Реферат

Данная пояснительная записка к научно-исследовательской работе студента на тему: «Расчет прогнозного значения сигнала датчика с учетом показаний всех остальных установленных на объекте датчиков.» содержит 32 страниц, 8 рисунков, 7 источников.

Ключевые слова: ЦИФРОВЫЕ ФИЛЬТРЫ, ЧАСТОТНАЯ ВЫБОРКА, ЧИСЛЕННЫЕ МЕТОДЫ, КИХ-ФИЛЬТРЫ, ДВУМЕРНЫЕ ФИЛЬТРЫ.

Объектом исследования, проводимого в данной работе, является модель реактора типа РБМК.

Цель работы – реализация методарасчета прогнозного значения сигнала датчика с учетом показаний всех остальных установленных на объекте датчиков.

Первый раздел посвящён анализу особенностей внутриреакторной информации и требований к её фильтрации. Рассмотрен метод расчета прогнозного значения сигнала датчика.

Во втором разделе рассматривается разработка алгоритма расчета прогнозного значения сигнала датчика с учетом показаний всех остальных установленных на объекте датчиков..

В третьем разделе описывается проектирование структуры программного комплекса для реализации алгоритма расчета прогнозного значения сигнала датчика с учетом показаний всех остальных установленных на объекте датчиков..

Четвёртый раздел посвящён непосредственной реализации метода расчета прогнозного значения сигнала датчика с учетом показаний всех остальных установленных на объекте датчиков.

Заключение включает общие выводы по проведённой работе, достигнутые результаты и возможные направления для дальнейших исследований и улучшений.

Содержание

Введение

В системах сбора и обработки измерительной информации, как правило, вначале осуществляется проверка сигналов датчиков на достоверность. При этом если сигнал i- го датчика в k-ый момент времени классифицируется как недостоверный, он обычно исключается из дальнейших алгоритмов обработки. В результате возникают пробелы в последовательности исходных данных, которые увеличивают, иногда значительно, погрешность выполняемых затем расчетов.

В связи с этим представляется целесообразным разработка эффективного алгоритма восстановления недостающих данных. Его использование, в частности, позволит не только повысить точность обработки информации в штатных системах контроля, но и даст возможность оперативно поверять датчики контроля физических полей.

Целью данной работы является изучение алгоритма расчета прогнозного значения сигнала датчика с учетом показаний всех остальных установленных на объекте датчиков. В рамках работы будет проведено исследование теоретических основ метода, разработан алгоритм расчета прогнозного значения сигнала датчика с учетом показаний всех остальных установленных на объекте датчиков, а также выполнена программная реализация с последующим анализом результатов применения метода к данным.

Задачи, поставленные для достижения цели, включают:

1. Анализ особенностей внутриреакторной информации и требований к её фильтрации.
2. Изучение алгоритма расчета прогнозного значения сигнала датчика с учетом показаний всех остальных установленных на объекте датчиков.
3. Разработка алгоритма расчета прогнозного значения сигнала датчика с учетом показаний всех остальных установленных на объекте датчиков.
4. Разработка структуры программного комплекса для реализации алгоритма расчета прогнозного значения сигнала датчика с учетом показаний всех остальных установленных на объекте датчиков.
5. Проведение анализа результатов расчета прогнозного значения сигнала.

1. Аналитическая часть

Цифровой фильтр, который описывается передаточной функцией в виде полинома:

$$H(z) = h_0 + h_1 z^{-1} + \ldots + h_{N-1} z^{-(N-1)} = \sum_{k=0}^{N-1} h(k)\, z^{-k}$$

называется цифровым фильтром с конечной импульсной характеристикой (КИХ-фильтром). Импульсная характеристика h(k) имеет N отсчетов. Такие фильтры широко используются на практике. Важным примером КИХ-фильтра является трансверсальный фильтр.

Основные достоинства КИХ-фильтров:

– легко создавать КИХ-фильтры с линейной фазовой характеристикой;

– КИХ-фильтры можно строить как по рекурсивной, так и по нерекурсивной схемам;

– при нерекурсивной реализации КИХ-фильтр всегда устойчив;

– при рекурсивной реализации шумы округления из-за конечного числа разрядов можно минимизировать.

Недостатки КИХ-фильтров:

– для получения частотных характеристик A(ω) с крутыми срезами импульсная характеристика должна иметь большое число отсчетов

$$H(\omega) = \sum_k h(k) \exp(j\omega k \Delta t);$$

это выражение имеет вид усеченного ряда Фурье; если A(ω) = |H(ω)| имеет крутые срезы, то слева и справа от этого среза возникают колебания, обусловленные явлением Гиббса;

– задержка в КИХ-фильтрах с линейной фазовой характеристикой не всегда равна целому числу тактов дискретизации Δt, что иногда нежелательно.

Вклад КИХ-фильтров в обработку сигналов

1. Стабильность:

– КИХ-фильтры всегда стабильны, так как их характеристика полностью зависит от конечного числа входных отсчётов.

2. Простота реализации:

– Благодаря конечному числу коэффициентов, такие фильтры легко реализуются в цифровых устройствах.

3. Применение в реальных задачах:

– Удаление шумов в аудиозаписях и изображениях.

– Усиление полезного сигнала в телекоммуникациях.

– Выделение частотных диапазонов в медицинских данных.

Двумерные КИХ-фильтры — это тип фильтров, которые используются для обработки изображений и сигналов, особенно в контексте анализа данных и сигналов в различных областях, включая ядерную промышленность.

В ядерной промышленности такие фильтры могут быть полезны для нескольких целей:

– Обработка изображений: Используются для фильтрации данных с различных датчиков или камер, например, для улучшения качества изображений из гамма или нейтронных камер, которые могут быть использованы для мониторинга реакторов или выявления утечек радиации.

– Обработка спектров: В ядерных установках часто используются спектрометры для анализа радиации. Двумерные ких фильтры могут применяться для сглаживания спектров и удаления шумов, что позволяет более точно измерять уровни излучения и проводить анализ.

– Моделирование физических процессов: Ких фильтры могут быть полезны для численного моделирования процессов, таких как нейтронный поток или температурное распределение в реакторе. Это помогает в обеспечении безопасности и оптимизации работы ядерных установок.

– Диагностика и контроль состояния оборудования: Применяются для анализа сигналов с датчиков и обеспечения своевременного обнаружения аномалий в работе оборудования, например, в системах контроля ядерных реакторов.

В целом, двумерные КИХ-фильтры могут служить для улучшения точности данных и повышения безопасности в ядерной энергетике, позволяя более точно отслеживать и контролировать параметры работы реакторов и других критически важных систем.

1.1. Анализ особенностей внутриреакторной информации и требований к её фильтрации.

Современные системы контроля распределения энерговыделения (РЭ) в активной зоне ядерного реактора основаны на обработке внутриреакторной информации, получаемой от различных типов датчиков. Особенности этой информации обусловлены

как техническими характеристиками детекторов, так и физическими условиями эксплуатации в активной зоне.

К числу факторов, влияющих на точность внутриреакторных измерений, относятся высокая радиационная и температурная нагрузка, необходимость долгосрочной работы без обслуживания, ограниченное количество измерительных каналов, а также наличие шумов и случайных отклонений, обусловленных изменением состояния топлива и оборудования. Внутриреакторные детекторы, как правило, не покрывают всю активную зону, что требует интерполяции и восстановления РЭ на основе ограниченного объёма измерений. Расположение датчиков в реакторе представленно на рисунке 1.

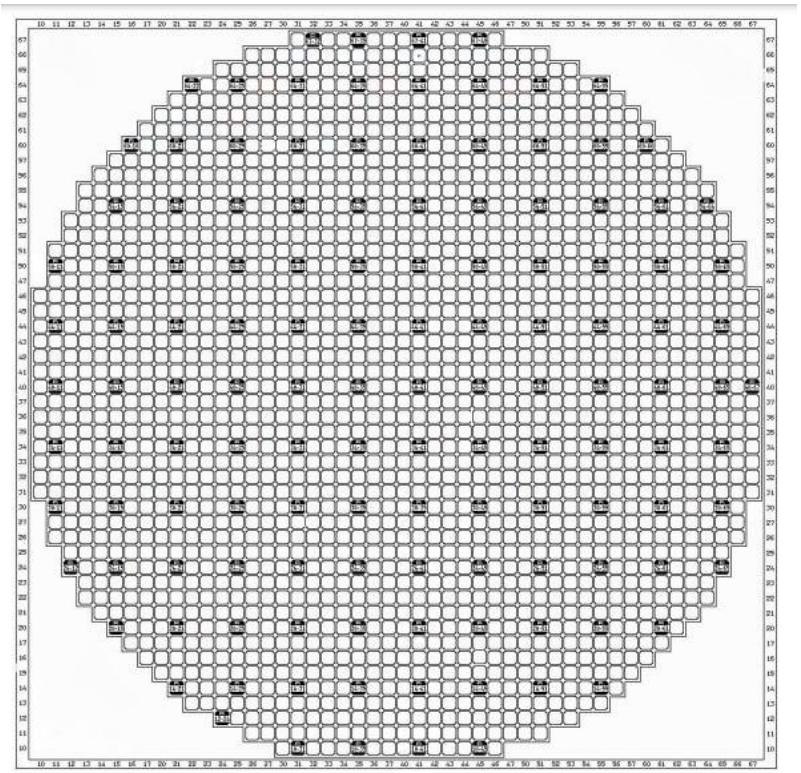

Рисунок 1 - Расположение датчиков в реакторе.

Таким образом, внутриреакторная информация представляет собой дискретные, зашумлённые данные, полученные в условиях сложной физической среды.

Для повышения точности и достоверности восстановления РЭ необходимо учитывать как случайную, так и детерминированную составляющие измеренного сигнала. Истинное распределение мощности можно представить следующим образом:

$$N(\vec{r}) = \tilde{N}(\vec{r}) + N^{(o)}(\vec{r})$$

Где $\tilde{N}(\vec{r})$ — математическое ожидание распределения (ДРЭ), получаемое из физического расчёта или статистического усреднения, а $N^{(o)}(\vec{r})$ — случайное отклонение, обусловленное измерительными шумами или моделируемыми ошибками [4].

Применение данного подхода позволяет точно оценивать влияние случайных факторов и, соответственно, формулировать требования к алгоритмам фильтрации. Например, при проектировании алгоритмов обработки внутриреакторных сигналов необходимо обеспечить:

- максимальную точность и пространственную детальность восстановления РЭ;
- способность алгоритма работать с сигналами от любых типов детекторов: локальных и нелокальных, внутри- и внереакторных, размещённых как по регулярной сетке, так и произвольно;
- независимость метода от геометрии активной зоны и конфигурации загрузки топлива;
- устойчивость к ошибкам измерений и физического моделирования;
- способность к адаптации и учёту деградации характеристик детекторов со временем[4].

Таким образом, требования к фильтрации внутриреакторной информации сводятся к необходимости использования гибких, высокоточных и вычислительно эффективных методов, которые могут обрабатывать дискретные и неполные данные с учётом шумов и ошибок. Этим требованиям в значительной степени соответствует использование аппроксимационных методов и оптимизационных подходов, например, линейного программирования или методов наименьших квадратов.

Дополнительно в задачах контроля РЭ важно уметь оценивать погрешность в любой точке активной зоны. Такая оценка может зависеть от расположения и состояния детекторов, степени их деградации, а также от распределения выгорания топлива и потока теплоносителя. В связи с этим требования к системам фильтрации расширяются до способности выполнять:

- автоматический учёт изменения конфигурации топливных сборок;
- учёт отказов отдельных датчиков;
- расчёт локальной погрешности в реальном времени;
- диагностику ложных срабатываний измерительных каналов.

Важно подчеркнуть, что внутриреакторная информация нередко сопровождается значительным количеством случайных возмущений, связанных с переходными режимами работы, температурными градиентами и радиационным фоном. Поэтому методы фильтрации не должны лишь устранять шумы, но и сохранять важные особенности сигнала, относящиеся к физическим аномалиям или аварийным ситуациям. Это требует

внедрения таких алгоритмов, которые обладают не только низкой рябью в полосе пропускания, но и высокой избирательностью в полосе подавления.

На практике для этих целей активно применяются методы частотной фильтрации, включая КИХ-фильтры (конечно-импульсные характеристики), синтез которых может быть адаптирован под конкретные частотные свойства сигналов. В сочетании с регулярной верификацией на основе физических расчётов и периодических поверок система контроля способна обеспечивать высокую точность восстановления РЭ на всех этапах работы реактора.

1.2. Изучение алгоритма расчета прогнозного значения сигнала датчика с учетом показаний всех остальных установленных на объекте датчиков.

Предлагаемый алгоритм вначале рассмотрим для одномерного случая на примере восстановления одного отсутствующего отсчета $v(l)$ последовательности $\{v(k)\}$. Затем приведем его обобщение для многомерных случаев. При этом будем предполагать, что при переходе от непрерывной функции $v(x)$ к ее дискретному аналогу $\{v(k)\}$ интервал дискретизации Δx выбран в соответствии с теоремой Котельникова [2] и наложение частот отсутствует.

При разработке алгоритма учтем, что идеальное качество восстановления будет гарантировать алгоритм интерполяции или аппроксимации с линейной фазой и амплитудно-частотной характеристикой $\left|H(e^{j\omega})\right|=1$ во всем нормированном частотном диапазоне сигнала, то есть

$$\left|H(e^{j\omega})\right|=1 \quad при \quad 0\le\omega\le\frac{\pi}{\Delta x}. \qquad (1)$$

Однако создание такого алгоритма на основе известных методов численного анализа не представляется возможным. Это связано как с нелинейностью фазовых характеристик классических алгоритмов, так и с невозможностью выбора конечного набора базисных функций, имеющего свойства всепропускающего фильтра вида (1) [3,4,5].

В связи с этим задачу восстановления отсутствующего отсчета $v(l)$ в настоящей работе предлагается решать методами цифровой обработки сигналов, так как только цифровые фильтры с конечной импульсной характеристикой (КИХ) имеют линейную фазу, позволяют наиболее полно учесть эффекты дискретности и спектральный состав

обрабатываемой последовательности $\{v(k)\}$, а современные методы их расчета [2,4,6] дают возможность воспроизвести желаемую частотную характеристику вида (1) с погрешностью 0.1% и менее.

При применении цифрового КИХ-фильтра будем использовать следующее соотношение между его входом и выходом [2]:

$$y(l) = \sum_{k=l-M}^{l+M} v(k) \cdot h(L+k-l) + h(L) \cdot v(l), \qquad (2)$$

где N –число отсчетов импульсной характеристики $\{h(k)\}$ КИХ-фильтра, M=(N-1)/2, L=M+1.

Выражение (2) позволяет сравнительно просто восстановить недостающий отсчет $v(l)$. В частности, если спектр последовательности $\{v(k)\}$ (по Фурье) не превосходит полосы пропускания цифрового фильтра (2), то

$$v(l) = y(l) = \frac{\sum_{\substack{k=l-M \\ k \neq l}}^{l+M} v(k) \cdot h(L+k-l)}{1-h(L)} \qquad (3)$$

Если спектр последовательности $\{v(k)\}$ превосходит полосу пропускания цифрового фильтра (2), то на основе соотношений (2) и (3) в настоящей работе предлагается следующий алгоритм вычисления недостающего отсчета $v(l)$, имеющий практически идеальную характеристику вида (1). Алгоритм включает три основных этапа.

Этап 1. Выполняется частотное разделение исходной последовательности $\{v(k)\}$ на низкочастотные $\{v_н(k)\}$ и высокочастотные $\{v_в(k)\}$ составляющие. Предположим, что применяемый для этого фильтр нижних частот (ФНЧ) имеет полосу пропускания $0 \le \omega \le \omega_п$, и полосу непропускания $\omega_н \le \omega \le \pi / \Delta x$ (Рис.1). Значения граничных частот $\omega_п$ и $\omega_н$ при этом можно выбирать произвольно в широких пределах, однако для упрощения последующих расчетов целесообразно положить $\omega_п \sim (0.4-0.5) \cdot \pi / \Delta x$, $\omega_н \sim (0.7-0.8) \cdot \pi / \Delta x$.

Этап 2. С использованием ФНЧ, имеющего полосу пропускания шире, чем спектр последовательности $\{v_н(k)\}$ (Рис.1), по формуле (3) вычисляется низкочастотная составляющая $v_н(l)$ недостающего отсчета $v(l)$.

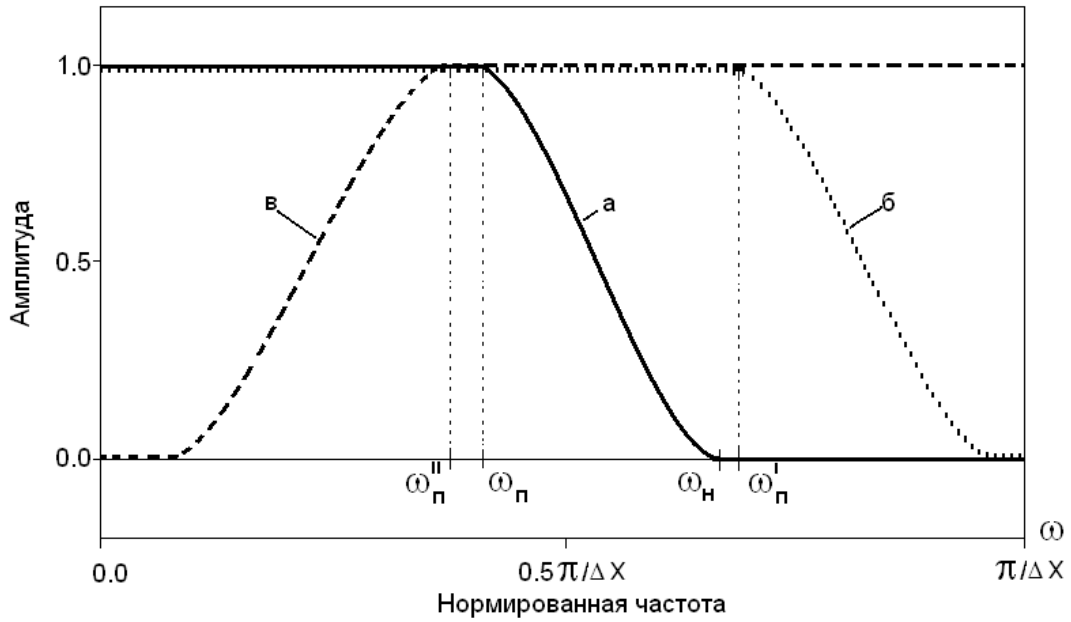

Рис.1. Амплитудно-частотные характеристики цифровых фильтров: а) ФНЧ этапа 1; б) ФНЧ этапа 2 ($\omega_п' > \omega_н$); в) ФВЧ этапа 3 ($\omega_п'' < \omega_п$).

Этап 3. С использованием фильтра верхних частот (ФВЧ), имеющего полосу пропускания шире, чем спектр последовательности $\{v_в(k)\}$ (Рис.1), по формуле (3) вычисляется высокочастотная составляющая $v_в(l)$ недостающего отсчета $v(l)$.

Этап 4. Складывая значения $v_н(l)$ и $v_в(l)$, окончательно получим значение недостающего отсчета $v(l)$:

$$v(l) = v_н(l) + v_в(l). \qquad (4)$$

Для обеспечения высокой точности восстановления отсутствующего отсчета $v(l)$ ФНЧ этапа 1 должен иметь возможно меньшие погрешности аппроксимации в полосах пропускания и непропускания, а ФНЧ и ФВЧ этапов 2 и 3 – удовлетворять этому условию только в полосах пропускания. Ширина переходных полос при этом может быть весьма большой, что дает возможность намного уменьшить число отсчетов импульсных характеристик используемых в алгоритме цифровых фильтров и минимизировать объем вычислений при выполнении фильтрации. Проведенные расчеты показывают, что при восстановлении $v(l)$ с погрешностью порядка 0.1% длина оптимальных КИХ-фильтров [2] не превышает 17 отсчетов (т.е. N=17, M=8).

Предложенный алгоритм без принципиальных изменений можно использовать и в многомерных случаях – для этого вместо одномерных КИХ-фильтров на этапах 1-3 необходимо использовать соответствующие двух- или трехмерные фильтры. Методы их расчета к настоящему времени хорошо разработаны [2,6] и не вызывают заметных трудностей при практической реализации.

2. Теоретическая часть

2.1. Разработка алгоритма расчета прогнозного значения сигнала датчика с учетом показаний всех остальных установленных на объекте датчиков.

Пусть фильтр имеет конечную импульсную характеристику длиной 2M+1, а его центральный коэффициент равен h(M).

Если спектр последовательности {v(k)} не превосходит полосу пропускания цифрового фильтра, то на основе уравнения свёртки пропущенный отсчёт v(l) может быть восстановлен по формуле:

$$v(l) = y(l) = \frac{\sum\limits_{\substack{k=l-M \\ k\neq l}}^{l+M} v(k)\cdot h(L+k-l)}{1-h(L)}$$

Рассмотрим теперь двумерный дискретный сигнал v(i,j), определённый на прямоугольной сетке. Пусть используется двумерный цифровой фильтр с конечной импульсной характеристикой

$$h(p,q), \quad p,q \in [-M, M]$$

причём центральный коэффициент фильтра равен h(0,0).

Спектр двумерного сигнала v(i,j) целиком лежит в полосе пропускания данного фильтра. Уравнение двумерной свёртки имеет вид:

$$y(i,j) = \sum_{p=-M}^{M} \sum_{q=-M}^{M} h(p,q)\, v(i-p, j-q).$$

Если в точке (i0,j0) значение сигнала отсутствует, а во всех остальных точках внутри окна фильтра значения известны, то, полагая y(i0,j0) = v(i0,j0), можно выделить неизвестный отсчёт:

$$v(i_0,j_0)\,(1-h(0,0)) = \sum_{p=-M}^{M} \sum_{q=-M}^{M} {}'h(p,q)\, v(i_0-p, j_0-q)$$

где штрих указывает на исключение члена при p=0, q=0.

Таким образом, пропущенный отсчёт двумерного сигнала может быть восстановлен по формуле:

$$v(i_0,j_0) = \frac{\sum\limits_{k=i_0-M}^{i_0+M} \sum\limits_{l=j_0-M}^{j_0+M} {}'v(k,l)\; h\big(L+k-i_0,\; L+l-j_0\big)}{1-h(L,L)}$$

3. Инженерная часть

3.1. Разработка структуры программного комплекса для реализации алгоритма расчета прогнозного значения сигнала датчика с учетом показаний всех остальных установленных на объекте датчиков.

Общая структура программного комплекса

Программный комплекс состоит из четырёх основных модулей, каждый из которых реализует определённую функциональность для удобного и эффективного выполнения задач синтеза, фильтрации и анализа данных.

Ниже приведены подробные описания модулей:

1. Модуль синтеза фильтра

Этот модуль отвечает за создание двумерных КИХ-фильтров методом частотной выборки.

- Функция генерации частотной сетки:
  - Формирует двумерную сетку частотных координат (u,v) на равномерной решётке размером $G \times G$.
  - Вычисляет расстояния до центра $R = u\text{^}2 + v\text{^}2$, чтобы определить маски полосы пропускания и задерживания.
  - Используется для задания частотных ограничений фильтра.
- Формирование матрицы частотных откликов:
  - Строится матрица A , описывающая, как каждый коэффициент фильтра влияет на частотный отклик в заданной точке.
  - Для каждой частоты (u k ,v k) используется экспоненциальное ядро $e^{-j\,2\,\pi\,(\,u\,k\,x + v\,k\,y\,)}$.
  - Отдельно формируются матрицы для полосы пропускания (A_pass) и полосы задерживания (A_stop).
- Постановка задачи линейного программирования:

  Для минимизации максимального отклонения амплитудно-частотной характеристики (АЧХ) от желаемого идеала в полосе пропускания и полосе задерживания была поставлена следующая задача линейного программирования:

  Переменные

  - M = N² — количество коэффициентов фильтра.

  - h ∈ ℝ^M — вектор коэффициентов фильтра.

  - δ ∈ ℝ — переменная, представляющая максимальное отклонение (рябь).

Вектор переменных задачи:

$x = [h_1, h_2, ..., h\_M, \delta]^T \in \mathbb{R}^{(M+1)}$

Целевая функция

Минимизировать рябь δ:

$\min \delta \Leftrightarrow \min c^T * x$, где $c = [0, ..., 0, 1]^T$

Ограничения

1. Нормировка по постоянной составляющей:

$\sum h\_ij = 1 \Leftrightarrow A\_eq * x = b\_eq$, где $A\_eq = [1, 1, ..., 1, 0]$

2. Ограничения в полосе пропускания (passband):

Для каждой частоты (u_k, v_k) с $\sqrt{(u\_k^2 + v\_k^2)} \leq f\_p$:

$1 - \delta \leq H(u\_k, v\_k) \leq 1 + \delta$

Это эквивалентно двум линейным ограничениям:

$A\_k * h - \delta \leq 1$

$-A\_k * h - \delta \leq -1$

3. Ограничения в полосе задерживания (stopband):

Для каждой частоты (u_k, v_k) с $\sqrt{(u\_k^2 + v\_k^2)} \geq f\_s$:

$|H(u\_k, v\_k)| \leq \delta \Leftrightarrow \begin{cases} A\_k * h - \delta \leq 0 \\ -A\_k * h - \delta \leq 0 \end{cases}$

4. Границы переменных:

$-1 \leq h\_i \leq 1$ для всех $i = 1, ..., M$

$\delta \geq 0$

Итоговая форма задачи ЛП

Искомое:

$\text{minimize } (c^T * x)$

при условиях:

$A\_ub * x \leq b\_ub$

$A\_eq * x = b\_eq$

и границы на x.

– Решение задачи оптимизации:
  – Используется scipy.optimize.linprog с методом 'highs'.
  – На выходе получается оптимальный вектор коэффициентов h.
  – Функция симметризации фильтра:
  – Обеспечивает пространственную симметрию ядра (нулевая фаза).
  – Усредняет значения по центрально-симметричным точкам ядра h (x , y).

2. Модуль работы с данными

Данный модуль отвечает за формирование входных данных, моделирование пропусков и выполнение операций фильтрации.

– Генерация тестовых сигналов

Для проверки работоспособности алгоритма используются синтетические двумерные сигналы, формируемые в виде суммы низкочастотной и высокочастотной составляющих. Это позволяет контролировать спектральный состав сигнала и анализировать качество разделения и восстановления.

– Формирование пропусков

Пропуски моделируются путём задания бинарной маски, определяющей отсутствующие значения сигнала. В соответствующих точках значения сигнала заменяются нулевыми, при этом маска используется на последующих этапах восстановления.

– Частотное разделение сигнала

Разделение сигнала на низкочастотную и высокочастотную компоненты осуществляется с использованием двумерной свёртки с синтезированным разделяющим КИХ-фильтром. Для уменьшения влияния пропусков применяется нормализованная свёртка, при которой вклад фильтра учитывается только по известным отсчётам сигнала.

3. Модуль восстановления пропущенных значений.

Восстановление пропущенных значений выполняется локально, отдельно для каждой спектральной компоненты сигнала.

Локальная формула восстановления:

Для каждой точки пропуска (i0,j0) восстановление осуществляется на основе значений соседних отсчётов в окне фильтра согласно выражению:

$$v(i_0, j_0) = \frac{\sum_{k=i_0-M}^{i_0+M} \sum_{l=j_0-M}^{j_0+M}{}' v(k,l)\, h\big(L+k-i_0,\ L+l-j_0\big)}{1 - h(L,L)}$$

где h(p,q) - импульсная характеристика восстанавливающего фильтра, а штрих означает исключение центрального коэффициента.

Восстановление выполняется только в точках пропусков, тогда как остальные значения сигнала сохраняются неизменными.

Алгоритм восстановления применяется независимо к низкочастотной и высокочастотной компонентам сигнала с использованием соответствующих восстанавливающих фильтров. После восстановления компоненты суммируются, формируя итоговую оценку сигнала.

4. Конфигурационный модуль

Данный модуль предназначен для оценки качества восстановления и наглядного представления результатов.

Метрики качества

В программной реализации рассчитываются следующие показатели:

- среднеквадратическая ошибка (MSE);
- отношение сигнал/шум (PSNR);
- средняя абсолютная ошибка в точках пропусков.

Визуализация результатов

Предусмотрена визуализация:

- исходного сигнала;
- сигнала с пропусками;
- восстановленного сигнала;
- ошибок восстановления;
- импульсных и частотных характеристик используемых фильтров.

Поток выполнения программы:

Этап 1. Ввод и инициализация данных

На первом этапе выполняется:

- загрузка или генерация двумерного входного сигнала;
- формирование бинарной маски пропусков;
- задание параметров фильтрации (размер фильтра, частоты среза, ширина переходной зоны).

Этап 2. Синтез двумерных КИХ-фильтров

На втором этапе осуществляется синтез фильтров, используемых в дальнейшем алгоритме:

- разделяющего низкочастотного фильтра;
- восстанавливающего низкочастотного фильтра;
- фильтра, используемого для формирования высокочастотной восстанавливающей компоненты.

Синтез фильтров выполняется методом линейного программирования с заданными ограничениями на амплитудно-частотную характеристику.

Этап 3. Частотное разделение сигнала

На третьем этапе входной сигнал с пропусками разделяется на низкочастотную и высокочастотную составляющие.

Для уменьшения влияния пропусков применяется нормализованная двумерная свёртка, при которой вклад фильтра учитывается только по известным значениям сигнала. В результате формируются оценки низкочастотной и высокочастотной компонент, содержащие пропуски в тех же точках, что и исходный сигнал.

Этап 4. Локальное восстановление пропущенных значений

На четвёртом этапе выполняется восстановление пропущенных значений отдельно для каждой спектральной компоненты.

Для каждой точки пропуска осуществляется:

- формирование локального окна вокруг восстанавливаемой точки;
- вычисление восстановленного значения по двумерной формуле восстановления;
- замена пропущенного значения на вычисленную оценку.

Восстановление выполняется только в точках, отмеченных в маске пропусков.

Этап 5. Формирование итогового сигнала

На пятом этапе восстановленные низкочастотная и высокочастотная компоненты суммируются, формируя итоговую оценку восстановленного сигнала.

Этап 6. Анализ результатов и визуализация

На заключительном этапе выполняется:

- вычисление метрик качества восстановления;
- построение изображений исходного, повреждённого и восстановленного сигналов;
- визуализация частотных характеристик фильтров и спектров сигналов.

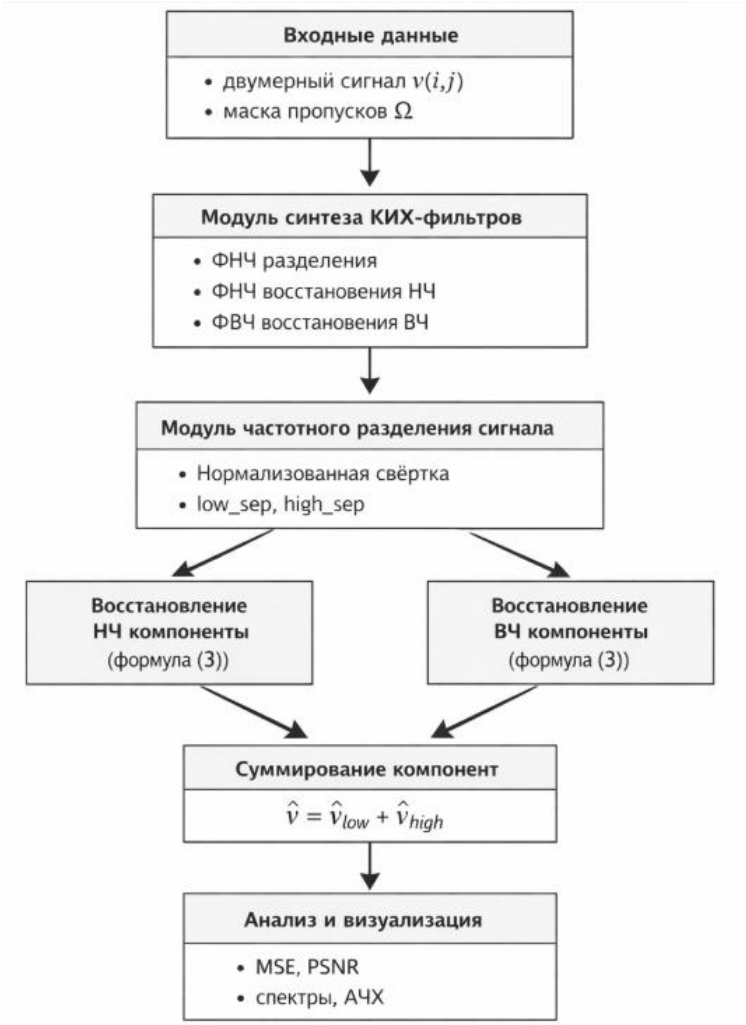

Рисунок 3 – Архитектура программного комплекса.

На рисунке 3 представлена архитектура программного комплекса для реализации расчета прогнозного значения сигнала датчика с учетом показаний всех остальных установленных на объекте датчиков.

Структурная схема алгоритма восстановления включает следующие функциональные блоки:

1. Блок входных данных

   Принимает двумерный сигнал и маску пропусков.

2. Блок синтеза фильтров

   Формирует набор двумерных КИХ-фильтров на основе заданных параметров.

3. Блок частотного разделения

   Выполняет разделение сигнала на низкочастотную и высокочастотную компоненты с учётом пропусков.

4. Блок восстановления НЧ-компоненты

   Выполняет локальное восстановление пропущенных значений в низкочастотной составляющей.
5. Блок восстановления ВЧ-компоненты

   Выполняет локальное восстановление пропущенных значений в высокочастотной составляющей.
6. Блок суммирования компонент

   Формирует итоговый восстановленный сигнал.
7. Блок оценки качества

   Производит анализ точности восстановления и визуализацию результатов.

Эта структура обеспечивает модульность, лёгкость масштабирования и удобство в использовании.

4. Технологическая и практическая часть

4.1. Программная реализация алгоритма расчета прогнозного значения сигнала датчика с учетом показаний всех остальных установленных на объекте датчиков.

```
def design_lp_filter_2d(N, cutoff, transition, G=81):
    assert N % 2 == 1, "N должно быть нечётным"

    u = (np.arange(G) - G // 2) / G
    U, V = np.meshgrid(u, u, indexing="ij")
    R = np.sqrt(U**2 + V**2)

    pass_mask = R <= cutoff
    stop_mask = R >= cutoff + transition

    coords = np.array([(i - N // 2, j - N // 2)
                       for i in range(N) for j in range(N)])

    def build_A(mask):
        rows = []
        idx = np.where(mask)
        for k, l in zip(idx[0], idx[1]):
            du, dv = U[k, l], V[k, l]
            rows.append(np.real(np.exp(
                -2j * np.pi * (du * coords[:, 0] + dv * coords[:, 1])
            )))
        return np.asarray(rows)

    A_pass = build_A(pass_mask)
    A_stop = build_A(stop_mask)

    M = N * N
    c = np.zeros(M + 1); c[-1] = 1.0

    A_eq = np.zeros((1, M + 1))
    A_eq[0, :M] = 1.0
    b_eq = np.array([1.0])

    A_ub, b_ub = [], []

    A_ub += [np.hstack([ A_pass, -np.ones((A_pass.shape[0], 1)) ])]
    b_ub += [ np.ones(A_pass.shape[0]) ]
    A_ub += [np.hstack([-A_pass, -np.ones((A_pass.shape[0], 1)) ])]
    b_ub += [-np.ones(A_pass.shape[0])]
    A_ub += [np.hstack([ A_stop, -np.ones((A_stop.shape[0], 1)) ])]
    b_ub += [ np.zeros(A_stop.shape[0]) ]
    A_ub += [np.hstack([-A_stop, -np.ones((A_stop.shape[0], 1)) ])]
    b_ub += [ np.zeros(A_stop.shape[0]) ]
```

```
    A_ub = np.vstack(A_ub)
    b_ub = np.hstack(b_ub)

    bounds = [(-1, 1)] * M + [(0, None)]

    res = linprog(c, A_ub, b_ub, A_eq, b_eq,
                  bounds=bounds, method="highs")

    h = res.x[:M].reshape(N, N)

    for i in range(N):
        for j in range(N):
            i2, j2 = N - 1 - i, N - 1 - j
            mean = (h[i, j] + h[i2, j] + h[i, j2] + h[i2, j2]) / 4.0
            h[i, j] = h[i2, j] = h[i, j2] = h[i2, j2] = mean

    return h, res.x[-1]
```

Создание двумерного КИХ-фильтра производится методом линейного программирования, адаптированным под частотную выборку:

Создание частотной сетки:

- Формируется двумерная сетка координат u, v в частотной области размером G×G.
- Вычисляется матрица расстояний R от каждой точки сетки до центра (нулевая частота).
- Определяются области пропускания ($R \leq$ cutoff) и подавления ($R \geq$ cutoff + transition).

Составление матрицы ограничений:

- Вычисляется АЧХ фильтра в точках полосы пропускания и задерживания с помощью косинусного преобразования.
- Формируются системы неравенств, отражающие требования:
  - В полосе пропускания: амплитуда должна находиться в пределах [1 - δ, 1 + δ].
  - В полосе подавления: амплитуда ≤ δ.

Постановка задачи линейного программирования:

- Целевая функция — минимизация ряби δ в полосах.
- Используется linprog() из библиотеки SciPy с методом "highs".
- Полученные коэффициенты свёрточного ядра фильтра преобразуются в квадратную матрицу N×N.

Симметризация фильтра:

- Для обеспечения нулевой фазы фильтр делается симметричным относительно центра: каждый элемент усредняется с соответствующими ему зеркальными элементами.

```
# Восстановление
def recover_component_local(data_band, h, mask_missing):
    H, W = data_band.shape
    N = h.shape[0]
    M = N // 2
    h0 = h[M, M]
    denom = 1.0 - h0

    restored = data_band.copy()
    miss_idx = np.argwhere(mask_missing)

    for i0, j0 in miss_idx:
        s = 0.0
        for di in range(-M, M+1):
            for dj in range(-M, M+1):
                if di == 0 and dj == 0:
                    continue
                ii = (i0 - di) % H
                jj = (j0 - dj) % W
                s += h[M+di, M+dj] * data_band[ii, jj]
        restored[i0, j0] = s / denom

    return restored
```

Функция recover_component_local реализует локальное восстановление пропущенных значений двумерной компоненты сигнала (низкочастотной или высокочастотной) на основе значений соседних отсчётов и коэффициентов восстанавливающего КИХ-фильтра.

Инициализация параметров:

В начале функции определяется размер входных данных и параметры фильтра:

H, W = data_band.shape

N = h.shape[0]

M = N // 2

Здесь H и W — размеры двумерного сигнала, N — размер фильтра, а M — радиус окна фильтра, используемого для восстановления.

Далее извлекается центральный коэффициент фильтра и вычисляется постоянный знаменатель, используемый при восстановлении:

```
h0 = h[M, M]
denom = 1.0 - h0
```

Подготовка данных к восстановлению:

Создаётся копия входного массива, в которую будут записываться восстановленные значения:

```
restored = data_band.copy()
```

Затем формируется список координат всех пропусков на основе маски:

```
miss_idx = np.argwhere(mask_missing)
```

Восстановление пропущенных значений:

Основной цикл функции проходит по всем координатам пропусков:

```
for i0, j0 in miss_idx:
```

Для каждой точки пропуска выполняется суммирование вкладов соседних значений внутри окна фильтра. Два вложенных цикла перебирают смещения по строкам и столбцам относительно центра окна:

```
for di in range(-M, M+1):
    for dj in range(-M, M+1):
```

Центральная точка окна исключается из расчёта:

```
if di == 0 and dj == 0:
    continue
```

Для корректной обработки граничных точек используется периодическое продолжение индексов:

```
ii = (i0 - di) % H
jj = (j0 - dj) % W
```

После этого значение соседнего отсчёта умножается на соответствующий коэффициент фильтра и добавляется к сумме:

```
s += h[M+di, M+dj] * data_band[ii, jj]
```

Запись восстановленного значения:

После обхода всех соседних точек восстановленное значение вычисляется и записывается в соответствующую позицию массива restored:

```
restored[i0, j0] = s / denom
```

4.2 Разработка программного блока применения фильтра к данным.

```
def low_high_separation_normalized(data_missing, mask_missing, h_sep):
```

```
    mask_valid = (~mask_missing).astype(float)

    num = convolve2d(data_missing, h_sep, mode='same', boundary='wrap')

    denom = convolve2d(mask_valid, h_sep, mode='same', boundary='wrap')

    low_sep = np.zeros_like(num)

    eps = 1e-6
    good = np.abs(denom) > eps
    low_sep[good] = num[good] / denom[good]

    low_sep_raw = convolve2d(data_missing, h_sep, mode='same', boundary='wrap')
    low_sep[~good] = low_sep_raw[~good]

    high_sep = data_missing - low_sep

    return low_sep, high_sep
```

Функция low_high_separation_normalized выполняет частотное разделение сигнала с пропусками на низкочастотную и высокочастотную компоненты с использованием нормализованной свёртки. В отличие от обычной свёртки, нормализованный вариант компенсирует влияние пропусков за счёт учёта только известных значений, что уменьшает искажения при фильтрации.

Подготовка маски известных значений:

Сначала формируется маска, где известные значения помечены единицами, а пропуски — нулями:

mask_valid = (~mask_missing).astype(float)

Это позволяет учитывать только те точки, где исходное значение действительно присутствует.

Вычисление числителя нормализованной свёртки:

Далее выполняется обычная двумерная свёртка входного сигнала с разделяющим фильтром:

num = convolve2d(data_missing, h_sep, mode='same', boundary='wrap')

Здесь используется режим:

- mode='same' — результат того же размера, что и входные данные;
- boundary='wrap' — периодическое продолжение границ.

Поскольку в data_missing в точках пропусков обычно стоят нули, вклад пропусков автоматически исключается из суммирования.

Вычисление знаменателя — эффективной суммы весов:

Чтобы нормировать результат и компенсировать потерянные веса фильтра в области пропусков, выполняется свёртка маски известных значений тем же фильтром:

denom = convolve2d(mask_valid, h_sep, mode='same', boundary='wrap')

Важно, что используется тот же h_sep, а не модуль abs(h_sep), так как изменение знаков фильтра приводит к нарушению его частотных свойств.

Нормирование результата и обработка проблемных случаев:

Создаётся массив для низкочастотной компоненты:

low_sep = np.zeros_like(num)

Затем задаётся порог для предотвращения деления на очень малые значения:

eps = 1e-6

good = np.abs(denom) > eps

Далее выполняется нормализация только в тех точках, где знаменатель достаточно велик:

low_sep[good] = num[good] / denom[good]

Резервный вариант для “пустых” областей:

Если в некоторой точке вокруг фильтра слишком много пропусков, denom может стать близким к нулю. В этом случае применяется резервный вариант — обычная свёртка без нормировки:

low_sep_raw = convolve2d(data_missing, h_sep, mode='same', boundary='wrap')

low_sep[~good] = low_sep_raw[~good]

Это позволяет избежать нестабильных значений и гарантирует, что алгоритм выдаст корректный результат даже в областях с высокой плотностью пропусков.

Формирование высокочастотной компоненты:

После получения низкочастотной составляющей высокочастотная вычисляется как разность между исходными данными и НЧ-оценкой:

high_sep = data_missing - low_sep

4.3 Анализ результатов применения метода

Входной сигнал:

```
o_x = o_y = 64
w_low  = 0.05
w_high = 0.28

X, Y = np.meshgrid(np.arange(o_x), np.arange(o_y), indexing="xy")

data_low  = 2 + np.sin(2*np.pi*w_low*X)  + np.sin(2*np.pi*w_low*Y)
```

```
data_high = np.sin(2*np.pi*w_high*X) + np.sin(2*np.pi*w_high*Y)
data_orig = data_low + data_high
```

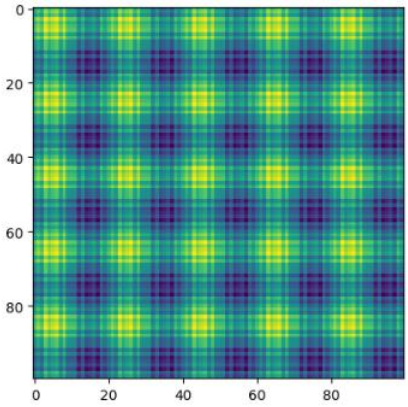

Рисунок 3 – Входной сигнал.

Применение разделяющего фильтра:

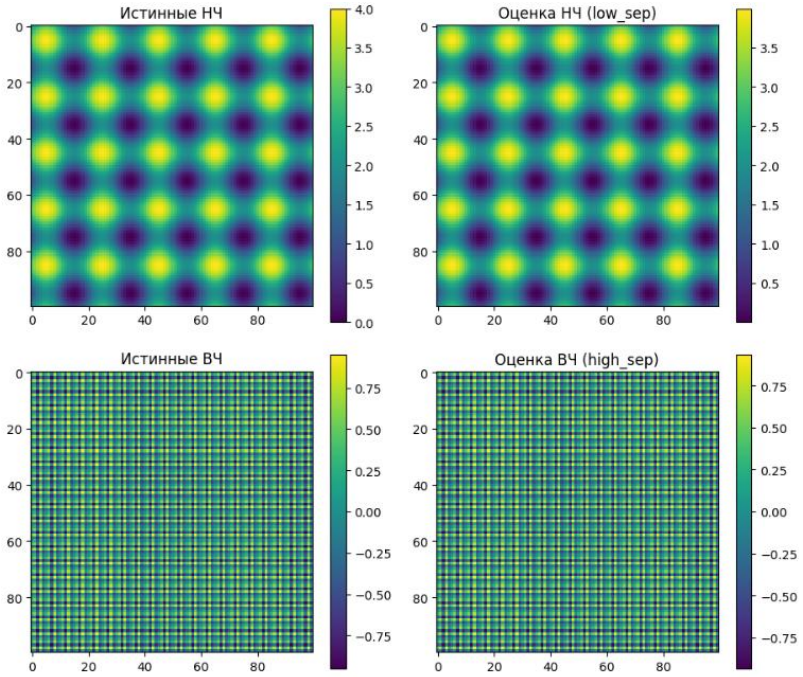

Рисунок 4 – Применение разделяющего фильтра.

Разделяющий фильтр идеально разделил сигнал на высокочастотную и низкочастотную состовляющую, что в будущем позволит обрабатывать компоненты поотдельности.

Восстановление низкочастотного сигнала:

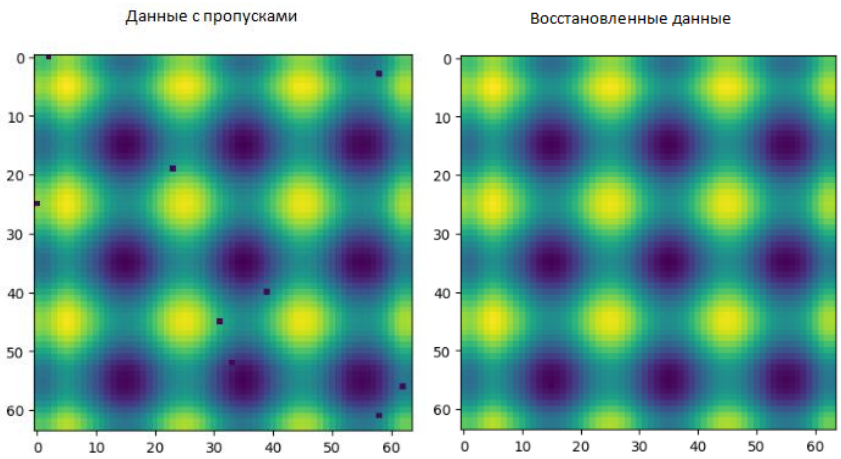

Рисунок 5 – Результат восстановления низкочастотного сигнала.

Восстановление высокочастотного сигнала:

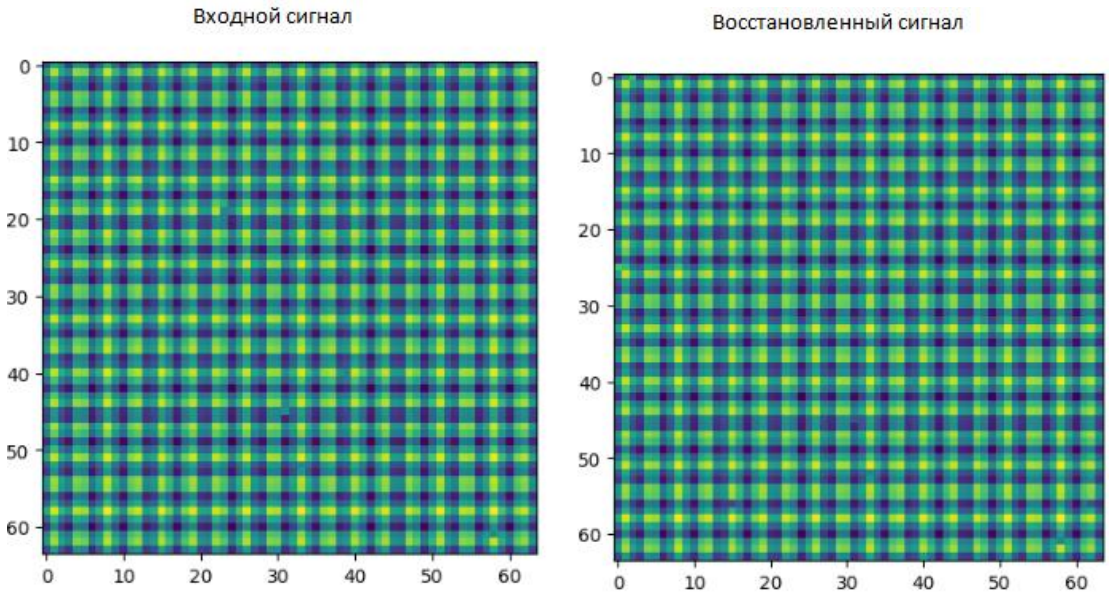

Рисунок 6 – Результат восстановления высокочастотного сигнала.

Восстановление комплексного сигнала:

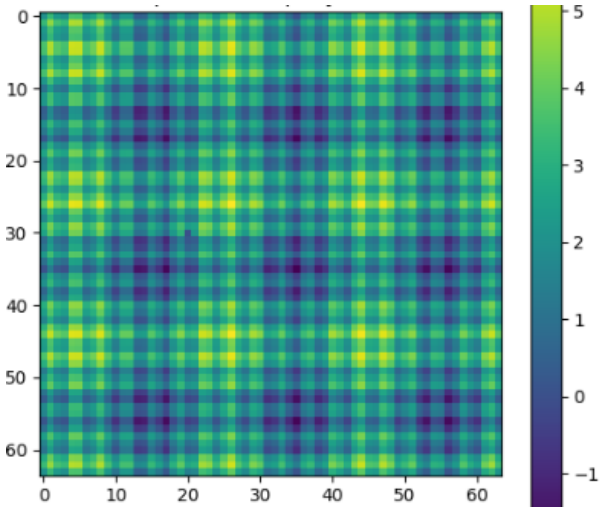

Рисунок 7 – Комплексный сигнал с пропусками.

Входной сигнал состоит из двух компонент, сначала создаются высокочастотный и низкочастотный сигнал, после чего они складываются.Так же задаются пропуски, которые необходимо восстановить. Таким образом мы получаем комплексный сигнал имеющий пропуски для проверки метода восстановления.

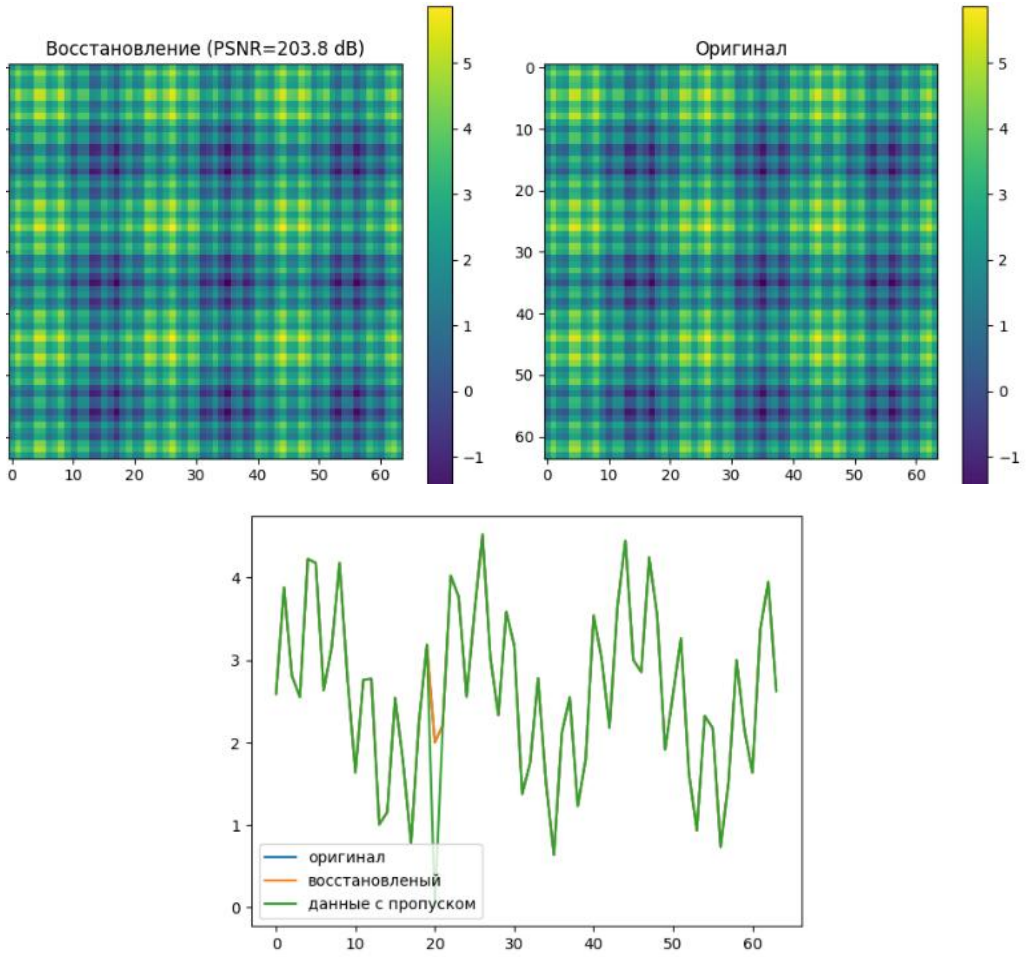

Рисунок 8 – Результат восстановления.

Визуально видно, что сигнал восстановлен идеально, что подтвержается показателями качества:

MSE по всей картинке: 0.00000000

PSNR по всей картинке: 203.84 dB

MAE в точках дырок: 0.000000

Средняя относит. ошибка в дырках: 0.00 %

Нулевая среднеквадратическая и средняя абсолютная ошибки указывают на отсутствие заметных отклонений как в области пропусков, так и вне их. Значение PSNR, превышающее 200 дБ, характерно для случаев, когда различия между исходным и восстановленным сигналами находятся на уровне численной погрешности вычислений.

Таким образом, в рассматриваемом эксперименте алгоритм восстановления корректно воспроизводит пропущенные значения сигнала без внесения искажений в остальную часть изображения.

Заключение

В рамках данной работы были Проанализированы особенности внутриреакторной информации.

Рассмотрена разработка алгоритма расчета прогнозного значения сигнала датчика с учетом показаний всех остальных установленных на объекте датчиков..

Спроектирование структуры программного комплекса для реализации алгоритма расчета прогнозного значения сигнала датчика с учетом показаний всех остальных установленных на объекте датчиков.

На основе спроектированной структуры программы реализован программный комплекс для расчета прогнозного значения сигнала датчика с учетом показаний всех остальных установленных на объекте датчиков.

Проведен анализ результатов расчета прогнозного значения сигнала.

Список литературы

1. Oppenheim A. V., Schafer R. W. Digital Signal Processing. – Prentice Hall, 1975. – 585 с.
2. Proakis J. G., Manolakis D. K. Digital Signal Processing: Principles, Algorithms, and Applications. – Pearson, 2006. – 1000 с.
3. Медведев С. И. Цифровая обработка сигналов. – Москва: Физматлит, 2012. – 432 с.
4. Постников В.В., Якунин И.С. Контроль распределения энерговыделения в активной зоне ядерного реактора: Учебное пособие. – М.: НИЯУ МИФИ, 2012. – 92 с.
5. Рабинер Л., Гоулд Б. Теория и применение цифровой обработки сигналов. – Москва: Мир, 1978. – 856 с.
6. Каппеллини В., Константинидис А. Дж., Эмилиани П. Цифровые фильтры и их применение. – Пер. с англ. – Москва: Энергоатомиздат, 1983. – 360 с.
7. Яковлев А., Соколова Д. Цифровая фильтрация и синтез цифровых фильтров. – Litres, 2022.