксации, а окончанием события движения – начало события фиксации. Обнаруженные события сохраняются в базу данных. Для каждого события фиксируется его длительность, время окончания и координаты (для событий движения – конечные координаты).

На основе обнаруженных событий раз в M секунд на временном интервале длиной N секунд вычисляются и сохраняются в базу данных статистические параметры, описывающие динамику движения глаз:

а) AVG_FIX – среднее время фиксации глаза;

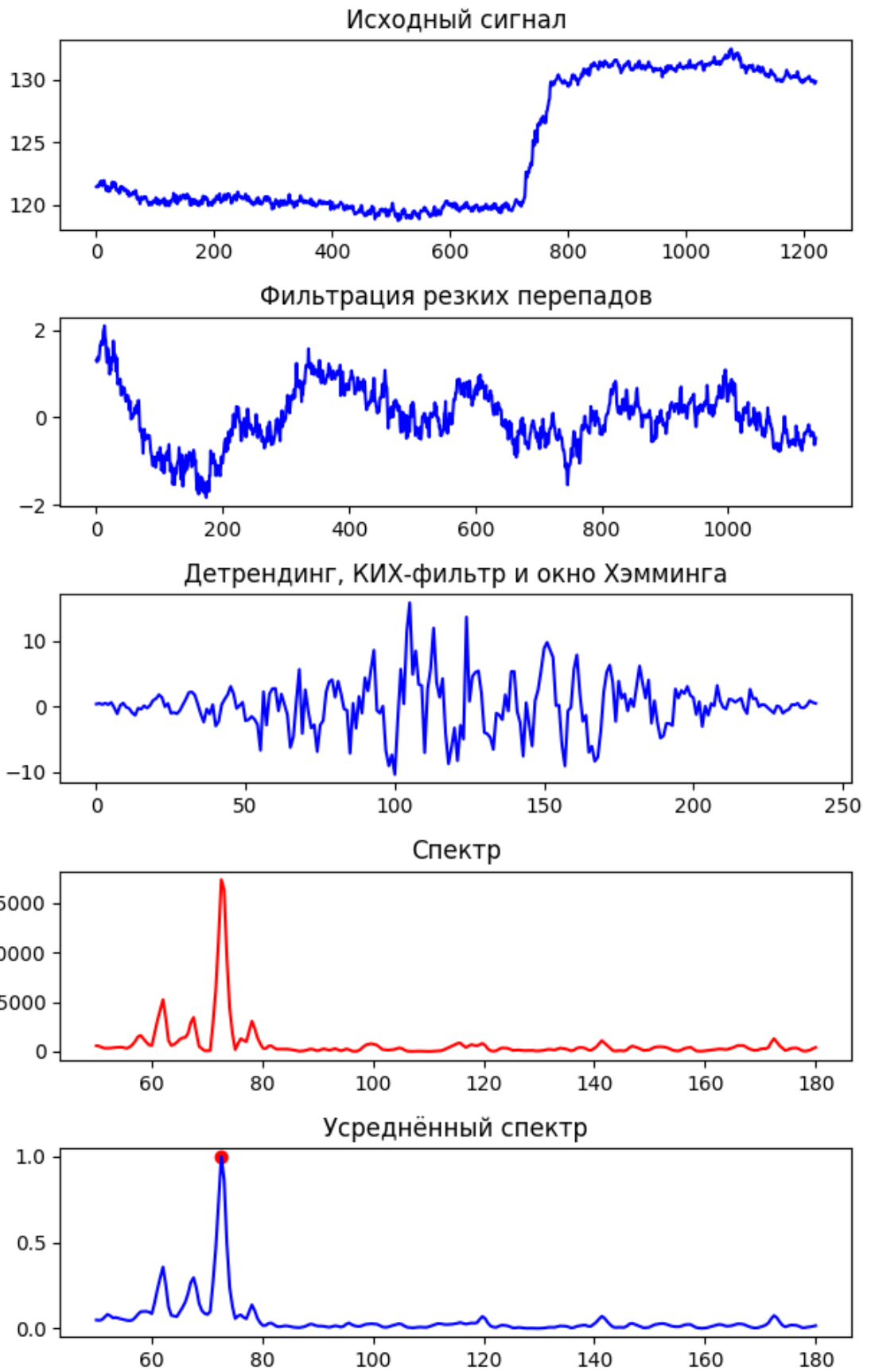


Рисунок 44 — Графики сигнала и спектров на основных этапах обработки при вычислении пульса

- б) DEV_FIX – среднеквадратичное отклонение времени фиксации глаза;
- в) AVG_DST – среднее расстояние между точками фиксации глаза;
- г) DEV_DST – среднеквадратичное отклонение расстояний между точками фиксации глаза;
- д) AVG_MOV – среднее время движения глаза.

Пороги перемещения зрачка по осям были введены из-за наличия шумов, связанных как с естественными колебаниями и дрожанием мышц человека, так и с дрожанием точек зрачка и сетки лица, определённых нейронной сетью, вызванным шумами камеры и колебаниями освещённости. Горизонтальное и вертикальное перемещения зрачка вычисляются в относительных единицах, нормированных на длину прямых, соединяющих точки на глазнице. Длина этих прямых различна и, следовательно, единицы измерения горизонтального и вертикального перемещения не равны между собой. Кроме того, перемещение зрачка по горизонтали, как правило, больше, чем по вертикали. Поэтому пороги перемещения зрачка по осям задаются отдельно. Вычисленные статистические метрики AVG_FIX, DEV_FIX, AVG_DST, DEV_DST, AVG_MOV позволяют оценить вовлеченность пользователя при чтении текста и поиске необходимой информации на экране монитора и используются в качестве входных данных для обучения и работы кластерной модели оценки вовлеченности.

Получение и анализ аудиоданных

Программный компонент аудиоанализа частично основана на алгоритме, представленном в работе [40], который использует спектральный метод определения частоты дыхания. В ходе разработки прототипа было установлено, что в случае использования микрофона с оголовным исполнением необходимо применять иные шаги фильтрации и параметры фильтров. В дополнение к спектральному методу в программном компоненте представлен расчет частоты дыхания интервальным методом. Кроме того повышена устойчивость работы программного модуля в условиях воздействия посторонних шумов (использование клавиатуры и компьютерной мыши).

Программный компонент аудиоанализа написан на языке программирования Python 3 и состоит из одного файла - `breath_analyzer.py`. Для вычисления частоты дыхания требуются аудиоданные, полученные в течение довольно продолжительного периода времени (не менее 60 секунд), так как частоты дыхания измеряется в единицах герц. Однако работа прототипа предполагает достаточно частое обновление собираемых данных, поэтому программный компонент аудиоанализа обрабатывает данные в режиме скользящего окна. В этом случае массив данных для аудиоанализа обновляется за счет отбрасывание части наиболее старых данных и добавления взамен равного по объему массива новых данных. В начале работы компонента выполняется инициализация, включающая в себя подключение к базе данных и присвоение значений параметрам обработки аудиоданных. Захват

аудиопотока осуществляется с помощью библиотеки `pyaudio` и производится с микрофона, определенного в операционной системе как микрофон по умолчанию (`default`). Оцифрованные данные поступают в буфер до тех пор пока длительность записанных данных не будет превышать 5 секунд, после этого данные записываются в массив для аудиоанализа. Если после этого объем данных становится достаточен (по длительности) для проведения аудиоанализа, Типичный вид оцифрованного сигнала с микрофона представлен на рисунке 45. Сигнал достаточно зашумлен, но на фоне общего шума можно зафиксировать максимумы, соответствующие моментам выдоха воздуха через нос. При этом в сигнале преобладают высокочастотные шумы, связанные по всей видимости с шумом усилительного тракта микрофона и наведенными помехами. Затем сигнал подвергается фильтрации

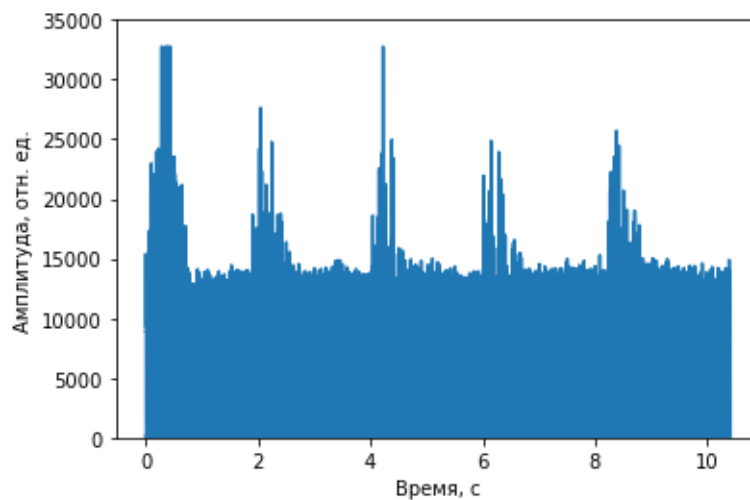


Рисунок 45 — Оцифрованный сигнал до обработки

цифровым полосовым фильтром Баттерворта 6-го порядка с граничными частотами 250 и 5000 Гц, после чего вычисляется огибающая звукового сигнала с помощью преобразования Гильберта. Это позволяет снизить высокочастотную и низкочастотную компоненту шума. Обработанный сигнал представлен на рисунке 46. После этого производится субдискретизация сигнала за счет выборки из всего массива только k -ых элементов. Это позволяет уменьшить объем анализируемых данных и снизить нагрузку на рабочую станцию при дальнейших расчетах. Затем производится фильтрация сигнала с помощью порогового фильтра, который необходим для уменьшения влияния посторонних шумов (от клавиатуры, компьютерной мыши или любых других резких аперриодических звуков). Следующим шагом является фильтрация сигнала с помощью Фурье-фильтра в диапазоне частот детектирования частоты дыхания 0.25-2.5 Гц. Результаты обработки сигнала представлены на рисунке 47. Указанный сигнал позволяет достаточно точно определять моменты выдохов воздуха и по ним вычислять частоту дыхания. Программный компонент аудиоанализа делает это с помощью двух методов: интервального и спектрального. Интервальный метод определения частоты дыхания использует среднее значение интервала между двумя последовательными выдохами. Для этого используется функция

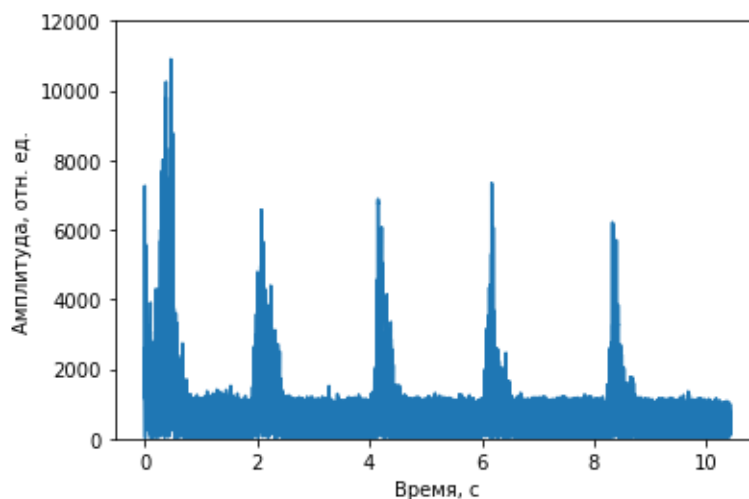


Рисунок 46 — Звуковой сигнал после обработки с помощью цифрового фильтра Баттерворта и преобразования Гильберта

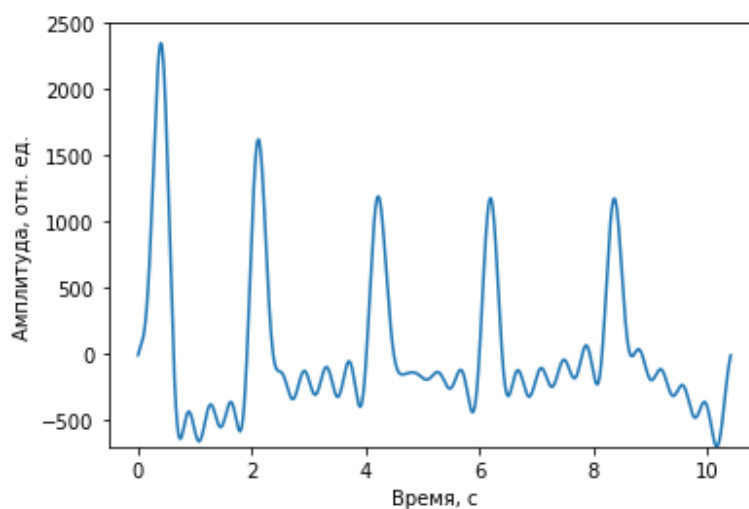


Рисунок 47 — Звуковой сигнал после обработки Фурье-фильтром

`find_peaks` библиотеки `scipy`. Определение минимальной высоты подсчета максимумов на графике делается за счет поимка минимального среднеквадратичного отклонения длительности интервалов между двумя последовательными выдохами. В этом методе из рассмотрения исключаются максимумы, чье амплитудное значение больше некоторого предустановленного порога, для снижения влияния посторонних шумов. На рисунке 48 показаны точки локальных максимумов, по которым определяются моменты выдохов при расчете частоты дыхания интервальным методом. Спектральный метод предполагает вычисление максимумов с использованием кривой спектральной плотности мощности. Частота дыхания в этом случае определяется по частоте максимума кривой на графике. На рисунке 49 представлена типовая зависимость спектральной плотности мощности.

Вычисленные значения записываются в таблицу "breath_rate" СУБД MySQL вместе с временной меткой измерений.

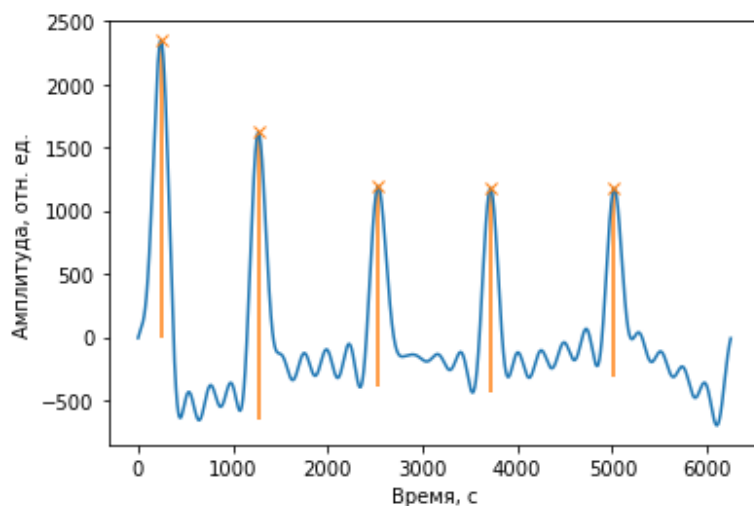


Рисунок 48 — Временные метки выдохов, найденные на графике при расчеты частоты дыхания интервальным методом

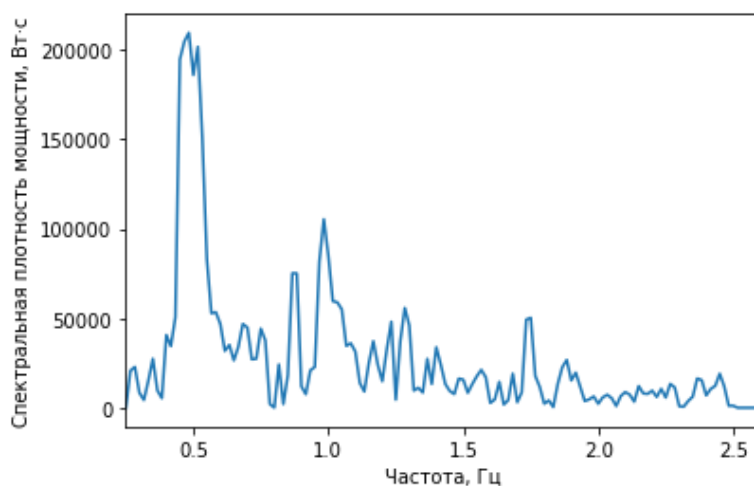


Рисунок 49 — Спектральная плотность мощности обработанного звукового сигнала

Получение и анализ данных с клавиатуры и компьютерной мыши

Для получения и анализа данных с клавиатуры и компьютерной мыши используются два программных компонента. Первый компонент (программный компонент сбора данных с клавиатуры и компьютерной мыши) используется для регистрации действий пользователя при работе с указанным оборудованием, а второй (программный компонент расчета статистических параметров на основе данных с клавиатуры и компьютерной мыши) - для статистической обработки полученных данных. Программные компоненты написаны на языке программирования Python и используют библиотеки `rupput`, `ipython` и `mysql_connector_python`. Для работы программного модуля требуется СУБД MySQL версии 8. Все необходимые настройки для работы программного модуля вынесены в отдельный конфигурационный файл.

Программный компонент сбора данных с клавиатуры и компьютерной мыши используется для сбора первичных данных от клавиатуры и компьютерной мыши, подключаемых

к рабочей станции через интерфейс USB. Для того, чтобы начать регистрацию событий клавиатуры и компьютерной мыши необходимо запустить интерпретатор языка python версии 3 и передать ему путь к файлу `key_log.py`. После запуска программного обеспечения происходит захват системных событий связанных с использованием клавиатуры и мыши с помощью библиотеки `rumpit`. Для клавиатуры захватываются события нажатия (`on_press`) и отпускание клавиши, а для компьютерной мыши - нажатие/отжатие левой и правой клавиш (`on_click`), а также движение курсора (`on_move`). В обработчике событий происходит запись данных о событии в таблицу `log` СУБД MySQL. Для клавиатуры записывается временная метка, тип события (нажатие или отпускание клавиши) и значение кода клавиши (в том числе для специальных клавиш `alt`, `ctrl`). Для компьютерной мыши записывается временная метка, тип события (нажатие левой кнопки мыши, нажатие правой кнопки мыши, движение курсора) и координаты курсора в момент события. Стоит отметить, что перемещение курсора сохраняется с частотой около 10 Герц (раз 0.1 секунды) для уменьшения нагрузки на СУБД.

Для определения эмоционального состояния необходимо проводить статистический анализ использования клавиатуры и компьютерной мыши, используя данные за некоторый интервал времени. В работах [29, 41, 42] представлены статистические метрики, которые помогают оценить уровень вовлеченности пользователя при использовании клавиатуры и компьютерной мыши. Использование клавиатуры оценивается по временным параметрам динамики нажатия клавиш. Среди характерных особенностей, на которые исследователи обычно обращают внимание стоит выделить следующие:

- а) Скорость набора текста;
- б) Длительность нажатие клавиши;
- в) Длительность задержки между последовательными нажатиями клавиш, в том числе внутри ди- и триграфов;
- г) Длительность печати ди- и триграфов;
- д) Длительность пауз между набором текста.

Использование компьютерной мыши оценивают по временным характеристикам нажатия клавиш мыши и по траектории курсора между двумя последовательными нажатиями клавиш мыши. Обычно рассматривают следующие особенности движения курсора:

- а) Расстояние между двумя последовательными нажатиями клавиш (эвклидово, путьевое);
- б) Скорость движение курсора;
- в) Ускорение движения курсора;
- г) Преимущественное направление движения курсора.

Для указанных характеристик использования клавиатуры и мыши в качестве статистической обработки используют функции нахождения: минимума, максимума, среднего значения, среднеквадратичного отклонения. При разработке прототипа были выбран ряд наиболее часто используемых статистических метрик, которые будут представлены далее.

Вычислением статистических метрик занимается программный компонент расчета статистических параметров на основе данных с клавиатуры и компьютерной мыши. Его запуск осуществляется с помощью интерпретатор языка python версии 3, которому необходимо передать ему путь к файлу `key_analysys.py`. Алгоритм работы статистической обработки исходных данных предполагает периодический запуск расчета (раз в 1-10 секунд), с указанием интервала времени в пределах которого производится поиск событий клавиатуры и компьютерной мыши. Этот временной интервал указывается в конфигурационных настройках программного модуля. Исходные данных для статистической обработки извлекаются из таблицы `log` СУБД MySQL. Полученный массив событий подвергается предварительной обработке (поиск ди- и триграфов, вычисление временных промежутков между нажатиями клавиш т.д.), результаты которой записываются в таблицу `temp_log_stat` СУБД MySQL, которая предварительно очищается от результатов предыдущих вычислений.

Затем на основе предварительно обработанных данных из таблицы `temp_log_stat` с помощью методов вычисления среднего и среднеквадратичного отклонения вычисляются следующие статистические метрики.

Для клавиатуры были выбраны следующие статистические параметры:

- а) `MN_SPD` – средняя скорость набора текста;
- б) `AVG_KD` – среднее время нажатия клавиши;
- в) `DEV_KD` – среднеквадратичное отклонение времени нажатия клавиши;
- г) `AVG_2K` – среднее время между нажатием двух клавиш;
- д) `DEV_2K` – среднеквадратичное отклонение времени между нажатием двух клавиш;
- е) `AVG_DG` – среднее длительность диграфа;
- ж) `DEV_DG` – среднеквадратичное отклонение длительности диграфа;
- з) `AVG_DG12K` – среднее время между двумя нажатиями клавиш внутри диграфа;
- и) `AVG_DG2K` – среднее время нажатия второй клавиши внутри диграфа;
- к) `AVG_TG` – среднее длительность триграфа;
- л) `AVG_TG3K` – среднее время нажатия 3 клавиши в триграфе;

Для компьютерной мыши были выбраны следующие статистические параметры:

- а) AVG_LKM: среднее время нажатия левой кнопки мыши;
- б) AVG_PKM – среднее время нажатия правой кнопки мыши;
- в) AVG_2LKM – среднее время между двумя нажатиями левой кнопки мыши;
- г) C_LKM – количество нажатий левой кнопки мыши в течение временного отрезка;
- д) C_PKM – количество нажатий правой кнопки мыши в течение временного отрезка;
- е) AVG_DST – среднее евклидово расстояние между двумя последовательными нажатиями кнопки мыши;
- ж) AVG_SPD – средняя скорость движения курсора между двумя последовательными нажатиями кнопки мыши;
- з) AVG_DST2 – средняя длина пути курсора между двумя последовательными нажатиями кнопки мыши;
- и) AVG_SPD2 – средняя путевая скорость - отношение путевого расстояния к времени между двумя последовательными нажатиями кнопки мыши;
- к) AVG_DST_RT – среднее отношение путевого расстояния к евклидову расстоянию для двух последовательных нажатий кнопки мыши;
- л) DEV_DST_RT – среднеквадратичное отклонение отношения путевого расстояния к евклидову расстоянию для двух последовательных нажатий кнопки мыши

Стоит отметить, что путевое расстояние между двумя последовательными нажатиями кнопки компьютерной мыши вычисляется как сумма евклидовых расстояний между событиями движения курсора.

Вычисленные статистические метрики использования клавиатуры и компьютерной мыши характеризуют вовлеченность пользователя в рабочий процесс и применяются в качестве входных данных для обучения и работы кластерной модели оценки вовлеченности. Для этого вычисленные значения этих метрик в месте с текущей меткой времени сохраняются в таблицах `stat_kdb_a` (метрики использования клавиатуры) и `stat_ms_a` (метрики использования компьютерной мыши) СУБД MySQL.

Эмуляция данных с бионического костюма

TODO: Димитров, расписать про какие есть майки и какие параметры снимают, привязать тезис Рассмотрим некоторые решения умного текстиля, которые представляют собой надеваемый на тело человека текстильный объект с закрепленными на нем датчиками и устройством сбора информации с датчиков.

Майка `hexoskin` [<https://hexoskin.com>] от канадской компании `Carre` предоставляет следующий набор датчиков:

- а) три электрода для регистрации ЭКГ, двух канальный датчик для регистрации дыхания и трехосевой акселерометр для определения положения тела. Два отведения для ЭКГ находятся на уровне груди (слева и справа), одно отведение — снизу справа на уровне брюшной полости;
- б) датчик дыхания, который закреплён на уровне груди.
- в) акселерометр находится посередине майки.

Майка hexoskin имеет всего три отведения (в отличие от классической электрокардиограммы, измеряемой в учреждениях медицинской помощи, где на пациента закрепляют 12 отведений). Это не позволяет ей строить полноценную ЭКГ, однако этого достаточно, чтобы измерять ритмы работы сердца, на основе которых можно диагностировать некоторые виды заболеваний, например, фибрилляцию предсердий (один из видов аритмий).

Датчик анализа частоты дыхательных движений измеряет объем груди в соответствии с этим можно делать предположения о вдыхании/выдыхании воздуха человеком. Помимо самой частоты дыхания имеется возможность расчета объема получаемого воздуха.

Датчик акселерометра позволяет регистрировать изменения перемещения тела человека по трем плоскостям. Частая смена положений тела может свидетельствовать об отсутствии концентрации человека на процессе, а следовательно и не вовлеченности в него. Для определения изменения положения тела предлагается использовать фильтр, регистрирующий только достаточно большие перемещения (например, откидывание на спинку стула/кресла, пересаживание с одного края стула на другой и др.). Параметр частой смены предлагается установить экспериментальным путём.

Архитектура взаимодействия майки hexoskin с компьютером сбора информации представлена 50. На майке имеется блок сбора информации, который имеет Bluetooth Low Energy (BLE) интерфейс, что позволяет подключать его к устройству с модулем BLE. Поэтому на компьютере сбора информации предполагается разместить модуль BLE и получать данные прямо на компьютер, где уже кластерная модель будет работать с данными.

TODO: Беляев, связать измеряемые параметры с параметрами, обычно снимаемыми с майки

Модель оценки вовлеченности человека

TODO: Беляев, добавить воды из лит. обзора TODO: Димитров, расписать работы программного модуля в общем: извлечение значений, кластерный анализ, запись обратно в базу, какие библиотеки используются, собственно код основного скрипта, но без указания конкретных полей Для расчета оценки вовлеченности используются уже предобу-

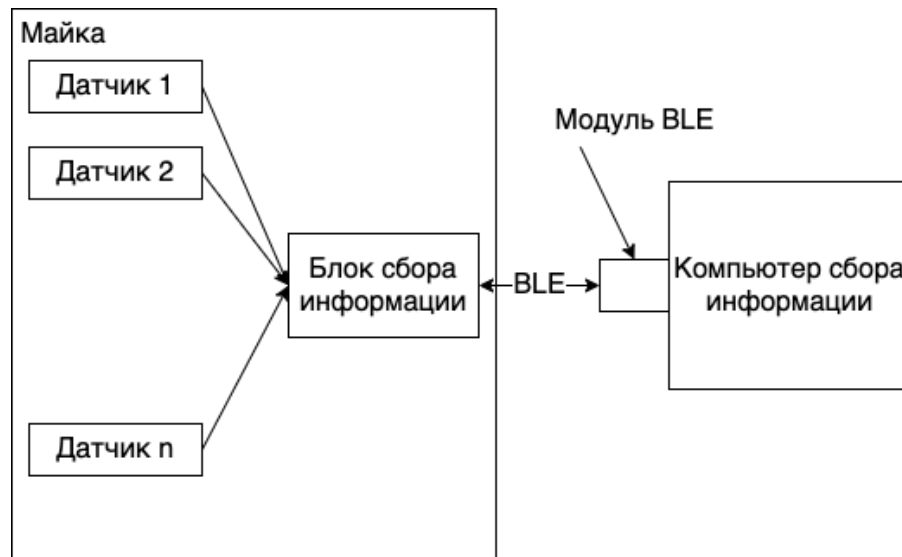


Рисунок 50 — Принципиальная схема получения данных с майки hexoskin

ченные модели для кластерного анализа. Основная программа работает по следующим шагам:

- а) Загрузить предобученные модели по клавиатуре, мыши и движению глаз.
- б) Извлечь последние записи из базы данных из таблиц статистики клавиатуры, мыши и движению глаз.
- в) Применить соответствующую модель к соответствующим данным для вычисления вовлеченности. В результате получим 3 значения вовлеченности.
- г) Рассчитать общую вовлеченность.

Рассмотрим выбор параметров кластеризации при анализе модели кластерного анализа для каждого из видов данных.

Для данных с клавиатуры наилучшие разбиения получаются при интервале по 2 минуты. Список параметров используемых для кластеризации: AVG_KD, DEV_KD, DEV_2K, AVG_DG, AVG_DG2K. Количество кластеров 4. Результат предсказаний отображается на вовлеченность следующим образом (слева результат, который выдает модель, справа уровень вовлеченности):

0 => 3
 1 => 1
 2 => 2
 3 => 0

Уровень вовлеченности от 0 до 3 соответствует уровням вовлеченности от более сильного к более слабому.

Для данных, получаемых с мыши наилучшие разбиения получаются при интервале по 2 минуты. Список параметров используемых для кластеризации: AVG_LKM, AVG_2LKM, C_LKM', AVG_DST, AVG_SPD, AVG_SPD2. Количество кластеров 4. Ре-

зультат предсказаний отображается на вовлеченность следующим образом (слева результат, который выдает модель, справа уровень вовлеченности):

0 => 1
1 => 3
2 => 0
3 => 2

Для движения глаз наилучшие параметры получились для $thX = 0.08$, $thY = 0.02$ при 90 секундах. Список параметров используемых для кластеризации: AVG_FIX, DEV_FIX, AVG_DST, DEV_DST. Количество кластеров 4. Результат предсказаний отображается на вовлеченность следующим образом (слева результат, который выдает модель, справа уровень вовлеченности):

0 => 2
1 => 3
2 => 1
3 => 0

Расчет общего уровня вовлеченности в процентах выполняется по следующей формуле:

$$((I_{kbd} + I_{ms} + I_{eye})/9) * 100,$$

где I_{kbd} - уровень вовлеченности для статистических данных с клавиатуры, I_{ms} - уровень вовлеченности для статистических данных с мыши, I_{eye} - уровень вовлеченности для статистических данных движения глаз.

Таким образом после расчета всех метрик в базу данных в таблицу *involvement* сохраняются 5 полей:

- а) keyboard — уровень вовлеченности по данным с клавиатуры;
- б) mouse — уровень вовлеченности по данным с мыши;
- в) eye — уровень вовлеченности по данным движения глаз;
- г) aggregate — общий уровень вовлеченности;
- д) timestamp – момент времени вычисления характеристик.

Пользовательский интерфейс

TODO: Ермаков

Пользовательский интерфейс предназначен для отображения параметров с каждого из потока данных, а так же рассчитанных статистических характеристик.

Для построения, отображения и обновления графиков в режиме реального времени используется Python-библиотека Dash. Параметры подключения к источнику данных для отображения определяются конфигурационным файлом. Список параметров для

отображения определяется автоматически после подключения к источнику данных. Пользовательский интерфейс доступен в интернет-браузере по адресу, предоставляемым терминальным выводом запущенного приложения в формате ip:port.

Пользователю доступен следующий набор элементов:

- а) Выпадающий список со множественным выбором, содержащий набор параметров для построения графика. Пользователь может выбрать несколько параметров для построения графиков.
- б) Слайдер выбора количества точек для построения графика.
- в) Слайдер выбора частоты опроса базы данных. Данный параметр определяет частоту обновления графиков.
- г) Флаг использования временной метки в качестве значения по оси X.
- д) Вывод последнего значения из запроса формате db_name/column_name: value.
- е) Область построения графиков. Количество графиков определяется выбранными параметрами из выпадающего списка.

Пример пользовательского интерфейса изображен на рисунках 51 и 52 соответственно. Количество точек для построения графика варьируется от 5 до 1000 с шагом в 5 точек. Время опроса БД варьируется от 100 до 1000 миллисекунд, с шагом в 100 миллисекунд. Для каждого графика применяется автомасштабирование. Дополнительно выводится последнее значение из результата запроса к БД по всем выбранным из выпадающего списка параметрам.

2.1.3 Испытания программно-аппаратного прототипа

2.1.3.1 Построения модели оценки вовлеченности человека

TODO: Беяев, указать каак отсеивались признаки, какие выбраны, скомпоновать

Для оценки оптимальных значений порогов обнаружения движения зрачка и порога расстояния между верхним и нижним веком для обнаружения моргания был собран набор данных при чтении текста с экрана в течении одного часа. Полученные гистограммы распределения перемещения зрачка от кадра к кадру представлены на рисунках 53 и 54. Полученный график изменения расстояния между верхним и нижним веком с течением времени приведён на рисунке 55.

На полученных гистограммах видно линейное возрастание плотности распределения, сменяющееся экспоненциальным спадом. Для перемещения по горизонтали максимум распределения наблюдается в районе значения 0.08, для перемещения по вертикали – в районе значения 0.02. Данные значения были выбраны в качестве пороговых для обнаружения движения зрачка. На графике изменения расстояния между верхним и нижним



Рисунок 51 — Пользовательский интерфейс

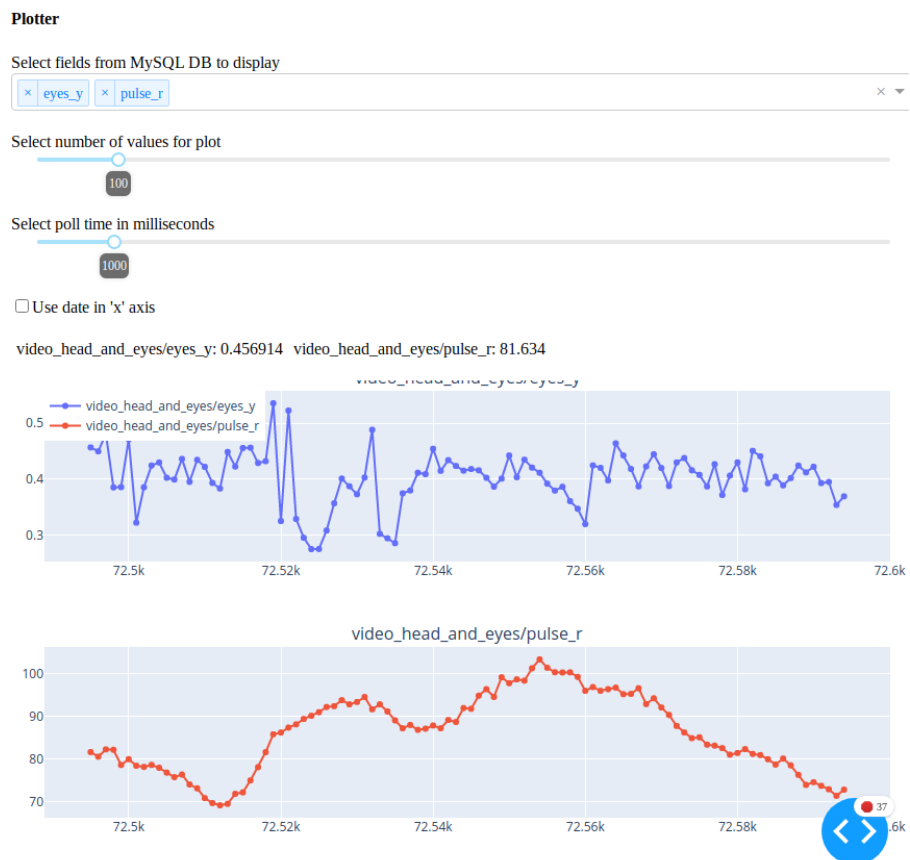


Рисунок 52 — Пользовательский интерфейс, отображение двух графиков

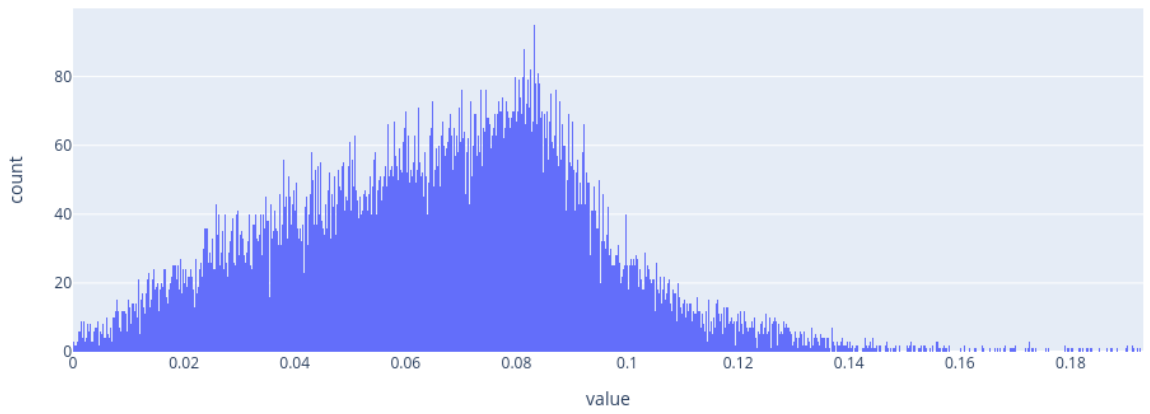


Рисунок 53 — Гистограмма распределения перемещения зрчка по горизонтали от кадра к кадру

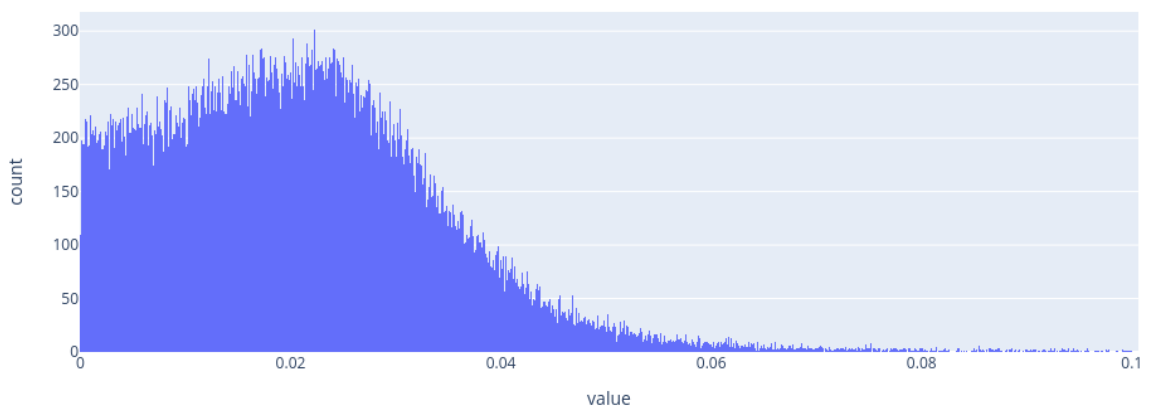


Рисунок 54 — Гистограмма распределения перемещения зрчка по вертикали от кадра к кадру

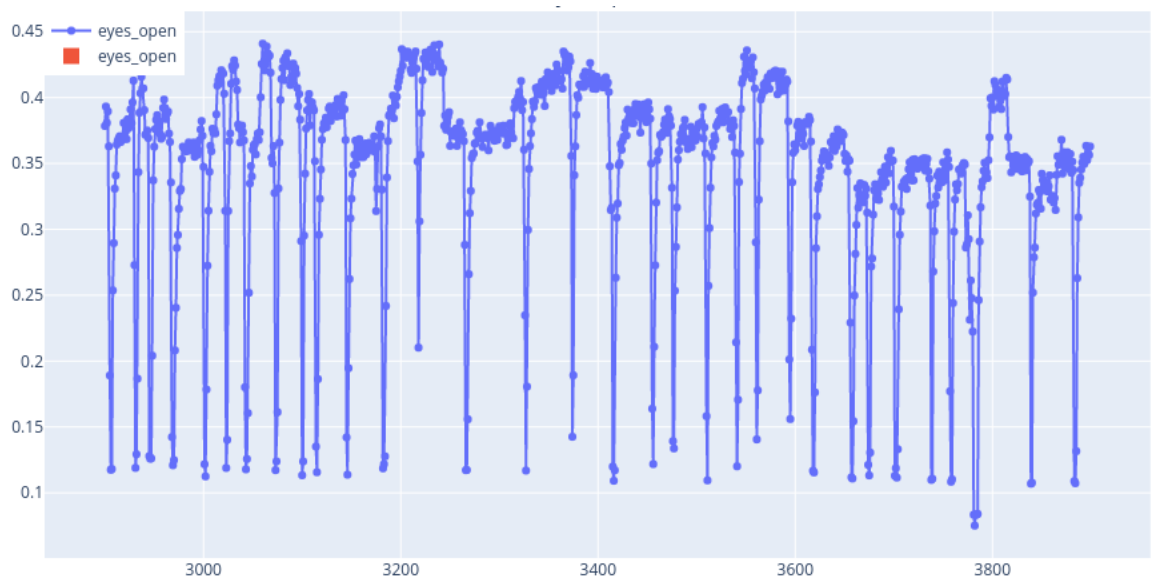


Рисунок 55 — График изменения расстояния между верхним и нижним веком с течением времени

веком с течением времени хорошо прослеживаются кратковременные резкие падения, соответствующие морганиям. Исходя из полученных данных значение порога было выбрано равным 0.3.

2.1.3.2 Результаты испытаний программно-аппаратного прототипа

TODO: Указать результаты испытаний по ПМИ

2.1.4 Альтернативные методы оценки вовлеченности человека

Раздел посвящен исследованию методики оценки эмоционального состояния человека по его манере печатать текст на клавиатуре с помощью нейронной сети LogNNet. Для обучения и тестирования сети использовалась база данных EmoSurv.

Описание базы EmoSurv

База EmoSurv применяется для тестирования нейронных алгоритмов распознавать эмоциональное состояние человека по его манере печатать текст на клавиатуре. База данных содержит набор данных о нажатии кнопок клавиатуры с сопоставленными метками эмоций. Данные о времени и частоте нажатий записывались во время набора произвольного или фиксированного текста после того, как у участников возникали определенные эмоции. Градация эмоций имеет 5 состояний: гнев, счастье, спокойствие, печаль и нейтральное состояние. Изначально данные собирались пока участник находился в нейтральном состоянии. Затем участник смотрел эмоциональное видео, и как только у него возникала определенная эмоция, он набирал фиксированный или свободный текст.

База содержит 4 файла .csv:

- а) Файл 1: Содержит набор данных собранный во время печати фиксированного текста, и включает в себя следующие колонки: идентификатор пользователя, индекс эмоций, код клавиши, временные интервалы D1U1, D1U2, D1D2, U1D2, U1U2, D1U3, D1D3 и ответы пользователя если они подразумевались.
- б) Файл 2: Содержит набор данных собранный во время печати произвольного текста.
- в) Файл 3: Содержит индикаторы, связанные с частотой набора.
- г) Файл 4: Содержит демографические данные, такие как идентификатор пользователя, печать двумя или одной рукой, печать одним пальцем, двумя или полной кистью, время провозждения за компьютером, возраст -интервал, пол, статус, степень и страну.

Временные интервалы D1U1, D1U2, D1D2, U1D2, U1U2, D1U3, D1D3 соответствуют интервалам характерным при наборе триграфа (трех последовательных символов) определяемые по схеме, показанной на рисунке 56.

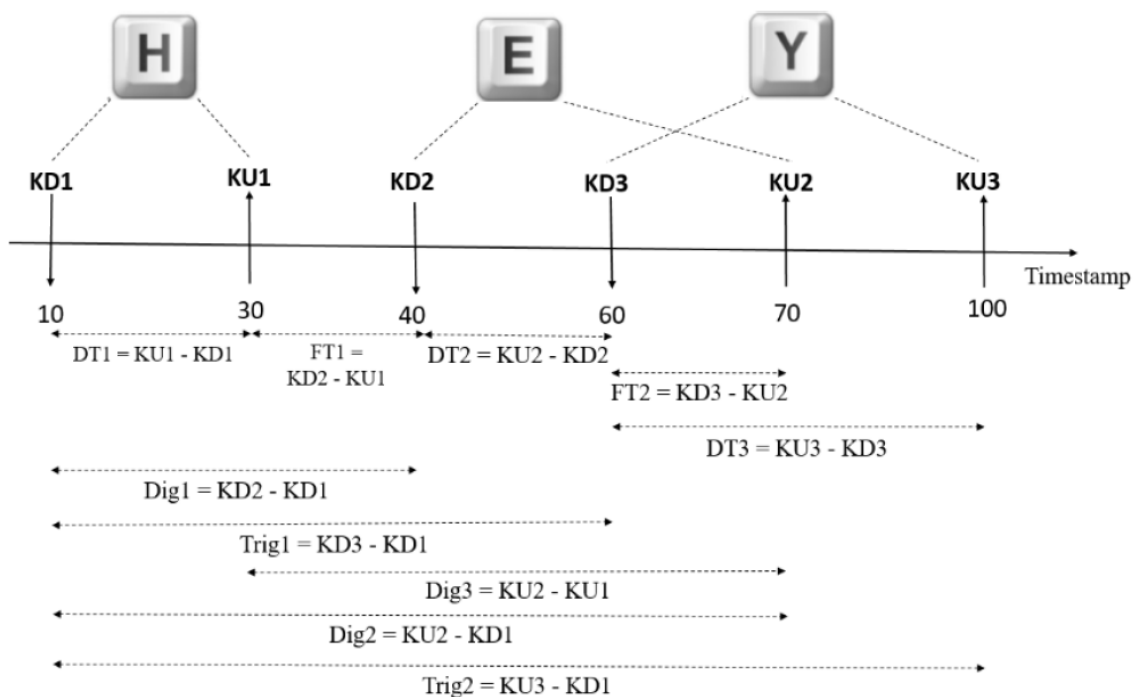


Рисунок 56 — Принцип определения временных интервалов при печати триграфа

Методика реорганизации базы данных, балансировка

Для обучения и тестирования сети LogNNNet база EmoSurv была представлена в нескольких вариантах с постепенным увеличением набора признаков. Балансировка обучающего множества проводилась по методике, описанной в параграфе ??

Первый вариант обучающего множества состоял из 10 столбцов, и содержал параметры, указанные в Таблице 9. В таблице приведены примеры первых пяти элементов множества. Множество состояло из триграфа символов набираемые в порядке Symbol 1, Symbol 2, Symbol 3, и соответствующих значений D1U1, D1U2, D1D2, U1D2, U1U2, D1U3, D1D3. Номер эмоции также указан в таблице. Соответствие номера и эмоции определялось по следующим правилам: 1 - H (for Happy - счастье), 2 - S (for Sad - печаль), 3 - A (for Angry - гнев), 4 - C (for Calm- спокойствие), и 5 - N (for Neutral state - нейтральное состояние). Всего множество содержало 106819 векторов.

Таблица 6 — Вариант-1 обучающего множества

1	2	3	4	5	6	7	8	9	10	
Symbol	Symbol	Symbol								
3	2	1	D1U1	D1U2	D1D2	U1D2	U1U2	D1U3	D1D3	Emotion
99	32	101	94	404	338	244	310	1644	1554	1
101	104	116	101	323	212	111	222	422	316	2
107	97	106	81	246	149	68	165	409	327	3

Продолжение таблицы 9

1	2	3	4	5	6	7	8	9	10	
117	97	101	78	307	218	140	229	512	443	4
99	110	111	90	2556	2479	2389	2466	2737	2610	5

Второй вариант обучающего множества состоял из 15 столбцов, и содержал помимо параметров из Варианта-1, еще 5 параметров, характеризующих пользователя и его стиль печати, указанные в Таблице 7. Всего множество содержало 106819 векторов.

Значения определялись по следующим условиям:

- а) typeWith - печать одной (1) или двумя руками (2);
- б) typistType - печать одним пальцем (1), двумя пальцами (2) и полной кистью (3);
- в) pcTimeAverage - время проводимое за компьютером: менее одного часа в день (1), более часа но менее трех в день (2), больше трех часов в день (3);
- г) ageRange - возраст испытуемого 16-19 лет (1), 20-29 лет (2), 30-39 лет (3), более 50 лет (4);
- д) gender - пол: женский (1), мужской (2).

Таблица 7 — Вариант-2 обучающего множества (добавление к Варианту-1)

11	12	13	14	15	
typeWith	typistType	pcTimeAverage	ageRange	gender	Emotion
2	3	3	2	1	1
2	3	3	3	2	2
2	3	3	3	2	3
2	3	3	2	1	5

Третий вариант обучающего множества состоял из 25 столбцов (Таблица 8), и содержал помимо параметров из Варианта-2, еще 10 параметров, характеризующих положение кнопок, нажатых пользователем и скорость перемещения пальцев между кнопками. Данное дополнение является импровизацией автора отчета, в надежде на то, что добавленные признаки увеличат точность классификации. Всего множество содержало 106819 векторов.

Таблица 8 — Вариант-3 обучающего множества (добавление к Варианту-2)

16	17	18	19	20	21	22	23	24	25	
V2	V1	S2	S1	X3	Y3	X2	Y2	X1	Y1	Emotion
35.6237	307.834	40.9672	75.1115	94.6877	34.6924	131.1627	16.0406	81.0556	71.9961	1
-9084.1	268.7829	63.5887	29.8349	81.0556	71.9961	141.8473	53.3443	118.2675	71.6231	2
1623.848	1657.956	131.5317	112.741	180.1646	53.3443	48.6334	52.9712	161.3744	52.9712	3
806.4728	268.5129	109.6803	37.5918	156.5847	72.3691	48.6334	52.9712	81.0556	71.9961	4
1043.989	23.9954	56.3754	57.3249	94.6877	34.6924	151.0582	35.4385	194.9019	72.36915	5

Координаты кнопок клавиатуры вычислялись по макету клавиатуры, представленному на рисунке 57.

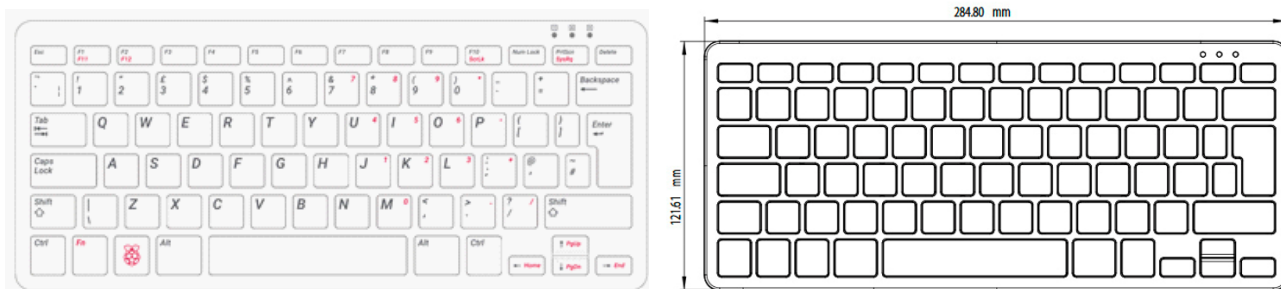


Рисунок 57 — Макет клавиатуры и его размеры

TODO: Марченков, вставить оставшуюся часть отчета Величко

2.1.5 Перспективы развития программно-аппаратного прототипа

2.2 Программно-аппаратный прототип распознавателя приоритетных событий из множественных источников при движении робота на основе адаптивной стратегии оперативного отбора информации о текущей ситуации

2.2.1 Проблема распознавания приоритетных событий программно-аппаратных комплексов по данным из множественных источников.

Автономные роботизированные устройства, в т.ч. автономные мобильные устройства (АМУ) являются сложными многоцелевыми программно-аппаратными комплексами и находят применение в широком спектре приложений во многих видах экономической деятельности и персональной активности [43], в частности: в промышленности, торговле, природоохранных проектах, образовании, сельском хозяйстве, деятельности домохозяйств и ряде других [44, 45]. Основные усилия исследователей и разработчиков в настоящее время направлены, в том числе, на разработку средств и методов позволяющих таким устройствам действовать в частично или полностью незнакомой среде автономно или с ограниченным вмешательством удаленного оператора. В случае если АМУ оперирует в известной среде, которая описана с помощью 2D, 3D карты, графа или иным образом, тем не менее, такая среда может изменяться за счет деятельности человека, деятельности других автономных устройств или случайных факторов. Заметим также, что в работе самого АМУ могут происходить сбои, независимо или в силу агрессивных факторов окружающей среды. Такие изменения в подавляющем большинстве случаев являются плохо предсказуемыми, однако АМУ должно идентифицировать их и, впоследствии, генерировать реакцию/управление.

С целью извлечения сведений о свойствах окружения и для осуществления навигации в полностью или частично неизвестной среде АМУ используют данные микро-электро-механических сенсоров (МЕМС), которые могут поставлять разнообразную информацию, как об объектах окружающей среды, так и о состоянии подсистем АМУ. В настоящее время существует большое разнообразие МЕМС устройств, которые поставляют обширные разнородные объемы исходных данных. В литературе приняты различные способы классификации сенсоров. Например:

- а) основанные на природе данных, поставляемых сенсором (пространственные, электромагнитные, механические);
- б) основанные на природе активности сенсора (внутренние — уровень заряда аккумулятора, внешние — расстояние до препятствия, активные — сонар, пассивные — камера)
- в) ряд других.

Данные, поставляемые МЕМС устройствами, избыточны, подвержены шумам и искажениям, поставляются, как правило, с весьма высокой частотой и содержат большие объемы как данных о состоянии АМУ и окружающей среды, так и шумов, ошибок, несущественных сведений о незначительных флуктуациях системы. Поэтому первичные данные подвергаются обработке перед использованием их в подсистемах распознавания, построения выводов и управления. С этой целью, как правило, используются известные методы обработки цифровых сигналов и временных рядов. Основная цель этого этапа - скорректировать ошибки и устранить неопределенность данных МЕМС. Затем, обработанные данные рассматриваются как источники информации для модулей интеграции данных, которые используют ряд методов, призванных идентифицировать события в АМУ и среде и построить управление АМУ. Таким образом можно выделить две основные проблемы обработки данных МЕМС:

- а) первичные данные МЕМС содержат неопределенность из-за их низкой точности и влияния внешних факторов, вносящих погрешность в измерения требуемых величин, т.н. шум;
- б) значимость элементов соответствующих временных рядов варьируется. Некоторые из этих данных имеют критическое значение для целостности АМУ и/или окружающей среды, некоторые — имеют существенное значение для правильной работы системы и выполнения плановых заданий, некоторые — незначительны или бесполезны для целей идентификации и управления.

Как было отмечено выше, для решения первой задачи используются методы анализа цифровых сигналов и анализа временных рядов. Наиболее популярными являются различные сглаживающие фильтры, такие как арифметическое и геометрическое скользящее среднее, медианный фильтр и некоторые другие. Также в литературе широко применяются фильтр Калмана и последовательный метод Монте-Карло (англ. Particle Filter) [46].

Заметим, что многие производители МЕМС встраивают первичные фильтры в свои продукты, однако не всегда должным образом документируют их.

После сглаживания в алгоритмах интеграции данных данные нескольких сенсоров комбинируются с информацией соответствующих баз данных, для построения конкретных выводов о событиях в системе и в окружающей среде [47]. Заметим, что некоторые авторы относят фильтр Калмана [48] к методам интеграции данных, так как он также может использоваться в задачах локализации и навигации.

В настоящее время в подсистемах интеграции данных широко используются нечеткие интеллектуальные алгоритмы для распознавания событий АМУ и прогнозирования состояния системы. В частности это методы нейронных сетей [49], и методы машинного обучения.

Заметим, что несмотря на высокую интенсивности исследований методы, направленные на решение первой проблемы, в подавляющем большинстве случаев имеют высокую вычислительную сложность, а методы решения второй проблем весьма ограничены и сводятся к выбору сенсоров и их месторасположению на АМУ. При этом само АМУ не обладает большими вычислительными мощностями и имеет ограниченный заряд батареи. При этом все операции, связанные с механическими манипуляциями и передачей данных по беспроводным сетям имеют наиболее высокий уровень энергопотребления.

В связи с этим в настоящем разделе мы рассматриваем двух-этапный метод распознавания приоритетных событий. На первом этапе для обработки первичных данных МЕМС используется нечеткий метод интеллектуального отбора данных (НМИОД), который на основе адаптивной стратегии [50], [51] выбирает из потока данных МЕМС те результаты измерений, которые могут нести потенциально новые сведения о состоянии системы и среды. На втором этапе данные, отобранные НМИОД используются алгоритмом распознавания приоритетных событий, который основан на графе состояний АМУ и функциях переходов между ними.

2.2.2 Двух-этапный метода распознавания приоритетных событий

Отдельный сенсор (например, камера или инерциальный модуль) как правило не может предоставить достаточно данных для целей управления и не является достаточно надежным. Если этот единственный сенсор выйдет из строя, то система немедленно лишится внешних источников данных и, возможно части данных о состоянии собственных подсистем. Поэтому АМУ оснащают набором сенсоров, которые в свою очередь порождают весьма обширные массивы данных. Число событий в системе среда-АМУ, которые могут быть идентифицированы по этим данным также весьма велико. Между тем, введенные выше атрибуты событий (критическое, существенное, бесполезное) не могут быть ассоциированы с одним отдельным МЕМС. Например, рассмотрим уровень заряда батареи АМУ. Если батарея заряжена, то нет необходимости опрашивать данные об уровне

заряда часто, но если уровень заряда низкий или он снижается быстрее, чем ожидалось, то уровень заряда следует контролировать на регулярной основе и, может быть, перепланировать работу устройства. Таким образом данные одного и того же сенсора могут иметь различные атрибуты в различных контекстах.

Любой МЕМС может поставлять как критические, так и бесполезные данные. При этом атрибуты их значимости и политика измерений взаимосвязаны и зависят от нашей оценки уровня неопределенности состояния АМУ и окружающей среды. Если уровень заряда батареи высок, мы ожидаем надежной работы системы и уровень оптимизма при оценке уровня неопределенности высокий, потребность в управлении - низкая. Если уровень заряда близок к критическому, то уровень оптимизма при оценке уровня неопределенности - низкий и все действия АМУ требуют дополнительного планирования и анализа.

Поэтому для извлечения информации, существенной для управления и навигации, мы предлагаем двух этапную схему обработки первичных данных, которая использует нечеткие методы интеллектуального анализа на обоих этапах и позволяет сократить вычислительные затраты на обработку данных. На первом этапе выбираются данные, которые потенциально могут нести информацию о приближении изменений в состоянии АМУ или окружающей его среды, означать наступление событий в системе, определенных как приоритетные. Первичные данные, которые не прошли отбор на первом этапе отбрасываются и в дальнейшем не используются. Для отбора потенциально значимых данных используется адаптивная стратегия, которая определяет баланс между оптимистическими и пессимистическими ожиданиями уровня неопределенности и позволяет сократить объем трафика между модулем отбора данных и модулями интеграции данных и распознавания событий. На втором этапе данные, прошедшие отбор используются для распознавания т. н. приоритетных событий в работе АМУ. Нами определена графовая модель состояний системы и функции перехода между этими состояниями, которые в качестве аргументов принимают первичные данные, прошедшие отбор.

2.2.2.1 Нечеткий алгоритм интеллектуального отбора данных

Для построения абстрактной модели сенсора используем метод, описанный в работе [52]. Будем рассматривать сенсор как отображение $S_m \mapsto S_n$, где $S_m \subset \mathbb{R}^m$ и $S_n \subset \mathbb{R}^n$. В подавляющем большинстве случаев $|S_m| > |S_n|$ при этом $m > n$. Соотношение размерностей в этом отображении обусловлено необходимостью интеграции непрерывных и дискретных величин в рамках современных кибер-физических систем. Как правило, физические величины, подлежащие измерению с помощью МЕМС, являются непрерывными, например: ускорение, скорость, расстояние, температура. Однако дальнейшая обработка данных измерений в вычислительной системе предполагает дискретизацию, ввиду использования двоичного представления этих данных в памяти в доступной разрядной сетке. Поэтому, фактически, интервалы значений исходной физической величины будут отобра-

жаться на выходе МЕМС в одно значение. Первичные фильтры, применяемые производителями, могут произвести дополнительное преобразование элементов пространства S_n .

Рассмотрим два элемента $x, y \in S_n$ и определим норму $\|x - y\| \in \mathbb{R}$, как расстояние между двумя значениями, полученные МЕМС. Заметим, что конкретное определение нормы зависит от природы измеряемой величины и должно быть введено отдельно для каждого потока первичных данных. Будем считать, что эта норма аддитивна и $\|ax\| = a\|x\|$, $a \in \mathbb{R}$, $x \in S_n$. Теперь определим последовательность $s^i = \{s_0^i, s_1^i, \dots\}$, $s_k^i \in S_n$ данных, предоставленных сенсором i и последовательность $t^i = \{t_0^i, t_1^i, \dots\}$ временных меток, соответствующих элементам последовательности s^i .

С целью дальнейшего анализа определим меру неопределенности в работе АМУ и окружающей среды, основанную на первичных данных, представленных в последовательности s^i . Представим одно измерение сенсора как сумму

$$\|s_k\| = \|\sigma_k\| + r_k,$$

где σ_k истинное значение измеряемой переменной и $r_k \in \mathbb{R}$ случайная ошибка измерений.

Если измеряемая переменная остается неизменной или изменяется незначительно, это означает, что нет необходимости передавать повторно эти новые значения модулям интеграции данных — они не несут новой информации о состоянии системы. Пусть $\mathbb{E}[r_k] = 0$. Это предположение означает, что МЕМС калиброван правильно. Тогда $\mathbb{E}[s_k] = \mathbb{E}[\sigma_k]$. Если функция распределения r_k непрерывна и симметрична, то

$$\mathbb{P}\{r_k > 0\} = \mathbb{P}\{r_k < 0\} = 1/2.$$

Обозначим p_m вероятность, что m значений r_n подряд больше или меньше нуля. Тогда $p_m = 1/2^m$. То есть, наблюдая такую последовательность, мы можем ожидать, что значение измеряемой величины изменяется вероятностью $1 - p_m$ и остается неизменным с вероятностью p_m .

В большинстве практических случаев распределение r_k неизвестно. Однако, если можно сделать разумное предположение о виде функции распределения r_k , то p_m , в рамках такого предположения, можно использовать как верхнюю или нижнюю оценку вероятности изменения данных при наблюдении монотонной последовательности во временном ряду первичных данных.

Определим задержку τ_n , которая применяется при отправке данных сенсора модулям интеграции данных и управления. Задержка определяется динамически, в соответствии со свойствами потока данных и выражает соотношение между оптимистическими и пессимистическими ожиданиями неопределенности. Низкая задержка означает преобладание пессимистических ожиданий, высокая — оптимистических. Величина τ_n возрастает ли-

нейно, если данные не содержат указаний о необходимости дополнительного управления и уменьшается по показательному закону, если ожидания неопределенности возрастают.

Как было указано выше, периоды монотонного роста или убывания первичных данных могут использоваться как индикатор нарастания пессимизма. Однако многие сенсоры предполагают наличие монотонных последовательностей в периоды нормальной работы АМУ. Это, например, данные позиционирования, дальномеры, сонары, иногда данные скорости, камеры. Если эти изменения этих данных должны происходить в рамках некоторого шаблона, они также могут предаваться с задержкой. Для этой цели необходимо ввести дополнительные метрики, которые будут сохранять свои значения неизменными в периоды нормальной работы АМУ.

Помимо идентификации монотонных последовательностей для управления также используются сигналы обратной связи, которые модули интеграции и контроля данных направляют НМИОД. Эти величины содержат оценку степени новизны информации, которая содержится во вновь поступивших данных. Сигналы обратной связи образуют последовательность $\{k_n\}_{n=0}^{\infty}$, где k_n оценивает важность полученных данных после каждого интервала τ_n . Значение $k_n \in \mathbb{R}$ и $0 < k_n \leq 1$. Здесь 1 означает критически важные, а 0 означает незначительные. Таким образом задержка τ_n вычисляются по следующей схеме:

$$\tau_{n+1} = \begin{cases} \frac{\alpha}{k_n + 1} \tau_n, & \text{если произошло событие изменения} \\ \tau_n + \delta, & \text{иначе.} \end{cases} \quad (1)$$

Здесь $0 < \alpha < 1$ коэффициент уменьшения задержки или коэффициент пессимизма и $\delta > 0$ константа роста или коэффициент оптимизма. Коэффициент пессимизма выбирается из интервала $[\alpha/2, \alpha]$. Указанные выше события изменений определяются следующим образом:

Таблица 9 — Параметры НМИОД

Параметр	Значение
α	Параметр мультипликативного убывания.
δ	Параметр аддитивного роста.
m	Длина монотонной последовательности в потоке.
InitDelay	Начальное значение задержки.

Продолжение таблицы 9

Параметр	Значение
UpperLimit	Верхний предел задержки.

- а) Значение $\|s_k^i - s_{k-1}^i\| > 0$ или $\|s_k^i - s_{k-1}^i\| < 0$ для m значений s_k^i подряд, где m - параметр метода.
- б) Значение $\|\hat{s}_k^i - s_k^i\| > 0$ или $\|\hat{s}_k^i - s_k^i\| < 0$ для m значений подряд, где m параметр метода.
- в) Определено критическое событие. Список критических событий формируется заранее. Критическими считаются события, требующие немедленной реакции. В этом случае отправляется сигнал k_{urgent} .

Полное описание параметров метода приведено в таблице 9.

Если сигналы обратной связи не используются модулями интеграции данных и управления, то эволюция задержки примет вид

$$\tau_{n+1} = \begin{cases} \alpha\tau_n, & \text{если произошло событие изменения} \\ \tau_n + \delta, & \text{иначе.} \end{cases} \quad (2)$$

2.2.2.2 Графовая модель как основа распознавания приоритетных состояний.

Для распознавания приоритетных событий в работе АМУ определим множество состояний системы P . Элементы множества P определяются экспертным путем, в них включаются режимы стабильной работы системы и функциональные состояния, которые не требуют применения интеллектуальных методов для построения дополнительного управления. Размерность множества P зависит от нескольких факторов, в частности:

- а) объема функций, выполняемых системой;
- б) степени детализации, необходимой для анализа и распознавания;
- в) сложности окружающей среды и уровня ее неопределенности;
- г) надежности АМУ.

Также определим возможные переходы между состояниями множества P . Будем считать, что между элементами множества P есть связь, если между ними возможен переход. Определим множество связей C , где каждая связь является направленной. Пара $\{P, C\}$ образует направленный граф. С каждым элементом множества $c_{ij} \in C$ ассоциирована функция перехода $F_{ij}(x_1, \dots, x_n)$, значение которой является бинарная величина 1/0, а аргументами является набор данных сенсоров s_i^k , отобранных НМИОД. Заметим, что для каждой функции используется собственный набор потоков первичных данных. Опре-

делим множество F всех возможных функций перехода. Тогда тройка $\{P, C, F\}$ образует графовую модель АМУ.

Определение 1. *Приоритетным событием называется событие перехода между элементами множества P .*

Таким образом, задача сводится к определению событий переходов между элементами множества P и регистрации их.

Пример. Рассмотрим четырехколесное АМУ, оснащенное одним мотором, инерциальным модулем, содержащим трехосные акселерометр и гироскоп, термометр, а также лазерным дальномером. АМУ передвигается внутри помещения по заданным маршрутам, перемещая предметы или является проводником на маршруте для тех, кто с ним не знаком. Определим следующие состояния АМУ:

- а) P_1 АМУ стоит. Мотор выключен
- б) P_2 АМУ стоит. Мотор включен
- в) P_3 АМУ движется прямолинейно, равномерно.
- г) P_4 Поворот
- д) P_5 Нештатная ситуация:
 - 1) Температура системы вне рабочего диапазона
 - 2) Опрокидывание
 - 3) Незапланированное ускорение
 - 4) Препятствие

Для распознавания событий переходов между этими состояниями будут использоваться следующие источники данных:

- а) акселерометр: потоки ax_i, ay_i, az_i — значения ускорения по трем осям;
- б) гироскоп: потоки hx_i, hy_i, hz_i — значения угловой скорости по трем осям;
- в) дальномер d_i — значения данных дальномера;
- г) термометр t_i — данные термометра.

Временной ряд первичных данных по каждой оси акселерометра и гироскопа образует отдельный источник данных, обрабатывается НМИОД независимо и поддерживает собственное значение задержки адаптивной стратегии. Регистрация первичных данных происходит равномерно во времени с периодом δt^k где k — номер потока. Для каждого сенсора определяется собственный интервал регистрации.

В силу того, что данные дальномера могут демонстрировать монотонный рост/убывание в периоды нормальной работы АМУ, построим дополнительное преобразование. А именно, по данным дальномера перед направлением их в НМИОД предварительно вычисляется метрика $\delta d_i = d_i - d_{i+1}$. Так во время нормальной работы АМУ

движется равномерно, а интервал между измерениями остается постоянным, метрика δd_i удовлетворяет требованиям НМИОД.

Примеры функций, являющихся элементами множества F :

- а) $F_{12}(ax_i, az_i) = |ax_i| > \epsilon \text{ AND } |az_i| > \epsilon$
- б) $F_{21}(ax_i, az_i) = |ax_i| < \epsilon \text{ AND } |az_i| < \epsilon$
- в) $F_{23}(\sigma_{ax_i}^2, \sigma_{ay_i}^2) = |\sigma_{ax_i}^2| \geq \epsilon_1 \text{ AND } |\sigma_{ay_i}^2| \geq \epsilon_1$
- г) $F_{32}(\sigma_{ax_i}^2, \sigma_{ay_i}^2) = |\sigma_{ax_i}^2| < \epsilon_1 \text{ AND } |\sigma_{ay_i}^2| < \epsilon_1$
- д) $F_{34}(hz_i) = |hz_i| > \epsilon_2$
- е) $F_{43}(hz_i) = |hz_i| < \epsilon_2$
- ж) $F_{i5_1}(t_i) = (T_{min} > t_i) \text{ OR } (t_i > T_{max})$
- з) $F_{i5_2}(hx_i, hy_i) = |hx_i| > H_{max} \text{ OR } |hy_i| > H_{max}$
- и) $F_{i5_3}(ax_i, ay_i) = |ax_i| > A \text{ OR } |ay_i| > A$
- к) $F_{i5_4}(\delta d_i) = |\delta d_i| < D_{min}$
- л) $F_{i1}(ax_i, az_i, hx_i, hy_i, hz_i, t, \delta d_i) = |ax_i| < \epsilon \text{ AND } |az_i| < \epsilon \text{ AND } |hx_i| < \epsilon_2 \text{ AND } |hy_i| < \epsilon_2 \text{ AND } |hz_i| < \epsilon_2 \text{ AND } T_{min} < t_i \text{ AND } t_i < T_{max}$

Константы $\epsilon, \epsilon_1, \epsilon_2$ означают пренебрежимо малые значения данных акселерометра (ϵ), дисперсии (ϵ_1) и гироскопа (ϵ_2). Они определяются экспериментально или на основе документации МЕМС для каждой реализации. Константы $A, D_{min}, H_{max}, T_{min}, T_{max}$ означают максимальные значения ускорения, пройденного за фиксированный интервал расстояния, угловой скорости и колебаний температуры, которые допустимы при нормальной работе АМУ. **Описание стадий исследований и разработок** На рисунке 58 представлен граф приоритетных состояний, отображающий связь между состояниями:

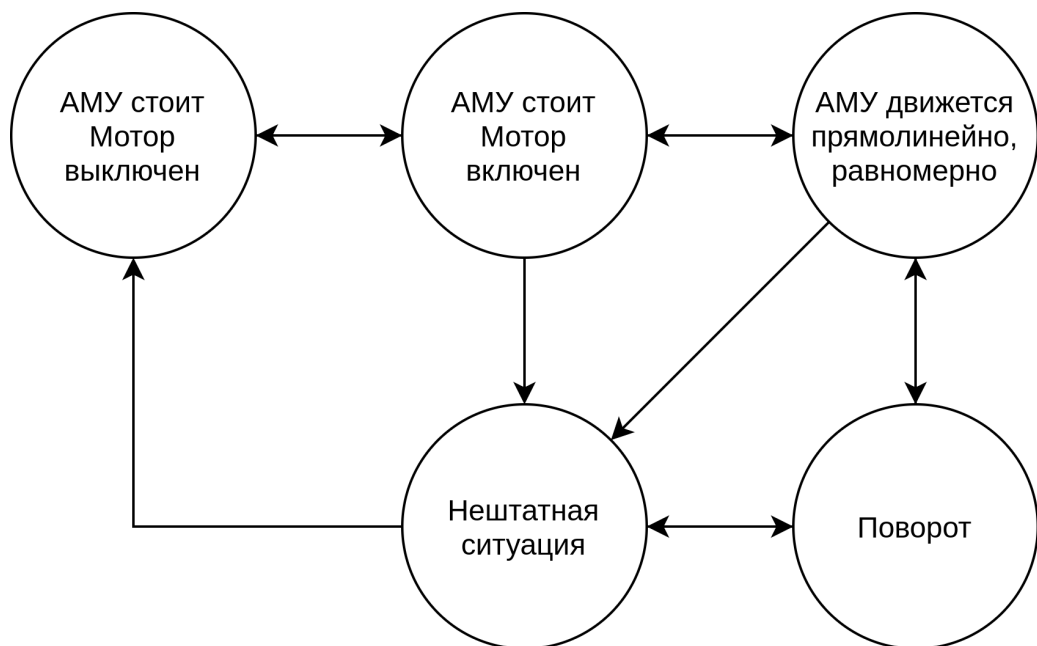


Рисунок 58 — Граф приоритетных состояний

2.2.3 ПАК распознавания приоритетных событий

2.2.3.1 Концептуальная модель программно-аппаратного прототипа

Предлагается двухэтапная схему распознавания событий. Первый этап — отбор исходных данных сенсоров, которые потенциально могут представлять интерес для задач распознавания. Второй этап - распознавание приоритетных событий по данным, прошедших первый этап. Отметим следующие особенности.

- а) На первом этапе используется нечеткий метод интеллектуального отбора данных (НМИОД), основанный на адаптивной стратегии управления задержкой передачи данных.
- б) На втором этапе данные, отобранные НМИОД, передаются модулю распознавания, который определяет состояние системы.

В алгоритме НМИОД используется переменная задержку, которая выражает баланс оптимистических и пессимистических ожиданий в отношении окружающей среды и работы АМУ [52]. Также метод определяет ряд событий, которые могут понижать уровень оптимизма. Если по истечении периода задержки ни одно из определённых событий не наступило, то задержка увеличивается аддитивно, в противном случае — уменьшается мультипликативно.

Предполагается, что монотонные последовательности роста или убывания во временном ряду исходных данных сенсора могут сигнализировать об изменении измеряемой сенсором величины. В случае, если эта величина должна иметь постоянное значение или изменяться незначительно такие последовательности могут сигнализировать об необходимости дополнительного анализа окружающей обстановки. Например, при равномерном прямолинейном движении АМУ значения ускорения должны быть близки к нулю. Если нормальная работа АМУ предполагает наличие изменений измеряемой величины, то к таким переменным могут применяться следующие два подхода.

- а) Исходные данные преобразуются с помощью некоторой функции, значение которой должно сохранять постоянное величину. Например, для последовательности данных дальномера $d_{ij} > 1$ вычисляется разность $\Delta_i = d_{i+1} - d_i$. Если АМУ движется равномерно прямолинейно, а данные дальномера передаются через равные промежутки времени, то элементы последовательности Δ_i характеризуют скорость движения АМУ и должны незначительно колебаться относительно некоторой величины V .
- б) Исходные данные должны изменяться с незначительными отклонения от некоторого шаблона. Для распознавания шаблона могут использоваться нейросети [53], идентификация цифровых двойников и ряд других методов.

Данные, отобранные НМИОД, передаются модулю распознавания, который определяет состояние системы. С этой целью мы определяем граф состояний АМУ – среда и функции перехода между этими состояниями, параметрами которых являются данные сенсоров, отобранные НМИОД.

В качестве состояний определяются существенные этапы нормальной работы АМУ и события окружающей среды которые могут повлиять на работу АМУ или привести к сбоям. Функции перехода трактуют данные сенсоров, специфические для каждого перехода. Функции выполняются при поступлении новых данных от НМИОД. По результату работы функции система может перейти в состояние, определённое графом переходов или остаться в том же состоянии.

В соответствии с рисунком 59 представлена высокоуровневая архитектура приложения по выявлению приоритетных событий на базе конвейерной модели обработки данных.

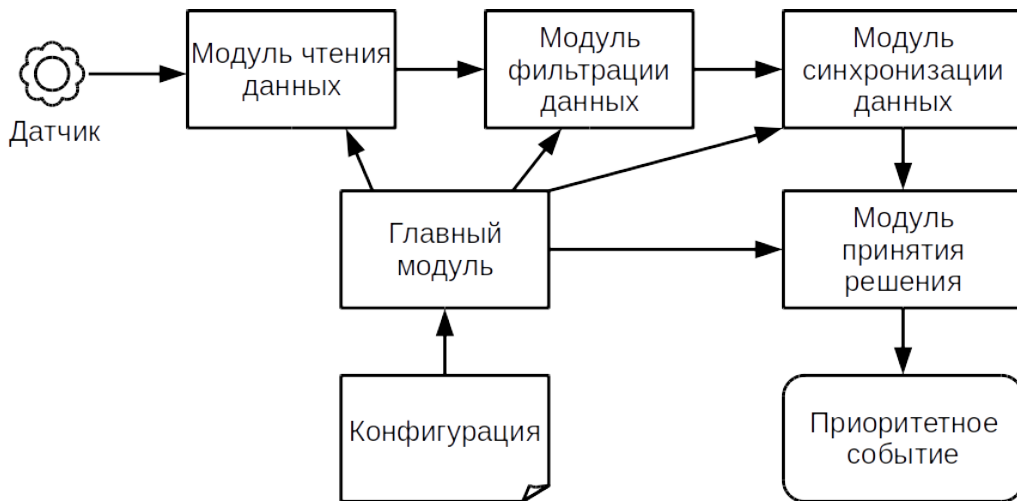


Рисунок 59 — Высокоуровневая архитектура приложения

Главный модуль выполняет инициализацию работы приложения, чтение конфигурации и организацию связей в конвейере. Модули чтения данных организуют подключение к датчику и выполняют циклические опросы данных в зависимости от частоты работы датчиков. Модули фильтрации реализуют алгоритм НМИОД для каждого потока данных по отдельности. Отобранные данные сопоставляются в модуле синхронизации данных и на выходе организуются комплекты данных. Модуль принятия решения анализирует каждый комплект данных и на основе графа состояний АМУ определяет возможный переход, т. е. приоритетное событие.

В соответствии с рисунком 60 представлен поток данных

Модуль чтения данных получает данные с определенного датчика и отправляет их в модуль фильтрации. После прохождения фильтрации данные отправляются в модуль синхронизации, где из них формируется комплект из 8 показаний. На основе комплекта показаний модуль принятия решения выявляет наступление приоритетного события

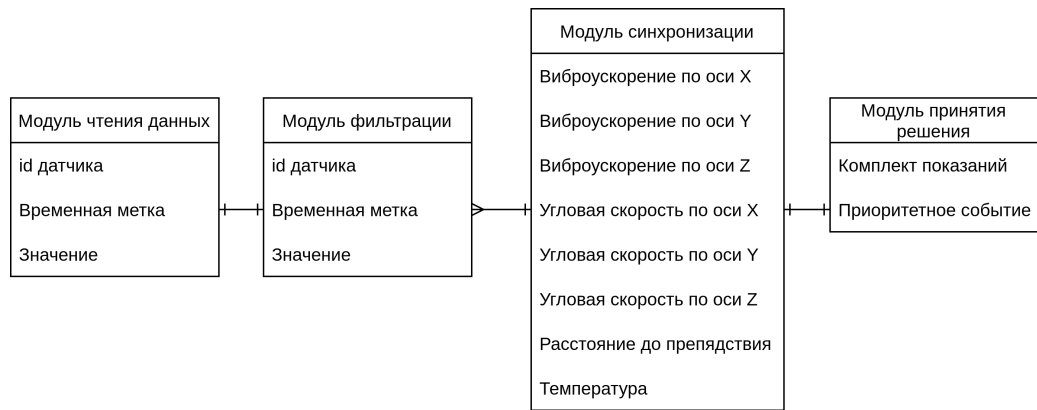


Рисунок 60 — Поток данных

2.2.3.2 Аппаратная часть

Программно-аппаратный прототип представляет из себя колесного робота, изображен на рисунке 61 с установленным набором следующих компонентов:

- а) Одноплатный компьютер Raspberry Pi 3B
- б) Инерциальный модуль IMU10dof состоящий из следующих наборов сенсоров:
 - 1) Трехосевой акселерометр ADXL345;
 - 2) Трехосевой гироскоп ITG3200;
 - 3) Трехосевой магнитометр HMC5883L;
 - 4) Датчик атмосферного давления BMP085 со встроенным термосенсором;
 - 5) Лазерный дальномер URG-04LX-UG01.
 - 6) Драйвер двигателей постоянного тока
 - 7) 4 приводных колеса с мотор-редуктором.

Каждая ось акселерометра и гироскопа считается источником данных и обрабатывается как отдельный поток. Дополнительно используются датчик температуры и лазерный дальномер.

Сводная таблица используемых датчиков слоя сенсорики

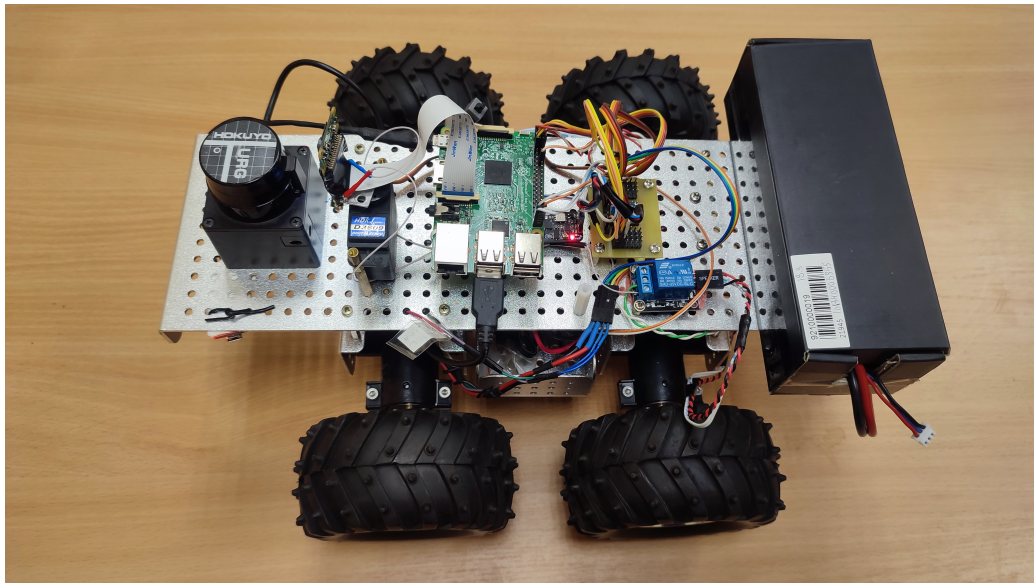


Рисунок 61 — Программно-аппаратный прототип на примере используемого АМУ

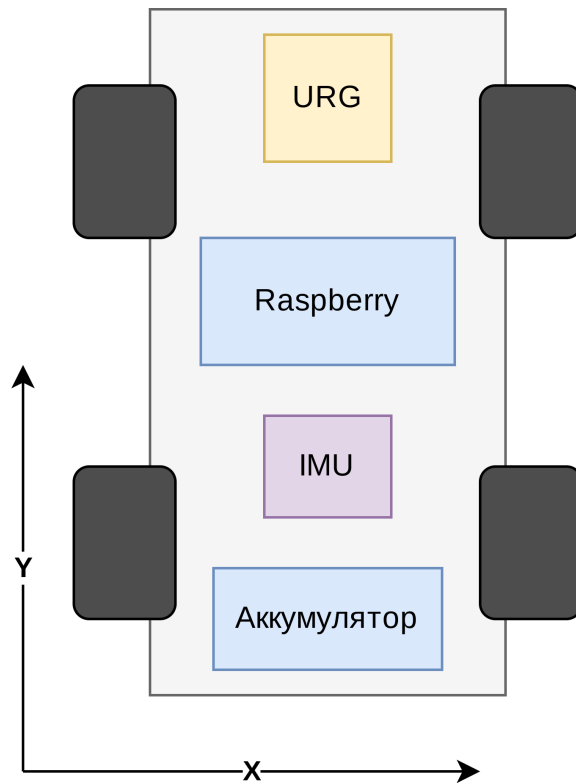


Рисунок 62 — Схема размещения датчиков на АМУ

Продолжение таблицы 10

1	2	3	4
---	---	---	---

Таблица 10 — Сводная таблица слоя сенсорики

Название	Из- меряемая физиче- ская величина	Погрешность, ча- стота дискретиза- ции, диапазон из- мерений	Описание
IMU10dof	-	-	Трехосевой инерциальный измерительный модуль. В состав входит трехосевые акселерометр ADXL345, гироскоп ITG3200, магнитометр HMC5883L и датчик атмосферного давления BMP085 со встроенным термосенсором. Интерфейс подключения модуля I ² C.
ADXL345	Виброускорение, мм/с ²	Эффективная частота дискретизации 100 Гц. Погрешность измерений ±1%. Диапазон измерений ±16g	Сенсор выполняет измерение виброускорения по трем осям, включая ускорение свободного падения по оси Z. Входит в состав трехосевого инерциального модуля IMU10dof. Интерфейс подключения датчика I ² C, адрес 0x1D.
ITG3200	Угловое переме- щение, град/сек	±6%. Диапазон измерений ±2000 град/сек. Эффективная частота дискретизации 100 Гц	Входит в состав трехосевого инерциального модуля IMU10dof. Интерфейс подключения датчика I ² C, адрес 0x68.

Продолжение таблицы 10

1	2	3	4
ВМР085	Давление и температура в Па и градусах Цельсия		Производит измерение расстояния до препятствия, расположенного перед собой. Входит в состав трехосевого инерциального модуля IMU10dof. Интерфейс подключения датчика I ² C, адрес 0x77.
URG 04LX-UG01	Производит измерение расстояния до препятствия, расположенного перед собой, мм	Погрешность измерения ± 300 мм, при расстояниях более 1 м — $\pm 3\%$. Эффективная частота дискретизации до 10 Гц. Диапазон измерений: до 5600 мм	Смонтирован в передней части АМУ для определения расстояния до препятствия. Интерфейс подключения USB.

Для сбора данных слоя сенсорики реализован программно-аппаратный модуль, содержащий в себе набор датчиков, а так же программное обеспечение для получения показаний с датчиков, их предварительную обработку и дальнейшую отправку. Схема слоя сенсорики представлена на рисунке 63

Модуль реализован в виде программной компоненты, содержащий один процесс, состоящий из трех потоков:

- а) Поток, выполняющий работу с модулем IMU10dof:
 - 1) конфигурирование модуля;
 - 2) сбор показаний с трехосевого акселерометра;
 - 3) сбор показаний с трехосевого гироскопа;
 - 4) получение температуры с датчика атмосферного давления и температуры;
 - 5) нормировку полученных показаний;
 - 6) логирование полученных показаний;
 - 7) инкапсуляцию и отправку полученных показаний по UDP.
- б) Поток, выполняющий работу с дальномером URG:
 - 1) конфигурирование дальномера;

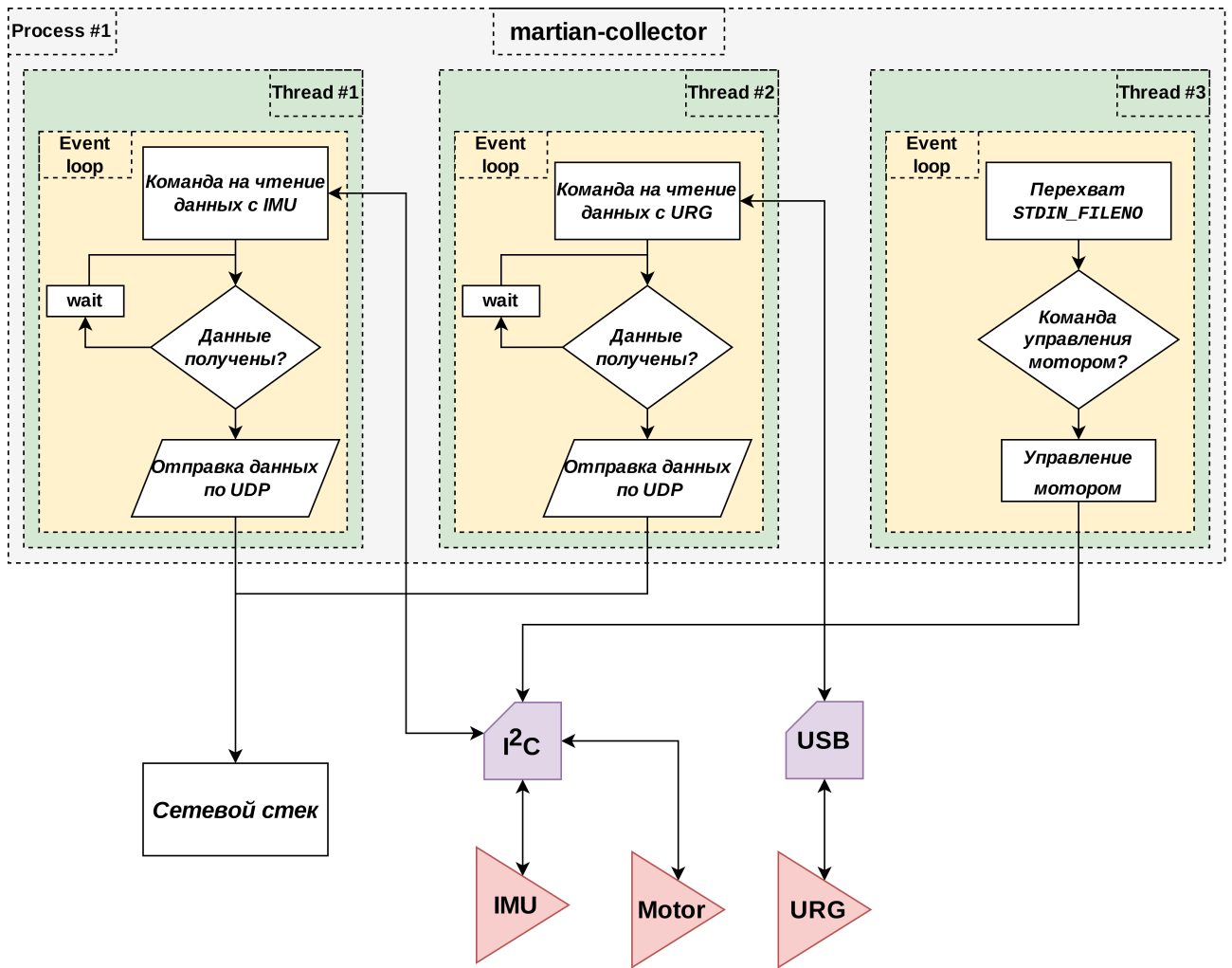


Рисунок 63 — Схема слоя сенсорики АМУ

- 2) получение расстояния до препятствия в секторе 4° от лазерного дальнотемера;
 - 3) логирование полученного показания;
 - 4) инкапсуляцию и отправку полученного показания по UDP.
- в) Поток, выполняющий работу с управление драйвером двигателей с помощью команд в терминале:
- 1) чтение нажатой клавиши;
 - 2) преобразование кода нажатой клавиши в соответствующую команду драйвера двигателя;
 - 3) выполнение команды драйвером двигателя (осуществление движения).
 - 4) логирование команд управления;
 - 5) вывод напряжения на аккумуляторной батарее.

Разделяемым ресурсом для нескольких потоков является символьное устройство I²C: i2c-1. Синхронизация потоков не предусмотрена ввиду отсутствия ситуаций с критическим доступом к разделяемым ресурсам.

2.2.3.3 Алгоритм отбора данных

Описание алгоритма

Фильтрация потоков данных происходит на основе поиска монотонной последовательности, в случае нахождения такой последовательности, то задержка отправки данных мультипликативно уменьшается, в противном случае задержка отправки данных аддитивно увеличивается.

Описание программной реализации

Алгоритм фильтрации в виде класса `FilterFunction`, в котором используются следующие переменные:

- а) `alpha` - параметр мультипликативного уменьшения;
- б) `delta` - параметр аддитивного увеличения;
- в) `m` - длина осмысленной монотонной последовательности в потоке данных;
- г) `initDelay` - начальное значение задержки;
- д) `upperLimit` - верхний предел задержки, который не должен быть превышен;
- е) `syncModule` - указатель на объект `SyncDataModule`, реализующий функцию синхронизации;
- ж) `last_filter_value` - значение датчика с прошлой итерации;
- з) `inc` - счетчик монотонного увеличения последовательности;
- и) `dec` - счетчик монотонного уменьшения последовательности;
- к) `seqFound` - флаг нахождения достаточной монотонной последовательности
- л) `received` - счётчик полученных пакетов;
- м) `sent` - счётчик отправленных пакетов;
- н) `m_name` - название датчика.

Также данный модуль реализует следующие методы:

- а) инициализация класса `FilterFunction` - при инициализации класса передаётся конфигурационная строка имеющая `json` структуру, далее происходит её парсинг и определяются следующие объекты: параметр мультипликативного уменьшения, параметр аддитивного увеличения, длина осмысленной монотонной последовательности в потоке данных, начальное значение задержки и верхний предел задержки, который не должен быть превышен;
- б) метод `setSync`- на вход получает указатель на объект класса `SyncDataModule` и сохраняет указатель в переменной `syncModule`;

- в) метод `getStats` - возвращает объект `tuple` состоящий из количества полученных объектов в методе `insertDataPiece` и количества отправленных объектов в метод `insertDataPiece` класса `SyncDataModule`;
- г) метод `getName` - возвращает `string` содержащий название датчика;
- д) метод `insertDataPiece` - на вход получает объект класса `DataPiece`, после получения первого объекта, алгоритм отправляет данные в модуль синхронизации и начинает итерировать все поступающие объекты. При нахождении монотонно увеличивающейся или уменьшающейся последовательности, происходит отправка полученного объекта в модуль синхронизации и мультипликативное уменьшение значения задержки. В случае достижения задержки при отсутствии монотонной последовательности, то объект также отправляется в модуль синхронизации и значение задержки аддитивно увеличивается. Если последовательность не является монотонной и задержка еще не закончилась, то отправка данных не происходит.

2.2.3.4 Алгоритм синхронизации данных

(Мадрахимова, Кулаков)

2.2.3.5 Алгоритм выявления приоритетных событий

Возможность перехода и его приоритетность определяется на основе функций и матрицы переходов, функции перехода проверяют возможность выполнения перехода на основе входных данных, а матрица перехода регулирует приоритетность доступных переходов. Матрица переходов имеет следующую структуру: индекс строки соответствуют состоянию системы установленному на настоящий момент времени, индексы столбцов соответствуют номерам состояний для перехода. Значение матрицы на пересечении строк и столбцов соответствующих выбранному переходу отвечают за приоритетность перехода, для значений равных нулю переход невозможен.

$$\begin{pmatrix} 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.2 & 0 & 0.5 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0.2 & 0 & 0.5 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0.5 & 0 & 1 & 1 & 1 & 1 \\ 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Конечный автомат реализован в виде класса `FiniteStateMachine`, в котором используются следующие переменные:

- а) `condition` - строка сохраняющая название текущего состояния.
- б) `weight_matrix` - указатель на массив, представляющий собой матрицу переходов
- в) `weight_matrix_elements` - количество элементов в матрице переходов.
- г) `weight_matrix_size_x` - количество столбцов матрицы переходов
- д) `names` - массив строк обозначающих названия для состояний (номер строки в массиве соответствует номеру состояния в матрице переходов)
- е) `id_to_var` - объект `map`, устанавливающий соответствие между `id`-датчиков и переменными в функциях переходов
- ж) `memorized_params` - объект `map`, устанавливающий список названий переменных функций перехода, значение которых сохраняется до обновления на следующей итерации (ключ - название переменной, значение - значение переменной)
- з) `delta_params` - объект `map`, устанавливающий список новых переменных определяемых как разность значений одной переменной, полученных на текущей и предыдущей итерациях
- и) `transition_funcs` - указатель на объект `TransitionFunction`, реализующий функции перехода
- к) `m_output` - указатель на объект `OutputModule`, реализующий функцию вывода текущего состояния.
- л) `received` - счётчик полученных пакетов из модуля синхронизации
- м) `sent` - счётчик отправленных пакетов (счётчик обращений к методу `PrintState` класса `OutputModule`)

и реализующего следующие методы:

- а) Инициализация класса `FiniteStateMachine` - при инициализации класса передаётся конфигурационная строка имеющая `json` структуру, далее происходит её разбор и определяются следующие объекты: массив названий состояний, начальное состояние системы, количество элементов в матрице состояний, количество столбцов в матрице состояний, матрица состояний, словарь сопоставлений `id`-датчиков и переменных, список переменных для которых требуется вычислять разность значений между итерациями, инициализация объекта класса `TransitionFunction`.
- б) Метод `setOutputModule` - на вход получает указатель на объект класса `OutputModule` и сохраняет указатель в переменной `m_output`.
- в) Метод `getStats` - возвращает объект `tuple` состоящий из количества полученных векторов в методе `appendData` и количества отправленных состояний в метод `PrintState` класса `OutputModule`.
- г) Метод `appendData` - на вход получает вектор состоящий из объектов класса `DataPiece`, далее происходит преобразование значений вектора в соответствующее

переменные и вычисление разности значений для тех указанных переменных. Далее, на основе указанного состояния системы определяется возможность перехода в одно из доступных состояний с помощью функций перехода: если значение матрицы перехода с номером строки соответствующему номеру текущего состояния и с номером столбца соответствующего состоянию для перехода равняется нулю то проверка перехода с помощью функций перехода не выполняется, иначе проводится проверка выполнения соответствующей функции перехода и полученное значение умножается на соответствующее значение из матрицы переходов, далее определяется максимальное из этих значений и выполняется переход в соответствующее ему состояние, которое передается методу PrintState объекту класса OutputModule, если максимальное значение равно нулю, то состояние не меняется.

Функции перехода описываются в конфигурационном файле в разделе transition_functions и имеют следующую структуру:

- а) start_position - текущее состояние
- б) finish_position - состояние в которое совершается переход
- в) conditions - функция, соответствующая переходу

Для описания функций могут быть использованы константы, описанные в разделе constants и имеющие следующую структуру:

- а) variable - имя константы
- б) value - значение

Или переменные, которые обозначают показания с датчиков и описаны в разделе id_names

Для разбора и проверки функций перехода реализован класс TransitionFunction, в котором используются следующие поля:

- а) conditions - объект типа map, где ключом является строка в формате start_position + "_" + finish_position, а значением соответствующая функция перехода в виде объекта типа Condition
- б) constants - объект типа map, где ключом является имя константы, а значением является соответствующее значение константы

и реализованы следующие методы:

- а) Инициализация класса TransitionFunction - при инициализации класса в качестве параметра передается строка имеющая json структуру, описывающую функции переходов и значение констант
- б) Метод parseConditions - на вход получает объект типа json, который является частью строки, переданной при инициализации. Метод вызывается рекурсивно на каждую логическую функцию, пока не будут достигнуты листья дерева условий

- в) Метод `parseConstants` - на вход получает объект типа `json`, который описывает список используемых констант и их значения. В результате работы метода возвращается список констант с соответствующими значениями
- г) Метод `checkTransition` - на вход получает следующие объекты:
 - 1) `start_position` - текущее состояние
 - 2) `finish_position` - состояние, в которое происходит переход
 - 3) `var` - набор показаний с датчиков сенсорики

и возвращает возможность данного перехода

Для хранения дерева условий реализован класс `Condition`, в котором используются следующие поля:

- а) `first_val` - левое значение выражения
- б) `second_val` - правое значение выражения
- в) `statement` - знак выражения (`>`, `<`, `>=`, `<=`, `=`)
- г) `isAbsFirst` - указатель использования модуля для левой части выражения
- д) `isAbsSecond` - указатель использования модуля для правой части выражения
- е) `isNot` - логическое отрицание
- ж) `or_conditions` - поддереву внутри условия ИЛИ
- з) `and_conditions` - поддереву внутри условия И

и реализуются следующие методы:

- а) Метод `checkData` - на вход получает показания с датчиков сенсорики и набор констант. В результате работы метода возвращается логическое значение - результат проверки условия перехода
- б) Метод `checkStatement` - на вход получает значение правой и левой части логического условия. В результате работы метода возвращается логическое значение - результат проверки логического условия

2.2.4 Экспериментальный анализ

2.2.4.1 Описание экспериментов

В рамках экспериментального анализа выполняется модульное тестирование и серия экспериментов в соответствии с программой и методикой испытаний.

Модульное тестирование выполняется автоматическим способом. В модульном тестировании участвует весь код за исключением кода инициализации (см. архитектуру на рисунке 59). Модульное тестирование реализовано с помощью библиотеки `Google Test` [54]. Тестирование выполняется путем подключения исходного кода приложения в требуемых комбинациях модулей и запуска на тестовых наборах данных.

Для контроля ввода-вывода модулей приложения используются специальные объекты-обманки («mock» объекты). Mock объекты реализуются с помощью технологии Google Mock [54]. В рамках модульного тестирования реализованы mock объекты для всех модулей и объектов данных.

Эксперименты проводятся в закрытом помещении размером 3 на 6 метров. Экспериментальный стенд состоит из АМУ в соответствии с рисунком ?? и ЭВМ наблюдения. В серию входят следующие эксперименты:

- а) Прямолинейное движение. В ходе эксперимента АМУ из состояния покоя выполняет прямолинейное движение и останавливается.
- б) Поворот. В ходе эксперимента АМУ из состояния покоя выполняет последовательно прямолинейное движение, поворот, прямолинейное движение и остановку.
- в) Препятствие. В ходе эксперимента АМУ из состояния покоя последовательно выполняет прямолинейное движение, поворот, подъезд к стене.
- г) Разворот. В ходе эксперимента АМУ из состояния покоя последовательно выполняет прямолинейное движение, разворот на 180 градусов, прямолинейное движение и остановку.
- д) Крен/Переворот. В ходе эксперимента АМУ из состояния покоя последовательно выполняет прямолинейное движение, наезд левой/правой стороной на возвышенность с образованием угла наклона не менее 60 градусов и остановку.
- е) Перегрев. В ходе эксперимента АМУ нагревается до 40 градусов цельсия.
- ж) Незапланированное ускорение. В ходе эксперимента АМУ из состояния покоя последовательно выполняет прямолинейное движение, ускорение и остановку.

При проведении эксперимента на ЭВМ наблюдения организуется подключение к АМУ с использованием протокола SSH, запуск приложения распознавания приоритетных событий с подключением к АМУ и приложения управления АМУ. Результат эксперимента анализируется по отладочному выводу приложения распознавания приоритетных событий и соответствию сообщений фактическому состоянию АМУ.

2.2.4.2 Результаты анализа данных

Ермаков: тестирование ввода

Ремшу: тестирование фильтрации

Мадрахимова: тестирование синхронизатора

Ткаченко: тестирование функций условий

Сафонов: тестирование графа переходов

Тестирование работы системы в целом проводилось в два этапа:

- а) Подключение модулей в цепочку и проверка взаимодействия модулей;
- б) Проверка утечки памяти в ходе работы системы в целом.

В ходе тестирования подключения модулей выполнялась проверка чтения настроек функцией «startup()» из файлов с правильным и неправильным содержанием, проверка создания цепочки модулей в соответствии с конфигурацией, проверка чтения данных с файлов с тестовыми наборами данных, проверка фильтрации данных тестовых наборов, проверка синхронизации данных с тестовых наборов, проверка определения состояния по тестовым наборам. В результате выполнено 10 тестов успешно.

Для проверки утечки памяти был использован mock объект класса «DataPiece». В ходе тестирования каждому модулю на входе подавалось заданное количество экземпляров mock объекта и выполнялся расчет числа объектов созданных внутри модуля, числа объектов удаленных внутри модуля и числа объектов на выходе. Если число созданных объектов совпадало с числом удаленных объектов, то тест считался успешным. В результате выполнено 3 теста успешно.

Ермаков: тестирование на тележке

2.2.5 Раздел про Лиончопинг

- 3 Разработка концептуального прототипа цифрового образа территории, создаваемого с помощью робототехнических сенсорных систем
 - 3.1 Программно-аппаратный прототип распознавателя знаний о территории движения человека на основе технологий тактильного интернета и периферийных робототехнических IoT-систем на стороне мобильного пользователя
 - 3.2 Программно-аппаратный прототип распознавателя текущих интересов мобильного пользователя на основе данных, получаемых от сопровождающих IoT-устройств

4 Интеграция образовательных программ с научной и производственной деятельностью

Описание модернизированных образовательных программ магистратуры под задачи разработки интеллектуальных сенсорных систем для робототехники, а также специализированных образовательных модулей по технологиям и приложениям искусственного интеллекта в программы практик и выпускных работ

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Баланев, Никита Владимирович. Анализ факторов, влияющих на точность позиционирования промышленного робота и методы обеспечения заданной точности [Текст] / Никита Владимирович Баланев, Роман Андреевич Янов // Достижения науки и образования. — 2016. — № 1 (2).
- 2 A Study of Accuracy and Time Delay for Bilateral Master-Slave Industrial Robotic Arm Manipulator System [Text] / Nuratiqa Natrah Mansor, Muhammad Herman Jamaluddin, Ahmad Zaki Shukor, Chee Cong Lok // MATEC Web of Conferences / EDP Sciences. — Vol. 150. — [S. l. : s. n.], 2018. — P. 01015.
- 3 Mechanical design and control of inflatable robotic arms for high positioning accuracy [Text] / Hye-Jong Kim, Akihiro Kawamura, Yasutaka Nishioka, Sadao Kawamura // Advanced Robotics. — 2018. — Vol. 32, no. 2. — P. 89–104.
- 4 Geometrical calibration of a 6-axis robotic arm for high accuracy manufacturing task [Text] / Luca Lattanzi, Cristina Cristalli, Daniele Massa [et al.] // The International Journal of Advanced Manufacturing Technology. — 2020. — Vol. 111, no. 7. — P. 1813–1829.
- 5 Крахмалев, ОН. Определение динамической точности манипуляционных систем роботов с упругими шарнирами [Текст] / ОН Крахмалев, ЛИ Блейшмидт // Проблемы машиностроения и надежности машин. — 2014. — № 1. — С. 29–36.
- 6 Повышение точности и производительности роботизированного фрезерования на основе траекторно-импедансного управления [Текст] / Владимир Андреевич Гречишников, Юрий Владимирович Илюхин, Александр Вячеславович Исаев [и др.] // Вестник МГТУ Станкин. — 2016. — № 4. — С. 8–16.
- 7 Кольцов, Александр Германович. Проверка пятикоординатного фрезерного обрабатывающего центра на точность [Текст] / Александр Германович Кольцов, Алексей Валерьевич Торопов, Александр Анатольевич Петухов // Вестник Уфимского государственного авиационного технического университета. — 2012. — Т. 16, № 4 (49).
- 8 Improvement of position accuracy for inflatable robotic arm using visual feedback control method [Text] / Hye-Jong Kim, Yuto Tanaka, Akihiro Kawamura [et al.] // 2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM) / IEEE. — [S. l. : s. n.], 2015. — P. 767–772.
- 9 Robotic arm calibration and teaching method based on binocular vision [Text] / Ming Chen, Xin Hong, Longsheng Wei [et al.] // 2020 39th Chinese Control Conference (CCC) / IEEE. — [S. l. : s. n.], 2020. — P. 5963–5969.
- 10 Yaniger, S.I. Force Sensing Resistors: A Review Of The Technology [Text] / S.I. Yaniger // Electro International, 1991. — [S. l. : s. n.], 1991. — P. 666–668.

- 11 2002.
- 12 J., Fraden. Handbook of modern sensors [Text]. — 2013.
- 13 Suzuki, K. A 1024-element high-performance silicon tactile imager [Text] / K. Suzuki, K. Najafi, K.D. Wise // Technical Digest., International Electron Devices Meeting. — [S. l. : s. n.], 1988. — P. 674–677.
- 14 Пейтон А.Дж. — 1994.
- 15 Tao, Jianhua. Affective Computing: A Review [Text] / Jianhua Tao, Tieniu Tan // Affective Computing and Intelligent Interaction / Ed. by Jianhua Tao, Tieniu Tan, Rosalind W. Picard. — Berlin, Heidelberg : Springer Berlin Heidelberg, 2005. — P. 981–995.
- 16 Luca, Joseph. Does emotional intelligence affect successful teamwork [Text] / Joseph Luca, Pina Tarricone // Proc. 18th Ann. Conf. of the Australasian Society for Computers in Learning in Tertiary Education / Ed. by G. Kennedy, M. Keppell, C. McNaught, T. Petrovic. — [S. l.] : Biomedical Multimedia Unit, The University of Melbourne, 2001. — Dec. — P. 367–376.
- 17 Golan, Maya. A framework for operator – workstation interaction in Industry 4.0 [Text] / Maya Golan, Yuval Cohen, Gonen Singer // International Journal of Production Research. — 2020. — Vol. 58, no. 8. — P. 2421–2432.
- 18 Bann, Eugene. Measuring cultural relativity of emotional valence and arousal using semantic clustering and twitter [Text] / Eugene Bann, Joanna Bryson // Proc. 35th Annual Meeting of the Cognitive Science Society / Ed. by Markus Knauff, Natalie Sebanz, Michael Pauen, Ipke Wachsmuth. — [S. l.] : Curran Associates, Inc., 2013. — Jul. — P. 1809–1815.
- 19 The Faces of Engagement: Automatic Recognition of Student Engagement from Facial Expressions [Text] / Jacob Whitehill, ZewelANJI Serpell, Yi-Ching Lin [et al.] // IEEE Transactions on Affective Computing. — 2014. — Vol. 5, no. 1. — P. 86–98.
- 20 COVID-19 and Remote Work: An Early Look at US Data [Text] : Working Paper : 27344 / National Bureau of Economic Research ; Executor: Erik Brynjolfsson, John J Horton, Adam Ozimek [et al.] : 2020. — Jun.
- 21 Designing Online Platforms Supporting Emotions and Awareness [Text] / César A. Col-lazos, Habib Fardoun, Deema AlSekait [et al.] // Electronics. — 2021. — Vol. 10, no. 3.
- 22 Detection and Evaluation of Emotions in Massive Open Online Courses [Text] / Der-ick Leony, Pedro J. M. Merino, José A. R. Valiente [et al.] // J. Univers. Comput. Sci. — 2015. — Vol. 21. — P. 638–655.
- 23 Multimodal Human Attention Detection for Reading from Facial Expression, Eye Gaze, and Mouse Dynamics [Text] / Jiajia Li, Grace Ngai, Hong Va Leong, Stephen C. F. Chan // SIGAPP Appl. Comput. Rev. — 2016. — Vol. 16, no. 3. — P. 37–49.

- 24 Ren, Zhiyun. Predicting Performance on MOOC Assessments Using Multi-Regression Models [Text] / Zhiyun Ren, Huzefa Rangwala, Aditya Johri // Proc. 9th International Conference on Educational Data Mining / Ed. by T. Barnes, M. Chi, M. Feng. — [S. l.] : NC State University, 2016. — Jun. — P. 484–490.
- 25 Building an Automated Engagement Recognizer Based on Video Analysis [Text] / Minsu Jang, Cheonshu Park, Hyun-Seung Yang [et al.] // Proceedings of the 2014 ACM/IEEE International Conference on Human-Robot Interaction. — New York, NY, USA : Association for Computing Machinery, 2014. — P. 182–183.
- 26 A Deep Learning Approach to Detecting Engagement of Online Learners [Text] / M. Ali Akber Dewan, Fuhua Lin, Dunwei Wen [et al.] // 2018 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCCom/IOP/SCI). — [S. l. : s. n.], 2018. — P. 1895–1902.
- 27 Automatic Detection of Learner Engagement Using Machine Learning and Wearable Sensors [Text] / M. Carroll, M. Ruble, M. Dranias [et al.] // J. Behav. Brain Sci. — 2020. — Vol. 10. — P. 165–178.
- 28 Frank, Maria. Engagement Detection in Meetings [Text]. — 2016. — 1608.08711.
- 29 Salmeron-Majadas, Sergio. An Evaluation of Mouse and Keyboard Interaction Indicators towards Non-intrusive and Low Cost Affective Modeling in an Educational Context [Text] / Sergio Salmeron-Majadas, Olga C. Santos, Jesus G. Boticario // Procedia Computer Science. — 2014. — Vol. 35. — P. 691–700.
- 30 Student behavior analysis to measure engagement levels in online learning environments [Text] / Khawlah Altuwairqi, Salma Kammoun Jarraya, Arwa Allinjawi, Mohamed Hammami // Signal, Image and Video Processing. — 2021. — Vol. 15, no. 7. — P. 1387–1395.
- 31 Face Mesh - mediapipe [Text]. — [S. l. : s. n.], 2021. — Access mode: https://google.github.io/mediapipe/solutions/face_mesh.html.
- 32 Blazeface: Sub-millisecond neural face detection on mobile gpus [Text] / Valentin Bazarevsky, Yury Kartynnik, Andrey Vakunov [et al.] // arXiv preprint arXiv:1907.05047. — 2019.
- 33 Real-time facial surface geometry from monocular video on mobile GPUs [Text] / Yury Kartynnik, Artsiom Ablavatski, Ivan Grishchenko, Matthias Grundmann // arXiv preprint arXiv:1907.06724. — 2019.
- 34 Training deep networks for facial expression recognition with crowd-sourced label distribution [Text] / Emad Barsoum, Cha Zhang, Cristian Canton Ferrer, Zhengyou Zhang // Proceedings of the 18th ACM International Conference on Multimodal Interaction. — [S. l. : s. n.], 2016. — P. 279–283.

- 35 Challenges in representation learning: A report on three machine learning contests [Text] / Ian J Goodfellow, Dumitru Erhan, Pierre Luc Carrier [et al.] // International conference on neural information processing / Springer. — [S. l. : s. n.], 2013. — P. 117–124.
- 36 Kingma, Diederik P. Adam: A method for stochastic optimization [Text] / Diederik P Kingma, Jimmy Ba // arXiv preprint arXiv:1412.6980. — 2014.
- 37 Low-cost gaze and pulse analysis using RealSense [Text] / Qiang Qiu, Zhuoqing Chang, Mark Draelos [et al.] // Proceedings of the 5th EAI International Conference on Wireless Mobile Communication and Healthcare. — [S. l. : s. n.], 2015. — P. 276–279.
- 38 Remote heart rate measurement from face videos under realistic situations [Text] / Xiaobai Li, Jie Chen, Guoying Zhao, Matti Pietikainen // Proceedings of the IEEE conference on computer vision and pattern recognition. — [S. l. : s. n.], 2014. — P. 4264–4271.
- 39 Algorithmic principles of remote PPG [Text] / Wenjin Wang, Albertus C den Brinker, Sander Stuijk, Gerard De Haan // IEEE Transactions on Biomedical Engineering. — 2016. — Vol. 64, no. 7. — P. 1479–1491.
- 40 Nam, Yunyoung. Estimation of Respiratory Rates Using the Built-in Microphone of a Smartphone or Headset [Text] / Yunyoung Nam, Bersain A. Reyes, Ki H. Chon // IEEE Journal of Biomedical and Health Informatics. — 2016. — Vol. 20, no. 6. — P. 1493–1501.
- 41 Mining of Keystroke and Mouse Dynamics to Increase the Engagement of Students with Programming Assignments [Text] / Mario Garcia Valdez, Juan-J. Merelo, Amaury Hernandez Aguila, Alejandra Mancilla Soto // Computational Intelligence: 9th International Joint Conference, IJCCI 2017 Funchal-Madeira, Portugal, November 1-3, 2017 Revised Selected Papers / Ed. by Christophe Sabourin, Juan Julian Merelo, Kurosh Madani, Kevin Warwick. — Cham : Springer International Publishing, 2019. — P. 41–61. — ISBN: 978-3-030-16469-0.
- 42 Arapakis, Ioannis. Understanding Within-Content Engagement through Pattern Analysis of Mouse Gestures [Text] / Ioannis Arapakis, Mounia Lalmas, George Valkanas // Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. — CIKM '14. — New York, NY, USA : Association for Computing Machinery, 2014. — P. 1439–1448.
- 43 Ambient Intelligence Services in IoT Environments: Emerging Research and Opportunities [Text] / Dmitry Korzun, Ekaterina Balandina, Alexey Kashevnik [et al.]. — [S. l.] : IGI Global, 2019. — ISBN: 9781522589730.
- 44 Rubio, Francisco. A review of mobile robots: Concepts, methods, theoretical framework, and applications [Text] / Francisco Rubio, Francisco Valero, Carlos Llopis-Albert // International Journal of Advanced Robotic Systems. — 2019. — 03. — Vol. 16. — P. 172988141983959.

- 45 Mobile Robot Navigation in Indoor Environments: Geometric, Topological, and Semantic Navigation [Text] / R. Barber, J. Crespo, C. Gómez [et al.] // Applications of Mobile Robots. — [S. l. : s. n.], 2018.
- 46 Raaj, Yaadhav. 3D Object Localization using Forward Looking Sonar (FLS) and Optical Camera via particle filter based calibration and fusion [Text] / Yaadhav Raaj, A. John, T. Jin // OCEANS 2016 MTS/IEEE Monterey. — 2016. — P. 1–10.
- 47 Erdem, A. T. Fusing Inertial Sensor Data in an Extended Kalman Filter for 3D Camera Tracking [Text] / A. T. Erdem, A. Ö. Ercan // IEEE Transactions on Image Processing. — 2015. — Vol. 24, no. 2. — P. 538–548.
- 48 van Drongelen, Wim. Chapter 19 - Kalman Filter [Text] / Wim van Drongelen // Signal Processing for Neuroscientists (Second Edition) / Ed. by Wim van Drongelen. — [S. l.] : Academic Press, 2018. — P. 361–374.
- 49 Using Neural Networks to Identify Driving Style and Headway Control Behavior of Drivers [Text] / Charles MacAdam, Zevi Bareket, P. Fancher, Donna Ervin // Vehicle System Dynamics - VEH SYST DYN. — 1998. — 01. — Vol. 29. — P. 143–160.
- 50 Individual Client Strategies for Active Control of Information-Driven Service Construction in IoT-enabled Smart Spaces [Text] / Olga Bogoiavlenskaia, Andrey Vdovenko, Dmitry Korzun, Alexey Kashevnik // International Journal of Embedded and Real-Time Communication Systems (IJERTCS). — 2019. — Vol. 10, no. 2. — P. 20–36.
- 51 Allman, M. TCP Congestion Control [Text] / M. Allman, V. Paxson, E. Blanton // RFC 5681, September. — [S. l. : s. n.], 2009.
- 52 Bogoiavlenskaia, Olga. Intelligent Data Selection in Autonomous Robot Movement [Text] / Olga Bogoiavlenskaia, Dmitry Korzun // Proc. 29th Conf. Open Innovations Association FRUCT. — [S. l. : s. n.], 2021. — May. — P. 47–53.
- 53 Casale, Pierluigi. Human Activity Recognition from Accelerometer Data Using a Wearable Device [Text] / Pierluigi Casale, Oriol Pujol, Petia Radeva. — Vol. 6669. — [S. l. : s. n.], 2011. — 06. — P. 289–296.
- 54 google/googletest: GoogleTest - Google Testing and Mocking Framework [Text]. — [S. l. : s. n.], 2021. — Access mode: <https://github.com/google/googletest>.