

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ» (НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ, НГУ)

Факультет **ФИЗИЧЕСКИЙ**

Кафедра автоматизации физико-технических исследований

Направление подготовки **03.03.02 ФИЗИКА**

Образовательная программа: **БАКАЛАВРИАТ**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

Янко Павел Евгеньевич

(Фамилия, Имя, Отчество автора)

Тема работы Разработка программного обеспечения для автоматизации работы напорного  
кавитационного стенда

**«К защите допущен»**

И. о. зав. кафедрой

к.т.н.

**Научный руководитель**

к.ф.-м.н.

н. с., ИТ СО РАН

Лысаков К. Ф./.....  
(фамилия И., О.) / (подпись, МП)

«.....».....20...г.

Кравцова А. Ю./.....  
(фамилия И., О.) / (подпись, МП)

«.....».....20...г.

Дата защиты: «.....».....20...г.

Новосибирск, 2021

## Оглавление

1. Введение.....	4
2. Обзор области.....	7
2.1. Система с обратной связью.....	7
2.2. Типовые законы управления.....	10
2.2.1. Пропорциональное управление.....	11
2.2.2. Пропорционально-дифференциальное управление .....	12
2.2.3. Интегральное управление .....	14
2.2.4. ПИД-регулирование .....	16
2. Предыдущие исследования .....	17
3. Устройство экспериментальной установки.....	20
4. Интерфейс RS-485.....	22
4.1. Описание интерфейса и общие характеристики.....	22
4.2. Топология сети на основе интерфейса RS-485 .....	23
5. Основы протокола Modbus.....	26
5.1. Modbus ASCII, Modbus RTU .....	26
5.2. Регистры Modbus.....	27
6. Процедура подключения ПК к ЧП и установления соединения.....	28
6.1. Подготовка общей шины для интерфейса RS-485 .....	28
6.2. Установление соединения с помощью программы VFD Soft .....	30
6.3. Установление соединения с помощью библиотеки PyModbus.....	31
7. Разработка пользовательского приложения для работы с ЧП .....	32
7.1. Средства разработки, требования и ограничения.....	32
7.2. Диаграмма юзкейсов программы .....	33
7.3. Sequence-диаграммы работы ПО.....	35
7.3.1. Задание параметров подключения .....	35
7.3.2. Подключение ПК к ЧП.....	37
7.3.3. Задание постоянной частоты ЧП.....	38
7.3.4. Создание сигнала .....	39
7.3.5. Отправка сигнала .....	41
7.4. Архитектура ПО .....	42
8. Результаты, детали реализации ПО.....	44

8.1. Метрики эффективности ПО .....	44
8.2. Методы снижения метрик .....	45
8.2.1. Стандартное отклонение сигнала.....	45
8.2.2. Временные задержки .....	48
9. Экспериментальная визуализация переходных режимов.....	54
10. Заключение .....	57
11. Список литературы .....	59

## 1. Введение

На сегодняшний день гидроэнергетика является преобладающим источником возобновляемой, экологичной энергии. В 2019 году, практически четверть источников энергии является возобновляемыми. Среди возобновляемых источников, около 70% приходится на гидроэнергетику (Рис. 1) [1].

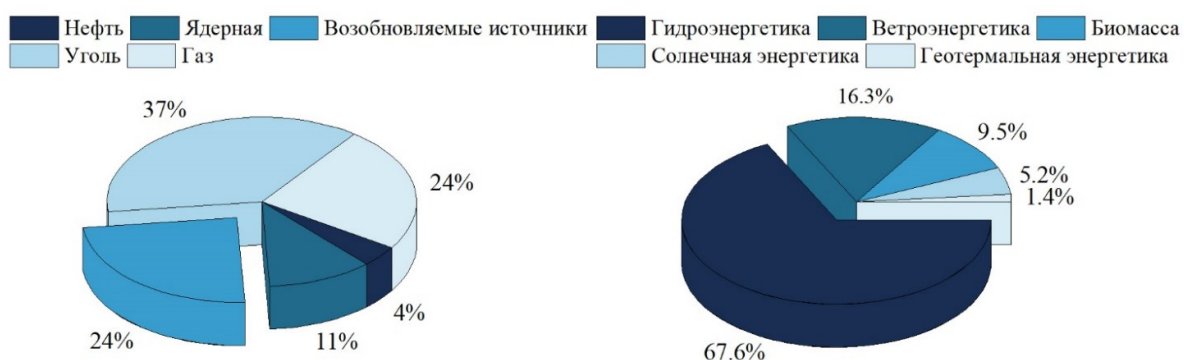


Рис. 1. (а) Доля возобновляемых источников энергии в мире, 2019г.(б) Состав возобновляемой энергетики [1]

Ответственность за преобразование энергии потока воды в электроэнергию лежит на Гидро Электростанциях. Для этого (Рис. 2) поток воды, создаваемый перепадом высот между нижним и верхним бьефами, приводит в движение рабочее колесо гидротурбины. Вращение вала гидротурбины запускает генератор, преобразующий механическую энергию в электрическую. Таким образом, гидротурбина является ключевым узлом ГЭС, от которого зависит эффективность преобразования энергии.

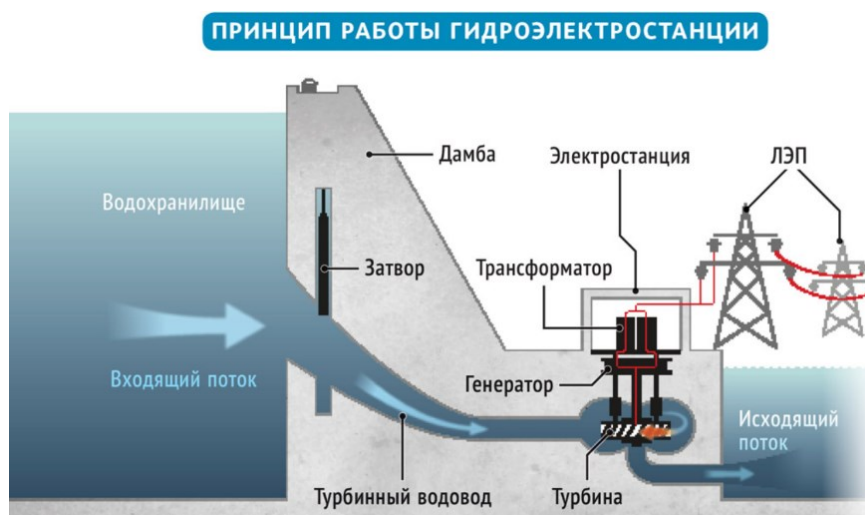


Рис. 2. Принцип работы ГЭС

Помимо генерации экологичной энергии, гидроэлектростанции так же несут регулируемую функцию для стабилизации работы электросетей. При регулировании гидротурбины работают в том числе в неоптимальных режимах, с расходом жидкости, отличным от оптимального. Согласно предыдущим исследованиям, при работе турбины в неоптимальном режиме, поток в области за рабочим колесом турбины имеет остаточную закрутку, что приводит к формированию вихревых структур [2, 3, 4].

Образующиеся вихревые структуры приводят к пульсациям давления и избыточным вибрациям оборудования ГЭС [5]. Это крайне негативно сказывается на безопасности ГЭС и КПД гидротурбин [6]. Так же, в области пониженного давления вихревых структур возникает явление кавитации – скопления парогазовых полостей. Схлопывание таких полостей в областях повышенного давления порождает ударную волну, приводящую к кавитационной эрозии стенок отсасывающей трубы (Рис. 3) [7].

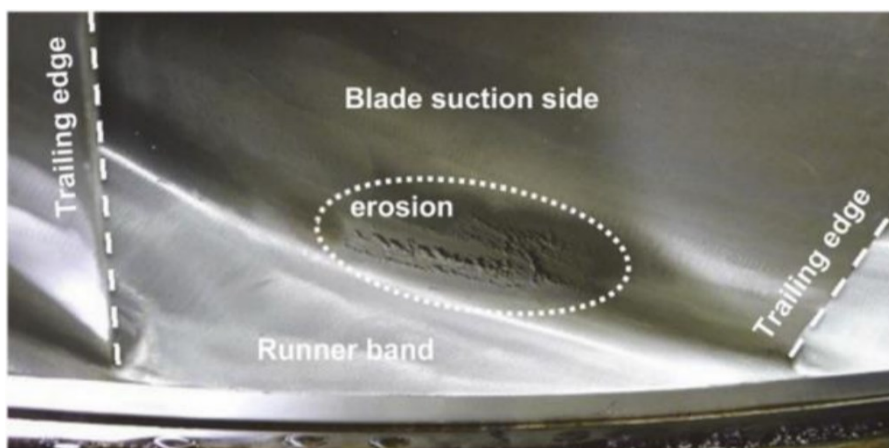


Рис. 3. Кавитационная эрозия стенок отсасывающей трубы турбины Френсиса [7]

Таким образом, систематическая работа гидротурбин в неоптимальных режимах приводит к снижению КПД электростанций и нарушению безопасности работы ГЭС.

Актуальность работы состоит в том, что автоматизация кавитационного стенда сделает возможным экспериментальное исследование переходных процессов в работе моделей гидротурбин. На данный момент, обнаружена нехватка экспериментальных исследований между неоптимальными режимами работы турбин. В области исследований переходных процессов преобладают численные методы моделирования потока [2, 3, 4, 8, 9].

Ранее, на кавитационном стенде была возможна постановка эксперимента в стационарных условиях с постоянным расходом жидкости. Поэтому, целью работы является реализация программного обеспечения, позволяющего динамически изменять расход жидкости на кавитационном стенде. Использование реализованного ПО позволит глубже исследовать физику нестационарных вихревых течений и верифицировать численное моделирование потока в гидротурбинах.

Для достижения цели были поставлены следующие задачи:

1. Провести литературный обзор экспериментальных и численных исследований переходных режимов течения в гидротурбинах

2. Изучить принцип работы экспериментального стенда с точки зрения теории автоматического регулирования
3. Изучить предыдущие программные решения для динамического управления расходом
4. Изучить мануал частотного преобразователя Delta CP 2000, описание регистров устройства, а также поддерживаемые протоколы обмена данными с внешними устройствами
5. Изучить протокол Modbus для отправки запросов на чтение данных от ПК к ЧП
6. Успешно установить соединение ПК с ЧП по протоколу Modbus с помощью программного обеспечения производителя
7. Разработать расширяемую, поддерживаемую архитектуру для реализуемого ПО
8. Реализовать ПО согласно требованиям сотрудников лаборатории
9. Использовать реализованное ПО для проведения экспериментальной визуализации переходных режимов работы модели турбины

## **2. Обзор области**

### **2.1. Система с обратной связью**

Областью работы является теория автоматического регулирования (ТАР). Согласно терминологии ТАР, экспериментальный стенд является замкнутым контуром регулирования (Рис. 4). Ключевыми элементами контура регулирования являются объект управления (ОУ) и регулятор, так же известный как устройство управления (УУ) [10].



Рис. 4. Обобщённая схема замкнутого контура регулирования.  $w(t)$  – заданное значение,  $z(t)$  – возмущающее воздействие,  $x(t)$  – регулируемая величина,  $y(t)$  – управляющее воздействие,  $y_R(t)$  – управляющий сигнал,  $e(t)$  – рассогласование регулируемой величины

Согласно ТАР, процесс управления состоит в подаче управляющего воздействия  $y(t)$  от УУ на ОУ [11]. Объектом управления является часть установки, для которой необходимо поддерживать необходимую регулируемую величину  $x(t)$ . На значение регулируемой величины  $x(t)$  возможно влиять с помощью управляющего воздействия  $y(t)$ . Поскольку в ряде случаев регулятор не способен задавать управляющее воздействие непосредственно, управляющий сигнал  $y_R(t)$  подаётся на исполнительные элементы, связанные с объектом регулирования.

Цель регулятора состоит в том, чтобы добиться равенства между регулируемой величиной  $x(t)$  и заданным значением  $w(t)$ . При этом, вследствие возмущающего воздействия  $z(t)$ , возникает разность между фактической и целевой величиной. Эта разность называется рассогласованием регулируемой величины  $e(t)$  и передаётся на регулятор посредством обратной связи.

Необходимо установить, что является объектом управления в данной работе. Для этого был изучен принцип работы центробежного насоса, использующегося в экспериментальной установке (Рис. 5). Ключевым



элементом является рабочее колесо насоса. Жидкость, поступающая из входного патрубка, смещается к периферийной области в спиральный отвод, после чего поступает в диффузор. Таким образом, расход жидкости связан с частотой вращения рабочего колеса. Вращение рабочего колеса управляется валом электродвигателя, который является объектом управления в данной работе.

Устройством управления (УУ) является частотный преобразователь Delta SP 2000, поскольку скорость вращения вала зависит от частоты выходного напряжения. Частотный преобразователь является источником управляющего воздействия  $u(t)$ . В данной работе, реализуемое ПО ответственно за своевременную подачу заданного значения  $w(t)$ , для задания необходимой формы частотного профиля на преобразователе (Рис. 4).

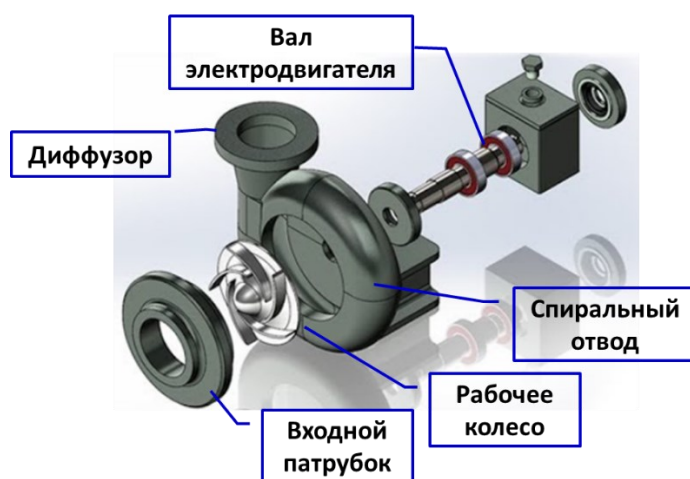


Рис. 5. Компоненты центробежного насоса

Ключевым принципом автоматического регулирования является наличие системы с обратной связью (Рис. 6). Для этого, рассогласование регулируемой величины определяется как разность между регулируемой и заданной величинами:

$$e(t) = w(t) - x(t) \quad (1)$$

В случае, когда сигнал ошибки  $e(t)$  подаётся на вход УУ, возможно скорректировать управление ОУ для компенсации измеренного рассогласования.

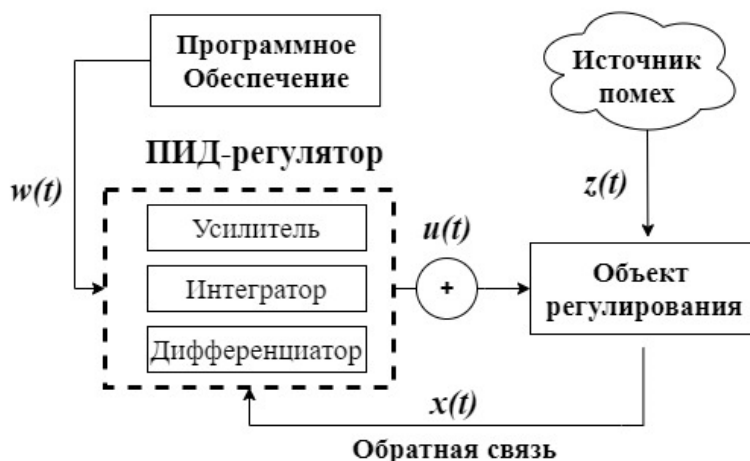


Рис. 6. Механизм работы системы с обратной связью.  $W(t)$  – заданное значение,  $u(t)$  – управляющее воздействие,  $z(t)$  – возмущающее воздействие,  $x(t)$  – регулируемая величина

## 2.2. Типовые законы управления

Основой для построения системы с обратной связью является рассогласование регулируемой величины  $e(t)$ , передаваемое от ОУ на УУ с целью корректировки дальнейшего управления. В зависимости от действий, совершаемых с величиной  $e(t)$ , различают четыре типовых закона управления – пропорциональный, пропорционально-дифференциальный, интегральный и пропорционально-интегрально-дифференциальный.

Перед обзором четырёх типовых законов управления, необходимо объяснить значение терминов, используемых для описания режима работы управляемой системы. Так, установившимся (стационарным) называют режим работы, при котором параметры системы остаются неизменными. Если же в системе происходит один или несколько переходных процессов, режим её работы называется переходным.

При задании новой целевой величины  $y(t)$  в контуре регулирования, период времени, в течение которого фактическое значение  $x(t)$  впервые

сравнивается с заданной величиной, называется временем регулирования  $T_{\text{рег}}$  (Рис. 7).

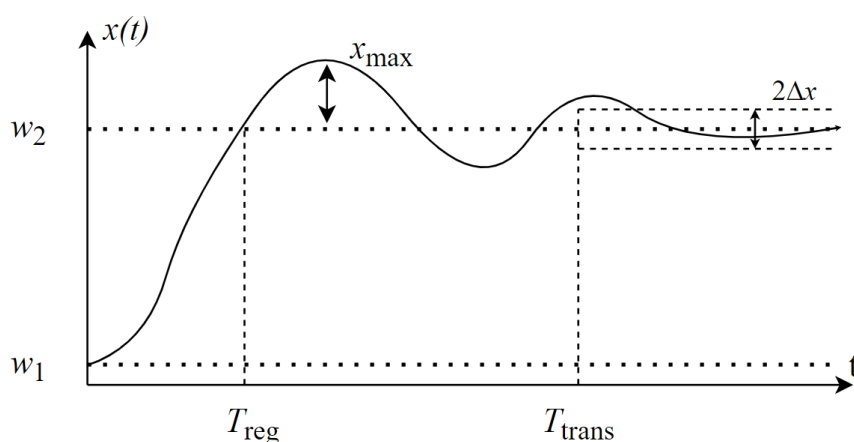


Рис. 7. Критерии переходного процесса

На Рис. 7, при переходе от предыдущего заданного значения  $w_1$  к новому  $w_2$ , максимальное отклонение фактической величины  $x(t)$  от заданной называется пиком перерегулирования  $x_{\text{max}}$ . Период времени, в течение которого фактическая величина выходит на заданный уровень, с точностью до интервала  $\Delta x$ , называется временем переходного процесса  $T_{\text{trans}}$ .

### 2.2.1. Пропорциональное управление

Пропорциональный закон управления является наиболее распространённым и простым в реализации. Управляющим сигналом является рассогласование  $e(t)$ , умноженное на постоянную величину. В таком случае, в качестве УУ служит усилитель, а сигнал управления  $u(t)$  пропорционален сигналу ошибки  $e(t)$ , переданному посредством обратной связи:

$$u_p(t) = k_p \times e(t) = k_p \times (w(t) - x(t)), \quad (2)$$

где коэффициент  $k_p$  называется пропорциональным передаточным коэффициентом,  $w(t)$  — заданная величина, а  $x(t)$  — фактическая величина. Такую систему называют П-регулятором. Преимущества пропорционального

управления заключаются в простоте реализации и скорости отклика на сигнал ошибки. Из-за скорости операции умножения на константу, П-регулятор изменяет сигнал управления без временной задержки.

Однако, схемы, построенные по принципу пропорционального управления, очень чувствительны к подбору передаточного коэффициента  $k_p$ . Так, при подаче единичного скачка на вход системы, вид переходной характеристики П-регулятора сильно зависит от выбранного коэффициента (Рис. 8). На Рис. 8 так же видно, как большой передаточный коэффициент является причиной перерегулирования и длительного перехода в установившийся режим.

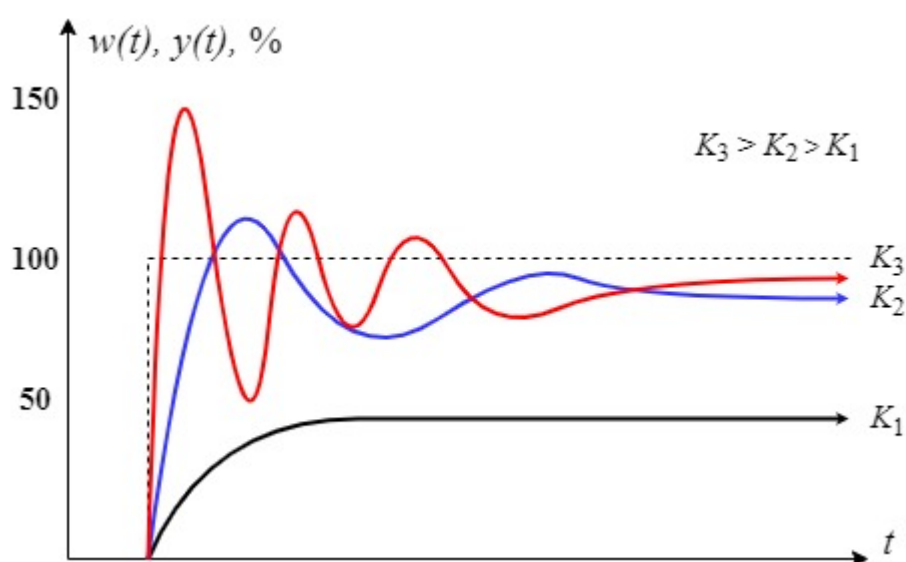


Рис. 8. Вид переходной характеристики П-регулятора при подаче на вход единичного скачка.  $K$  – передаточный коэффициент П-регулятора

### 2.2.2. Пропорционально-дифференциальное управление

Подход пропорционально-дифференциального управления подходит для случаев резких перепадов в управляющем сигнале  $y(t)$ . В случае резких изменений, необходимо знать не только текущий, но и ожидаемый сигнал ошибки  $e(t)$ . Элементом схемы, способным измерить скорость изменения ошибки, является дифференциатор. Дифференциатор дополнительно учитывает разность между смежными показаниями фактической величины  $x(t)$ , с точностью до коэффициента  $k_d$ .

$$u_d(t) = k_d \times (x(t) - x(t - dt)) \quad (3)$$

Данная схема называется ПД-регулятором. В итоге, управляющий сигнал представляет себя сумму из пропорциональной и дифференциальной компонент:

$$u_{pd}(t) = u_p(t) + u_d(t) = k_p \times e(t) + k_d \times (x(t) - x(t - dt)) \quad (4)$$

Для случая скачка заданного значения  $w(t)$ , динамика дифференциальной и пропорциональной компонент управляющего сигнала была рассмотрена в литературе [10, 12]. Так, Рис. 9 демонстрирует мгновенный скачок пропорциональной составляющей, в момент задания нового значения  $w_2(t)$ . В процессе нарастания фактического значения  $x(t)$ , рассогласование регулируемой величины  $e(t)$  снижается, что ведёт к уменьшению пропорциональной составляющей. При этом, из-за линейного возрастания регулируемой величины  $x(t)$ , скорость её изменения остаётся постоянной. Следовательно, дифференциальная компонента, пропорциональная  $\Delta x(t)$  сохраняет постоянное значение большую часть времени регулирования.

Таким образом, дифференциальная составляющая оказывает демпфирующее действие, уменьшая управляющий сигнал тем сильнее, чем выше скорость изменения регулируемой величины  $x(t)$ .

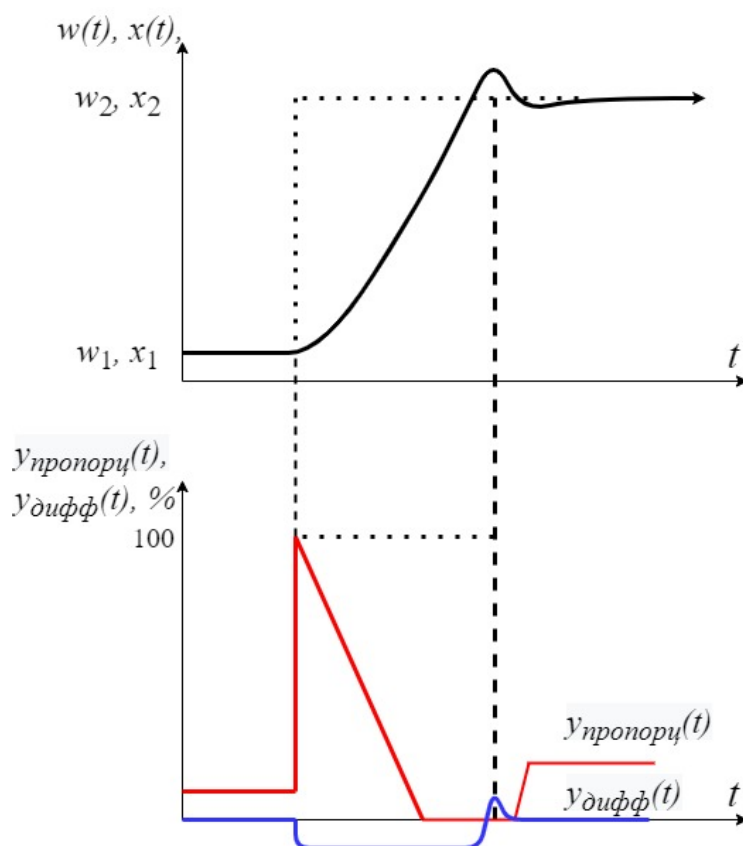


Рис. 9. Пропорциональная и дифференциальная составляющие управляющего сигнала ПД-регулятора замкнутого контура

Важно отметить, что дифференциатор никогда не используется отдельно, а только совместно с П либо ПИ регуляторами. Достоинствами ПД-регулирования является скорость отклика, поскольку дифференциальная компонента учитывает скорость изменения выходного сигнала. Недостатки включают в себя чувствительность к высокочастотным шумам выходного сигнала. С целью подавить высокочастотные шумы, на входе дифференциатора используются низкочастотные фильтры.

### 2.2.3. Интегральное управление

Для того, чтобы снизить чувствительность системы к шумам выходного сигнала, в составе системы с обратной связью используется интегратор. Интегрирующее звено суммирует рассогласование  $e(t)$ , накопленное за всё время работы системы. Таким образом, суммирование сглаживает шумы,

обеспечивая своевременную реакцию на долгосрочные отклонения выходного сигнала от задающего воздействия.

$$u_i(t) = \frac{1}{T_N - T_1} \times \sum_{j=1}^{j=N} e(t_j) \Delta t_j = k_i \times \sum_{j=1}^{j=N} e(t_j) \Delta t_j \quad (5)$$

где  $k_i$  – коэффициент, обратно пропорциональный времени интегрирования. Чем меньше время интегрирования, тем быстрее отклик И-регулятора на изменения в системе. Переходная характеристика И-регулятора наглядно показана на Рис. 10. В момент  $t_1$  происходит задание новой целевой величины  $w(t)$ . Из-за моментального возрастания рассогласования заданной и фактической величин, И-регулятор начинает наращивать управляющий сигнал  $y(t)$ . По мере уменьшения рассогласования, значение управляющего сигнала выходит на постоянную величину.

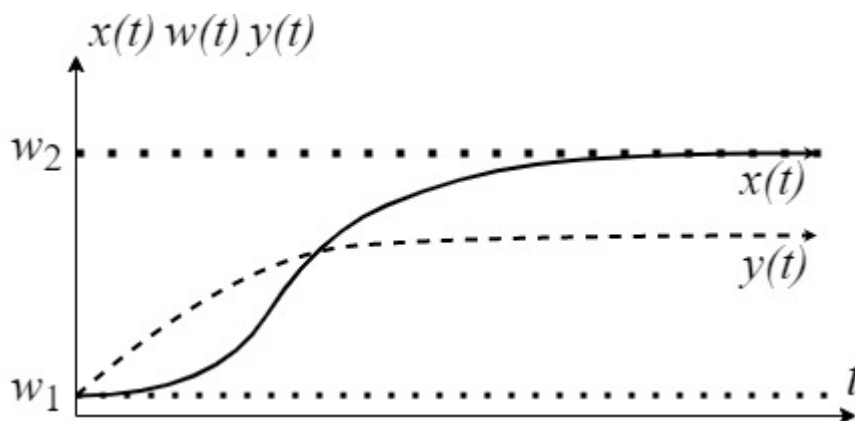


Рис. 10. Переходная характеристика И-регулятора.  $w(t)$  – заданное значение,  $x(t)$  – регулируемая величина,  $y(t)$  – сигнал управления

Преимуществами И-регуляторов является полное устранение рассогласования. Однако, такие системы обладают долговременным откликом на изменения заданной величины, поскольку для полной компенсации рассогласования требуется время. Поэтому, одной из важнейших характеристик И-регуляторов является время отставания  $T_{\text{delay}}$ . Так же, переходная характеристика И-регулятора чувствительна к подбору

времени интегрирования. Так, Рис. 11 демонстрирует, как выбор малого времени интегрирования приводит к перерегулированию.

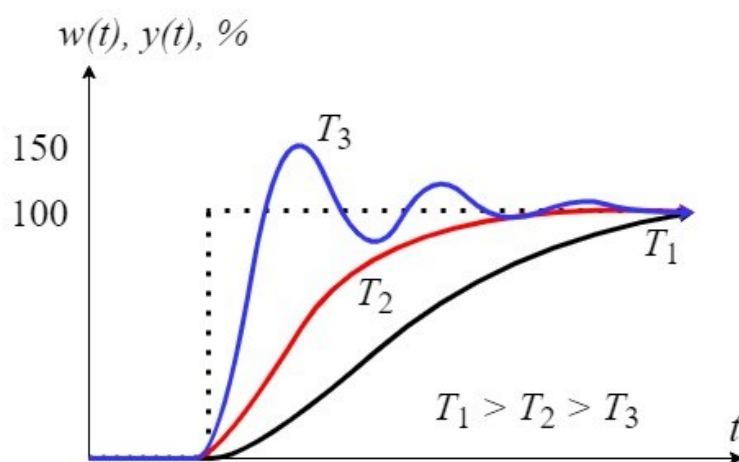


Рис. 11. Переходная характеристика И-регулятора при подаче на вход единичного скачка.  $T_{1,2,3}$  – время интегрирования

В схемах с обратной связью, интегратор так же используют совместно с пропорциональным звеном. В случае совместного использования интегральной и пропорциональной компонент, система называется ПИ-регулятором:

$$u_{pi}(t) = u_p(t) + u_i(t) = k_p \times e(t) + k_i \times \sum_{j=1}^{j=N} e(t_j) \Delta t_j \quad (6)$$

#### 2.2.4. ПИД-регулирование

В итоге, наиболее гибким и точным законом управления является пропорционально-интегрально-дифференциальный, совместивший в себе одновременную работу всех описанных звеньев. Выходы дифференциатора, интегратора и усилителя для получения полной компоненты сигнала управления:



$$u_{pid}(t) = k_p \times e(t) + k_i \times \sum_{j=1}^{j=N} e(t_j) \Delta t_j + k_d \times (y_{\text{вых}}(t) - y_{\text{вых}}(t - dt)) \quad (7)$$

В итоге, УУ в схеме с обратной связью состоит из трёх различных устройств, ответственных за обработку сигнала ошибки, поступившего на вход УУ посредством обратной связи. Усилитель ответственен за обработку «текущей» реакции на ошибку, тогда как дифференциатор компенсирует ошибку наперёд. В то же время, интегратор ориентирован на компенсацию сигнала ошибок, накопленных за прошедшее время работы системы. Полученная схема называется ПИД-регулятором (Рис. 12).

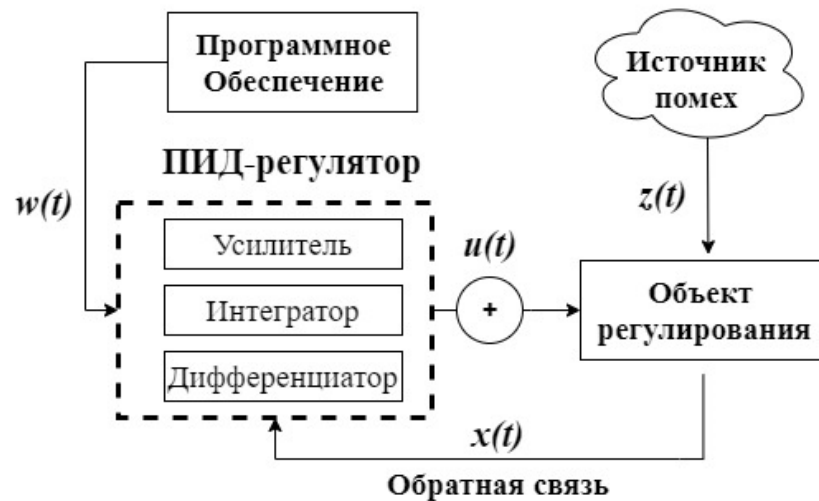


Рис. 12. Схема ПИД-регулятора.  $W(t)$  – задающее воздействие,  $u(t)$  – управляющий сигнал,  $z(t)$  – сигнал помех,  $x(t)$  – регулируемая величина

## 2. Предыдущие исследования

Как показано ранее, при работе гидротурбин в неоптимальных режимах, в закрученном потоке за рабочим колесом формируются вихревые структуры. Так, в работе Скрипкина et al. [5] проведено исследование прецессирующего вихревого жгута в отсасывающей трубе гидротурбины. В данной работе представлена визуализация формирования вихревого кольца, синхронизированная с осциллограммой давления потока на стенки

отсасывающей трубы. Авторы обнаружили резкий скачок давления потока в момент касания вихревым кольцом стенок трубы. Однако, визуализация выполнена в стационарных условиях. Например, расход жидкости стабилизировался магнитным расходомером с точностью до 0,5%.

Для исследования формирования вихревых структур, проведено множество экспериментальных и численных исследований [3, 13, 14]. В работе Farvel et al. [15] уделено большое внимание явлению резонанса. Авторы утверждают, что в режиме частичной нагрузки, образующийся вихревой жгут за рабочим колесом турбины является причиной периодических возмущений. В таком случае, совпадение частоты возбуждения с собственными частотами деталей турбины приводит к резкому росту амплитуды пульсаций давления.

Для исследований поперечных и продольных компонент скорости потока, авторы использовали метод лазерной доплеровской velocиметрии (ЛДВ) на модели турбины Френсиса. В данном исследовании удалось поставить эксперимент в условиях резонанса. Метод ЛДВ позволил измерить флуктуации продольной и поперечной компонент скорости потока, порождённых вихревым жгутом в режиме частичной нагрузки. Более того, авторы измерили пульсации давления в конусе отсасывающей трубы. Было обнаружено, что рост области кавитации (cavitation volume) приводит к снижению интенсивности пульсаций давления. Также, не удалось установить периодический характер изменения продольной компоненты скорости, что говорит об отсутствии влияния резонанса на продольную скорость потока. В работе показано существенное влияние вихревого ядра на пульсации поперечной компоненты скорости потока.

В работе Trivedi et al. [8] проведено экспериментальное исследование синхронных и асинхронных пульсаций давления в переходных режимах работы турбин Френсиса. В работе рассмотрены турбины вертикальной и горизонтальной ориентации. Для измерения давления потока были

использованы четыре сенсора, распределённые по периметру конуса отсасывающей трубы. Авторы обнаружили значительные амплитуды пульсаций в режимах изменения нагрузки и пуска турбины. Так, среднеквадратичное отклонение пульсаций в переходных режимах достигало 5,4 кПа, тогда как полной нагрузке соответствует давление около 85 кПа.

Исследование переходных режимов так же проводится на аэродинамических стендах. В работах И. Литвинова и Д. Суслова [16, 17] рассмотрено формирование прецессирующего вихревого ядра в воздушной модели отсасывающей трубы гидротурбины. В работе [17], полученная карта пульсаций давления выделила диапазон расходов и частот вращения завихрителя, способствующих формированию ПВЯ. Для перехода от режима частичной нагрузки к режиму с наибольшим КПД турбины, построенная вейвлет диаграмма позволила рассчитать время жизни ПВЯ.

Исследование режимов работы гидротурбин не ограничивается экспериментальными данными. Например, в работе Авдюшенко et al. [4] проведено численное моделирование потока в натурной турбине Френсиса для нескольких переходных процессов – пуска в турбинный режим, снижения мощности и мгновенного сброса нагрузки. В работе рассчитано распределение давления на лопасти, а также линии тока в сечениях рабочего колеса.

В работе J. Nicolle et al. [9] проведено сравнение экспериментальных и численных исследований режимов пуска и холостого хода рабочего колеса. Авторам удалось верифицировать численный расчёт скорости вращения рабочего колеса турбины Френсиса. С помощью датчика давления, расположенного на рабочем колесе, сопоставлены расчётная и экспериментальная осциллограммы давления в режиме пуска. Однако, для верификации моделирования структуры потока необходимы дополнительные экспериментальные исследования.

Главным результатом моделирования S. Cherny et al. [2] является визуализация изо-поверхностей давления потока, а также расчёт пульсаций давления на стенки отсасывающей трубы. Изо-поверхности наглядно демонстрируют развитие вихревого жгута, ранее отмеченное в экспериментальных исследованиях [5].

### **3. Устройство экспериментальной установки**

Кавитационный стенд представляет из себя замкнутый вертикальный гидродинамический контур (Рис. 13). В рабочий участок контура помещается модель гидротурбины, состоящая из двух лопаточных завихрителей, установленных на входе. Подача жидкости в контур достигается за счёт центробежного насоса. Под действием переменного напряжения от частотного преобразователя Delta CP 2000, вращение вала центробежного насоса приводит в действие крыльчатку, прокачивающую жидкость через рабочий участок. В данной установке центробежный насос 3LMH 80-160/18.5 фирмы Ebara обеспечивает расход в контуре до  $200 \text{ м}^3/\text{ч}$ . Частота выходного напряжения регулируется с помощью управляющего пульта КРС-СЕ01. Контроль расхода жидкости осуществляется с помощью электромагнитного расходомера «Взлёт Лайт М+».

Система обратной связи между расходомером и частотным преобразователем позволяет работать в режиме постоянного расхода, несмотря на изменение внешних параметров (например, снижения вязкости жидкости из-за роста температуры). Вакуумный насос в верхней части контура предназначен для создания дополнительного разрежения.



Рис. 13. Устройство кавитационного стенда

Модель гидротурбины состоит из конической отсасывающей трубы и двух лопаточных завихрителей на её входе (Рис. 14). Рабочее колесо одного из завихрителей может вращаться вокруг своей оси, имитируя работу гидротурбины. Вращение завихрителя достигается за счёт ременной передачи между колесом и электродвигателем.

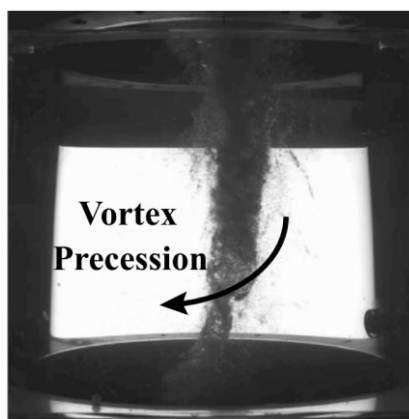


Рис. 14. (а) – Модель закручивающего устройства; (б) – Визуализация вихревого жгута в модели конуса отсасывающей трубы [15]

В данной работе, с целью автоматического контроля расхода жидкости, необходимо найти способ управления частотой выходного напряжения частотного преобразователя. Благодаря автоматизации задания частот напряжения, станет возможным управления частотой вращения вала

центробежного насоса, а следовательно, и расходом жидкости. Согласно пользовательскому руководству частотного преобразователя, управление частотой с устройств, отличных от пульта КРС-СЕ01, возможно с помощью интерфейса RS-485 и протокола Modbus.

## **4. Интерфейс RS-485**

### **4.1. Описание интерфейса и общие характеристики**

RS-485 (RS – “Recommended Standard”) – стандарт физического уровня, используемый для промышленной автоматизации. Официальное название стандарта TIA/EIA-485-A. Популярность интерфейса связана с поддержкой полудуплексной передачи данных, возможностью работы с несколькими устройствами, подключёнными на шину. Интерфейс RS-485 позволяет достаточно быструю связь устройств, расположенных на расстояниях до километра. Так, на максимальной дистанции в 1200 м, скорость передачи данных достигает 62,5 кБит / с [18].

RS-485 основан на дифференциальном способе передачи сигнала. При таком способе передачи, напряжение, соответствующее логическому биту (0 или 1), измеряется как разность потенциалов между двумя передающими линиями: Signal+ и Signal-. Напряжение каждой из линий Signal+ и Signal- может быть произвольным, однако не должно превышать значений от -7 до +12 В.

Аналогичным образом, приёмники сигнала, передающегося по линиям Signal+, Signal-, воспринимают только разность между напряжениями на них. Такие приёмники называют дифференциальными. Согласно стандарту RS-485, диапазон разности напряжений от 200 мВ до +12 В соответствует переданной логической единице, от -200 мВ до -7 В – логическому нулю. Стандарт RS-485 так же регламентирует величину дифференциального напряжения – она не должна быть менее 1,5 В. Данное требование стандарта

связано с предотвращением помех из-за падения напряжения на длинной линии с большим омическим сопротивлением.

Стандарт передачи данных RS-485 устойчив к помехам, в случае одинаковых по форме и величине наводок на линиях Data+, Data-. Такой тип помех называется синфазным. Это связано с тем, что приёмник определяет величину передаваемого сигнала как разность напряжений линий Data+ и Data-. Следовательно, симметричные наводки ослабляются после расчёта разности напряжений.

Для снижения влияния наводок и помех, в качестве линии передачи данных в работе использована витая пара. Соблюдение условий стандарта даёт возможность для коммуникации двух и более устройств на расстоянии до 1200 м. С ростом расстояния от 10 м до 1200 м, скорость передачи данных падает от 10000 кБит / с до 62,5 кБит / с.

#### **4.2. Топология сети на основе интерфейса RS-485**

Общее свойство всех топологий сети на основе интерфейса RS-485 – наличие общей шины. В качестве общей шины, для снижения индуцируемых электромагнитных помех используется витая пара.

Сети RS-485 работают в полудуплексном режиме. Каждое устройство, подключенное к сети, может являться как приёмником, так и передатчиком данных, но из-за использования для передачи одной линии связи, в каждый момент времени данные могут передаваться только в одном направлении, поэтому в каждый момент времени только одно устройство в сети может являться передатчиком.

В связи с данным ограничением, простейшей топологией является соединение точка-точка – непосредственное соединение приёмника и передатчика общей шиной (Рис. 15). Топология точка-точка реализована в данной работе, поскольку необходимо реализовать управление только одним устройством – частотным преобразователем.

В случае необходимости подключения нескольких устройств к общей шине RS-485, используется топология подключения «общая шина» (Рис. 15). Согласно данному типу подключения, подразумевается наличие общего кабеля, называемого шиной или магистралью, к которому подключаются все рабочие станции.

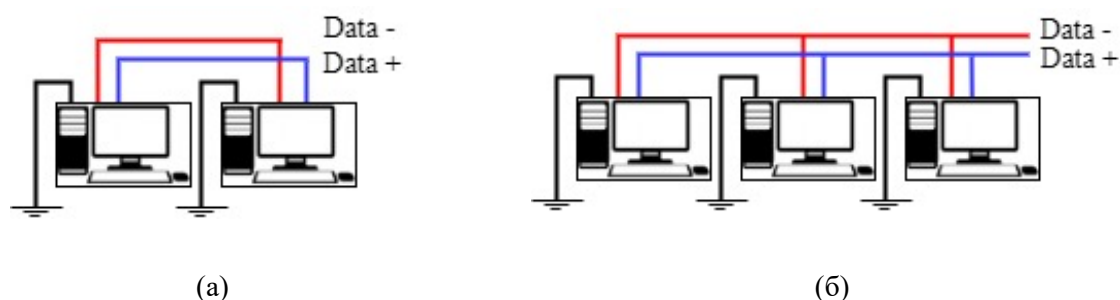


Рис. 15. (а) – Схема подключения «точка-точка» устройств в сети RS-485,  
(б) – Схема подключения нескольких устройств к сети RS-485, согласно  
топологии «общая шина»

Стоит отметить наличие ограничения на количество устройств, подключаемых к общей шине. Поскольку приемники и передатчики имеют конечное сопротивление, их подключение к сети увеличивает нагрузку на сеть. Для того, чтобы рассчитать максимальное количество устройств, способных к подключению разрабатываемой сети RS-485, введено понятие единичной нагрузки. Все устройства, подключаемые к сети RS-485, должны характеризоваться определённой долей от единичной нагрузки. Например, устройство приёмопередатчик MAX3485 специфицирован как 1 единичная нагрузка, тогда как MAX487 характеризуется как 1/4 от единичной нагрузки. При подключении нескольких приёмопередатчиков к сети RS-485, максимальное число единичных нагрузок на витой паре не должно превышать 32. Это означает, что к одной общей шине могут быть подключены до 32 устройств типа MAX3485.



Необходимо так же уделить внимание выбору кабеля для соединения устройств. Например, скрутки кабелей и длинные ответвления негативно сказываются на сообщении устройств в сети (Рис. 16). Ухудшение качества сигнала связано с эффектом длинной линии, проявляющимся при высокой скорости передачи данных, а также с ограниченной скоростью распространения электромагнитных волн в проводниках. График на Рис. 17 показывает, что наличие длинных ответвлений накладывает ограничение на выбор скорости передачи данных (Кбит / с). Таким образом, значительные расстояния (порядка сотен метров) от устройств до общей шины, совместно с отражением сигнала от ответвлений, приводят к передаче искажённых данных и негативно сказываются на производительности сети.

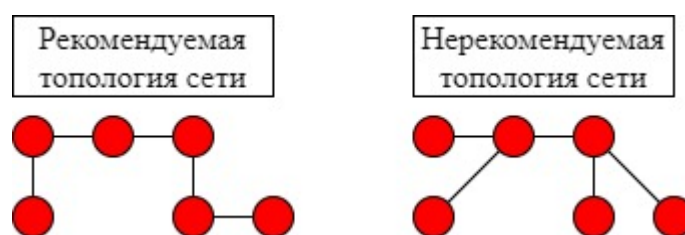


Рис. 16. Пример рекомендуемой и не рекомендуемой топологии сети из нескольких устройств, подключенных к общей шине RS-485. Красным отмечены узлы сети с подключённым согласующим резистором – терминатором

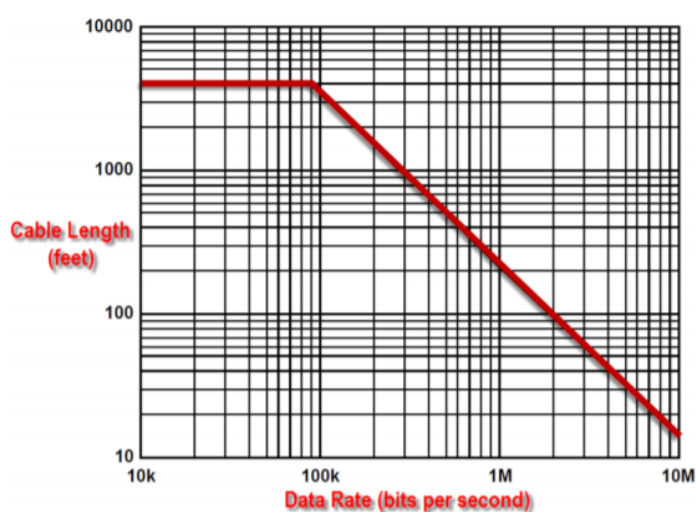


Рис. 17. Зависимость скорости передачи данных по интерфейсу RS-485 от длины проводного соединения [18]

## 5. Основы протокола Modbus

Modbus – промышленный протокол, представленный компанией Modicon в 1979 году. Благодаря данному стандарту, возможна коммуникация оборудования, поставляемого различными производителями, при условии поддерживаемого обеими сторонами протокола Modbus. Сообщение устройств по протоколу Modbus позволяет решать широкий спектр задач, касающихся сбора данных с датчиков, удалённого управления промышленным оборудованием, отслеживания параметров взаимодействующих устройств.

В данной работе, используемый частотный преобразователь Delta CP 2000 поддерживает взаимодействие по протоколу Modbus RTU и Modbus ASCII.

### 5.1. Modbus ASCII, Modbus RTU

Согласно протоколу Modbus ASCII, данные в передаваемых фреймах кодируются символами из таблицы ASCII и передаются в шестнадцатеричном формате. Начало каждого пакета обозначается символом двоеточия, а конец — символами возврата каретки и переноса строки.

Отличие протоколов Modbus ASCII и Modbus RTU главным образом заключается в структуре пакетов данных (Рис. 18). Так, в протоколе Modbus RTU кодирование данных осуществляется в двоичном формате, отсутствует символ разделения пакетов друг от друга. На Рис. 18 так же отображена терминология, используемая при описании работы протоколов Modbus.

- ADU – application data unit – термин для описания целого пакета, включающего заголовки, основную часть с передаваемыми данными, контрольную сумму и адрес устройства-получателя.
- PDU – protocol data unit – часть пакета с передаваемыми данными, так же называемый payload.

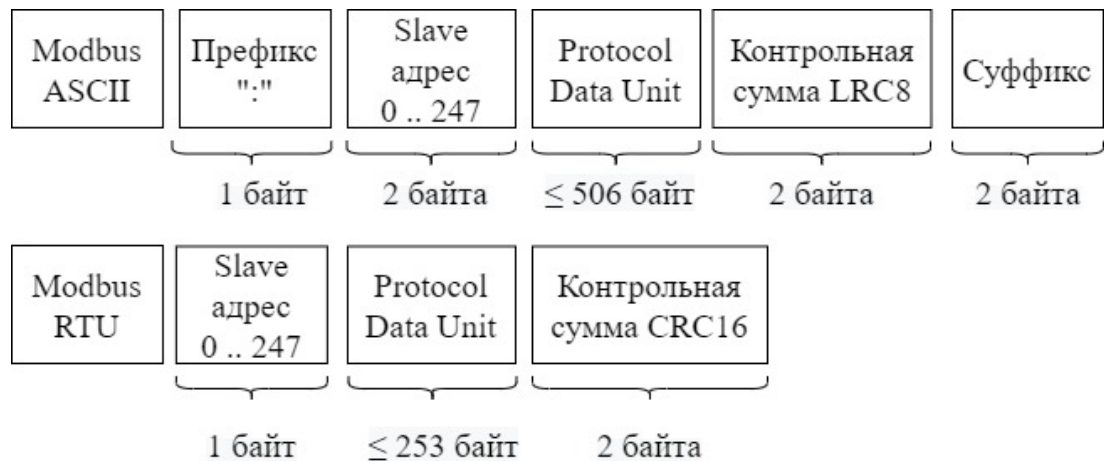


Рис. 18. Структура пакетов данных, согласно протоколам Modbus ASCII и Modbus RTU

Протокол основан на архитектуре master-slave (ведущий-ведомый). Для передачи данных по протоколу Modbus, используются такие интерфейсы как RS-232, RS-485, RS-422. Например, в данной работе, обмен данными с частотным преобразователем Delta CP 2000 осуществляется по протоколу Modbus, с помощью интерфейса RS-485.

## 5.2. Регистры Modbus

Поскольку взаимодействие с оборудованием по протоколу Modbus сводится к чтению и изменению данных регистров устройства, необходимо дать описание регистров согласно протоколу Modbus.

Две основные функции для реализации работы с устройством по протоколу Modbus – это функции чтения и записи. В отправляемых пакетах данных, коды этих функций передаются вместе с регистром устройства, с которым производится операция. Терминология протокола Modbus унаследовала названия типов регистров от языков релейной логики. Различают четыре типа регистров: discrete inputs, coils, inputs и holding registers.

Дискретными входами (discrete inputs) называют регистры, доступные только для чтения. Термин «катушки (coils)» обозначает дискретные выходы

устройства, либо внутренние значения, доступные для чтения и записи. Входные регистры (input registers) – регистры, доступные только для чтения, а регистры хранения (holding registers) – доступные для чтения и записи 16-битные выходы устройства.

Как правило, описание функции тех или иных регистров может быть найдена в пользовательском руководстве устройств (Рис. 19). Например, в данной работе, информация о регистрах для чтения и записи частотного преобразователя Delta CP 2000 найдена в руководстве по эксплуатации [18]. Таблица описания адресов содержит всю необходимую информацию о типе регистра (только для чтения, для чтения и записи и т.д.), а также о соответствии битов регистра и режимом работы частотного преобразователя (команды задания выходной частоты, команды Пуск, Стоп и т.д.).

Индикация состояния Только чтение	13~15	
	2102H	Заданная частота (F) (XXX.XXГц)
	2103H	Выходная частота (H) (XXX.XXГц)
	2104H	Выходной ток (XX.XXA. Если ток выше 655.35, то XXX.XA. Отброшенный разряд будет записан в старшем байте 211F.)

Рис. 19. Описание регистров для чтения в руководстве по эксплуатации частотного преобразователя Delta CP 2000 [18]

## 6. Процедура подключения ПК к ЧП и установления соединения

### 6.1. Подготовка общей шины для интерфейса RS-485

В данной работе, в качестве регулятора используется частотный преобразователь Delta CP 2000. Согласно руководству пользователя, данное устройство поддерживает обмен данными по протоколу Modbus ASCII и Modbus RTU. При этом, сообщающиеся устройства обязаны поддерживать обмен данными посредством интерфейса RS-485.

Для подключения частотного преобразователя к линиям передачи сигнала по дифференциальному интерфейсу RS-485 (Data+ и Data-), на лицевой панели устройства предусмотрены специальные клеммы (Рис. 20).

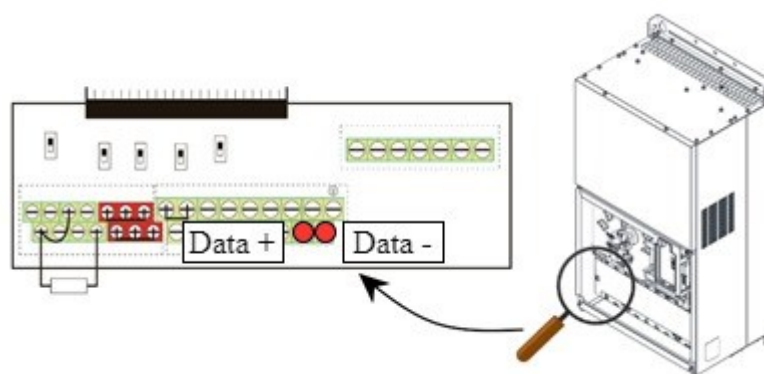


Рис. 20. Клеммы Data+ и Data- на лицевой панели преобразователя, предусмотренные для сообщения с внешними устройствами посредством интерфейса RS-485

Для снижения влияния электромагнитных помех, в качестве проводника, соединяющего ПК с ЧП используется витая пара. В данной работе, конструкция ПК не предусматривает наличие интерфейса RS-485. Однако, в большинстве современных компьютеров поддерживается интерфейс USB. Поэтому, для обеспечения взаимодействия, использован конвертер интерфейсов RS-485 / USB (Рис. 21).



Рис. 21. Используемый в работе адаптер интерфейсов RS-485 / USB

Расстояние от преобразователя до компьютера в ходе работы не превышало трёх метров. Малое расстояние, совместно с использованием витой пары для соединения устройств, позволяет минимизировать искажения сигнала, а также обеспечить скорость передаваемых данных до 1000 Кбит / сек.

## 6.2. Установление соединения с помощью программы VFD Soft

Первым этапом в подключении ПК к ЧП является использование программного обеспечения VFD Soft от производителя частотного преобразователя. Выбор VFD Soft обусловлен удобной функцией авто-подключения, реализованной в данной программе. С помощью авто-подключения, можно узнать текущие параметры протокола для установления соединения.

Для подключения устройств по протоколу Modbus, необходимо определить параметры подключения, которые обязательно должны совпадать на обоих устройствах. Данные характеристики включают:

- Разновидность протокола (Modbus-ASCII, Modbus-RTU)
- Номер COM-порта, к которому подключён проводник, соединяющий ПК с устройством
- Наличие Stopbits и Parity bits в структуре пакетов данных
- Baudrate – скорость передачи данных в Бит / сек, согласованная для обоих устройств.

Функция авто-подключения, реализованная в программе VFD Soft, осуществляет перебор всех упомянутых параметров, с целью успешного установления соединения с устройством по протоколу Modbus RTU или Modbus ASCII на одном из COM-портов ПК.

Пример главного окна программы VFD Soft приведён на Рис. 22. Зелёный цвет индикатора в левом верхнем углу свидетельствует об успешном соединении устройств. После соединения ЧП с ПК с помощью витой пары и адаптера интерфейсов, функция авто-подключения успешно установила соединение с преобразователем Delta CP 2000.

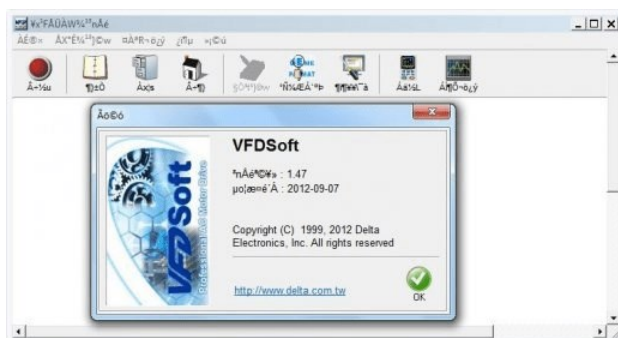


Рис. 22. Главное окно программы VFD Soft, использованной в данной работе

### 6.3. Установление соединения с помощью библиотеки PyModbus

В предыдущей части было установлено, что программное обеспечение VFD Soft позволяет установить соединение между ПК и частотным преобразователем. Так же, данное обеспечение позволяет считывать данные регистров устройства и редактировать их значения. Однако, программа VFD Soft не решает задачи, поставленные в рамках данной работы – динамическое управление регистром выходной частоты, согласно заданной пользователем зависимости.

Для того, чтобы реализовать требуемую функциональность, необходимо самостоятельно разработать программное обеспечение. Для этого в работе используется библиотека PyModbus, реализующая обмен данными с регистрами устройства по протоколу Modbus.

Согласно документации библиотеки, сообщающееся устройство является клиентом, взаимодействующим по протоколу Modbus. Поэтому, для установления соединения, на ПК необходимо создать библиотечный объект-клиент, названный ModbusSerialClient. Сигнатура конструктора класса ModbusSerialClient требует определения следующих параметров:

- Method – разновидность протокола Modbus (RTU, ASCII);
- Port – номер COM-порта ПК, к которому подключено устройство
- Timeout – временной диапазон, в течение которого ModbusSerialClient ожидает отклика от устройства при установлении соединения

- StopBits – параметр, определяющий внутреннюю структуру пакета данных
- Bytesize – размер одного байта передаваемых данных
- Parity – наличие битов чётности во внутренней структуре пакета
- Baudrate – скорость передачи данных в кБит / сек

Параметры подключения должны полностью совпадать на всех общающихся устройствах. В данной работе, параметры протокола Modbus настраивались в соответствии с пользовательским руководством частотного преобразователя. Настройка производилась с помощью пульта управления преобразователем.

После вызова конструктора объекта, для установления соединения необходимо вызвать функцию connect(). Значение, возвращаемое функцией connect – это логическая величина, определяющая успешностью подключения.

В данной работе, результат вызова функции connect() положителен (True), что свидетельствует об успешном подключении с использованием собственной программы.

## **7. Разработка пользовательского приложения для работы с ЧП**

### **7.1. Средства разработки, требования и ограничения**

В качестве программного средства для реализации программного обеспечения (ПО), был выбран интерпретируемый язык программирования Python. Выбор языка Python обусловлен визуальной простотой кода для дальнейших пользователей, поддерживающих ПО. Программа разрабатывается согласно парадигме Объектно-Ориентированного Программирования (ООП). Выбор парадигмы ООП обусловлен простотой расширения функциональности ПО, посредством добавления новых модулей приложения и выделения абстрактных классов. Однако, преимущества ООП-подхода проявляются только при соблюдении пяти принципов разработки SOLID:



1. Принцип одной зоны ответственности (SRP)
2. Принцип открытости/закрытости (OCP)
3. Принцип Подстановки Барбары Лисков (LSP)
4. Принцип разделения интерфейса (ISP)
5. Принцип инверсии зависимостей (DIP)

Согласно перечисленным принципам разработки, от программного обеспечения требуется:

1. Модульность и расширяемость ПО
2. Простота добавления новых видов частотного профиля
3. Выделение общих закономерностей работы ПО в абстрактные классы

## **7.2. Диаграмма юзкейсов программы**

Перед реализацией ПО, согласована необходимая функциональность программы. Главные способы использования ПО представлены на use-case диаграмме (Рис. 23).

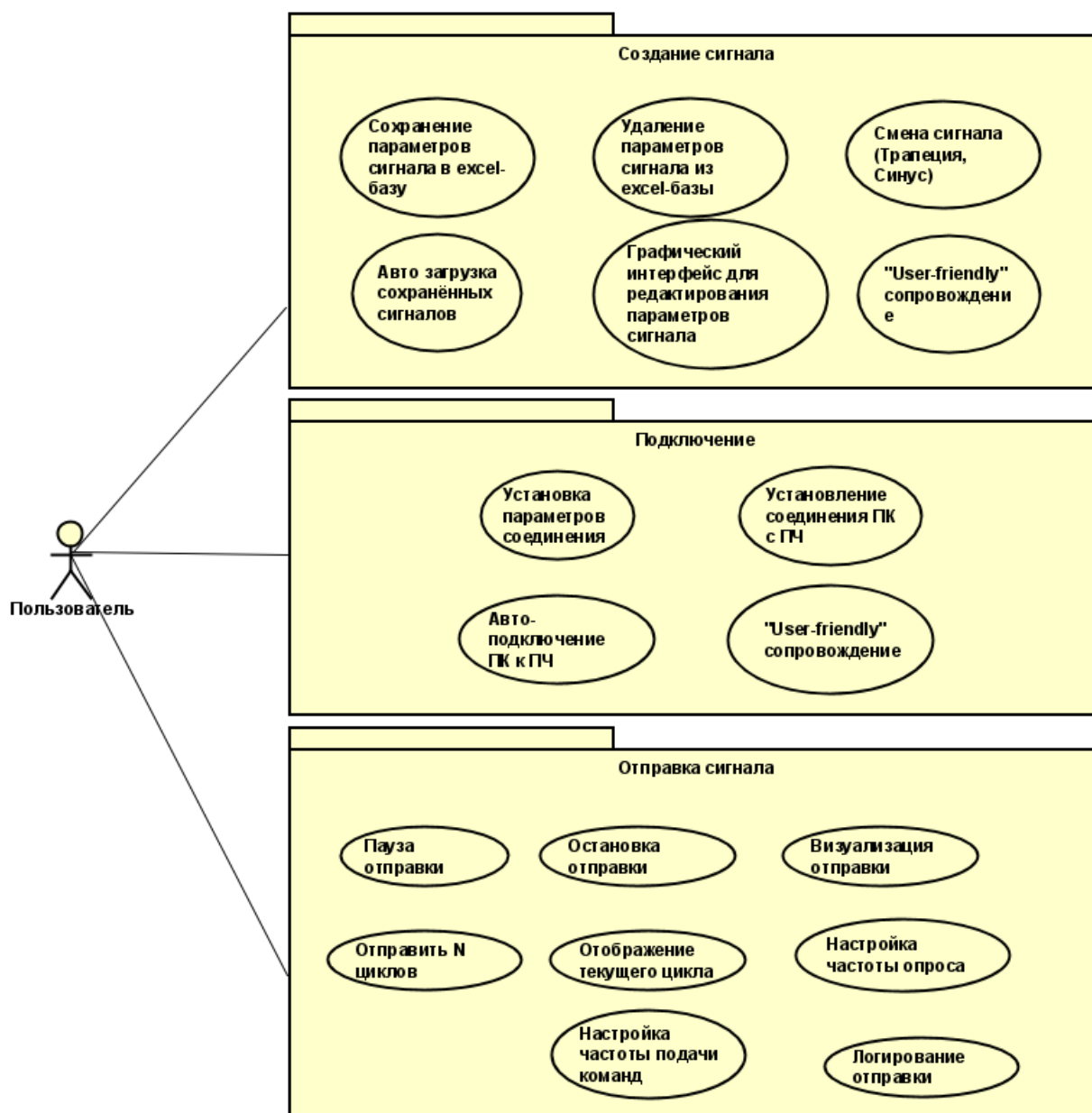


Рис. 23. Use-case диаграмма реализуемого ПО

Представленная на Рис. 23 диаграмма обобщает функциональность, согласованную с сотрудниками лаборатории. Согласно диаграмме, интерфейс программного обеспечения необходимо разделить на несколько модулей:

1. Модуль подключения к частотному преобразователю
2. Модуль редактирования формы профиля
3. Модуль визуализации задания частоты выходного напряжения

Особое внимание требуется уделить интуитивному пользовательскому интерфейсу. Работа с программой должна быть удобной для пользователя, не

знакомого с деталями подключения по протоколу Modbus. Так же, с целью экономии времени, разрабатываемое ПО должно обеспечивать возможность сохранять и загружать предзаданные сигналы (пресеты). В случае некорректной работы с приложением, или сбоев в работе программы, необходимо обеспечить взаимодействие между пользователем и ПО за счёт всплывающих, поясняющих окон («Pop-Up windows»).

С целью реализации user-friendly интерфейса, юзкейс диаграмма на Рис. 23 предусматривает авто-подключение к частотному преобразователю. Так же, юзкейсы включают в себя сохранение, загрузку и удаление предзаданных сигналов. Под юзкейсом «User-friendly сопровождение» подразумевается появление всплывающих окон в случае неудачного подключения либо неверного взаимодействия с пользовательским интерфейсом.

### **7.3. Sequence-диаграммы работы ПО**

Перед тем, как перейти к написанию кода, необходимо графически описать последовательность исполняемых команд. Данный подход использует так называемые sequence-диаграммы – UML-диаграммы, отображающие жизненный цикл объектов, выполняющих определённую задачу. Так же, sequence-диаграмма иллюстрирует последовательность взаимодействия объектов системы.

#### **7.3.1. Задание параметров подключения**

На Рис. 24 представлена sequence-диаграмма работы ПО при подключении компьютера к частотному преобразователю. Диаграмма описывает задание пользователем параметров подключения.

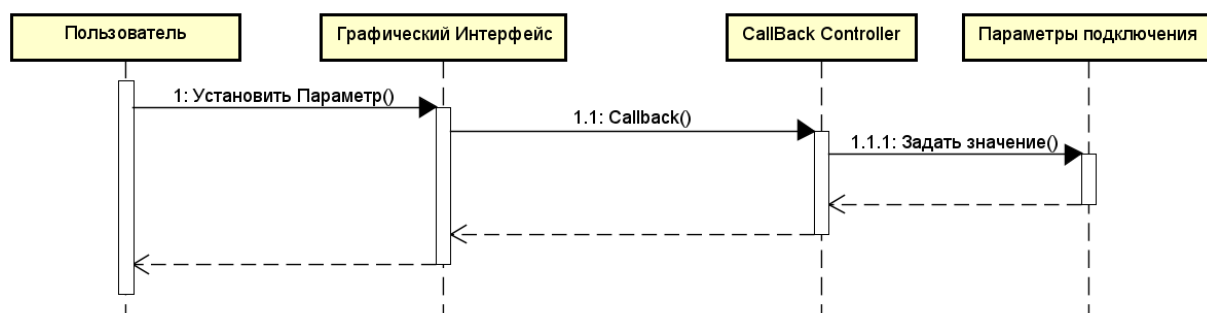


Рис. 24. Диаграмма последовательности для задания параметров подключения по протоколу Modbus

Согласно диаграмме, пользователь задаёт параметр подключения на графическом интерфейсе (например, номер USB-порта). При этом, соответствующий объект графического интерфейса сигнализирует об изменении своего состояния. Согласно принципу разделения ответственности, за реакцию на изменение состояния объекта ответственен определённый класс `CallBackOperator`. Цель экземпляров класса `CallBackOperator` – своевременная реакция на изменение объектов графического интерфейса. Реакция на новое состояние зависит от конкретной задачи. Например, при подключении по протоколу Modbus, реакцией является обновление параметра подключения (Рис. 24). Таким образом, `CallBackOperator` обновляет данные параметра подключения, опираясь на значение, введённое пользователем в окно графического интерфейса.

Уже на данном этапе можно увидеть предпосылки к созданию класса интерфейса, структуры данных параметров подключения, а также абстрактного класса `CallBackOperator`. Наследники класса `CallBackOperator` ответственны за непересекающиеся группы элементов графического интерфейса. Преимущества такой архитектуры заключаются в соблюдении SOLID принципа одной зоны ответственности.

### 7.3.2. Подключение ПК к ЧП

Предыдущая часть описывает взаимодействие объектов ПО при задании параметров подключения. После редактирования параметров подключения, необходимо реализовать соединение ПК с частотным преобразователем по протоколу Modbus. На этапе подключения, все необходимые параметры уже заданы пользователем. Диаграмма на Рис. 25 иллюстрирует последовательность работы ПО после нажатия кнопки подключения.

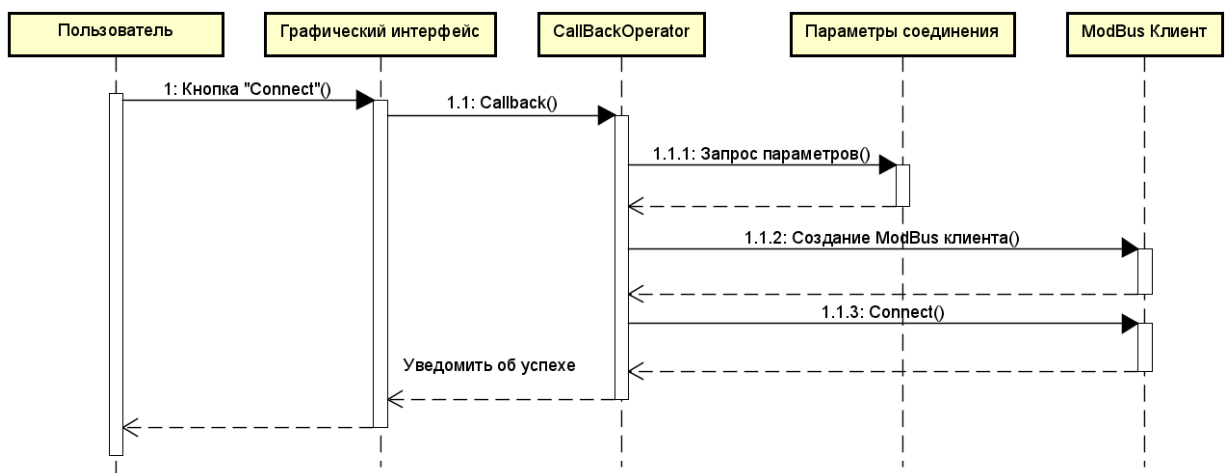


Рис. 25. Диаграмма последовательности для установления соединения ПК с ЧП

Стоит отметить, что `CallbackOperator` – это лишь абстрактный класс. Каждый экземпляр класса `CallbackOperator` переопределяет свою реакцию. Так, на диаграмме (Рис. 25), экземпляр класса `CallbackOperator` реагирует на нажатие пользователем кнопки «Connect» графического интерфейса. Ответ оператора включает себя запрос параметров подключения, создание Modbus клиента с заданными параметрами и установление соединения с помощью функции `Connect()`, реализованной в библиотеке `PyModbus`.

Для интуитивной работы с программой, результат работы сопровождается всплывающим окном, сигнализирующим об успешном подключении. В противном случае, всплывающее окно предлагает попробовать другой набор параметров.

Для успешного обмена данными с частотным преобразователем, необходимо соединение с единственным экземпляром класса ModBus клиента. С другой стороны, к методам клиента должны получить доступ все операторы, взаимодействующие с частотным преобразователем. С целью обеспечения единственного экземпляра класса клиента использован паттерн Singleton.

Паттерн Singleton позволяет каждому из операторов использовать локальную «копию» клиента, гарантируя при этом его единственность.

### 7.3.3. Задание постоянной частоты ЧП

На Рис. 26 приведена диаграмма последовательности для модуля программы, ответственного за задание постоянной частоты. Пользовательский интерфейс копирует функциональность управляющего пульта – графическое окно содержит текстовое поле со слайдером для задания частоты, а также кнопки Set Frequency, Start и Stop.

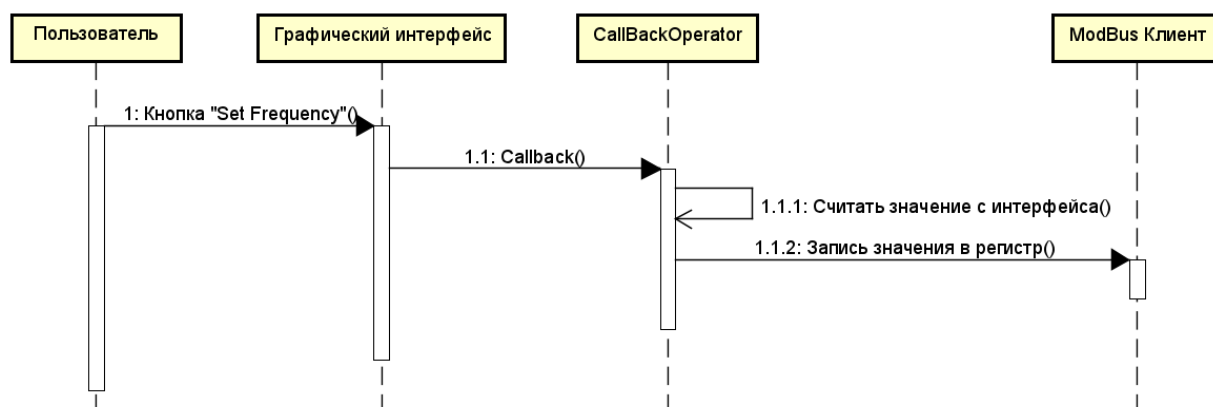


Рис. 26. Диаграмма последовательности для задания постоянной частоты на частотном преобразователе

В примере на Рис. 26 проиллюстрирована работа программы в случае нажатия кнопки Set Frequency. Стоит заметить, на этапе задания частоты соединение Modbus клиента с частотным преобразователем уже установлено (Рис. 25). Далее, переопределённый класс CallbackOperator агрегирует клиента, установившего соединение с преобразователем. Задание частоты

осуществляется с помощью снятия значения с графического интерфейса и записи в соответствующий регистр преобразователя.

### 7.3.4. Создание сигнала

Следующим этапом после подключения становится редактирование трапецевидного частотного профиля (Рис. 27). Редактирование сигнала реализовано с помощью объектно-ориентированного паттерна Model-View-Controller.

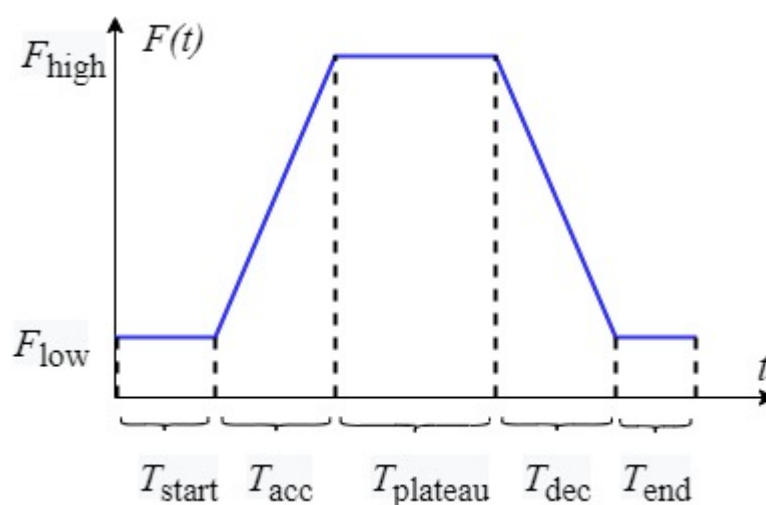


Рис. 27. Пример трапецевидного сигнала. Промежутки времени, обозначенные на рисунке, соответствуют старту, разгону, плато, замедлению и конечному периодам

Для соблюдения принципа открытости-закрытости, необходимо реализовать архитектуру программы, открытую для расширения (добавления новых типов сигналов), но закрытую для модификации уже имеющихся сигналов. Для этого, класс контроллера (Controller) является абстрактным. При добавлении нового типа сигнала, необходимо реализовать новый класс контроллера.

Класс контроллера агрегирует графический интерфейс для создания сигнала, обновляемую модель данных (Model) и визуальное представление модели для пользователя. Помимо этого, Controller агрегирует набор

операторов, реагирующих на действия пользователя и обновляющих параметры модели.

Аналогично, для соблюдения принципа открытости-закрытости, класс модели так же является абстрактным. Модель содержит в себе всю необходимую информацию о параметрах сигнала – минимальной и максимальной частоте и длинах всех временных промежутков (Рис. 27). При изменении внутренних параметров, модель оповещает класс View о необходимости обновить представление модели.

Таким образом, последовательность работы программы при редактировании сигнала показана на Рис. 28. Основные этапы включают в себя:

1. Действие пользователя
2. Реакция соответствующего класса CallbackOperator, изменение модели
3. Оповещение моделью класса View
4. Обновление классом View представления модели

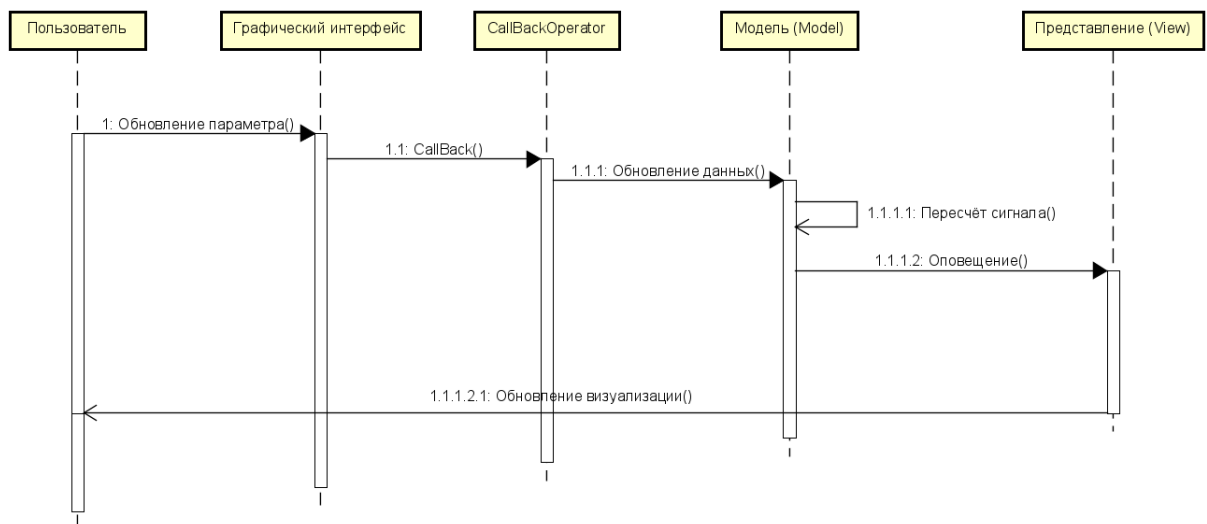


Рис. 28. Диаграмма последовательности для редактирования сигнала



### 7.3.5. Отправка сигнала

В предыдущем пункте было отмечено, что редактирование сигнала реализовано с помощью паттерна проектирования Model-View-Controller. При этом, класс контроллера (Controller) агрегирует:

1. Класс графического интерфейса для отображения сигнала
2. Класс, содержащий данные сигнала (Model)
3. Класс, ответственный за визуальное представление
4. Набор классов CallBackOperator, связывающих взаимодействие пользователя с графическим интерфейсом и обновление данных сигнала

Последним классом, содержащимся в контроллере, является оператор, ответственный за отправку сигнала, то есть за своевременное задание выходной частоты переменного напряжения.

На Рис. 27 представлена зависимость выходной частоты ЧП от времени, заданная пользователем приложения. Отправка сигнала включает в себя использование таймера для своевременного задания нужных значений по протоколу Modbus. Для снижения временных задержек, механизм отправки сигнала построен на принципе прерываний.

Под прерыванием подразумевается сигнал таймера о наступлении высокоприоритетного события. Высокоприоритетным событием является задание выходной частоты либо опрос регистра истинной частоты. Например (Рис. 27), после задания начальной частоты, с помощью таймера засекается время старта  $t_{start}$ . По истечении промежутка времени  $t_{start}$ , наступает момент для задания следующего значения выходной частоты. Следовательно, в момент завершения  $t_{start}$  таймер вызывает специальную функцию – обработчик прерывания (Рис. 29).

Обработчик прерывания добавляет новую задачу на опрос, либо на задание частоты напряжения. Одновременно с основным потоком исполнения, осуществляющим замер времени по таймеру, параллельный поток отслеживает и выполняет задачи, добавленные в очередь (Рис. 29).

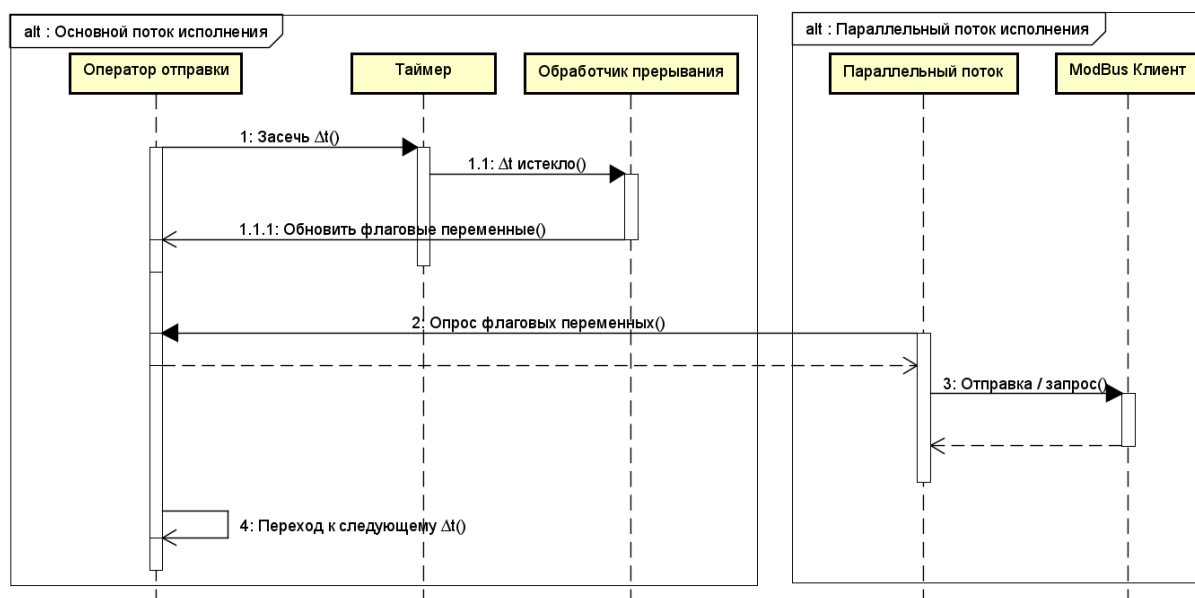


Рис. 29. Диаграмма последовательности для механизма прерываний  
(алгоритм отправки сигнала)

## 7.4. Архитектура ПО

На этапе создания архитектуры программы, необходимо соблюдать принципы SOLID объектно-ориентированного программирования. Для того, чтобы обеспечить расширяемость программного обеспечения, важным аспектом является модульность приложения. Каждый из разработанных модулей обязан соблюдать принцип одной зоны ответственности.

В связи с этим, центральным классом ПО является *ApplicationManager*, агрегирующий несколько модулей с непересекающимися зонами ответственности:

1. Модуль подключения ПК к частотному преобразователю
2. Модуль эмуляции управляющего пульта (задание постоянной частоты)

3. Модуль создания и редактирования частотного профиля
4. Модуль задания целевых частот по таймеру
5. Модуль визуализации лог-файлов

Каждый из реализованных модулей наследуется от абстрактного класса `ApplicationModule` (Рис. 30). Таким образом, каждый наследник абстрактного класса содержит свой личный набор объектов, определяющих интерфейс абстрактного класса `CallBackOperator`. Данная архитектура соблюдает принцип разделения интерфейсов и принцип одной зоны ответственности, а также упрощает расширение ПО за счёт добавления новых модулей.

Класс `CallBackOperator` ответственен за связь взаимодействия пользователя с графическим интерфейсом и реакцию ПО на действия пользователя. Абстрактным методом класса `CallBackOperator` является `ConnectCallBack()`. Следовательно, каждый наследник переопределяет реакцию ПО на взаимодействие пользователя с определёнными виджетами графического интерфейса.

Поскольку каждый модуль содержит набор операторов, ответственных за разные виджеты графического интерфейса, данная архитектура подчиняется принципам разделения интерфейса, открытости-закрытости и одной зоны ответственности. Также, при добавлении новых пользовательских виджетов, упрощается расширение ПО, за счёт добавления новых объектов `CallBackOperator`, вместо внесения изменений в уже отлаженный код.

Помимо выбора архитектуры, в ходе разработки ПО, возникла необходимость в использовании паттернов объектно-ориентированного программирования. Так, поскольку Modbus-клиент должен быть единственным, класс `ModBusClient` реализован с помощью паттерна `Singleton`. Модуль создания сигнала реализован с помощью паттерна `Model-View-Controller`.

На Рис. 30 показана итоговая диаграмма классов ПО на уровне абстрактных классов-интерфейсов.

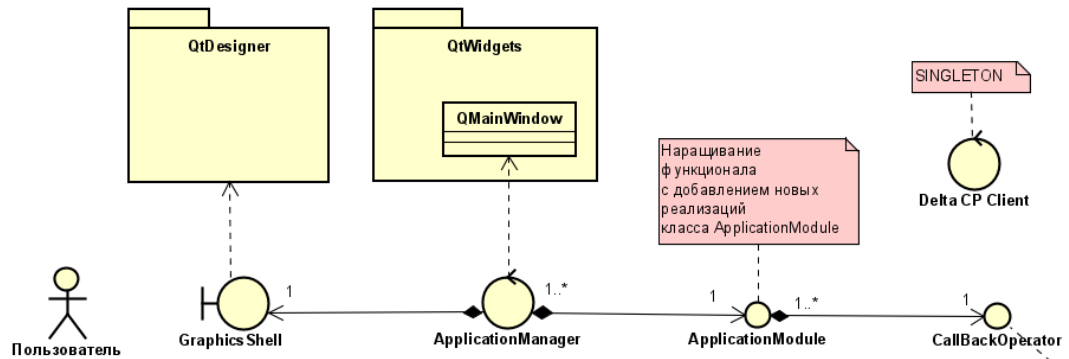


Рис. 30. Диаграмма абстрактных классов ПО

## 8. Результаты, детали реализации ПО

### 8.1. Метрики эффективности ПО

Чтобы оценить эффективность разработанного ПО, необходимо минимизировать две метрики. Первой метрикой является средняя временная задержка на цикл:

$$\Delta t_{\text{delay}} = \frac{1}{N_{\text{cycles}}} \sum_{i=1}^{N_{\text{cycles}}} (\Delta t_{\text{cycle real}} - \Delta t_{\text{cycle expect}}) \rightarrow \min. \quad (8)$$

Действительно, в качестве средства разработки был выбран интерпретируемый язык Python. В отличие от компилируемых языков программирования (C, C++ и др.), исполнение кода в интерпретируемых языках осуществляется построчно, что негативно сказывается на времени исполнения кода. Поэтому, необходимо компенсировать временные издержки, накапливающиеся от цикла к циклу в результате исполнения кода.

Вторая метрика касается соответствия реального и ожидаемого частотных профилей. Реальный частотный профиль обладает характерной кривизной начальных участков разгона и замедления. Форму участков можно

редактировать с помощью параметров преобразователя, ответственных за сглаживание фронтов кривой разгона и замедления.

Поэтому, для измерения соответствия профилей предлагается синхронизировать кривые по времени, после чего рассчитать среднеквадратичное отклонение, как меру схожести форм сигнала:

$$\sigma = \sqrt{\frac{1}{N_{\text{cycles}}} \sum_1^{N_{\text{cycles}}} (F_{\text{real}} - F_{\text{expect}})^2} \rightarrow \min. \quad (9)$$

## 8.2. Методы снижения метрик

### 8.2.1. Стандартное отклонение сигнала

Для постановки эксперимента в условиях постоянной частоты напряжения, частотный преобразователь был настроен на автоматический режим разгона. Автоматический режим предполагает как можно более быстрый разгон до частоты, заданной на пульте управления. Однако, такой режим работы не соответствует желаемой форме частотной зависимости. В автоматическом режиме форма реального профиля значительно расходится с желаемой, что соответствует максимальному среднеквадратичному отклонению  $\sigma = 7,94$ .

Согласно пользовательскому руководству [18], преобразователь поддерживает режим линейного разгона, что соответствует желаемой форме профиля. При этом, скорость достижения заданной частоты регулируется с помощью нескольких регистровых переменных:

1. Регистр минимальной частоты ( $F_{\text{max}}$ )
2. Регистр максимальной частоты ( $F_{\text{min}}$ )
3. Регистр времени разгона от минимальной до максимальной частот ( $t_{\text{acc}}$ )
4. Регистр времени замедления от максимальной до минимальной частот ( $t_{\text{dec}}$ )

Для корректного использования режима линейного разгона, перед отправкой необходимо произвести пересчёт регистровых переменных времени разгона и замедления.

Обозначим за  $a_{\text{goal}}$  целевое ускорение, заданное пользователем на этапе создания сигнала, а за  $a_{\text{current}}$  – аппаратное ускорение преобразователя, настроенное на данный момент. Так же, обозначим за  $F_{\text{high}}$  и  $F_{\text{low}}$  минимальную и максимальную частоты сигнала, заданного пользователем на графическом интерфейсе ПО. Стоит отметить, что регистры минимальной и максимальной частоты оставались постоянными и равны 0 и 50 Гц соответственно.

$$a_{\text{goal}} = \frac{F_{\text{high}} - F_{\text{low}}}{t_{\text{acc.goal}}} \quad (10)$$

$$a_{\text{current}} = \frac{F_{\text{max}} - F_{\text{min}}}{t_{\text{acc.current}}} \quad (11)$$

Необходимо, чтобы целевое ускорение соответствовало действительному. Потому, из равенства ускорений получены формулы для пересчёта регистровых переменных времён ускорения и замедления:

$$t_{\text{acc.current}} = \frac{(F_{\text{max}} - F_{\text{min}}) \times t_{\text{acc.goal}}}{F_{\text{high}} - F_{\text{low}}} \quad (12)$$

$$t_{\text{dec.current}} = \frac{(F_{\text{max}} - F_{\text{min}}) \times t_{\text{dec.goal}}}{F_{\text{high}} - F_{\text{low}}}. \quad (13)$$

Следующим шагом стала оптимизация S-кривой разгона. Согласно пользовательскому руководству [18], есть ряд параметров, ответственных за сглаживание трапецевидной кривой разгона (Рис. 31). Поставленная задача не предполагает сглаживание частотного профиля, поэтому параметры сглаживания частотной зависимости установлены в ноль.

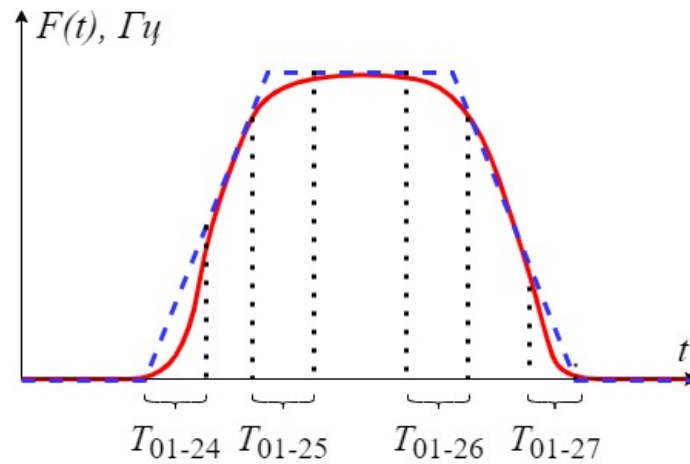


Рис. 31. Вид S-кривой частотного профиля. Параметры частотного преобразователя 01-24, 01-25, 01-26, 01-27 преобразователя регулируют длительность сглаженных участков

В результате, принятые меры по устранению несоответствия реальной желаемой форм частотного профиля снизили стандартное отклонение на 92% (Рис. 32). В результате, экспериментальный профиль с нулевым S-сглаживанием и линейным разгоном наиболее соответствует заданному пользователем (Рис. 33).

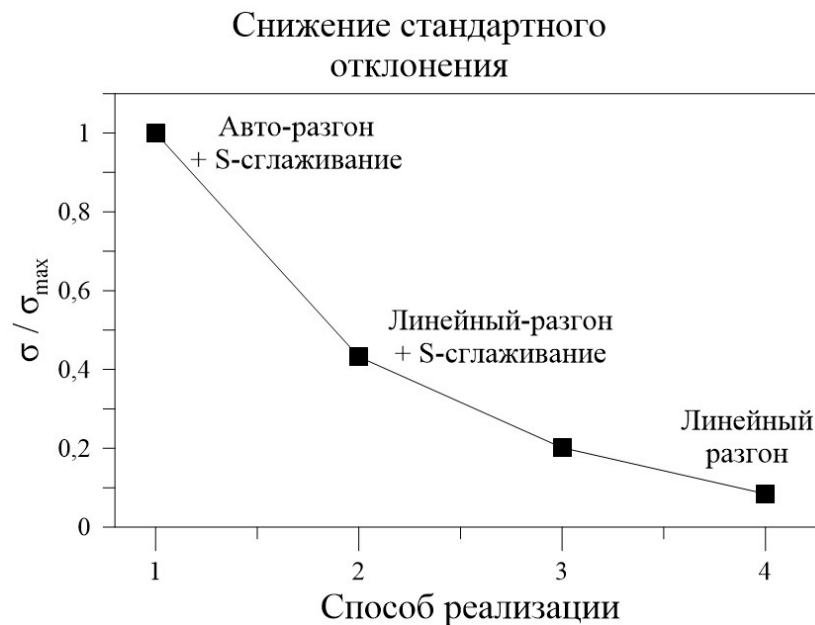


Рис. 32. Нормированное среднеквадратичное отклонение.

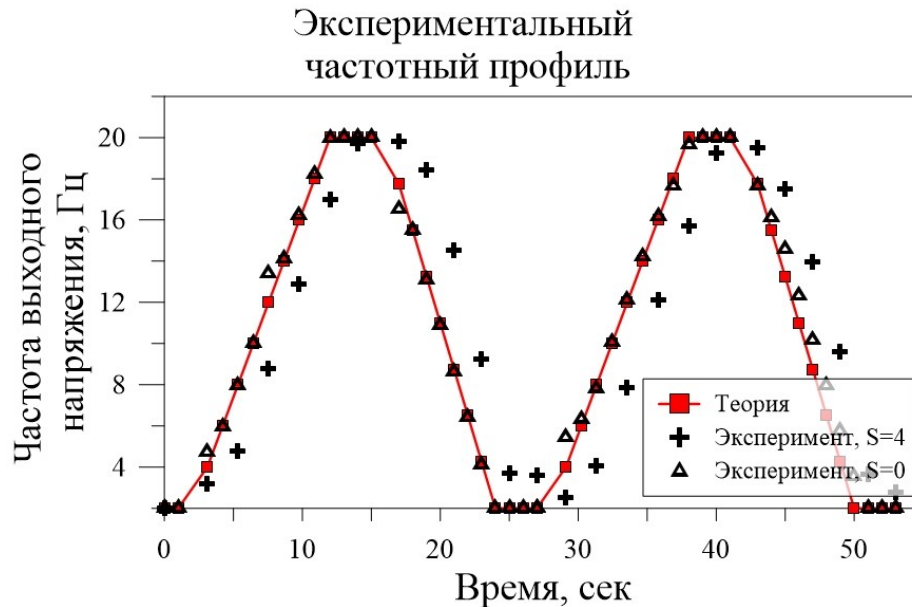


Рис. 33. Ожидаемый и реальные частотные профили, соответствующие различным методам снижения стандартного отклонения. Параметр  $S$  соответствует длительности периода сглаживания в секундах

### 8.2.2. Временные задержки

Алгоритм отправки сигнала реализован на основе принципа прерываний (См. часть 5.3.4. «Отправка сигнала»).

В случае однопоточной реализации, обработчик прерывания выполняет опрос и редактирование регистров в основном потоке исполнения. Это приводит к существенным временным задержкам, накапливающимся с каждым циклом работы программы. Установлено несколько основных причин временных задержек.

Во-первых, работа программы в интерпретируемом языке Python осуществляется построчно, при этом исполнение каждой строки кода требует некоторого времени. Помимо редактирования регистровых переменных, исполнение кода включает в себя отображение графического окна, пересчёт массива времени, обновление визуализации отправки сигнала.

Во-вторых, с помощью логирования работы программы было установлено, что обмен данными по протоколу Modbus является самой длительной



операцией. Команды задания частоты могут занимать до двух секунд, а опрос регистра в среднем занимает 0,13 секунды (Рис. 34). Поэтому, длительное исполнение обработчика прерываний в основном потоке приводит к накапливающейся временной задержке (Рис. 35) и несоответствию расчётного и реального времени цикла (Рис. 36).

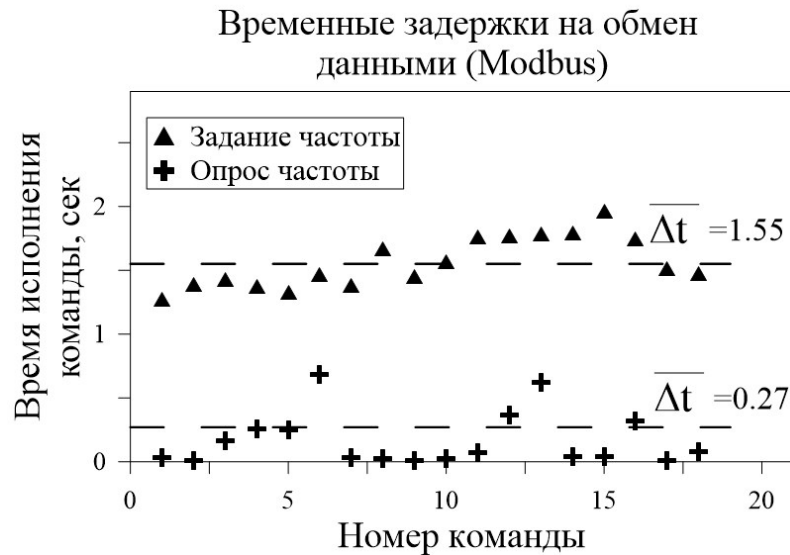


Рис. 34. Время, необходимое для исполнения кода по заданию выходной частоты и опросу регистра текущей частоты преобразователя

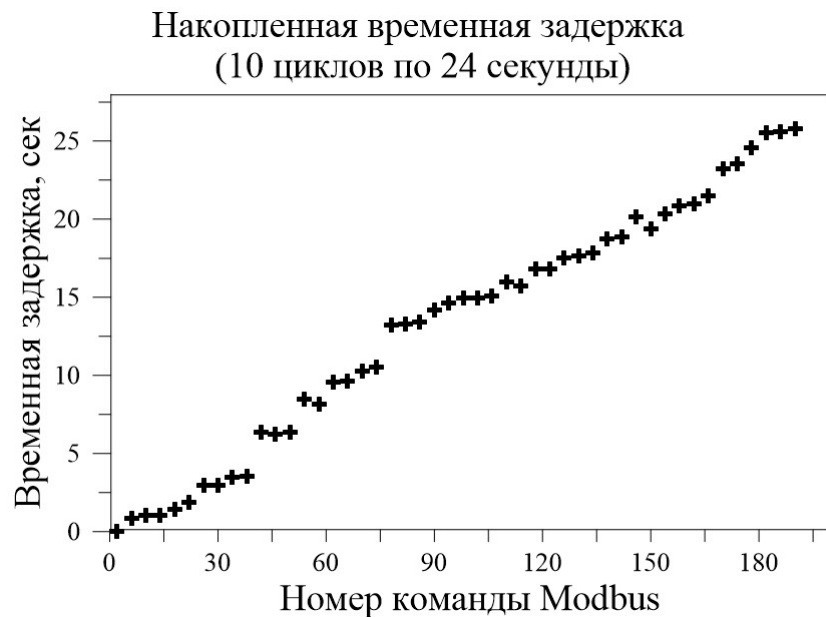


Рис. 35. Временная задержка, накопленная в результате отправки десяти циклов по 24 секунды

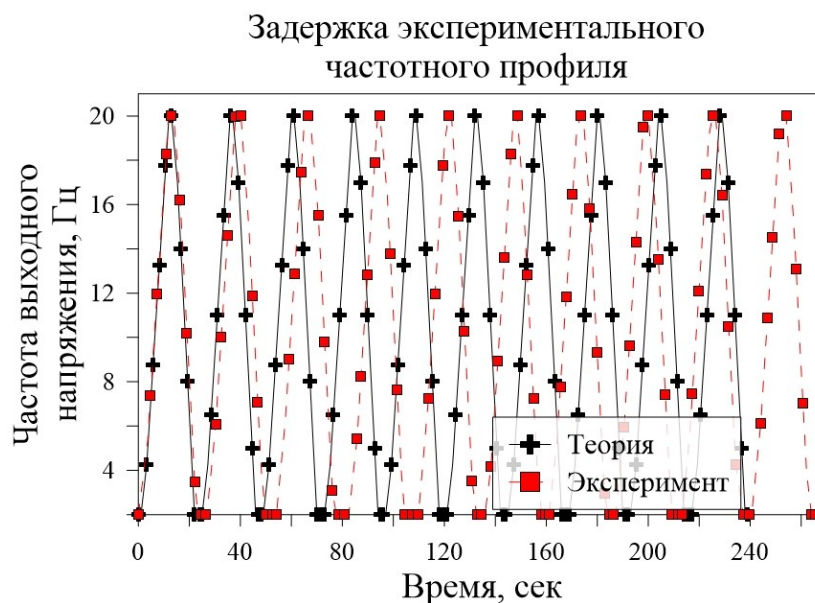


Рис. 36. Пример запаздывания реального профиля в следствие временных задержек

Для того, чтобы сократить время на обработку прерывания, в работе предложена многопоточная реализация отправки. Основной поток выполняется согласно уже описанному алгоритму, основанном на прерываниях таймера (Рис. 29). Однако, в новой реализации, обработчик прерывания не исполняет опрос и редактирование регистров частотного преобразователя. Вместо этого, срабатывая по прерыванию, обработчик прерывания добавляет новую задачу в специальную очередь (Queue).

Очередь является FIFO («First In First Out») структурой данных. Таким образом, оператор, ответственный за отправку сигнала, агрегирует несколько задач на редактирование и опрос регистров, ожидаемых решения. Сложность добавления элемента в очередь и удаления элемента постоянна ( $O(1)$ ), что гарантирует быстрое завершение работы обработчика прерывания. В результате, среднее время обработчика прерываний сократилось до 0,005 секунд.

Одновременно с основным потоком исполнения, параллельный поток (Рис. 29) исполняет бесконечный цикл, отслеживая состояние очереди задач на опрос и редактирование регистровых переменных. В случае непустой

очереди, параллельный поток выполняет все поступающие задачи, а также выполняет визуализацию отправки сигнала.

В дополнение, несмотря на длительную коммуникацию по протоколу Modbus, параллельный поток гарантирует, что исполнит вновь поступившие немедленно, не задерживая при этом работу таймера в основном потоке исполнения.

Помимо этого, с целью компенсировать задержки из-за выполнения кода, в конце каждого цикла отправки, расчётное время сравнивается с реальным с помощью системного таймера. В случае, если реальное время превосходит расчётное, полученная разница делится на количество команд в следующем цикле:

$$\Delta t_{\text{delay}} = \frac{t_{\text{real}} - t_{\text{expect}}}{N_{\text{commands}}}. \quad (14)$$

В течение следующего цикла, каждая команда подаётся на  $\Delta t_{\text{delay}}$  ранее. Такой подход позволяет противостоять временным задержкам на обновление графического окна и построчное исполнение кода. Данный способ позволил снизить временную до 0,08 секунд на цикл, что на 97% ниже начальных показателей. Более того, с помощью пересчёта величины  $\Delta t_{\text{delay}}$  с каждым циклом, устранён эффект накапливания задержки (Рис. 37).

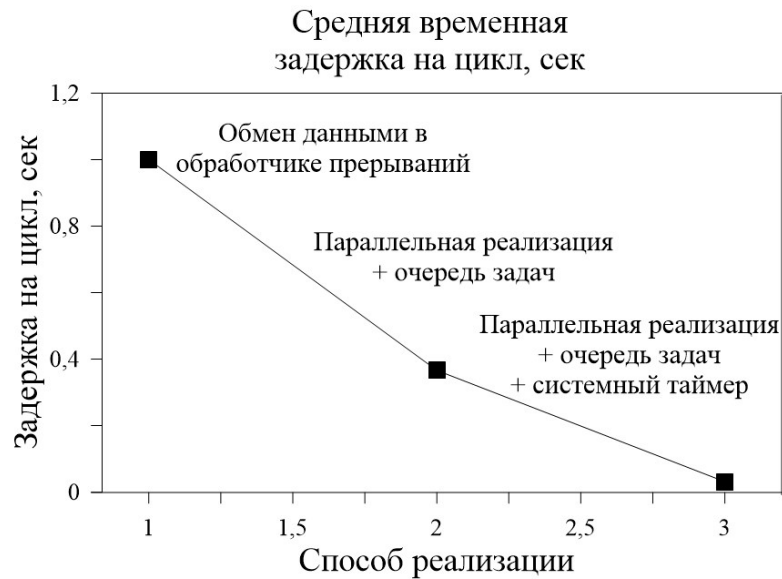


Рис. 37. Результаты снижения средней временной задержки на цикл

Для оценки суммарной временной задержки, проведено тестирование последней версии алгоритма – с очередью задач и поправкой на цикл. Для тестирования выбрано два режима (быстрый и медленный), различающиеся временем разгона и замедления (Табл. 1).

**Таблица 1.** Параметры тестируемых режимов.

Тип режима	Минимальная частота, Гц	Максимальная частота, Гц	Время разгона, сек	Время замедления, сек
Быстрый	0	20	10	8
Медленный	0	20	15	15

Каждый из режимов был протестирован на экспериментальном стенде в течение нескольких периодов, от 20 минут до 1 часа работы. Результаты измерений представлены на Рис. 38. Таким образом, суммарная временная задержка не превосходит 1 секунды, вне зависимости от выбора режима разгона и количества циклов работы программы.

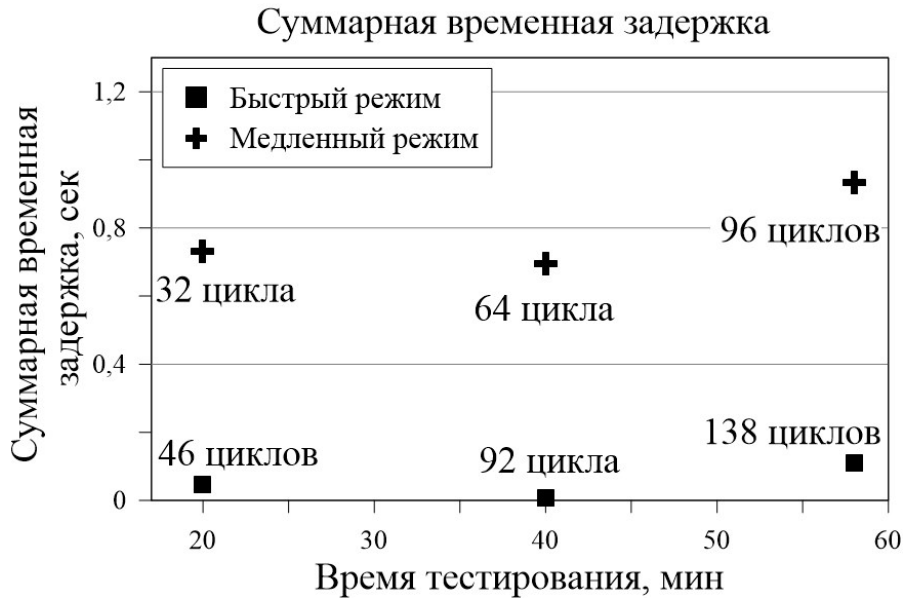


Рис. 38. Суммарная временная задержка, накопленная в результате работы алгоритма

Последним шагом является получение расходно-частотной характеристики преобразователя, для перевода частотной зависимости  $F(t)$  в профиль объёмного расхода жидкости  $Q(t)$ .

Необходимо установить, по какому закону расход жидкости, протекающей через проточный тракт, связан с частотой выходного напряжения преобразователя. Для этого, с помощью реализованного ПО, были заданы выходные частоты от 5 до 45 Гц, с шагом 5 Гц, и измерены соответствующие показания электромагнитного расходомера. Согласно полученной расходно-частотной характеристике на Рис. 39, установлена линейная связь между частотой выходного напряжения и расходом жидкости. Следовательно, после перевода частотной зависимости  $F(t)$  в профиль расхода  $Q(t)$ , форма профиля сохранится.

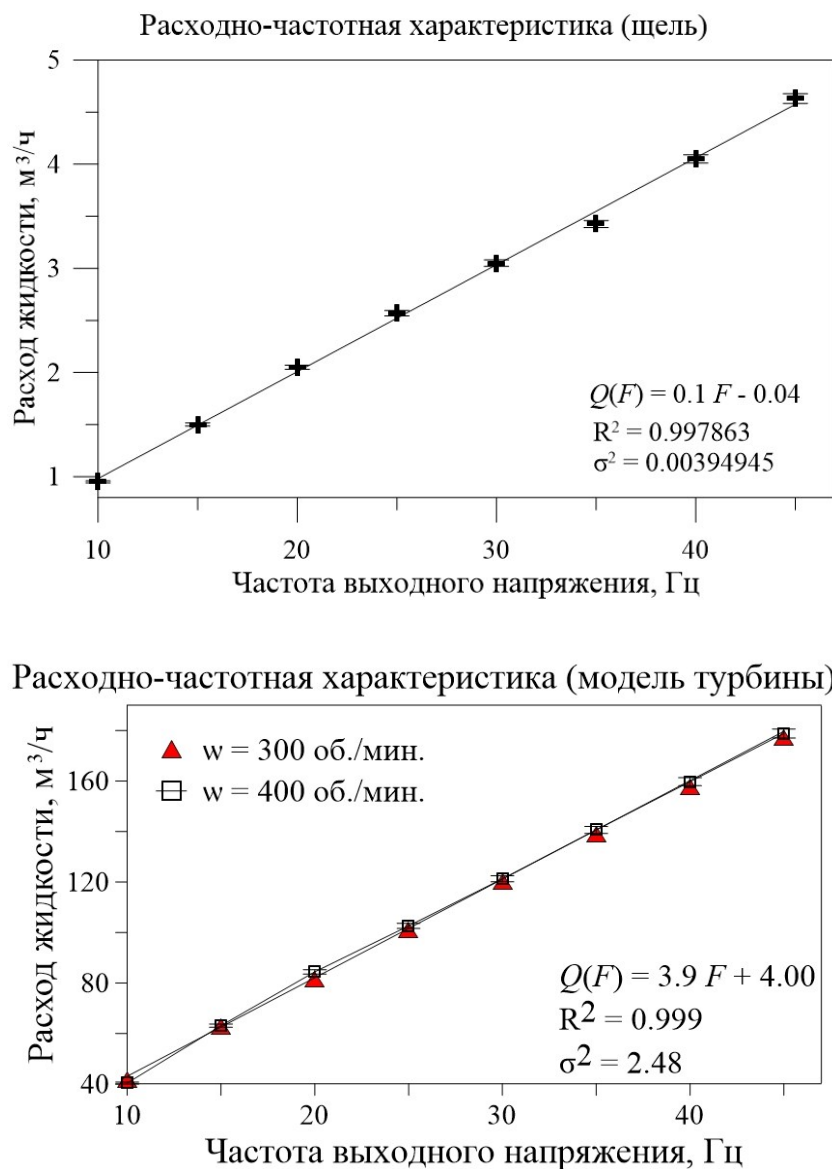


Рис. 39. Расходно-частотная характеристика кавитационного стенда для щели и модели гидротурбины. Частота вращения рабочего колеса турбины обозначена как  $\omega$ .  
Погрешность измерений расходомера составляет 1% от измеряемого значения

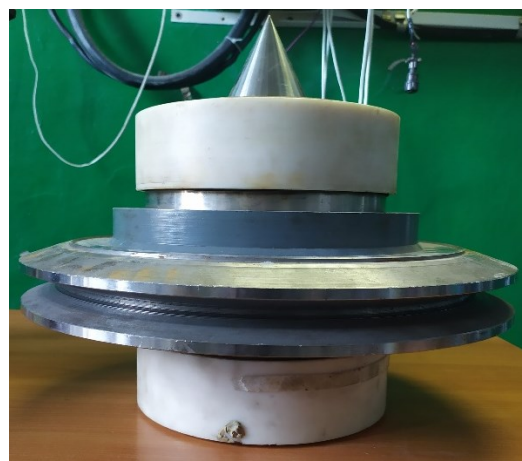
## 9. Экспериментальная визуализация переходных режимов

Разработанное программное обеспечение было использовано для визуализации переходных режимов работы гидротурбины. В рабочем участке кавитационного стенда располагается модель гидротурбины со стационарным лопаточным завихрителем, закручивающем поток жидкости (Рис. 40 (а)). Помимо лопаточного завихрителя, в конструкцию турбины входит вращающееся рабочее колесо (Рис. 40 (б)). Вал рабочего колеса

турбины соединён с валом электродвигателя посредством ременной передачи. Таким образом, скорость вращения рабочего колеса контролируется электродвигателем, управляемым частотным преобразователем фирмы АВВ.



(а)

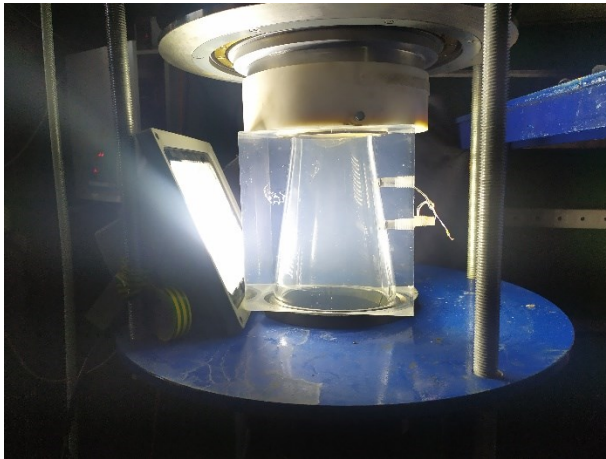


(б)

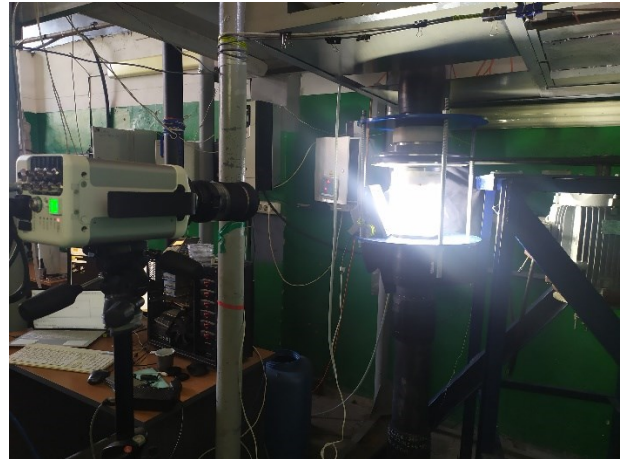
Рис. 40. (а) - Модель рабочего колеса гидротурбины (вид сверху) со стационарным лопаточным завихрителем. (б) – Рабочее колесо модели гидротурбины (вид сбоку)

Исследуемым объектом является модель отсасывающей трубы, расположенная за рабочим колесом турбины (Рис. 41 (а)). Для повышения контрастности снимков, модель отсасывающей трубы была освещена с помощью прожектора. Визуализация переходных режимов проводилась с помощью скоростной камеры NOVA S12 (Рис. 41 (б)).

Параметры съёмки, а также параметры трапециевидного профиля расхода, выбранного для экспериментальной визуализации, приведены в таблицах 2 и 3 соответственно. Стоит отметить, переходные режимы реализуются на участках динамического изменения расхода, т. е. перехода от минимального к максимальному расходу. Поскольку на частоте в 2000 кадров в секунду, скоростная камера позволяет провести съёмку видео длительностью в 10,91 секунды, длительность участков разгона и замедления была установлена на 10,9 секунд.



(a)



(б)

Рис. 41. Рабочий участок кавитационного стенда. (а) – Модель отсасывающей трубы гидротурбины, подсвечиваемая с помощью прожектора. (б) – Скоростная камера, используемая для съёмки переходных режимов работы турбины

**Таблица 2.** Параметры съёмки.

Количество кадров в секунду	Разрешения, пикс.	Количество кадров	Длительность съёмки, сек
2000	1024 × 1024	21821	10,91

**Таблица 3.** Параметры трапециевидного профиля расхода.

Время разгона, сек	Время замедления, сек	Минимальный расход, м <sup>3</sup> /час	Максимальный расход, м <sup>3</sup> /час
10,9	10,9	0	179,5

Результатом съёмки является серия снимков прецессирующего вихревого ядра, развивающегося в конусе отсасывающей трубы, в период роста расхода жидкости от 0 м<sup>3</sup>/час до 179,5 м<sup>3</sup>/час за 10,9 секунд (Рис. 42, Рис. 43).



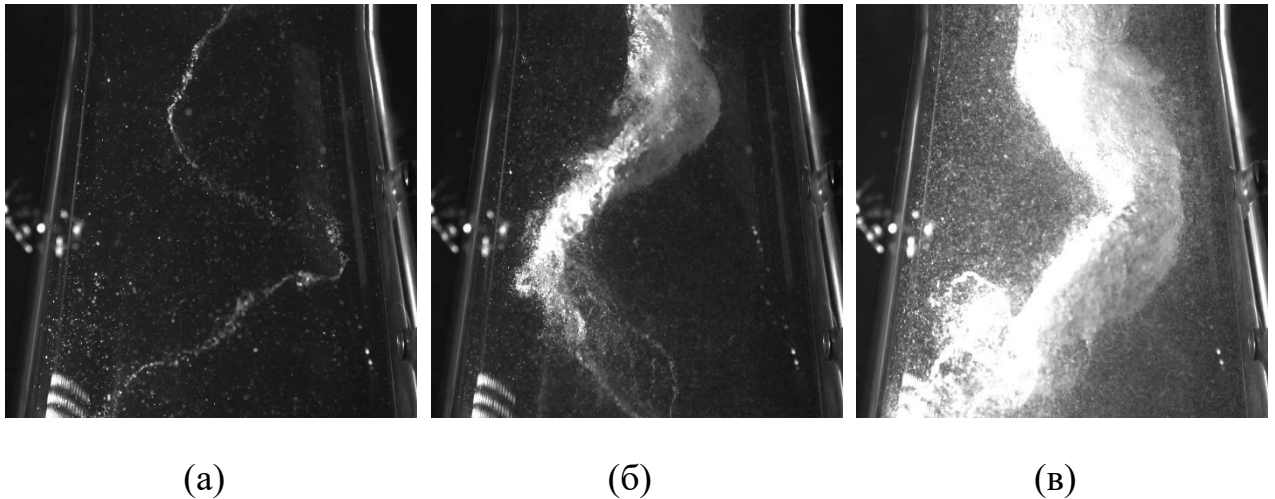


Рис. 42. Развитие прецессирующего вихревого ядра в отсасывающей трубе модели гидротурбины. Кадры (а), (б) и (в) соответствуют началу, середине и концу участка разгона на профиле расхода (Рис. 43)

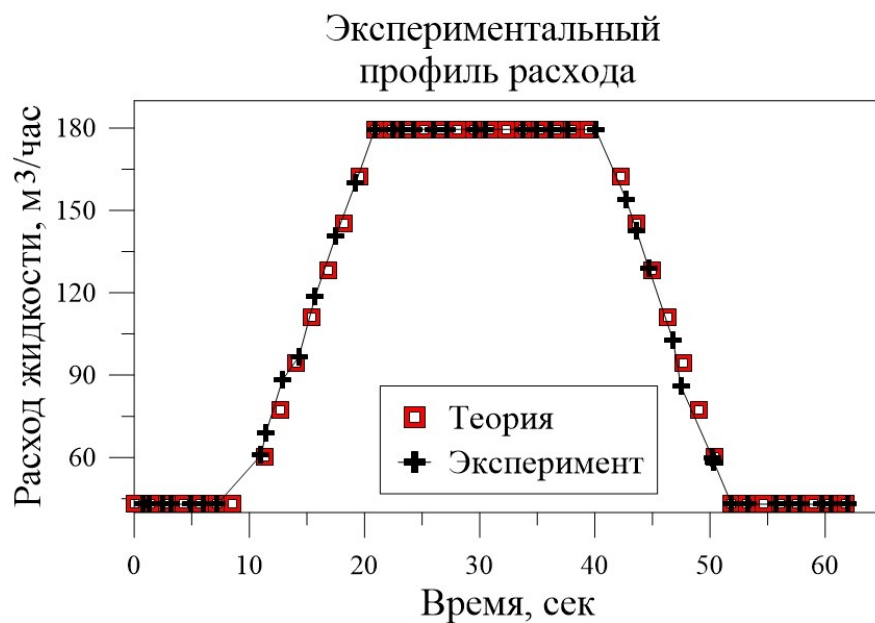


Рис. 43. Профиль расхода для экспериментальной визуализации переходных режимов работы турбины. Участки разгона и замедления соответствуют росту и снижению расхода жидкости от 0 до 179,5 м³/час

## 10. Заключение

Целью работы является реализация программного обеспечения для автоматизации управления расходом на напорном кавитационном стенде. Актуальность работы заключается в том, что динамическое управление

расходом на кавитационном стенде позволит экспериментально исследовать переходные неоптимальные режимы работы моделей турбин, а также верифицировать численные методы математического моделирования потока.

Для достижения поставленной цели, проведён обзор существующих программных решений. Было установлено, что реализованные решения не обеспечивают функциональность для автоматизированного задания целевых значений. Следовательно, основной задачей является реализация программного обеспечения.

В ходе разработки ПО, предложена поддерживаемая и расширяемая SOLID архитектура программы, изучены библиотеки PyModbus, PyQt для обмена данными по протоколу Modbus и реализации пользовательского интерфейса. Для повышения эффективности ПО, использованы такие метрики, как временная задержка алгоритма и стандартное отклонение теоретического и экспериментального профилей.

В ходе работы, метрики эффективности ПО минимизированы с помощью специальных регистров частотного преобразователя, а также многопоточной реализации задания целевых значений. Устранение сглаживания S-кривой разгона и использование линейного режима разгона позволили сократить стандартное отклонение на 92%. Благодаря многопоточной реализации с очередью задач, временная задержка на исполнение кода сокращена на 97% и не превосходит 1,2 секунд, вне зависимости от количества циклов работы алгоритма.

Реализованное ПО использовано для экспериментальной визуализации переходных режимов работы турбины. Дальнейшая разработка приложения заключается в расширении набора частотных профилей. Практическая ценность работы состоит в возможности динамического изменения расхода жидкости для исследования вихревых нестационарных течений. Реализованное ПО внедрено в лаборатории эффективного использования реакторных установок.

## 11. Список литературы

1. IRENA, Renewable Energy Statistics: Global Status Repot, Int. Renew. Energy Agency. (2019).
2. S. Cherny, D. Chirkov, D. Bannikov, V. Lapin, V. Skorospelov, I. Eshkunova, A. Avdushenko. 3D numerical sinulation of transient processes in hydraulic turbines // IOP Conference Series: Environmental Science 12, 012071, 2010
3. A. Ruprecht, T. Helmrich, T. Aschenbrenner, T. Scherer. Simulation of vortex rope in a turbine draft tube // Proceedings of the Hydraulic Machinery and Systems 21<sup>st</sup> IAHR Symposium, Lausanne, September 9-12, 2002
4. А. Ю. Авдюшенко, С. Г. Черный, Д. В. Чирков, В. А. Скороспелов, П. А. Турук. Численное моделирование переходных процессов в гидротурбинах // Теплофизика и аэромеханика, том 20, №5, 2013
5. С. Г. Скрипкин, М. А. Цой, С. И. Шторк, П. А. Куйбин. Исследование пульсаций давления в модельной отсасывающей трубе гидротурбины, вызванных вихревыми кольцами // Вестник НГУ, Серия: Физика, т. 11, №4, 2016
6. Причины аварии на Саяно-Шушенской ГЭС. Выводы Ростехнадзора, основные тезисы 3 октября 2009 г.
7. I. K. Vilberg, M. Kjeldsen, X. Escaler, J. V. Ekanger, T. K. Nielsen. Influence of draft tuve water injection system on cavitation begaviour in a full-scale Francis turbine with visual access // IOP Conference Series: Earth and Environmental Science, 2019
8. C. Trivedi, P. J. Gogstad, O. G. Dahlhaug. Investigation of the unsteady pressure pulsations in the prototype Francis turbines during load variation and startup // Journal of Renewable and Sustainable Energy, 9, 064502, 2017
9. J. Nicolle, A. M. Giroux, J. F. Morissette. CFD configurations for hydraulic turbine startup // IOP Conference Series: Earth and Environmental Science, 22, 032021, 2014

10. М. Шляйхер. Техника автоматического регулирования для практиков. Издательство JUMO GmbH
11. Карпов В.Э. ПИД-управление в нестрогом изложении. Москва, 2012
12. Интернет ресурс «Энциклопедия АСУ ТП» (bookasutp.ru)
13. Romeo Susan-Resiga, Thi C. Vu, Sebastian Muntean, Gabriel Dan Ciocan, Bernd Nennemann. Jet control of the draft tube vortex rope in Francis turbines at partial discharge // 23<sup>rd</sup> IAHR Symposium – Yokohama, 1(14), 2006
14. An Yu, Zhipeng Zou, Daqing Zhou, Yuan Jpyng, Xianwu Luo. Investigation of the correlation mechanism between cavitation rope behavior and pressure fluctuations in a hydraulic turbine // Renewable Energy, 147, pp. 1199-1208, 2020
15. A. Farvel, A. Muller, C. Landry, K. Yamamoto, F. Avellan. LDV survey of cavitation and resonance effect on the precessing vortex rope dynamics in the draft tube of Francis turbines // Exp Fluids, 57:168, 2016
16. I. Litvinov, D. Suslov, E. Gorelikov, S. Shtork. Experimental Study of Transient Flow Regimes in a Model Hydroturbine Draft Tube // Energies, 14(5), 2021
17. Д. А. Суслов, И. В. Литвинов, С. И. Шторк, Е. Ю. Гореликов. Влияние переходных режимов на нестационарные вихревые явления в модели отсасывающей трубы гидротурбины // Сибирский физический журнал, Том 14, №4, 2019
18. RS-422 and RS-485 Applications eBook. International Headquarters, B+B SmartWorx, 2010
19. Delta Fan / Pump Vector Control Drive CP2000 Series User Manual. Delta Electronics, 2010