

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования «Московский государственный технологический
университет «СТАНКИН»

На правах рукописи



ШЛАЕВ ВИКТОР ИВАНОВИЧ

**ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ КОНТРОЛЯ КАЧЕСТВА ИЗДЕЛИЙ
НА ОСНОВЕ АВТОМАТИЗАЦИИ ПРОЦЕССА ПОКАНАЛЬНОЙ
ОБРАБОТКИ СИГНАЛОВ МНОГОКАНАЛЬНЫХ СИСТЕМ
ПРЕОБРАЗОВАТЕЛЕЙ**

2.3.3. Автоматизация и управление технологическими процессами и
производствами

диссертация на соискание ученой степени
кандидата технических наук

Научный руководитель:
кандидат технических наук, доцент
Бильчук Мария Викторовна

Москва, 2025 г.

ОГЛАВЛЕНИЕ

Введение	5
Глава 1. Информационно-аналитический обзор методов и средств автоматизации процесса обработки данных контроля качества многоканальных систем преобразователей и методов автоматизации процесса разработки технических требований к преобразователям	12
1.1. Информационно-аналитический обзор методов и средств автоматизации процесса контроля качества многоканальных систем преобразователей.	12
1.2. Информационно-аналитический обзор методов и средств автоматизации процесса разработки технических требований к преобразователям	26
1.3. Выводы по Главе 1	29
Глава 2. Разработка структурно-параметрической модели автоматизированного процесса контроля качества многоканальных систем преобразователей	31
2.1. Основные параметры процесса автоматизированного контроля качества многоканальных систем преобразователей	31
2.2. Структурно-параметрическая модель автоматизации процесса поканальной обработки сигналов многоканальных систем преобразователей	35
2.3. Структурная модель поканальной коммутации для автоматизации процесса контроля качества многоканальных систем преобразователей	40
2.4. Выводы по Главе 2	44
Глава 3. Разработка математических моделей и алгоритмов автоматизированного процесса контроля качества многоканальных систем преобразователей и автоматизации процесса разработки технических требований к преобразователям	45
3.1. Математическая модель автоматизации процесса разработки технических требований к преобразователям	45
3.2. Алгоритм автоматизации процесса разработки технических требований к преобразователям	48

3.3. Математическая модель автоматизированного процесса контроля качества многоканальных систем преобразователей	52
3.4. Алгоритм системы автоматизации процесса контроля качества многоканальных систем преобразователей	58
3.5. Выводы по Главе 3	66
Глава 4. Разработка программно-аппаратного комплекса автоматизированного процесса контроля качества многоканальных систем преобразователей и программного модуля автоматизации процесса разработки технических требований к преобразователям	67
4.1. Программно-аппаратный комплекс автоматизированного процесса контроля качества многоканальных систем преобразователей	67
4.2. Программный модуль автоматизации процесса разработки технических требований к преобразователям	85
4.3. Выводы по Главе 4	88
Глава 5. Оценка эффективности использования системы автоматизации процесса контроля качества многоканальных систем преобразователей и модуля автоматизации процесса разработки технических требований к преобразователям.....	90
5.1. Оценка эффективности использования системы автоматизации процесса контроля качества многоканальных систем преобразователей	90
5.2. Оценка эффективности использования модуля автоматизации разработки технических требований к преобразователям.....	95
5.3. Выводы по Главе 5	99
Заключение.....	100
Список источников	102
Приложение А. Свидетельство о государственной регистрации программы для ЭВМ	110
Приложение Б. Акт об использовании результатов работы в производстве	111

Приложение В. Акт внедрения результатов работы в учебный процесс	112
Приложение Г. Листинг программного комплекса автоматизированного процесса контроля качества многоканальных систем преобразователей	113
Приложение Д. Листинг фрагмента обучения и вызова обученной модели программной системы прогнозирования.	152

ВВЕДЕНИЕ

Актуальность исследования. В современных реалиях одним из основных направлений развития электронной промышленности является приборостроение, что отражено в Стратегии развития электронной промышленности Российской Федерации на период до 2030 года (распоряжение Правительства РФ от 17.01.2020 г. №20-р). Согласно документу – электронная промышленность относит к себе, приборы оптоэлектроники и фотоники, приборы квантовой электроники и пьезотехники, радиоэлектронные устройства и системы на кристалле. Одним из проблемных вопросов производственно-технологического характера в отрасли является дефицит современного российского контрольно-измерительного и испытательного оборудования, что оказывает существенное влияние на освоение производства конкурентоспособной высокотехнологичной электронной продукции. Практика концентрации ресурсов на закупке оборудования, в основном иностранного производства, не обеспечивает требуемый уровень развития производства. Применение автоматизированных систем управления и контроля технологических процессов носит фрагментарный характер.

При разработке изделий приборостроения ключевыми элементами являются различные преобразователи и датчики, которые позволяют превращать физическую энергию в электрическую и наоборот. Устройства, с системами подобных преобразователей, находят широкое применение в различных сферах жизни, включая телекоммуникации, энергетику, медицину, промышленность и др. Очевидно, что надежность и точность преобразователей, критически необходима, при разработке подобных систем. Контроль качества изделий приборостроения, представляющих собой многоканальные системы преобразователей, выполняется, преимущественно, с использованием иностранного оборудования или без использования средств автоматизации. Действующие санкции и запреты на доступ к зарубежным

технологиям, оборудованию и материалам, усложняют реализацию бизнес-процессов в отрасли, производство современной конкурентоспособной электронной продукции. В свою очередь, отсутствие средств автоматизации в процессе контроля качества, существенно снижает производительность и качество выпускаемой продукции. В рамках проведения контроля качества изделий приборостроения, согласно стандартам ГОСТ Р МЭК 62127-3-2010, ГОСТ Р ГОСТ Р 53188.1-2008 (МЭК 61672-1:2002) 53566-2009, ГОСТ Р 53576-2009 (МЭК 60268-4:2004), ГОСТ 32783-2014 и другим, является обработка и регистрация сигналов для каждого преобразователя системы. Таким образом актуальной является задача повышения эффективности контроля качества изделий приборостроения на основе автоматизации поканальной обработки сигналов многоканальных систем преобразователей. В настоящее время разработка технических требований к преобразователям физической энергии в электрическую, таких как гидрофоны и вибродатчики, зачастую осуществляется методом множественных экспериментов, что приводит к увеличению времени контроля качества изделий, а также к увеличению временных затрат на использование специализированных лабораторий, в которых проводятся исследования производимой продукции. В условиях отсутствия инструментов предварительного прогнозирования параметров преобразователей на стадии формирования технических требований возрастает риск неоптимальных решений на ранних этапах проектирования. Таким образом, возникает необходимость в разработке методов и моделей, обеспечивающих автоматизацию процесса подготовки на стадии разработки технических требований к преобразователям, позволяющих получать оценочные характеристики изделий без проведения длительных экспериментальных процедур.

Степень разработанности проблемы. Вопросами повышения эффективности контроля качества изделий приборостроения, обработкой сигналов и методами коммутации занимались Овчарук В.Н., Пестриков П.П., Пестрикова Т.В., Николаева Д.В., Зубарев П.В., Миронова Е.Ю., Тубольцев

Ю.В., Еремин И.В., Устинов Д.А., Пурисев Ю.А., Казначеев П.А., Непейна К.С., Козлов С.В., Сергеев И.Ю. и другие. Однако вопрос автоматизации поканальной обработки сигналов многоканальных систем преобразователей недостаточно изучен. Наиболее распространенным оборудованием для проведения контроля качества являются устройства американской компании National Instruments. Проблемы с использованием иностранного оборудования описаны выше. Кроме того, оборудование в рассмотренных решениях позволяет проводить измерения с количеством преобразователей в системе, не превышающих число измерительных каналов оборудования. Имеется возможность расширения числа измерительных каналов, путем добавления дополнительных модулей, что приводит к увеличению габаритов и способна вызвать ряд проблем при проведении измерений.

Проблема разработки технических требований к преобразователям путем множественных экспериментальных исследований заключается в трудоемкости и трудозатратности. Другой подход разработки технических требований к преобразователям основан преимущественно на конечно-элементных и спектральных методах, требующих детальной физико-математической модели, знания свойств материалов и условий эксплуатации. Рассмотренные методы обладают высокой точностью, но являются трудоёмкими и плохо подходят для быстрого прогноза характеристик по результатам массовых измерений.

Целью работы является повышение эффективности контроля качества изделий приборостроения на основе автоматизации процесса поканальной обработки данных многоканальных систем преобразователей и автоматизации процесса подготовки производства на стадии разработки технических требований к преобразователям.

Для достижения поставленной цели, необходимо решить следующие **задачи**:

1. Выполнить информационно-аналитический обзор существующих методов и средств автоматизации процесса обработки данных

многоканальных систем преобразователей, а также методов и средства автоматизации процесса подготовки производства на стадии разработки технических требований к преобразователям по заданными амплитудно-частотными характеристиками.

2. Разработать структурно-параметрическую модель автоматизированного процесса поканальной обработки данных многоканальных систем преобразователей.

3. Выявить взаимосвязи и установить зависимости автоматизированного процесса разработки технических требований к преобразователям.

4. Разработать математические модели и алгоритмы автоматизированного процесса поканальной обработки данных многоканальных систем преобразователей и автоматизации процесса подготовки производства на стадии разработки технических требований.

5. Разработать программно-аппаратный комплекс поддержки автоматизированного процесса контроля качества многоканальных систем преобразователей и автоматизации процесса подготовки производства на стадии разработки технических требований к преобразователям.

6. Провести апробацию и внедрение разработанного программно-аппаратного комплекса в реальное производство, провести оценку эффективности его работы.

Объект исследования – процессы автоматизации обработки данных многоканальных систем преобразователей и автоматизации подготовки производства на стадии разработки технических требований.

Предмет исследования – контроль качества многоканальных систем преобразователей и подготовка производства на стадии разработки технических требований.

Научная новизна работы заключается в следующем:

Выявлены взаимосвязи и установлены зависимости автоматизированного процесса разработки технических требований к преобразователям.

Разработана структурно-параметрическая модель автоматизированного процесса поканальной обработки данных многоканальных систем преобразователей.

Разработаны математические модели и алгоритмы автоматизированного процесса поканальной обработки данных многоканальных систем преобразователей и автоматизированного процесса разработки технических требований к преобразователям.

Теоретическая значимость исследования заключается в совершенствовании теоретических основ проведения контроля качества изделий приборостроения и разработки технических требований к преобразователям, обеспечивающих сокращение времени выполнения измерительных и проектных работ. Полученные результаты могут использоваться при проведении контрольных испытаний, а также при разработке технических требований к преобразователям.

Практическая значимость исследования состоит в разработанном программно-аппаратном комплексе поддержки автоматизированного процесса поканальной обработки данных многоканальных систем преобразователей и модуле автоматизации процесса разработки технических требований к преобразователям по заданным амплитудно-частотным характеристикам. Применение разработанных программно-аппаратных средств позволяет сократить время выполнения конструкторско-технологических и контрольно-измерительных работ.

Достоверность и обоснованность результатов исследования подтверждается корректностью применяемого математического аппарата, результатами проводимых измерительных операций.

Методы исследования. В работе использовались методы математического анализа, структурное моделирование, математическое моделирование, численные методы, методы обработки сигналов, методы корреляции.

Положения, выносимые на защиту:

1. Структурно-параметрическая модель системы автоматизации процесса контроля качества многоканальных систем преобразователей.
2. Математическая модель автоматизации процесса контроля качества многоканальных систем преобразователей;
3. Алгоритм функционирования модели автоматизации процесса контроля качества многоканальных систем преобразователей;
4. Математическая модель и алгоритм автоматизации процесса подготовки производства на стадии разработки технических требований к преобразователям.

Степень достоверности и апробация выполненных исследований. Результаты, полученные в диссертационной работе, подтверждаются соответствием экспериментальных и теоретических исследований, разработкой оригинальной модели системы автоматизации контроля качества многоканальных систем преобразователей, разработкой программно-аппаратного комплекса систем автоматизации поканальной обработки данных многоканальных систем преобразователей. Произведена государственная регистрация разработанной программы для ЭВМ. Получен акт об использовании результатов работы по автоматизации процесса контроля качества многоканальных систем преобразователей на АО «Акустический институт имени академика Н.Н. Андреева» (г. Москва).

Работа выполнена при финансовой поддержке Министерства науки и высшего образования Российской Федерации (проект No FSFS-2024-0012).

Основные положения диссертационной работы докладывались и обсуждались на международных, всероссийских, региональных и межвузовских научно-технических конференциях: International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), 2021 г.; международная научная конференция «Самарские чтения», 2022 г.; XIV всероссийская конференция с международным участием МАШИНОСТРОЕНИЕ: традиции и инновации, 2021г.; студенческие научно-

практические конференции «Автоматизация и информационные технологии» (АИТ-2021, АИТ-2023), 2021 г., 2023 г.

Публикации по теме работы. По теме диссертации опубликовано 9 научных публикаций : 3 статьи в изданиях, рекомендованных ВАК РФ, 1 научная работа – в издании, входящем в базу данных Scopus, 4 публикации – в материалах конференций, получено 1 свидетельство о государственной регистрации программы для ЭВМ.

Соответствие паспорту специальности. Научная работа соответствует формуле научной специальности 2.3.3 — «Автоматизация и управление технологическими процессами и производствами» в пунктах 2 – «Автоматизация контроля и испытаний» и 3 – «Методология, научные основы, средства и технологии построения автоматизированных систем управления технологическими процессами (АСУТП) и производствами (АСУП), а также технической подготовкой производства (АСТПП) и т. д.».

Структура и объем. Диссертация изложена на 155 страницах машинописного текста, содержит 49 рисунков и 2 таблицы, состоит из введения, 5 глав, заключения и списка литературы из 53 наименований.

ГЛАВА 1. ИНФОРМАЦИОННО-АНАЛИТИЧЕСКИЙ ОБЗОР МЕТОДОВ И СРЕДСТВ АВТОМАТИЗАЦИИ ПРОЦЕССА ОБРАБОТКИ ДАННЫХ КОНТРОЛЯ КАЧЕСТВА МНОГОКАНАЛЬНЫХ СИСТЕМ ПРЕОБРАЗОВАТЕЛЕЙ И МЕТОДОВ АВТОМАТИЗАЦИИ ПРОЦЕССА РАЗРАБОТКИ ТЕХНИЧЕСКИХ ТРЕБОВАНИЙ К ПРЕОБРАЗОВАТЕЛЯМ

Приборостроение, как важнейшая отрасль машиностроения, занимается разработкой и производством средств измерений, обработки и представления информации. При разработке многоканальных систем преобразователей, ключевыми элементами которых являются различные преобразователи и датчики, которые позволяют превращать физическую энергию (звук, свет, давление и т.д.) в электрическую и наоборот. Подобными датчиками являются микрофоны, гидрофоны, электролюминесцентные датчики и др. Устройства, с системами подобных преобразователей, находят широкое применение в различных сферах жизни, включая телекоммуникации, энергетику, медицину, промышленность и др.

1.1. Информационно-аналитический обзор методов и средств автоматизации процесса контроля качества многоканальных систем преобразователей

Для регулирования процесса разработки и производства преобразователей, в России, как и во многих других странах, предусмотрены соответствующие стандарты.

Так, например [2] содержит требования для гидрофонов с пьезоэлектрическими чувствительными элементами, разработанными для измерений ультразвуковых полей импульсного или непрерывного излучения,

генерируемого ультразвуковым оборудованием, а также для гидрофонов, используемых для измерений в воде и для гидрофонов с предварительным усилителем или без него. В стандарте описываются основные параметры гидрофонов, которые необходимо учитывать при разработке подобных устройств, например чувствительности по приему и излучению, импедансы и другие характеристики, которые рассчитываются по амплитудно-фазовым частотным характеристикам.

Устанавливает требования к электроакустическим характеристикам для приборов трех видов, предназначенных для измерения звука [3]. Частотные характеристики данных приборов должны удовлетворять приведенному стандарту. Данные показатели рассчитываются с использованием амплитудно-фазовых частотных характеристик.

Стандарт [4] устанавливает требования проведения приемо-сдаточных испытаний для измерения электроакустических параметров микрофонов. Устанавливаются допустимые погрешности измерений и указываются проверяемые характеристики, основная часть которых рассчитываются с использованием амплитудно-фазовых частотных характеристик.

Стандарт [5] устанавливает требования безопасности и контроль индуктивно-проводных датчиков, применяемых в системах железнодорожной автоматики и телемеханики, отвечающих за безопасность движения железнодорожных поездов. В данном стандарте приводятся требования безопасности, определяемых вибрациями, многократными и одиночными ударами. Показатели определяются по амплитудно-фазовым частотным характеристикам.

Стандарт [6] регулирует методы калибровки датчиков вибрации и удара. Устанавливаются требования к методам и средствам испытаний и измерений для первичной калибровки датчиков. Данный стандарт распространяется на измерения в определенных диапазонах частот и амплитуд. Из стандарта следует, что проведение измерений и калибровки датчиков вибрации и удара

основываются на амплитудно-фазовых частотных характеристиках преобразователей.

Стандарт [7] устанавливает требования на распределенные волоконно-оптические датчики температуры и определяет методы испытаний по определению их основных характеристик. Определение параметров влияния номинальной температуры и времени прогрева осуществляется с по амплитудно-фазовых частотным характеристикам датчика.

Стандарт [8] устанавливает требования безопасности, конструктивные и функциональные характеристики датчиков контроля давления пневматического типа. Приводятся требования к отклонениям от установочной точки, дрейф установочной точки и гистерезис. Данные показатели рассчитываются с использованием амплитудно-фазовых частотных характеристик преобразователя.

Стандарт [9] регулирует методы приемочных и типовых испытаний датчиков, имеющих стандартные аналоговые выходные сигналы или стандартные аналоговые пневматические выходные сигналы. Основные рассчитываемые характеристики, приведенные в стандарте, вычисляются с использованием АФЧХ.

Из приведенных стандартов, можно сделать вывод, что основные параметры датчиков и преобразователей, при проведении контроля качества, рассчитываются с использованием амплитудно-фазовых частотных характеристик. Для вычисления требуемых характеристик преобразователя, необходимо представить описание сложного периодического сигнала непрерывного времени в виде разложения по системе простых(базисных) функций. Для этого можно использовать быстрое преобразование Фурье [12].

Из [10] можно сделать вывод, что базовой формой непрерывного периодического сигнала, может является синусоидальный сигнал.

Для расчета требуемых характеристик во время проведения измерений, необходимо рассмотреть существующие средства, методы, алгоритмы и решения процесса регистрации и обработки оцифрованных данных

непрерывных периодических сигналов, а также способы проведения данных измерений в многоканальных системах преобразователей (Таблица 1).

В работе [13] описаны методики регистрации и анализа спектральных характеристик сигналов акустической эмиссии при решении задач мониторинга состояния промышленных объектов. Приводится структурная схема программного комплекса сбора и обработки сигналов, которая реализована на базе оборудования американской компании National Instruments. Различные модели данного оборудования позволяют расширять управляемые каналы АЦП/ЦАП, путем включения дополнительных измерительных плат. Программный комплекс позволяет реализовать различные методы постобработки данных. В качестве языка программирования использовался LabVIEW.

Оборудованием компании National Instruments реализован способ многоканальной регистрации на базе DAQ устройств в работе [14]. Каждый из приведенных модулей способен обрабатывать до 8 каналов. Комплекс позволяет в реальном времени проводить анализ входного сигнала с предысторией. Такой подход позволяет обеспечить помехоустойчивость и позволяет регистрировать временную форму сигнала на внешние устройства для последующего анализа.

В работе [15] рассматривается способ обработки многоканальной системы преобразователей. Представлена схема коммутации электрических сигналов, основанная на получении на выходе схемы попарных разностей потенциалов между измерительными электродами. Оборудованием является

Схема коммутации электрических сигналов представлена в работе [3]. Она заключается в получении на выходе схемы попарных разностей потенциалов между измерительными электродами. В качестве аппаратных средств используется оборудование National Instruments

В [16] приводится описание многоканальной системы сбора и регистрации данных диагностик плазмы установки ИТЭР. Система разделена на две части, одна из частей располагается на минимальном расстоянии от

детекторных помещений, а вторая располагается непосредственно в помещении. Такой подход позволяет проводить измерения в условиях высоких радиационных и электромагнитных полей. В качестве измерительного оборудования использовался модуль National Instruments.

В системе регистрации сигналов со стриповых полупроводниковых детекторов [17], приводится способ регистрации данных 64 канальной системы. Источником триггерного сигнала может служить внешний сигнал или детектор, регистрирующий прохождение частицы ионизирующего излучения через стрипы, амплитуда сигнала с которого может использоваться для спектрометрии частиц. Узел управления и считывания сигналов IDE1140, выполненный на программируемой логической интегральной схеме EP3C16Q240, служит для обработки, накопления и передачи данных. Приведены основные характеристики системы: интегральная нелинейность. В соответствии с числом стрипов детектора микросхема содержит 64 спектрометрических тракта. Выходным сигналом каждого тракта является напряжение, полученное путем преобразования входного заряда, зарегистрированного в определенный момент времени. Коммутация каналов основывается на мультиплексорах и дешифраторах.

В работе [18] рассказывается о разработке аппаратного и алгоритмического обеспечения, для применения пьезорезонансных чувствительных элементов в составе прецизионных датчиков физических величин. Описаны основные преимущества пьезорезонансных чувствительных элементов. Приведен принцип работы и принципиальное устройство разработанного многоканального измерителя. А также приведена сравнительная характеристика получившегося измерителя с уже существующими. Разработанная многоканальная система регистрации содержит пять измерительных каналов, способных работать в диапазоне рабочих частот ПЧЭ от 32 кГц до 20 МГц. Каждый измерительный канал формируется с помощью микроконтроллера Atiny 24, на вход внешней синхронизации которого подается частотный сигнал от ПЧЭ. Встроенные

средства в микроконтроллере Atiny 24 позволяют измерять частоту с погрешностью до ± 1 Гц. Далее полученное значение частоты в качестве параметра передается по интерфейсу SPI в центральный микроконтроллер Atmega 328P.

В [19] излагаются основы акустико-эмиссионных методов неразрушающего контроля. Проводится подробный анализ свойств современных информационно-измерительных систем акустико-эмиссионной диагностики, даются рекомендации по их дальнейшему совершенствованию. Обосновывается необходимость использования методов спектрального анализа при исследовании процесса трещинообразования. Анализируются результаты математического моделирования энергетических характеристик сигналов акустической эмиссии, дается их сравнительный анализ с характеристиками, полученными при испытании образцов сварных соединений и элементов конструкций. Приводятся рекомендуемая методика корректировки и обработки спектральных функций, методика проведения испытаний. В качестве аппаратных средств используется оборудование Nationa Instruments. Применение спектрального анализа в многоканальных системах регистрации сигналов акустической эмиссии рассматривается в работе [20] Метод акустической эмиссии используется как средство анализа состояния материалов и конструкций, неразрушающего контроля и диагностики их во время работы. Важное преимущество перед другими методами состоит в том, что он реагирует только на развитие действительно опасных дефектов и способен контролировать большие объемы и поверхности без их сканирования. В статье приведены результаты разработки алгоритмов и программ акустико-эмиссионной системы «EMISS-2», предназначенных для анализа акустических сигналов, и связи их с данными, полученными на более ранних стадиях эксперимента. Корреляция и соотношение сигнал-шум были оценены при помощи алгоритмов, основанных на частотном анализе. Алгоритм регистрации сейсмоакустических сигналов с многоканальной

коммутацией при неизвестном уровне высокочастотных помех рассматривается в [21]

Указана необходимость оцифровки нескольких низкочастотных сигналов с помощью минимального набора аппаратных средств. Описывается минимальный вариант аппаратной реализации такой системы предполагает использование компьютера с аналого-цифровым преобразователем в составе интегрированной звуковой карты и внешнего коммутатора сигналов с необходимым согласованием. Приводится конфигурация коммутация сигналов для последующей оцифровки и учет возможных высокочастотных помех в сигналах. Описан алгоритм захвата данных применительно к разработанной градиентной установке регистрации сейсмоакустического сигнала.

В работе [22] приведены одно- и двухэтапные алгоритмы обработки дружно- и быстрофлуктуирующего отраженного сигнала (ОС) в радиолокационном обнаружителе-измерителе угловых координат в условиях внешних маскирующих помех в свободном пространстве и при наличии мешающих отражений (МО) с неизвестными параметрами. Алгоритмы основаны на операциях пространственной компенсации помех (ПКП) в каждом периоде повторения, оценивании доплеровского сдвига частоты МО, их когерентной компенсации (КК), обеления результатов обработки во временной области, оценивании средней мощности и коэффициента междупериодной корреляции ОС с последующим построением и максимизацией логарифма функции отношения правдоподобия (ФОП). Приведены результаты моделирования предлагаемых алгоритмов, свидетельствующие об их работоспособности и эффективности. В работе [23] обоснованы вариант оптимального и модификации квазиоптимальных алгоритмов оценивания пеленга цели в обзорной двухкоординатной радиолокационной станции с подсистемой пространственной компенсации помех. Алгоритмы основаны на операциях внутривременной компенсации помех с «обелением» их результатов во временной области путем деления на

оценку среднеквадратического значения напряжения суммы взвешенных внутренних шумов и некомпенсированных остатков внешних помех, вычисления ожидаемого сигнала для всех возможных положений цели в пределах области неопределенности, построение и максимизацию функции правдоподобия в виде взаимной корреляционной функции «обеленной» реализации и опорного сигнала. Приведены результаты оценки среднеквадратических ошибок оценивания угловых координат цели.

В [24] описана многоканальная методика регистрации сигналов детекторов с различными интервалами усреднения сигнала для каждого канала. Методика позволяет повысить надежность обнаружения радиоактивных источников, расположенных на различной глубине движущегося транспортного средства при непостоянстве его скорости в зоне контроля за счет оптимизации величины интервала усреднения сигналов различной длительности.

В работе [25] определена структура систем обработки на основе предварительного некогерентного накопления доплеровских спектров при корреляционной пространственновременной обработке для пространственно многоканальных РЛС. Оценена возможность применения квазиоптимальных алгоритмов, основанных на стробировании принимаемого сигнала по задержке с помощью эквидистантной совокупности стробов. Приведены результаты моделирования системы ПВ обработки при наличии стробирования и некогерентного накопления доплеровских спектров.

В [26] показано, что применение современных пассивно-активных систем радиолокации повышает эффективность контроля воздушного пространства и снижает его стоимость. Представлено общее описание принципов функционирования пассивноактивных системах радиолокации, а также предложен вариант многоканальной обработки сигналов в таких системах, позволяющий с большей эффективностью использовать поле подсвета, создаваемое разнотипными источниками радиоизлучений. Установлено, что основной проблемой при многоканальной обработке

является устранение неоднозначности, вызванной различным временем прихода отраженных сигналов от одного объекта, формируемых разнесенными на местности передающими мачтами телевизионного и радиовещания, а также различная частота Доплера. Предложены пути по уменьшению влияния данных факторов, основанные на многоканальной параллельной цифровой обработке сигналов.

В [27] рассмотрены вопросы, связанные с особенностями многоканальной обработки сигналов на основе методов сверхразрешения с использованием системы с квазиколевым расположением приемников. Проведено сравнение выходной характеристики при использовании классического метода обработки (формируется диаграмма направленности) и метода сверхразрешения (формируется пеленгационный рельеф). В качестве основы алгоритма обработки сигналов использовано свойство 2π -периодичности угловой зависимости характеристик направленности приемников и мощности сигнала. Исследованы вопросы повышения разрешающей способности по угловому направлению принимаемых сигналов при многоканальной обработке в системе приемников с кольцевым расположением. Показана возможность значительного улучшения разрешающей способности при переходе от классического метода к методу сверхразрешения.

В [28] исследованы существующие подходы к пространственно-временной обработке полезных сигналов на фоне помех. Рассмотрено условие узкополосности (факторизуемости) в пространственно-временном смысле при обработке полезных сигналов на фоне воздействия широкополосных помех в пространственно-временном смысле. Показаны сложности достижения условий факторизуемости (разделяемости) пространственно-временных структур, встречающиеся в радиотехнических системах с крупноапертурными антенными решетками, которые используют для достижения больших дальностей обнаружения высокоскоростных и малоразмерных целей с высокими точностями измерения координат и разрешающими способностями.

Проведен анализ путей реализации выполнения условия пространственно-временной узкополосности в радиотехнических системах с антенными решетками с использованием управляемых линий задержки и с использованием фазовращателей.

В [29] рассмотрены возможности повышения дальности обнаружения объектов многоканальной оптико-электронной системой в двух аспектах: выбор статистического критерия принятия решения и рациональное построение приемной оптической системы. Выбраны способы интеграции каналов на алгоритмическом и конструктивном уровнях. Выбран критерий качества поиска минимума среднего риска по Байесу. Для программно-аппаратных комплексов автоматизации контроля и диагностирования с использованием различных критериев при измерении и регистрации данных от датчиков разработана методика многоточечных измерений на основе мультиверсионного подхода к формированию измеряемых параметров технологического процесса в [30]. В результате выполненного анализа определяются места размещения датчиков для отбора проб, а также количество и относительное расположение точек размещения. Полученный план размещения датчиков многоточечной измерительной системы позволяет реализовать мультиверсионное голосование при формировании результирующего значения измеряемых параметров технологического процесса, основанное на получении N -го количества измеряемых данных. K – обобщенный критерий качества технологического процесса рассчитывается по $n-1$ входам.

Встраиваемое программное обеспечение многоканальных адаптивных систем регистрации сигналов [31], способно настраивать рабочие каналы АЦП, изменять значения нулевого уровня и коэффициентов усиления по каналам, настраивать частоту дискретизации, способна измерять аналоговые сигналы, временно хранить полученные результаты. Основано на ARM Cortex-M7. Информации о возможности данным программным средством коммутировать каналы не приводится.

Специальное (прикладное) программное обеспечение многоканальной системы регистрации звуковых сигналов и радиолокационной информации [32] способна к многоканальной записи звуковых (аналоговых и цифровых) и радиолокационных источников. текущий контроль записи всех типов данных; создание, хранение и воспроизведение фрагментов записей, содержащих информацию за выбранный промежуток времени; внесение изменений в конфигурацию программы, определяющих режимы и параметры записи источников информации данных. В описании также не приводится система коммутации данных. Данный программный продукт позволяет регистрировать и работать с данными со всех каналов.

SeismoStation — программа для регистрации многоканальных сейсмоакустических данных [33]. Программа предназначена для приёма цифровой информации с выхода многоканального 24 разрядного АЦП, организации ее в многоканальные массивы данных, визуализации для оперативного контроля качества принимаемой информации и сохранения преобразованной информации в файлы формата SEG-Y. Программа позволяет осуществить выбор шага дискретизации сигналов, числа каналов и длины трасс. В зависимости от используемой аппаратуры программа позволяет регистрировать сигналы от 16 до 128 каналов при шаге дискретизации от 0,0625 мс до 4,0 мс. Запуск регистрации может осуществляться как от самого компьютера, так и от внешнего источника. Минимальный интервал времени между двумя запусками составляет от 1 с до 2 с. Визуализации данных осуществляется в виде временных разрезов и в виде многоканальных сейсмограмм с выбором масштаба и детальности изображения. Изобретение [34] служит для повышения скорости обмена данными, обеспечении автоматической адаптации к изменяющимся условиям радиосвязи, обеспечении работы в многоканальном режиме. Для этого система передачи данных предназначена для работы в диапазоне частот от 1,5 до 30,0 МГц для передачи информации: четырьмя автономными каналами

номинальной мощностью 500 Вт на 4 антенны; четырьмя автономными каналами на две или одну общую антенну; на одной частоте со сложением мощностей двух или четырех каналов в одной антенне с номинальной суммарной выходной мощностью соответственно 1 кВт и 2 кВт; для приема информации по восьми каналам приема и формирования данных панорамного обзора рабочего диапазона частот; для обмена цифровыми данными (файлами) на скоростях до 9,6 кбит/с в каждом канале (от одного до восьми каналов на верхней и нижней боковых полосах). Коммутация происходит программно и система способна работать в различных режимах. Содержит способна обрабатывать данные по 8 каналам.

Программа модуля многоканальной обработки биопотенциалов приводится в [35]. Программа предназначена для приема, автоматического регулирования усиления сигналов от многоканальной системы регистрации биопотенциалов. Реализован алгоритм распознавания жестов по входным сигналам от многоканальной (до восьми каналов) системы регистрации биопотенциалов.

Программное средство для сбора и первичной обработки сигналов с распределенной сети пьезодатчиков представлена в [36]. Программа предназначена для автоматизации сбора данных с системы пьезодатчиков с использованием многоканального управляемого коммутатора и цифрового запоминающего осциллографа - приставки к ЭВМ. Реализован функционал обработки измеряемых сигналов с применением усреднения по серии замеров, скользящего среднего, устранения средней величины измеренного электрического напряжения, оконных функций в частотной и/или временной области. Графический интерфейс программы позволяет пользователю контролировать процесс сбора и обработки данных и представляет результаты измерений в наглядной форме в виде графиков зависимостей электрических напряжений от времени или частоты. Программа может применяться при решении задач ультразвукового неразрушающего контроля тонкостенных конструкций распределенной сетью пьезодатчиков, а также в учебных целях

для изучения особенностей обработки экспериментальных волновых сигналов.

В [37] рассматривается использование многоканального полупроводникового преобразователя в системе стабилизации тока для формирования поля компенсационной обмотки. Описывается разработка системы адаптивного управления с алгоритмом сигнальной настройки и эталонной моделью для многоканального преобразователя. Эталонная модель представляется в пространстве переменных состояния. Приводится модель разработанной системы стабилизации, выполненная в САПР Matlab/Simulink, и графики выходного тока, полученные в результате моделирования. Система коммутации основывается на использовании диодов.

Наиболее распространенным оборудованием, для проведения регистрации и обработки электрических сигналов, являются устройства американской компании National Instruments. Измерительный потенциал данного оборудования можно описать «Один канал модуля – один канал преобразователя». Т.е. при проведении измерений данное оборудование позволяет проводить измерения с количеством преобразователей в системе в количестве, не превышающих число измерительных каналов оборудования. Имеется возможность расширения числа измерительных каналов, путем добавления дополнительных модулей. Данная особенность способна вызвать ряд проблем при проведении измерений. Например, при проведении контроля качества многоканальной системы гидроакустических преобразователей, данная система помещается в водную среду. Из-за больших габаритов, измерительное оборудование указанного вида, помещается в воздушной среде на некотором расстоянии от каналов преобразователей. Электрический сигнал на выходе преобразователей поступает в канал АЦП оборудования по кабелю. В данном примере для достижения точности измерений необходимо учитывать физические параметры каждого кабеля для каждого преобразователя, такие как длина, материал, сечение, учитывать параметры оплетки и другие характеристики, влияющие на качество переданных данных.

Рассмотренные методы обработки основываются на использовании быстрого преобразования Фурье. Наиболее распространенным подходом при обработке сигналов, является использование преобразования на полном наборе входных данных, число которых варьируется от нескольких сотен тысяч до миллионов. Данная процедура может препятствовать параллельной регистрации и обработки сигналов и вносить значительное замедление в процесс контроля качества. Другим популярным подходом, является поиск области полезного сигнала с последующей обработкой узкой области. Данный подход является некоторой оптимизацией, но в данной ситуации процедура поиска является замедлителем.

Для поканальной регистрации электрических сигналов приводились различные системы коммутации. Применение схем коммутации позволяет обрабатывать большее количество каналов преобразователей, по сравнению с моделью «один канал АЦП/ЦАП – один канал преобразователя». Однако реализация схем коммутации, например с использованием диодов, является плохо масштабируемой и имеет ограничения в значениях используемого напряжения.

Для минимизации потери данных, предлагается, при проведении измерений, поместить измерительное оборудование в непосредственную близость к каналам преобразователей. Задачу увеличения числа каналов преобразователей предлагается с использованием схемы коммутации. Для увеличения производительности процесса обработки, предлагается обрабатывать только «полезный» сигнал преобразователя. Для минимизации операций поиска положения во времени «полезного» сигнала предлагается синхронизировать излучение и прием преобразователей таким образом, что местоположение во времени «полезного» сигнала будет одинаковым для всех каналов преобразователей на всех измеряемых частотах.

1.2. Информационно-аналитический обзор методов и средств автоматизации процесса разработки технических требований к преобразователям

Разработка технических требований к преобразователям является одной из важных задач приборостроения. Одной из задач является установка физических параметров преобразователей по заданным амплитудно-частотным характеристикам. Проведение измерений для проверки требуемых АЧХ преобразователя по физическим параметрам трудоемкий процесс, который требует использования измерительных лабораторий и работу квалифицированного персонала для проведения контрольных испытаний. Перед разработкой физических моделей преобразователей, проводится математическое или программное моделирование устройств различными методами. Так, например в [38] используется метод конечных элементов для расчета элементов пьезоэлектрического преобразователя с учетом реальных условий эксплуатации в газовой среде. В рассмотренной работе разрабатывается модель пьезоэлектрического преобразователя по требуемым характеристикам, на основе расчетов и теоретических основ.

В работе [39] рассматривается новый способ конечно-элементного моделирования, позволяющий учитывать влияние встроенных механических напряжений на механические свойства диэлектрической мембраны. Модель в работе разработана на основе теоретических данных влияния механических напряжений на свойства диэлектрической мембраны.

В статье [40] рассмотрен вопрос использования пространственных векторов для описания переменных электромеханического преобразователя переменного тока. Особенностью модели является учет дополнительной обмотки, установленной между подвижной частью и статором двигателя. Определены проектные параметры задачи.

В работе [41] излагается схема применения метода спектральных элементов для моделирования динамического поведения пьезоэлектрического преобразователя сложной формы в трехмерном случае. Предлагаемая в данной работе математическая модель необходима для обобщения гибридного метода для описания возбуждения и измерения бегущих волн в протяженных конструкциях в трехмерном случае. Рассматриваются различные типы электродирования пьезоэлектрических преобразователей, а также различные граничные условия для описания процессов возбуждения (актуаторы) и измерения (сенсоры).

В статье [42] предлагается конечно-элементная модель, которая соответствует реальному гидроакустическому преобразователю. Основное внимание уделяется модулю электрического импеданса, смещению в различных точках поверхности преобразователя, а также акустическому давлению, создаваемому излучателем в воде на расстоянии 1 метр от его поверхности.

В рассмотренных работах основной упор делается на теоретические данные о свойствах преобразователей физической энергии в электрическую. Основным подходом в рассмотренных работах является создание конечно-элементной модели преобразователя с последующим расчетом параметров в системах моделирования.

Предлагается для моделирования преобразователей использовать модель машинного обучения, основанную на методах градиентного бустинга. В работе [43] с использованием методов градиентного бустинга происходит прогнозирование ключевых параметров промышленного комплекса. Для решения задачи прогнозирования физических параметров преобразователей предлагается использовать данные измерений процесса контроля качества соответствующих преобразователей для обучения модели, предварительно выявив ключевые физические параметры преобразователей, которые влияют на соответствующие амплитудно-частотные характеристики.

Результаты анализа представлены в Таблице 1.

Таблица 1 – Результаты анализа источников.

Группа метода	Подход	Краткое описание	Недостатки	Предложения
Спектральный анализ	Быстрое преобразование Фурье (БПФ)	Преобразование временног о сигнала в спектр	Обработка полного массива данных → большое время вычислений. Затрудняет параллельную многоканальную обработку.	Обрабатывать только заранее синхронизированную область «полезного» сигнала. Исключить поиск полезного фрагмента — обеспечить одинаковые временные позиции сигналов синхронизацией возбуждения и приема.
	Сужение окна анализа (обработка сегмента сигнала)	БПФ применяется не ко всему сигналу, а к найденному фрагменту	Поиск полезного участка → вычислительные затраты. Ошибки локализации фрагмента.	Полная временная синхронизация каналов: место полезного сигнала известно заранее → окно фиксированное и одинаковое.
Корреляционный анализ	Автокорреляция, взаимная корреляция	Выделение структуры сигнала, определение задержек	Высокие вычислительные затраты. Уязвимость к помехам при большом количестве каналов.	Использовать корреляцию только для контрольных каналов. Далее обработку вести по заранее синхронизированным сигналам по однократному поиску.
Многоканальная обработка	Увеличение числа каналов АЦП	Расширение числа каналов с увеличением числа АЦП	Низкая масштабируемость. Требуется использования большого числа модулей АЦП для обработки соответствующего числа каналов.	Использовать компактное оборудование, размещённое рядом с преобразователями. Использовать систему коммутации, основанную на модели демультиплексоров соединенных в каскадную схему.
Усреднение, скользящее окно, оконные функции	Используются в системах с большими шумами	Снижение шумов, подготовка данных	Увеличение времени обработки. Потеря детализации сигналов преобразователей.	Применять только при необходимости. Основной упор — на физическую синхронизацию и уменьшение кабельных влияний.
Триггерная регистрация	Использование внешнего триггера	Запись начинается при событии	Не подходит для непрерывных контролируемых процессов преобразователей. Может смещать временное положение сигнала.	Использовать жестко определенное время излучения — время записи фиксированное.
Конечно-элементные методы	Классический МКЭ	3D/2D моделирование технических параметров	Требуют длительных расчётов. Нужны точные параметры преобразователей, их измерение — трудоемко. Варьирование конструкции требует многократных пересчетов.	Машинное обучение (градиентный бустинг): обучать модель на реальных измерениях для быстрого предсказания технических характеристик.

1.3. Выводы по Главе 1

1. Амплитудно-фазовые частотные характеристики (АФЧХ) являются базовой измеряемой величиной практически для всех типов преобразователей – гидрофонов, микрофонов, оптоэлектронных, пьезорезонансных, вибрационных и др. Анализ нормативных документов показал, что именно на основе АФЧХ рассчитываются ключевые электроакустические, динамические, температурные и надёжностные параметры датчиков при проведении контроля качества.

2. Процедуры измерений и контроля многоканальных систем преобразователей в действующих стандартах и в большинстве исследованных работ базируются на методах спектрального анализа, преимущественно на быстром преобразовании Фурье. При этом выполняется либо обработка полного массива данных, либо выделение области полезного сигнала при обработке каждого измерения. В первом случае увеличивается объем и сложность вычислений, во втором случае поиск сигнала создает дополнительные задержки.

3. Существующие средства регистрации основаны в основном на использовании измерительного оборудования National Instruments. Для них характерна архитектура «один канал АЦП/ЦАП – один канал преобразователя». Это накладывает ограничения на масштабируемость многоканальных систем, приводит к увеличению стоимости комплексов и усложняет учёт влияния длинных кабельных линий при удалённом размещении аппаратуры.

4. Способы многоканальной коммутации, реализованные в анализируемых работах, позволяют расширять число обслуживаемых преобразователей, однако зачастую ограничены по напряжению, плохо масштабируются и требуют сложной схемотехники. При этом большинство

отечественных и зарубежных решений не обеспечивают гибкую программную перенастройку каналов.

5. На основании анализа установлено, что для повышения точности и производительности контроля качества многоканальных систем преобразователей целесообразно размещать измерительное оборудование максимально близко к преобразователям, для снижения влияния физических свойств соединителей. Для увеличения количества измеряемых преобразователей в системе без расширения аппаратных каналов АЦП следует применять схему многоканальной коммутации. Для уменьшения объема обрабатываемых данных целесообразно искать область полезного сигнала. Предварительная синхронизация излучаемого и принимаемого сигналов обеспечивает возможность однократного поиска позиции для сокращения времени обработки.

6. В ходе анализа методов автоматизации процесса разработки технических требований к преобразователям установлено, что стандартный подход основан преимущественно на конечно-элементных и спектральных методах, требующих детальной физико-математической модели, знания свойств материалов и условий эксплуатации. Рассмотренные методы обладают высокой точностью, но являются трудоёмкими и плохо подходят для быстрого прогноза характеристик по результатам массовых измерений.

7. В качестве альтернативного направления предложено использование методов машинного обучения, в частности градиентного бустинга, что позволяет строить автоматизированные модели разработки технических требований к преобразователям по экспериментальным данным контроля качества.

ГЛАВА 2. РАЗРАБОТКА СТРУКТУРНО-ПАРАМЕТРИЧЕСКОЙ МОДЕЛИ АВТОМАТИЗИРОВАННОГО ПРОЦЕССА КОНТРОЛЯ КАЧЕСТВА МНОГОКАНАЛЬНЫХ СИСТЕМ ПРЕОБРАЗОВАТЕЛЕЙ

2.1. Основные параметры автоматизированного процесса контроля качества многоканальных систем преобразователей

Для разработки структурной и математической моделей автоматизации процесса поканальной обработки многоканальных систем преобразователей необходимо выяснить основные параметры. Основным параметром разрабатываемых моделей автоматизации, является сигнал преобразователя.

Основными параметрами электрических импульсов, характерных для всех импульсных сигналов независимо от формы относят амплитуду U , длительность t_c и, для периодических последовательностей импульсов, период T или частотой $f = \frac{1}{T}$. Наиболее простым типом электрических импульсов является постоянный импульс, который можно описать законом:

$$u(t) = U, \quad -\infty < t < +\infty \quad (2.1)$$

Данный сигнал задается постоянной амплитудой и действует в заданном промежутке времени.

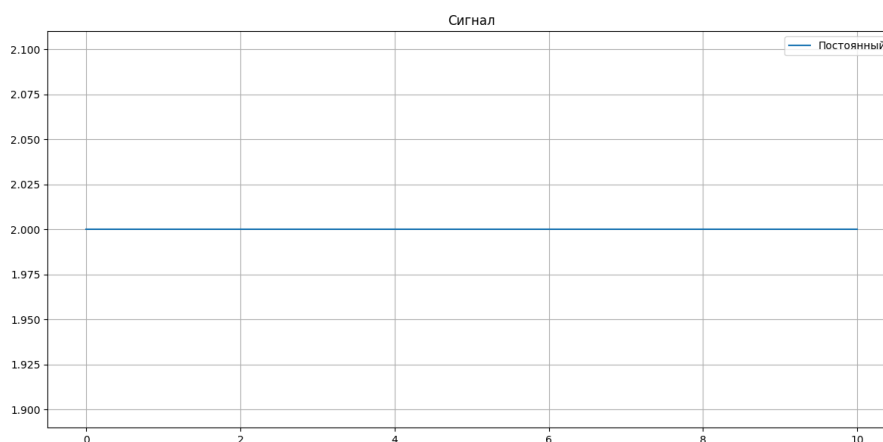


Рисунок 2.1. – Постоянный сигнал

При усложнении постоянного сигнала, можно получить прямоугольный периодический сигнал:

$$u(t) = \begin{cases} U, & \frac{t}{T} - \left[\frac{t}{T} \right] \leq t_c, \\ 0, & \frac{t}{T} - \left[\frac{t}{T} \right] > t_c \end{cases}, \quad (2.2)$$

где функция (2.2) принимает значение U – амплитуды, когда промежуток времени t на период T меньше длительности сигнала t_c и 0 в противном случае (Рисунок 2.2.).

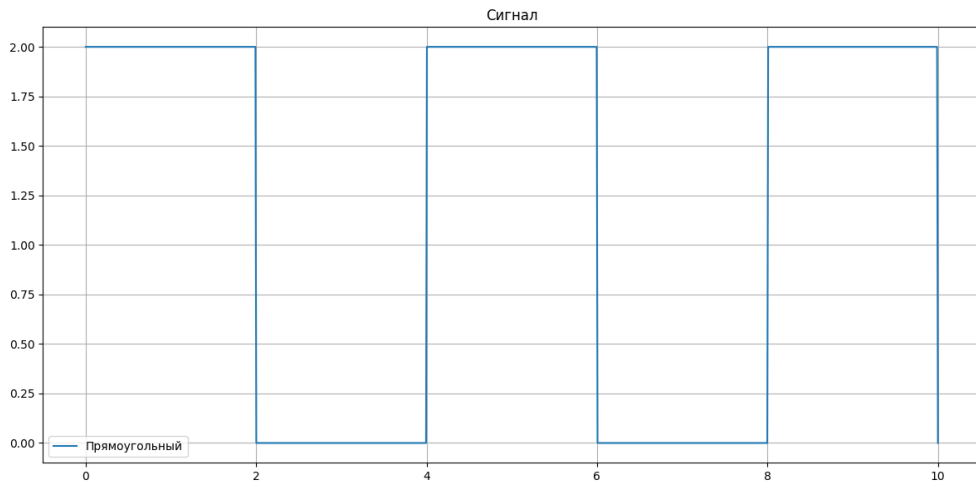


Рисунок 2.2. – Периодическая прямоугольная форма волны.

При задании отрицательной амплитуды задается меандр (Рисунок 2.3):

$$u(t) = \begin{cases} U, & \frac{t}{T} - \left[\frac{t}{T} \right] < t_c, \\ -U, & \frac{t}{T} - \left[\frac{t}{T} \right] \geq t_c \end{cases}, \quad (2.3)$$

где $\frac{t}{T} - \left[\frac{t}{T} \right]$ – остаток от деления текущего времени на период

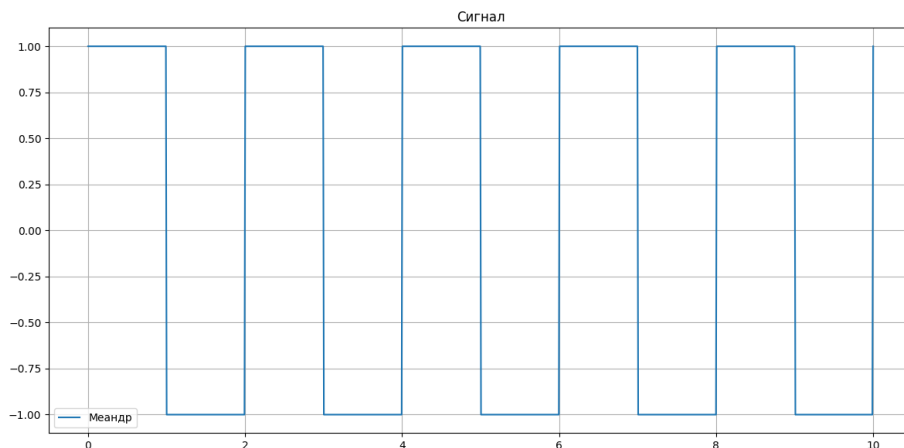


Рисунок 2.3. – Волна в форме меандра

Существует пилообразный тип электрического сигнала, который можно представить как:

$$u(t) = 2U \left(\frac{t}{T} - \left[\frac{1}{2} + \frac{t}{T} \right] \right) \quad (2.4)$$

В сигнале пилообразной формы происходит нарастание весь период T , а в конце периода мгновенно падает до начального значения (Рисунок 2.4).

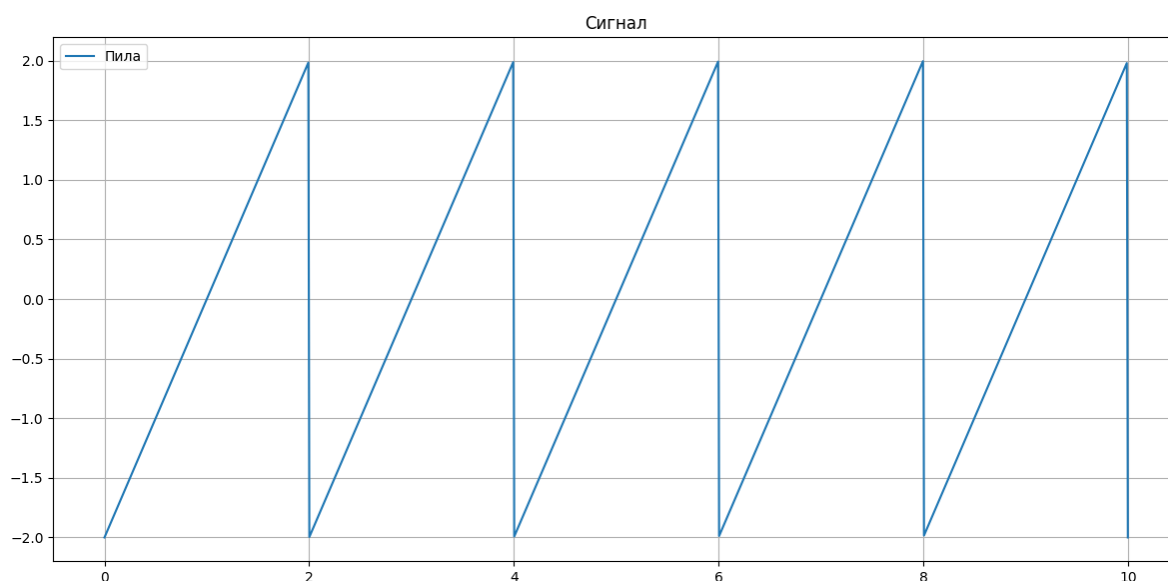


Рисунок 2.4. – Пилообразная форма волны.

Базовой формой электрического сигнала, согласно [10], является синусоидальный или тональный сигнал:

$$u(t) = U \sin(\omega t + \varphi_0), \quad (2.5)$$

где U – амплитуда сигнала, ω – циклическая частота, показывающая, на сколько радиан изменяется фаза колебания за одну секунду, φ_0 – начальная фаза колебаний, определяющая значение полной фазы в момент времени $t = 0$. Иначе уравнение (2.5) можно переписать через частоту сигнала f :

$$u(t) = U \sin(2\pi f t + \varphi_0) \quad (2.6)$$

где U – амплитуда сигнала, f – частота сигнала, φ_0 – начальная фаза колебаний, определяющая значение полной фазы в момент времени $t = 0$. Амплитуда сигнала синусоидальной или тональной формы изменяется в соответствии с тригонометрической функцией (Рисунок 2.5.).

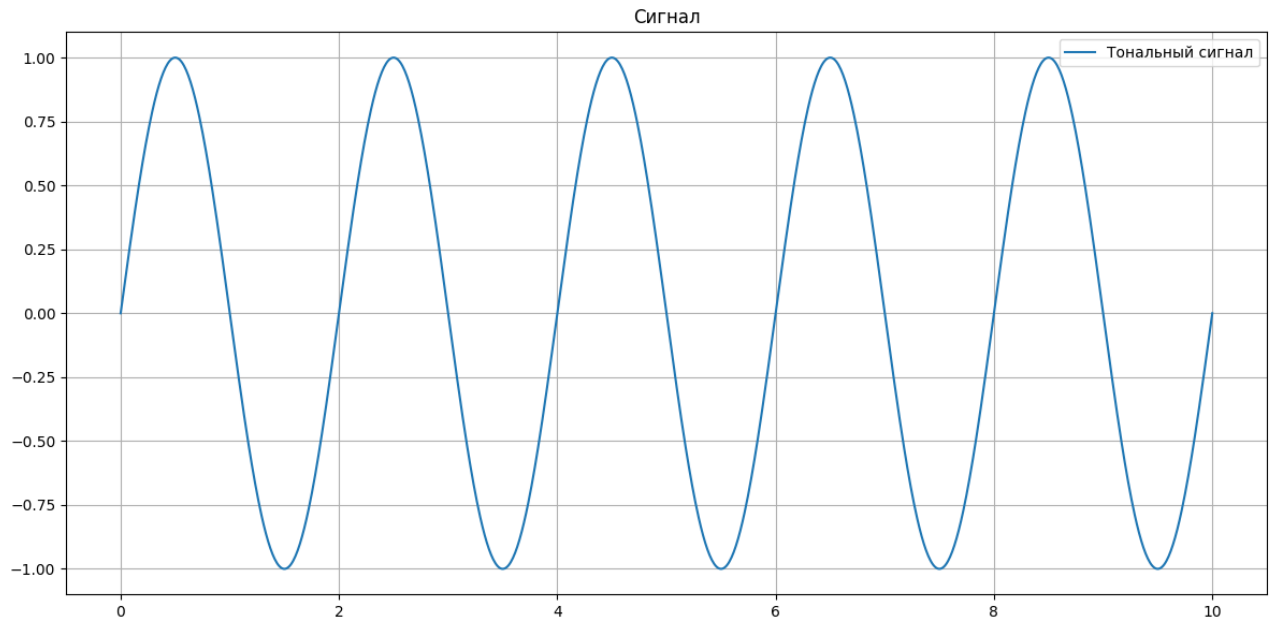


Рисунок 2.5. – Синусоидальная (тональная) форма волны

Для проведения измерений необходимо реализовать функционал генерации приведенных типов волн, в особенности сигналов синусоидальной формы. В общем случае излучаемый сигнал имеет (Рисунок 2.6) комбинированную форму периодического сигнала, например тонального, с нулевым. Данная форма необходима для минимизации наложений и помех полезного импульса при проведении измерений.

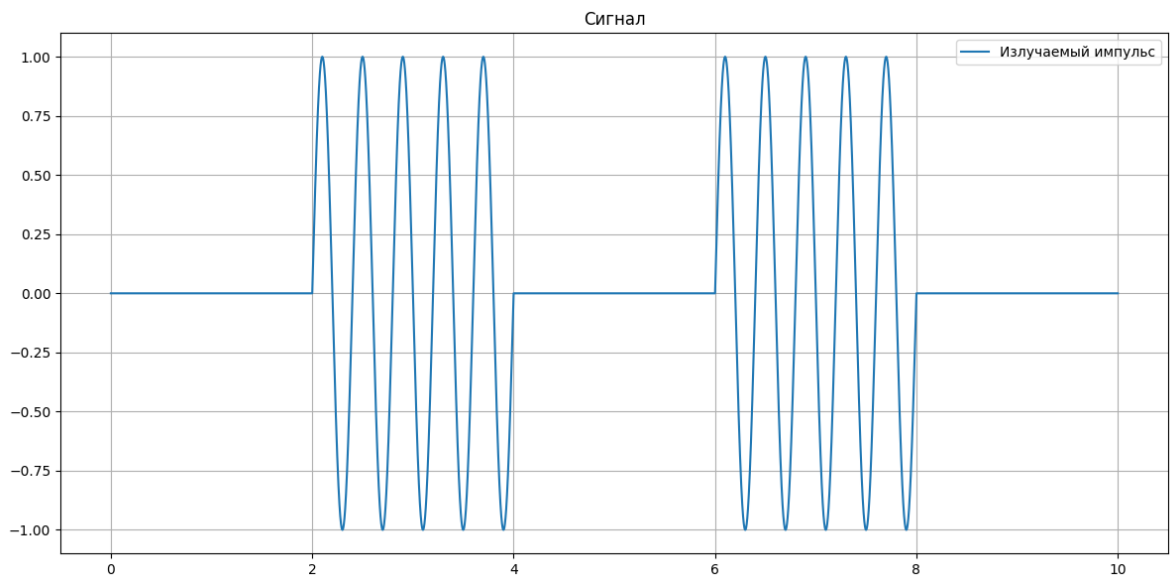


Рисунок 2.6. Пример общей формы генерируемого сигнала

Данную форму с полезным тональным сигналом можно описать как:

$$u(t) = \begin{cases} U \sin(2\pi f t + \varphi_0), & t \in [T_0, T_0 + l_s] \\ 0, & t \in [T_1, T_1 + l_z] \end{cases}, \quad (2.7)$$

где T_0 – период повторения тонального сигнала, l_s – длительность тонального импульса, T_1 – период нулевого сигнала, l_z – длительность нулевого импульса.

2.2 Структурно-параметрическая модель автоматизации процесса поканальной обработки сигналов многоканальных систем преобразователей

Контроль качества многоканальных систем преобразователей на предприятиях производится согласно [11]. Схема проведения контроля качества приводится на рисунке 2.7.



Рисунок 2.7. – Схема проведения контроля качества согласно нормативным документам.

Методику проведения для преобразователей схематично можно изобразить как на рисунке 2.8. Где некое излучающее устройство подает импульс на канал системы и затем регистрируется выходной сигнал.

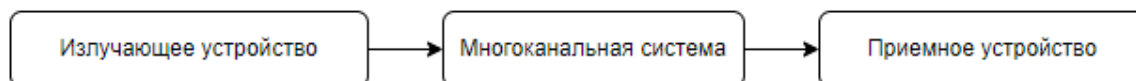


Рисунок 2.8. – Методика контроля качества для системы преобразователей.

В некоторых, более общих случаях в процессе измерений добавляются различные контрольные излучатели или контрольные приемные устройства (Рисунок 2.9). Например, в случаях измерения гидроакустических датчиков используются контрольный излучающий гидрофон и контрольный гидрофон. Сигнал поступает на излучатель, далее от излучателя сигнал идет к контрольному приемнику и одновременно на канал системы. Затем эти сигналы регистрируются и проводится расчет характеристик.



Рисунок 2.9. – Схема измерений с излучателем и приемником.

В более общем случае возникает необходимость подавать сигнал через усилитель. В том же примере с гидродатчиками необходимо подавать сигналы высоких амплитуд, которые большинство используемых генераторов не могут выдать. Для этого сигнал пропускается через усилитель и подается к излучателю (Рисунок 2.10). Усиленный сигнал поступает на контрольный приемник и канал системы и оттуда, пропускается либо через делитель, т.к. большинство осциллографов ограничены в амплитуде приема, либо через усилитель, если выходной сигнал составляет микровольты. Например, при измерении датчиков света необходимо, чтобы к источнику излучения поступало напряжение, амплитудой тысячи вольт. На выходе получают подобные же значения. Для регистрации данных измерений необходимо пропустить эти сигналы через делитель, зафиксировать коэффициент и зарегистрировать сигналы.



Рисунок 2.10. – Общая схема проведения контроля качества преобразователей

Проведение контроля качества включает в себя, в общем случае, измерения по приему (Рисунок 2.11) и по излучению (Рисунок 2.12).

В режиме по приему формируется сигнал определенной формы, который является опорным и регистрируется каналом АЦП, предварительно проходя через измерительный усилитель. Опорный сигнал регистрируется на выходе некоторого контрольного принимающего устройства. Другим каналом АЦП регистрируются данные с выхода измеряемого канала преобразователя. Далее на основе зарегистрированных данных происходит расчет необходимых параметров. В общем случае каналы АЦП и генератор могут находиться на значительном расстоянии, следовательно для проведения подобных измерений потребуется участие двух квалифицированных сотрудников.

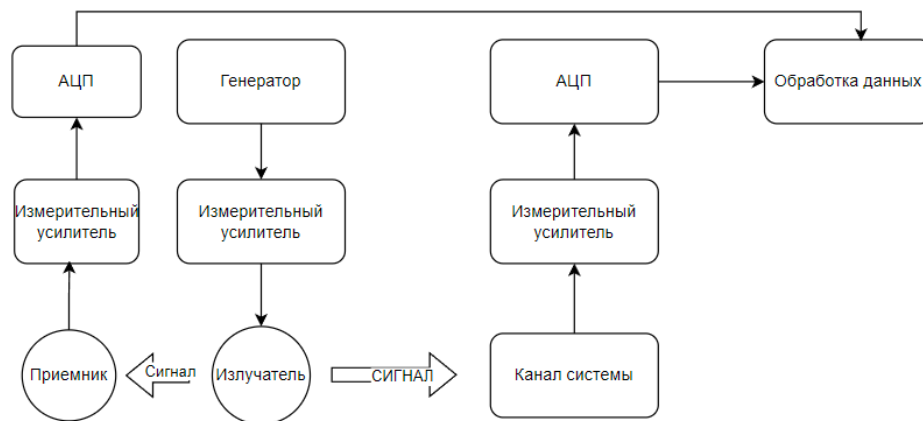


Рисунок 2.11. – Условная схема измерений по приему

В процессе проведения измерений в режиме излучения, опорный сигнал так же регистрируется приемным устройством или же происходит регистрация сигнала на входе канала преобразователя. Остальной процесс проведения измерения схож с измерениями в режиме приема преобразователя.

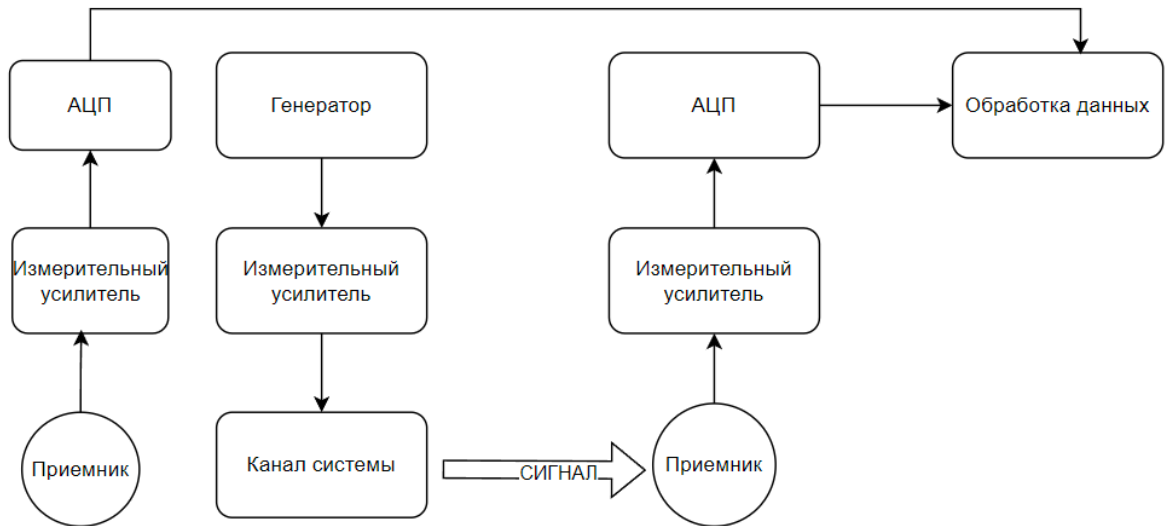


Рисунок 2.12. – Схема излучения

Упрощенная структурная модель автоматизации [47] процесса контроля качества представлена на рисунке 2.13.

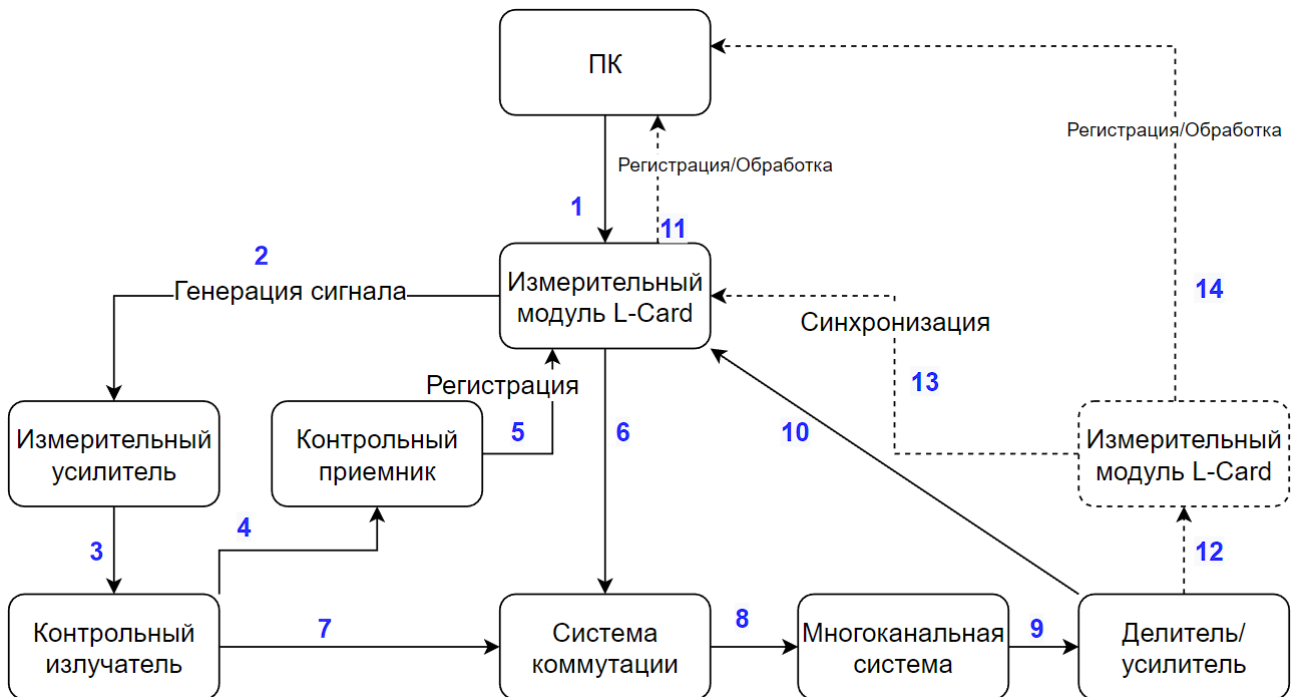


Рисунок 2.13. – Упрощенная модель автоматизации процесса контроля качества.

Персональный компьютер дает команду Измерительному модулю начать работу, условно нажимается кнопка старт. Что происходит. Модуль генерирует сигнал, которые идет к измерительному усилителю, тот в свою очередь передает этот сигнал контрольному излучателю. Дальше сигнал идет

одновременно к контрольному приемнику (4), оттуда регистрируется измерительным модулем, одновременно включается один из каналов (6) и на него подается после включения сигнал (7). Т.к. каналов может быть несколько сотен, то для переключения была разработана схема коммутации (Рисунок 6), она представляет собой каскадную модель из переключателей. Данную модель можно использовать совокупно с логической моделью управляемого демультиплексора.

Данный подход позволяет управлять $2^n - 2$ каналами, где n – количество цифровых выходов. Из этого количества вычитается 2, т.к. значение 0 будет использоваться в качестве всех выключенных каналов, а максимальное значение – в качестве всех активных каналов. При разработке модели коммутации следует оставить один цифровой выход на кодирование выключения и включения всех каналов системы. Существует множество способов реализации подобной схемы, в зависимости от измерительного модуля и требований к измерениям, например можно реализовать модель коммутации с использованием управляемых релейных модулей. Далее сигнал проходит через канал системы и поступает на делитель или усилитель. Дальше в обычном случае сигнал идет в измерительный модуль (10) и подается к ПК.

Для случая, когда необходимо регистрировать сигнал с канала системы в непосредственной близости, еще один модуль помещается рядом с системой. Модули имеют возможность синхронной работы. Таким образом второй измерительный модуль, (12) включается по команде модуля основного (1) и уже от этого модуля (14) данные поступают к персональному компьютеру.

Возможность расчета АЧХ может быть на стадии передачи данных от модулей. Но также часто требуется регистрировать данные для последующей обработки. Важную роль в этом имеет синхронизация регистрации опорного сигнала и сигнала выходного из канала. Необходимо, чтобы промежуток времени между принятием этих сигналов был для каждой частоты, для каждого канала одним и тем же. В случае работы двух модулей синхронизация

позволяет это делать. В случае работы одного модуля возможна реализация синхронизации программным образом.

Процесс проведения измерений по приему сводится к тому, что генератор сигналов излучает электрический сигнал определенной формы. Данный сигнал подается к измерительному усилителю откуда поступает к контрольному излучателю. Далее поступает к контрольному приемнику и каналу системы. Данные из приемника и канала системы поступают через измерительные усилители на каналы АЦП. Далее происходит оценка значения амплитуды и вычисление необходимых характеристик. Далее меняется частота излучаемого сигнала. При измерении частот нужного диапазона происходит вывод устройства из рабочей среды, переключение канала и проведение описанных действий для другого канала преобразователя. В режиме по излучению в качестве излучателя используется канал системы. Регистрация опорного сигнала осуществляется либо контрольным приемником, либо регистрируется сигнал перед подачей на канал излучения. Одной из важных задач является автоматизации коммутации каналов.

2.3. Структурная модель поканальной коммутации для автоматизации процесса контроля качества многоканальных систем преобразователей

При разработке схемы коммутации [46] нужно исходить из имеющегося количества управляемых цифровых выходов выбранного измерительного модуля (16 цифровых выходов). Предлагается использовать каскадную схему переключателей (Рисунок 2.14.). Для управления потребуются соединить параллельно каждую линейку переключателей. Подавая на каждую линейку логический 0 или 1, в результате переключается один из каналов.

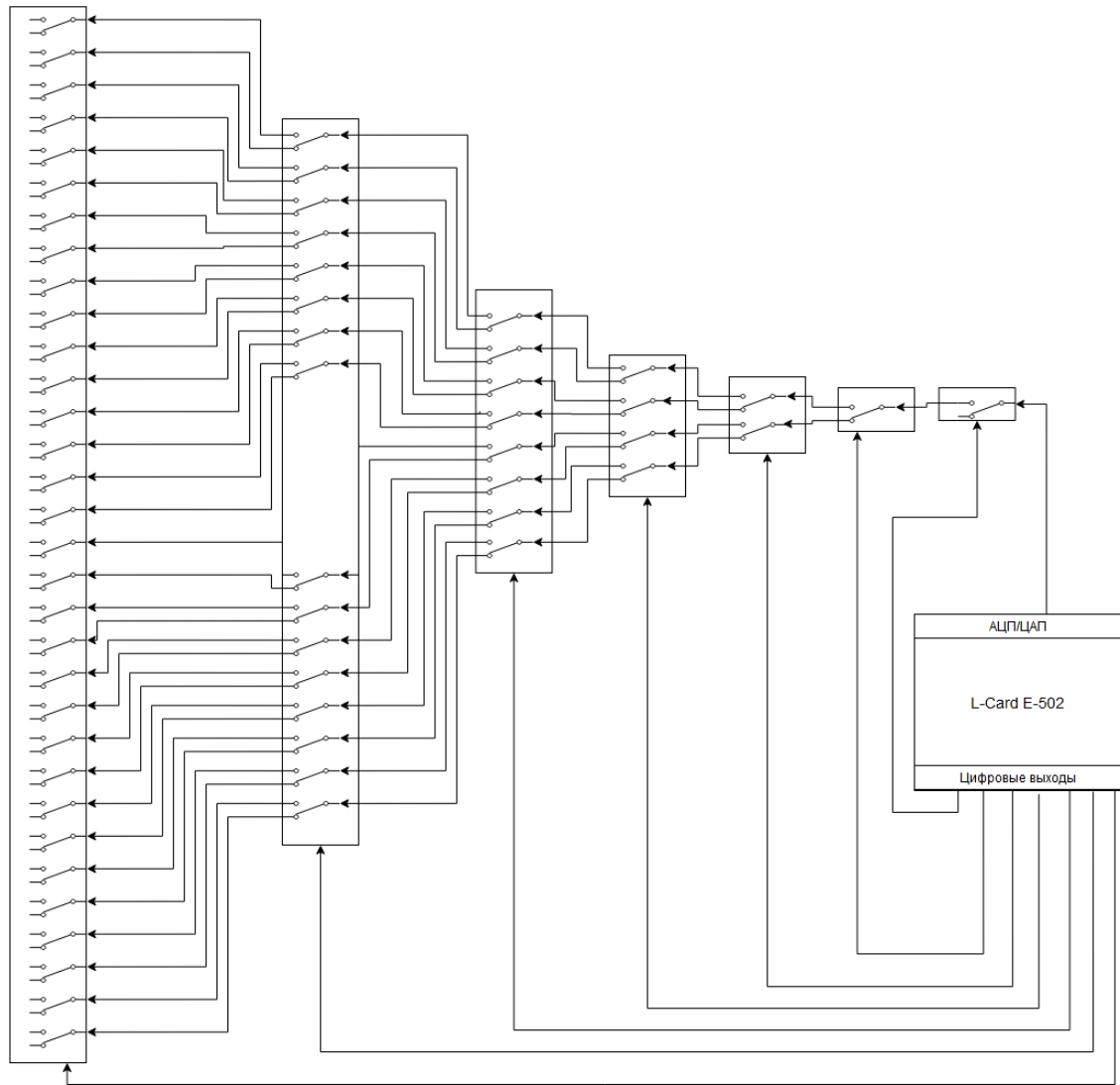


Рисунок 2.14. – Каскадная схема переключателей

Для физической реализации предлагается в качестве переключателей использовать электромеханические реле. Выбор обосновывается тем, что данный вид реле способен пропускать через себя электрические сигналы высоких напряжений, а управление можно делегировать цифровым выводам. Для масштабируемости системы предлагается использовать восьмиканальные релейные модули и на них реализовать схему демультиплектора 3 на 8. Параллельно с платой пропуска излучаемого или принимаемого сигнала, предлагается добавить плану управления питанием платы (Рисунок 2.15.).

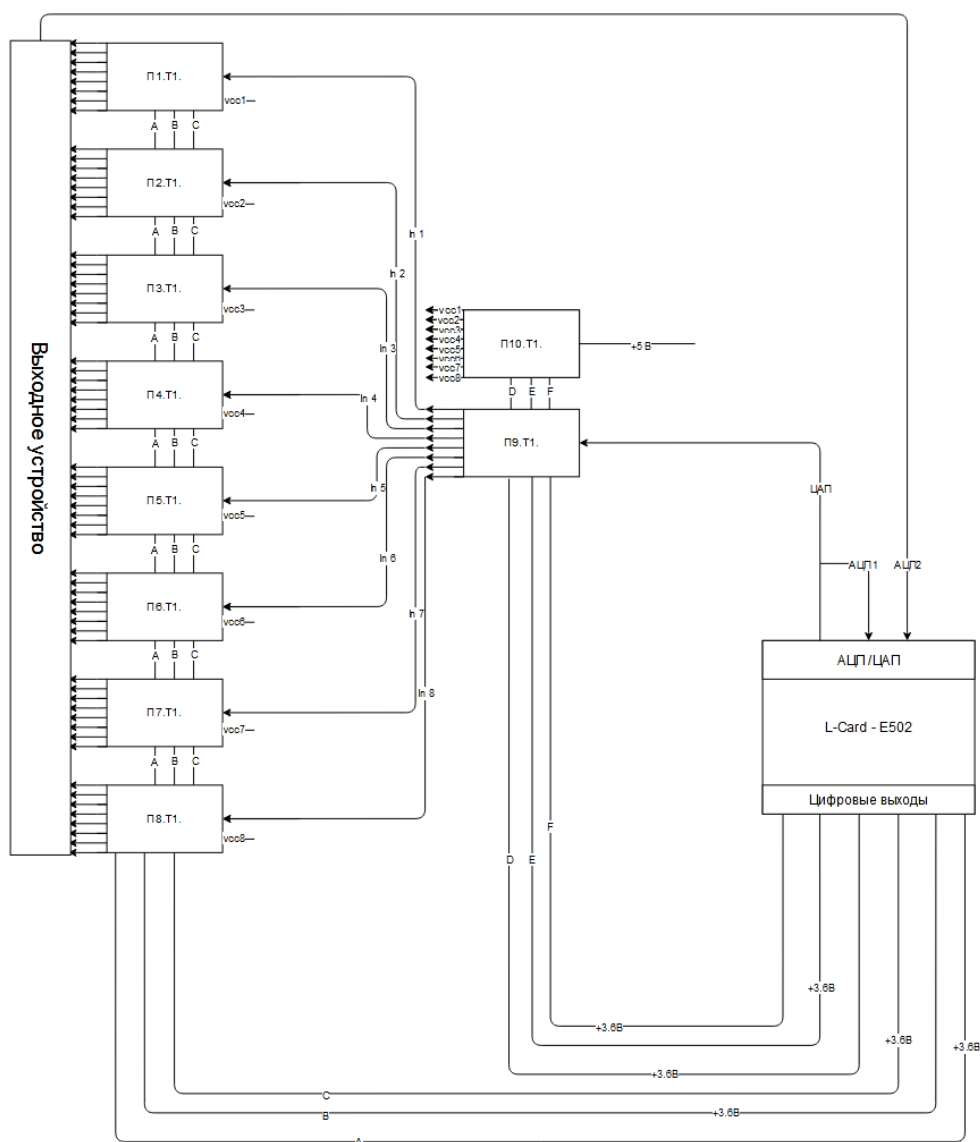


Рисунок 2.15. – Схема коммутации на основе релейных модулей.

Для повышения надежности системы, вводится плата усиления сигнала коммутации реле (Рисунок 2.16.). Данная плата управляется цифровыми выходами и в качестве коммутируемого сигнала используется постоянный ток, который подается на линейки плат в качестве логических 0 и 1. Приведенная схема является масштабируемой. На рисунках представлена схема для 64 каналов преобразователей. Добавление следующего каскада позволяет обрабатывать 512 каналов. Количества цифровых выходов измерительного модуля достаточно для реализации последующих каскадов. Однако, можно предположить, что ограничение даже в 512 каналов системы

является условным, т.к. многоканальных систем с большим числом каналов преобразователей небольшое количество.

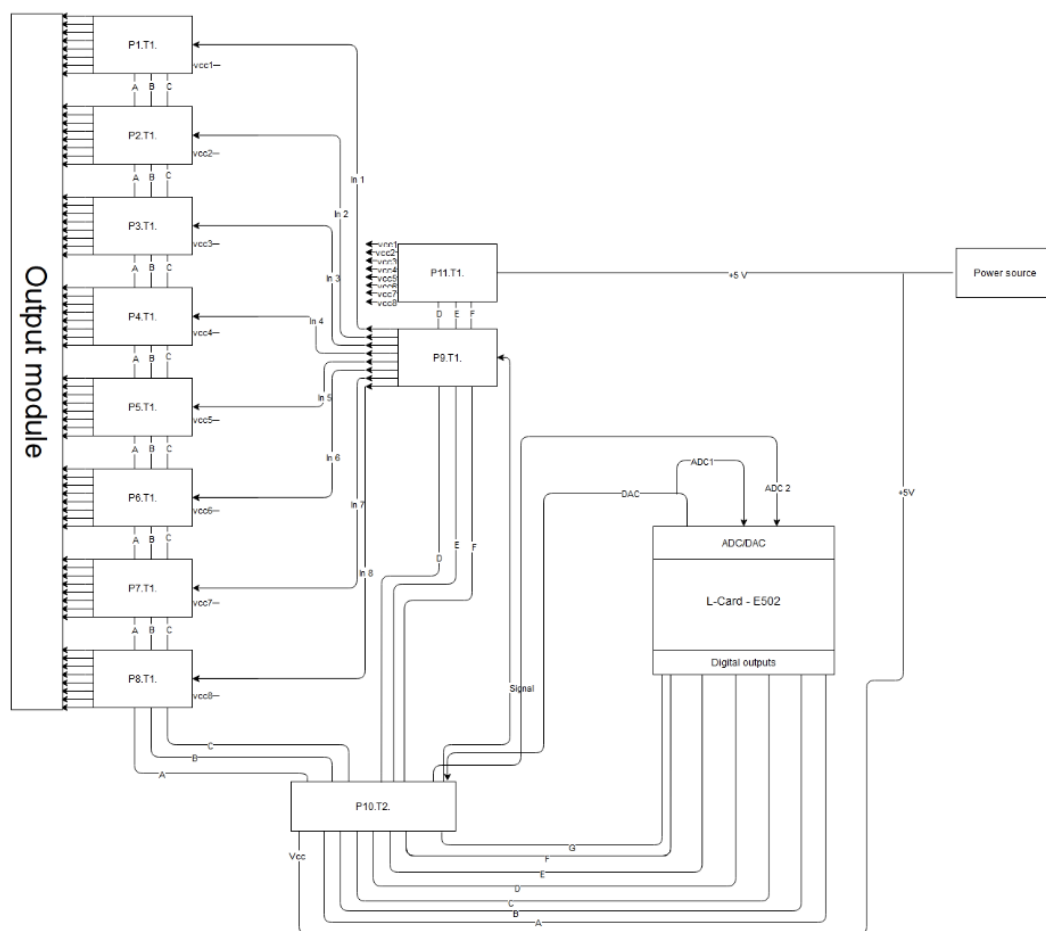


Рисунок 2.16. – Каскадная схема коммутации сигналов с платой усиления сигнала

Приведенная схема является масштабируемой. На рисунках представлена схема для 64 каналов преобразователей. Добавление следующего каскада позволяет обрабатывать 512 каналов. Количества цифровых выходов измерительного модуля достаточно для реализации последующих каскадов. Однако, можно предположить, что ограничение даже в 512 каналов системы является условным, т.к. многоканальных систем с большим числом каналов преобразователей небольшое количество. Но даже при необходимости добавления следующего каскада, появляется возможность измерений 4096 каналов преобразователей. Количества цифровых выходов измерительного модуля достаточно для реализации подобной схемы коммутации. А габариты плат увеличиваются пропорционально измеряемым

каналам, т.о. схема коммутации может располагаться в непосредственной близости к каналам преобразователей.

2.4. Выводы по Главе 2

1. При проведении контроля качества многоканальных систем преобразователей, согласно стандартам, используются электрические импульсы различной формы, например в форме меандра, пилообразной или тональной форме. В общем виде используется комбинированная форма сигналов.

2. Основными параметрами электрических импульсов, характерных для всех импульсных сигналов независимо от формы относят амплитуду, длительность, и, для периодических последовательностей импульсов, период или частоту.

3. Согласно нормативным документам, проводятся измерительные работы по контролю качества многоканальных систем преобразователей в режимах по приему и по излучению.

4. Разработанная структурная модель автоматизированного процесса контроля качества многоканальных систем преобразователей учитывает все необходимые требования проведения измерений без использования средства автоматизации с добавлением измерительного модуля, коммутирующего устройства и средств синхронизации сигналов.

5. Разработанная структурная модель коммутации основана на демультимплексорах соединенных в каскадную схему переключателей.

ГЛАВА 3. РАЗРАБОТКА МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ И АЛГОРИТМОВ АВТОМАТИЗИРОВАННОГО ПРОЦЕССА КОНТРОЛЯ КАЧЕСТВА МНОГОКАНАЛЬНЫХ СИСТЕМ ПРЕОБРАЗОВАТЕЛЕЙ И АВТОМАТИЗАЦИИ ПРОЦЕССА РАЗРАБОТКИ ТЕХНИЧЕСКИХ ТРЕБОВАНИЙ К ПРЕОБРАЗОВАТЕЛЯМ

3.1. Математическая модель автоматизации процесса разработки технических требований к преобразователям

Математическая модель, для автоматизации процесса разработки технических требований к преобразователям, основана на модели прогнозирования физических параметров преобразователей по заданным амплитудно-частотным характеристикам. Для разработки математической модели прогнозирования физических параметров преобразователей по заданным амплитудно-частотным характеристикам, необходимо определить основные параметры преобразователей и степень влияния на общую математическую модель.

Частотная характеристика преобразователя определяется его резонансными свойствами и общей реакцией на входные колебания. Ее изменение напрямую связано с конструктивными и материальными параметрами. Ключевым законом для резонансных систем является соотношение [44]:

$$f_c = \frac{1}{2\pi} \sqrt{\frac{k}{m}}, \quad (3.1)$$

где k — эффективная жёсткость системы, а m — эффективная масса. Рост массы снижает резонансную частоту, а увеличение жёсткости — повышает её.

Основными характеристиками являются:

Материалы – механические свойства материалов корпуса и мембраны, такие как плотность, упругость, акустическая проводимость. Плотность влияет на массу: $f \propto \frac{1}{\sqrt{\rho}}$. Упругий модуль определяет жёсткость: $f \propto \sqrt{E}$. Акустическая проводимость влияет на ширину полосы: высокая проводимость расширяет полосу, снижая пиковое усиление.

Размеры – линейные размеры чувствительного элемента и корпуса преобразователя. Габариты преобразователя и его элементов определяют акустические резонансные частоты. Например для, длинных полых корпусов или трубок первых порядков резонансы выполняется соотношение $\frac{\lambda}{4}: f_c \approx \frac{c}{4L}$, где c - скорость звука, L - характерный размер канала. В объёмных резонаторах [45] основная резонансная частота задается формулой:

$$f_H = \frac{c_0}{2\pi} \sqrt{\frac{S}{CL_{eff}}}, \quad (3.2)$$

где S – площадь сечения горловины, L_{eff} - ‘эффективная длина канала (с учетом коррекции концов), V – объем резонатора, а c_0 - скорость звука. Отсюда видно, что увеличение размеров корпуса уменьшает частоту, а увеличение горловины или уменьшение ее длины – повышает. Аналогично, линейные размеры чувствительного элемента влияют на собственную жёсткость: более длинные или большие плиты низкочастотнее из-за меньшей жёсткости на растяжение/сжатие. Таким образом геометрия и размеры согласуются с материальными свойствами, определяя полный спектр резонансов системы

Масса – общая масса устройства, которая влияет на его резонансные частоты. Увеличение массы движущихся элементов (корпус, мембрана, рули и т.д.) снижает резонансную частоту. Поэтому большие по массе и объёму преобразователи имеют более низкие резонансы, тогда как облегчённые — более высокие. Увеличение массы можно использовать для целенаправленного сдвига полосы вниз. Например, в вибродатчиках добавление груза в точках крепления понижает резонанс.

Геометрия – форма и конструкция корпуса, которые определяют направление и диапазон чувствительности. Форма влияет на моды колебаний и распределение массы. Компактная форма повышает жёсткость, увеличивая частоту. Объёмные формы или полости, напротив, снижают частоту.

Способ установки – тип крепления влияет на граничные условия: жёсткое крепление увеличивает частоту, гибкое – снижает. Монтирование через демпфирующие элементы уменьшает жёсткость и сдвигает полосу вниз.

Моделирование таких преобразователей "вручную" — сложный и трудоемкий процесс, требующий множества измерений и экспериментов для точной настройки и калибровки параметров. Поскольку влияние каждого из параметров на частотные характеристики может быть нелинейным и взаимозависимым, традиционные методы требуют значительных ресурсов. Чтобы упростить этот процесс и сократить количество экспериментов, целесообразно разработать общую математическую модель, в которую будут заложены зависимости амплитудно-фазовых частотных характеристик от перечисленных параметров.

В обобщенном виде общую математическую модель частотной характеристики представима как:

$$f_c = F(M, L, W, G, S) = \frac{1}{2\pi} \sqrt{\frac{k_{eff}(M, L, G, S)}{m_{eff}(M, W, G, S)}} \quad , \quad (3.3)$$

где:

f_c — целевая полоса пропускания или частотная характеристика преобразователя,

M — параметры материалов (например, плотность, акустическая проводимость),

L — размеры устройства,

W — масса преобразователя,

G — геометрия корпуса,

S — способ установки,

k_{eff} и m_{eff} – эффективные жесткость и масса, зависящие от указанных параметров. Каждый из параметров влияет на частотную характеристику преобразователя согласно их физическим свойствам.

3.2. Алгоритм автоматизации процесса разработки технических требований к преобразователям

Разработанная модель прогнозирования (3.3) позволяет задать зависимости между основными характеристиками преобразователя и его частотными характеристиками. Для повышения точности и автоматизации разработки технических требований предлагается обучить модель на основе искусственного интеллекта. Данные, полученные из уже проведенных экспериментов и измерений, будут использоваться для обучения модели. На основе данных ИИ сможет формировать рекомендации по оптимальным параметрам для достижения заданной полосы пропускания. Для обучения модели используются данные, полученные при проведении измерений по контролю качества соответствующих преобразователей. Таким образом, предлагается по заданному пользователем диапазону частот выводить оптимальные для преобразователя параметры, описанные (3.3). Для обучения модели необходимо подготовить данные. Для этого выполняется поиск спектральных характеристик, описанный в обработке сигналов при проведении измерений контроля качества. К этим данным прибавляются параметры измерений и преобразователей. Таким образом подготовленные данные могут быть представлены в виде таблицы в файле с расширением *.csv (Рисунок 3.1). Для решения задачи прогноза оптимальных параметров преобразователя при заданном диапазоне рабочих частот [48] наиболее эффективными являются модели градиентного бустинга. В отличие от линейных методов, такие модели способны учитывать сложные нелинейные зависимости между входными признаками и целевыми параметрами. Среди

доступных реализаций особый интерес представляет библиотека CatBoost, показывающая высокую устойчивость к разнородности данных, автоматическую обработку категориальных признаков и способность работать с ограниченными выборками.

```

transducerInputData.csv
1 Material,Geometry,Installation_Method,Mass (g),Size_X (cm),Size_Y (cm),Size_Z (cm),AFC_Max_Frequency (Hz),AFC_Bandwidth (Hz)
2 Aluminum,Cone-shaped,Bolted,33.14,3.35,2.93,2.66,932.3,126.8,-11.31
3 Steel,Cubic,Glued,51.85,8.68,8.51,1.35,250.3,191.5,-16.61
4 Titanium,Cylindrical,Bolted,30.43,8.77,4.78,1.32,340.6,67.2,-10.08
5 Aluminum,Cylindrical,Bolted,76.95,7.95,6.19,8.48,156.9,183.6,-6.74
6 Aluminum,Cone-shaped,Clamped,92.73,2.52,8.86,6.32,815.9,191.0,-10.05
7 Titanium,Cubic,Welded,48.14,8.4,8.77,4.22,659.6,68.4,-6.68
8 Aluminum,Spherical,Welded,40.79,7.92,4.34,9.05,383.4,155.8,-18.67
9 Aluminum,Cylindrical,Welded,96.54,6.37,2.88,1.63,247.4,63.0,-2.48
10 Steel,Cone-shaped,Glued,50.04,4.03,7.98,9.82,136.1,152.8,-1.22
11 Aluminum,Cubic,Bolted,54.01,3.36,4.9,6.55,250.1,163.5,-7.64
12 Polymer,Cone-shaped,Welded,41.65,2.1,2.73,2.65,508.0,64.5,-2.37
13 Titanium,Spherical,Clamped,91.86,5.59,9.14,7.11,322.5,148.9,-17.37
14 Steel,Cylindrical,Welded,61.21,6.27,2.78,7.87,333.1,166.2,-4.05
15 Steel,Cone-shaped,Welded,78.6,4.23,1.23,4.86,888.0,84.5,-17.62
16 Polymer,Cone-shaped,Bolted,83.06,5.28,1.18,8.75,563.6,50.2,-11.84
17 Titanium,Cone-shaped,Bolted,40.77,4.27,6.6,2.36,756.3,36.6,-13.52
18 Titanium,Spherical,Bolted,49.03,6.77,6.96,3.51,132.1,123.6,-8.29
19 Steel,Cone-shaped,Glued,86.36,9.11,3.8,1.84,428.6,196.1,-13.97
20 Aluminum,Spherical,Glued,42.69,2.48,3.88,2.9,531.3,168.1,-7.25
21 Steel,Spherical,Welded,81.45,4.43,4.06,3.15,267.8,98.5,-12.65
22 Titanium,Spherical,Bolted,94.69,2.27,7.12,9.26,208.2,116.6,-2.78
23 Polymer,Cylindrical,Welded,66.09,6.62,1.52,1.7,883.3,70.3,-7.32
24 Steel,Cylindrical,Clamped,76.75,7.56,9.36,4.85,945.5,49.4,-15.69
25 Titanium,Cone-shaped,Clamped,82.46,9.39,3.0,5.79,170.9,106.8,-13.28
26 Steel,Cone-shaped,Bolted,86.05,3.51,10.0,9.09,414.1,103.4,-3.05
27 Aluminum,Cylindrical,Bolted,45.55,9.55,2.54,9.98,962.5,113.9,-12.27
28 Steel,Cubic,Clamped,14.61,1.57,7.94,7.78,174.8,191.5,-9.28
29 Steel,Cylindrical,Bolted,46.17,6.43,3.01,9.74,679.3,128.7,-7.02
30 Titanium,Spherical,Glued,70.64,1.51,1.73,8.08,975.0,129.0,-13.04
31 Titanium,Cylindrical,Glued,50.53,9.04,8.87,9.67,980.4,43.0,-14.12
32 Titanium,Spherical,Bolted,77.41,5.77,8.73,7.37,524.1,199.4,-10.49
33 Steel,Cone-shaped,Clamped,42.13,7.02,4.63,1.72,246.5,159.1,-15.7
34 Aluminum,Cubic,Welded,67.13,8.48,7.32,4.62,281.9,177.9,-14.32
35 Polymer,Cone-shaped,Bolted,26.78,2.35,6.01,6.82,900.6,94.6,-18.76
36 Polymer,Spherical,Bolted,77.44,1.35,8.68,4.16,920.5,110.4,-13.92
37 Titanium,Spherical,Welded,59.11,8.91,3.84,3.46,647.1,114.0,-13.59
38 Titanium,Spherical,Bolted,28.37,3.83,5.71,6.2,789.6,54.9,-0.76
39 Steel,Cubic,Welded,36.27,2.16,5.37,9.4,765.9,144.1,-8.21
40 Polymer,Cylindrical,Clamped,77.37,5.22,4.19,1.31,695.6,182.5,-4.25
41 Titanium,Cone-shaped,Glued,53.43,6.38,9.02,7.92,844.5,49.5,-19.09
42 Aluminum,Cone-shaped,Clamped,29.44,1.28,2.49,7.5,716.9,35.5,-17.87
43 Aluminum,Cylindrical,Glued,97.05,3.47,2.35,9.68,580.6,152.9,-3.71
44 Titanium,Cone-shaped,Welded,74.36,1.3,6.19,9.38,313.3,69.2,-7.53
45 Titanium,Cubic,Glued,96.14,4.78,6.92,7.86,769.5,132.3,-16.18
46 Polymer,Cone-shaped,Clamped,48.44,9.06,2.75,5.68,526.1,145.2,-11.9
47 Polymer,Spherical,Glued,52.76,9.84,4.52,2.84,365.3,77.9,-9.23
48 Steel,Cylindrical,Welded,90.82,6.73,9.52,5.91,821.8,166.7,-3.13
49 Polymer,Cubic,Glued,57.25,2.45,3.22,7.89,727.0,119.0,-18.79
50 Aluminum,Cylindrical,Bolted,73.66,9.15,7.02,9.06,943.6,176.4,-8.07
51 Aluminum,Cubic,Clamped,32.7,5.65,7.66,8.11,248.0,157.4,5.66

```

Normal text file length: 202 290 lines: 3 181 Ln: 1 Col: 1 Pos: 1 Unix (LF) UTF-8 INS

Рисунок 3.1. – Фрагмент подготовленных данных

На рисунке 3.2 представлен структурный алгоритм функционирования модуля машинного обучения, используемого для прогнозирования параметров преобразователей. Алгоритм включает следующие этапы:

Загрузка данных. На вход модулю подается таблица в формате CSV, содержащая записи с параметрами преобразователей, включающие используемые материалы, геометрические размеры, массу колебательной

системы, конструктивные особенности и способ установки. Для каждого экземпляра приводятся измеренные амплитудно-частотные характеристики (АЧХ), полученные в лабораторных условиях или в рамках производственного контроля. Такая структура данных позволяет учитывать как конструктивные факторы, так и реальные динамические свойства преобразователей.

Предобработка данных. На данном шаге выполняется фильтрация записей с пропущенными измерениями, исправление или удаление аномальных значений, а также устранение выбросов, возникающих вследствие ошибок измерения или некорректной сборки экспериментального стенда. Числовые признаки нормализуются для обеспечения корректной работы алгоритма бустинга, а категориальные признаки преобразуются в числовое представление.

Разделение на обучающую и тестовую выборки. Для корректной оценки обобщающей способности модели исходная таблица разделяется на две подвыборки. Обучающая выборка (70–80% данных) используется непосредственно при оптимизации параметров модели, тогда как тестовая выборка (20–30%) применяется исключительно для финальной проверки качества. Такой подход предотвращает переобучение и обеспечивает объективность оценивания.

Выбор модели и гиперпараметров. В качестве основной модели используется градиентный бустинг на решающих деревьях (CatBoost), способный эффективно обрабатывать как числовые, так и категориальные признаки. На этом шаге задаются параметры глубины деревьев, скорости обучения, количества итераций и другие метрики.

Обучение модели. Производится итеративная оптимизация параметров на тренировочных данных. После каждой итерации измеряется ошибка на валидационной выборке, и модель сохраняется в виде бинарного файла *.pkl для последующего использования.

Оценка точности. Для проверки качества обученной модели выполняется тестирование на отложенной выборке. В качестве

количественных показателей используются распространённые метрики регрессии: средняя абсолютная ошибка, среднеквадратичная ошибка и коэффициент детерминации.

Прогнозирование. После финальной настройки обученная модель готова к использованию для предсказания параметров по заданной частотной характеристике (или наоборот).

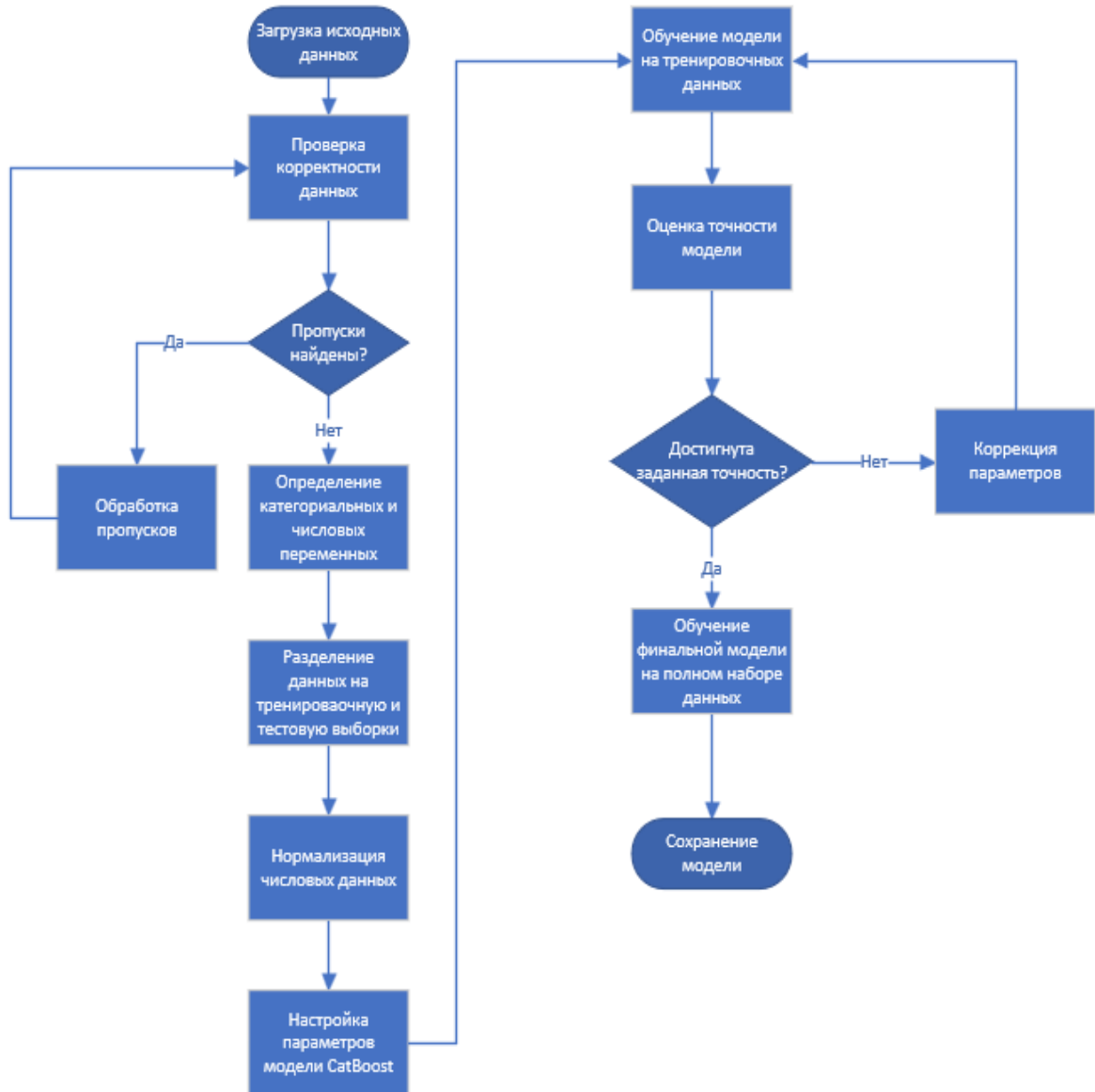


Рисунок 3.2. – Алгоритм модели машинного обучения.

Алгоритм использования обученной модели примитивна, по заданному диапазону частот с использованием внутренних механизмов алгоритма машинного обучения происходит вывод результатов прогнозирования.

3.3. Математическая модель автоматизированного процесса контроля качества многоканальных систем преобразователей

Математическая модель системы автоматизации поканальной обработки данных преобразователей (3.11) основывается на поиске АФЧХ исследуемых сигналов преобразователей с единовременным поиском участков полезных импульсов методом кросс-корреляции, для сигналов, полученных из системы коммутации, к которым применяется быстрое преобразование Фурье.

Представление сигнала $u(t)$, который подается на выход канала демультиплексора с тремя управляющими входами по управляющему воздействию $\overline{u_{\text{ВХ}}} = \overline{A_0 A_1 A_2}$ описывается как:

$$\phi(A_0, A_1, A_2,) = \begin{cases} u(t), & \overline{u_i} \cap \overline{u_{\text{ВХ}}} = 1 \\ 0, & \overline{u_i} \cap \overline{u_{\text{ВХ}}} = 0 \end{cases} \quad (3.4)$$

где $u(t)$ – входной сигнала, который передается на канал преобразователя, $\overline{u_i}$ - комбинация логического воздействия i -го управляющего выхода

Функция (3.4) описывает, что при подаче различных битовых комбинаций A_0, A_1, A_2 , при подаче сигнала переключения активируется только один выход. Демультиплексор питания, соединяется с демультиплексором сигнала, и функционирует по принципу (3.4), где в качестве $u(t)$ используется напряжение, необходимое для питания активной платы.

При расширении данного закона на каскадную схему можно обеспечить управление 8^n каналами преобразователей, где n – количество уровней каскада, например при $n = 3$, схема позволяет управлять 512 каналами преобразователей. Для одного уровня каскада при $\overline{u_{\text{ВХ},i}} = \overline{A_{i,0} A_{i,1} A_{i,2}}$ описывается как:

$$\phi_i(A_i) = \begin{cases} u(t), & \overline{u_{i,j}} \cap \overline{u_{\text{ВХ},i}} = 1 \\ 0, & \overline{u_{i,j}} \cap \overline{u_{\text{ВХ},i}} = 0 \end{cases} \quad (3.5)$$

где $\overline{u_{i,j}}$ – битовая комбинация канала j уровня i , т.е. набор $[B_{i,j,0}, B_{i,j,1}, B_{i,j,2}]$.

Тогда уравнение для каскада уровней по управляющим воздействиям $\overline{u_{\text{ВХ},0}}, \overline{u_{\text{ВХ},1}}, \dots, \overline{u_{\text{ВХ},n-1}}$ ВЫГЛЯДИТ как:

$$\Phi_c(A_i) = \prod_{i=0}^n \begin{cases} u(t), \overline{u_{i,j}} \cap \overline{u_{\text{ВХ},i}} = 1 \\ 0, \overline{u_{i,j}} \cap \overline{u_{\text{ВХ},i}} = 0 \end{cases}, \quad (3.6)$$

где n – уровень каскада, а Φ_c - ненулевой только если все уровни каскада активируют свой единственный канал.

При моделировании подобной каскадной схемы, нужно учитывать, что для коммутации питания на уровень каскада, необходимо параллельно к предпоследним уровням каскада добавить дубль демультиплексора управляющий коммутацией питания, т.е. для обслуживания 64 каналов достаточно добавить еще один демультиплексор к управляющему первому в каскаде, в случае обработки 512 каналов преобразователей, возникает необходимость в использовании дополнительных 9 демультиплексоров для коммутации питания – 1 демультиплексор на первом каскаде и 8 на втором каскаде. При разработке математической модели и определении количества используемых демультиплексоров, стоит учитывать физическую реализацию, при которой могут возникнуть сложности с недостаточной мощностью сигнала переключения (EN), в таком случае может потребоваться использование дополнительного демультиплексора, коммутирующего усиленный сигнал на каскады. Таким образом для управления 8^n каналами можно посчитать количество демультиплексоров по закону:

$$D_{CNT} = n + 8^n + \sum_{k=0}^n 8^k, \quad (3.7),$$

где n – номер уровня каскада, $n \in \mathbb{Z}_+$.

Количество цифровых выходов измерительного модуля, необходимых для управления n каскадами можно вычислить как:

$$DO_{CNT} = 3(n + 1) \quad (3.8).$$

Таким образом, аппаратного обеспечения выбранного измерительного модуля [44] достаточно для решения задачи коммутации и реализации описанной

модели коммутации каналов. При физическом моделировании системы коммутации, необходимо учитывать ограничения в количестве используемых демультиплексоров и их габариты.

Для управления каналом системы необходимо активировать цифровые выходы измерительного модуля. Может потребоваться возможность по известным логическим значениям цифровых выходов найти десятичное число, которое будет подаваться непосредственно на измерительный модуль. Данное число можно найти с помощью:

$$D = \sum_{i=0}^{n-1} B_i 2^{n-i-1}, \quad (3.9)$$

где D - искомое десятичное число, B_i – значение i -го канала цифрового выхода измерительного модуля, n – количество цифровых выходов.

Например, передается входной массив $B = [0,0,1,0,0,0,1,1]$, соответствующий одному каналу системы коммутации. Рассчитывая по (3.9): $D = 0 * 2^7 + 0 * 2^6 + 1 * 2^5 + 0 * 2^4 + 0 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0 = 35$.

Может потребоваться активация выбранного канала обратным способом. Т.е. может потребоваться при вводе десятичного номера канала получить набор логических значений для каждого канала цифрового выхода измерительного модуля. Для этой задачи можно воспользоваться:

$$B = \bigcup_{i=0}^{n-1} B_i, \text{ где } B_i = \langle int \rangle \left(\left[\frac{D}{2^{n-i-1}} \right] \bmod 2 = 1 \right), \quad (3.10)$$

где \cup – оператор объединения значений в массив, B -список каналов цифровых выходов измерительного модуля, количества n , которое можно представить как объединение $B(i)$, где каждое i -ое значение может принимать значение 1 или 0. Значение 1 принимается, если выполняется условие $\left(\left[\frac{D}{2^{n-i-1}} \right] \bmod 2 = 1 \right)$, зависящее от введенного десятичного числа D , $0 \leq D \leq 2^n - 1$.

Например, при $D = 35: B_0 = \left[\frac{35}{27} \right] \bmod 2 = \left[\frac{35}{128} \right] \bmod 2 = 0$, $B_1 = \left[\frac{35}{26} \right] \bmod 2 = \left[\frac{35}{64} \right] \bmod 2 = 0$, $B_2 = \left[\frac{35}{25} \right] \bmod 2 = \left[\frac{35}{32} \right] \bmod 2 = 1$, $B_3 = \left[\frac{35}{24} \right] \bmod 2 = \left[\frac{35}{16} \right] \bmod 2 = 0$, $B_4 = \left[\frac{35}{23} \right] \bmod 2 = \left[\frac{35}{8} \right] \bmod 2 = 0$, $B_5 = \left[\frac{35}{22} \right] \bmod 2 = \left[\frac{35}{4} \right] \bmod 2 = 0$, $B_6 = \left[\frac{35}{21} \right] \bmod 2 = \left[\frac{35}{2} \right] \bmod 2 = 1$, $B_7 = \left[\frac{35}{20} \right] \bmod 2 = \left[\frac{35}{1} \right] \bmod 2 = 1$.

Таким образом $B = [0, 0, 1, 0, 0, 0, 1, 1]$, что соответствует проверочному массиву.

Следующей важной задачей является хранение данных перед записью результата. Т.к. предполагается использование модели для измерений многоканальных систем на различных параметрах измерений, то ожидается большое количество данных. АЦП измерительного модуля на выходе выдает два массива оцифрованных данных, представленных в виде вещественных чисел. Для регистрации данных на носитель, оптимальным является использование преобразованных в двоичный формат данных после слияния, например $x = [a_0, c_0, \dots, a_n, c_n]$, где $a_i \in A, c_i \in C$ – вещественные числа исходных массивов.

Создание массива бинарных чисел описывается как:

$$B = \bigcup_{j=0}^{N-1} B_j, \quad (3.11),$$

$$B_j = \begin{cases} b_{i,k}^{(I)}, & 0 \leq j < \sum_{r=0}^i m_r, \quad k = j - \sum_{r=0}^{i-1} m_r \\ f_{i,k}, & \sum_{r=0}^i m_r \leq j < \sum_{r=0}^i (m_r + n_r), \quad k = j - \sum_{r=0}^{i-1} m_r \end{cases}, \quad (3.12)$$

где $x_i = I_i + F_i$, $I_i = [x_i]$, $F_i = x_i - I_i$, $b_{i,k}^{(I)} = \left[\frac{I_i}{2^k} \right] \bmod 2, k = 0, 1, \dots, m_i - 1$, $f_{i,0} = [2F_i]$, $F_{i,1} = 2F_i - f_{i,0}, \dots, f_{i,k} = [2F_{i,k-1}]$, $F_{i,k} = 2F_{i,k-1} - f_{i,k}$.

Полное двоичное число обозначается как:

$$x_i^2 = b_{i,m_i-1}^{(I)} \dots b_{i,0}^{(I)} \cdot f_{i,0} f_{i,1} \dots f_{i,n_i-1}, \quad (3.13)$$

Для обратного перевода используется:

$$X = \bigcup_{i=0}^{N-1} X_i, \quad X_i = I_i + F_i, \quad (3.14)$$

$$I_i = \sum_{k=0}^{m_i-1} b_{i,k}^{(I)} \cdot 2^k, \quad (3.15)$$

$$F_i = \sum_{k=0}^{n_i-1} f_{i,k} \cdot 2^{-(k+1)}, \quad (3.16)$$

где X_i вещественное число, полученное суммой восстановленных целой I_i и дробной F_i частей по заданному бинарному числу. Например, при $x = 24.335$.

$$I = [24.335] = 24, F = 24.335 - 24 = 0.335. \quad \text{Используя} \quad b_k^{(I)} =$$

$$\left[\frac{I}{2^k} \right] \bmod 2: b_0 = \left[\frac{24}{2^0} \right] \bmod 2 = 24 \bmod 2 = 0, b_1 = \left[\frac{24}{2^1} \right] \bmod 2 =$$

$$[12] \bmod 2 = 0, b_2 = \left[\frac{24}{2^2} \right] \bmod 2 = [6] \bmod 2 = 0, b_3 = \left[\frac{24}{2^3} \right] \bmod 2 =$$

$$[3] \bmod 2 = 1, b_4 = \left[\frac{24}{2^4} \right] \bmod 2 = [1.5] \bmod 2 = 1.$$

$$I = 24_{10} = 11000_2.$$

В примере для точности берется 8 бит. Для снижения ошибки вычислений следует использовать большее количество бит, например 32.

$$F_0 = 0.335, 2F_0 = 0.67 \Rightarrow f_0 = 0, F_1 = 0.67, 2F_1 = 1.34 \Rightarrow f_1 = 1, F_2 =$$

$$0.34, 2F_2 = 0.67 \Rightarrow f_2 = 0, F_3 = 0.68, 2F_3 = 0.67 \Rightarrow f_3 = 1, F_4 =$$

$$0.36, 2F_4 = 0.67 \Rightarrow f_4 = 0, F_5 = 0.72, 2F_5 = 0.67 \Rightarrow f_5 = 1, F_6 = 0.44,$$

$$2F_6 = 0.67 \Rightarrow f_6 = 0, F_7 = 0.88, 2F_7 = 0.67 \Rightarrow f_7 = 1, F_8 = 0.76.$$

$$F \approx 0.01010101_2, x = 24.335 \approx 11000.01010101_2.$$

$$\text{Обратное преобразование: } I = 0 \cdot 2^0 + 0 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 + 1 \cdot 2^4 = 8 + 16 = 24,$$

$$F^{(8)} = 2^{-2} + 2^{-4} + 2^{-6} + 2^{-8} = 0.25 + 0.0625 + 0.015625 + 0.00390625 = 0.33203125. \text{ Приближенное значение будет } x^{(8)} = 24 +$$

$$0.33203125 = 24.33203125. \text{ Ошибка по сравнению с исходным числом составляет: } \Delta^{(8)} = x^{(8)} - 24.335 = -0.00296875. \text{ Если для точности взять}$$

$$\text{первые 32 бита дробной части, то дробная часть: } \text{frac_bits}_{0..31} = 01010101110000101000111101011100. \text{ При использовании обратного}$$

преобразования: $F^{(32)} \approx 0.33499999996274709701538085937500$. Тогда $x^{(32)} = 24 + F^{(32)} \approx 24.33499999996274709701538085937500$.

Погрешность составит:

$$\Delta^{(32)} = x^{(32)} - 24.335 \approx -3.7252902984619140625 \times 10^{-11}.$$

При обработке сигнала важнейшей задачей становится поиск начала полезного сигнала для дальнейшего поиска АФЧХ. По возможности рекомендуется и заложено в модели сделать синхронизацию сигналов, т.е. необходимо получить данные, где все полезные импульсы на всех частотах для всех каналов будут располагаться на одних и тех же позициях. Данное требование возможно реализовать при помощи синхронизации АЦП и ЦАП цифрового модуля путем межмодульной синхронизации или по спаду или фронту сигнала. Таким образом для обработки сигналов потребуется один раз найти позицию полезного импульса и в дальнейшем использовать эту информацию. Т.к. полезные сигналы, как правило, состоят из сравнительно небольшого количества данных, то дальнейшая обработка будет происходить быстро. Для поиска позиции полезного импульса можно воспользоваться методом кросс-корреляции. Зная длительность сигнала $x(t)$, частоту и частоту дискретизации, можно создать сигналу $y(t)$ аналогичной формы, с длительностью, например, в двое меньше генерируемого и той же частоты.

Тогда кросс-корреляции ищется как:

$$C(t) = \sum_{\tau=0}^T [x(\tau) * y(t - \tau)], \quad (3.17)$$

где $C(t)$ - результат кросс-корреляции, τ – сдвиг времени, который может изменять от 0 до T – длительности периодического сигнала. Вычисляя значения кросс-корреляции для различных t , максимальное значение будет указывать на место начала полезного сигнала. Кросс-корреляцию необходимо выполнить как для опорного, так и для измеряемого сигнала. Вычисляя значения кросс-корреляции для различных t , максимальное значение будет указывать на позицию по времени начала полезного сигнала. Обозначим результат поиска полезного сигнал, как:

$$S(t) = S(C(t)) \quad (3.18)$$

Для поиска АФЧХ необходимо выполнить быстрое преобразование Фурье для полезных импульсов опорного и измеряемого сигналов по закону:

$$S[k] = \sum_{n=0}^{N-1} \left(S[n] e^{-\frac{j2\pi kn}{N}} \right), \quad k = 0, \dots, N-1 \quad (3.19)$$

где $s[n]$ – дискретный во времени сигнал $\prod_{i=0}^{n-1} \phi_i(A_i, EN_i, u_x(n))$, $n = 0, \dots, N-1$, N – количество точек быстрого преобразования Фурье, x – *input* – для входного сигнала или *output* – для выходного. При необходимости добавляются нули для достижения требуемого N . f_s – частота дискретизации, $T_s = \frac{1}{f_s}$ – период дискретизации. Индекс частоты $k = 0, \dots, N-1$ соответствуют частоте $f_k = \frac{k}{N} f_s$, j – мнимая единица.

Тогда поиск АФЧХ осуществляется по закону:

$$H[k] = \frac{S_{out}[k]}{S_{input}[k]}, \quad k = 0, \dots, N-1, \quad (3.20)$$

где $S_{out}[k]$, $S_{input}[k]$ – результаты быстрого преобразования Фурье выходного и входного дискретных импульсов соответственно. Тогда математическая модель системы автоматизации поканальной обработки многоканальных систем преобразователей описывается как:

$$H[k] = \frac{\sum_{n=0}^{N-1} \left(\prod_{i=0}^{n-1} \phi_i(A_i, EN_i, u_{output}(n)) \right) e^{-j2\pi kn/N}}{\sum_{n=0}^{N-1} \left(\prod_{i=0}^{n-1} \phi_i(A_i, EN_i, u_{input}(n)) \right) e^{-j2\pi kn/N}} \quad (3.21)$$

3.4. Алгоритм системы автоматизации процесса контроля качества многоканальных систем преобразователей

Для разработки программного обеспечения, необходимо разработать наиболее общий алгоритм, который позволяет решать задачи, приведенные на рисунке 3.3. Перечислим основные требования к разрабатываемой программе.

1. Возможность проводить измерение характеристик приема и излучения каналов многоканальных систем.
2. Возможность в автоматизированном режиме обрабатывать поочередно как все каналы, так и один указанный канал.
3. Возможность задавать диапазон измеряемых частот и возможность проводить измерения на одной выбранной частоте.
4. Возможность графического отображения процесса измерения и обрабатываемых данных.
5. Возможность регистрации результатов прямых измерений, обработанных сигналов и обработанных рассчитанных значений.
6. Возможность выбора типа сигнала и конфигурирование.
7. Возможность настройки характера полезных импульсов.

Разрабатываемый алгоритм [49] должен учитывать вышеперечисленные требования. В рабочий алгоритм (Рисунок 3.3.) поступают данные из графической формы, определяющие параметры измерений. Входными данными могут являться: номера измеряемых каналов, тип и характеристики сигналов (амплитуда, диапазон частот, шаг), параметры полезных импульсов (длительность общего сигнала, количество полезных импульсов, длина полезного импульса и др.) и другие параметры, для реализации указанных требований.

Вторым шагом алгоритма является настройка измерительных модулей, задается частота дискретизации, происходит настройка цифровых выходов, ЦАП и АЦП.

Запускается цикл по диапазону каналов. Диапазон каналов может иметь различную длину, как по всем указанным каналам, так и по одному выбранному. Происходит активация выбранного канала. Подается специальный код на цифровой выход измерительного модуля, который по схеме поступает в систему коммутации. В момент активации определенного канала система должна быть готова либо считывать сигнал с этого канала, либо подавать сгенерированный сигнал.

Запускается цикл по диапазону заданных частот с определенным шагом. На основе текущей частоты и введенных параметров, происходит генерация опорного сигнала. Если происходит режим измерений характеристик приема преобразователей, то сигнал подается на излучающую аппаратуру. Выбранный канал переводится в режим регистрации и данные сигнала поступают на АЦП измерительного модуля. При измерении характеристик излучения преобразователей, сгенерированный сигнал подается на указанный канал. Вместе с этим активируется канал АЦП для принимающего устройства.

Далее происходит регистрация опорного и выходного сигналов в буфер. Принятые сигналы подаются на отображение в графическую форму. Параллельно запускаются процессы регистрации и обработки сигналов при необходимости. Выполняется аналогичный цикл по всему диапазону частот и по всем выбранным каналам.

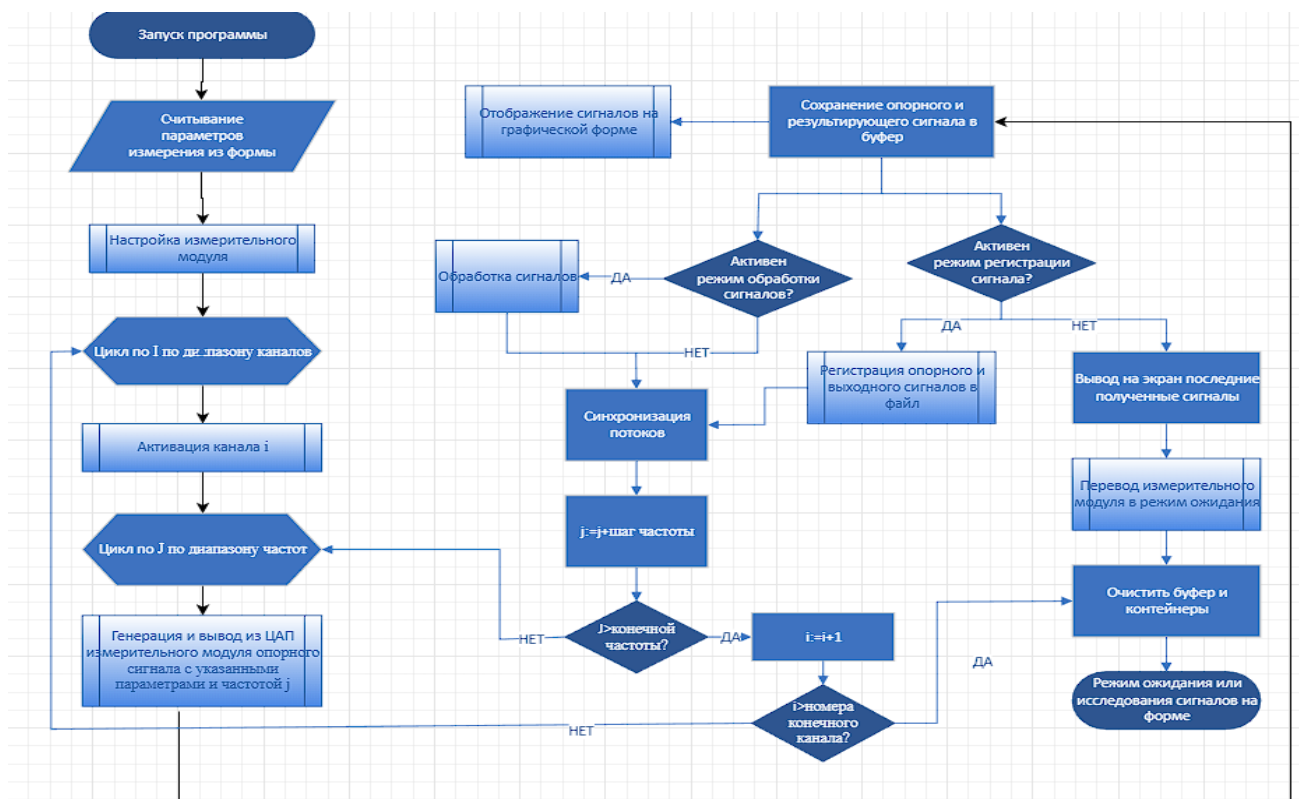


Рисунок 3.3. – Общий алгоритм автоматизации

Рассмотрим подробнее процесс регистрации результатов прямых измерений (Рисунок 3.4).

На вход алгоритму поступают данные прямых измерений. Происходит выбор рабочей директории. Рабочую директорию можно создавать различными способами. Например, можно создать директорию с датой и режимом измерений и для каждого измеряемого канала создать подпапку.

Создается двоичный массив, куда будут записываться результаты измерений. Создается структура шапки файла, куда можно занести основные параметры измерений. Важно, чтобы шапка имела определенный и фиксированный размер, который можно записать в первый байт. В цикле в результирующий массив записываются данные измерений. Можно в четные индексы записать данные опорного сигнала, в нечетные – выходного. Затем создается директория с номером канала, если такой не существует для данного режима измерений. На основе параметров сигнала и шапки, формируется имя файла. Пользователю предлагается поменять шаблон названия на свой. Это делается один раз для унификации названий. Создается пустой файл в режиме бинарной записи. Записывается шапка и результирующий массивы. Происходит закрытие файла и выход из процедуры регистрации.

Обработка результатов прямых измерений (Рисунок 3.5) запускается при соответствующем требовании. Данную процедуру можно запускать как в процессе измерений, так и по завершению, на основе зарегистрированных результатов (Рисунок 3.4). Процедура обработки сигналов [19], [21-27] является сложной задачей, особенно в режиме измерений в реальном времени, т.к. количество поступающих данных может измеряться сотнями тысяч точек. В рамках разрабатываемого алгоритма, данная задача решается с использованием опции синхронизации цифровых модулей. Идея заключается в том, что в любом из режимов измерений момент начала регистрации опорного и выходного сигналов в буфер происходит синхронно, а это означает что для любого канала, на любой измеряемой частоте, полезные сигналы будут располагаться на одних и тех же позициях относительно точек всего

зарегистрированного сигнала. На основе данной идеи будет построен основной алгоритм обработки.

После поступления сигналов в функцию обработки происходит конфигурирование параметров обработки. Данное действие происходит единожды. Производится настройка степени точности выполнения быстрого преобразования Фурье, устанавливается число потоков обработки и другие данные. Затем поступивший сигнал добавляется в конец очереди на обработку. Данное действие служит подстраховкой для корректного выполнения операции. Затем происходит проверка на наличие данных о положении полезного сигнала.

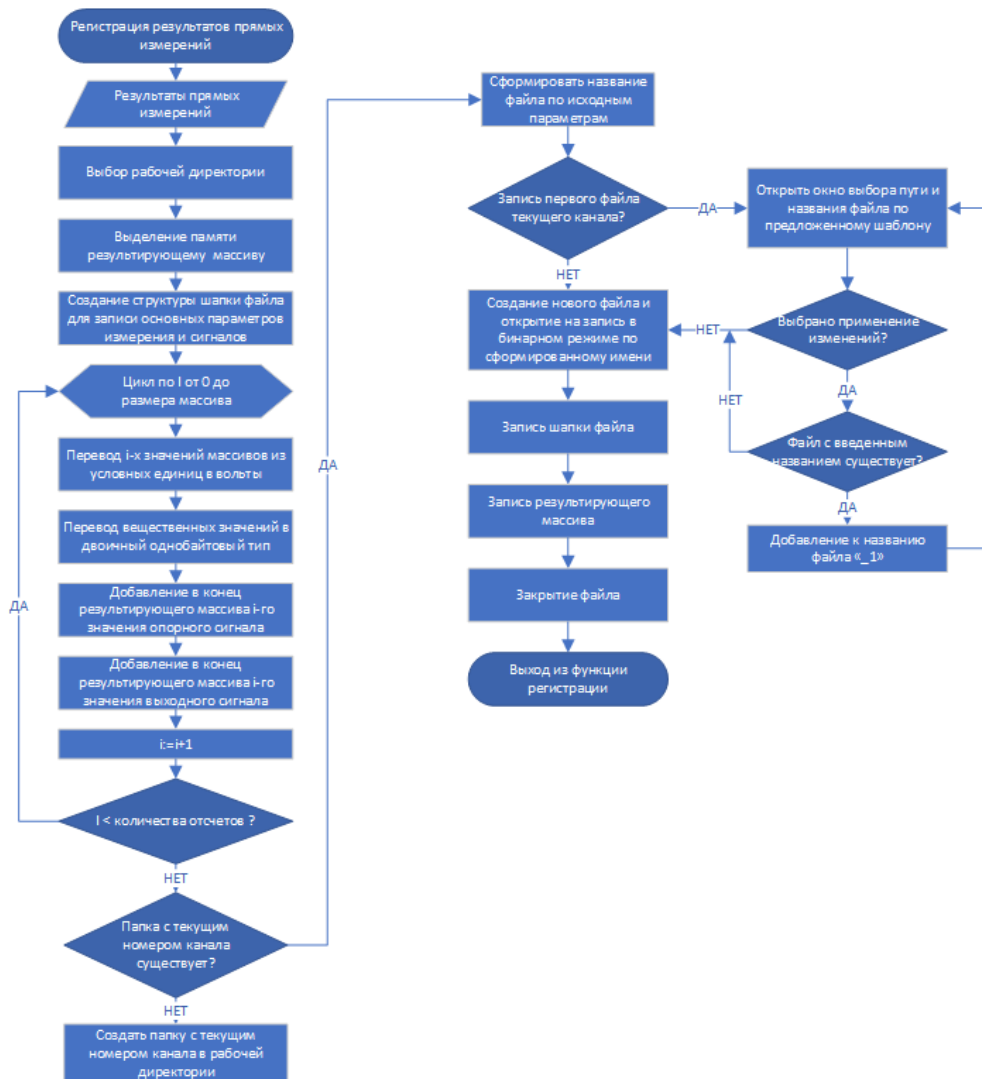


Рисунок 3.4. – Алгоритм регистрации результатов прямых измерений.

Положение полезного сигнала можно определять различными способами. Наиболее простым способом является единоразовое выделение строба полезного сигнала на графической форме оператором. Если данных о положении полезного сигнала нет, то запускается процедура поиска (Рисунок 3.6).

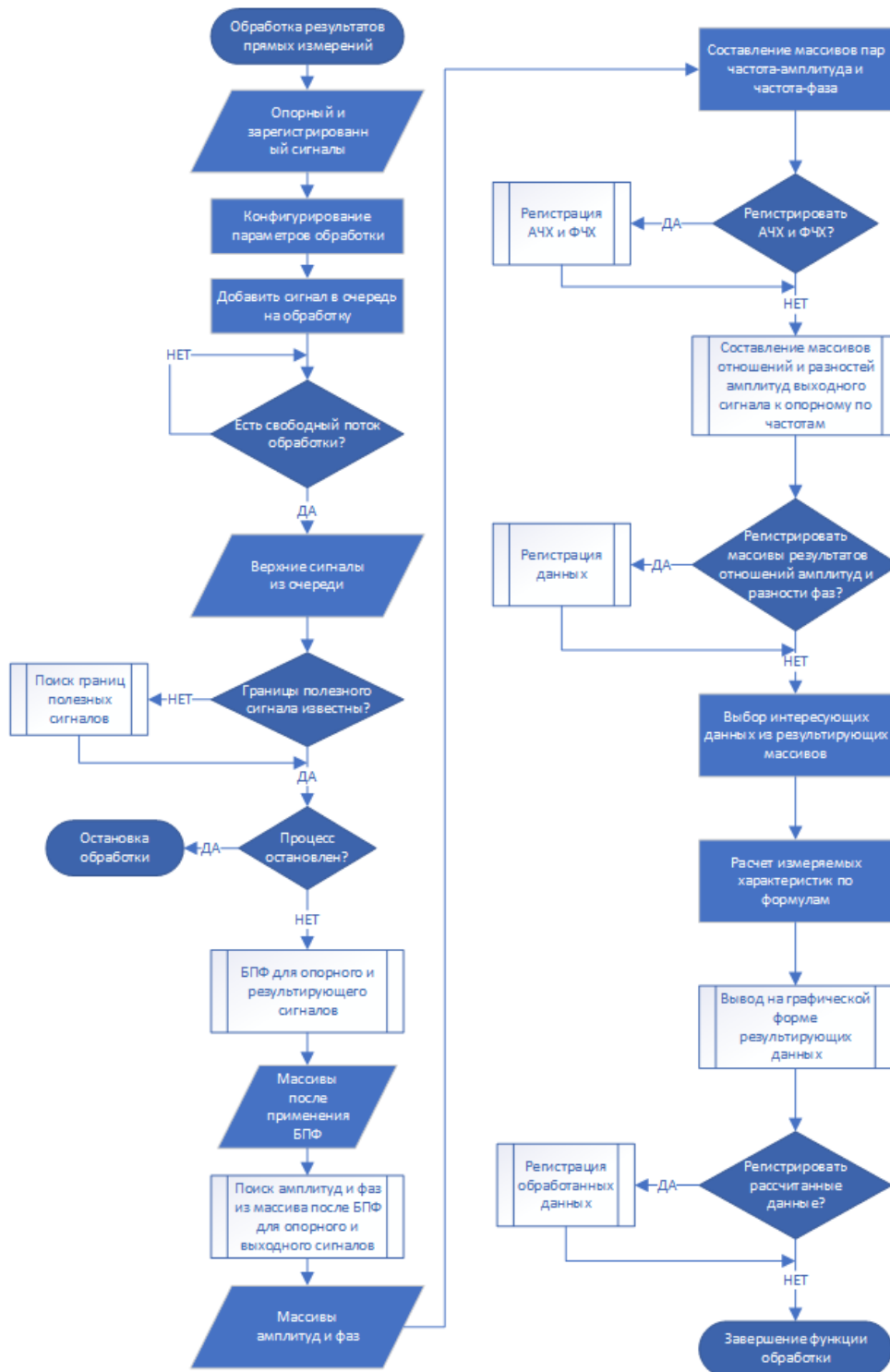


Рисунок 3.5. – Алгоритм обработки сигналов.

В данной процедуре создается шаблонный сигнал по указанным характеристикам определенной длины. Далее выделяются стробы из полученного сигнала такой же длины и происходит наложение шаблонного сигнала на измеренный. Происходит расчет корреляции данных сигналов и в позиции наибольшей корреляции будет находиться полезный сигнал. Данную операцию нужно проделать для опорного и выходного сигналов. Т.к. известно, что все полезные сигналы будут находиться на одинаковых позициях, то достаточно вызвать данную функцию поиска единойжды. По известным границам полезных сигналов, выбирается сигнал оптимальной длительности. Выполняется БПФ над выбранными сигналами. Происходит поиск АЧХ и ФЧХ сигналов. Опционально происходит регистрация обработанных данных на различных этапах процедуры. На основе полученных АЧХ и ФЧХ происходит расчет измеряемых характеристик, например чувствительностей, резонансов, импедансов и других параметров.

Наиболее сложными задачами в обработке сигналов является процесс поиска полезных сигналов. Основным механизмом обработки является выполнение быстрого преобразования Фурье. Сложность алгоритма составляет $O(N \log N)$, где N - размер сигнала. В рассмотренных методах [20-27] обработки БПФ применяется ко всему сигналу, либо происходит поиск полезного сигнала для каждой частоты. Поиск границ полезного сигнала имеет сложность $O(N)$. Разработанный алгоритм позволяет выполнить БПФ над оптимальным стробом, а не над всем массивом данных, а также поиск полезных сигналов можно выполнить на графической форме или вызвать функцию поиска единойжды. Можно выделить несколько преимуществ данного алгоритма:

1. Выполняются все поставленные для измерений задачи.
2. Выполнение БПФ над массивами с размером на порядок ниже входных данных.

3. Функцию поиска полезных сигналов нужно выполнять единожды или не выполнять с добавлением опции выбора полезных сигналов на графический интерфейс пользователем.

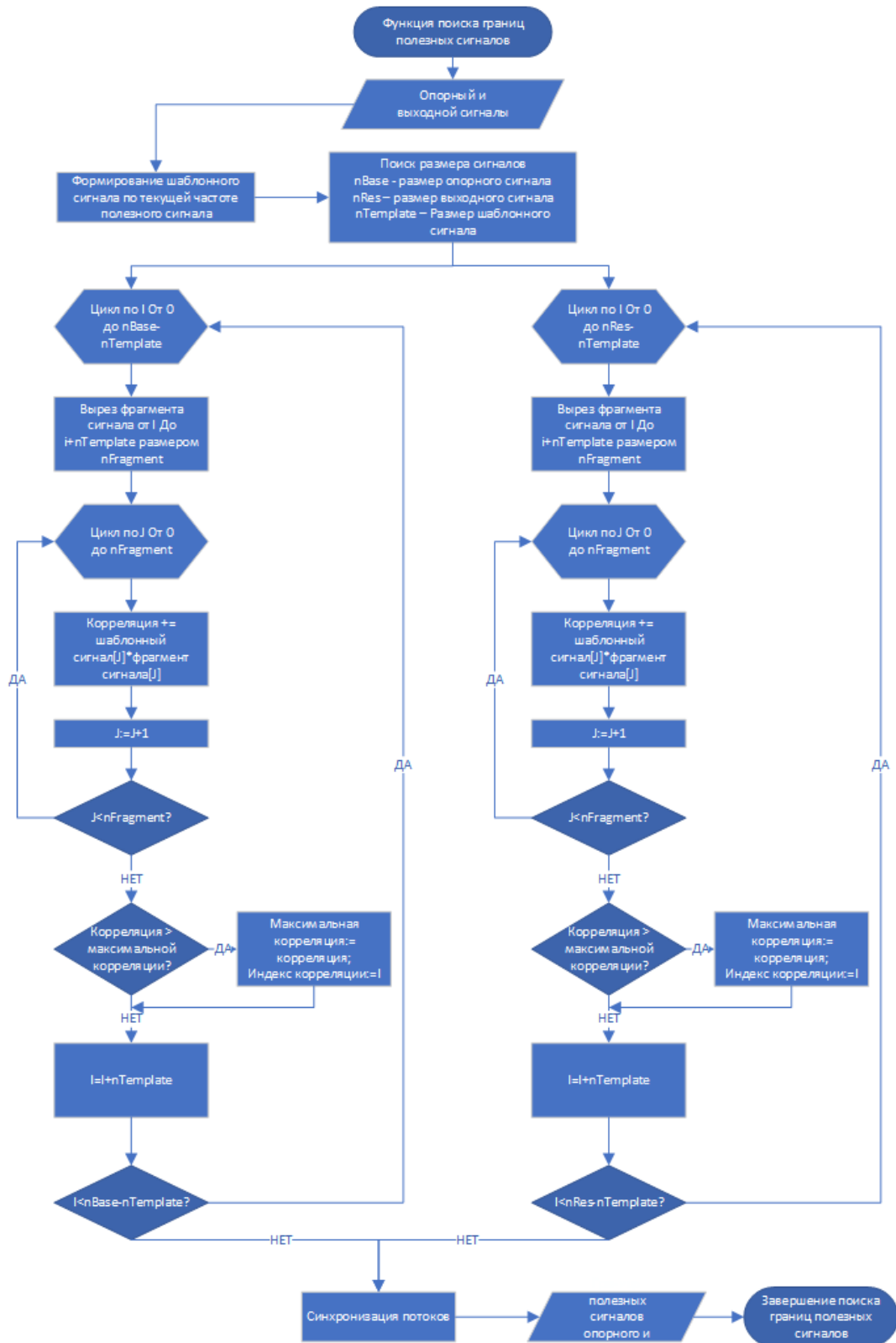


Рисунок 3.6. – Поиск границ полезных сигналов.

3.5. Выводы по Главе 3

1. Основными параметрами автоматизации процесса разработки технических требований к преобразователям являются: параметры материалов, размеры устройства, масса преобразователя, геометрия корпуса и способ установки.

2. Для автоматизации процесса разработки технических требований к преобразователям выполняется обучение модели на результатах контроля качества многоканальных систем преобразователей. В качестве модели машинного обучения используется реализация градиентного бустинга.

3. Алгоритм автоматизации разработки технических требований к преобразователям базируется на соответствующем алгоритме для обучения модели машинного обучения и отображает загрузку и предобработку данных, разделение на обучающую и тестовую выборки, выбор и обучение модели и оценку точности.

4. Математическая модель автоматизированного процесса контроля качества основана на поиске временных границ полезных импульсов методом кросс-корреляции применяемой на сигналах из системы коммутации, с последующим поиском АФЧХ с использованием быстрого преобразования Фурье.

5. Алгоритм автоматизированного процесса контроля качества отображает процессы конфигурирования измерений, коммутации канала, изменения частоты сигнала, единовременный поиск временных границ полезного сигнала, возможность регистрации и обработки, а также синхронизацию работы измерительных модулей для оптимизации обработки результатов измерений.

**ГЛАВА 4. РАЗРАБОТКА ПРОГРАММНО-АППАРАТНОГО
КОМПЛЕКСА АВТОМАТИЗИРОВАННОГО ПРОЦЕССА КОНТРОЛЯ
КАЧЕСТВА МНОГОКАНАЛЬНЫХ СИСТЕМ ПРЕОБРАЗОВАТЕЛЕЙ
И ПРОГРАММНОГО МОДУЛЯ АВТОМАТИЗАЦИИ ПРОЦЕССА
РАЗРАБОТКИ ТЕХНИЧЕСКИХ ТРЕБОВАНИЙ К
ПРЕОБРАЗОВАТЕЛЯМ**

**4.1. Программно-аппаратный комплекс автоматизированного процесса
контроля качества многоканальных систем преобразователей**

Апробация результатов исследования проводится на многоканальной системе гидрофонов, т.к. данная система в полной мере отображает основные свойства многоканальных систем преобразователей.

Для реализации схемы предлагается использовать управляемые восьмиканальные релейные модули (4.1).

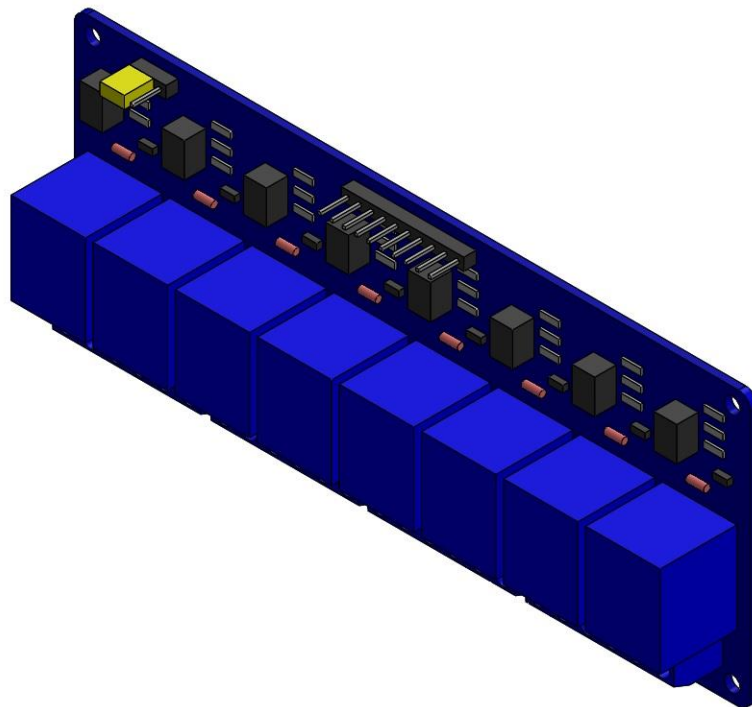


Рисунок 4.1. – Восьмиканальный релейный модуль (модель из САПР).

Собрать схему демультиплексора 3 на 8 и собрать в схему, как показано на рисунке 4.2.

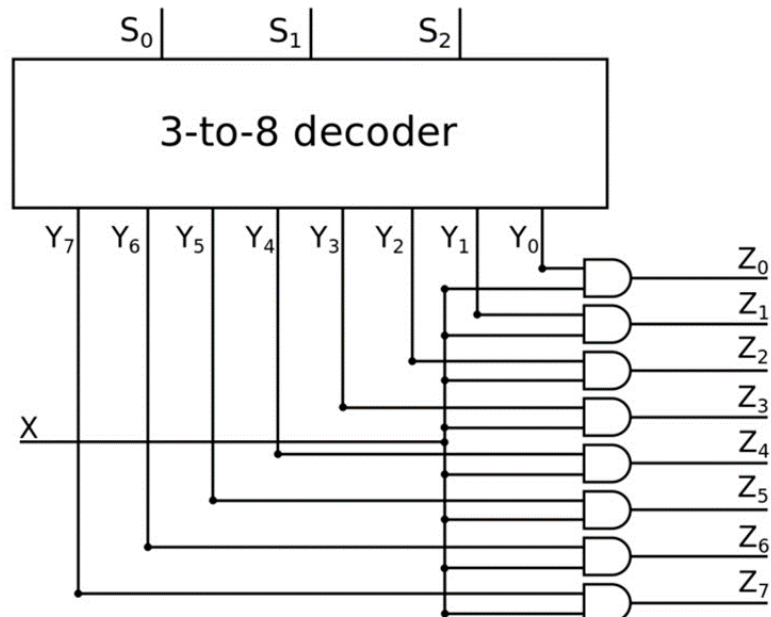


Рисунок 4.2. – Схема демультиплексора 3 на 8.

Для решения проблемы питания переключаемой платы, параллельно плате управления переключением каналов (Рисунок 4.3.) вводится аналогичная по схеме плата, только вместо сигнала подается постоянный ток для питания. Таким образом будет включаться только выбранная плата, куда будет поступать сигнал, минимизируя затраты.

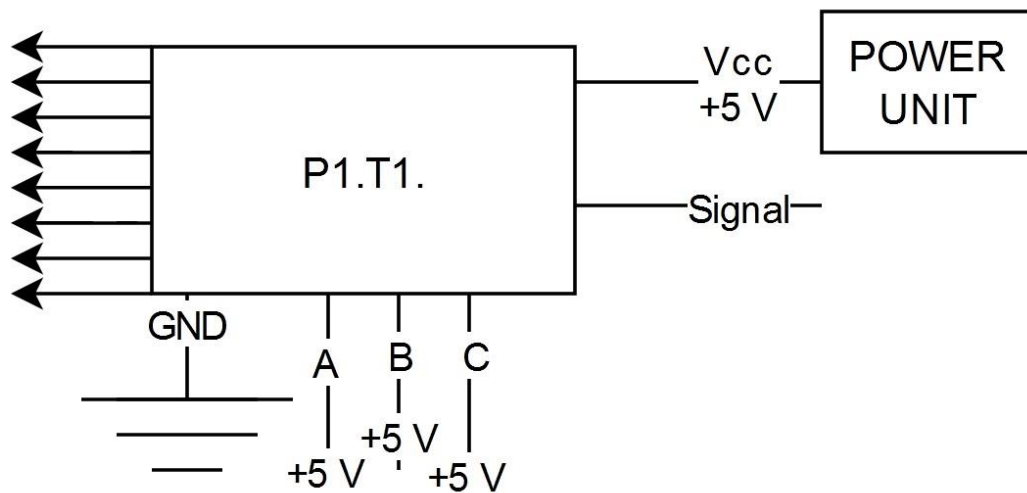


Рисунок 4.3. – Схема релейного модуля переключения каналов.

Из-за большого количества плат, соединяемых параллельно, может возникнуть проблема недостатка мощности при переключении, т.к. проходящий электрический ток встречает сопротивление при прохождении

через каскад. Для решения этой проблемы вводится специальная управляющая плата, на который подается постоянный ток из источника питания (Рисунок 4.4).

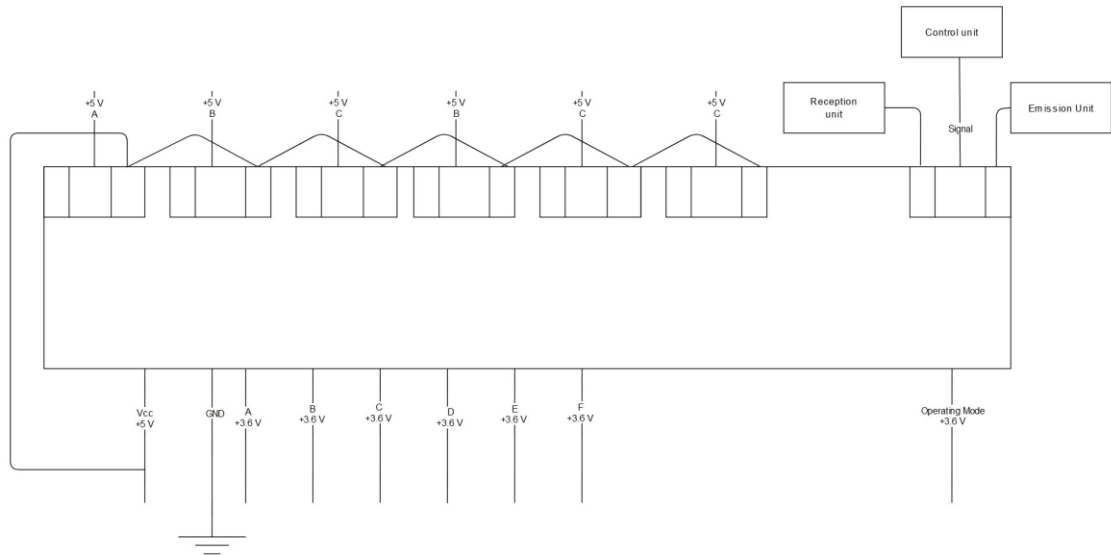


Рисунок 4.4. – Схема управляющей платы.

К данному релейному модулю поступает управляющее воздействие от цифрового выхода управляющей платы. На коммутируемые выходы релейной платы подаётся постоянный ток, обеспечивающий питание исполнительных цепей. В процессе переключения каналов управляющей платы на вход релейного каскада поступает не непосредственно сигнал цифрового выхода, а постоянный ток от внешнего источника питания. Его величины достаточно для надежного управления всеми релейными элементами каскада, что позволяет минимизировать нагрузку на цифровые линии управления и повысить общую отказоустойчивость системы.

Мощности воздействия цифрового выхода достаточно для управления сразу несколькими релейными платами, что обеспечивает масштабируемость и возможность расширения числа каналов без существенного увеличения аппаратных затрат. Дополнительно один из выходов релейного модуля используется для формирования управляющего сигнала переключения режима измерения. Через цифровой выход измерительного модуля управляющая программа подаёт соответствующее воздействие, инициируя

переключение между режимом подачи тестового сигнала на систему коммутации и режимом приёма сигнала от контролируемого преобразователя. Это обеспечивает автоматизацию процедур измерений и исключает необходимость ручного вмешательства оператора при смене режимов работы.

Схема коммутации, разработанная для управления 64 каналами, представлена на Рисунке 4.5. Реализованная структура обеспечивает параллельное и последовательное объединение релейных модулей, что позволяет гибко конфигурировать коммутационные пути и гарантирует стабильную работу системы при увеличении количества подключаемых преобразователей.

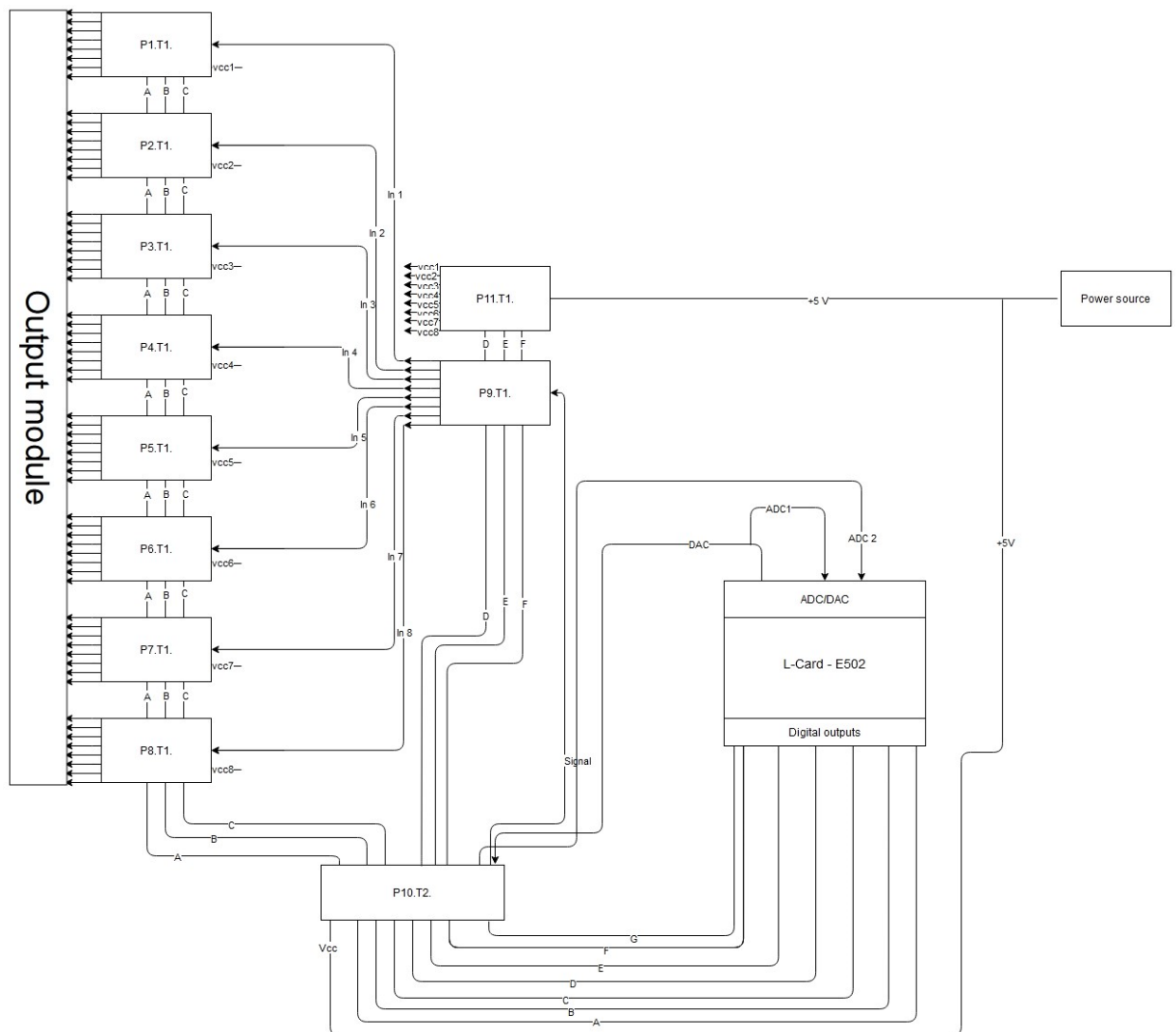


Рисунок 4.5 – Схема коммутации многоканальной системы

В общем случае разработанная модель автоматизации [53] позволяет проводить измерения в различных физических средах, включая подводные условия, зоны повышенной радиации и области с экстремальными температурными режимами. В таких условиях существенно усложняется задача передачи измерительной информации на большие расстояния. Поскольку принимаемые сигналы имеют сравнительно малую амплитуду (порядка микровольт), их передача по проводнику протяжённостью в десятки метров неизбежно сопровождается потерями, искажениями и воздействием внешних электромагнитных шумов. Даже использование измерительных усилителей лишь частично компенсирует данные эффекты, поскольку вносимые усилением улучшения нивелируются шумами линии, паразитными сопротивлениями и ёмкостями.

Для минимизации потерь применяются встроенные усилители, обеспечивающие первичное повышение уровня сигнала. Однако даже в усиленном виде аналоговый сигнал, передаваемый по длинному проводнику, остаётся чувствительным к характеристикам среды и параметрам линии связи. В расчётах необходимо учитывать длину кабеля, его материал, площадь поперечного сечения, паразитную ёмкость, индуктивность, а также влияние внешнего электромагнитного поля. Указанные параметры могут существенно изменяться в реальных условиях эксплуатации, что приводит к снижению точности восстановления истинного значения сигнала и усложняет калибровку системы.

В подобных условиях достижение требуемой точности измерений становится затруднительным при использовании традиционных схем удалённого съёма аналогового сигнала. Для устранения данных проблем предлагается располагать один из измерительных модулей в непосредственной близости от системы коммутации (рисунок 4.6). Такое решение позволяет принимать сигнал непосредственно от выбранного преобразователя, исключая необходимость его передачи на значительные расстояния в аналоговом виде. Минимизация длины аналогового тракта

существенно снижает уровень наводок и шумов, обеспечивает стабильность амплитудно-фазовых характеристик и повышает точность измерений. В результате достигается оптимальный баланс между компактностью системы, надёжностью передачи данных и высокой точностью измерительных операций.

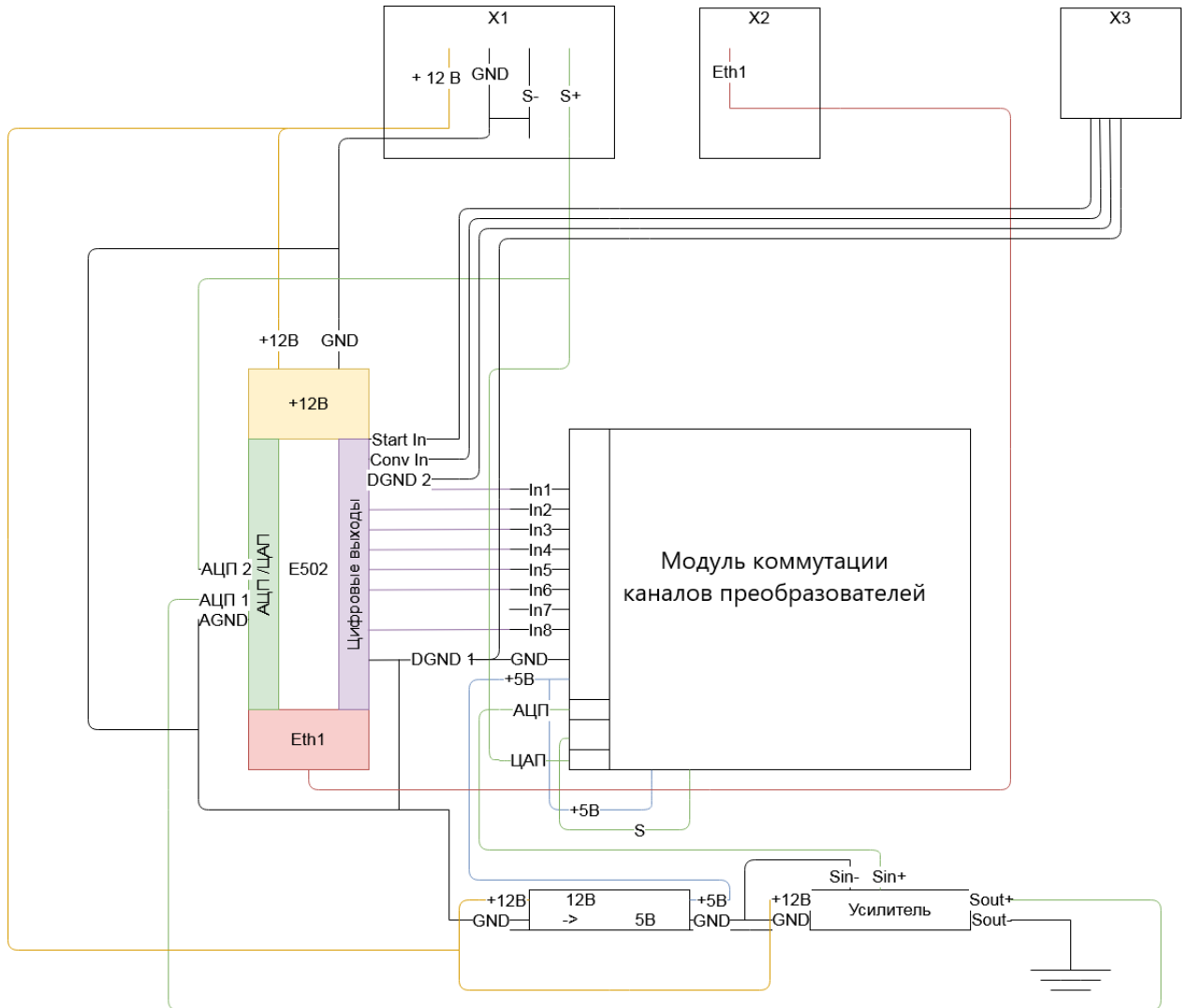


Рисунок 4.6. – Схематичное соединение оснастки многоканальной системы преобразователей

Рассмотрим процесс моделирования в системе автоматизированного проектирования многоканальной системы преобразователей. Происходит моделирование основания с необходимыми стойками под измерительные устройства, систему коммутации и другие приборы (Рисунок 4.7). Блок с преобразователями проектируется модулем цилиндрической формы. На нем располагаются стойки для оснастки и модулей.

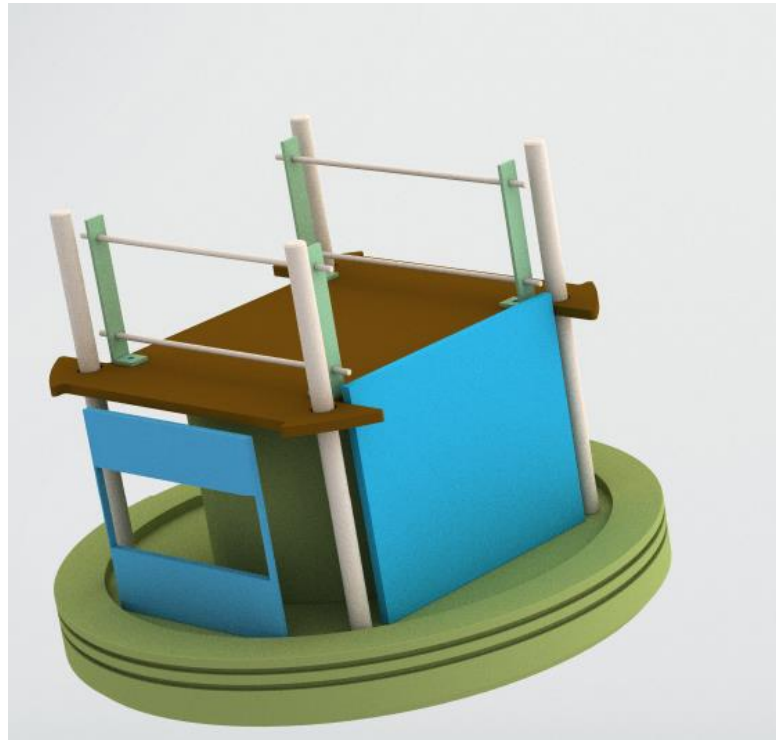


Рисунок 4.7. – 3D модель основания многоканальной системы преобразователей.

Происходит моделирование и установка оснастки и измерительного модуля на основание системы (Рисунок 4.8).

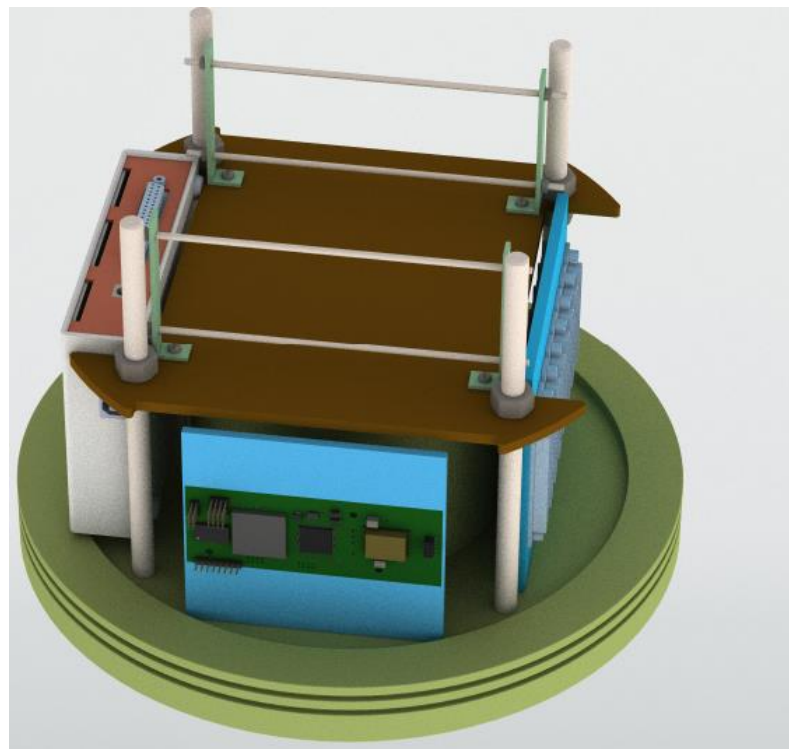


Рисунок 4.8. – 3D модель оснастки на основании системы.

Далее происходит моделирование и установка системы коммутации и управления на соответствующую позицию системы (4.9).

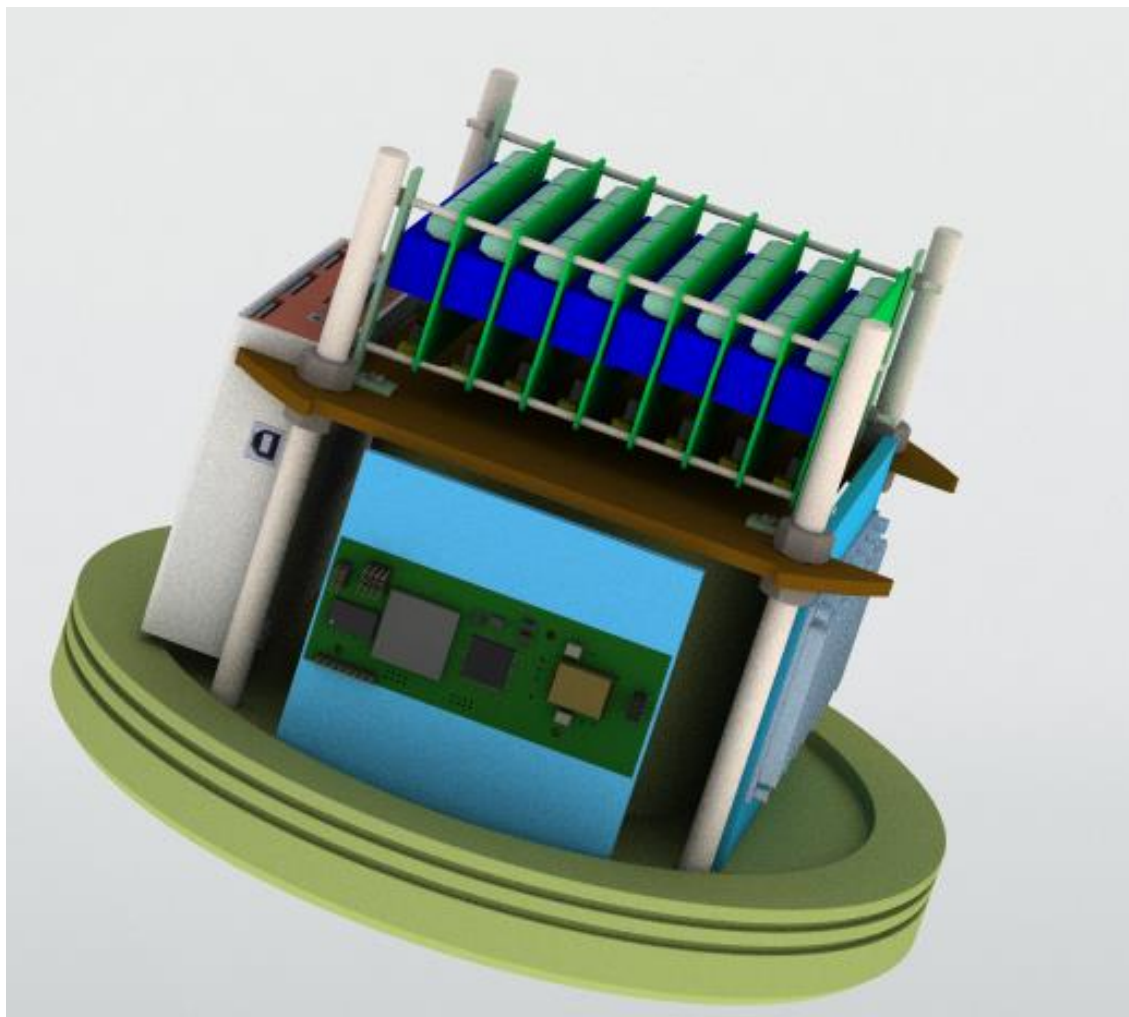


Рисунок 4.9. – 3D модель многоканальной системы преобразователей с измерительным модулем, необходимой оснасткой и системой коммутации.

Далее на основе разработанных 3D-моделей осуществляется формирование соответствующей физической модели системы. На этом этапе производится изготовление или сборка элементов конструкции, их монтаж и последующая интеграция в единый узел. После изготовления компонент выполняется сборка системы коммутации и её подключение к измерительному модулю с целью проведения предварительных испытаний «на столе» (рисунок 4.10). Такой подход позволяет оценить корректность функционирования аппаратных узлов до установки оборудования в реальных условиях эксплуатации.

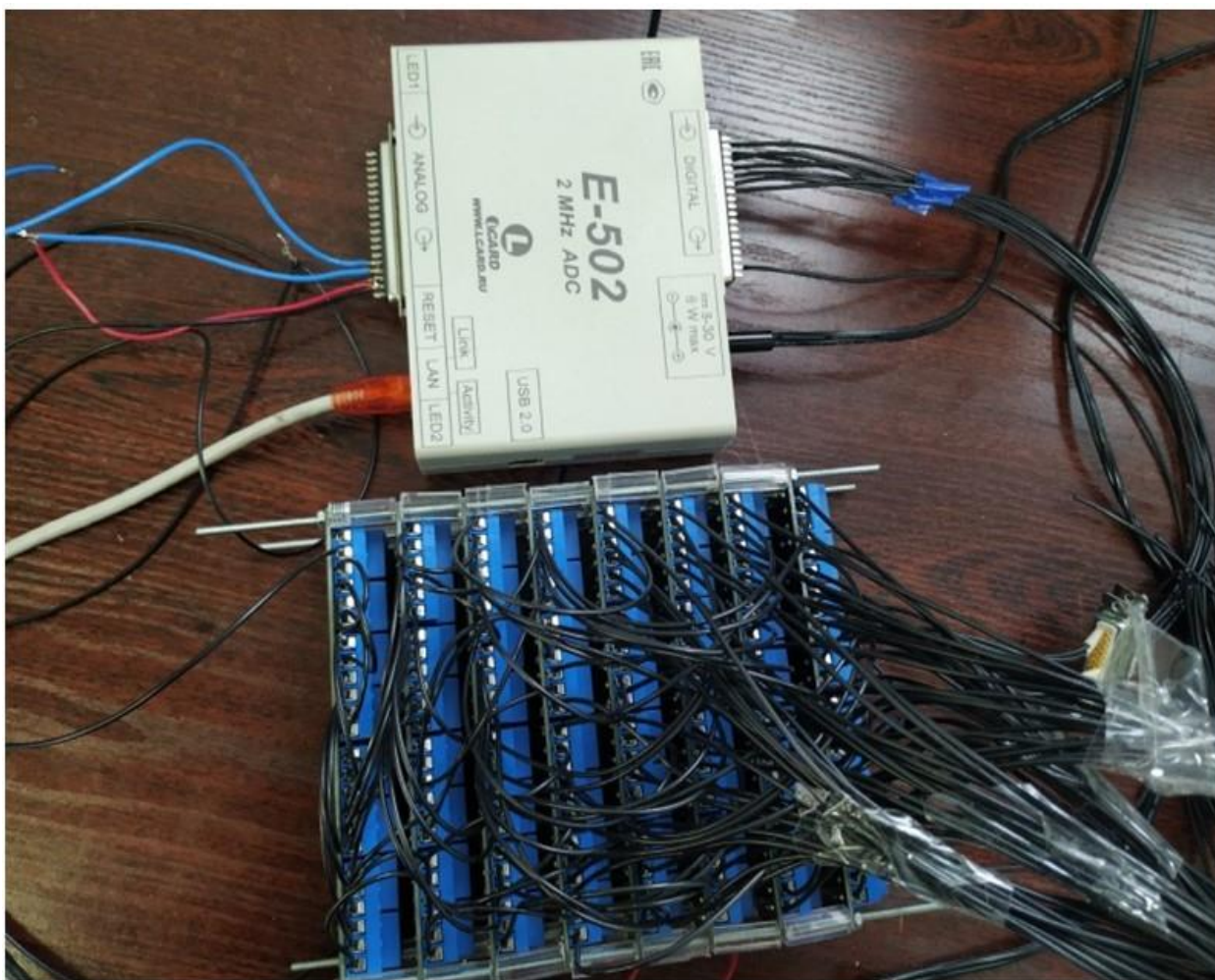


Рисунок 4.10. – Измерительный модуль и система коммутации.

На данном этапе осуществляется сборка всех необходимых компонентов комплекса, их подключение и проверка работоспособности (рисунок 4.11). Аппаратная измерительная часть включает в себя систему коммутации, измерительный модуль и измерительный усилитель, предназначенный для повышения уровня сигналов, формируемых преобразователями. Кроме того, в состав комплекса входят соединительные жгуты и разъёмы, обеспечивающие подачу питания, синхронизацию работы модулей, а также кабели Ethernet для передачи цифровых данных к ведущему измерительному модулю.

Ведущий измерительный модуль располагается в другой физической среде, на рабочем месте, вблизи компьютера с установленным программным обеспечением комплекса. Такая архитектура обеспечивает разделение измерительной и управляющей частей системы, минимизирует влияние внешних факторов на точность измерений, а также позволяет централизованно

контролировать процессы сбора и обработки данных. Организация каналов передачи информации в цифровом виде через Ethernet обеспечивает высокую скорость обмена данными, стабильность сигналов и совместимость с различными программными компонентами комплекса, что особенно важно при работе с многоканальными измерительными системами и в условиях повышенных требований к точности и надёжности измерений.

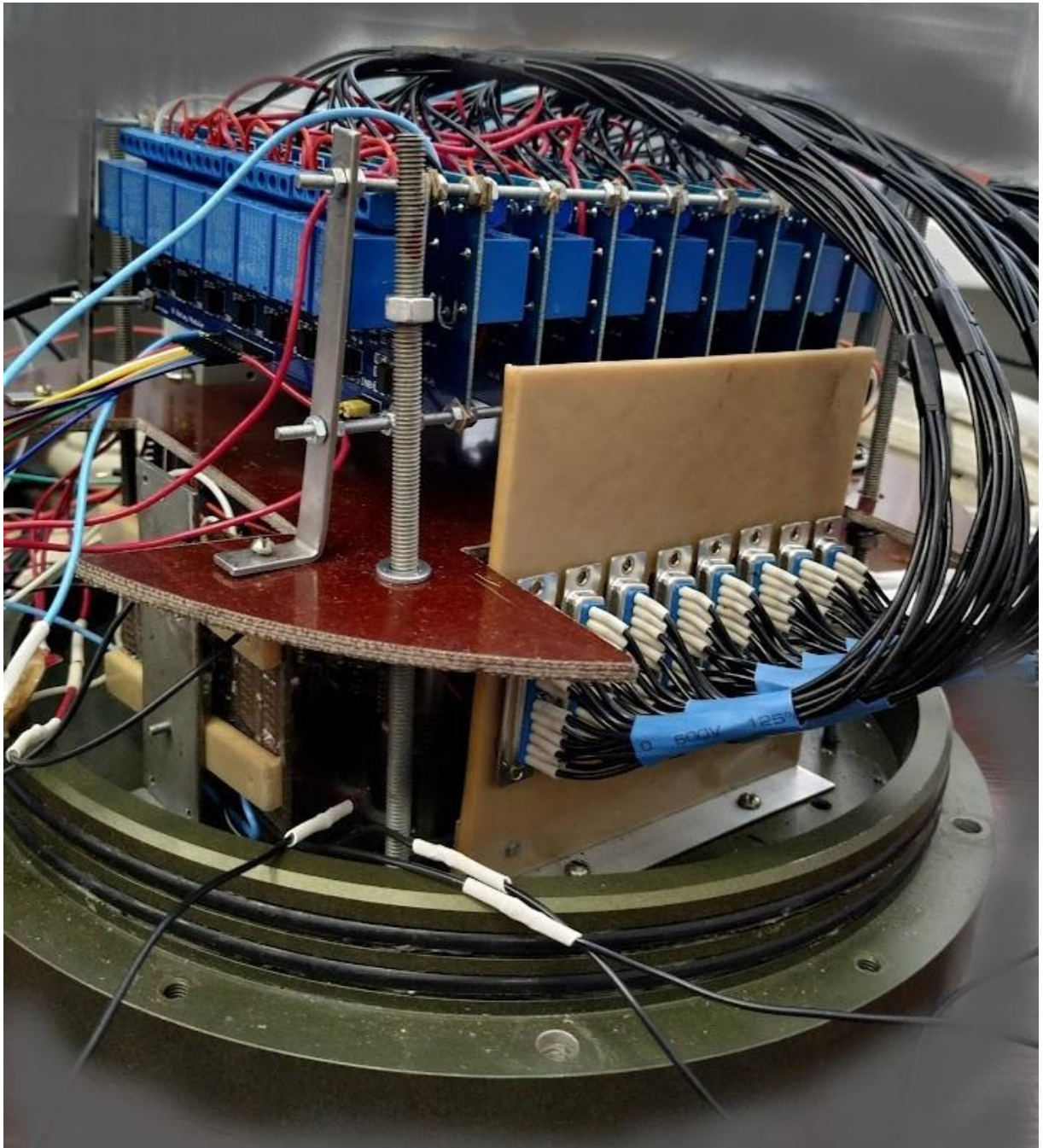


Рисунок 4.11. – Сборка аппаратной части комплекса.

В режиме приема, сигнал преобразователя, при использовании встроенного усилителя, поступает к измерительному модулю. Встроенный усилитель можно не использовать, настроив измерения таким образом, чтобы принимаемый сигнал был достаточен для регистрации измерительным модулем. Аналоговый сигнал преобразуется измерительным модулем в цифровой и передается по кабелю Ethernet к персональному компьютеру.

Измерительный модуль соединяется с модулем-мастером, для синхронизации работы. Таким образом каждый сигнал на каждой частоте для каждого канала будет находиться на одинаковых позициях для других каналов. Данное условие позволяет проводить обработку принятых данных в процессе измерений без повторного поиска полезного сигнала.

В режиме излучения модуль в непосредственной близости к преобразователям решает задачу регистрации опорного сигнала. Т.к. для наибольшей точности необходима информация о сигнале в момент излучения. Если регистрировать данные опорного сигнала в момент излучения, то при передаче через проводник происходят аналогичные принимаемым данным искажения, а следовательно точность измерений падает. Встроенный модуль регистрирует опорный сигнал в момент излучения, минимизируя искажения излучаемого сигнала.

В качестве измерительного модуля, для реализации модели, был выбран модуль L-Card E-502. Для разработки программного обеспечения было принято решение использовать язык программирования с++ и библиотека QT для создания графического интерфейса пользователя (Приложение Г). На рисунке 4.12 представлен графический интерфейс пользователя разрабатываемого программного обеспечения [50]. Группа 1 содержит кнопки обеспечивающие управление измерениями. Кнопки «Старт» и «Стоп» отвечают за запуск и приостановку программы измерений в установленных режимах соответственно. В данной группе имеется возможность задать режим работы (излучение или приём). При установке флага «Регистрация» будет происходить регистрация данных измерений по выбранным параметрам. Если

не устанавливая данный флаг, то программное обеспечение будет работать в режиме осциллографа, позволяя провести настройку измерений. Флаг «Обработка» запускает модуль обработки, который представляет собой отдельный процесс. Это необходимо для возможности автономной и независимой работы по обработке зарегистрированных данных. Данный флаг запускает процесс обработки.

Группа 2 позволяет задавать диапазон измеряемых каналов. На данном этапе можно установить значения от и до в одинаковое значение, тогда будет произведена обработка одного выбранного канала. При установке значения «От» в 0, будет произведена работа со всеми выбранными каналами, число которых установлено в поле «До».

Группа 3 позволяет производить настройку диапазона измеряемых частот и шаг частоты. В режиме, когда отключен флаг «Регистрация» группы 1, итерация частот не будет происходить. В качестве рабочей частоты будет использоваться значение, установленное в поле «От».

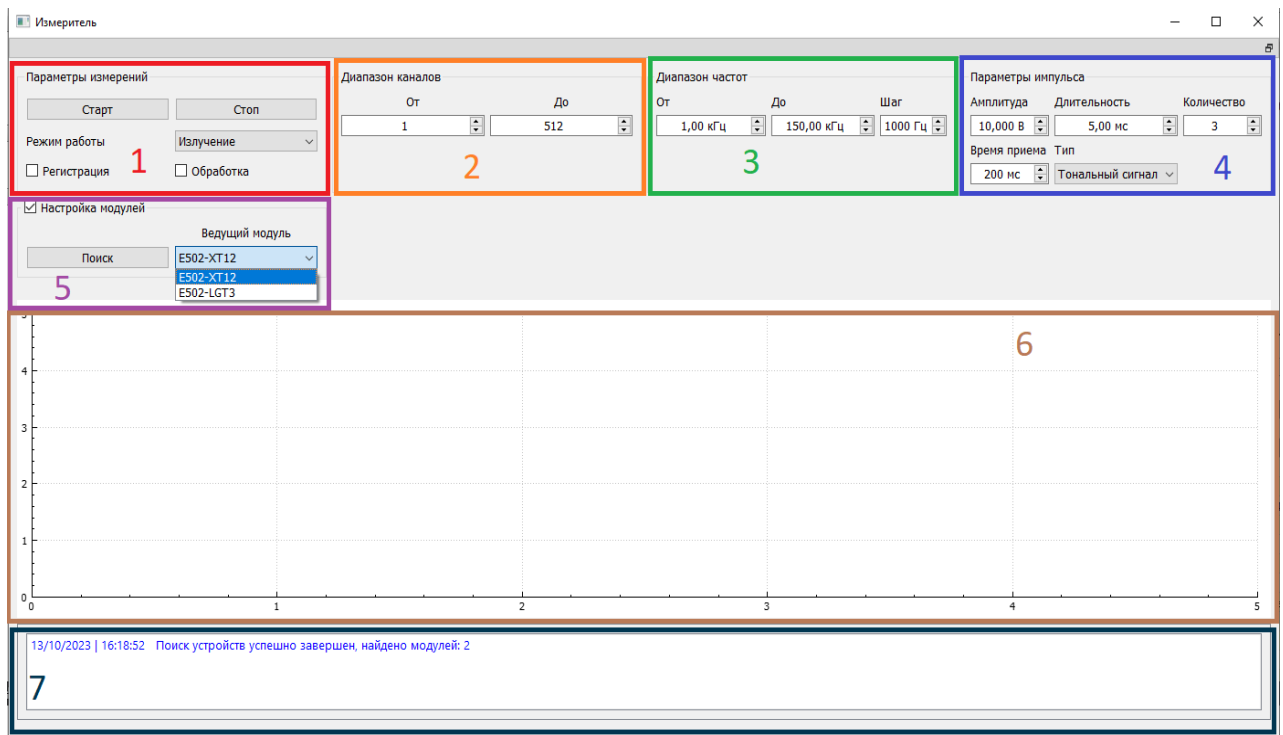


Рисунок 4.12. – Графический интерфейс пользователя модуля регистрации

Группа 4 задает параметры импульса, такие как амплитуда, длительность, количество полезных импульсов, время приема и тип. Данные параметры будут отправляться в измерительный модуль.

Группа 5 позволяет производить поиск устройств. Все найденные устройства будут заноситься в выпадающий список, в котором можно выбрать ведущий модуль. Ведущий модуль является основным рабочим модулем программы при работе в режиме с несколькими модулями. Именно от него происходит отправка команд на зависимые модули.

Полотно 6 – это элемент управления графической формы, куда будут выводиться данные прямых измерений и рисоваться графики. По оси абсцисс отображается время. По оси ординат – амплитуда. Все ошибки и сообщения, статус обработки и обрабатываемый канал выводятся в поле 7.

Группы 1-5 содержатся на панели, который можно сворачивать или перемещать вне графической формы, для увеличения размера полотна. После поиска устройств группу настройки модулей можно свернуть, нажав на соответствующий флаг (Рисунок 4.13).

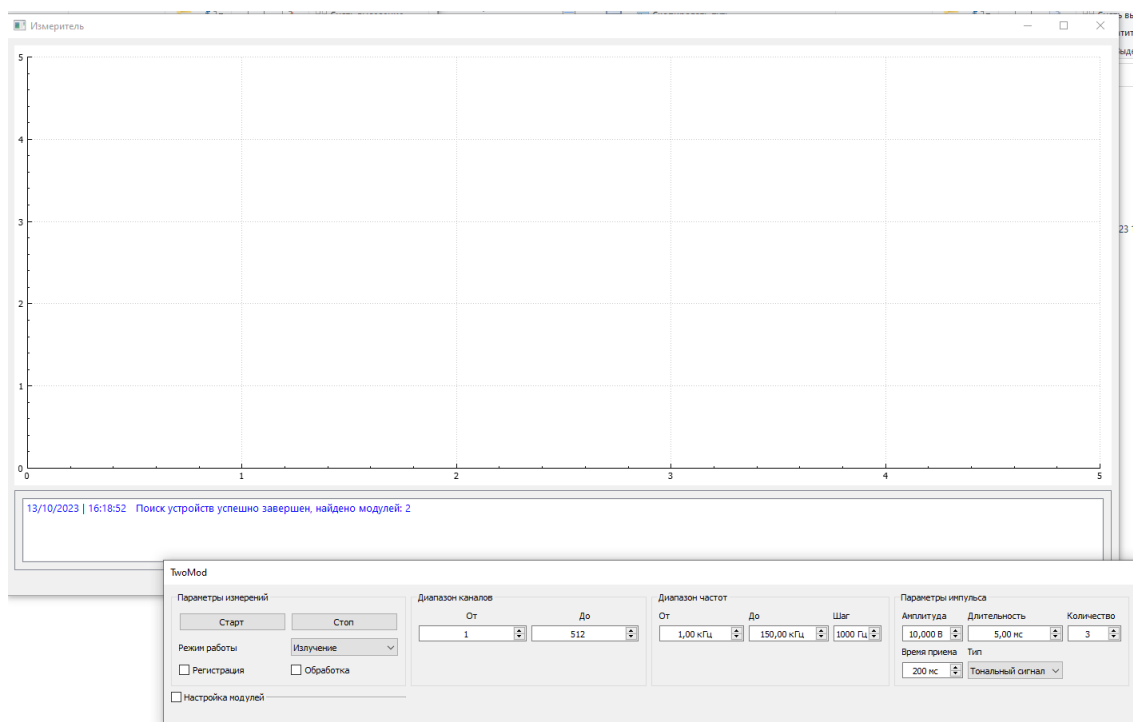


Рисунок 4.13. – Графический интерфейс пользователя.

При запуске программы измерений с выбранным режимом обработки, запускается модуль обработки (Рисунок 4.13). В меню 1 можно выбрать папку или файл для обработки (Рисунок 4.14). При выборе файла или папки, программа переходит в автономный режим работы. Появляются все зарегистрированные файлы папки, которые можно просмотреть и выбрать параметры обработки.

Параметры обработки 2 аналогичные режиму работы при измерениях. Можно задать при помощи селекции границы полезного сигнала. Если этого не сделать, то модуль в автоматическом режиме произведет поиск границ полезного сигнала и расстояния между ними «Дельта» согласно вышеприведенному алгоритму. Данный процесс занимает определенное время, по этой причине, имеется возможность выбора расстояния между импульсами и границ полезных сигналов с использованием селектора. Выбирается коэффициент базы БПФ. В режиме обработки без регистрации, также настраивается базу БПФ, при котором будет синхронная работа системы.

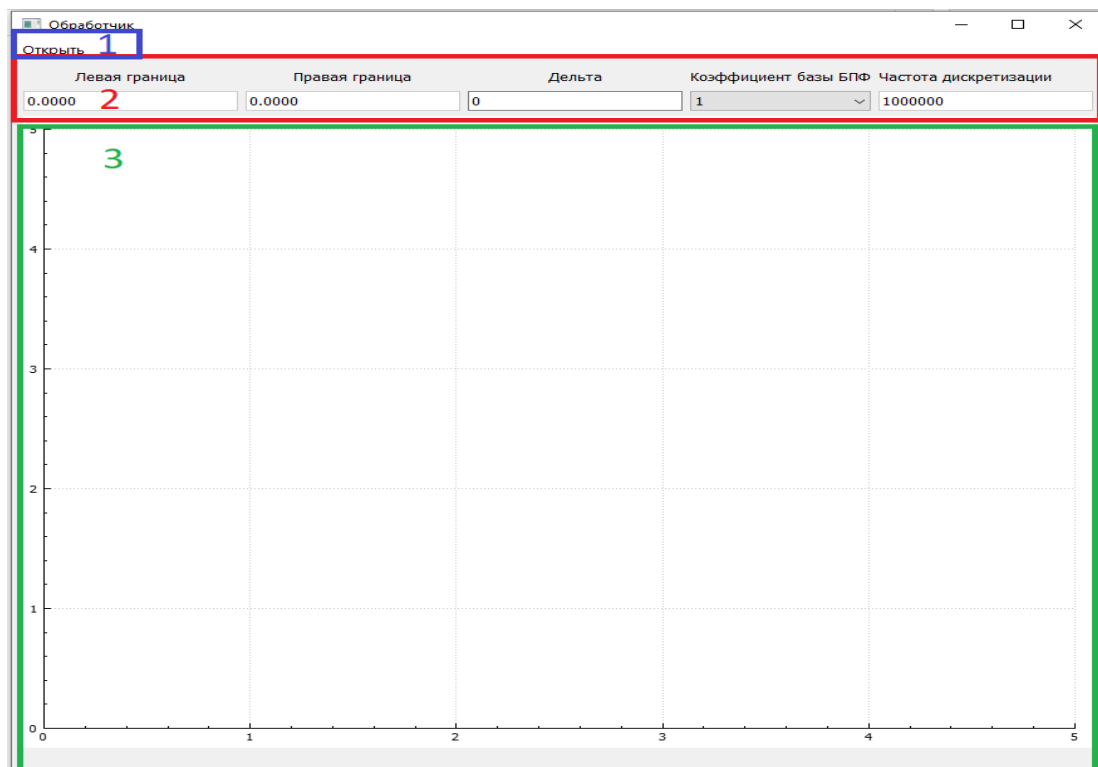


Рисунок 4.14. – Модуль обработки.

Для тестирования программного комплекса собирается схема «на столе» (Рисунок 4.15). Для моделирования настоящего эксперимента, необходимо использовать два измерительных модуля, которые через Ethernet порт соединяются с персональным компьютером [52]. Между собой модули соединяются витой парой для межмодульной синхронизации. Генератор измерительного модуля 1, который в программном комплексе должен быть выбран мастером, передает сгенерированный сигнал на канал АЦП1 этого же модуля и на приемно-передающее устройство, которое с некоторым искажением передает сигнал к измерительному модулю 2. Данный модуль с некоторой задержкой принимает сигнал и передает на персональный компьютер.

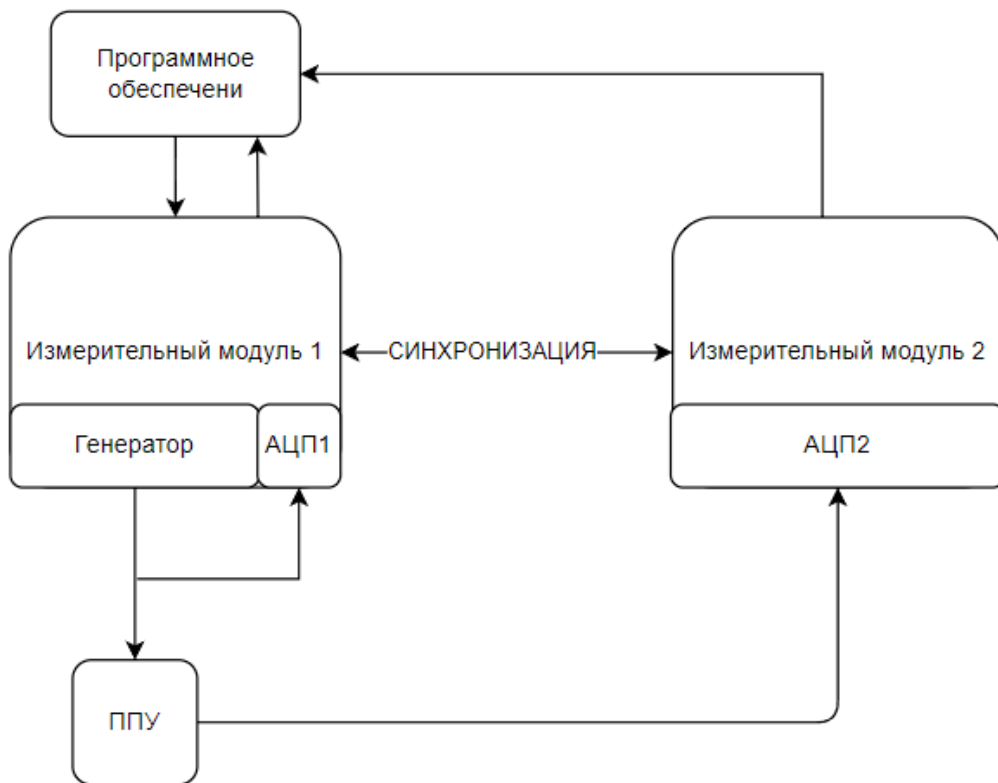


Рисунок 4.15. – Схема для тестирования программного комплекса

Принятые данные с измерительных модулей передаются обратно к персональному компьютеру (Рисунок 4.16). На экране программы можно оценить опорный сигнал и сигнал после ППУ. Происходит регистрация данных в бинарный файл и запуск модуля обработки (Рисунок 4.17).

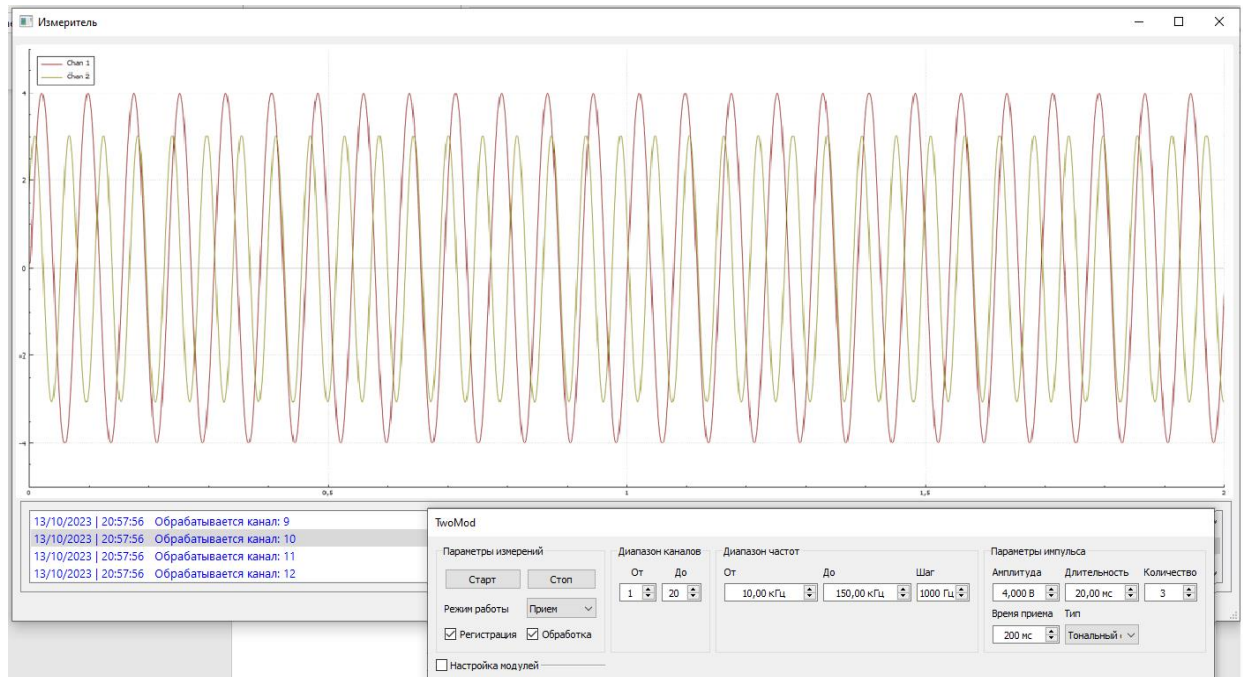


Рисунок 4.16. – Тестовый запуск программного комплекса (регистратор).

Модуль обработки данных выполняет функции идентификации и выделения границ полезного сигнала в потоке измерений [51]. На первом этапе устанавливается дельта, определяющая пороговые значения для определения начала и конца сигнала, после чего запускается процесс обработки. В рамках обработки осуществляется вычисление амплитудно-частотной характеристики (АЧХ) выбранного канала, что позволяет оценить частотные и амплитудные параметры исследуемого преобразователя.

Результаты обработки отображаются в пользовательском интерфейсе модуля, что обеспечивает оперативный контроль качества измерений и визуальный анализ сигнала. Одновременно производится регистрация всех вычисленных данных в соответствующих файлах для последующей обработки, анализа и архивирования. Такой подход обеспечивает возможность многократного использования измерений, проведения статистической обработки, а также интеграции данных с другими модулями комплекса для автоматизированного прогнозирования и оптимизации параметров преобразователей.

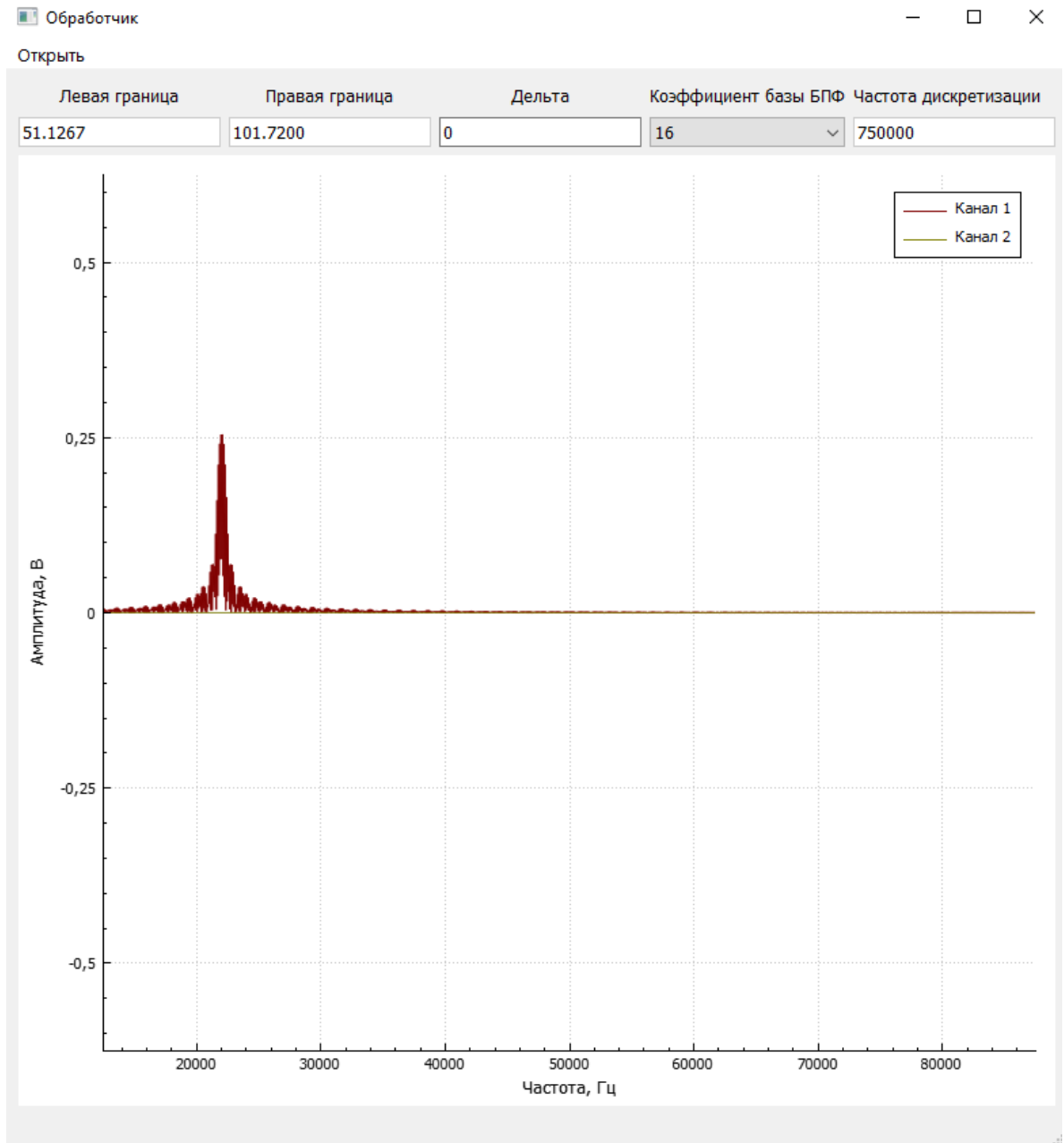


Рисунок 4.17. – Модуль обработки

Модуль обработки может работать в автономном режиме (Рисунок 4.18). Для этого необходимо открыть папку с зарегистрированными файлами. Активируется выбранная папка и в списке объектов будут отображаться все выбранные файлы директории. При клике по файлу происходит отображение выбранного файла на графической форме. Здесь же можно селекцией выбрать диапазон полезных сигналов. Если этого не сделать, то модуль обработки, как и в режиме работы измерений, при запуске измерений автоматически единоразово определит позиции полезных сигналов.

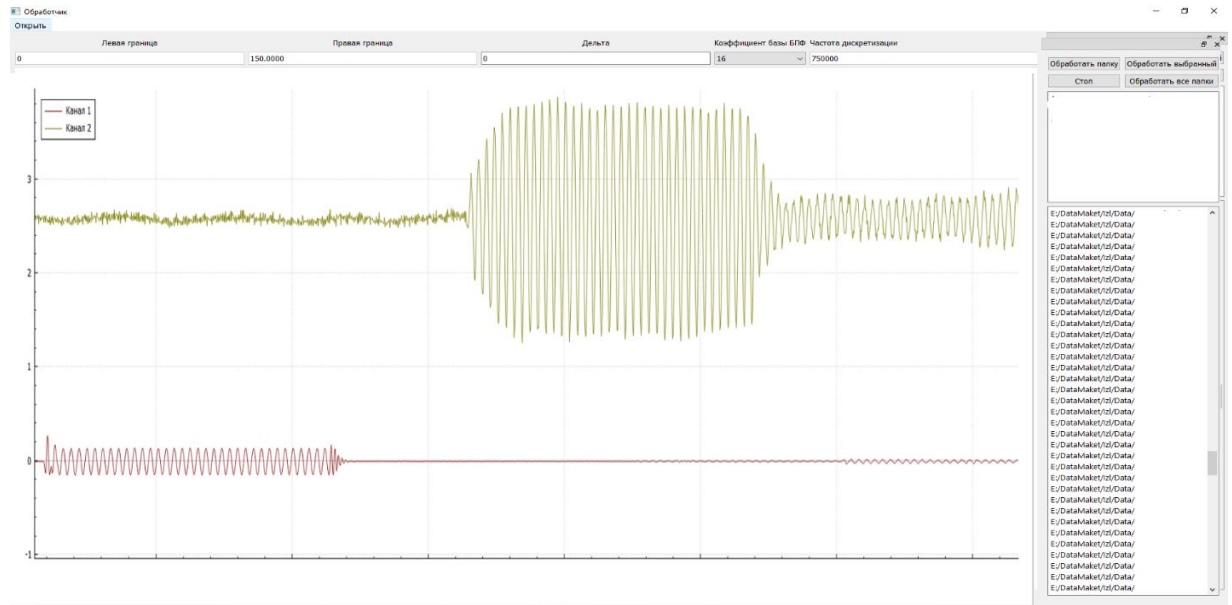


Рисунок 4.18. – Автономная работа обработчика сигналов

Разработанный модуль обработки предоставляет возможность анализа как всех файлов, находящихся в заданной директории, так и отдельного выбранного файла (рисунок 4.19). Обработка сигналов выполняется в соответствии с разработанным алгоритмом, который включает предварительную фильтрацию, определение границ полезного сигнала и вычисление амплитудно-фазовой частотной характеристики (АФЧХ) системы.

Все результаты обработки регистрируются в отдельный файл, что обеспечивает возможность последующего анализа, архивирования и интеграции данных с другими модулями комплекса. Одновременно вычисленные АФЧХ отображаются на графической форме модуля, что позволяет оператору визуально оценивать динамические и частотные характеристики исследуемой системы. Такой подход обеспечивает как автоматизацию обработки большого объема данных, так и наглядное представление результатов измерений, что повышает точность анализа и удобство работы с многоканальными измерительными системами.

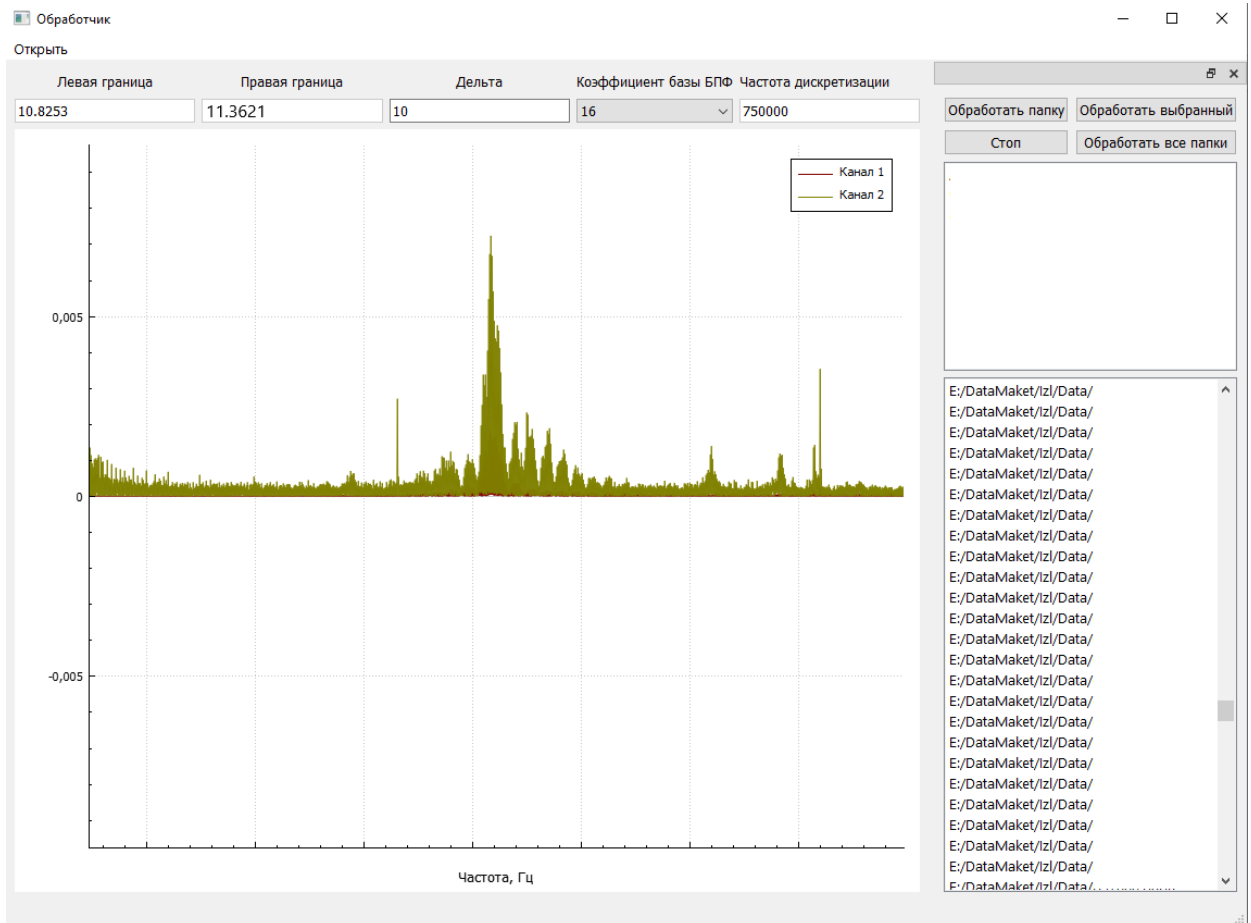


Рисунок 4.19. – Обработка сигналов из файла

Программный комплекс разработан по приведенному алгоритму и реализовывает модель автоматизации процесса контроля качества многоканальных систем преобразователей. Была проведена государственная регистрация разработанного программного обеспечения (Приложение А).

4.2 Программный модуль автоматизации процесса разработки технических требований к преобразователям

Для разработки программного модуля автоматизации разработки технических требований к преобразователям, необходимо решить задачу определения физических параметров преобразователей по требуемым частотным характеристикам. Для выполнения задачи определения базовых параметров преобразователей в программном модуле-измерителе

предусмотрена специализированная клавиша «Задача параметров преобразователей» (Рисунок 4.20). Использование данной функции позволяет загружать конфигурирующие файлы в формате *.csv, содержащие обработанные результаты измерений, которые применяются для обучения модели прогнозирования.

Файлы *.csv включают как исходные параметры исследуемых преобразователей, так и соответствующие амплитудно-фазовые частотные характеристики на выбранных частотных точках.

Фрагмент измеренных АЧХ для группы преобразователей с заданными параметрами представлен на Рисунке 4.19, что демонстрирует распределение амплитудных и фазовых характеристик в диапазоне исследуемых частот. Эти данные служат основой для последующего анализа, калибровки модели и прогнозирования оптимальных параметров новых преобразователей с учётом заданных диапазонов частот и конструктивных особенностей.

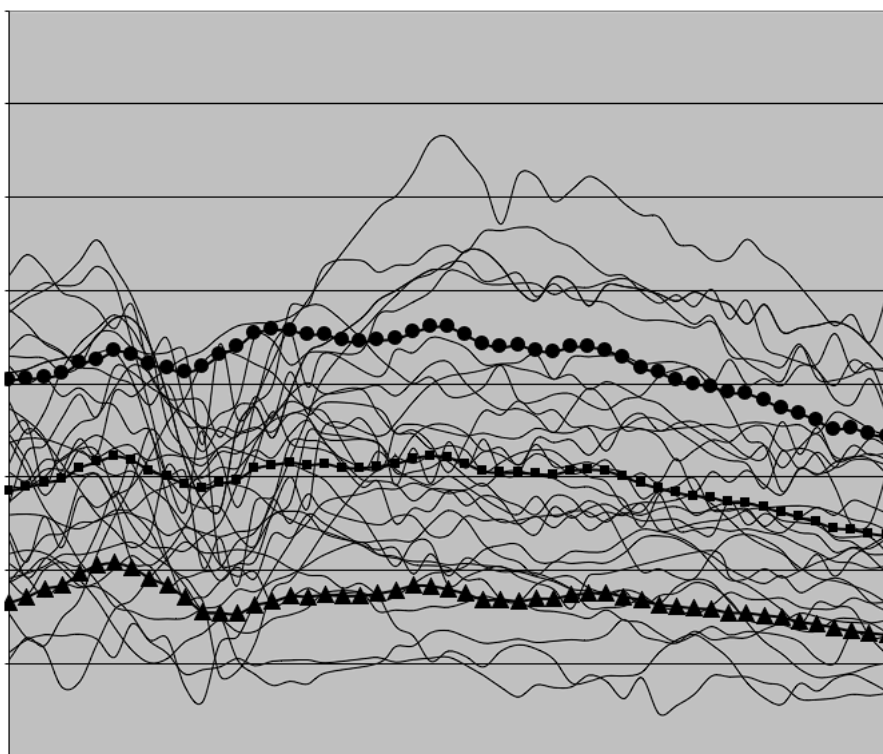


Рисунок 4.19. – Фрагмент результатов измерений.

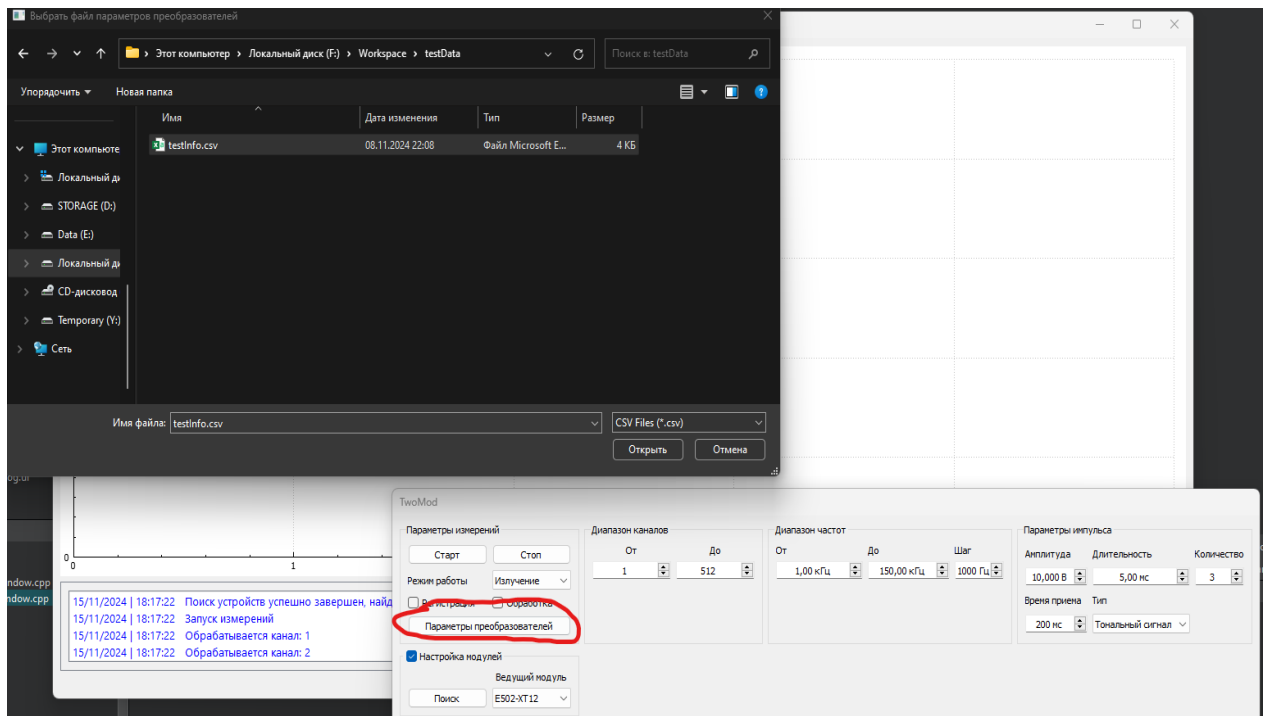


Рисунок 4.20. – Выбор параметров преобразователей.

Для составления прогноза необходимо запустить модуль прогнозирования. На рисунке 4.21 представлен графический интерфейс программного модуля прогнозирования, разработанного с использованием технологий C++/Qt и Python (Приложение Д). Имеется возможность ввода пользователем диапазона частот, в пределах которого требуется подобрать параметры преобразователя. Это может быть, например, центральная резонансная частота или диапазон рабочей полосы. Кнопка «Старт» используется для запуска процесса прогнозирования. При первом запуске открывается диалоговое окно для выбора файла данных: CSV-файл: запускается процесс обучения новой модели по этим данным. RKL-файл: используется ранее обученная модель. При выборе CSV-файла начинается обучение модели на основе текущих параметров. При выборе модели в формате RKL – производится прямое прогнозирование. После выполнения прогноза таблица заполняется вычисленными параметрами преобразователя (например, материал, масса, геометрия), соответствующими заданной частотной характеристике.

Прогнозирование

Диапазон частот: От 1000,00 Гц До 5000,00 Гц Старт

Материал	Геометрия	Способ установки	Масса (г)	Размер X (см)	Размер Y (см)	Размер Z (см)	Частота	пропускания	чувствительность	ность прогнозирования
Алюминий	Цилиндрическая	Сварное	58.23	4.55	3.87	2.9	12235.1	1323.5	-10.44	94.2...
Сталь	Цилиндрическая	Сварное	34.51	8.12	4.3	7.51	11234.2	1123.4	-12.68	93.5
Сталь	Кубическая	Склеенное	44.87	5.63	6.98	9.1	12132.5	1213.3	-13.68	93.2
Алюминий	Кубическая	Склеенное	72.44	3.42	5.99	8.42	10845.6	1084.6	-13.11	92.1
Полимер	Кубическая	Склеенное	41.73	7.34	2.97	4.24	10425.7	1070.2	-9.53	91.2
Титан	Сферическая	Болтовое	52.97	5.2	4.45	8.03	13640.5	1364.1	-15.02	90.4
Полимер	Цилиндрическая	Спаянное	63.17	6.57	4.88	3.21	12623.8	1262.4	-14.32	89.3
Сталь	Сферическая	Болтовое	48.83	7.17	6.59	2.73	11472.5	1147.3	-7.82	88.6
Сталь	Конусоидальная	Спаянное	38.51	8.02	7.6	5.17	11556.7	1155.7	-14.56	87.8...
Алюминий	Конусная	Склеенное	69.42	4.78	5.24	4.92	11472.5	1147.3	-7.82	87.2

Рисунок 4.21. – Пример использования модуля прогнозирования

4.3 Выводы по Главе 4

1. Апробация результатов моделирования автоматизированного процесса контроля качества многоканальной системы преобразователей проводится на многоканальной системе гидрофонов, т.к. данная система и преобразователи в полной мере отображают возможности модели.

2. Аппаратная часть комплекса разработана на основе соответствующей 3D – модели, созданной в САПР. Система коммутации основана на реализации схемы демультиплексора на восьмиканальных релейных модулях.

3. Программная часть комплекса позволяет задавать параметры сигналов и измерений, а также регистрировать физические параметры преобразователей, которые используются для обучения модуля прогнозирования.

4. Программная часть комплекса состоит из модуля проведения измерений и модуля обработки данных. Модуль обработки данных реализован отдельной программой, для возможности проведения расчетов в лабораторных условиях, помимо возможности использования функции обработки результатов в процессе измерений.

5. Программный модуль автоматизации разработки технических требований к преобразователям основан на прогнозировании физических характеристик преобразователей по заданным частотным характеристикам. Программный модуль позволяет выбирать результаты измерений для обучения модели и использовать обученную модель для составления прогноза.

ГЛАВА 5. ОЦЕНКА ЭФФЕКТИВНОСТИ ИСПОЛЬЗОВАНИЯ СИСТЕМЫ АВТОМАТИЗАЦИИ ПРОЦЕССА КОНТРОЛЯ КАЧЕСТВА МНОГОКАНАЛЬНЫХ СИСТЕМ ПРЕОБРАЗОВАТЕЛЕЙ И МОДУЛЯ АВТОМАТИЗАЦИИ ПРОЦЕССА РАЗРАБОТКИ ТЕХНИЧЕСКИХ ТРЕБОВАНИЙ К ПРЕОБРАЗОВАТЕЛЯМ

5.1. Оценка эффективности использования системы автоматизации процесса контроля качества многоканальных систем преобразователей

Для старта проведения контроля качества, после проведения отладочных работ, пользователь устанавливает параметры измерений и нажимает на кнопку «Старт» (Рисунок 5.1).

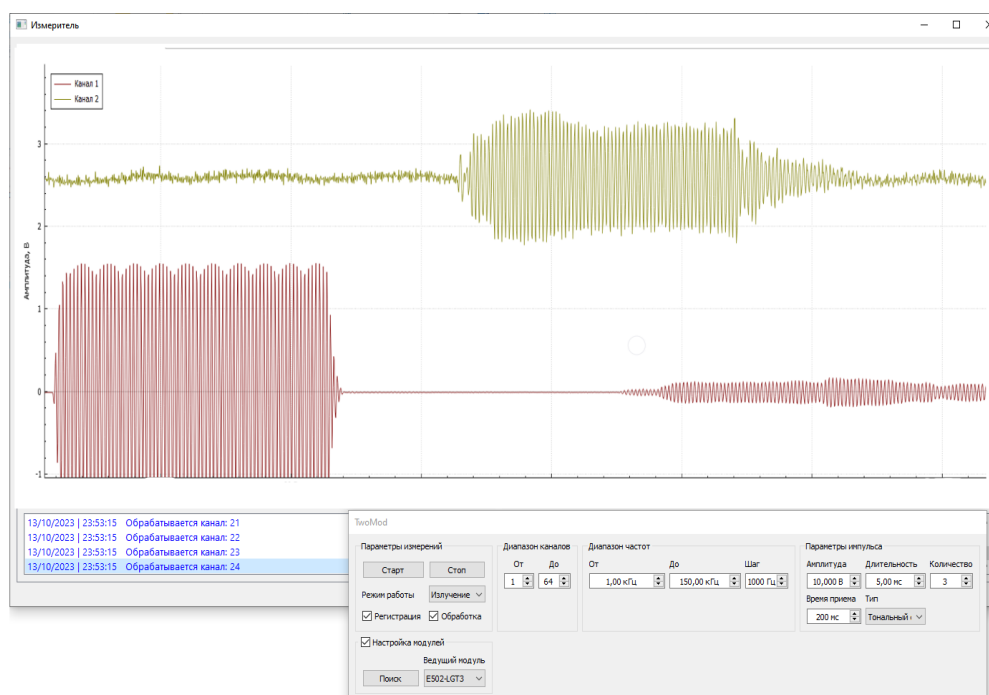


Рисунок 5.1. – Проведение измерений.

Одновременно с этим запускается модуль обработки (Рисунок 5.2). Процесс обработки и регистрации происходят параллельно. Также происходит отображение данных на графических формах модулей.

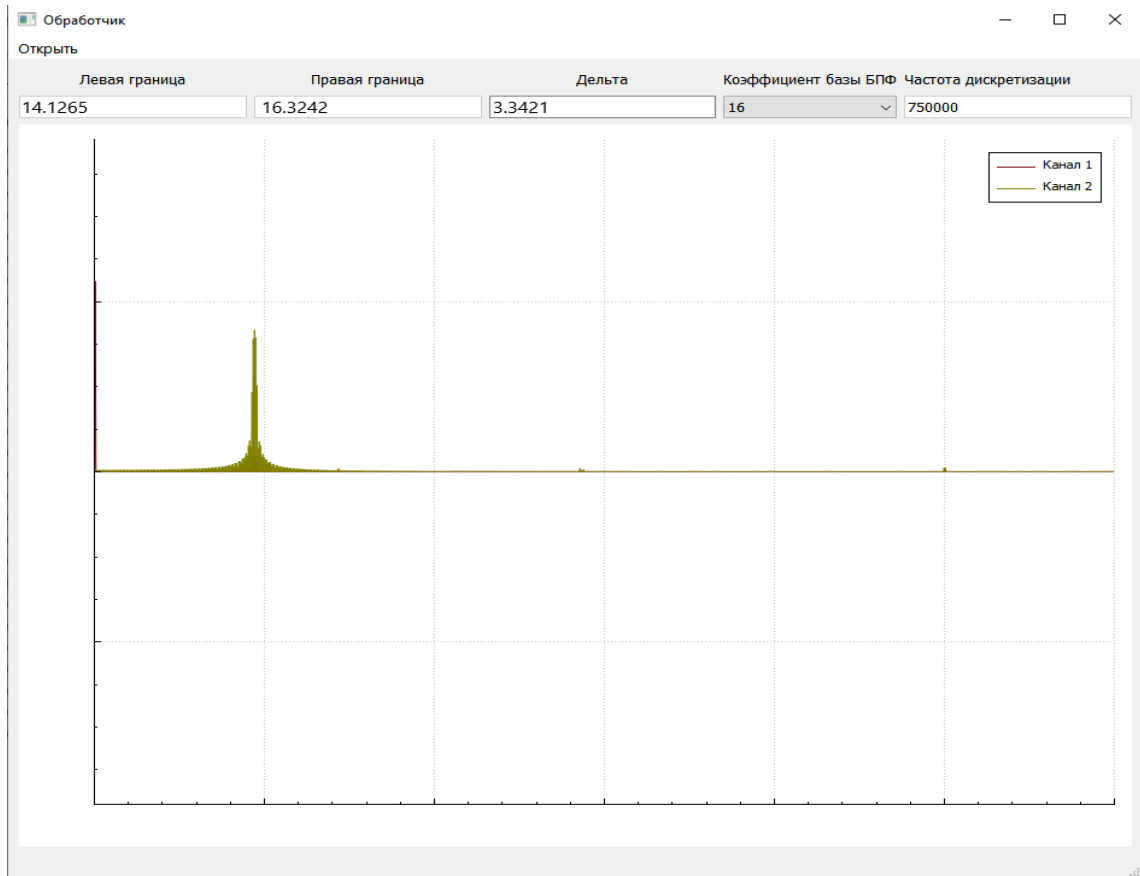


Рисунок 5.2. – Обработка сигнала канала

Раздельная реализация модулей обработки и регистрации данных измерений обеспечивает возможность проведения анализа сигналов в лабораторных условиях после фактического выполнения измерений. Такая архитектура позволяет отделить процесс сбора данных от их последующей обработки, что повышает гибкость работы с измерительной информацией и облегчает повторное использование зарегистрированных данных для различных видов анализа.

Модуль обработки предоставляет возможность оценки зарегистрированных сигналов, выделения области полезного сигнала и последующего запуска автоматизированной обработки, полностью аналогичной процессу, выполняемому непосредственно во время проведения измерений (Рисунок 5.3). Это обеспечивает воспроизводимость результатов и позволяет проводить постфактум проверку корректности измерений, оценку качества сигналов и выявление аномалий без необходимости повторного проведения экспериментальных процедур. Такой подход существенно повышает

эффективность работы лаборатории, сокращает время на обработку данных и обеспечивает возможность интеграции результатов с другими модулями комплекса для дальнейшего анализа, визуализации и прогнозирования параметров исследуемых преобразователей.

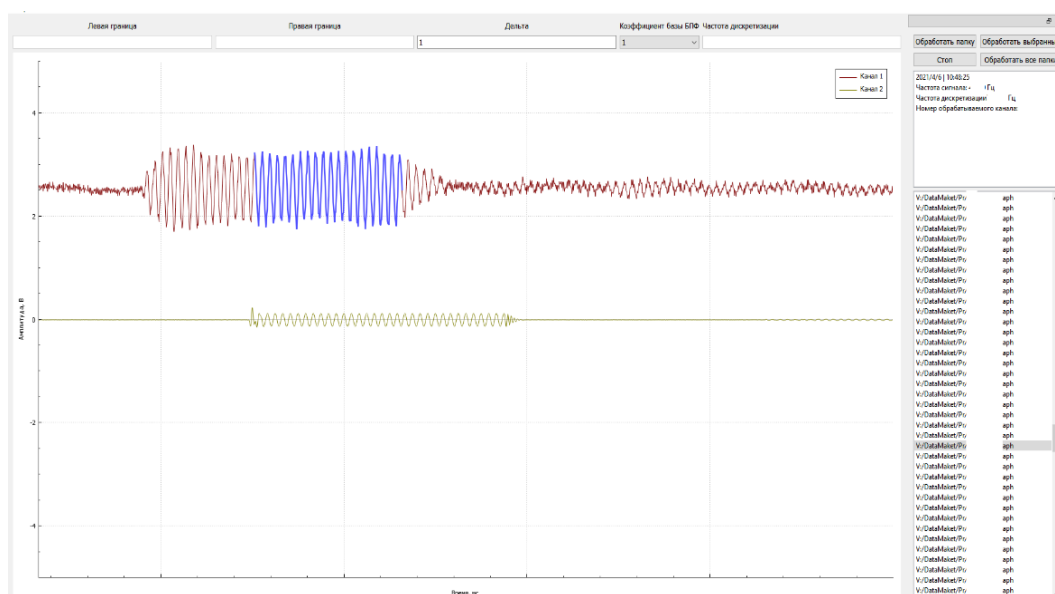


Рисунок 5.3. – Открытие файлов измерений модулем обработки

Обработка сигналов происходит в автоматизированном режиме. Происходит поиск АЧХ, ФЧХ или АФЧХ каналов по требованию. При этом происходит отображение результатов измерений на графической форме модуля обработки (Рисунок 5.4).

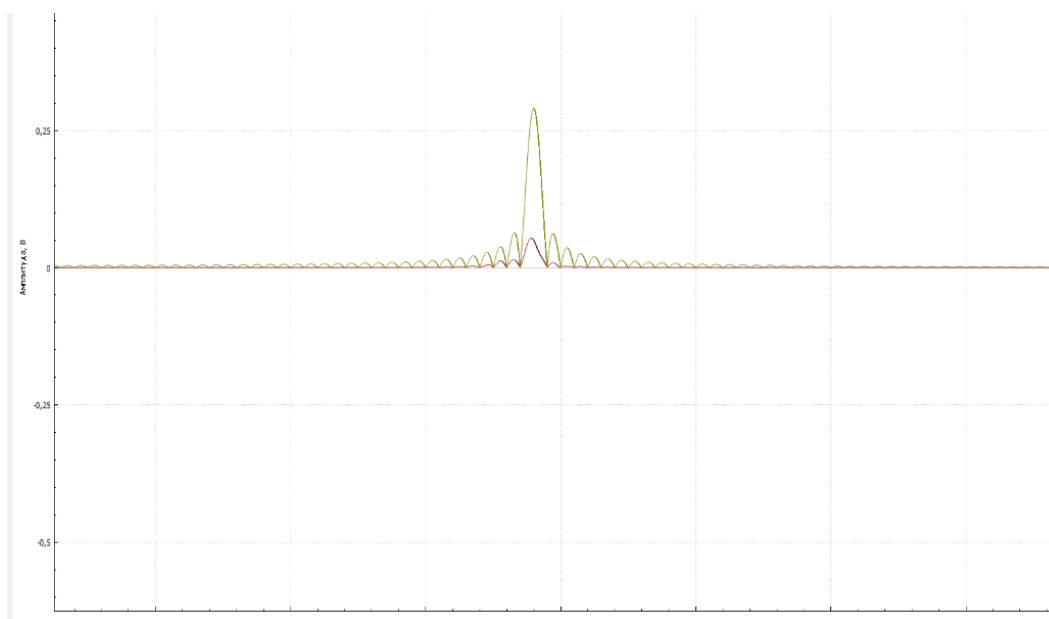


Рисунок 5.4. – Пример работы модуля обработки

По результатам измерений по требованию выводятся необходимые показатели как для одного канала преобразователя (Рисунок 5.5), так и общий результирующий график для всех каналов преобразователей многоканальной системы.

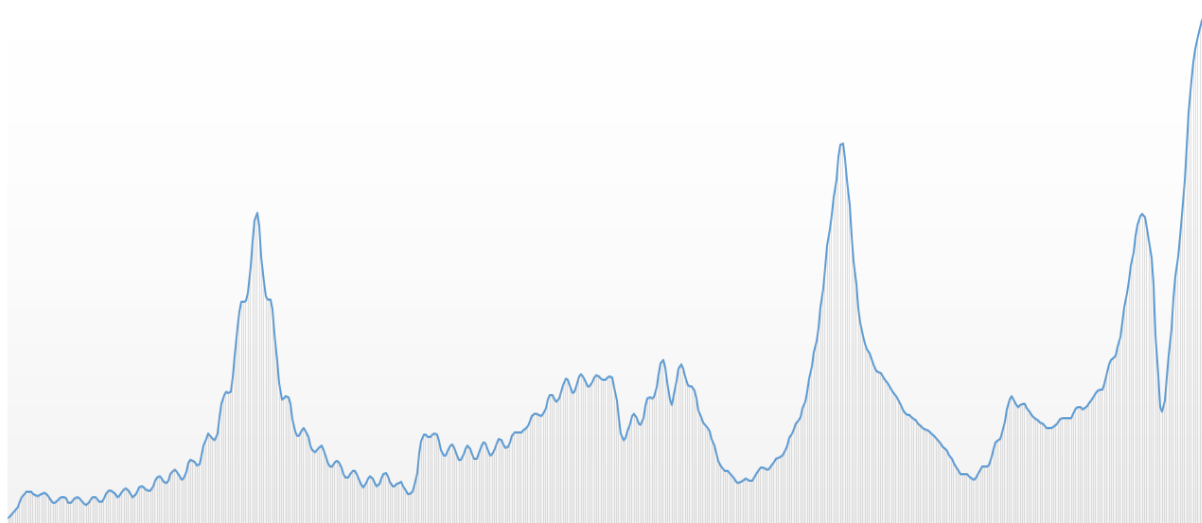


Рисунок 5.5. – Вывод результата системы преобразователей

Для проверки корректности проведенных измерений, результирующие данные сравнивались с результатами ручной проверки на аналогичных каналах и параметрах измерений. Отклонение результатов автоматизированного комплекса от результатов, полученных в ручном режиме, в среднем составило порядка 3%. Был проведен статистический анализ по проведению аналогичных измерений многоканальных систем в ручном режиме. Для проведения аналогичных измерений в ручном режиме, как правило, требуется участие двух квалифицированных инженеров. Один занимается управлением генератором сигналов и переключением каналов. Задачей второго инженера является управление осциллографом, фиксация результата в таблицу и расчет. Статистически (по данным, собранным внутри предприятия), среднее время проведения измерения одного канала на одной частоте с учетом переключения канала, генерации импульса, фиксации, регистрации и расчета необходимых показателей, составляет от 40 до 80 секунд. Время проведения одного измерения увеличивается с увеличением

количества измерений, т.е. чем больше измерений совершается за рабочее время, тем медленнее происходит процесс измерений, это связано с человеческим фактором. Программный комплекс одну аналогичную итерацию выполняет за 2 секунды. Это время не увеличивается с увеличением количества измерений. Нижнюю границу прироста производительности, исходя из собранной статистики, можно оценить как $30 \cdot t$, где t - время, затрачиваемое на проведение одного измерения. Например, при проведении измерений в диапазоне частот от 10кГц, до 100кГц с шагом в 1кГц для одного канала составляет 91 итерацию. Для 58 каналов преобразователей необходимо провести 5278 итераций. Общее время для проведения 5278 измерений с использованием средств автоматизации составит 2.93 часа. Аналогичные измерения без использования средств автоматизации потребуют примерно 88.1 час. Таким образом при проведении контроля качества рассматриваемой системы сокращение времени измерений составляет около 85 часов рабочего времени. Результаты моделирования автоматизированного процесса контроля качества многоканальных систем преобразователей были использованы в действующем производстве (Приложение Б). При использовании разработанной модели автоматизации и программного комплекса в действующее производство, по причине перехода от ручного перехода от ручного способа измерений к автоматизированному, по предварительной оценке, сокращение времени при проведении контроля качества многоканальных систем гидрофонов составит 170 часов в год.

Методика и программный модуль автоматизации процесса контроля качества многоканальных систем преобразователей внедрены в учебный процесс кафедры «Автоматизированные системы обработки информации и управления» ФГАОУ ВО «МГТУ «СТАНКИН» при подготовке бакалавров и магистров по направлению 27.03.04 «Управление в технических системах» в рамках учебной дисциплины «Теория управления».

5.2. Оценка эффективности использования модуля автоматизации разработки технических требований к преобразователям

Результаты использования модуля автоматизации разработки технических требований к преобразователям, позволяют разработчику преобразователей минимизировать время на подбор параметров преобразователя для достижения искомой полосы пропускания или диапазона частот. Таким образом происходит сокращение времени моделирования преобразователей. При проведении измерений макета смоделированных преобразователей, с использованием измерительного модуля, результаты измерений используются для обучения модели прогнозирования, что способствует увеличению точности дальнейших предсказаний.

Зависимость между использованием модели искусственного интеллекта и моделированием, путем проведения экспериментов можно выразить через время, необходимое для достижения определенной точности (например, минимальной ошибки MAE).

Вводятся обозначения.

T_{total}^{exp} – общее время эксперимента.

T_{total}^{model} – общее время использования модели

MAE_{exp} – ошибка при экспериментальном подходе.

MAE_{model} – ошибка при использовании модели.

N – количество комбинаций параметров в экспериментах.

M – количество запросов для модели.

T_{exp} – время одного эксперимента.

T_{train} – время обучения модели.

T_{pred} – Время одного предсказания.

Получаем зависимости времени:

$$T_{total}^{exp} = N * T_{exp} \quad (5.1)$$

$$T_{\text{total}}^{\text{model}} = T_{\text{train}} + M * T_{\text{pred}} \quad (5.2)$$

$$\frac{T_{\text{total}}^{\text{model}}}{T_{\text{total}}^{\text{exp}}} = \frac{T_{\text{train}} + M * T_{\text{pred}}}{N * T_{\text{exp}}} \quad (5.3)$$

Если $M \ll N$ и $T_{\text{pred}} \ll T_{\text{exp}}$, то $T_{\text{total}}^{\text{model}} \ll T_{\text{total}}^{\text{exp}}$, что делает использование модели значительно быстрее.

Зависимость ошибок. Для экспериментов точность напрямую зависит от количества протестированных комбинаций параметров:

$$\text{MAE}_{\text{exp}}(N) \propto \frac{1}{\sqrt{N}} \quad (5.4)$$

Для модели точность зависит от объема данных, на которых она обучена:

$$\text{MAE}_{\text{model}}(D) = \frac{a}{\sqrt{D}} + b \quad (5.5)$$

При одинаковом объеме данных, модель и эксперименты могут достичь схожей точности, но время использования модели гораздо меньше.

В результате получается зависимость эффективности времени и точности:

$$\frac{\text{MAE}_{\text{model}}(D)}{\text{MAE}_{\text{exp}}(N)} \approx \frac{T_{\text{train}} + M * T_{\text{pred}}}{N * T_{\text{exp}}} \quad (5.6)$$

Т.е. с увеличением объема данных модель становится точнее, а общее время использования остается значительно меньше.

Рассмотрим использование данных соотношений на практике.

При $N = 1000$ – количество комбинаций параметров, $T_{\text{exp}} = 1$ час – время одного эксперимента.

$T_{\text{train}} = 10$ часов – время обучения модели.

$M = 1000$ – количество предсказаний(запросов).

$T_{\text{pred}} = 0.01$ часа (36 секунд) – время одного предсказания.

В результате расчетов можно сделать выводы: время использования модели в 50 раз быстрее:

$$\frac{T_{\text{total}}^{\text{model}}}{T_{\text{total}}^{\text{exp}}} = \frac{1000}{20} = 50$$

На старте модель менее точная из-за дополнительной ошибки $b = 0.5$. Однако с увеличением данных ошибка модели приближается к экспериментальной, таким образом если $D = 10000$, то $MAE_{\text{model}}(D) = \frac{a}{\sqrt{D}} + b = \frac{5}{\sqrt{10000}} + 0.5 = 0.55$.

Таблица 2. – Сравнение моделей.

Метод	Время обучения, ч	MAE, Гц	MSE, (Гц) ²	R ²
Эксперименты	1000 ч	0.10	0.015	0.92
CatBoost (ML)	10	0.55	0.50	0.65
CatBoost (ML)	30 (расшир. данные)	0.15	0.03	0.90

Для экспериментов расчет ошибок описывается как:

$$MAE_{\text{exp}}(N) = \frac{k}{\sqrt{N}}, \quad (5.7)$$

где $k = 5$ – эмпирический коэффициент.

Таким образом время составит.

Экспериментальный подход:

$$T_{\text{total}}^{\text{exp}} = N * T_{\text{exp}} = 1000 * 1 = 1000 \quad \text{часов.}$$

Использование модели:

$$T_{\text{total}}^{\text{model}} = T_{\text{train}} + M * T_{\text{pred}} = 10 + 1000 * 0.01 = 20 \text{ часов.}$$

Ошибки составят.

Для экспериментов :

$$MAE_{\text{exp}}(N) = \frac{k}{\sqrt{N}} = \frac{5}{\sqrt{1000}} \approx 0.158$$

Для модели:

$$MAE_{\text{model}}(D) = \frac{a}{\sqrt{D}} + b = \frac{5}{\sqrt{1000}} + 0.5 \approx 0.658$$

График ошибок представлен на рисунке 5.6.

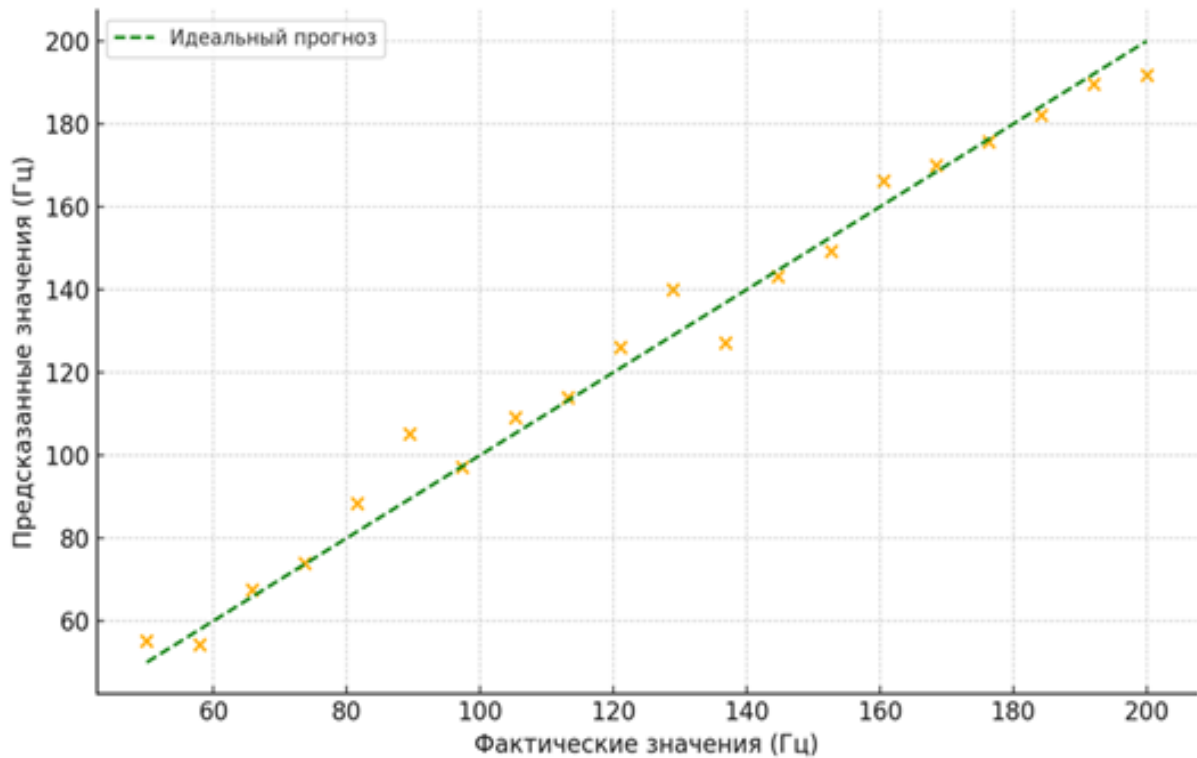


Рисунок 5.6. – График ошибок при использовании программного модуля.

Разработанная математическая модель и программный модуль на основе машинного обучения позволяют сократить затраты времени и ресурсов при проектировании электромеханических преобразователей (Таблица 2). Например, при проведении 1000 измерений, использование программного модуля позволяет ускорить процесс проектирования в 50 раз. В сравнении с традиционными экспериментальными методами, предложенный подход демонстрирует существенное преимущество за счет минимизации количества физических испытаний. Анализ временных затрат показывает, что использование машинного обучения ускоряет процесс разработки в десятки раз, сохраняя при этом высокий уровень точности прогнозирования параметров. С увеличением объема обучающих данных модель становится еще более точной, что делает ее применение перспективным для автоматизации и оптимизации проектирования в данной области. Практическое применение программного модуля позволяет ускорить процесс проектирования преобразователей требуемых частотных характеристик.

5.3 Выводы по Главе 5.

1. Нижняя граница прироста производительности с использованием результатов автоматизированного процесса контроля качества, исходя из собранной статистики, оценивается как $30t$, где t - время, затрачиваемое на проведение одного измерения без средств автоматизации.

2. Использование разработанной модели автоматизации и программного комплекса в действующем производстве, по причине перехода от ручного способа измерений к автоматизированному, сокращение времени при проведении контроля качества многоканальных систем гидрофонов составит 170 часов в год.

3. При использовании модуля автоматизации процесса разработки технических требований к преобразователям сокращение затрат времени и ресурсов при проектировании преобразователей оценивается как $50t$, где t - время, затрачиваемое на проведение одного измерения без системы моделирования. Точность увеличивается при увеличении количества результатов измерений, необходимых для обучения модели.

ЗАКЛЮЧЕНИЕ

Выполненные исследования и практическая работа позволили получить следующие выводы и результаты:

1. Диссертация является законченной научно-квалификационной работой, в которой содержится решение задачи в области развития приборостроения – повышение эффективности контроля качества изделий приборостроения за счет применения автоматизированной системы поканальной обработки данных многоканальных систем преобразователей и автоматизации процесса подготовки производства на стадии разработки технических требований к преобразователям.

2. Проведенный информационно-аналитический обзор позволил выявить взаимосвязи и установить зависимости для автоматизации процесса разработки технических требований к преобразователям.

3. Разработана структурно-параметрическая модель системы автоматизации–процесса поканальной обработки данных многоканальных систем преобразователей для повышения эффективности процесса контроля качества.

4. Разработана математическая модель контроля качества многоканальных систем преобразователей, отображающая поиск АЧХ системы по полезным импульсам входного и измеряемого сигналов, полученных в результате коммутации.

5. Разработана математическая модель системы автоматизации процесса подготовки производства на стадии разработки технических требований к преобразователям для повышения эффективности проектирования изделий приборостроения.

6. Разработан алгоритм автоматизации процесса контроля качества многоканальных систем преобразователей, отображающий решение задач

коммутации сигнала, поиска полезного импульса, а также расчета АЧХ системы.

7. Разработан алгоритм автоматизации процесса разработки технических требований к преобразователям, отображающий решение задачи прогнозирования физических параметров преобразователей по требуемым амплитудно-частотным характеристикам с использованием модели машинного обучения и данных результатов измерений.

8. Использование разработанного на основе соответствующих алгоритмов автоматизированного комплекса контроля качества многоканальных систем преобразователей, позволяет ускорить процесс проведения измерений без потери качества примерно в 30 раз. Использование разработанного программного модуля автоматизации процесса разработки технических требований к преобразователям, позволят ускорить разработку технических требований вдвое. Производительность увеличивается при увеличении количества результатов измерений, необходимых для обучения модели.

9. Разработанная методика и программный модуль автоматизации процесса контроля качества многоканальных систем преобразователей внедрены в учебный процесс кафедры «Автоматизированные системы обработки информации и управления» ФГАОУ ВО «МГТУ «СТАНКИН» при подготовке бакалавров и магистров по направлению 27.03.04 «Управление в технических системах» в рамках учебной дисциплины «Теория управления».

СПИСОК ИСТОЧНИКОВ

1. Стратегия развития электронной промышленности Российской Федерации на период до 2030 года
2. ГОСТ Р МЭК 62127-3-2010
3. ГОСТ Р ГОСТ Р 53188.1-2008(МЭК 61672-1:2002)53566-2009
4. ГОСТ Р 53576—2009 (МЭК 60268-4:2004)
5. ГОСТ 32783— 2014
6. ГОСТ Р ИСО 16063-15-2012
7. ГОСТ Р 59166-2020
8. ГОСТ Р 55205-2012
9. ГОСТ Р МЭК 60770-2-2015
10. ГОСТ Р ИСО/МЭК 19762-4-2011
11. ГОСТ Р ИСО. 9001
12. Основы теории сигналов : учебное пособие / С. И.Кумков, А.В.Цветков ; Министерство науки и высшего образования РФ.— Екатеринбург : Изд-во Урал. ун-та, 2023. — с. 114
13. Овчарук, В. Н. Регистрация и корректировка спектральных функций акустических сигналов многоканальной акустико-эмиссионной системы / В. Н. Овчарук // Информационные технологии XXI века : сборник научных трудов. – Хабаровск : Тихоокеанский государственный университет, 2019. – С. 67-72. – EDN YOVTSN
14. Овчарук, В. Н. Особенности регистрации акустических сигналов в многоканальных акустико-эмиссионных системах / В. Н. Овчарук, В. А. Павлов // Информационные технологии XXI века : Сборник научных трудов / Отв. ред. В.В. Воронин. – Хабаровск : Тихоокеанский государственный университет, 2019. – С. 117-123. – EDN YPQXZZ
15. Пестриков, П. П. Многоканальная измерительная система для регистрации сигналов поверхностной электромиографии мышц предплечья /

П. П. Пестриков, Т. В. Пестрикова // Молодые ученые - Хабаровскому краю : Материалы XXI краевого конкурса молодых ученых и аспирантов, Хабаровск, 15–18 января 2019 года / Редакционная коллегия: И. Н. Пугачев, А. М. Сундуков, А. А. Самойлик. – Хабаровск: Тихоокеанский государственный университет, 2019. – С. 108-113. – EDN ZAEXJR

16. Многоканальная система регистрации и сбора данных диагностик плазмы установки ИТЭР / Д. В. Николаева, П. В. Зубарев, Е. Ю. Миронова [и др.] // XLV Международная Звенигородская конференция по физике плазмы и управляемому термоядерному синтезу : Сборник тезисов докладов, Звенигород, 02–06 апреля 2018 года. – Звенигород: ЗАО НТЦ ПЛАЗМАИОФАН, 2019. – С. 349. – EDN OYDCDD

17. Многоканальная спектрометрическая система регистрации сигналов со стриповых полупроводниковых детекторов / Ю. В. Тубольцев, И. В. Еремин, А. А. Богданов [и др.] // Приборы и техника эксперимента. – 2019. – № 6. – С. 39-45. – DOI 10.1134/S0032816219060168. – EDN LBCCMF

18. Устинов, Д. А. Многоканальная система регистрации рабочей частоты пьезорезонансных датчиков / Д. А. Устинов, Е. Н. Балобанов, А. А. Смирнов // Актуальные научные исследования в современном мире. – 2020. – № 6-1(62). – С. 303-308. – EDN POYJZO

19. Овчарук В.Н., Пурисев Ю.А. Регистрация и обработка акустико-эмиссионной информации в многоканальных системах: монография / В.Н. Овчарук., Ю.А. Пурисев -Хабаровск: Изд-во Тихоокеан. гос. ун-та, 2019. -116 с.

20. Овчарук, В. Н. Применение методов спектрального анализа в многоканальных системах регистрации сигналов акустической эмиссии / В. Н. Овчарук, Чье Ен Ун // Приборы. – 2021. – № 11(257). – С. 35-39. – EDN AZXCZW.

21. Казначеев, П. А. Алгоритм регистрации сейсмоакустических сигналов с многоканальной коммутацией при неизвестном уровне высокочастотных помех / П. А. Казначеев, К. С. Непейна // Современные

техника и технологии в научных исследованиях : Сборник материалов XI Международной конференции молодых ученых и студентов. В 2-х томах, Бишкек, 24–26 апреля 2019 года. – Бишкек: Федеральное государственное бюджетное учреждение науки Научная станция Российской академии наук в г. Бишкеке, 2019. – С. 85-88. – EDN CDHPUS.

22. Козлов, С. В. Алгоритмы обработки сигналов в радиолокационных измерителях угловых координат со сканирующей многоканальной антенной системой / С. В. Козлов, Т. Х. Ву // Журнал радиоэлектроники. – 2019. – № 11. – С. 15. – DOI 10.30898/1684-1719.2019.11.10. – EDN WMRHAU

23. Козлов, С. В. Оценивание угловых координат в обзорных радиолокационных станциях с подсистемами пространственной компенсации помех / С. В. Козлов, Т. Х. Ву // Доклады Белорусского государственного университета информатики и радиоэлектроники. – 2019. – № 4(122). – С. 48-56. – EDN TGWCAS

24. Сергеев, И. Ю. Многоканальная методика обработки сигналов детекторов систем обнаружения радиоактивных источников в движущемся транспорте / И. Ю. Сергеев, Н. П. Валуев, П. М. Юданов // Сибирский пожарно-спасательный вестник. – 2020. – № 2(17). – С. 85-90. – DOI 10.34987/vestnik.sibpsa.2020.17.2.005. – EDN QWREVJ

25. Петешов, А. В. Структура систем корреляционной пространственно-временной обработки сигналов многоканальных РЛС / А. В. Петешов, А. И. Полубехин, В. Л. Румянцев // Известия Тульского государственного университета. Технические науки. – 2020. – № 9. – С. 215-225. – EDN OYQNHW

26. Многоканальная обработка сигналов в пассивно-активных системах радиолокации / С. А. Батчев, А. Б. Талалаев, Д. А. Тимаков, И. В. Чеботарь // Успехи современной радиоэлектроники. – 2022. – Т. 76. – № 4. – С. 14-20. – EDN AXGZZB

27. Левченков, А. Н. Особенности многоканальной обработки сигналов при квазиколецевом расположении приемных элементов Аннотация

/ А. Н. Левченков // Инженерный вестник Дона. – 2020. – № 3(63). – С. 16. – EDN ULASSF

28. Гурский, С. М. Исследование метода пространственно-временной обработки сигналов на фоне нефакторизуемых помех в радиотехнических системах с крупноапертурными антенными решетками / С. М. Гурский, И. В. Кочетков, А. М. Кузинков // Современные наукоемкие технологии. – 2022. – № 5-1. – С. 32-37. – DOI 10.17513/snt.39146. – EDN HBEWIM

29. Аникьев, А. А. Методы оптимизации дальности обнаружения объектов многоканальной оптико-электронной системой / А. А. Аникьев, Н. В. Барышников, В. И. Заварзин // Вестник Московского государственного технического университета им. Н.Э. Баумана. Серия Приборостроение. – 2020. – № 4(133). – С. 134-152. – DOI 10.18698/0236-3933-2020-4-134-152. – EDN RVYHIF

30. Математическое моделирование и алгоритмизация функций мониторинга технологических процессов на основе многоточечных измерительных систем / И. В. Ковалев, Д. И. Ковалев, В. В. Лосев [и др.] // Современные наукоемкие технологии. – 2021. – № 6-1. – С. 29-38. – DOI 10.17513/snt.38693. – EDN ERYLYC

31. Свидетельство о государственной регистрации программы для ЭВМ № 2022662125 Российская Федерация. Встраиваемое программное обеспечение многоканальных адаптивных систем регистрации сигналов : № 2022661768 : заявл. 24.06.2022 : опубл. 29.06.2022 / А. В. Шипатов, А. В. Козаренко ; заявитель Общество с ограниченной ответственностью «Электронные микросистемы». – EDN SGFUQI

32. Свидетельство о государственной регистрации программы для ЭВМ № 2019619334. Специальное (прикладное) программное обеспечение многоканальной системы регистрации звуковых сигналов и радиолокационной информации (Магнитофон "Гранит") (НКПГ.10202-04) : № 2019618260 : заявл. 05.07.2019 : опубл. 16.07.2019 ; заявитель Общество с

ограниченной ответственностью "Фирма "Новые информационные технологии в авиации". – EDN RGHHXY

33. Свидетельство о государственной регистрации программы для ЭВМ № 2018664669 Российская Федерация. SeismoStation - программа для регистрации многоканальных сейсмоакустических данных : № 2018662508 : заявл. 08.11.2018 : опубл. 20.11.2018 / А. Л. Марченко, В. Г. Гайнанов ; заявитель Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный университет имени М.В.Ломоносова» (МГУ). – EDN IGGWBT

34. Патент № 2779148 С1 Российская Федерация, МПК Н04В 7/02. Высокоскоростная многоканальная система передачи данных КВ-диапазона : № 2021119282 : заявл. 18.11.2021 : опубл. 05.09.2022 / П. А. Дикушин, Д. А. Минин, С. А. Помазунов [и др.] ; заявитель Акционерное общество "Научно-исследовательский институт автоматизированных систем и комплексов связи "Нептун". – EDN ANQLLA

35. Свидетельство о государственной регистрации программы для ЭВМ № 2019665701 Российская Федерация. Программа модуля многоканальной обработки биопотенциалов : № 2019663046 : заявл. 09.10.2019 : опубл. 28.11.2019 / И. В. Кречетов, А. А. Скворцов ; заявитель Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский политехнический университет» (Московский Политех). – EDN NNBWRX

36. Свидетельство о государственной регистрации программы для ЭВМ № 2020619202 Российская Федерация. Программное средство для сбора и первичной обработки сигналов с распределенной сети пьезодатчиков "DaCap 1.0" : № 2020618418 : заявл. 30.07.2020 : опубл. 13.08.2020 / И. А. Барейко, А. А. Еремин ; заявитель Федеральное государственное бюджетное образовательное учреждение высшего образования «Кубанский государственный университет» (ФГБОУ ВО «КубГУ»). – EDN CHNBHX

37. Сереброва, Н. С. Адаптивная система управления многоканальным полупроводниковым стабилизатором тока с сигнальной настройкой / Н. С. Сереброва, В. С. Лавриновский // Труды Крыловского государственного научного центра. – 2019. – № S2. – С. 212-218. – DOI 10.24937/2542-2324-2019-2-S-I-212-218. – EDN EWNQNH

38. Дедович, Н. Н. Моделирование элементов конструкции пьезоэлектрического преобразователя методом конечных элементов / Н. Н. Дедович, А. Ф. Романов // Приборостроение - 2024 : материалы 17-й Международной научно-технической конференции, Минск, 26–29 ноября 2024 года. – Минск: Б. и., 2024. – С. 44-46. – EDN AHYEIX.

39. Конечно-элементное моделирование механических напряжений в многослойной диэлектрической мембране для МЭМС-структуры газочувствительного преобразователя / Д. М. Григорьев, С. В. Никифоров, В. В. Кривецкий [и др.] // Наноиндустрия. – 2023. – Т. 16, № S9-2(119). – С. 316-323. – DOI 10.22184/1993-8578.2023.16.9s.316.323. – EDN DGKFMV.

40. Голоколос, Д. А. Моделирование электромеханического преобразователя системой пространственных векторов / Д. А. Голоколос, К. К. Ким, С. Н. Иванов // Молодежь и наука: актуальные проблемы фундаментальных и прикладных исследований : Материалы VI Всероссийской национальной научной конференции молодых учёных. В 3-х частях, Комсомольск-на-Амуре, 10–14 апреля 2023 года. Том Часть 1. – Комсомольск-на-Амуре: Комсомольский-на-Амуре государственный университет, 2023. – С. 315-318. – EDN ZHVPVS.

41. Шпак, А. Н. Моделирование колебаний пьезоэлектрического преобразователя с загнутым электродом с помощью метода спектральных элементов / А. Н. Шпак, С. А. Сиухина, М. В. Голуб // Вестник Южно-Уральского государственного университета. Серия: Математическое моделирование и программирование. – 2022. – Т. 15, № 3. – С. 96-110. – DOI 10.14529/mmp220307. – EDN LDTTHT.

42. Зими́на, К. Д. Математическое моделирование гидроакустического излучателя на основе пьезокерамического пакетного преобразователя / К. Д. Зими́на, В. К. Бахтин, М. С. Дерябин // Экология 2023 - море и человек : Сборник трудов XII Всероссийской научной конференции и молодёжной школы-семинара, Таганрог, 21–23 сентября 2023 года. – Ростов-на-Дону ; Таганрог: Южный федеральный университет, 2023. – С. 104-109. – EDN AESFVZ.

43. Лукоянов, М. Р. прогнозирование ключевых параметров промышленного комплекса с использованием методов градиентного бустинга / М. Р. Лукоянов // Студенческий вестник. – 2025. – № 6-3(339). – С. 66-67. – EDN SRUNDJ.

44. Блехман, И. И. Теория вибрационных процессов и устройств. Вибрационная механика и вибрационная техника / И. И. Блехман. – Санкт-Петербург : Издательский дом "Руда и металлы", 2013. – 640 с. – ISBN 978-5-98191-074-6. – EDN VMPXKD.

45. Петров, Б. М. Электродинамика и распространение радиоволн / Б. М. Петров. Том Часть 1. – Москва : Горячая линия-Телеком, 2017. – 296 с. – ISBN 978-5-9912-0520-7. – EDN NUKCQD.

46. Шлаев, В. И., Бильчук, М. В., Тясто, С. А. Сравнительный анализ инструментов и разработка схемы коммутации для автоматизации процесса регистрации электрических сигналов многоканальных систем / В. И. Шлаев, М. В. Бильчук, С. А. Тясто // Вестник МГТУ «СТАНКИН». – 2021. – №1 (56). С. 13-18.

47. Шлаев, В. И., Бильчук, М. В. Обобщенная модель автоматизации процесса контроля качества изделий приборостроения. / В. И. Шлаев, М. В. Бильчук // Автоматизация и измерения в машино- приборостроении. – 2025. № 4 (32). – С 43-53.

48. Шлаев, В. И. Модуль прогнозирования параметров преобразователей по заданным амплитудно-частотным характеристикам. / В.

И. Шлаев // Известия ЮФУ. Технические науки. – 2025. – № 5 (247). – С. 93-104.

49. Shlaev, V.I., Bilchuk, M.V., Tyasto, S.A. Development of a Switching Circuit for the Operation of a Multichannel System in Reception and Emission Modes / V. I. Shlaev, M. V. Bilchuk, S. A. Tyasto // 2021 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM). – 2021. – Pages: 461-465.

50. Шлаев, В. И., Бильчук, М. В., Тясто, С. А. Программный комплекс для автоматизации процесса регистрации данных многоканальной системы электрических сигналов / В. И. Шлаев, М. В. Бильчук, С. А. Тясто // Самарские чтения. Материалы международной конференции. – Москва: Янус-К, 2022. – С. 246-247.

51. Шлаев, В. И., Бильчук, М. В., Тясто, С. А. Применение автоматизированного комплекса регистрации данных для многоканальной системы электрических сигналов / В. И. Шлаев, М. В. Бильчук, С. А. Тясто // Материалы XIV всероссийской конференции с международным участием МАШИНОСТРОЕНИЕ: традиции и инновации. – Москва, 2021. – С. 372-377.

52. Шлаев, В. И. Программный комплекс для автоматизации процесса регистрации данных многоканальной системы электрических сигналов / В.И. Шлаев // МАТЕРИАЛЫ студенческой научно-практической конференции Автоматизация и информационные технологии. – Москва. – 2021. – № 2. – С. 90.

53. Шлаев, В. И., Бильчук, М. В. Модель автоматизации процесса регистрации и обработки данных многоканальной системы электрических преобразователей / В. И. Шлаев, М. В. Бильчук // МАТЕРИАЛЫ студенческой научно-практической конференции Автоматизация и информационные технологии. – Москва. – 2023. – № 2. – С. 53.

Приложение А. Свидетельство о регистрации ЭВМ.

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2024616183

**Программное обеспечение для автоматизации
обработки и регистрации сигналов многоканальных
систем преобразователей**

Правообладатель: *федеральное государственное бюджетное
образовательное учреждение высшего образования
«Московский государственный технологический
университет «СТАНКИН» (RU)*

Авторы: *Шлаев Виктор Иванович (RU), Бильчук Мария
Викторовна (RU)*

Заявка № 2024615007

Дата поступления 12 марта 2024 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 18 марта 2024 г.



Руководитель Федеральной службы
по интеллектуальной собственности

Ю.С. Зубов

Приложение Б. Акт об использовании результатов работы в производстве.



АКТ

об использовании результатов работы по автоматизации процесса контроля качества многоканальных систем электрических преобразователей

17 января 2024 г.

№ 125

Настоящим актом подтверждается использование результатов работы, выполненной аспирантом Шлаевым В.И. в 2024 г. В производственный процесс АО «АКИН» внедрены:

1. методика автоматизации контроля качества характеристик (АЧХ, ФЧХ, чувствительности по приему, чувствительности по излучению и т.д.) многоканальных систем гидрофонов;
2. справочный лист с рекомендациями по автоматизации контроля качества;
3. диагностический лист для систематизации данных и исключения случайных ошибок при проведении работ.

Решение об использовании результатов работы было принято после анализа экономической эффективности внедрения. По предварительной оценке, сокращение времени при проведении мероприятий контроля качества многоканальных систем электрических преобразователей составит 170 часов в год, по причине перехода от ручного способа измерений к автоматизированному.

Первый заместитель генерального директора

Селянский Р.В.

Заместитель директора филиала
по научно-технической работе



Титаренко Д.В.

Приложение В. Акт внедрения результатов работы в учебный процесс.

МИНОБРНАУКИ РОССИИ
федеральное государственное автономное образовательное учреждение
высшего образования
«Московский государственный технологический университет «СТАНКИН»
(ФГАОУ ВО «МГТУ «СТАНКИН»)

УТВЕРЖДАЮ

И.о. проректора по учебной работе

Д.т.н., доц. Шехтман С.Р.

**АКТ**

внедрения в учебный процесс
результатов диссертационной работы Шлаева Виктора Ивановича
на соискание ученой степени кандидата технических наук

Материалы диссертационной работы аспиранта кафедры «Информационные системы» Шлаева Виктора Ивановича «Повышение эффективности контроля качества изделий на основе автоматизации процесса поканальной обработки сигналов многоканальных систем преобразователей» внедрены в учебный процесс кафедры «Автоматизированные системы обработки информации и управления» ФГАОУ ВО «МГТУ «СТАНКИН» при подготовке бакалавров по направлению 27.03.04 «Управление в технических системах» профиль подготовки «Управление в технических системах» и магистров по направления 27.04.04 «Управление в технических системах» профиль подготовки «Управление в технических системах» в рамках учебной дисциплины «Теория управления».

Заведующий кафедрой АСОИиУ,
д.т.н., профессор

Капитанов А.В.

Приложение Г. Листинг программного комплекса автоматизированного процесса контроля качества многоканальных систем преобразователей.

```
#pragma once
#include <headers.hpp>
#include <map>
struct doManager
{
    uint32_t activChan(t_x502_hnd& hnd, uint32_t chanNum, int method);
    void disable(t_x502_hnd &hnd);
private:
    int getCount(uint32_t& keyNum, int base);
    uint32_t getNumber(uint32_t& ind, int method);
};
#ifndef MAINWINDOW_H
#define MAINWINDOW_H
#include <QMainWindow>
#include <x502manager.h>
#include <QDebug>
#include <QDateTime>
#include <QThread>
#include <memory>
#include <qcustomplot.h>
#include <QKeyEvent>
#include <QShortcut>
#include <x502configurator.h>
#include <painter.h>
QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
```

```

QT_END_NAMESPACE
class MainWindow : public QMainWindow
{
    Q_OBJECT
public:
    MainWindow(QWidget *parent = nullptr);
    ~MainWindow() override;
private slots:
    void search();
    void fillDevNames(std::vector<QString> &devs);
    void applySettings();
    void start();
    void stop();
    void printClip(double ampl);
    void printErr(const char* msg, int err);
private:
    bool isRunning = false;
    std::unique_ptr<Painter> plot;
    std::unique_ptr<QThread> mThr;
    std::unique_ptr<X502Manager> mngr;
    X502Configurator conf;
    Ui::MainWindow *ui;
    QString workDir;
    bool isFirstFile = true;
    void printLog(const QString& msg, bool isErr);
    void fillRadHead(RHead& rHead);
};
#endif // MAINWINDOW_H
#include "manager.h"
Manager::Manager(QObject *parent) : QObject(parent)

```

```

{
}
Manager::Manager(std::vector<hnd> &_devs, rHead rh)
{
}
hnd Manager::master() const
{
    return devs[0];
}
hnd Manager::slave() const
{
    return devs[1];
}
double Manager::fGenSin(int n, int freq, double amp, double offset, double phase,
double freqMod) const
{
    const double PI = 3.14159265358979323846;
    return amp*sin(2*PI*n*freq/freqMod +phase)+offset;
}
QVector<double>      Manager::mergeImp(QVector<double>      &buf1,
QVector<double> &buf2)
{
    QVector<double> res;
    for(int i = 0; i < buf1.size (); ++i){
        res.append (buf1[i]);
        res.append (buf2[i]);
    }
    return res;
}
#endif MANAGER_H

```

```

#define MANAGER_H
#include <QObject>
#include <vector>
#include <headers.hpp>
#include <math.h>
class Manager : public QObject
{
    Q_OBJECT
public:
    explicit Manager(QObject *parent = nullptr);
    Manager(std::vector<hnd> &_devs, rHead rh);
    virtual void start();
    virtual void stop();
    virtual ~Manager();
private:
    std::vector<hnd> devs;
protected:
    virtual void createSign (std::vector<double> &imp,int freq);
    virtual int dataCollection(hnd & dev,QVector<double> &adcBuf, size_t fullSize);
    hnd master() const;
    hnd slave() const;
    double fGenSin(int n, int freq, double amp, double offset, double phase, double
freqMod) const;
    QVector<double> mergeImp(QVector<double> & buf1, QVector<double>&
buf2);
signals:
};
#endif // MANAGER_H
#include "painter.h"
#include <QColor>

```

```

#include <array>
Painter::Painter(QCustomPlot *plt,QWidget *parent):QWidget (parent)
{
    plot =static_cast<std::unique_ptr<QCustomPlot>>(plt);
    init();
}
Painter::~Painter()
{
}
void Painter::init()
{
    //CTRL + Plus
    keyCtrlPlus =std::make_unique<QShortcut>(plot.get ());
    keyCtrlPlus->setKey (static_cast<Qt::Key>(Qt::CTRL)+ Qt::Key_Plus);
    connect (keyCtrlPlus.get (), &QShortcut::activated, this, [=]() {
        plot->yAxis->scaleRange(.50,plot->yAxis->range().center());
        plot->replot();
    });
    //CTRL + Minus
    keyCtrlMinus = std::make_unique<QShortcut>(plot.get ());
    keyCtrlMinus->setKey (static_cast<Qt::Key>(Qt::CTRL) + Qt::Key_Minus);
    connect (keyCtrlMinus.get(), &QShortcut::activated, this, [=]() {
        plot->yAxis->scaleRange(1/.50,plot->yAxis->range().center());
        plot->replot();
    });
    //Shift + Plus
    keyShiftPlus = std::make_unique<QShortcut>(plot.get ());
    keyShiftPlus->setKey (static_cast<Qt::Key>(Qt::SHIFT) + Qt::Key_Plus);
    connect (keyShiftPlus.get(), &QShortcut::activated, this, [=]() {
        plot->xAxis->scaleRange(.50,plot->xAxis->range().center());
    });
}

```

```

    plot->replot();
});
//Shift + Minus
keyShiftMinus= std::make_unique<QShortcut>(plot.get ());
keyShiftMinus->setKey (static_cast<Qt::Key>(Qt::SHIFT) + Qt::Key_Minus);
connect (keyShiftMinus.get(), &QShortcut::activated, this, [=]() {
    plot->xAxis->scaleRange(1/.50,plot->xAxis->range().center());
    plot->replot();
});
//Up
keyUp = std::make_unique<QShortcut>(plot.get ());
keyUp->setKey (Qt::Key_Up);
connect (keyUp.get(), &QShortcut::activated, this, [=]() {
    if(plot->yAxis->range ().upper <= 20){
        double step =plot->yAxis->range ().size ()/3.;
        plot->yAxis->setRange ( plot->yAxis->range ().lower + step,
                               plot->yAxis->range ().upper + step);
        plot->replot ();
    }
});
//Right
keyRight = std::make_unique<QShortcut>(plot.get ());
keyRight->setKey (Qt::Key_Right);
connect (keyRight.get(), &QShortcut::activated, this, [=]() {
    if(plot->xAxis->range ().upper <= 200){
        double step =plot->xAxis->range ().size ()/3.;
        plot->xAxis->setRange ( plot->xAxis->range ().lower + step,
                               plot->xAxis->range ().upper + step);
        plot->replot ();
    }
}

```

```

});
//Down
keyDown = std::make_unique<QShortcut>(plot.get ());
keyDown->setKey (Qt::Key_Down);
connect (keyDown.get(), &QShortcut::activated, this, [=]() {
    if(plot->yAxis->range ().lower >= -20){
        double step = plot->yAxis->range ().size ()/3.;
        plot->yAxis->setRange (plot->yAxis->range ().lower - step,
                               plot->yAxis->range ().upper - step);
        plot->replot ();
    }
});
//Left
keyLeft = std::make_unique<QShortcut>(plot.get ());
keyLeft->setKey (Qt::Key_Left);
connect (keyLeft.get(), &QShortcut::activated, this, [=]() {
    if(plot->xAxis->range ().lower >0 ) {
        double step =plot->xAxis->range ().size ()/3.;
        plot->xAxis->setRange ( plot->xAxis->range ().lower - step,
                               plot->xAxis->range ().upper - step);
        plot->replot ();
    }
});
plot->setInteractions (QCP::iRangeZoom | QCP::iRangeDrag );
plot->axisRect()->setRangeDrag(Qt::Horizontal | Qt::Vertical);
plot->axisRect()->setRangeZoom(Qt::Horizontal);
plot->xAxis->setTickLabelFont(QFont(QFont().family(), 8));
plot->yAxis->setTickLabelFont(QFont(QFont().family(), 8));
plot->axisRect()->insetLayout()->setInsetAlignment(0,
Qt::AlignLeft|Qt::AlignTop);

```

```

}
void Painter::sendBufDrawVec(const QVector<double> &adcData)
{
    std::array<Qt::GlobalColor, 5> colors{Qt::darkRed, Qt::darkYellow,
Qt::darkBlue, Qt::darkGreen, Qt::black};
    int chCnt = 2;
    auto minMax = std::minmax_element(adcData.begin (), adcData.end ());
    if(abs (*minMax.first) > abs(*minMax.second))
        emit sendAmpl (*minMax.first);
    else
        emit sendAmpl (*minMax.second);
    QVector <QVector <double>> u (chCnt);
    for(auto i = 0; i < adcData.size (); ++i){
        u[i%chCnt].append (adcData[i]);
    }
    QVector <double> t;
    for(int i = 0; i < adcData.size()/chCnt; ++i){
        t.append ( i/freqD);
    }
    plot->legend->setVisible (true);
    plot->clearGraphs ();
    for(int i = 0; i < chCnt; ++i){
        plot->addGraph ();
        plot->graph (i)->setData (t, u[i]);
        plot->graph (i)->setPen (QColor(colors[static_cast<size_t>(i)]));
        plot->graph (i)->setLineStyle (QCPGraph::lsLine);
        if(i == 0)
            plot->graph (i)->setName ("Канал 1");
        if(i == 1)
            plot->graph (i)->setName ("Канал 2");
    }
}

```

```

    }
    plot->xAxis->setLabel ("Время,");
    plot->yAxis->setLabel ("Амплитуда,");
    plot->replot ();
}
#ifdef PAINTER_H
#define PAINTER_H
#include <QObject>
#include <qcustomplot.h>
#include<memory>
class Painter: public QWidget
{
    Q_OBJECT
public:
    Painter (QCustomPlot *plt, QWidget *parent = nullptr);
    ~Painter();
    void init();
private:
    std::unique_ptr<QCustomPlot> plot;
    std::unique_ptr<QShortcut> keyCtrlPlus ;
    std::unique_ptr<QShortcut> keyCtrlMinus;
    std::unique_ptr<QShortcut> keyShiftPlus;
    std::unique_ptr<QShortcut> keyShiftMinus;
    std::unique_ptr<QShortcut> keyUp;
    std::unique_ptr<QShortcut> keyRight;
    std::unique_ptr<QShortcut> keyDown;
    std::unique_ptr<QShortcut> keyLeft;
public slots:
    void sendBufDrawVec(const QVector<double> &adcData);
signals:

```

```

void sendAmpl(double ampl );
};
#endif // PAINTER_H
#include <Vfft.h>
#include<QDebug>
double myFFT::processHilbert(complexArray &arr)
{
    return Hilbert (arr);
}
double myFFT::processSpectrum(complexArray &arr, double freq, double Fd,
size_t kNf, size_t chanNum, complexArray& resArr)
{
    size_t nt= arr.size ();
    size_t nf = getDesLen (arr.size ());
    arr.resize (nf);
    fft(arr, false);
    std::vector<double> absF (nf/2);
    std::transform(arr.begin      (),      arr.begin      ()+nf/2,      absF.begin(),
[=](std::complex<double> val){
    return abs(val)/nt;
});
    int ng = static_cast<int>(round (nf* freq/Fd));
    int delN = static_cast<int>(round (nf/Fd));
    resArr.resize (arr.size ());
    std::copy(arr.begin (), arr.end(), resArr.begin ());
    size_t nMax =static_cast<size_t>(std::distance(absF.begin (),
std::max_element(absF.begin      ()+ng-delN,      absF.begin
()+ng+delN+1)));
    if(chanNum ==0)
        freq = nMax*Fd/nf;
}

```

```

return *std::max_element(absF.begin ()+ng-delN, absF.begin ()+ng+delN+1);
}
std::complex<double> myFFT::processPhase(complexArray arr, double freq,
double Fd, size_t kNf, complexArray &resArr)
{
size_t nt= arr.size ();
size_t nf = getDesLen (arr.size ())*1;
arr.resize (nf);
fft(arr, false);
std::vector<double> absF (nf/2);
std::transform(arr.begin (), arr.begin ()+nf/2, absF.begin(),
[=](std::complex<double> val){
return 2*abs(val)/nt;
});
int ng = static_cast<int>(round (nf* freq/Fd));
int delN= static_cast<int>(round (nf* 500/Fd));
size_t newSize = static_cast<size_t>(std::distance(arr.begin ()+ng-delN, arr.begin
()+ng+delN+1));
resArr.resize (newSize);
resArr.resize (arr.size ());
std::copy(arr.begin (), arr.end(), resArr.begin ());
return arr[static_cast<size_t>(ng)];
}
std::complex<double> myFFT::process(complexArray &arr, double freq, double
Fd, size_t kNf, complexArray &resArr)
{
resArr.clear ();
size_t nt= arr.size ();
size_t nf = getDesLen (arr.size ())*kNf;
size_t ng = static_cast<size_t>(round (nf* freq/Fd));

```

```

arr.resize (nf);
fft(arr, false);
std::vector<double> absF (nf/2);
std::transform(arr.begin (), arr.begin ()+nf/2, arr.begin(),
[=](std::complex<double> val){
    return 2.*val/static_cast<double>(nt);
});
std::copy(arr.begin (), arr.begin ()+nf/2-1,std::back_inserter(resArr));
return arr[ng];
}
void myFFT::fft(complexArray &arr, bool inv)
{
    const double Pi = 3.141592653589793238462643383279;
    size_t n = arr.size ();
    if (n==1) return;
    complexArray a0(n/2), a1(n/2);
    for(size_t i = 0, j = 0; i < n; i+=2, ++j){
        a0[j]=arr[i];
        a1[j]=arr[i+1];
    }
    fft(a0, inv);
    fft(a1, inv);
    double ang = 2*Pi/n*(inv ? 1 : -1);
    std::complex<double> w(1), wn(cos(ang), sin(ang));
    for (size_t i = 0; i < n/2; ++i){
        arr[i] = a0[i] + w* a1[i];
        arr[i+n/2] = a0[i] - w*a1[i];
        if(inv){
            arr[i] /=2;
            arr[i+n/2] /=2;
        }
    }
}

```

```

    }
    w*=wn;
}
}
double myFFT::Hilbert(complexArray &arr)
{
    size_t nt = arr.size ();
    size_t nf = getDesLen (arr.size());
    arr.resize(nf);
    fft(arr, false);
    for(size_t i = 1; i < nf/2; ++i) arr[i] *= 2;
    for(size_t i = nf/2+1; i < nf; ++i) arr[i] = {0 ,0};
    fft(arr, true);
    std::vector<double> mods (nt);
    for(size_t i = 0; i < nt; ++i){
        mods[i] = abs(arr[i]);
    }
    return getAmpl (mods);
}
double myFFT::getAmpl(std::vector<double> &mods)
{
    double sum = std::accumulate(mods.begin (), mods.end (),0.0);
    return sum/mods.size ();
}
size_t myFFT::getDesLen(size_t arrSize)
{
    size_t n = 1;
    while(n<arrSize) n <<=1;
    return n;
}

```

```
}  
#pragma once  
#include <QMainWindow>  
#include <QDebug>  
#include <QFile>  
#include <QDir>  
#include <QDataStream>  
#include <QFileDialog>  
//stl and my  
#include <map>  
#include <functional>  
#include <complex>  
#include <vector>  
#include <algorithm>  
#include <QFile>  
#include <QTextStream>  
struct myFFT{  
    double processHilbert(complexArray &arr);  
    double processSpectrum(complexArray& arr, double freq, double Fd, size_t kNf,  
size_t chanNum, complexArray& resArr);  
    std::complex<double> processPhase(complexArray arr, double freq, double Fd,  
size_t kNf,complexArray& resArr);  
    std::complex<double> process(complexArray &arr, double freq, double Fd, size_t  
kNf, complexArray& resArr);  
    void test();  
private:  
    size_t getDesLen(size_t arrSize);  
    void fft( complexArray& arr, bool inv);  
    double Hilbert (complexArray& arr);  
    double getAmpl(std::vector<double>& mods);
```

```
void writeArr(const complexArray& arr, QString path);
```

```
void writeVec(const std::vector<double>& arr, QString path);
```

```
};
```

```
#include "worker.h"
```

```
#include <QThread>
```

```
#include <fstream>
```

```
#include<iostream>
```

```
Worker::Worker(QObject *parent) : QObject(parent){ }
```

```
Worker::Worker(const std::vector<QString>& paths , QString wDir, int b, int e,int  
delta, int metNum, size_t koef ): methodNum(metNum), fPaths (paths){
```

```
    stepToBegin =b;
```

```
    stepToEnd = e;
```

```
    this->delta = delta;
```

```
    isStopped = false;
```

```
    dirPath = wDir;
```

```
    kNf = static_cast<size_t>(pow(2, koef));
```

```
    qRegisterMetaType<complexArray>("complexArray");
```

```
qRegisterMetaType<std::vector<complexArray>>("std::vector<complexArray>");
```

```
qRegisterMetaType<std::map<double,std::vector<double>>>("std::map<double,st  
d::vector<double>>");
```

```
    qRegisterMetaType<size_t>("size_t");
```

```
}
```

```
Worker::Worker(const QString &path, int b, int e,int delta, size_t koef)
```

```
{
```

```
    isStopped = false;
```

```
    stepToBegin =b;
```

```
    stepToEnd = e;
```

```

this->delta = delta;
fPath = path;
kNf =static_cast<size_t>(pow(2, koef));
qRegisterMetaType<complexArray>("complexArray");

qRegisterMetaType<std::vector<complexArray>>("std::vector<complexArray>");

qRegisterMetaType<std::map<double,std::vector<double>>>("std::map<double,st
d::vector<double>>");
    qRegisterMetaType<size_t>("size_t");
}
void Worker::stop()
{
    isStopped = true;
}
void Worker::startWorker()
{
    std::vector<QString> paths = fPaths;
    emit sendFileCnt (paths.size ());
    myFFT worker;
    std::vector<double> amps;
    complexArray phases;
    std::map<double, std::vector<double>> listFA;
    double freq = 0;
    double fd =0;
    size_t fNum =1;
    std::vector<complexArray> signs ;
    complexArray arrForPrint;
    for(const auto& path: paths){
        if(!isStopped){

```

```

emit sendCurrentFileNum (fNum);
amps.clear ();
signs = getSignalFromFile (path, freq,fd);
for(size_t i =0; i < signs.size (); ++i){
    phases.emplace_back(worker.processPhase (signs[i], freq, fd, kNf,
arrForPrint));
    emit sendData (arrForPrint, freq);
    if(methodNum == 0){
        amps.emplace_back(worker.processSpectrum (signs[i], freq, fd, kNf, i
,arrForPrint));
        emit sendData (arrForPrint, freq);}
    if(methodNum ==1)
        amps.emplace_back(worker.processHilbert (signs[i]));
    }
double ampl =amps[1]/amps[0];
std::complex<double> phDiv =phases[1]/phases[0];
double phase = arg (phDiv);
emit sendInfo (freq, ampl, phase);
listFA[freq].emplace_back(ampl);
listFA[freq].emplace_back(phase);
fNum++;
}else{
    emit finished ();
}
}
emit sendTotal (listFA);
emit finished ();
writeFile (dirPath, listFA);
}
void Worker::startOne()

```

```

{
  QString path = fPath;
  myFFT worker;
  std::vector<double> amps;
  complexArray phases;
  std::map<double, std::vector<double>> listFA;
  double freq = 0;
  double fd =0;
  size_t fNum =1;
  std::vector<complexArray> signs ;
  std::vector<complexArray> dataForPrint (2);
  complexArray arrForPrint;
  amps.clear ();
  signs = getSignalFromFile (path, freq,fd);
  methodNum = 0;
  for(size_t i =0; i < signs.size (); ++i){
    phases.emplace_back(worker.processPhase (signs[i], freq, fd, kNf,
dataForPrint[0]));
    if(methodNum == 0){
      amps.emplace_back(worker.processSpectrum (signs[i], freq, fd, kNf, i
,dataForPrint[1]));
      if(methodNum ==1)
        amps.emplace_back(worker.processHilbert (signs[i]));
    }
    emit sendAllData (dataForPrint, freq);
  }
  double ampl =amps[1]/amps[0];
  std::complex<double> phDiv =phases[1]/phases[0];
  double phase = arg (phDiv);
  emit sendInfo (freq, ampl, phase);
  emit finished ();
}

```

```

}
void Worker::startNewWorker()
{
    QString path = fPath;
    myFFT worker;
    std::vector<double> amps;
    complexArray phases;
    std::map<double, std::vector<double>> listFA;
    double freq = 0;
    double fd =0;
    size_t fNum =2;
    std::vector<complexArray> signs ;
    std::vector<complexArray> dataForPrint (2);
    complexArray arrForPrint;
    amps.clear ();
    signs = getSignalFromFile (path, freq,fd);
    methodNum = 0;
    signs = getSignalFromFile (path, freq,fd);
    std::complex<double>  sf1  =  worker.process  (signs[0],  freq,  fd,kNf,
dataForPrint[0]);
    std::complex<double>  sf2  =  worker.process  (signs[1],  freq,  fd,kNf,
dataForPrint[1]);
    std::complex<double> Kf = sf2/sf1;
    double ampl =abs(Kf);
    double phase = arg (Kf);
    emit sendInfo (freq, ampl, phase);
    listFA[freq].emplace_back(ampl);
    listFA[freq].emplace_back(phase);
    emit sendAllCData (dataForPrint, freq, fd);
    writeFile (dirPath, listFA);

```

```

    emit finished ();
}
void Worker::startAllWorker()
{
    std::vector<QString> paths = fPaths;
    myFFT worker;
    double freq = 0;
    int keyFreq = 0;
    double fd =0;
    double fNum =1;
    std::vector<complexArray> dataForPrint (2);
    std::map<double, std::vector<double>> listFA;
    std::vector<complexArray> signs ;

    for(const auto& path: paths){
        if(!isStopped){
            methodNum = 0;
            signs = getSignalFromFile (path, freq,fd);
            keyFreq = static_cast<int>(freq);
            std::complex<double> sf1 = worker.process (signs[0], freq, fd,kNf,
dataForPrint[0]);
            std::complex<double> sf2 = worker.process (signs[1], freq, fd,kNf,
dataForPrint[1]);
            std::complex<double> Kf = sf2/sf1;
            double ampl =abs(Kf);
            double phase = arg (Kf);
            double gamma = ampl*(g[keyFreq])/(us[keyFreq]);
            emit sendInfo (freq, ampl, phase);
            listFA[freq].emplace_back(gamma);
            emit sendAllCData (dataForPrint, freq, fd);
        }
    }
}

```

```

        fNum++;
    }
    else{
        emit finished ();
    }
}
emit sendTotal (listFA);
emit finished ();
writeFile (dirPath, listFA);
}
double Worker::getAmpl(std::vector<complexArray> &arr)
{
    std::vector<double> temp;
    for(const auto& v: arr[0]){
        temp.push_back (v.real());
    }
    double min = *std::min_element(temp.begin (), temp.end ());
    double max = *std::max_element(temp.begin (), temp.end ());
    return (max-min)/2.;
}
std::vector<QString> Worker::getPaths(const QString &path)
{
    QDir dir(path);
    std::vector<QString> paths;
    if(dir.exists ()){
        dir.setSorting (QDir::Name);
        QFileInfoList fList = dir.entryInfoList ();
        for(int i = 0; i < fList.size (); ++i){
            paths.emplace_back(fList.at (i).filePath ());
        }
    }
}

```

```

    }
    return paths;
}
std::vector<complexArray> Worker::getSignalFromFile(const QString &path,
double &freq, double &fd)
{
    QFile fIn(path);
    QByteArray hd;
    QByteArray fullData;
    int headSize = sizeof(InputFileHead);
    hd.reserve (headSize);
    if(fIn.open(QIODevice::ReadOnly)){
        hd = fIn.read (headSize);
        fullData = fIn.readAll();
        fIn.close ();
    }
    InputFileHead* fHead = reinterpret_cast<InputFileHead*>(&hd.data ()[0]);
    freq = fHead->freqSign;
    fd = fHead->freqChan;
    std::vector<complexArray> temp(2);
    std::vector<complexArray> signs(2);
    complexArray sOne;
    complexArray sTwo;
    QDataStream ds(fullData);
    ds.setByteOrder (ds.LittleEndian);
    double a;
    int cntr = 0;
    while (!ds.atEnd ()) {
        ds>>a;
        if(cntr%2 ==0)

```

```

    sOne.emplace_back(a,0);
else
    sTwo.emplace_back(a,0);
++cntr;
}
auto b= sOne.begin ()+stepToBegin-delta;
auto e= sOne.begin ()+stepToEnd-delta;
std::copy(b,e,std::back_inserter(temp[0]));
b= sTwo.begin ()+stepToBegin;
e= sTwo.begin ()+stepToEnd;
std::copy(b,e,std::back_inserter(temp[1]));
std::complex<double> t1 ;
for(const auto& t: temp[1]){
    t1+= t;
}
t1=t1/static_cast<double>(temp[1].size ());
for(auto& t: temp[1]){
    t-=t1;
}
return temp;
}
std::map<int, double> Worker::getKoeffsFromFile(const QString &path)
{
    QFile file(path);
    std::map<int,double> res;
    if(file.open (QIODevice::ReadOnly)){
        QTextStream ts(&file);
        QString line;
        while(!ts.atEnd ()){
            line = ts.readLine ();

```

```

        res[line.split ("\t").at(0).toInt ()] = line.split ("\t").at(1).toDouble ();
    }
    file.close ();
}
return res;
}
#endif WORKER_H
#define WORKER_H
#include <QObject>
#include <Vfft.h>
class Worker : public QObject
{
    Q_OBJECT
public:
    explicit Worker(QObject *parent = nullptr);
    explicit Worker (const std::vector<QString> &paths,QString wDir, int b, int e,int
delta, int metNum = 0, size_t koef = 0);
    Worker(const QString& path, int b, int e, int delta, size_t koef = 0);
    void stop();
public slots:
    void startWorker();
    void startOne();
    void startNewWorker();
    void startAllWorker();
private:
    int methodNum;
    size_t kNf;
    std::vector<QString> fPaths;
    QString dirPath;
    QString fPath;

```

```

bool isStopped;
int stepToBegin, stepToEnd, delta;
std::vector<complexArray> signs;
double getAmpl(std::vector<complexArray>& arr);
std::vector<QString> getPaths(const QString& path);
std::vector<complexArray> getSignalFromFile(const QString &path, double
&freq, double &fd);
void writeFile(QString dir, const std::map<double, std::vector<double> > &data);
void printLog (QString msg, QString log);
std::map<int, double> getKoefsFromFile(const QString& path);
signals:
void sendFileCnt(size_t cnt);
void sendCurrentFileNum(size_t num);
void sendInfo(double freq, double ampl, double phase);
void sendCDData(complexArray arr, double freq, double Fd);
void sendData(complexArray arr, double freq);
void sendAllData(std::vector<complexArray> arr, double freq);
void sendAllCDData(std::vector<complexArray> arr, double freq, double fd);
void sendTotal(std::map<double, std::vector<double>> res);
void finished();
};
#endif // WORKER_H
#include "x502configurator.h"

X502Configurator::X502Configurator(QObject *parent) : QObject(parent)
{}
void X502Configurator::search()
{
for(size_t i = 0; i < devs.size(); ++i){
X502_Free (devs[i]);

```

```

    }
    devs.clear ();
    devRecs.clear ();
    t_e502_eth_svc_browse_hnd *pContext = new t_e502_eth_svc_browse_hnd;
    t_e502_eth_svc_record_hnd *svc = new t_e502_eth_svc_record_hnd;
    t_x502_devrec *devRec = new t_x502_devrec;
    int err = E502_EthSvcBrowseStart (pContext, 0);
    std::vector<QString> devNames;
    if(err == X502_ERR_OK){
        uint32_t event = 1;
        while(err == X502_ERR_OK){
            err = E502_EthSvcBrowseGetEvent (*pContext, svc, &event, nullptr, 1000);
            if(!event)
                break;
            err = E502_MakeDevRecordByEthSvc (devRec, *svc, 0, 1000);
            devRecs.emplace_back(*devRec);
        }
        size_t findDevCnt = devRecs.size ();
        for(size_t i = 0; i < findDevCnt; ++i){
            hnd hnd = nullptr;
            hnd = X502_Create ();
            X502_OpenByDevRecord (hnd, &devRecs[i]);
            devNames.push_back (devRecs[i].location);
            devs.emplace_back(hnd);
        }
        if (findDevCnt !=0){
            emit sendDevNames (devNames);
            E502_EthSvcRecordFree (*svc);
        }
        err = E502_EthSvcBrowseStop (*pContext);
    }

```

```

    delete devRec;
    delete svc;
    delete pContext;
}
}
int X502Configurator::setConfigure(size_t masterInd, RHead rh, size_t mode)
{
    for(size_t i = 0; i < devs.size (); ++i){
        X502_OpenByDevRecord (devs[i], &devRecs[i]);
    }
    rHead = rh;
    nMaster = masterInd;
    int err = 0;
    err = confMaster ();
    if(err!=0) return err;
    err = confSlave ((mode+1)%2);
    return err;
}
std::vector<hnd>& X502Configurator::getDevs()
{
    outDevs.push_back (master ());
    outDevs.push_back (slave());
    return outDevs;
}
void X502Configurator::setRadParams(RHead rh)
{
    rHead = rh;
}
void X502Configurator::exit()
{

```

```

for(auto dev: devs){
    X502_Close (dev);
    X502_Free (dev);
}
}
void X502Configurator::close()
{
    for(auto dev: devs){
        X502_Close (dev);
    }
}
int X502Configurator::confMaster()
{
    uint32_t chanCnt = 1;
    uint32_t mode = 0;
    uint32_t range = rHead.range;
    double freqMod = 1e6;
    double freqCadre = 0;
    hnd dev = master();
    int err= X502_SetLChannelCount (dev, chanCnt);
    if (err != 0) return err;
    err = X502_SetLChannel (master(),
        0 ,
        0 ,
        mode,
        range,
        1);
    if(err != 0) return err;
    err = X502_SetAdcFreq (dev, &freqMod, &freqCadre);
    if(err != 0) return err;
}

```

```

err = X502_SetSyncMode (dev, X502_SYNC_INTERNAL);
if(err !=0) return err;
err = X502_SetSyncStartMode (dev, X502_SYNC_INTERNAL);
if(err !=0) return err;
err = X502_SetRefFreq (dev, X502_REF_FREQ_1500KHZ);
if(err !=0) return err;
err = X502_StreamsEnable (dev, X502_STREAM_ADC |
X502_STREAM_DAC1);
if(err !=0) return err;
if(err !=0) return err;
err = X502_Configure (dev, 0);
return err;
}
int X502Configurator::confSlave(uint32_t chanNum)
{
int err = 0;
uint32_t chanCnt = 1;
uint32_t mode = 0;
uint32_t range = rHead.range;
double freqCadre = 0;
double freqAdc = 1e6;
hnd dev = slave();
err = X502_SetLChannelCount (dev,chanCnt);
if (err != 0) return err;
err = X502_SetLChannel (dev,
0,
chanNum,
mode,
range,
1);

```

```

if(err != 0) return err;
err = X502_SetAdcFreq (dev,
                    &freqAdc,
                    &freqCadre);
if(err !=0) return err;
err = X502_SetSyncMode (dev, X502_SYNC_EXTERNAL_MASTER);
if(err !=0) return err;
err = X502_SetSyncStartMode (dev, X502_SYNC_EXTERNAL_MASTER);
if(err !=0) return err;
err = X502_SetRefFreq (dev, X502_REF_FREQ_1500KHZ);
if(err !=0) return err;
err = X502_StreamsEnable (dev, X502_STREAM_ADC);
if(err !=0) return err;
err = X502_SetDigInPullup (dev, X502_PULLDOWN_CONV_IN |
X502_PULLDOWN_START_IN);
if(err !=0) return err;
err = X502_Configure (dev, 0);
return err;
}
hnd X502Configurator::slave()
{
    return devs[1-nMaster];
}
hnd X502Configurator::master()
{
    return devs[nMaster];
}
#pragma once
#include <headers.hpp>
//stl

```

```

#include <vector>
#include <string>
#include <math.h>
//QT
#include <QObject>
class X502Configurator : public QObject
{
    Q_OBJECT
public:
    explicit X502Configurator( QObject *parent = nullptr);
    void search();
    int setConfigure(size_t masterInd, RHead rh ,size_t mode = 0);
    std::vector<hnd>& getDevs();
    void setRadParams(RHead rh);
    void exit();
    hnd slave() ;
    hnd master() ;
public slots:
    void close();
private:
    std::vector<hnd> devs;
    std::vector<hnd> outDevs;
    std::vector<t_x502_devrec> devRecs;
    size_t nMaster;
    RHead rHead;
    int confMaster();
    int confSlave(uint32_t chanNum);

signals:
    void sendDevNames( std::vector<QString>& devs);

```

```
};  
#pragma once  
#include <do_manager.h>  
#include<thread>  
#include <vector>  
#include <string>  
#include <math.h>  
//QT  
#include <QVector>  
#include <QFile>  
#include <QTextStream>  
#include <QDebug>  
#include <QObject>  
#include <QDir>  
#include <QDateTime>  
#include <QDataStream>  
#include <QThread>  
class X502Manager: public QObject  
{  
    Q_OBJECT  
public:  
    explicit X502Manager(QObject *parent = nullptr);  
    X502Manager(std::vector<hnd> &_devs, RHead rh);  
public slots:  
    void start();  
    void stop();  
private:  
    bool isStopped = false;  
    std::vector<hnd> devs;
```

```

hnd slave() const;
hnd master() const;
double fGenSin(int n, int freq, double amp, double offset, double phase, double
freqMod);
void createSign (std::vector<double> &imp,int freq, int method=0);
    RHead rHead;
void writeFile(const QVector<double> &vec, uint32_t chanNum, int freq);
int cycleStart(const std::vector<double> &imp, int method = 0);
const char* chErr(int err);
int dataCollection( hnd & dev,QVector<double> &adcBuf, size_t fullSize);
QVector<double> mergeImp(QVector<double> & buf1, QVector<double>&
buf2);
signals:
void sendDevNames( std::vector<QString>& devs);
void sendData(const QVector<double>& adcData);
void finished();
void sendMsg(const QString& msg, bool isErr);
void sendCurrentData(const QString& msg);
void sendErr(const char* log, int err);
};
#include "x502manager.h"
X502Manager::X502Manager(QObject *parent):QObject (parent)
{
    qRegisterMetaType<QVector<double>>("QVector<double>");
}
X502Manager::X502Manager(std::vector<hnd>& _devs, RHead rh)
{
    qRegisterMetaType<QVector<double>>("QVector<double>");
    devs = _devs;
    rHead = rh;
}

```

```

}
void X502Manager::start()
{
    hnd m = master ();
    int err =0;
    hnd s = slave ();
    X502_StreamsEnable (m, X502_STREAM_ADC | X502_STREAM_DAC1);
    X502_StreamsEnable (s, X502_STREAM_ADC );
    size_t fullSize = rHead.bufSize/2;
    std::vector<uint32_t> recvBuf(fullSize);
    QVector <double> adcBufMaster(static_cast<int>(fullSize));
    QVector <double> adcBufSlave(static_cast<int>(fullSize));
    std::vector<double> imp(fullSize);
    doManager doWorker;
    uint32_t ch = rHead.chanNum;
    while (ch <= static_cast<uint32_t>(rHead.chanCnt) && !isStopped){
        uint32_t activChan = doWorker.activChan (s, ch, rHead.method);
        Q_UNUSED (activChan)
        int freq = rHead.freqBegin;
        while(freq<= rHead.freqEnd && !isStopped){
            createSign (imp, freq, 1);
            err = cycleStart (imp, 1);
            if(err != 0) {
                X502_StreamsStop (s);
                X502_StreamsStop (m);
                emit sendErr (chErr (err), err);
                doWorker.disable (s);
                emit finished ();
                break;
            }
        }
    }
}

```

```

std::thread thr1 ([&](){
    err = dataCollection (s ,adcBufSlave,recvBuf.size ());
});
std::thread thr2 ([&](){
    err = dataCollection (m ,adcBufMaster,recvBuf.size ());
});
if(err != 0) {
    emit sendErr (chErr (err), err);
    X502_StreamsStop (m);
    X502_StreamsStop (s);
    doWorker.disable (s);
    emit finished ();
    break;
}
if(!adcBufMaster.isEmpty ()){
    QVector <double> res = mergeImp (adcBufMaster, adcBufSlave);
    if(!rHead.isOsc) writeFile (res, ch, freq);
    emit sendData (res);
}
if(!rHead.isOsc) freq += rHead.freqStep;
}
if(!rHead.isOsc) ch ++;
}
X502_StreamsStop (m);
X502_StreamsStop (s);
emit finished ();
}
void X502Manager::stop()
{
    isStopped = true;

```

```

X502_StreamsStop (master ());
X502_StreamsStop (slave ());
}
int X502Manager::cycleStart(const std::vector<double>& imp, int method )
{
    hnd dev = master();
    hnd s = slave ();
    std::vector<uint32_t> outBuf(imp.size());
    if(X502_IsRunning (dev)== X502_ERR_OK){
        X502_StreamsStop (dev);
    }
    int err = X502_OutCycleLoadStart (dev, imp.size());
    if(err !=0) return err;
    err = X502_PrepareData (dev,
                            imp.data(),
                            nullptr,
                            nullptr,
                            imp.size(),
                            X502_DAC_FLAGS_VOLT | X502_DAC_FLAGS_CALIBR,
                            outBuf.data ());
    if(err != 0) return err;
    err = X502_Send(dev, outBuf.data (), outBuf.size (), 500);
    if(err > 0){
        err= X502_OutCycleSetup (dev,
X502_OUT_CYCLE_FLAGS_WAIT_DONE);
        if (method == 1) X502_StreamsStart (s);
        if(X502_IsRunning (dev) == X502_ERR_STREAM_IS_NOT_RUNNING)
            err = X502_StreamsStart (dev);
    }else{
        return err;
    }
}

```

```

    }
    return err;
}
const char *X502Manager::chErr(int err)
{
    return X502_GetErrorString (err);
}
int X502Manager::dataCollection( hnd & dev,QVector<double> &adcBuf, size_t
fullSize)
{
    int err =0;
    std::vector<uint32_t> recvBuf(fullSize);
    int dataCnt = X502_Recv(dev, recvBuf.data (), fullSize, 500);
    if(dataCnt > 0 ){
        err = X502_StreamsStop (dev);
        if(err != 0) return err;
        err = X502_ProcessAdcData (dev,
                                recvBuf.data (),
                                adcBuf.data (),
                                &fullSize,
                                X502_PROC_FLAGS_VOLT
                                );
        if(err != 0) return err;
    }else if (dataCnt == 0) emit sendMsg ("Отсутствуют данные ", false);
    else return dataCnt;
    return err;
}
QVector<double>      X502Manager::mergeImp(QVector<double>      &buf1,
QVector<double> &buf2)
{

```

```

QVector<double> res;
for(int i = 0; i < buf1.size (); ++i){
    res.append (buf1[i]);
    res.append (buf2[i]);
}
return res;
}
void X502Manager::createSign(std::vector<double> &imp, int freq , int method)
{
    Q_UNUSED (method)
    double freqChan = 2000000;
    if(rHead.impCnt == 0){
        for(size_t i = 0; i < imp.size (); ++i){
            imp[i] = fGenSin (static_cast<int>(i), freq, rHead.ampl, 0,0, freqChan);
        }
    }else{
        size_t impSize = rHead.impSize;
        size_t          perSize          =
static_cast<size_t>(((static_cast<double>(rHead.bufSize)/freqChan/2.)/rHead.impC
nt*freqChan);
        size_t currSize = 0;
        int cycle =0;
        while(cycle!= rHead.impCnt){
            for(size_t i =currSize; i < impSize + currSize; ++i ){
                imp[i] = fGenSign (static_cast<int>(i), freq, rHead.ampl, 0,0, freqChan);
            }
            for(size_t i = impSize+currSize; i < perSize+currSize; ++i){
                imp[i] = 0;
            }
            currSize += perSize;

```

```
        cycle++;  
    }  
}  
}  
hnd X502Manager::master() const  
{  
    return devs[0];  
}  
hnd X502Manager::slave() const  
{  
    return devs[1];  
}
```

Приложение Д. Листинг фрагмента обучения и вызова обученной модели программной системы прогнозирования.

```
import pandas as pd
import sys
import joblib
from catboost import CatBoostRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
def train_model(input_csv, output_pkl):
    print("Loading dataset:", input_csv)
    df = pd.read_csv(input_csv)
    target_column = "f_resonance"
    if target_column not in df.columns:
        raise Exception("CSV file must contain 'f_resonance' column")
    X = df.drop(columns=[target_column])
    y = df[target_column]
    cat_features = [i for i, col in enumerate(X.columns)
                    if X[col].dtype == "object" or X[col].dtype.name == "category"]
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.2, random_state=42
    )
    model = CatBoostRegressor(
        iterations=1500,
        depth=8,
        learning_rate=0.03,
        loss_function='RMSE',
        verbose=200
    )
```

```

print("Training model...")
model.fit(X_train, y_train, cat_features=cat_features, eval_set=(X_test, y_test))
print("Saving model to:", output_pkl)
joblib.dump({"model": model, "columns": X.columns.tolist(), "cat_features":
cat_features},
            output_pkl)
y_pred = model.predict(X_test)
print("MAE:", mean_absolute_error(y_test, y_pred))
print("MSE:", mean_squared_error(y_test, y_pred))
print("R2:", r2_score(y_test, y_pred))
if __name__ == "__main__":
    if len(sys.argv) < 3:
        print("Usage: train_model.py input.csv output.pkl")
        sys.exit(1)
    train_model(sys.argv[1], sys.argv[2])
import joblib
import pandas as pd
import sys
def predict(model_path, csv_input=None):
    data = joblib.load(model_path)
    model = data["model"]
    cols = data["columns"]
    if csv_input:
        df = pd.read_csv(csv_input)
    else:
        raise Exception("No input data provided!")
    df = df[cols]
    y_pred = model.predict(df)
    for v in y_pred:
        print(v)

```

```

if __name__ == "__main__":
    if len(sys.argv) < 3:
        print("Usage: predict.py model.pkl input.csv")
        sys.exit(1)
    predict(sys.argv[1], sys.argv[2])
#include <QProcess>
#include <QString>
#include <QFile>
#include <QTextStream>
#include <QDebug>
class PredictorML
{
public:
    PredictorML(const QString &pythonPath)
        : m_python(pythonPath)
    {}
    bool trainModel(const QString &csv, const QString &outPkl)
    {
        QString script = "train_model.py";
        QStringList args;
        args << script << csv << outPkl;
        QProcess p;
        p.start(m_python, args);
        p.waitForFinished(-1);
        QString out = p.readAllStandardOutput();
        QString err = p.readAllStandardError();
        qDebug() << "Training output:" << out;
        if (!err.isEmpty())
            qDebug() << "Training errors:" << err;
        return QFile::exists(outPkl);
    }
};

```

```

}
QVector<double> predict(const QString &modelPkl, const QString &inputCsv)
{
    QString script = "predict.py";
    QStringList args;
    args << script << modelPkl << inputCsv;
    QProcess p;
    p.start(m_python, args);
    p.waitForFinished(-1);
    QString out = p.readAllStandardOutput();
    QString err = p.readAllStandardError();
    if (!err.isEmpty())
        qDebug() << "Prediction errors:" << err;
    QVector<double> result;
    for (QString line : out.split("\n"))
    {
        bool ok = false;
        double val = line.trimmed().toDouble(&ok);
        if (ok)
            result.append(val);
    }
    return result;
}
private:
    QString m_python;
};

```