

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Локтионова Оксана Геннадьевна

Должность: проректор по учебной работе

Дата подписания: 10.10.2024 12:15:57

Уникальный программный ключ:

0b817ca911e6668abb13a5d476d39e5f1c11eabbf73e943df4a4851fda56d089

## МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Юго-Западный государственный университет»  
(ЮЗГУ)

Кафедра механики, мехатроники и робототехники

УТВЕРЖДАЮ

Проректор по учебной работе

 О. Локтионова

«26» 09



## ЦИФРОВЫЕ ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ В РОБОТОТЕХНИКЕ

Методические указания по выполнению самостоятельной работы  
для студентов направления «Мехатроника и робототехника»

Курск 2024

УДК 51-74

Составитель: П.А. Безмен

Рецензент

Кандидат технических наук, доцент кафедры механики, мехатроники и  
робототехники

Е.Н. Политов

**Цифровые интеллектуальные системы в робототехнике:**  
методические указания по выполнению самостоятельной работы для  
студентов направления «Мехатроника и робототехника» / Юго-Зап.  
гос. ун-т; сост.: П.А. Безмен; Курск, 2024. 106 с.

Содержат сведения по вопросам основных методов теории  
цифровой обработки сигналов, используемых при предварительной  
обработке сигналов различной физической природы. Приведены  
примеры и задания по цифровой обработке сигналов.

Методические указания соответствуют требованиям  
Федерального государственного образовательного стандарта.

Методические указания предназначены для студентов  
направлений технического профиля всех форм обучения.

Текст печатается в авторской редакции

Подписано в печать *26.09.24*. Формат 60x84 1/16.  
Усл. печ. л. 6,16 . Уч.-изд. л. 5,58 . Тираж 100 экз. Заказ. *984* Бесплатно.  
Юго-Западный государственный университет.  
305040, Курск, ул. 50 лет Октября, 94.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	4
1 ОСНОВНЫЕ ПОНЯТИЯ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ В МЕХАТРОННЫХ СИСТЕМАХ .....	6
1.1 Понятие о первичной и вторичной обработке сигналов.....	6
1.2 Технические средства комплекса обработки сигналов.....	8
1.3 Основные типы алгоритмов цифровой обработки сигналов.....	12
1.4 Линейные и нелинейные преобразования .....	14
1.5 Переход от непрерывных сигналов к дискретным .....	17
1.6 Циклическая свертка и корреляция.....	20
1.7 Аперриодическая свертка и корреляция.....	21
1.8 Двумерная аперриодическая свертка и корреляция .....	22
1.9 Контрольные вопросы и задания.....	25
2 ДИСКРЕТНЫЕ ОРТОГОНАЛЬНЫЕ ПРЕОБРАЗОВАНИЯ.....	28
2.1 Введение в теорию ортогональных преобразований .....	28
2.2 Интегральное преобразование Фурье .....	30
2.3 Интегральное преобразование Хартли .....	33
2.4 Дискретное преобразование Фурье .....	36
2.5 Дискретное преобразование Хартли .....	39
2.6 Двумерные дискретные преобразования Фурье и Хартли .....	41
2.7 Ортогональные преобразования в диадных базисах .....	45
2.8 Понятие о Wavelet-преобразованиях. Преобразование Хаара .....	52
2.9 Контрольные вопросы и задания.....	57
ПРИЛОЖЕНИЕ 1. ГЛОССАРИЙ.....	59
ПРИЛОЖЕНИЕ 2. ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ В СРЕДАХ МАТЛАВ И МАТНСАД .....	70
П.2.1 Пример фильтрации сигнала с использованием быстрого прямого и обратного преобразований Фурье .....	70
П.2.2 Пример дифференцирования сигнала .....	88
П.2.3 Пример интегрирования сигнала.....	97
П.2.4 Пример работы с изображениями.....	101
ЛИТЕРАТУРА .....	106

## ВВЕДЕНИЕ

Бурный прогресс вычислительной техники (ВТ) в последние десятилетия привел к широкому внедрению методов цифровой обработки информации практически во всех областях научных исследований и народно-хозяйственной деятельности. При этом среди различных применений средств вычислительной техники одно из важнейших мест занимают системы цифровой обработки сигналов (ЦОС), нашедшие использование при обработке данных дистанционного зондирования, медико-биологических исследований, решении задач навигации аэрокосмических и морских объектов, связи, радиофизики, цифровой оптики и в ряде других приложений [1,6,7,8].

Особое значение цифровая обработка сигналов приобретает в мехатронных объектах, в том числе в робототехнике, где, по полученной с разных датчиков информации, вырабатываются соответствующие управляющие сигналы.

Цифровая обработка сигналов (ЦОС) - это динамично развивающаяся область ВТ, которая охватывает как технические, так и программные средства. Родственными областями для цифровой обработки сигналов являются теория информации, в частности, теория оптимального приема сигналов и теория распознавания образов. При этом в первом случае основной задачей является выделение сигнала на фоне шумов и помех различной физической природы, а во втором - автоматическое распознавание, т.е. классификация и идентификация сигнала.

В теории информации под сигналом понимается материальный носитель информации. В цифровой же обработке сигналов под сигналом будем понимать его математическое описание, т.е. некоторую вещественную функцию, содержащую информацию о состоянии или поведении физической системы при каком-нибудь событии, которая может быть определена на непрерывном или

дискретном пространстве изменения времени или пространственных координат.

В широком смысле под системами ЦОС понимают комплекс алгоритмических, аппаратных и программных средств. Как правило, системы содержат специализированные технические средства предварительной (или первичной) обработки сигналов и специальные технические средства для вторичной обработки сигналов. Средства предварительной обработки предназначены для обработки исходных сигналов, наблюдаемых в общем случае на фоне случайных шумов и помех различной физической природы и представленных в виде дискретных цифровых отсчетов, с целью обнаружения и выделения (селекции) полезного сигнала, его пеленгования и оценки характеристик обнаруженного сигнала. Полученная в результате предварительной обработки полезная информация поступает в систему вторичной обработки для классификации, архивирования, структурного анализа и т.д. [6,7].

Основными процедурами предварительной обработки сигналов являются процедуры быстрых дискретных ортогональных преобразований (БДОП), реализуемых в различных функциональных базисах, процедуры линейной алгебры, линейной и нелинейной фильтрации. Указанные процедуры и быстрые алгоритмы их реализации рассматриваются в данном учебном пособии.

# 1 ОСНОВНЫЕ ПОНЯТИЯ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ В МЕХАТРОННЫХ СИСТЕМАХ

## 1.1 Понятие о первичной и вторичной обработке сигналов

В задачах ЦОС выделяют этапы предварительной (первичной) и вторичной обработки сигналов. Это связано с тем, что в общем случае на входе системы ЦОС наблюдается смесь  $V(t)$  полезного сигнала  $x(t)$ , некоторого шума  $n(t)$  и различных помех разной природы  $p(t)$ :

$$V(t) = x(t) + n(t) + p(t), \quad (1.1)$$

где  $n(t)$  является характеристикой самого технического устройства, а  $p(t)$  - некоторое искажающее воздействие самой физической среды, в которой распространяется сигнал (например, затухание).

Важнейшей задачей предварительной обработки сигнала является подавление  $n(t)$  и  $p(t)$  (шума и помехи). Такая задача может быть решена только на основе использования избыточности представления исходного сигнала, а также имеющихся сведений о свойствах полезного сигнала, помехи и шума для увеличения вероятности правильного приема [6,8].

Вследствие того, что на вход приемного устройства системы поступает сумма полезного сигнала и помехи, вероятность правильного приема будет определяться отношением полезного сигнала к помехе. Для повышения вероятности правильного приема сигнала должна быть произведена предварительная обработка принятого сигнала, обеспечивающая увеличение отношения сигнал/помеха. Таким образом, средства предварительной обработки при приеме должны содержать два основных элемента (рис.1.1): фильтр  $\Phi$ , обеспечивающий улучшение отношения сигнал/помеха, и решающее устройство РУ, выполняющее главные функции приема (обнаружения, различения и восстановления сигналов).

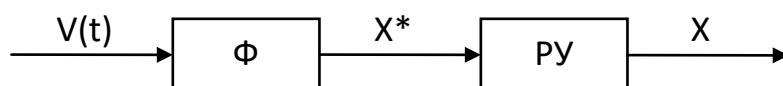


Рисунок 1.1 Структура оптимального приемного устройства

Известны следующие методы фильтрации, обеспечивающие улучшение соотношения сигнал/помеха:

- метод накопления;
- частотная фильтрация;
- корреляционный метод;
- согласованная фильтрация;
- нелинейная фильтрация.

Все эти методы основаны на использовании различий свойств полезного сигнала и помехи.

Кроме того, при предварительной обработке решается задача обнаружения сигнала и определения местоположения его источника. На этапе предварительной обработки в ряде случаев формируются также некоторые количественные оценки сигнала (амплитуда, частота, фаза).

Во входной смеси может и не быть полезного сигнала  $x(t)$ , поэтому на выходе системы предварительной обработки не будет никакого сигнала; следовательно, интенсивность потока данных на выходе будет ниже, чем на входе.

Система вторичной обработки сигнала предназначена для идентификации обнаруженного сигнала, его классификации и выдачи информации об обнаруженных сигналах оператору или формирования управляющего воздействия.

Характерной чертой предварительной обработки сигнала является

постоянство алгоритма обработки при его достаточно высокой вычислительной сложности. Этап вторичной обработки характеризуется большей гибкостью используемых алгоритмов, необходимостью поддержки обмена с другим техническим средством или диалога с оператором. Поэтому системы вторичной обработки чаще всего строятся на основе программируемых вычислительных средств. Системы же предварительной обработки могут быть построены как на программируемых вычислительных средствах, так и на основе специальных вычислителей с жесткой логикой [7].

## 1.2 Технические средства комплекса обработки сигналов

Комплекс цифровой обработки сигналов содержит ЭВМ, специализированные устройства ввода и соответствующее программное обеспечение. В общем случае подобный комплекс должен также обеспечивать ввод, вывод и передачу сигналов различной физической природы. Общие требования к системам ЦОС представлены в таблице 1.1.

При этом особый интерес представляет обработка двумерных сигналов - изображений, получаемых от различных приемных устройств.

Многие задачи обработки изображений могут быть решены на современных персональных ЭВМ, если к скорости обработки не предъявляются высокие требования. В этом случае те или иные процедуры обработки изображений на ПЭВМ реализуются путем создания специального программного обеспечения. Для обеспечения ввода изображения в реальном масштабе времени используются специализированные устройства ввода. К такому типу систем относятся системы IMAGE-3 и Microsight-2. Заметим, что в них обработка изображений производится на ПЭВМ не в реальном масштабе времени. Для обработки сигнальной информации в реальном масштабе времени требуется производительность, превышающая производительность ПЭВМ. В



этом случае необходимы специализированные устройства обработки.

В настоящее время, согласно литературе, известны два типа систем обработки сигналов [6,7,8,10].

Таблица 1.1 - Основные требования к системам ЦОС

Параметр	Значение параметра	Область применения
Динамический диапазон обрабатываемых данных (бит)	Фиксированная точка: 8-16	Сжатие изображений. Радиолокация. Связь
	Плавающая точка: 16-32	Гидроакустика. Обработка речи. Обработка изображений
Емкость памяти буферных устройств (слов)	64-128 не менее 1К более 4К до 16К	Сжатие изображений. Радиолокация. Гидроакустика. Обработка изображений
Производительность (оп/сек)	$10^8$	Системы связи Обработка акустических сигналов Обработка изображений Радиолокация и гидроакустика
	$10^9$	
	$10^9-10^{10}$ $10^{10}-10^{11}$	

Первый тип систем ЦОС предусматривает построение конструктивно законченного блока. Как правило, такой блок имеет модульную структуру и строится на базе специализированных сверхбольших интегральных схем (СБИС), что позволяет обеспечить аппаратную реализацию подлежащего исполнению алгоритма и оптимизировать структуру аппаратных средств под особенности алгоритма. К этому направлению можно отнести системы Series- 151 и MaxVideo. В ряде случаев такие процессоры могут программироваться в целях выполнения тех или иных функций, как, например, WARP-процессор [1].

Отличительной чертой такой архитектуры является наличие отдельных магистралей ввода/вывода данных и возможность автономного функционирования. Блок со спецпроцессором при этом может быть выполнен в стандартном конструктиве типа VME, CAMAC, Multibys [6,7].

Такая система ЦОС допускает не только ввод, но и обработку изображений в реальном масштабе времени, поэтому подобный подход весьма эффективен при построении систем обработки видеоданных.

Второй тип систем ЦОС представляет собой ЭВМ со специализированным сопроцессором в виде платы, подключаемой к магистрали ЭВМ и конструктивно встраиваемый в ее корпус. Примером такой архитектуры могут служить наборы модулей фирмы Data Translation на базе сигнальных процессоров типа TMS и плат-акселераторов фирмы INMOS на базе транспьютеров T800. Указанные технические средства ориентированы на использование в качестве периферийных спецпроцессоров для построения систем на базе IBM PC/AT. Спецпроцессор, входящий в эту систему, имеет, как правило, конвейерную структуру и может выполнять процедуры обработки изображений, требующие больших вычислительных затрат, в реальном масштабе времени. Настройка на выполнение тех или иных конкретных алгоритмов обработки видеоинформации производится программированием спецпроцессора, что увеличивает функциональную гибкость подобных систем и расширяет области их возможного применения.

На практике первый тип систем ЦОС наиболее часто используется в составе средств предварительной обработки сигналов, причем соответствующие вычислительные средства строятся по принципу операционного автомата с жесткой логикой. Такой подход связан с автономностью функционирования средств предварительной обработки от управляющей ЭВМ при неизменном алгоритме обработки и высокой интенсивности входного потока данных.

Второй тип систем используется, как правило, для систем, сочетающих средства предварительной (спецпроцессоры) и вторичной (ЭВМ) обработки, когда требуется достаточно интенсивный обмен с оператором.

### 1.3 Основные типы алгоритмов цифровой обработки сигналов

Математические методы обработки сигналов можно подразделить на три группы, если в основу классификации положить принцип формирования отдельного элемента (отсчета) результата по некоторой совокупности элементов (отсчетов) исходного сигнала [10].

Точечные преобразования – в таких преобразованиях обработка каждого элемента исходных данных производится независимо от соседнего. Иначе говоря, значение каждого отсчета результата определяется как функция от одного отсчета исходного сигнала, причем номера отсчетов сигнала и результата одинаковы.

Иначе говоря, пусть требуется обработать вектор из  $n$  отсчетов сигнала:

$$X = [x_0 \ x_1 \ x_2 \ \dots \ x_n] \quad (1.2)$$

и получить последовательность чисел:

$$Y = [y_0 \ y_1 \ y_2 \ \dots \ y_n] \quad (1.3)$$

Причем

$$y_i = f(x_i) \quad (1.4)$$

Точечные преобразования достаточно просты и наименее громоздки с точки зрения вычислительных затрат. Если обрабатывается матрица размером  $N \times N$  элементов отсчетов исходного двумерного сигнала, то вычислительная

сложность процедуры точечных преобразований составит

$$Q_m = N^2 \text{ (БО)},$$

где под базовой операцией (БО) понимается операция вида (1.4).

Локальные преобразования - при локальных преобразованиях обеспечивается формирование каждого элемента матрицы или вектора результата как функции от некоторого множества соседних элементов матрицы или вектора отсчетов исходного сигнала, составляющих некоторую локальную окрестность. При этом полагается, что местоположение вычисляемого отсчета результата (или текущий индекс элемента) задается координатами (или текущими индексами) центрального элемента локальной окрестности. Для формирования следующего элемента матрицы результата выполняется смещение окрестности вдоль строки матрицы исходных данных или вдоль исходного вектора. Такая перемещаемая окрестность часто носит название окна сканирования. При обработке матрицы исходных данных после прохождения всей строки матрицы исходных данных окно сканирования смещается на одну строку и возвращается в начало следующей строки, после чего продолжается обработка. Просматриваемая при перемещении окна сканирования полоса строк матрицы носит название полосы сканирования. Иначе говоря, при такой обработке

$$y_i = F(X^{\wedge}); X^{\wedge} = \{x_{i-m/2}, x_{i-m/2+1}, \dots, x_i, \dots, x_{i+m/2-1}, x_{i+m/2}\} \quad (1.5)$$

где  $i = 0, N - 1$  - индекс отсчета результата,  $m$  - размер окна сканирования. Если  $i < m/2$  или  $i > N - m/2$ , что имеет место на практике при обработке начальных и конечных отсчетов вектора исходного сигнала, то элементы вектора исходных данных с "недостающими" индексами полагаются равными нулю.

Вычислительная сложность локального преобразования составляет

$$Q_{\text{л}} = N^2 m^2 \text{ (БО)},$$

где под базовой операцией понимается выполнение заданного преобразования для отдельного отсчета исходных сигналов. Примером

локальных преобразований могут служить аperiodическая свертка или корреляция (см. раздел 1.7), а также процедуры ранговой фильтрации.

Глобальное преобразование предусматривает формирование каждого отсчета результата как функции от всей совокупности отсчетов исходного сигнала и некоторого множества меняющихся от одного отсчета результата к другому по определенному правилу коэффициентов, составляющих так называемое ядро преобразования. В случае обработки одномерного исходного сигнала глобальное преобразование можно определить как

$$Y_i = F(G_i, X); \quad X = [x_0 \quad x_1 \quad x_2 \quad \dots \quad x_n] \quad i = 0, N - 1 \quad (1.6)$$

где  $G_i$  – изменяемое ядро преобразования. Вычислительная сложность глобального преобразования для случая обработки двумерного сигнала составляет

$$Q_r = N^4 \text{ (БО)},$$

где под базовой операцией понимается выполнение заданного преобразования вида (1.6) для отдельного элемента исходных данных. Примером подобных преобразований могут служить дискретные ортогональные преобразования типа преобразования Фурье, Хартли, Адамара.

#### 1.4 Линейные и нелинейные преобразования

Все преобразования ЦОС могут быть подразделены по своему типу на линейные и нелинейные преобразования [4,10,11,12].

Пусть  $x(t)$  - входная последовательность, а  $y(t_1)$  – выходная последовательность, связанная со входной через некоторое функциональное

преобразование  $T$ :

$$y(t_1) = T[x(t)] \quad (1.7)$$

Для линейных преобразований справедлив аддитивный закон:

$$T[ax_1(t) + bx_2(t)] = aT[x_1(t)] + bT[x_2(t)] = ay_1(t_1) + by_2(t_1), \quad (1.8)$$

где  $a$  и  $b$  – некоторые константы. Таким образом, линейное преобразование, применяемое к суперпозиции исходных сигналов эквивалентно по своему воздействию суперпозиции результатов преобразования каждого из сигналов. Свойство линейности является весьма важным для практических приложений, поскольку позволяет значительно упростить обработку различных сложных сигналов, являющихся суперпозицией некоторых элементарных сигналов. Так, в частности, за простейший элементарный сигнал может быть принят моногармонический сигнал  $x(t)$ , описываемый функцией:

$$x(t) = a \cos(2\pi vt - \varphi),$$

где  $a$  - амплитуда,  $v = \frac{1}{T}$  - частота,  $T$  - период,  $\varphi$  - начальная фаза.

Тогда более сложный полигармонический сигнал может быть записан как суперпозиция простейших моногармонических сигналов:

$$\hat{x}(t) = \sum_{k=0}^K a_k \cos(2\pi v_k t - \varphi_k)$$

Важное место в цифровой обработке сигналов имеет некоторый идеализированный простейший импульсный сигнал, называемый дельта-функцией или единичным импульсом:

$$\delta(x) = \begin{cases} 0, & x \neq 0 \\ 1, & x = 0 \end{cases}$$

Согласно теории цифровой обработки сигналов, любой сигнал может

быть представлен как суперпозиция взвешенных единичных импульсов следующим образом:

$$x(t) = \sum_{k=-\infty}^{\infty} x(t_k) \delta(t - t_k) \quad (1.9)$$

где  $x(t)$  – отсчет сигнала в некоторый момент времени.

Если на вход системы ЦОС, выполняющей линейное преобразование, поступает единичный импульс, то сигнал  $h(t)$ , снимаемый с выхода системы и являющийся откликом системы на единичный импульс, носит название импульсной характеристики (импульсного отклика) системы. Импульсный отклик является важнейшей характеристикой системы и позволяет описать ее как “черный ящик”, задав реакцию системы на некоторый простейший эталонный сигнал.

Если  $h(t)$  конечна, то такие системы называются КИХ- системами, т.е. системами с конечной импульсной характеристикой. Если  $h(t)$  бесконечна, то это БИХ-системы, т.е. системы с бесконечной импульсной характеристикой. В цифровой обработке сигналов имеет смысл рассматривать только КИХ-системы, поскольку время обработки, т.е. реакции системы на входной сигнал должно быть конечно.

Подставив (1.9) в (1.7), получаем для линейных преобразований:

$$\begin{aligned} y(t_1) &= T[x(t)] = T \left[ \sum_{k=-\infty}^{\infty} x(t_k) \delta(t - t_k) \right] = \sum_{k=-\infty}^{\infty} x(t_k) T[\delta(t - t_k)] = \\ &= \sum_{k=-\infty}^{\infty} x(t_k) h_k(t) \end{aligned} \quad (1.10)$$

Таким образом, для линейной системы результат обработки любого поступившего на вход сложного сигнала может быть определен как суперпозиция импульсных откликов системы на поступившие на вход единичные импульсы с соответствующей начальной задержкой и весом, определяемым весом соответствующего отсчета исходного сигнала.



Примерами линейных преобразований могут служить преобразования Фурье, Хартли, свертка и корреляция. К нелинейным преобразованиям относятся, в частности, многие алгоритмы распознавания, гистограммные преобразования и ранговая фильтрация.

### 1.5 Переход от непрерывных сигналов к дискретным

Процесс перехода от непрерывной области изменения аргумента (задания функции) к конечному множеству отдельных значений аргумента называется дискретизацией. Процесс перехода от непрерывной области изменения функции к конечному множеству определенных значений называется квантованием. Обычно полагается, что дискретизация и квантование выполняются с равными шагами, т.е. функция определена в равноотстоящих точках по оси абсцисс и по оси ординат. Переход от непрерывного сигнала к дискретному осуществляется с потерей информации. Восстановление непрерывного сигнала по дискретным значениям и устранение потерь информации зависит от параметров дискретизации - т.е. шага дискретизации, способа восстановления сигнала и от свойств сигнала.

Условие, при котором возможно восстановление сигнала без потерь, определяется из теоремы Котельникова [10,12].

Пусть функции  $f(x)$  и  $F(\xi)$  связаны обратным преобразованием Фурье, т.е.:

$$f(x) = \int_{-\xi_{max}}^{\xi_{max}} F(\xi) e^{j2\pi\xi x} dx$$

**Прямая формулировка теоремы Котельникова.** Если функция  $f(x)$  имеет ограниченный спектр, локализованный в диапазоне  $-\xi_{max} \leq \xi \leq \xi_{max}$ , то она полностью определена путем задания отсчетов на наборе точек,

отстоящих друг от друга на расстоянии  $1/2 \xi_{max}$ .

**Обратная формулировка теоремы Котельникова.** Если  $f(x)$  задана в ограниченной области  $-x_{max} \leq x \leq x_{max}$ , то ее спектр  $F(\nu)$  полностью определен набором отсчетов в точках, равноотстоящих друг от друга на расстоянии  $1/2 x_{max}$ . Поясним выбор шагов дискретизации по теореме Котельникова на рисунке 1.2.

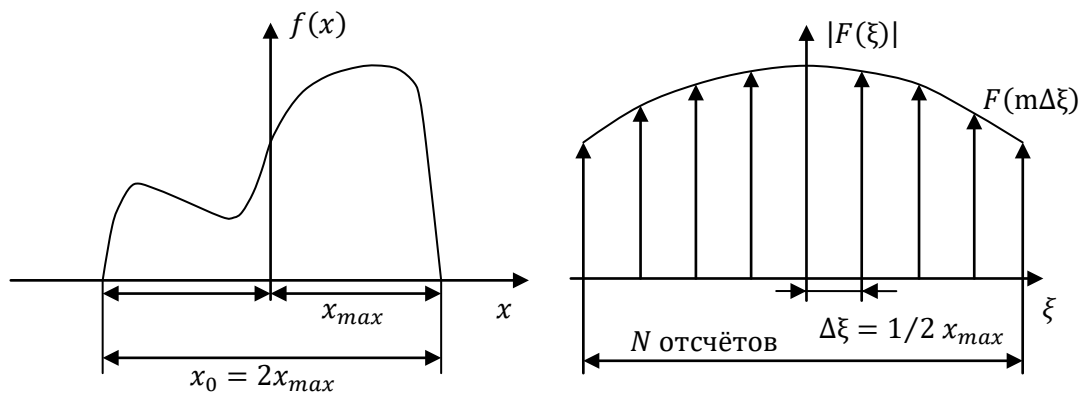


Рисунок 1.2 Выбор шага дискретизации по теореме Котельникова

При дискретизации согласно теореме Котельникова исходная функция  $f(x)$  может быть получена по ее дискретным значениям по формуле:

$$f(x) = \sum_{k=-\infty}^{\infty} f(k\Delta x) \frac{\sin 2\pi\nu \max(x-k\Delta x)}{\nu \max(k\Delta x)}, \quad (1.11)$$

причем шаг дискретизации составляет

$$\Delta x = \frac{1}{2\nu \max} = \frac{\pi}{\omega \max}$$

функции  $f(x)$  соответствует бесконечный спектр и наоборот, конечный спектр соответствует бесконечной исходной функции.

Поэтому для реальных сигналов условия теоремы Котельникова в строгом смысле слова не выполняются.

1. Все реальные сигналы ограничены во времени и имеют неограниченный спектр, т.е.  $f_B = \infty$ .
2. В соответствии с рядом Котельникова восстановление осуществляется по бесконечному числу отсчетов  $-\infty \leq k \leq \infty$
3. Поскольку сигнал восстанавливается по бесконечному числу отсчетов функций, то его восстановление осуществляется с бесконечной задержкой во времени.

Поэтому, чтобы получить конечный спектр, можно воспользоваться равенством Парсеваля:

$$\left| \int_0^T f(t) dt \right| \equiv \left| \int_{-v}^v F(v) dv \right| (1 - \varepsilon)$$

и, зная, что исходная функция  $f(x)$  конечна, вычислить значение интеграла в левой части равенства, после чего, задав величину погрешности  $\varepsilon$  в определении интеграла в правой части, определить максимальную частоту. Исходя из максимальной частоты определяется число отсчетов. Если размер области задания исходной функции  $f(x)$  равен  $X = 2X_{max}$ , а число отсчетов функции при дискретизации должен составлять  $N$ , то шаги дискретизации исходной функции и ее спектра составят:

$$\begin{cases} \Delta v \leq \frac{1}{2x_{max}}; & V_{max} = \frac{N}{2} \Delta v; \\ \Delta x \leq \frac{1}{2V_{max}}; & X_{max} = \frac{N}{2} \Delta x; \end{cases} \quad (1.12)$$

Итак, в результате дискретизации в соответствии с теоремой Котельникова от  $x(t)$  мы переходим к набору отсчетов или к вектору:

$$X = \{x_n\}; \quad X = [x_0 \quad x_1 \quad x_2 \quad \dots \quad x_{N-1}]. \quad n = 0, N - 1.$$

## 1.6 Циклическая свертка и корреляция

В дискретном виде линейные преобразования могут быть описаны в общем виде как векторно-матричные операции [3,4,10]:

$$Y = B_N X \quad (1.13)$$

где  $X$  - вектор отсчетов исходных данных, полученный в результате дискретизации непрерывного сигнала согласно теореме Котельникова,  $Y$  - вектор отсчетов результата,  $B_N$  - матрица размера  $N \times N$ , определяющая ядро выполняемого преобразования.

К числу подобных преобразований относится циклическая свертка последовательностей  $X = [x_0 \ x_1 \ x_2 \ \dots \ x_{N-1}]$  и  $G = [g_0 \ g_1 \ g_2 \ \dots \ g_{N-1}]$ , в этом случае строится матрица ядра свертки:

$$B_N^C = G_N = \begin{vmatrix} g_0 & g_{N-1} & g_{N-2} & \dots & g_1 \\ g_1 & g_0 & g_{N-1} & \dots & g_2 \\ g_2 & g_1 & g_0 & \dots & g_3 \\ \dots & \dots & \dots & \dots & \dots \\ g_{N-1} & g_{N-2} & g_{N-3} & \dots & g_0 \end{vmatrix}$$

Каждый элемент вектора  $Y$  может быть описан как:

$$y_m = \sum_{n=0}^{N-1} g_{m-n} x_n; \quad m = \overline{0, N-1} \quad (1.14)$$

Матрица ядра циклической взаимокорреляции может быть построена как транспонированная матрица ядра свертки, т.е. следующим образом:

$$K_N = B_N^K = \begin{vmatrix} g_0 & g_1 & g_2 & \dots & g_{N-1} \\ g_{N-1} & g_0 & g_{N-1} & \dots & g_{N-2} \\ g_{N-2} & g_{N-1} & g_0 & \dots & g_{N-3} \\ \dots & \dots & \dots & \dots & \dots \\ g_1 & g_2 & g_3 & \dots & g_0 \end{vmatrix}$$

Поэтому каждый отсчет результата может быть записан как:

$$Z_m = \sum_{n=0}^{N-1} g_{m+n} x_n; \quad m = \overline{0, N-1} \quad (1.15)$$

Причем  $g_{m+n} = g_n$  для  $m+n \geq N$ .

### 1.7 Аперриодическая свертка и корреляция

Аперриодическая свертка и корреляция в отличие от циклической относятся к классу локальных преобразований. При этом как правило полагается, что размер вектора исходных данных значительно больше размера ядра свертки, что приводит к следующему выражению для вычисления любого отсчета результата:

$$y_i = \sum_m g_m x_{i-m} \quad (1.16)$$

Вычисление свертки и корреляции лежит в основе корреляционного метода подавления помех.

Сущность такого метода заключается в использовании различия между корреляционными функциями сигнала и помехи. Данный метод эффективен лишь в случае обработки периодических или квазипериодических сигналов.

Рассмотрим сущность метода на примере, когда полезный сигнал является гармоническим, а помеха - типа белого гауссова шума [12].

Автокорреляционная функция сигнала является тоже гармонической и имеет ту же частоту. Метод автокорреляционного приема основан на анализе автокорреляционной функции принятого сигнала  $y(t) = x(t) + p(t)$ .

Если сигнал и помеха взаимно независимы (типичный для практики случай), то

$$K_y(\tau) = K_x(\tau) + K_r(\tau),$$

т.е. автокорреляционная функция принятого сигнала равна сумме

автокорреляционных функций сигнала и помехи.

Метод корреляционного приема позволяет обнаружить полезный сигнал, который имеет мощность значительно меньшую, чем мощность помехи.

## 1.8 Двумерная аperiodическая свертка и корреляция

Важное место среди операций линейной обработки сигналов занимает операция перемножения матриц одинаковой размерности. Такая операция имеет вид:

$$Z_N = B_N X_N, \quad (1.17)$$

где  $B_N$  и  $X_N$  исходные матрицы порядка  $N$ , а  $Z_N$  - результирующая матрица того же порядка. Каждый элемент матрицы  $Z_N$  формируется в соответствии с выражением:

где  $x_{ik}$ ,  $b_{kj}$ , - элементы исходных матриц.

На основе операций (1.17) и (1.18) выполняется вычисление функций двумерной аperiodической свертки/корреляции исходного двумерного сигнала (изображения) с двумерным ядром. Подобное преобразование часто используется как для удаления шумов, так и для выделения мелких объектов.

Математически двумерная аperiodическая свертка может быть описана следующим образом:

$$y_{ij} = \sum_{n=1}^M \sum_{m=1}^M g_{mn} x_{i+m-1, j+n-1} \quad (1.20)$$

где  $y_{ij}$  - отсчеты результатов вычислений (отсчеты свертки),  $g_{mn}$  - отсчеты весовой функции окна (ядра свертки) размерностью  $M \times N$  отсчетов, причем  $N \gg M$ . Очевидно, что размерность матрицы, описывающей двумерную

свертку, равна  $(N + M - 1) \times (N + M - 1)$  отсчетов. Поэтому матрица исходных данных также должна быть дополнена до указанного размера нулевыми элементами по краям кадра.

Отсчеты свертки формируются при перемещении окна вдоль строки исходного изображения. Для каждого положения окна формируется один отсчет свертки, после чего окно сдвигается на один элемент вдоль строки (т.е. на один столбец). Обработка начинается с элемента  $x$  исходного изображения. После прохождения  $i$ -й строки изображения ( $i = 1, N$ ) окно смещается на одну строку вниз и возвращается к началу следующей ( $i + 1$ )-й строки изображения. По окончании обработки кадра изображения окно перемещается в исходное положение.

При вычислении свертки можно распараллелить вычисления по столбцам окна сканирования, выполняя параллельно вычисление произведений столбцов

ядра свертки  $g_n^{\wedge}$  и столбцов  $X_{j+n-1}^{\wedge}$  текущей полосы сканирования изображения. Поэтому для окончательного вычисления отсчета свертки достаточно сформировать сумму:

$$Y_j = \sum_n g_n^{\wedge} X_{j+n-1}^{\wedge}, \quad (1.20)$$

соответствующую отсчету свертки с номером  $j$ , расположенному в средней строке полосы шириной  $M$ .

Представляет интерес распараллеливание по разрядным срезам, поскольку в этом случае практически исключается операция умножения [10]. При разрядно-срезовой обработке данные должны быть представлены в формате с фиксированной точкой. Представим значение элемента изображения в следующем виде:

$$x_{mn} = \sum_q b_{mn}^q 2^{q-1}, \quad (1.21)$$

где  $b_{mn}^q$  - $q$ -ый разряд,  $x_{mn}(q = 1, \dots, Q), (g = (1, Q))$ , где  $Q$  - разрядность данных.

С учетом этого процедура вычисления свертки принимает вид:

$$y_{ij} = \sum_q 2^{q-1} \sum_m \sum_n g_{mn} b_{i+m-1, j+n-1}^q = \sum_q 2^{q-1} Y_q \quad (1.22)$$

Таким образом, процедура двумерной аperiodической свертки для одного положения окна сводится к  $M \times N \times Q$  операциям сложения и  $(Q - 1)$  операциям сдвига. Умножение под знаком суммы сводится к операции вида:

$$g_{mn} b_{mn}^q = g_{mn} \text{ при } b_{mn}^q = 1 \text{ и } g_{mn} b_{mn}^q = 0 \text{ при } b_{mn}^q = 0 .$$

Поэтому разрядно-срезовый алгоритм вычисления свертки или корреляции для одного положения окна может быть представлен в виде [10]:

начало;

$S := 0;$

для  $q = 1, \dots, Q$  цикл:

$S1 := 0;$

для  $m = 1, \dots, M$  цикл:

для  $n = 1, \dots, M$  цикл:

если  $b_{i-m, j-n}^q = 1$ , то  $g_{mn} := g_{mn}$

иначе  $g_{mn} := 0;$

$S1 := S1 + g_{mn};$

конец цикла по  $n$ ;

конец цикла по  $m$ ;

$S := S + sh1_{q-1}(S1);$

конец цикла по  $q$ ;

$y_{ij} = S;$

конец.



Физический смысл функций свертки и корреляции состоит в том, что они являются количественной мерой совпадения (сходства) двух последовательностей  $f(x)$  и  $g(x)$ . При этом наиболее полно мера сходства может быть определена по функции корреляции, в связи с чем функция взаимокорреляции (или кросскорреляции) может быть использована для распознавания сигналов. Если распознаваемый сигнал  $f(t)$  точно соответствует эталонному сигналу  $g(t)$ , то результирующий сигнал  $\omega^k(t)$  принимает значение:

$$\omega^k(t) = \max \text{ при } f(t) \equiv g(t),$$

что соответствует функции автокорреляции. Если сигналы отличаются, то  $\omega^k(t) < \max$ .

Кроме того, при обработке двумерных сигналов (изображений объектов) координаты максимума данной функции определяют центр тяжести исходного распознаваемого объекта, что позволяет определить и его местоположение (т.е. запеленговать объект). По этим причинам вычисление одномерной или двумерной корреляции лежит в основе целого ряда методов распознавания.

Таким образом, функции линейной апериодической свертки и корреляции полезны для распознавания сигналов заданной формы. На этом принципе работают корреляционные методы распознавания. Свертка определяется путем скольжения эталона по вектору исходного сигнала, и максимум функции будет тогда, когда исходный сигнал совпал с эталоном. Функция апериодической свертки, кроме того, оказывается полезной для удаления, например, низкочастотных помех.

## 1.9 Контрольные вопросы и задания

1. Сигнал имеет частотный спектр, ограниченный частотой  $F_{\max} = 10$  кГц, причем разрешение по частоте составляет 100 Гц. В течении какого промежутка времени должен наблюдаться сигнал? Через какие промежутки времени должны сниматься отсчеты сигнала?

2. Сигнал наблюдается в течении 10 сек., причем отсчеты сигнала снимаются через 10 мксек. Какова предельная частота сигнала  $F_{\max}$  может быть зафиксирована. Какое разрешение по частоте будет обеспечиваться в этом случае?

3. Заданы последовательности  $G = [0; 1; 2]$  и  $X = [0; 1; 2]$ . Вычислить аperiodическую свертку и корреляцию.

Выполнить оценку вычислительной сложности разрядно-срезового алгоритма сверки/корреляции в сравнении с вычислительной сложностью прямого алгоритма свертки/корреляции.

4. Задан вектор  $X = [0, 0, 1, 1, 2, 3, 2, 1, 0, 1, 0, 0]$ . Определить вектор  $Y$  с отсчетами отфильтрованного сигнала при использовании рекурсивного линейного фильтра с коэффициентами  $H = [1, 3, 1]$  и  $B = [-1/2, 1]$  (краевыми эффектами пренебречь).

5. Реализация линейных пространственных фильтров требует перемещения центра маски по изображению и вычисления, для каждого из положений маски, суммы произведений коэффициентов маски на значения соответствующих пикселей. В случае низкочастотной фильтрации все коэффициенты равны 1, и можно использовать *однородный усредняющий фильтр* или алгоритм *скользящего среднего*, основанный на том, что при переходе от точки к точке обновляется только часть вычисляемых элементов.

(a) Сформулируйте такой алгоритм для фильтра размерами  $n \times n$ , демонстрирующий характер взаимосвязи вычислений с последовательностью сканирования, использующейся при передвижении маски по изображению.

(б) Отношение числа операций, требуемых для реализации метода «в лоб» к числу операций, используемых алгоритмом скользящего среднего называется *эффективностью алгоритма*. Подсчитайте эффективность алгоритма для данного случая и изобразите ее в виде графика зависимости от  $n$  для  $n > 1$ . Коэффициент  $1/n^2$  является общим для обоих случаев, и поэтому не должен приниматься во внимание. Считайте, что изображение окружено бордюром из нулей достаточной ширины, чтобы не учитывать влияние граничных эффектов при вычислениях.

## 2 ДИСКРЕТНЫЕ ОРТОГОНАЛЬНЫЕ ПРЕОБРАЗОВАНИЯ

### 2.1 Введение в теорию ортогональных преобразований

Две вещественные функции  $g(x)$  и  $h(x)$ , заданные на конечном или бесконечном интервале ( $a < x < b$ ), называются ортогональными друг другу на этом интервале, если

$$\int_a^b g(x)h(x)dx = 0$$

При этом функции предполагаются конечными либо бесконечными, но обязательно с абсолютно сходящимся интегралом. Интеграл называется абсолютно сходящимся если

$$\int_a^\infty |f(x)|dx < \infty$$

Система функций называется ортогональной на некотором интервале, если каждые две функции из этой системы ортогональны друг другу на этом интервале.

Пусть задана система функций

$$g_1(x), g_2(x), \dots, g_n(x) \tag{2.1}$$

ортогональная на некотором интервале ( $a < x < b$ ).

Может возникнуть задача о разложении произвольной функции  $f(x)$  на этом интервале в ряд по функциям (2.1), т.е. в ряд вида

$$f(x) = a_1g_1(x) + a_2g_2(x) + \dots + a_n g_n(x) + \dots = \sum_{n=1}^{\infty} a_n g_n(x), \tag{2.2}$$

где  $a_n$  – числовые коэффициенты. При этом возникают вопросы: возможно ли разложение для любой функции  $f(x)$  и как найти коэффициенты  $a_n$ .

Будем считать для простоты все рассматриваемые функции, а также интервал конечным. Ответ на первый вопрос зависит от выбора системы, по которой мы будем производить разложение. Если разложение возможно для любой функции  $f(x)$ , то система функций называется полной [5].

Перейдем теперь к нахождению коэффициентов  $a_n$  разложения, причем будем считать, что ни одна из функций (2.1) не равна тождественно нулю. Для этого умножим обе части уравнения (2.2) на  $g_n(x)$  и проинтегрируем результат по интервалу ( $a < x < b$ ).

$$\begin{aligned} \int_a^b f(x)g_n(x)dx &= \\ &= a_1 \int_a^b g_1(x)g_n(x) + a_2 \int_a^b g_2(x)g_n(x) + \dots + a_n \int_a^b g_n^2(x) + \dots \end{aligned}$$

В силу ортогональности системы (2.1), в правой части последнего равенства все интегралы равны нулю, за исключением интеграла от  $g_n^2(x)$ , и мы получаем формулу для коэффициентов

$$a_n = \frac{\int_a^b f(x)g_n(x)dx}{\int_a^b g_n^2(x)dx} \quad (n = 1, 2, 3, \dots).$$

## 2.2 Интегральное преобразование Фурье

Любой периодический сигнал  $x_{\Pi}(t)$  можно представить в виде ряда Фурье [12]:

$$x_{\Pi}(t) = \frac{A_0}{2} + \sum_{k=1}^{\infty} A_k \cos(k\omega_0 t - \varphi_k), \quad (2.3)$$

где  $A_0 = \frac{2}{T} \int_{t_1}^{t_1+T} x_n(t) dt$  - постоянная составляющая сигнала,  $\omega_0 = 2\pi/T = 2\pi\nu$ . Из выражения (2.3) следует, что периодический сигнал любой формы может быть представлен в виде суммы гармонических составляющих с различными амплитудами  $A_k$ , частотами  $k\omega_0$  - и фазами  $\varphi_k$ . Благодаря этому можно перейти от представления сигнала во временной области к частотной, где  $A_k = A_k(k\omega_0)$  - спектр амплитуд,  $\varphi_k = \varphi_k(k\omega_0)$  - спектр фаз.

Эти спектры являются линейчатыми. Для любого сигнала  $x(t)$  можно определить спектры амплитуд  $A_k$  и фаз  $\varphi_k$  следующим образом:

$$A_k(k\omega_0) = \sqrt{a_k^2 + b_k^2};$$

$$\varphi_k(k\omega_0) = \operatorname{arctg} \frac{b_k}{a_k}.$$

где, в свою очередь, величины  $a_k$  и  $b_k$  определяются следующим образом:

$$a_k = \frac{2}{T} \int_{t_1}^{t_1+T} x_{\Pi}(t) \cos k \omega_0 t dt; \quad b_k = \frac{2}{T} \int_{t_1}^{t_1+T} x_{\Pi}(t) \sin k \omega_0 t dt;$$

Любой периодический сигнал бесконечен во времени, что на практике не осуществимо, поэтому периодический сигнал – математическая абстракция, и все рассмотренное выше не применимо к реальным сигналам.

Реальный сигнал ограничен во времени и, следовательно, является непериодическим. Однако, условно его можно рассматривать как

периодический с периодом  $T \rightarrow \infty$ . Тогда  $\omega_0 = 2\pi/T \rightarrow 0$ , а спектры амплитуд и фаз становятся непрерывными (сплошными), сумма в разложении Фурье превращается в интеграл. В результате переходим к интегралу Фурье (обратное преобразование) [4,12]:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} s(j\omega) e^{j\omega t} d\omega.$$

В формуле  $s(j\omega)$  – спектральная плотность сигнала, определяющая как распределяются амплитуды и фазы по частотам непрерывного спектра. Иногда в задачах обработки сигналов ее называют фурье-образом или фурье-спектром сигнала.

От  $s(j\omega)$  можно перейти к спектральной плотности амплитуд ( $s(\omega)$ ) и фаз ( $\varphi(\omega)$ ).

Для решения этой задачи используется прямое преобразование Фурье:

$$s(j\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt.$$

Важно отметить, что  $s(\omega)$  – всегда убывающая функция, а  $\varphi(\omega)$  – всегда неубывающая функция. Кроме того,  $s(\omega) = s(-\omega)$  – четная функция; а  $\varphi(\omega) = -\varphi(-\omega)$  – нечетная функция.

Таким образом, с точностью до постоянного коэффициента, прямое и обратное преобразование Фурье могут быть определены по соотношениям:

– прямое преобразование Фурье:

$$F(\xi) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi\xi x} dx \tag{2.4}$$

– обратное преобразование Фурье:

$$f(x) = \int_{-\infty}^{\infty} F(\xi) e^{j2\pi\xi x} d\xi \tag{2.5}$$

где  $\xi$  – некоторая пространственно-частотная спектральная координата (аналог координаты  $\nu$  для сигналов во временной области). Напомним основные общие свойства преобразования Фурье:

**Инвариантность к линейному смещению.** Это свойство преобразования Фурье позволяет получать неизменный квадрат Фурье-образа при перемещении исходного объекта вдоль осей координат, что имеет исключительно важное значение при обработке и распознавании образов:

$$f(x) \rightarrow F(\xi)$$

$$f(x-c) \rightarrow F_1(\xi) = F(\xi)e^{-j2\pi\xi c}$$

или  $|F(\xi)| \equiv |F_1(\xi)|$ , где  $c$  – произвольное смещение функции вдоль оси.

**Теорема масштабов или теорема подобия.** Теорема определяет характер изменения спектра при изменении масштаба сигнала

$$f(x) \rightarrow F(\xi)$$

$$f(mx) \rightarrow F_1(\xi) = F(\xi/m)$$

т.е. при растяжении функции  $f(x)$  в  $m$  раз происходит сжатие в  $m$  раз ее Фурье-образа, и наоборот.

**Теорема о свертке.** Пускай требуется вычислить свертку функций  $f(x)$  и  $g(x)$ , причем  $F(\xi)$  и  $G(\xi)$  – соответственно их Фурье-образы:

$$\omega_c = f(x) \times g(x) = \int_{-\infty}^{\infty} f(x - x_1)g(x_1)dx_1$$

$$f(x) \rightarrow F(\xi); \quad g(x) \rightarrow G(\xi)$$

Если функции  $F(\xi)$  и  $G(\xi)$  есть Фурье-образы функции  $f(x)$  и ядра  $g(x)$  соответственно, то  $F(\xi)G(\xi)$  есть Фурье-образ свертки  $\omega_c(x)$ .



**Теорема о корреляции.** Если функции  $F(\xi)$  и  $G(\xi)$  есть Фурье-образы функции  $f(x)$  и ядра  $g(x)$  соответственно, то  $F^*(\xi)G(\xi)$  есть Фурье образ корреляции  $\omega_k(x)$  функций  $f(x)$  и  $g(x)$ .

Теоремы о свертке и корреляции свидетельствуют о возможности вычисления функций свертки и корреляции через преобразование Фурье. Тем самым на основе преобразования Фурье можно выполнять и распознавание, идентификацию сигналов и обнаружение координат источника сигналов.

**Теорема Парсеваля или закон сохранения энергии.** Эта теорема свидетельствует о том, что мощность исходного сигнала и мощность спектра сигнала одинаковы:

$$\left| \int_{-\infty}^{\infty} f(x) dx \right|^2 \equiv \left| \int_{-\infty}^{\infty} F(\xi) d\xi \right|^2$$

Преобразование Фурье является одним из важнейших ортогональных преобразований, используемых в цифровой обработке сигналов. Действительно, вполне физически ясен смысл перехода от временного описания исходного сигнала к его частотному описанию. Кроме того, двумерное преобразование Фурье описывает не что иное, как дифракцию электромагнитных и упругих волн в дальней зоне (дифракцию Фраунгофера) – т.е. на большом (по сравнению с размерами источника и длиной волны) расстоянии от источника [16].

### 2.3 Интегральное преобразование Хартли

Для одномерного случая прямое преобразование Хартли может быть определено как [3]

$$H(\xi) = \int_{-\infty}^{\infty} f(x) [\cos 2\pi\xi x + \sin 2\pi\xi x] dx \quad (2.6)$$

и, соответственно, обратное преобразование Хартли

$$f(x) = \int_{-\infty}^{\infty} H(\xi)[\cos 2\pi\xi x + \sin 2\pi\xi x]d\xi \quad (2.7)$$

Сравним эти выражения с (2.4.) и (2.5), разложив ядро по формуле Эйлера на действительную и мнимую части (т.е.  $\sin$  и  $\cos$  компоненты):

$$F(\xi) = \int_{-\infty}^{\infty} f(x)[\cos 2\pi\xi x - j \sin 2\pi\xi x]dx \quad (2.8)$$

$$f(x) = \int_{-\infty}^{\infty} f(\xi)[\cos 2\pi\xi x + j \sin 2\pi\xi x]d\xi$$

Из анализа (2.6) - (2.8), можно сделать следующие выводы:

- 1) Преобразование Хартли является преобразованием с действительным ядром;
- 2) Прямое и обратное преобразование Хартли вычисляются идентично;
- 3) Квадрат модуля преобразования Фурье  $|F(v)|^2$  равен:

$$|F(v)|^2 = \frac{H^2(v) + H^2(-v)}{2} \quad (2.11)$$

- 4) Действительная и мнимая компоненты преобразования Фурье могут быть вычислены на основе преобразования Хартли весьма простым образом:

$$Re\{F(\xi)\} = \frac{H(\xi) + H(-\xi)}{2} \quad (2.10)$$

$$Im\{F(\xi)\} = \frac{H(\xi) - H(-\xi)}{2} \quad (2.11)$$

- 5) Если  $f(x)$  – четная (т.е.  $f(-x)=f(x)$ ), то:

$$H(\xi) \equiv Re\{F(\xi)\}, Im\{F(\xi)\} \equiv 0.$$

Основные свойства преобразования Хартли соответствуют преобразованию Фурье:

1) Инвариантность к сдвигу (модуль  $H^2(\zeta) + H^2(\bar{\zeta})$  – неизменен).

2) Так же, как и для преобразования Фурье, для преобразования Хартли справедливы следующие соотношения согласно теоремы масштабирования:

$$f(x) \rightarrow H(\xi)$$

$$f(m_x x) \rightarrow H_1(\xi) = \frac{H_1\left(\frac{\xi}{m_x}\right)}{m_x}$$

3) Так же, как и для преобразования Фурье, для преобразования Хартли справедлива Теорема Парсеваля.

Отличие от Фурье-преобразования заключается в иной трактовке теоремы о свертке:

Если заданы функции  $f(x)$  и  $g(x)$ , причем  $H(\xi)$  и  $G(\xi)$  – соответственно их спектры Хартли:

$$f(x) \rightarrow H(\xi),$$

$$g(x) \rightarrow G(\xi),$$

то их свертка вычисляется следующим образом [3]:

1) вычисляются функции  $H(-\xi)$  и  $G(-\xi)$ ;

2) формируется функция:

$$\Phi(\zeta) = \frac{[(H(\xi) + H(-\xi))G(\xi) + (H(\xi) - H(-\xi))G(-\xi)]}{2}$$

3) вычисляется преобразование Хартли от функции  $\Phi(\xi)$ .

Очевидно, что если функция  $g(x)$  – четная, то:

$$G(\xi) \equiv G(-\xi)$$

$$\Phi(\xi) = H(\xi)G(\xi),$$

Если и функция  $f(x)$  – четная, то:

$$H(\xi) \equiv H(-\xi)$$

$$\Phi(\xi) = G(\xi)H(\xi)$$

Преобразование Хартли требует вычислений примерно вдвое меньшей сложности (поскольку его ядро действительная функция) и в то же время от его результата достаточно просто перейти к результату, эквивалентному результату преобразования Фурье. Поэтому на практике преобразование Хартли используется вместо преобразования Фурье в различных задачах ЦОС как некоторое искусственное синтетическое преобразование меньшей сложности, но обеспечивающее получение требуемого результата.

## 2.4 Дискретное преобразование Фурье

Перейдём от интегрального преобразования Фурье (2.3) к дискретному преобразованию Фурье (ДПФ), при условии, что точки дискретизации выбраны согласно теоремы отсчётов (теоремы Котельникова) [4,12]:

$$\begin{aligned} F(\Delta\xi) &= \sum_{k=0}^{N-1} x(k\Delta x)\delta(x - k\Delta x) * e^{-j2\pi k\Delta\xi\Delta x} = \sum_{k=0}^{N-1} x_k e^{-j2\pi \frac{2x_{max}}{N} * k * \frac{1}{2x_{max}}} \\ &= k * \sum_{k=0}^{N-1} x_k * e^{-j\frac{2\pi}{N}kn} \end{aligned}$$

$$\text{где } \begin{cases} \frac{2x_{max}}{N} = \Delta x \\ \frac{1}{2x_{max}} = \Delta \xi \end{cases} \quad (2.11)$$

Тогда нетрудно получить, что

$$\Delta x * \Delta \xi = \frac{1}{N}$$

Подобным же образом можно получить и для обратного преобразования

$$x_m = k \sum F_n * e^{j\frac{2\pi}{N}nm} \quad k = \frac{1}{\sqrt{N}} \quad (2.12)$$

Заметим, что происхождение множителя  $k = \frac{1}{\sqrt{N}}$  связано с заменой при дискретизации согласно теоремы Котельникова, восстанавливающую функцию в (1.12) на “гребёнку” отсчётов.

Таким образом, в матричной форме:

$$F = \frac{1}{\sqrt{N}} E_N X \quad (2.13)$$

$$E_N = \begin{pmatrix} W^0 & W^0 & W^0 & \dots & W^0 \\ W^0 & W^1 & W^2 & \dots & W^{N-1} \\ W^0 & W^2 & W^4 & \dots & W^{2(N-1)} \\ W^0 & W^3 & W^6 & \dots & W^{3(N-1)} \\ \dots & \dots & \dots & \dots & \dots \\ W^0 & W^{N-1} & W^{2(N-1)} & \dots & W^{(N-1)(N-1)} \end{pmatrix}$$

где  $W^k = e^{-j2\pi\frac{k}{N}}$ , а сама матрица ядра ДПФ носит название матрицы дискретных экспоненциальных функций (ДЭФ). При этом строки матрицы определяют набор ортогональных функций или базис разложения.

При выполнении преобразования Фурье строки матрицы ядра задают набор ортогональных функций, по которым выполняется разложение исходного сигнала. Каждый элемент вектора результата определяет вклад соответствующей ортогональной функции в формирование исходного сигнала.

Для преобразования Фурье, как и для любого ортогонального преобразования, матрица ядра преобразования  $E_N$  обратима (т.е. определитель отличен от “0”), что позволяет выполнить как прямое, так и обратное преобразования:

$$\begin{cases} F = \frac{1}{\sqrt{N}} E_N X \\ X = \frac{1}{\sqrt{N}} E_N^{-1} F \end{cases} \quad (2.14),$$

$$\text{Поскольку } \frac{1}{N} E_N * E_N^{-1} = \frac{1}{N} I_N = \begin{vmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \end{vmatrix}$$

При этом матрица ядра обратного преобразования обладает свойством  $E_N^{-1} = E_N^*$ , где  $E_N^*$  - эрмитово-сопряжённая матрица. Понятие эрмитово-сопряжённой матрицы предусматривает, что матрица обратного преобразования является транспонированной по отношению к  $E_N$  и элементы её есть комплексно сопряжённые к  $W_{ij}$ .

Рассмотрим основные свойства матрицы ядра преобразования  $E_N$ . Коэффициенты такой матрицы обладают следующими свойствами:

- 1) цикличностью:  $W^{k+N} \equiv W^k$  или  $W^{-k} \equiv W^{N-k}$
- 2) мультипликативностью  $W^{k+m} \equiv W^k * W^m$

Из указанных свойств следует, матрица  $E_N$  из  $N^2$  элементов содержит только  $N$  попарно различных элементов.

- 3) симметричностью  $W_{ij} \equiv W_{ji}$

Рассмотрим примеры матрицы  $E_N$  для некоторых  $N$ :

$N=2$

$$E_2 = \begin{vmatrix} 1 & 1 \\ 1 & -1 \end{vmatrix}$$

N=3

$$E_3 = \begin{vmatrix} W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 \\ W^0 & W^2 & W^1 \end{vmatrix}$$

$$W^4 \equiv W^{3+1} \equiv W^1$$

$$E_4 = \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{vmatrix}$$

и, наконец, для N=8

$E_8$

$$= \begin{vmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \frac{\sqrt{2}}{2}(1-j) & -j & -\frac{\sqrt{2}}{2}(1+j) & -1 & \frac{\sqrt{2}}{2}(-1+j) & j & \frac{\sqrt{2}}{2}(1+j) \\ 1 & -j & -1 & j & 1 & -j & -1 & j \\ 1 & -\frac{\sqrt{2}}{2}(1+j) & j & \frac{\sqrt{2}}{2}(1-j) & -1 & \frac{\sqrt{2}}{2}(1+j) & -j & \frac{\sqrt{2}}{2}(-1+j) \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & \frac{\sqrt{2}}{2}(1-j) & j & \frac{\sqrt{2}}{2}(1+j) & -1 & \frac{\sqrt{2}}{2}(1-j) & -j & \frac{\sqrt{2}}{2}(-1-j) \\ 1 & j & -1 & -j & 1 & -j & -1 & -j \\ 1 & \frac{\sqrt{2}}{2}(1+j) & j & \frac{\sqrt{2}}{2}(-1+j) & -1 & -\frac{\sqrt{2}}{2}(1+j) & -j & \frac{\sqrt{2}}{2}(1-j) \end{vmatrix}$$

## 2.5 Дискретное преобразование Хартли

Дискретное преобразование Хартли имеет вид [3]:

- прямое преобразование

$$h_n = \sum_{k=0}^{N-1} x_k \cos \frac{2\pi kn}{N} + \sin \frac{2\pi kn}{N} = \sum_{k=0}^{N-1} x_k \operatorname{cas} \frac{2\pi kn}{N} \quad (2.15)$$

$$\operatorname{cas}[\dots] = \cos[\dots] + \sin[\dots]$$

- обратное преобразование

$$x_m = \sum_{n=0}^{N-1} h_n \operatorname{cas} \frac{2\pi nm}{N}$$

Матрица ядра преобразования Хартли может быть записана как:

$$C_N = \begin{pmatrix} C_0 & C_0 & C_0 & \dots & C_0 \\ C_0 & C_1 & C_2 & \dots & C_{N-1} \\ C_0 & C_2 & C_4 & \dots & C_{2(N-1)} \\ \dots & \dots & \dots & \dots & \dots \\ C_0 & C_{N-1} & C_{2(N-1)} & \dots & C_{(N-1)^2} \end{pmatrix}$$

где  $C_k = \operatorname{cas} \frac{2\pi k}{N}$ , причем  $C_0 \equiv 1$

Матрица ядра преобразования обладает следующими свойствами:

- 1) цикличностью  $C_{k+N} \equiv C_k$  ;  $C_{N-k} \equiv C_{-k}$ ;
- 2) отсутствием мультипликативности, т.е.:  $C_{k+l} \neq C_k * C_l$
- 3) симметричностью  $C_{ij} \equiv C_{ji}$

Отсюда следует, что обратная матрица  $C_N^{-1} \equiv C_N$ , поскольку  $C_N \equiv C_N^T$ .

4) из свойства цикличности следует, что в матрице  $C_N$  имеется лишь  $N$  различных между собой коэффициентов  $C_{ij}$  из  $N^2$

Для  $N=2$ ,  $N=4$  и  $N=8$  матрицы ядра преобразования будут иметь вид:

$$N=2 \quad C_2 = \begin{vmatrix} 1 & 1 \\ 1 & -1 \end{vmatrix} = E_2$$



$$N=4 \quad C_4 = \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{vmatrix} \neq E_4$$

$$N=8 \quad C_8 = \begin{vmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \sqrt{2} & 1 & 0 & -1 & -\sqrt{2} & -1 & 0 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 0 & -1 & \sqrt{2} & -1 & 0 & 1 & -\sqrt{2} \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -\sqrt{2} & 1 & 0 & -1 & \sqrt{2} & -1 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 0 & -1 & -\sqrt{2} & -1 & 0 & 1 & \sqrt{2} \end{vmatrix}$$

Таким образом, в отличие от матрицы ДЭФ, матрица ядра преобразования Хартли содержит коэффициенты, большие единицы.

## 2.6 Двумерные дискретные преобразования Фурье и Хартли

Для обработки двумерных сигналов, в частности, изображений, широко используются двумерные спектральные преобразования. Двумерное дискретное преобразование Фурье имеет вид [8,12]:

$$f_{k,l} = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x_{m,n} e^{-j\frac{2\pi}{N}[km+ln]} \quad (2.16)$$

Поскольку ядро преобразования Фурье является разделимым по переменным интегрирования, то для выполнения двумерного ДПФ на практике используется строчно-столбцовый метод.

Действительно,

$$e^{-j\frac{2\pi}{N}[km+ln]} = e^{-j\frac{2\pi}{N}km} e^{-j\frac{2\pi}{N}ln}$$

откуда получаем с учетом (2.16):

Введя обозначение

$$\hat{f}_{k,l} = \sum_{m=0}^{N-1} x_{mn} e^{-j\frac{2\pi}{N}km} \quad (2.17)$$

получим:

$$\hat{f}_{k,l} = \sum_{m=0}^{N-1} \hat{f}_{k,l} e^{-j\frac{2\pi}{N}ln} \quad (2.18)$$

Согласно такому подходу, вначале выполняются одномерные ДПФ по строкам матрицы с отсчетами изображения, а затем выполняются одномерные ДПФ по столбцам матрицы промежуточных результатов, представляющих собой одномерные спектры по строкам изображения. В матричном виде это может быть записано следующим образом [9]:

$$F_N = [E_N X_N] E_N \quad (2.19).$$

Из (2.19) следует, что двумерное ДПФ сводится вначале к одномерному ДПФ матрицы  $X$  по столбцам, а затем к одномерному ДПФ по строкам матрицы промежуточных результатов  $\hat{F}_N$ .

Сложность вычислений на первом этапе составит  $N$  раз по  $N^2$  базовых операций, под которыми понимаются операции вычисления выражения под знаком суммы в (2.17) и (2.18), и столько же на втором, откуда:  $Q_{\text{ДПФ}} = 2N^3$  (Б.О.)

В то же время непосредственные вычисления двумерного ДПФ по формуле (2.16) требуют вычислительных затрат:  $Q_{\text{ДПФ}}^1 = 2N^4$  (Б.О.)

Таким образом, строчно-столбцовый метод позволят существенно снизить вычислительную сложность алгоритма двумерного ДПФ с  $N^4$  до  $2N^3$  операций умножения. Вычисление одномерного ДПФ по каждой из координат преобразование выполняется на основе процедуры быстрого

преобразования Фурье. Такой подход, однако, требует выполнения дополнительной процедуры транспонирования матрицы промежуточных результатов после выполнения обработки по одной из координат матрицы. При этом преобразование по следующей координате может выполняться только после того, как сформирована вся матрица промежуточных результатов и выполнено ее транспонирование.

Поэтому использование свойства разделимости ядра двумерного ДПФ по переменным суммирования приводит к снижению сложности вычислений в  $N/2$  раз.

Однако далеко не все ядра двумерных спектральных ортогональных преобразований в явном виде обладают свойством разделимости. К таким преобразованиям относится, например, двумерное дискретное преобразование Хартли [3]:

$$H_{kl} = \sum_m \sum_n x_{mn} \left[ \frac{\text{cas}(2\pi(km + ln))}{N} \right] \quad (2.20)$$

$$\text{cas}[\dots] = \cos[\dots] + \sin[\dots]$$

Рассмотрим применение специального или модифицированного преобразования Хартли, в котором искусственно выполнено разделение ядра по координатам [3]:

$$H_{kl}^{\wedge} = \sum_n \left[ \sum_m x_{mn} \text{cas}\left(\frac{2\pi km}{N}\right) \text{cas}(2\pi ln/N) \right] \quad (2.21)$$

Очевидно, что выражение (2.21) может быть представлено в виде:

$$H_{kl}^{\wedge} = \sum_m \left[ \cos\left(\frac{2\pi km}{N}\right) + \sin\left(\frac{2\pi km}{N}\right) \right] \\ * \left\{ \sum_n x_{mn} [\cos(2\pi ln/N) + \sin(2\pi ln/N)] \right\}$$

Представляет интерес установить взаимосвязь между традиционным двумерным преобразованием Хартли (выражение (2.20)) и модифицированным преобразованием Хартли, определяемым согласно выражению (2.21). После ряда тригонометрических и алгебраических преобразований ядра в выражении (2.21) нетрудно получить [13]:

$$H_{kl} = [H_{kl}^{\wedge} + H_{k,-l}^{\wedge} + H_{-k,l}^{\wedge} - H_{-k,-l}^{\wedge}]/2 \quad (2.22)$$

где в силу цикличности ядра  $(N-k) \bmod N = -k$ ;  $(N-l) \bmod N = -l$ .

От компонент спектра Хартли  $H_{kl}$  можно перейти к компонентам спектра Фурье, используя известное соотношение:

$$F_{kl} = \{[H_{kl} + H_{-k,-l}] - j[H_{kl} - H_{-k,-l}]\}/2$$

Если записать  $H_{-k,-l}$  в виде, подобном выражению (2.22), то получим выражение, определяющее связь между компонентами спектра Фурье и компонентами модифицированного преобразования Хартли:

$$F_{kl} = \{[H_{k,-l}^{\wedge} + H_{-k,l}^{\wedge}] - j[H_{k,l}^{\wedge} - H_{-k,-l}^{\wedge}]\}/2 \quad (2.23)$$

Следует подчеркнуть, что определение  $F_{kl}$  как через традиционное, так и через модифицированное преобразование Хартли требует вычисления только лишь  $H_{kl}$  или соответственно. Остальные требуемые слагаемые легко могут быть найдены путем перекомпоновки матрицы  $H_{kl}$  или  $H_{kl}^{\wedge}$

Выражения (2.22) и (2.23) справедливы для любых действительных функций  $x_{mn}$ . В частных случаях, когда  $x_{mn}$  обладает симметрией относительно одной или двух координат, эти выражения существенно упрощаются.

Пусть  $x_{mn} = x_{m,-n}$  Тогда  $H_{kl}^{\wedge} = H_{k,-l}^{\wedge}$ ,  $H_{-k,-l}^{\wedge} = H_{-k,l}^{\wedge}$ , а формулы (2.22) и (2.23) приобретают вид:

$$H_{kl} = \hat{H}_{kl}$$

$$F_{kl} = \{[\hat{H}_{kl} + \hat{H}_{-k,l}] - j[\hat{H}_{kl} - \hat{H}_{-k,l}]\}/2$$

Если  $x_{m,n} = x_{-m,n} = x_{m,-n}$ , то как преобразование Хартли, так и преобразование Фурье эквивалентны модифицированному преобразованию Хартли:

$$H_{kl} = \hat{H}_{kl}; F_{kl} = \hat{H}_{kl}$$

## 2.7 Ортогональные преобразования в диадных базисах

Ортогональные преобразования в диадных (или иначе говоря, двузначных знакопеременных) базисах определены для данных, представленных векторами длиной  $N=2^M$ . К таким преобразованиям относятся преобразования Адамара, Пэли, Уолша, Трахтмана, Качмарджа и ряд других [5,8,12]. Матрица ядра любого из подобных преобразований содержит целочисленные коэффициенты из множества  $\{-1; +1\}$ . Очевидно, что при выполнении подобных преобразований существенно сокращается объем вычислений за счет исключения умножения в каждой базовой операции.

Матрица ядра преобразования Уолша - Адамара для  $N=2^m$  может быть описана как результат кронекеровского произведения  $m$  матриц ДЭФ  $E_2$  размера  $2 \times 2$  [5,9]:

$$A_N = A_{2^m} = E_2 \otimes E_2 \otimes E_2 \otimes \dots \otimes E_2 = [E_2]^m = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}^m, \quad (2.24)$$

где символ  $\otimes$  - операция кронекеровского умножения векторов, в результате чего порождается матрица блочной структуры. Заметим, что операция кронекеровского умножения двух матриц и состоит в получении

блочной матрицы, блоками которой является умноженная на соответствующий элемент правой матрицы левая матрица, т.е.:

$$A_4 = E_2 \otimes E_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{array}{cc|cc} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ \hline 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{array}$$

аналогично можно получить и для  $N = 8$ :

$$A_8 = E_2 \otimes E_2 \otimes E_2 = A_4 \otimes E_2 = \begin{array}{cc|cc|cc|cc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ \hline 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ \hline 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ \hline 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{array}$$

Матрица ядра Адамара обладает следующими свойствами:

- 1) цикличностью  $a_{N+k} = a_k$ ;  $a_{N-k} = a_{-k}$
- 2) мультипликативностью  $a_{k+l} = a_k * a_l$
- 3) симметричностью  $A_N = A_N^T$

В задачах ЦОС используются и другие, подобные Адамару, преобразования - Пэли, Уолша, Трахтмана и других. Ядра (матрицы) этих преобразований могут быть получены на основе матрицы ядра преобразования Адамара при определенном переупорядочении их строк. Поэтому перед рассмотрением указанных преобразований рассмотрим правила переупорядочения матриц, применяемые для формирования ядер таких преобразований.

Переупорядочим элементы вектора  $[x_0, x_1, x_2, \dots, x_{N-1}]$  таким образом, чтобы первые элементы вектора были бы чётными, а вторые - нечётными. Повторим эту процедуру до тех пор, умножая каждый раз длину соответствующего вектора вдвое, пока длина подвектора не станет равной 2. В результате получится переупорядоченный вектор, элементы которого упорядочены по закону так называемой двоичной инверсии.

В частности, для  $N=8$  такое переупорядочение будет выполняться следующим образом:

$$X = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} \Rightarrow \begin{pmatrix} x_0 \\ x_2 \\ x_4 \\ x_6 \\ x_1 \\ x_3 \\ x_5 \\ x_7 \end{pmatrix} \Rightarrow \begin{pmatrix} x_0 \\ x_4 \\ x_2 \\ x_6 \\ x_1 \\ x_5 \\ x_3 \\ x_7 \end{pmatrix} = \hat{X}$$

Вектор  $\hat{X}$  можно получить из  $X$ , если задать следующее правило формирования индекса текущего элемента вектора  $\hat{X}$  из индекса элемента  $x_i$  исходного вектора  $X$ :

- 1) записывается двоичный код текущего индекса  $i$ ;
- 2) двоичный код записывается в обратном порядке как рефлексивный или отраженный код (т.е. начиная с младших разрядов);
- 3) полученный код определяет индекс текущего элемента вектора  $\hat{X}$ ;
- 4) переводим полученный индекс в десятичную систему.

Для рассмотренного выше случая ( $N=8$ ) получим:

$$\begin{array}{c} \left| \begin{array}{l} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{array} \right| \Rightarrow \left| \begin{array}{l} 000 \\ 100 \\ 010 \\ 110 \\ 001 \\ 101 \\ 011 \\ 111 \end{array} \right| \Rightarrow \left| \begin{array}{l} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{array} \right| \end{array}$$

Процедура перестановки в терминах векторно-матричных операций может быть записана как:

$$\hat{X} = S_N X \quad (2.25)$$

где  $S_N$  - перестановочная матрица. Для случая двоично-инверсной перестановки перестановочная матрица будет иметь вид:

$$S_8 = \left| \begin{array}{cccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right|$$

Элементы вектора могут быть также переупорядочены по коду Грея:

$$\left| \begin{array}{l} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{array} \right| \Rightarrow \left| \begin{array}{l} 000 \\ 001 \\ 011 \\ 110 \\ 110 \\ 111 \\ 101 \\ 100 \end{array} \right|$$

В терминах векторно-матричных операций подобная перестановка элементов вектора может быть описана так:

$$\hat{X} = \Gamma_N X$$



где  $\Gamma_N$  - перестановочная матрица, в частности, для  $N=8$ :

$$\Gamma_8 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Ядро преобразования Пэли можно получить из ядра преобразования Адамара, переупорядочив строки матрицы  $A_N$  по закону двоичной инверсии, в частности, для  $N=8$  получаем:

$$\Gamma_8 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 \end{pmatrix} \begin{matrix} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{matrix}$$

В общем случае правило перехода от матрицы Адамара к матрице Пэли может быть представлено как [5]:

$$A_N \xrightarrow{\text{Дв.инверсия}} P_N \quad (2.26)$$

Если теперь переупорядочить строки матрицы ядра преобразования Пэли по коду Грея, то получим матрицу ядра преобразования Уолша [7]

$$W_8 = \begin{array}{cccccccc|ccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 000 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 001 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 011 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 010 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 110 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 111 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 101 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 100 \end{array}$$

Или в общем виде:

$$A_N \xrightarrow{\text{Дв.инв.}} P_N \xrightarrow{\text{Грей}} W_N \quad (2.27)$$

$$P_N \xrightarrow{\text{Грей}} W_N \quad (2.28)$$

Очевидно, что свойства матриц  $P_N$  и  $W_N$  аналогичны свойствам матрицы  $A_N$ . Матрицы ядер преобразования Трахтмана и Качмарджа также могут быть получены на основе матрицы ядра Адамара, но при использовании обратной перестановки по коду Грея.

Заметим, что матрицы преобразований в указанных базисах отличаются порядком строк. Очевидно, что при обработке одного и того же вектора исходных данных  $X$  в различных базисах вектор результата будет содержать одинаковые по своей величине элементы, отличаясь для каждого преобразования лишь порядком их следования. Поэтому для всех подобных

преобразований можно использовать одну и ту же матрицу ядра (например,  $A_N$ ), а для получения вектора  $F$  с нужным для каждого преобразования порядком следования элементов, переупорядочить элементы исходного вектора  $X$ . Так, для преобразования Пэли необходима двоично-инверсная перестановка, а для преобразования Уолша - двоично-инверсная перестановка и затем перестановка по коду Грея.

Если проанализировать все рассмотренные во второй главе преобразования, то можно прийти к выводу, что их сущность состоит в разложении исходной функции на ряд чётных и нечётных составляющих, которые задаются строками матриц  $E_N$ ,  $H_N$ ,  $A_N$ ,  $W_N$  и  $P_N$ . Так, для матриц  $E_N$  и  $H_N$  это набор гармонических функций  $\cos[...]$  и  $\sin[...]$ , а для  $A_N$ ,  $W_N$  и  $P_N$  - наборы прямоугольных знакопеременных функций.

Тем самым, в зависимости от вида исходной функции, в её составе будут определены отдельные компоненты (и их частотное значение), которые задаются строками матрицы ядра преобразования. Очевидно, что разложение по ортогональным функциям, задаваемым строками матрицы  $E_N$ , позволяет определить частотный состав исходной функции (сигнала), что имеет вполне понятный физический смысл.

Точно также при разложении по функциям, задаваемым строками матриц  $A_N$ ,  $W_N$  и  $P_N$ , можно определить, какой вклад вносит та или иная знакопеременная функция в состав исходного сигнала.

Очевидно, что исходный сигнал может быть разложен по различным ортогональным составляющим. При этом вклад таких составляющих в исходный сигнал будет различен для разных матриц ортогональных преобразований.

Согласно теореме Коруэна-Лоева [12], может быть определено оптимальное разложение исходной функции по набору ортогональных функций в смысле наименьшего числа отличных от нуля компонент такого разложения.

На практике для сигналов гармонической природы удобно использовать различные гармонические функции, т.е. для определения частотного состава сигнала выполнить преобразование Фурье или Хартли (ДПФ или БПФ). Действительно, для моногармонического сигнала лишь одна компонента

разложения по строкам матрицы  $E_N$  будет отлична от “0” (при условии, что  $f_0$  кратно  $k\pi$ ).

Для сигналов, описываемых знакопеременными функциями, близким к оптимальному является разложение по знакопеременным функциям типа Уолша, Адамара. Поэтому в задачах кодирования и распознавания речи, где для представления сигналов широко используется метод широтно-импульсной модуляции, удобно выполнять разложение по ортогональным функциям, задаваемым строками матриц  $A_N$ ,  $W_N$  или  $P_N$ .

## 2.8 Понятие о Wavelet-преобразованиях. Преобразование Хаара

В ряде случаев оказывается более удобным в качестве базисов разложения использовать такие системы функций, для которых коэффициенты разложения учитывают поведение исходной функции лишь в нескольких близкорасположенных точках. Использование такого базиса по своей сути означает переход от частотного анализа к масштабному, т.е. функция  $f(x)$  анализируется с помощью некоторой “стандартной” математической функции, изменяемой по масштабу и сдвигу на некоторую величину.

Первое упоминание об этих функциях появилось в работах Хаара в 1909 году. В 30-е годы начались более детальные исследования возможностей представления сигналов с использованием базисных масштабируемых функций. Пол Леви, используя масштабируемую базисную функцию типа функции Хаара, исследовал разновидность случайного сигнала – броуновское движение. Он обнаружил преимущество в применении базисных функций Хаара перед функциями Фурье.

В период 60-х - 80-х годов Вейс и Кофман исследовали простейшие элементы функционального пространства, названные ими атомами, с целью обнаружить атомы для произвольной функции и найти "правила сборки", позволяющие реконструировать все элементы функционального пространства, используя эти атомы. В 1980 году Гроссман и Марлет определили такие функции как Wavelet-функции. В переводе с английского Wavelet – всплеск, поэтому в отечественной литературе встречается термин «разложение по всплескам» наряду с вейвлет-анализом.

В конце 80-х г.г. Мейер и Добичи на основе исследований Марлета создали ортогональные базисы Wavelet-функций, которые и стали основой современных Wavelet-функций. Сходство Wavelet и Фурье преобразований состоит в следующем:

1. Оба они являются линейными преобразованиями и предназначены для обработки блоков данных, содержащих  $\log_2 N$  элементов.

2. Обратные матрицы ДПФ и DWT (discret wavelet transform) равны транспонированной, причем строки самих матриц содержат функции  $\cos(x)$  и  $\sin(x)$ , а для DWT - более сложные базисные функции - wavelet.

Наиболее важное различие между этими двумя видами преобразований состоит в том, что отдельные функции wavelet локализованы в пространстве, а синусные и косинусные - нет. Благодаря этой особенности, DWT находит большое число применений, в том числе для сжатия данных, распознавания образов и подавления шумовой составляющей принимаемого сигнала.

Преобразование Wavelet состоит из неограниченного набора функций. Семейство Wavelet различают по тому, насколько компактны базисные функции в пространстве и насколько они гладки. Некоторые из них имеют фрактальную структуру. В каждом семействе они могут быть разбиты на подклассы по числу коэффициентов и уровню итераций. Чаще всего внутри семейства функция классифицируется по номеру моментов исчезновения.

Набор дискретных Wavelet-функций в общем виде может быть описан как

$$W_{[s,l]}(x) = 2^{-s/2}W(2^{-s}x-l), \quad (2.27)$$

где  $s$  и  $l$  – целые числа, которые масштабируют и сдвигают материнскую функцию  $W(x)$  для создания wavelet.

Индекс масштаба  $s$  показывает ширину wavelet, а индекс смещения  $l$  определяет ее позицию. Материнские функции масштабированы или растянуты коэффициентом, кратным степени 2 и приведены к целому. Таким образом, если известна материнская функция, то может быть построен и весь базис. В свою очередь, само Wavelet-преобразование может быть записано в виде [2,14]:

$$V(x) = \sum_{k=1}^{N-2} (-1)^k z_{k+l} W(2x + l) \quad (2.28)$$

где  $Z_{k+l}$  – отсчеты исходного сигнала.

Метод разложения по всплескам широко используется для выделения шумовой компоненты при обработке данных.

К wavelet-подобным функциям относятся функции Хаара [5]. Для  $N=4$  и  $N=8$  матрицы ядра преобразования имеют вид:

$$X_4 = \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{vmatrix}$$

$$X_8 = \begin{vmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{vmatrix}$$

Основные свойства матрицы ядра преобразования Хаара состоят в следующем:

а) ее элементы не мультипликативны

б) матрица не симметрична, откуда следует, что для обратного преобразования матрицу ядра необходимо транспонировать, т.е.

$$X_N^{-1} = X_N^T$$

в) строки матрицы определяют периодические функции с периодом  $N$ .

Однако матрица ядра преобразования Хаара является не ортонормированной, т.е. для данного преобразования не выполняется теорема Парсеваля. Поэтому для выполнения теоремы Парсеваля требуется ввести дополнительную нормировку, такая нормировка заключается в умножении на  $\sqrt{2}$  тех элементов вектора результатов, которые соответствуют строкам матрицы с нулевыми компонентами. Для строк, содержащих  $N/2$  ненулевых элемента такое умножение выполняется один раз, для строк с  $N/4$  ненулевыми элементами - два раза и так далее. Подобную нормировку необходимо выполнять как при выполнении прямого, так и обратного преобразования Хаара.

Близким к преобразованию Хаара является усеченное преобразование Адамара, отличающееся, по своей сути, лишь порядком следования строк матрицы ядра преобразования [5].

Итак, любое линейное преобразование может быть записано как векторно-матричная процедура:

$$F = kB_N X$$

где  $X = [x_0, x_1, \dots, x_{N-1}]^T$  - вектор исходных данных;

$B_N$  - квадратная матрица размера  $N \times N$  ядра преобразования;

$F = [f_0, f_1, \dots, f_{N-1}]^T$  - вектор результатов.

Если для матрицы  $B_N$  существует обратная матрица  $B_N^{-1}$  (что справедливо для всех ортогональных преобразований), то существует и обратное преобразование:

$$X = kB_N^{-1} F.$$

Методы дискретных ортогональных преобразований в настоящее время широко используются при обработке сигналов. В таблице 2.2 приведены основные области применения спектральных методов и задачи, решаемые с их помощью [8].



Задачи ЦОС, решаемые методами дискретных ортогональных преобразований

Решаемые задачи	Область применения	Выполняемые преобразования
Формирование характеристик направленности антенных устройств	Радиофизика, радиолокация, гидроакустика	Двумерное ДПФ матрицы данных
Выделение сигнала на фоне шумов	Радиофизика, радиолокация, обработка изображений	Прямое ДПФ + сглаживание спектра + обратное ДПФ
Обнаружение и определение координат объекта на изображении	Радиофизика, гидроакустика, обработка изображений	Прямое ДПФ + согласованная фильтрация спектра + обратное ДПФ
Определение скорости движущегося объекта	Радиолокация, гидроакустика	Двумерное косинусное прямое или обратное преобразование по окну
Сжатие и восстановление изображений	Системы передачи, обработки и архивации изображений	Двумерное косинусное прямое или обратное преобразование по окну
Восстановление изображения по проекциям	Томография	Преобразование Радона для векторов данных через ДПФ

### 2.9 Контрольные вопросы и задания

1. Проверить, является ли ортогональным ядро преобразования для  $N=4$

а)  $f_k = \sum x_n (-1)^{kn}$  .

б)  $f_k = \sum x_n \cos[2\pi kn/N]$

$$в) f_k = \sum x_n \sin[2\pi kn/N]$$

2. Покажите, что преобразование Фурье и обратное преобразование суть линейные операции (относительно линейности см. раздел 2.6).

3. Пусть  $F(u,v)$  — ДПФ изображения  $f(x,y)$ . Из обсуждения в Разделе 4.2.3 нам известно, что умножение  $F(u,v)$  на функцию фильтра  $H(u,v)$  и вычисление обратного преобразования Фурье изменит вид изображения в соответствии с характером используемого фильтра. Пусть  $H(u,v)=A$ , где  $A$  — некоторая положительная константа. Используя теорему о свертке, объясните математически, почему элементы изображения в пространственной области умножаются на ту же самую константу.

4. Постройте перестановочную матрицу, с помощью которой можно было бы сгруппировать вначале нечетные элементы вектора, а затем четные (и те и другие - в порядке возрастания индексов) для  $N = 8$ .

5. Выполнить двоично-инверсную перестановку вектора данных

$$X = [x_0 ; x_1 ; x_2 ; x_3 ; \dots ; x_{N-2} ; x_{N-1} ]^T \quad (N = 16)$$

6. Выполнить перестановку вектора данных

$$X = [x_0 ; x_1 ; x_2 ; x_3 ; \dots ; x_{N-2} ; x_{N-1} ]^T \text{ с тройным прореживанием } (N = 9).$$

## ПРИЛОЖЕНИЕ 1. ГЛОССАРИЙ

### Б

**Базис (Basis)** – совокупность векторов пространства  $v$ , обладающих следующим свойством: любой вектор пространства может быть представлен единственным образом как их линейная комбинация  $v = \sum b_k v_k$ . Число базисных векторов равно размерности пространства. Аналогично и последовательность функций  $v_k(t)$  является базисом функционального пространства, если любая функция из этого пространства может быть представлена единственным образом как  $\sum b_k v_k(t)$ . Числа  $b_k$  называются коэффициентами разложения.

**Банк фильтров (Filter Bank)** – совокупность фильтров и следующих за ними дециматоров или следующих перед ними интерполяторов. Под дециматором понимается устройство, осуществляющее децимацию (прореживание) сигнала. Интерполятор выполняет интерполяцию сигнала. Банки фильтров бывают равномерные и неравномерные, ортогональные, биортогональные, двухканальные и многоканальные и т.д. Каждый фильтр банка фильтров образует канал. Поэтому говорят об  $M$ -канальном банке фильтров. Сигнал в канале называется субполосой. Отсюда название субполосная фильтрация (субполосное кодирование). В случае, если число каналов равно коэффициенту децимации (интерполяции) говорят о максимально (критически) децимированном банке фильтров. Быстрый алгоритм вычисления вейвлет-преобразования строится именно на основе таких банков фильтров. Равномерный банк фильтров – децимация в каждом канале одинаковая. В противном случае – неравномерный банк фильтров. Частный случай неравномерного банка фильтров - древовидный банк фильтров.

**Быстрое вейвлет-преобразование (Fast Wavelet Transform)** - пирамида Малла (Mallat). Входной сигнал подается на пару фильтров-дециматоров. Выход ВЧ фильтра-дециматора считается коэффициентами, а выход НЧ фильтра поступает на точно такую же схему (то есть в алгоритме выполняется итерация по НЧ каналу). Сложность - не более  $2LN$ , где  $L$  - длина фильтра,  $N$  -длина сигнала. В самом деле, на первом уровне сложность выполнения фильтрации  $LN$ , на втором -  $LN/2$  и т.д.; сумма последовательности  $1+1/2+1/4+...$  сходится к 2.

### В

**Вейвлет-пакеты (Wavelet Packet)**. Если в пирамиде Малла выполнять разложения не только по НЧ, но и по ВЧ каналу, то, в конечном счете, мы получим дерево базисов, аналогичное получаемому при выполнении

быстрого преобразования Фурье. Это множество базисов и называется вейвлет-пакетами. Функции вейвлет-пакетов ортогональны между собой на каждом уровне, а также и между уровнями.

**Вейвлет-фрейм (Frame)** - преобразование, сочетающее в себе свойства вейвлет-рядов и непрерывного вейвлет-преобразования. Вейвлеты соседних уровней являются сжатыми копиями друг друга (как у вейвлет-рядов), а преобразование ведется для всех сдвигов вейвлета (как при непрерывном ВП).

**Вейвлеты (Wavelet)** - это математические функции, обладающие некоторыми свойствами. В научном сообществе до сих пор не решен вопрос, какие функции относить к вейвлетам. В узком смысле это семейство функций, получающихся путем масштабирования и сдвигов одной, материнской функции. Именно за счет изменения масштабов вейвлеты способны выявить особенности сигнала на разных шкалах, а за счет сдвигов они способны проанализировать сигнал во всех точках. В широком смысле вейвлеты - это функции, обладающие хорошей частотно-временной локализацией, чье среднее значение равно нулю. При этом они могут вовсе не иметь функции-прототипа (например, вейвлеты второго поколения, названные так В.Свелденсом). В математических кругах вейвлеты назывались одно время всплесками, но сейчас этот термин встречается редко. В приложениях чаще всего используют дискретные вейвлеты, так как непрерывные вейвлеты не образуют ортонормированного базиса и бесконечно протяжены (имеют экспоненциальный спад). С другой стороны, применение непрерывного вейвлет-анализа позволяет лучше визуализировать результаты анализа (получить более «красивые картинки») и, возможно, выявить какие-то скрытые от других видов анализа свойства сигнала.

**Все пропускающий (Allpass) фильтр** – фильтр, частотная характеристика которого вещественна, а ее модуль равен 1.

Г

**Гладкость (Smoothness)** - параметр, ограничивающий сверху число производных функции. Различные вейвлет-фильтры обладают различной гладкостью (то есть, конечно, гладкостью обладают функции, получаемые в пределе при бесконечном числе итераций пирамиды Малла) Например, считается что сигнал изображения имеет гладкость 1,5-2. Поэтому, для его анализа, сжатия можно применять вейвлет-фильтры соответствующей или большей гладкости. Как правило, чем больше коэффициентов у вейвлет-фильтра, тем больше гладкость соответствующей ему функции.

## Д

**Децимация (прореживание) (Decimation)** – операция, заключающаяся в выбрасывании отсчетов, чей порядковый номер кратен определенному числу. Например, при децимации в два раза выбрасывается каждый второй отсчет, при прореживании в три раза – каждый третий и т.д. Совокупность фильтра и дециматора называется фильтром-дециматором.

**Дискретное преобразование Хаара (Haar transform)** – разложение сигналов по ортонормальному базису вейвлетов Хаара.

## И

**Избыточное (redundancy) и безизбыточное преобразования.** Преобразование считается безизбыточным, если в коэффициентах содержится ровно столько информации, чтобы можно было совершить обратное преобразование. При анализе сигнала этой информации нам может показаться мало, а обратное преобразование и не потребоваться. В этом случае прибегают к избыточному преобразованию. Например, в случае вейвлет-фреймов число коэффициентов на каждом уровне может равняться числу отсчетов в исходном сигнале, а число уровней быть весьма значительным (например, 64, 128).

**Интерполяция (Interpolation)** – операция, заключающаяся во встраивании между отсчетами, чей порядковый номер кратен определенному числу, некоторой константы. Совокупность интерполятора и фильтра называется фильтром-интерполятором.

## К

**Каузальный (Causal) фильтр** -  $h(k) = 0$  при  $k < 0$ , то есть реакция фильтра не может предшествовать приложенному воздействию. Система каузальна, если выходной сигнал  $y(n)$  зависит от входного сигнала только в моменты времени  $n$  и раньше.

**Компактный носитель** масштабирующей функции или вейвлета. Носитель - часть области определения функции, где она не равна нулю. Свойство компактности функций означает, что соответствующие им фильтры будут фильтрами с конечной импульсной характеристикой, то есть иметь конечное число коэффициентов.

**Контейнер** – сообщение, в которое внедряется скрытая информация. Обычно представляет собой речь, изображение, видео, аудиосигнал. Различают потоковый (формируемый в реальном времени) и фиксированный (например,

изображение) контейнеры. Контейнеры могут быть случайными, выбранными из некоторого множества, навязанными, а также являться функцией скрытого сообщения. Ясно, что в зависимости от типа контейнера, появляются различные особенности встраивания скрытых сообщений.

**Кратномасштабный (многомасштабный) анализ (КМА) – Multiresolution Analysis (MRA)** - математическая конструкция, схема представления сигналов. Эта конструкция позволяет «с другой стороны» зайти к построению вейвлет-рядов и заключается в представлении пространства (например,  $L^2(R)$ ) в виде бесконечной последовательности вложенных подпространств, являющихся отмасштабированными версиями друг друга и связанных определенными свойствами. Различают ортогональный и биортогональный КМА. Значение КМА заключается также и в том, что при его помощи мы можем дать более точные определения масштабирующей и вейвлет-функциям, а также найти связь между ними.

## Л

**Линейное преобразование (Linear transform)** – преобразование, удовлетворяющее условию: преобразование суммы сигналов равно сумме преобразований каждого сигнала. Большинство рассматриваемых в ЦОС преобразований – линейно.

## М

**Маска (Mask)** - в литературе по аппроксимации так называется вектор, в ЦОС называемый вектором коэффициентов фильтра.

**Масштабограмма (Scalogram)** - квадрат амплитуды коэффициентов непрерывного вейвлет-преобразования. Изображается обычно на плоскости. По оси абсцисс откладывается, например, время, а по оси ординат - масштабы, на которых сигнал анализировался. Масштабограмма аналогична спектрограмме, построенной при спектральном анализе с постоянной относительной полосой (constant-Q). Под спектрограммой понимается квадрат амплитуды коэффициентов оконного преобразования Фурье.

**М-полосные вейвлеты.** В этом случае имеется одна масштабирующая функция и несколько вейвлет-функций на каждом уровне анализа. В пирамиде Малла сигнал на каждом уровне пропускается через М фильтров.

**Мультивейвлеты** получаются за счет отказа от стационарности характеристик фильтров. Они могут применяться для многоканальных сигналов, причем каждый канал обрабатывается «своим» фильтром. Мультивейвлеты могут применяться и для обычных сигналов после их

полифазной декомпозиции. Интересной особенностью мультивейвлетов является возможность строить ортогональные симметричные базисы, что было невозможным в случае обычных вейвлетов (кроме фильтров Хаара).

## Н

**Нульдерев**о – одна из наиболее эффективных структур, упорядочивающих множество вейвлет-коэффициентов изображения для целей их сжатия. Заключается в следующем: вейвлет-коэффициенты более НЧ областей «отвечают» за четыре коэффициента более ВЧ областей. Тем, в свою очередь, соответствуют 16 коэффициентов еще более ВЧ областей и т.д. Таким образом, получаем дерево коэффициентов с вершиной в самой НЧ области. Выберем некоторый порог и назовем меньшие этого порога (по абсолютной величине) коэффициенты незначимыми. Начиная с самой НЧ области проверяем значимость коэффициентов относительно текущего порога. Если коэффициент незначимый, проверяем его потомков по дереву. Если все потомки незначимы, то на месте коэффициента ставится признак нульдеревца. Таким образом одним символом обозначаем большое множество коэффициентов. После «обхода» всех коэффициентов порог уменьшается в два раза, и процесс повторяется. Сжатие на основе нульдеревца было впервые предложено Льюисом и Ноулесом (1992) и улучшено в дальнейшем Шапиро (1993) и Саидом-Перельманом (1995).

## О

**Оконное (Short-Time) преобразование Фурье (STFT) - (ОПФ)** - определяется как преобразование Фурье сигнала, умноженного на некоторую взвешивающую функцию  $w$  (окно). Так как окно локализовано по времени, ОПФ применяется для получения частотно-временного представления сигнала. Выражение для ОПФ сигнала  $f(t)$  может быть записано в виде:

$$F(\omega, \tau) = \int_{-\infty}^{\infty} w(t - \tau) f(t) e^{-j\omega t} dt$$

В настоящее время предложено большое число различных окон: прямоугольное, треугольное, Ханна, Хэмминга и др. Простейшим из них является прямоугольное, соответствующее случаю умножения сигнала на константу.

**Ортогональное дополнение и прямая (direct) сумма.** Эти термины надо знать для понимания сущности КМА. Пусть  $V_0$  - подпространство в  $V_1$ . Ортогональное дополнение  $W_0$  содержит все векторы из  $V_1$ , ортогональные каждому вектору из  $V_0$ . Тогда каждый вектор в  $V_1$  может быть представлен в виде суммы двух проекций - на подпространства  $V_0$  и  $W_0$ . Прямая сумма вычислима в случае, если объединение базисов  $V_0$  и  $W_0$  дает базис

пространства  $V_1$  и не подразумевает ортогональности этих подпространств. Прямая сумма обозначается знаком  $\oplus$ .

## II

**Перераспределение энергии** - одним из предназначений линейного преобразования изображения при его сжатии является перераспределение энергии. Оно заключается в том, что большая часть энергии изображения после преобразования оказывается сосредоточенной в малой части коэффициентов (низкочастотных). Поэтому многие коэффициенты можно «обнулить», не передавать приемнику и, тем самым, достичь сжатия. Кроме того известно, что квантование неравномерно распределенной случайной величины более эффективно, чем равномерно распределенной.

**Преобразование Карунена-Лоэва (ПКЛ)** – базисные функции есть собственные векторы ковариационной матрицы входного сигнала. Является оптимальным по критерию достижения декорреляции входного сигнала. Энергия входного сигнала максимально перераспределяется в коэффициентах ПКЛ. Гарантируется, что процентное содержание энергии входного сигнала в данном количестве наибольших коэффициентов будет не меньше, чем в том же числе коэффициентов любого другого преобразования. Вычислительно трудоемко, кроме того, для вычисления обратного преобразования декодеру надо передавать базисные функции, найденные кодером. Поэтому, на практике не применяется. Все остальные преобразования часто сравниваются с ПКЛ.

**Преобразование Фурье – (Fourier transform):**

- непрерывное:  $\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt$  и  $f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(\omega)e^{-i\omega t} d\omega$ ;
- дискретное:  $\hat{x}(k) = \sum_{l=0}^{N-1} \bar{W}^{kl} x(l)$  и  $x(l) = \sum_{k=0}^{N-1} W^{kl} \hat{x}(k)$ , где  $W = e^{2\pi i/N}$ ;
- ряды Фурье:  $c(n) = \frac{1}{2\pi} \int_0^{2\pi} f(t)e^{-int} dt$  и  $f(t) = \sum_{-\infty}^{\infty} c(n)e^{int}$ .

**Преобразование Хартли (Hartley)** – одно из альтернатив ДПФ, использует вещественные функции. Элементы матрицы ПХ имеют вид:

$$H_{kn} = \cos \frac{2\pi kn}{L} + \sin \frac{2\pi kn}{L}$$

**Преобразования сигналов (Signal Transform)** – понятие, вообще говоря, очень широкое и включает в себя любую операцию, производимую над сигналом. В рамках словаря под преобразованием мы будем понимать лишь обобщенное преобразование Фурье (см. Преобразование Фурье). Преобразование переводит сигнал из одной области представления в другую. Прямое преобразование  $F = Tf$  переводит сигнал из временной (пространственной) области в область спектра, которая еще называется



трансформантой. Обратное преобразование  $f = T^{-1}F$  переводит сигнал из области трансформанты во временную (пространственную) область. Преобразования сигналов используются для разных целей: его сжатия, анализа и т.д. Анализ заключается в выполнении каких-нибудь действий над сигналом и формулировке выводов на основе полученных данных.

**Принцип неопределенности (Uncertainty Principle) Гейзенберга** – в рамках рассматриваемой тематики означает, что невозможно одновременно произвольно точно зафиксировать частоту сигнала и время ее возникновения. То и другое фиксируется с ошибкой, то есть истинное значение параметров находится внутри некоторого окна. Если считать это окно прямоугольным, то его площадь будет равна произведению частоты и времени. Площадь окна величина постоянная. Именно поэтому улучшению разрешения по частоте соответствует ухудшение разрешения по времени и наоборот.

## С

**Свертка (Convolution).** Сверткой двух векторов  $h * x$  называется вектор,  $n$ -ый элемент которого равен  $\sum h(k)x(n-k) = \sum h(n-k)x(k)$ .

**Сжатие без потерь (lossless compression)** – восстановленное после сжатия изображение полностью идентично исходному («бит в бит»). То есть при сжатии используется только статистическая избыточность, имеющаяся в изображении. Ее используют так называемые энтропийные кодеры (Хаффмана, арифметический). Понятно, что таким образом большого коэффициента сжатия достичь нельзя (обычно, коэффициент сжатия не более 1,5-2). Сжатие без потерь находит ограниченное применение в некоторых областях, например, медицине или астрономии. Алгоритмы сжатия JPEG, JPEG2000 имеют режимы работы без потерь. Иногда рассматривается сжатие «почти без потерь», то есть с незаметными для человека потерями. При этом достигаются коэффициенты сжатия порядка (6..10). Для сжатия изображений без потерь находит применение целочисленное вейвлет-преобразование. Его преимуществом является то, что коэффициенты преобразования - целые числа, поэтому после обратного преобразования можно восстановить изображение «бит в бит». Целочисленное вейвлет-преобразование выполняется для перераспределения энергии исходного изображения, так как в этом случае можно построить более эффективные энтропийные кодеры.

**Сжатие изображений (Image compression)** – сокращение объема их цифрового представления. Под изображением понимается прямоугольная матрица, элементы которой обычно принимают значения  $(0..2^8-1)$  (полутоновые изображения),  $(0..2^{24}-1)$  (цветные изображения),  $(0,1)$  (бинарные изображения). Цветные изображения могут быть в RGB-формате (по 8 бит на канал), YUV-формате (яркостная и две цветоразностные

составляющие), а также и в других форматах. Как правило, большинство алгоритмов сжатия разрабатываются для полутоновых изображений (256 градаций серого). Это связано с тем, что такие алгоритмы легко обобщить на случай цветного изображения. Различают сжатие без потерь и с потерями.

**Сжатие с потерями (Lossy Compression)** - восстановленное изображение визуально мало отличается от исходного. Это различие называется искажением. Говорят, что сжатие с потерями вносит искажение. Мерой искажения может быть визуальное отличие двух изображений. Однако, эта характеристика плохо описывается на языке формул. Поэтому, на практике в качестве меры искажения используется среднеквадратическая ошибка. В среде исследователей не прекращаются споры об адекватности этой меры, однако ничего лучшего по сей день не придумано. Сжатие с потерями выполняется, как правило, в три этапа: линейное преобразование (дискретное косинусное преобразование, вейвлет), квантование, энтропийное кодирование. Целей первого этапа несколько: перераспределение энергии между отсчетами, декорреляция отсчетов и придание коэффициентам удобной для квантования структуры (например, нульдеревя). Квантование бывает скалярное (каждый отсчет в отдельности) и векторное. В качестве энтропийного кодера обычно используется арифметический (более эффективный, но и более вычислительно сложный) или кодер Хаффмана.

**Сигнал (Signal)** – то, что математики обычно называют функцией. Под сигналом можно понимать какое-то упорядоченное множество чисел, несущих информацию о некотором процессе. Обычно описывается двумя (одномерный сигнал), тремя (двумерный сигнал) или более (многомерный сигнал) параметрами. Представляется в виде конечной или бесконечной совокупности точек. Одним из параметров для всех типов сигналов является значение уровня сигнала (его энергии) во всех точках. В качестве других параметров обычно выступают время (одномерный сигнал), пространственные координаты (двумерный и многомерные сигналы). Значения всех параметров могут быть непрерывными или дискретными. Таким образом, для каждой размерности можно различать  $2^L$  сигналов, где  $L$  - размерность сигнала. Например, в одномерном случае существуют сигналы, непрерывные по времени и уровню, дискретные по времени и уровню, дискретные по времени и непрерывные по уровню, непрерывные по времени и дискретные по уровню. Преобразование непрерывного по времени и уровню сигнала в дискретный по времени и уровню сигнал называется аналого-цифровым преобразованием (АЦП), обратное преобразование - ЦАП.

**Симметрия (антисимметрия) базисных функций** - симметрия означает, что если перенести центр функции в начало координат, то функция будет четной. Антисимметрия означает, что функция после переноса будет нечетной.

**Система анализа-синтеза (А-С)** – совокупность банка фильтров анализа и банка фильтров синтеза. Сигнал вначале декомпозируется на субполосы, в каждой из них выполняется некоторая его обработка и, затем, выполняется синтез сигнала (реконструкция). Надо отметить, что иногда нужен обратный порядок операций: сначала синтез, потом анализ. Такая последовательность действий встречается при использовании банков фильтров в связи. При этом НЧ сигналы от разных источников интерполируются, фильтруются, объединяются и передаются по каналу связи. На приеме групповой сигнал подается на схему анализа для выделения сигналов отдельных абонентов.

**Спектр (Spectr)** – представление сигнала в виде конечной или бесконечной суммы некоторых элементарных сигналов, умноженных на некоторые числа. В качестве элементарных сигналов обычно выступают ортогональные функции, такие как Фурье, Уолша, Хаара, Адамара, Хартли, вейвлеты и т.д. Поэтому говорят о спектре Фурье, Уолша и т.д. Числа, на которые умножаются элементарные сигналы называются спектральными коэффициентами (коэффициентами трансформанты). Часто просто говорят о коэффициентах Фурье, Уолша и т.д.

**Стандарты на сжатие видео** – MPEG-2, MPEG-4 (дискретное косинусное преобразование и вейвлеты, соответственно), рекомендация H.263, H263+ и др. Сжатие видео состоит из двух частей: сжатие неподвижного кадра (во многом аналогично сжатию изображений) и межкадровое кодирование.

**Стандарты на сжатие изображений** – сжатие бинарных изображений - JBIG, сжатие полутоновых и цветных - JPEG, JPEG2000. В стандарте JPEG в качестве линейного преобразования используется дискретное косинусное преобразование, в JPEG2000 - вейвлет-преобразование. На роль стандарта де-факто для сжатия смешанных тексто-графических изображений претендует алгоритм DjVu («дежавю»), основанный на распознавании образов и последующем применении различных алгоритмов к различным типам изображений.

**Стеганографические методы.** Информация может встраиваться в области исходного сигнала (временной, пространственной) или в области преобразования. Для применения в стеганографии предлагались, наверное, все известные линейные преобразования сигналов. Наиболее часто предлагается использовать преобразование Фурье, дискретное косинусное преобразование, преобразование Хаара, вейвлет-преобразование. Некоторые авторы рекомендуют использовать для встраивания информации те же преобразования, что и при сжатии. Другие авторы, наоборот, указывают, что использование стегоалгоритмами и алгоритмами сжатия одних и тех же

“дыр” неэффективно. Методы встраивания можно разделить на линейные и нелинейные.

**Стеганография** – наука о скрытии самого факта наличия информации. В этом ее отличие от криптографии, которая занимается скрытием содержания информации. В связи с появлением компьютеров в последние годы получила определенное развитие компьютерная стеганография. Особенно активно ведутся исследования в области цифровой стеганографии, или встраивания одних данных в другие с применением методов цифровой обработки сигналов.

## У

**Унитарное преобразование (Unitary transform)** – преобразование, сохраняющее норму сигнала. Вещественное унитарное преобразование называется ортогональным. Его базисные функции ортогональны между собой. Важным свойством ортогонального преобразования является простая формула вычисления обратного преобразования.

## Ф

**Фильтр (Filter)** – линейная стационарная система, то есть свойства фильтра не зависят от времени. Независимость свойств фильтра от времени означает, что задержка входа приводит к такой же задержке выхода. К основным характеристикам фильтров относятся импульсная характеристика  $h(n)$ , передаточная характеристика  $H(z) = \sum h(n)z^{-n}$ , частотная характеристика  $H(e^{i\omega}) = \sum h(n)e^{-i\omega n}$ , порядок фильтра.

**Фильтр Баттерворта** (максимально плоский фильтр) – в полосе пропускания практически нет колебаний. Это является преимуществом для многих приложений, где требуется постоянство коэффициента ослабления фильтра для всех частот полосы пропускания. Недостатком такого фильтра является невысокая крутизна полосы перехода.

## Ц

**Целочисленное (Integer) вейвлет-преобразование** - применяется в двух смыслах: 1) использование целочисленных фильтров; 2) получение целых значений коэффициентов преобразования целочисленного сигнала. В первом случае повышается скорость вычислений, во втором случае появляется возможность выполнения преобразования без потерь.

Э

**Элайзинг (Aliasing)** – наложение спектра, то есть два сигнала накладываются один на другой.

## ПРИЛОЖЕНИЕ 2. ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ В СРЕДАХ MATLAB И MATHCAD

### П.2.1 Пример фильтрации сигнала с использованием быстрого прямого и обратного преобразований Фурье

В основе спектрального анализа лежит теория Фурье о возможности разложения любого периодического процесса с периодом  $T = 2\pi/\omega = 1/f$  (где  $\omega$  – круговая частота периодического процесса, а  $f$  – его частота в герцах) в бесконечную, но счетную сумму отдельных гармонических составляющих.

Любой процесс с периодом  $T$  может быть представлен в виде комплексного ряда Фурье. Совокупность комплексных амплитуд ряда может рассматриваться как изображение периодического процесса в частотной области. Наибольшее распространение получила следующая форма записи преобразования Фурье:

$$X(f) = \int_{-\infty}^{+\infty} (x(t) \cdot e^{-i(2\pi f)t}) dt \quad (1)$$

Данный интеграл (1) существует (является сходящимся к конечной величине) только для так называемых «двусторонне затухающих» процессов (т. е. таких, которые уменьшаются до нуля как при  $t \rightarrow +\infty$ , так и при  $t \rightarrow -\infty$ ). Иначе говоря, его нельзя применять к так называемым «стационарным» колебаниям.

Указанное противоречие, касающееся условий существования интеграла (1) несколько сглаживается при численных расчетах, так как в этом случае можно иметь дело только с процессами ограниченной длительности, причем сам процесс в заданном диапазоне времени должен быть задан своими значениями в ограниченном числе точек.

В этом случае интегрирование заменяется суммированием и вместо вычисления интеграла (1) расчеты ограничиваются вычислением суммы:

$$X[(k-1) \cdot \Delta f] = \sum_{m=1}^n x[(m-1) \cdot \Delta t] \cdot e^{-i(2\pi)[k-1][m-1]\Delta f \cdot \Delta t} \quad (2)$$

Где:  $\Delta t$  – шаг времени,  $m$  – номер точки от начала процесса ( $m = 0, 1, 2, \dots$ ),  $k$  – номер значения частоты ( $k = 0, 1, 2, \dots$ ),  $\Delta f = 1/T$  – шаг частоты, Гц,  $T$  – промежуток времени на котором задан процесс,  $i$  – мнимая единица.

Если обозначить  $\Delta t$  через  $Ts$  и ввести обозначения:

$$x(m) = x[(m-1) \cdot \Delta t],$$

$$X(k) = X[(k-1) \cdot \Delta f],$$

а также учесть то, что число точек  $n$ , в которых задан процесс, равно:

$$n = \frac{T}{\Delta t} = \frac{T}{T_s} = \frac{1}{(\Delta f \cdot \Delta t)} \quad (3)$$

то соотношение (2) можно представить в более удобной форме:

$$X(k) = T_s \sum_{m=1}^n x(m) \cdot e^{-i(2\pi/n)[k-1][m-1]}. \quad (4)$$

Данное выражение носит название дискретного преобразования Фурье (ДПФ). Формально ДПФ периодической последовательности и ДПФ конечной последовательности полностью совпадают и разница между ними заключается в трактовке результатов, а именно:

В среде MATLAB прямое (5) и обратное (6) преобразования Фурье осуществляются с помощью функций `fft` и `ifft` соответственно. Вычисления осуществляются в соответствии с формулами:

$$y(k) = \sum_{m=1}^n x(m) \cdot e^{-i(2\pi)[k-1][m-1]/n}, \quad (5)$$

$$x(m) = \frac{1}{n} \sum_{k=1}^n y(k) \cdot e^{i(2\pi)[k-1][m-1]/n}, \quad (6)$$

Для применения процедуры `fft` необходимо сформировать частоты следующим образом:  $f = 0 : \Delta f : Fmax$ ,  $Fmax = 1/T_s$ ,  $\Delta f = 1/T$ . В результате применения процедуры `fft` будет получено представление процесса в частотной области. Обратная процедура `ifft`, если ее применить к результатам первого преобразования, дает возможность восстановить исходный процесс во временной области.

Однако процедура `fft` не дает непосредственно *Фурье-изображения* сигнала (процесса). Чтобы получить Фурье-изображение, необходимо к полученному результату применить процедуру `fftshift` и перестроить вектор частот по алгоритму:

$$f = -Fmax/2 : 1/T : Fmax/2.$$

Комплексный *спектр* разложения стационарного сигнала (процесса) равен поделенному на число измерений результату применения процедуры `fft` к заданному вектору измеренного сигнала (процесса).

В документации MATLAB описано несколько вариантов вызова функции `fft`:

- 1) `fft(x)` выполняет дискретное преобразование Фурье, вычисляя сумму (5) с помощью алгоритма быстрого преобразования Фурье;
- 2) `fft(x, n)` выполняет дискретное преобразование Фурье, используя  $n$

отсчётов из вектора  $x$ . Если вектор  $x$  состоит из меньшего числа отсчётов, то в конец сигнала добавляется необходимое количество нулевых отсчётов;

- 3) `fft(x, [], dim)` выполняет дискретное преобразование Фурье многомерного массива  $x$ , параметр  $dim$  указывает номер индекса массива по которому выполняется преобразование. Такой вариант вызова функции может быть полезен, когда в одной переменной хранится сразу несколько дискретных сигналов. По умолчанию в MATLAB считается, что сигналы находятся в столбцах матрицы, то есть первый индекс двумерного массива чисел – это номер отсчёта, второй индекс – номер сигнала. Если задать параметр  $dim$  равным 2, то второй индекс массива  $x$  будет означать номер отсчёта, а первый – номер сигнала;
- 4) `fft(x, n, dim)` этот вариант объединяет в себе возможности двух предыдущих.

Редактируя Фурье-изображение дискретного сигнала (например, умножая значения коэффициентов Фурье-разложения некоторого диапазона частот образа на число) и, затем, выполняя обратное преобразование Фурье, получаем изменение исходного сигнала.

Одномерное обратное преобразование Фурье реализуется функцией `ifft`:

- 1) `ifft(F)` – возвращает результат дискретного обратного преобразования Фурье вектора  $F$ . Если  $F$  – матрица, то `ifft` возвращает обратное преобразование Фурье для каждого столбца этой матрицы;
- 2) `ifft(F, n)` – возвращает результат  $n$ -точечного дискретного обратного преобразования Фурье вектора  $F$ ;
- 3) `ifft(F, [], dim)` и `y = ifft(F, n, dim)` – возвращает результат обратного дискретного преобразования Фурье массива  $F$  по строкам или по столбцам в зависимости от значения скаляра  $dim$ , где  $dim$  – целое положительное число.

Далее приведены примеры спектрального анализа и фильтрации сигналов в среде MATLAB.

### Пример спектрального анализа одиночного прямоугольного импульса в среде MATLAB

```
%-----  
close all;
```



```

clear all;
%-----

%Построение графика одиночного прямоугольного импульса
Ts = 0.01;           %шаг времени, сек
T = 1;              %длительность периода моделирования импульса, сек
A = 0.75;          %амплитуда прямоугольного импульса
w = 0.5;           %ширина прямоугольного импульса
                    %(включая отрицательную полуось времени), сек
t = 0 : Ts : T;    %вектор значений времени, сек
y = A*rectpuls(t, w); %создание одиночного прямоугольного импульса в
                    %дискретной форме и сохранение его в векторе y

%создание окна с заданным заголовком для вывода графика
figure('Name','Одиночный прямоугольный импульс','NumberTitle','off');
%вызов функции построения графика с заданными параметрами начертания
plot(t, y, 'k-o', 'LineWidth', 2); hold on; grid on;
%установка параметров шрифта текста, отображаемого в окне
set(gca, 'FontName', 'Arial Cyr', 'FontSize', 10);
%задание заголовка графика
title('Одиночный прямоугольный импульс');
%задание подписи оси абсцисс
xlabel('Время t, сек');
%задание подписи оси ординат
ylabel('Y(t)');

%-----

%Выполнение FFT-преобразования прямоугольного импульса
x = fft(y);        %вызов функции быстрого преобразования Фурье (FFT)
                    %и сохранение его результатов в векторе x
df = 1 / T;        %шаг частоты, Гц
Fmax = 1 / Ts;     %максимальная анализируемая частота
f = 0 : df : Fmax; %диапазон анализируемых частот
a = abs(x);        %получение модуля результата FFT-преобразования
figure('Name', 'Модуль FFT-преобразования прямоугольного импульса', ...
'NumberTitle', 'off');
plot(f, a, 'r-', 'LineWidth', 2); hold on; grid on;
set(gca, 'FontName', 'Arial Cyr', 'FontSize', 10);
title('Модуль FFT-преобразования прямоугольного импульса');
xlabel('Частота, Гц');
ylabel('Модуль');

%-----

%Получение модуля Фурье-изображения прямоугольного импульса
xp = fftshift(x); %сдвиг результатов FFT-преобразования
                    %(смещение компоненты нулевой частоты в центр спектра)
                    %и сохранение его в векторе xp
f1 = -Fmax/2 : df : Fmax/2; %диапазон анализируемых частот
a = abs(xp);       %получение модуля Фурье-изображения
figure('Name', 'Модуль Фурье-изображения прямоугольного импульса', ...
'NumberTitle', 'off');
plot(f1, a, 'b-', 'LineWidth', 2); hold on; grid on;
set(gca, 'FontName', 'Arial Cyr', 'FontSize', 10);

```

```

title('Модуль Фурье-изображения прямоугольного импульса');
xlabel('Частота, Гц');
ylabel('Модуль');

%-----

%Получение комплексного Фурье-изображения прямоугольного импульса
dch = real(xp); %получение действительной части Фурье-изображения
mch = imag(xp); %получение мнимой части Фурье-изображения
figure('Name', 'Фурье-изображение прямоугольного импульса', ...
'NumberTitle', 'off');
plot(f1, dch, 'r-', 'LineWidth', 2); hold on; grid on;
plot(f1, mch, 'b--', 'LineWidth', 2); hold on; grid on;
set(gca, 'FontName', 'Arial Cyr', 'FontSize', 10);
%отображение легенды графика
legend('Действительная часть', 'Мнимая часть', 1);
title('Фурье-изображение прямоугольного импульса');
xlabel('Частота, Гц');

%-----

%Восстановление сигнала по результату его FFT-преобразования
y1 = ifft(x); %вызов функции обратного быстрого преобразования Фурье
%и сохранение его результатов в векторе y1
figure('Name', 'Результат обратного преобразования Фурье', ...
'NumberTitle', 'off');
plot(t, y1, 'm-o', 'LineWidth', 2); hold on; grid on;
set(gca, 'FontName', 'Arial Cyr', 'FontSize', 10);
title('Результат обратного преобразования Фурье');
xlabel('Время t, сек');
ylabel('Y(t)');

%-----

```

На рис. 1 – 5 представлены изображения окон, появляющихся на экране монитора персонального компьютера при выполнении текста программы в среде MATLAB.

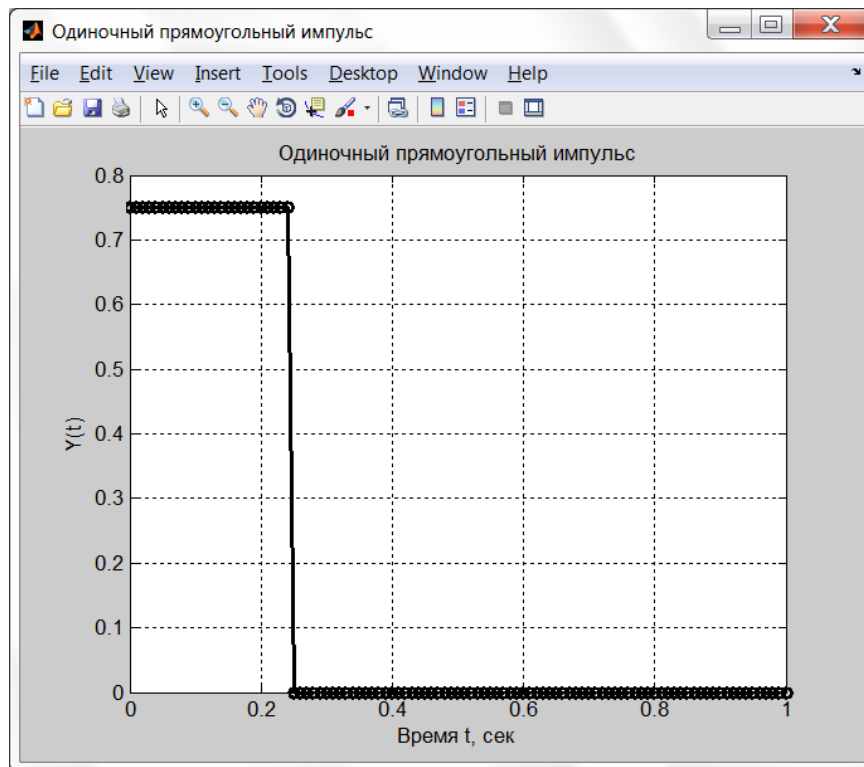


Рис. 1 Окно с графиком сигнала одиночного прямоугольного импульса

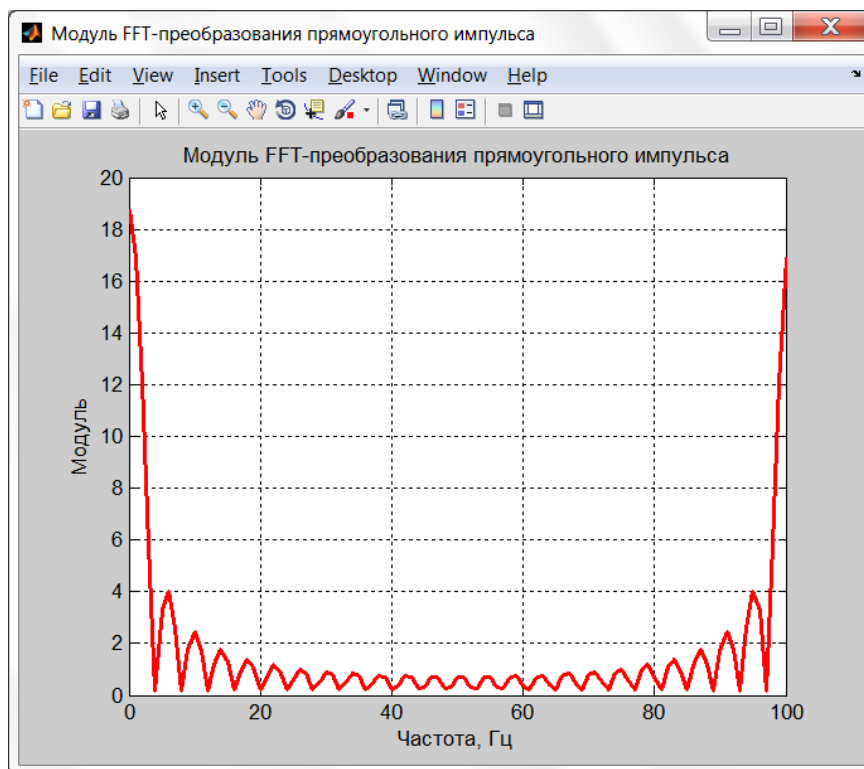


Рис. 2 Окно с графиком модуля FFT-преобразования сигнала

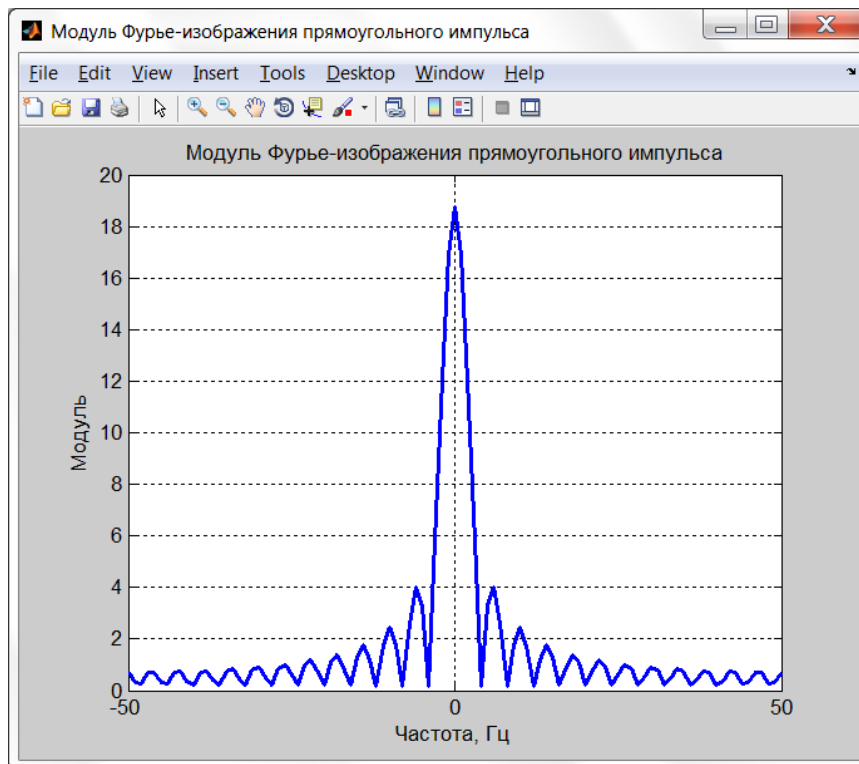


Рис. 3 Окно с графиком модуля Фурье-изображения сигнала (после перестроения вектора результата FFT-преобразования)

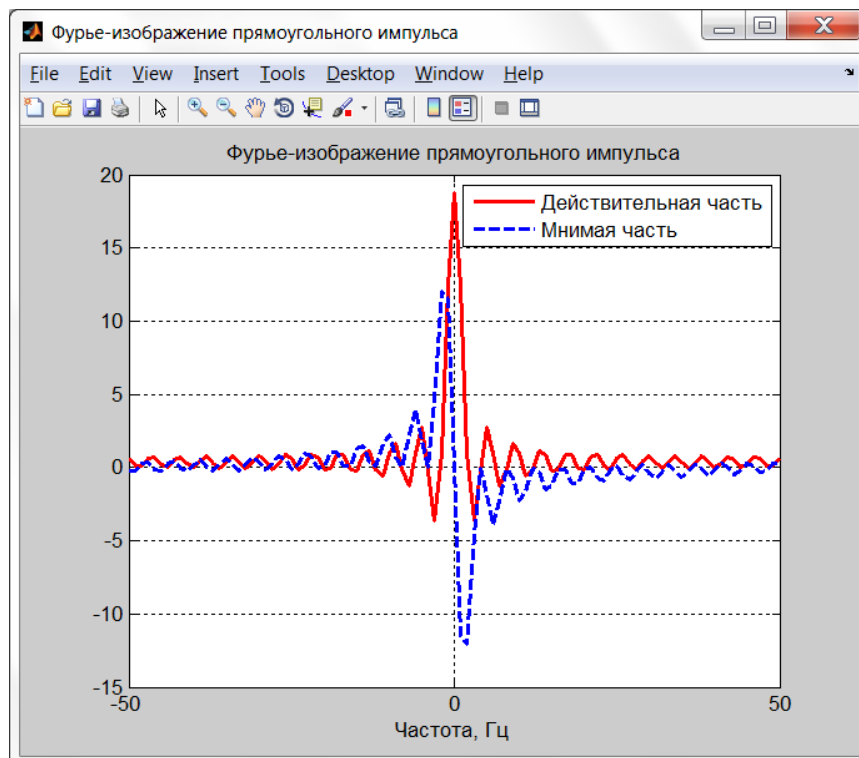


Рис. 4 Окно с графиками комплексного Фурье-изображения сигнала

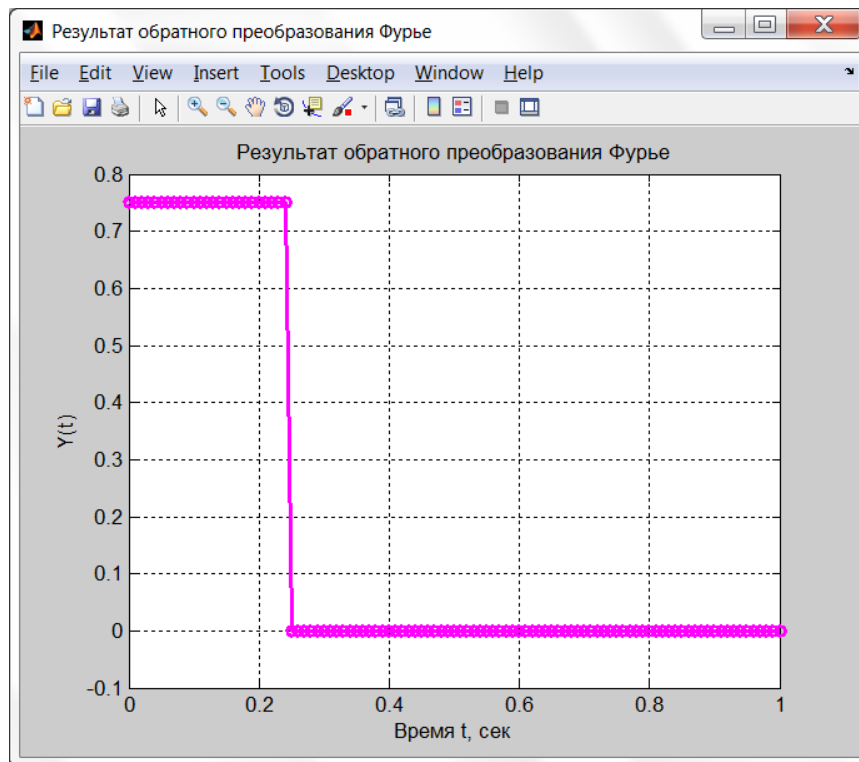


Рис. 5 Окно с графиком восстановленного сигнала после обратного быстрого преобразования Фурье

### Пример спектрального анализа и фильтрации полигармонического сигнала в среде MATLAB

```

%-----
close all;
clear all;
%-----

%Построение графика полигармонического сигнала
Ts = 0.01;           %шаг времени, сек
T = 100;            %длительность периода моделирования сигнала, сек
t = 0 : Ts : T;    %вектор значений времени, сек
w = [ 0.5  1  2 ]; %вектор частот гармоник сигнала
rng('default'); %задание настроек "по умолчанию" генератора случайных
                %чисел
                %(по умолчанию используется алгоритм "Вихрь Мерсенна")

%создание вектора-строки, содержащего случайные числа, полученные
%с помощью функции rand - диапазон значений случайных чисел: -1 ... 1
noise = (2 * rand(length(t), 1) - 1)';
%создание полигармонического сигнала в
%дискретной форме и сохранение его в векторе Y0
Y0 = cos(2*pi*t*w(1)) + 0.5*sin(2*pi*t*w(2)) + 0.3*cos(2*pi*t*w(3));
%добавление к полигармоническому сигналу шума - случайных чисел
Y = Y0 + noise;
figure('Name', 'Полигармонический сигнал', 'NumberTitle', 'off');
%построение графика полигармонического сигнала с шумом

```

```

plot(t, Y, 'r-', 'LineWidth', 1); hold on; grid on;
%построение графика полигармонического сигнала без шума
plot(t, Y0, 'k:', 'LineWidth', 3); hold on; grid on;
set(gca, 'FontName', 'Arial Cyr', 'FontSize', 10);
legend('Полигармонический сигнал с шумом', ...
'Полигармонический сигнал без шума', 1);
%функция ограничения диапазона значений времени,
%указанных на оси абсцисс графика
xlim([ 0 (Ts*1000) ]);
xlabel('Время, сек');
ylabel('Y(t)');
title('Полигармонический сигнал');

%-----

%Получение модуля Фурье-изображения полигармонического сигнала
df = 1 / T; %шаг частоты, Гц
Fmax = 1 / Ts; %максимальная анализируемая частота
f = -Fmax/2 : df : Fmax/2; %диапазон анализируемых частот
X = fft(Y); %вызов функции быстрого преобразования Фурье (FFT)
Xp = fftshift(X); %сдвиг результатов FFT-преобразования
% (смещение компоненты нулевой частоты в центр
% спектра)
A = abs(Xp); %получение модуля Фурье-изображения
fmin = -10; %минимальное отображаемое значение частоты, Гц
fmax = 10; %максимальное отображаемое значение частоты, Гц
figure('Name', 'Модуль Фурье-изображения сигнала', 'NumberTitle', 'off');
plot(f, A, 'b:d', 'LineWidth', 2); hold on; grid on;
set(gca, 'FontName', 'Arial Cyr', 'FontSize', 10);
%функция ограничения диапазона значений частоты,
%указанных на оси абсцисс графика
xlim([ fmin fmax ]);
xlabel('Частота, Гц');
ylabel('Модуль');
title('Модуль Фурье-изображения сигнала');

%-----

%Фильтрация Фурье-изображения сигнала
limit = 0.1; %задание максимальной величины окрестности ("допуска")
% значений частот, Гц
step = 0.01; %шаг частоты внутри окрестности, Гц

wlimit = 0 : step : limit; %задание окрестности значений частот (Гц)
%гармоник сигнала без шума

%формирование значений частот гармоник сигнала без шума
ww = [ w(1)-flip(wlimit) w(1)+wlimit(2:end) ...
w(2)-flip(wlimit) w(2)+wlimit(2:end) ...
w(3)-flip(wlimit) w(3)+wlimit(2:end) ];

%выполнение фильтрации
%полигармонического сигнала от частот,
%не принадлежащих гармоникам сигнала без шума
for (p = 1 : length(f))

```

```

    if (abs(f(p)) ~= ww(:))
        Xp(p) = 0;
    end;
end;

figure('Name', 'Модуль Фурье-изображения сигнала после фильтрации', ...
'NumberTitle', 'off');
plot(f, abs(Xp), 'r-o', 'LineWidth', 2); hold on; grid on;
set(gca, 'FontName', 'Arial Cyr', 'FontSize', 10);
title('Модуль Фурье-изображения сигнала после фильтрации');
xlim([ fmin fmax ]);
xlabel('Частота, Гц');
ylabel('Модуль');

%-----

%Получение комплексного Фурье-изображения полигармонического сигнала
dch = real(Xp); %получение действительной части Фурье-изображения
mch = imag(Xp); %получение мнимой части Фурье-изображения
figure('Name', 'Комплексное Фурье-изображение сигнала', ...
'NumberTitle', 'off');

subplot(2,1,1); %функция деления области построения окна на 2 части -
               %графики располагаются столбцом: первый график строится
               %в верхней половине окна
plot(f, dch, 'r-', 'LineWidth', 2); hold on; grid on;
set(gca, 'FontName', 'Arial Cyr', 'FontSize', 10);
xlim([ fmin fmax ]);
ylabel('Действит. часть');
title('Комплексное Фурье-изображение сигнала');

subplot(2,1,2); %функция деления области построения окна на 2 части -
               %графики располагаются столбцом: второй график строится
               %в нижней половине окна
plot(f, mch, 'b--', 'LineWidth', 2); hold on; grid on;
set(gca, 'FontName', 'Arial Cyr', 'FontSize', 10);
xlim([ fmin fmax ]);
xlabel('Частота, Гц');
ylabel('Мнимая часть');

%-----

%Восстановление сигнала по результату его FFT-преобразования
%обратный сдвиг Фурье-изображения сигнала
%(обратное смещение компоненты нулевой частоты)
X = ifftshift(Xp);

y1 = ifft(X); %вызов функции обратного быстрого преобразования Фурье
              %и сохранение его результатов в векторе y1
figure('Name', 'Результат обратного преобразования Фурье', ...
'NumberTitle', 'off');
plot(t, y1, 'k-', 'LineWidth', 2); hold on; grid on;
set(gca, 'FontName', 'Arial Cyr', 'FontSize', 10);
title('Результат обратного преобразования Фурье');
xlim([ 0 (Ts*1000) ]);

```

```

xlabel('Время t, сек');
ylabel('Y(t)');

%-----

```

На рис. 6 – 10 представлены изображения окон, появляющихся на экране монитора персонального компьютера при выполнении текста программы в среде MATLAB.

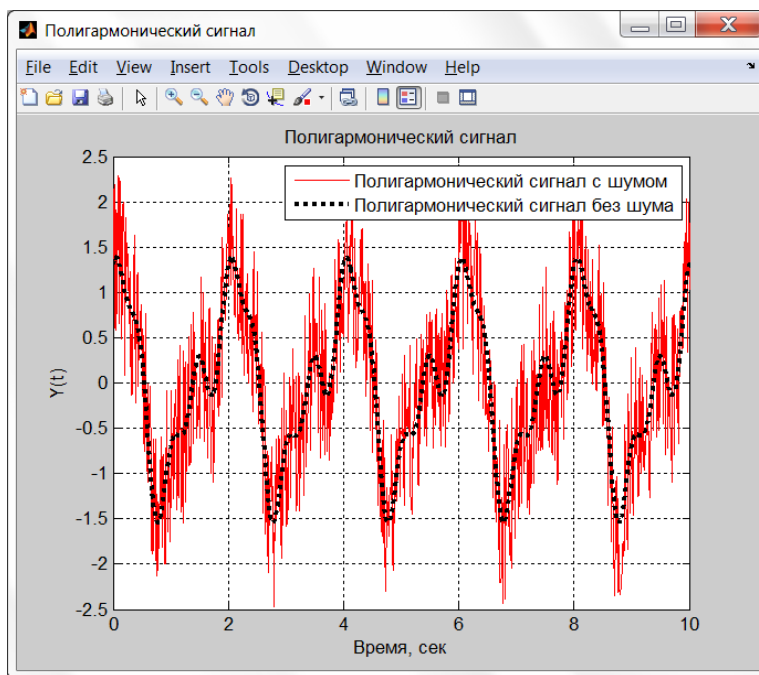


Рис. 6 Окно с графиками полигармонического сигнала с наложенным шумом и без него

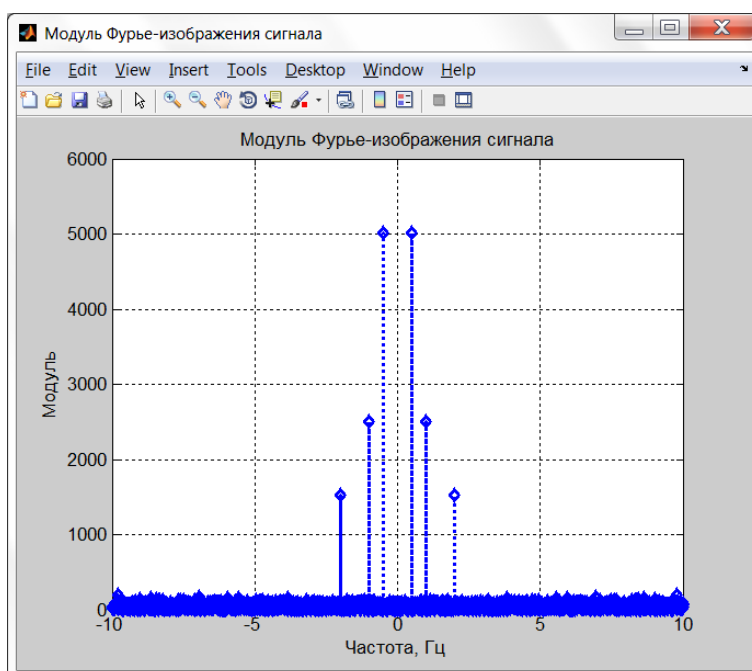




Рис. 7 Окно с графиком модуля Фурье-изображения сигнала

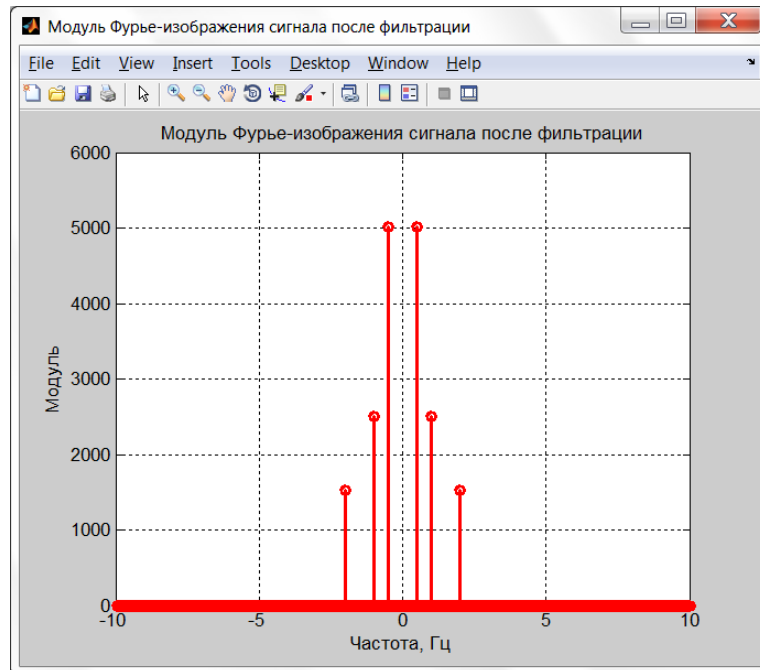


Рис. 8 Окно с графиком модуля Фурье-изображения сигнала после процедуры фильтрации

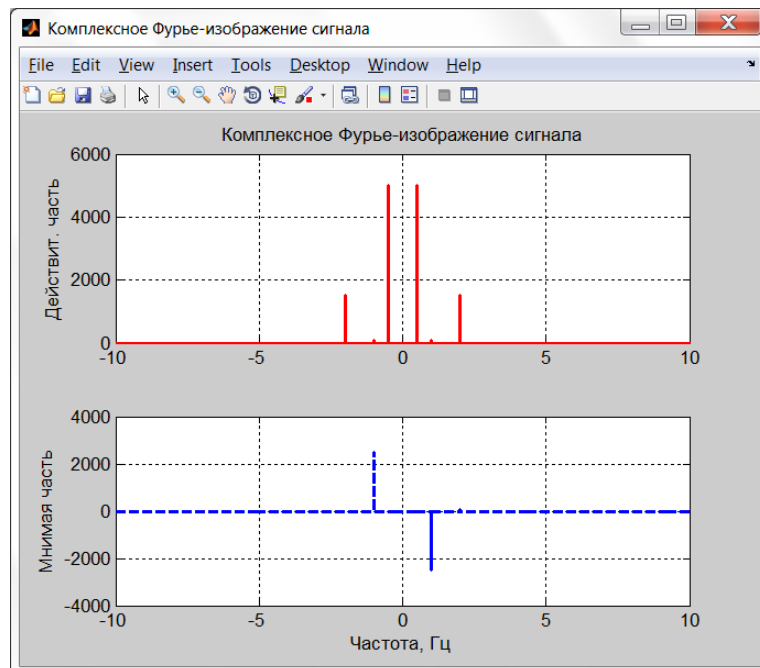


Рис. 9 Окно с графиками комплексного Фурье-изображения сигнала после процедуры фильтрации

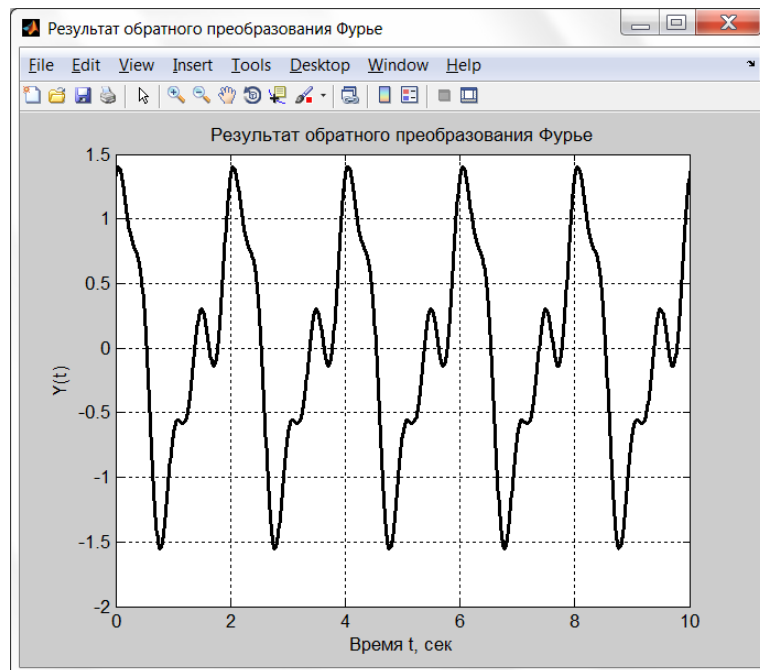


Рис. 10 Окно с графиком восстановленного сигнала после обратного быстрого преобразования Фурье

### Задания

1. По заданию преподавателя, пользуясь данными табл. 1, выбрать полигармонический сигнал и ввести его в среду MATLAB, с параметрами дискретизации:
 

$T_s = 0.001;$	<i>%шаг времени, сек</i>
$T = 100;$	<i>%длительность периода моделирования, сек</i>
$t = 0 : T_s : T;$	<i>%вектор значений времени, сек</i>
2. Построить временную диаграмму полигармонического сигнала средствами MATLAB.
3. Используя функции `fft` и `fftshift` получить модуль Фурье-изображения и комплексное Фурье-изображение полигармонического сигнала, построить их графики средствами MATLAB.
4. Согласно варианту задания выполнить фильтрацию исходного полигармонического сигнала.
5. Получить модуль Фурье-изображения сигнала после фильтрации, построить его график.
6. Используя функции `ifftshift` и `ifft` выполнить восстановление полученного сигнала по результату его FFT-преобразования.
7. Построить в среде MATLAB временную диаграмму сигнала после процедуры фильтрации.

## Варианты заданий

Таблица 1 – Варианты заданий

Вариант	Задание
1	<p>Задан полигармонический сигнал:  <math display="block">Y = A_1 \cos(2\pi t w(1)) + A_2 \sin(2\pi t w(2)) + \dots</math> <math display="block">A_3 \cos(2\pi t w(3)) + A_4 \sin(2\pi t w(4));</math>                     где:  <math>A_1 = 10, A_2 = 7, A_3 = 5, A_4 = 3,</math>  <math>w(1) = 17 \text{ Гц}, w(2) = 21 \text{ Гц}, w(3) = 70 \text{ Гц}, w(4) = 140 \text{ Гц}.</math>                      Отфильтровать сигнал от частот выше 20 Гц</p>
2	<p>Задан полигармонический сигнал:  <math display="block">Y = A_1 \cos(2\pi t w(1)) + A_2 \sin(2\pi t w(2)) + \dots</math> <math display="block">A_3 \cos(2\pi t w(3)) + A_4 \sin(2\pi t w(4));</math>                     где:  <math>A_1 = 10, A_2 = 3, A_3 = 10, A_4 = 1,</math>  <math>w(1) = 1 \text{ Гц}, w(2) = 5 \text{ Гц}, w(3) = 60 \text{ Гц}, w(4) = 100 \text{ Гц}.</math>                      Отфильтровать сигнал от частот ниже 60 Гц</p>
3	<p>Задан полигармонический сигнал:  <math display="block">Y = A_1 \cos(2\pi t w(1)) + A_2 \sin(2\pi t w(2)) + \dots</math> <math display="block">A_3 \cos(2\pi t w(3)) + A_4 \sin(2\pi t w(4));</math>                     где:  <math>A_1 = 10, A_2 = 7, A_3 = 5, A_4 = 3,</math>  <math>w(1) = 0,5 \text{ Гц}, w(2) = 5 \text{ Гц}, w(3) = 6 \text{ Гц}, w(4) = 10 \text{ Гц}.</math>                      Отфильтровать сигнал от частот от 4 до 6 Гц</p>
4	<p>Задан полигармонический сигнал:  <math display="block">Y = A_1 \cos(2\pi t w(1)) + A_2 \sin(2\pi t w(2)) + \dots</math> <math display="block">A_3 \cos(2\pi t w(3)) + A_4 \sin(2\pi t w(4));</math>                     где:  <math>A_1 = 10, A_2 = 7, A_3 = 20, A_4 = 1,</math>  <math>w(1) = 0,5 \text{ Гц}, w(2) = 9 \text{ Гц}, w(3) = 10 \text{ Гц}, w(4) = 20 \text{ Гц}.</math>                      Отфильтровать сигнал от частот выше 9 Гц</p>
5	<p>Задан полигармонический сигнал:  <math display="block">Y = A_1 \cos(2\pi t w(1)) + A_2 \sin(2\pi t w(2)) + \dots</math> <math display="block">A_3 \cos(2\pi t w(3)) + A_4 \sin(2\pi t w(4));</math>                     где:  <math>A_1 = 5, A_2 = 7, A_3 = 20, A_4 = 10,</math>  <math>w(1) = 0,5 \text{ Гц}, w(2) = 10 \text{ Гц}, w(3) = 11 \text{ Гц}, w(4) = 20 \text{ Гц}.</math>                      Выделить сигнал с частотами от 9 до 11 Гц</p>
6	<p>Задан полигармонический сигнал:  <math display="block">Y = A_1 \cos(2\pi t w(1)) + A_2 \sin(2\pi t w(2)) + \dots</math> <math display="block">A_3 \cos(2\pi t w(3)) + A_4 \sin(2\pi t w(4));</math>                     где:  <math>A_1 = 50, A_2 = 10, A_3 = 20, A_4 = 1,</math>  <math>w(1) = 0,5 \text{ Гц}, w(2) = 3 \text{ Гц}, w(3) = 10 \text{ Гц}, w(4) = 20 \text{ Гц}.</math>                      Отфильтровать сигнал от частот ниже 3 Гц</p>

Вариант	Задание
7	<p>Задан полигармонический сигнал:  <math display="block">Y = A1 \cdot \cos(2 \cdot \pi \cdot t \cdot w(1)) + A2 \cdot \sin(2 \cdot \pi \cdot t \cdot w(2)) + \dots</math> <math display="block">A3 \cdot \cos(2 \cdot \pi \cdot t \cdot w(3)) + A4 \cdot \sin(2 \cdot \pi \cdot t \cdot w(4));</math> где:  <math>A1 = 10, A2 = 5, A3 = 2, A4 = 1,</math>  <math>w(1) = 0,4 \text{ Гц}, w(2) = 0,8 \text{ Гц}, w(3) = 10 \text{ Гц}, w(4) = 20 \text{ Гц}.</math>  Отфильтровать сигнал от частот выше 0,8 Гц</p>
8	<p>Задан полигармонический сигнал:  <math display="block">Y = A1 \cdot \cos(2 \cdot \pi \cdot t \cdot w(1)) + A2 \cdot \sin(2 \cdot \pi \cdot t \cdot w(2)) + \dots</math> <math display="block">A3 \cdot \cos(2 \cdot \pi \cdot t \cdot w(3)) + A4 \cdot \sin(2 \cdot \pi \cdot t \cdot w(4));</math> где:  <math>A1 = 20, A2 = 50, A3 = 100, A4 = 10,</math>  <math>w(1) = 1 \text{ Гц}, w(2) = 10 \text{ Гц}, w(3) = 50 \text{ Гц}, w(4) = 100 \text{ Гц}.</math>  Отфильтровать сигнал от частот от 9 до 11 Гц и от частот выше 50 Гц</p>
9	<p>Задан полигармонический сигнал:  <math display="block">Y = A1 \cdot \cos(2 \cdot \pi \cdot t \cdot w(1)) + A2 \cdot \sin(2 \cdot \pi \cdot t \cdot w(2)) + \dots</math> <math display="block">A3 \cdot \cos(2 \cdot \pi \cdot t \cdot w(3)) + A4 \cdot \sin(2 \cdot \pi \cdot t \cdot w(4));</math> где:  <math>A1 = 1, A2 = 5, A3 = 20, A4 = 10,</math>  <math>w(1) = 0,5 \text{ Гц}, w(2) = 1 \text{ Гц}, w(3) = 10 \text{ Гц}, w(4) = 12 \text{ Гц}.</math>  Выделить сигнал с частотами от 9 Гц до 10 Гц</p>
10	<p>Задан полигармонический сигнал:  <math display="block">Y = A1 \cdot \cos(2 \cdot \pi \cdot t \cdot w(1)) + A2 \cdot \sin(2 \cdot \pi \cdot t \cdot w(2)) + \dots</math> <math display="block">A3 \cdot \cos(2 \cdot \pi \cdot t \cdot w(3)) + A4 \cdot \sin(2 \cdot \pi \cdot t \cdot w(4));</math> где:  <math>A1 = 8, A2 = 4, A3 = 2, A4 = 1,</math>  <math>w(1) = 0,5 \text{ Гц}, w(2) = 2 \text{ Гц}, w(3) = 4 \text{ Гц}, w(4) = 10 \text{ Гц}.</math>  Выделить сигнал с частотами от 2 до 4 Гц</p>
11	<p>Задан полигармонический сигнал:  <math display="block">Y = A1 \cdot \cos(2 \cdot \pi \cdot t \cdot w(1)) + A2 \cdot \sin(2 \cdot \pi \cdot t \cdot w(2)) + \dots</math> <math display="block">A3 \cdot \cos(2 \cdot \pi \cdot t \cdot w(3));</math> где:  <math>A1 = 10, A2 = 5, A3 = 5,</math>  <math>w(1) = 1 \text{ Гц}, w(2) = 5 \text{ Гц}, w(3) = 50 \text{ Гц}.</math>  Отфильтровать сигнал от частот ниже 50 Гц</p>
12	<p>Задан полигармонический сигнал:  <math display="block">Y = A1 \cdot \cos(2 \cdot \pi \cdot t \cdot w(1)) + A2 \cdot \sin(2 \cdot \pi \cdot t \cdot w(2)) + \dots</math> <math display="block">A3 \cdot \cos(2 \cdot \pi \cdot t \cdot w(3));</math> где:  <math>A1 = 4, A2 = 2, A3 = 5,</math>  <math>w(1) = 1 \text{ Гц}, w(2) = 5 \text{ Гц}, w(3) = 10 \text{ Гц}.</math>  Отфильтровать сигнал от частот выше 6 Гц</p>
13	<p>Задан полигармонический сигнал:  <math display="block">Y = A1 \cdot \cos(2 \cdot \pi \cdot t \cdot w(1)) + A2 \cdot \sin(2 \cdot \pi \cdot t \cdot w(2)) + \dots</math> <math display="block">A3 \cdot \cos(2 \cdot \pi \cdot t \cdot w(3));</math> где:  <math>A1 = 10, A2 = 5, A3 = 5,</math></p>

Вариант	Задание
	$w(1) = 1 \text{ Гц}, w(2) = 5 \text{ Гц}, w(3) = 50 \text{ Гц}.$ Отфильтровать сигнал от частот от 4 до 6 Гц
14	Задан полигармонический сигнал: $Y = A_1 \cos(2\pi t w(1)) + A_2 \sin(2\pi t w(2)) + \dots$ $A_3 \cos(2\pi t w(3));$ где: $A_1 = 5, A_2 = 5, A_3 = 2,$ $w(1) = 1 \text{ Гц}, w(2) = 10 \text{ Гц}, w(3) = 20 \text{ Гц}.$ Отфильтровать сигнал от частот от 10 до 11 Гц
15	Задан полигармонический сигнал: $Y = A_1 \cos(2\pi t w(1)) + A_2 \sin(2\pi t w(2)) + \dots$ $A_3 \cos(2\pi t w(3));$ где: $A_1 = 3, A_2 = 5, A_3 = 7,$ $w(1) = 1 \text{ Гц}, w(2) = 10 \text{ Гц}, w(3) = 50 \text{ Гц}.$ Выделить сигнал с частотами от 9 до 11 Гц
16	Задан полигармонический сигнал: $Y = A_1 \cos(2\pi t w(1)) + A_2 \sin(2\pi t w(2)) + \dots$ $A_3 \cos(2\pi t w(3));$ где: $A_1 = 1, A_2 = 2, A_3 = 5,$ $w(1) = 1 \text{ Гц}, w(2) = 5 \text{ Гц}, w(3) = 50 \text{ Гц}.$ Отфильтровать сигнал от частот ниже 5 Гц
17	Задан полигармонический сигнал: $Y = A_1 \cos(2\pi t w(1)) + A_2 \sin(2\pi t w(2)) + \dots$ $A_3 \cos(2\pi t w(3));$ где: $A_1 = 10, A_2 = 5, A_3 = 1,$ $w(1) = 1 \text{ Гц}, w(2) = 2,5 \text{ Гц}, w(3) = 5 \text{ Гц}.$ Отфильтровать сигнал от частот выше 1 Гц
18	Задан полигармонический сигнал: $Y = A_1 \cos(2\pi t w(1)) + A_2 \sin(2\pi t w(2)) + \dots$ $A_3 \cos(2\pi t w(3));$ где: $A_1 = 10, A_2 = 5, A_3 = 5,$ $w(1) = 10 \text{ Гц}, w(2) = 5 \text{ Гц}, w(3) = 100 \text{ Гц}.$ Отфильтровать сигнал от частот от 9 до 11 Гц и от частот от 99 Гц до 100 Гц
19	Задан полигармонический сигнал: $Y = A_1 \cos(2\pi t w(1)) + A_2 \sin(2\pi t w(2)) + \dots$ $A_3 \cos(2\pi t w(3));$ где: $A_1 = 1, A_2 = 10, A_3 = 50,$ $w(1) = 1 \text{ Гц}, w(2) = 50 \text{ Гц}, w(3) = 100 \text{ Гц}.$ Выделить сигнал с частотами от 99 Гц до 100 Гц

Вариант	Задание
20	<p>Задан полигармонический сигнал:  <math display="block">Y = A1 \cdot \cos(2 \cdot \pi \cdot t \cdot w(1)) + A2 \cdot \sin(2 \cdot \pi \cdot t \cdot w(2)) + \dots</math> <math display="block">A3 \cdot \cos(2 \cdot \pi \cdot t \cdot w(3));</math> где:  <math>A1 = 1, A2 = 5, A3 = 10,</math>  <math>w(1) = 1 \text{ Гц}, w(2) = 4 \text{ Гц}, w(3) = 10 \text{ Гц}.</math>  Выделить сигнал с частотами от 3 до 5 Гц</p>
21	<p>Задан полигармонический сигнал:  <math display="block">Y = A1 \cdot \cos(2 \cdot \pi \cdot t \cdot w(1)) + A2 \cdot \sin(2 \cdot \pi \cdot t \cdot w(2)) + \dots</math> <math display="block">A3 \cdot \cos(2 \cdot \pi \cdot t \cdot w(3)) + A4 \cdot \sin(2 \cdot \pi \cdot t \cdot w(4));</math> где:  <math>A1 = 20, A2 = 10, A3 = 5, A4 = 1,</math>  <math>w(1) = 15 \text{ Гц}, w(2) = 50 \text{ Гц}, w(3) = 80 \text{ Гц}, w(4) = 100 \text{ Гц}.</math>  Отфильтровать сигнал от частот ниже 100 Гц</p>
22	<p>Задан полигармонический сигнал:  <math display="block">Y = A1 \cdot \cos(2 \cdot \pi \cdot t \cdot w(1)) + A2 \cdot \sin(2 \cdot \pi \cdot t \cdot w(2)) + \dots</math> <math display="block">A3 \cdot \cos(2 \cdot \pi \cdot t \cdot w(3)) + A4 \cdot \sin(2 \cdot \pi \cdot t \cdot w(4));</math> где:  <math>A1 = 16, A2 = 8, A3 = 1, A4 = 10,</math>  <math>w(1) = 1 \text{ Гц}, w(2) = 5 \text{ Гц}, w(3) = 10 \text{ Гц}, w(4) = 20 \text{ Гц}.</math>  Отфильтровать сигнал от частот выше 10 Гц</p>
23	<p>Задан полигармонический сигнал:  <math display="block">Y = A1 \cdot \cos(2 \cdot \pi \cdot t \cdot w(1)) + A2 \cdot \sin(2 \cdot \pi \cdot t \cdot w(2)) + \dots</math> <math display="block">A3 \cdot \cos(2 \cdot \pi \cdot t \cdot w(3)) + A4 \cdot \sin(2 \cdot \pi \cdot t \cdot w(4));</math> где:  <math>A1 = 10, A2 = 5, A3 = 2, A4 = 1,</math>  <math>w(1) = 1 \text{ Гц}, w(2) = 3 \text{ Гц}, w(3) = 6 \text{ Гц}, w(4) = 12 \text{ Гц}.</math>  Отфильтровать сигнал от частот от 3 до 6 Гц</p>
24	<p>Задан полигармонический сигнал:  <math display="block">Y = A1 \cdot \cos(2 \cdot \pi \cdot t \cdot w(1)) + A2 \cdot \sin(2 \cdot \pi \cdot t \cdot w(2)) + \dots</math> <math display="block">A3 \cdot \cos(2 \cdot \pi \cdot t \cdot w(3)) + A4 \cdot \sin(2 \cdot \pi \cdot t \cdot w(4));</math> где:  <math>A1 = 8, A2 = 8, A3 = 2, A4 = 1,</math>  <math>w(1) = 0,5 \text{ Гц}, w(2) = 5 \text{ Гц}, w(3) = 10 \text{ Гц}, w(4) = 20 \text{ Гц}.</math>  Отфильтровать сигнал от частот от 4 до 5 Гц</p>
25	<p>Задан полигармонический сигнал:  <math display="block">Y = A1 \cdot \cos(2 \cdot \pi \cdot t \cdot w(1)) + A2 \cdot \sin(2 \cdot \pi \cdot t \cdot w(2)) + \dots</math> <math display="block">A3 \cdot \cos(2 \cdot \pi \cdot t \cdot w(3)) + A4 \cdot \sin(2 \cdot \pi \cdot t \cdot w(4));</math> где:  <math>A1 = 5, A2 = 10, A3 = 20, A4 = 1,</math>  <math>w(1) = 5 \text{ Гц}, w(2) = 10 \text{ Гц}, w(3) = 11 \text{ Гц}, w(4) = 15 \text{ Гц}.</math>  Выделить сигнал с частотами от 9 до 11 Гц</p>
26	<p>Задан полигармонический сигнал:  <math display="block">Y = A1 \cdot \cos(2 \cdot \pi \cdot t \cdot w(1)) + A2 \cdot \sin(2 \cdot \pi \cdot t \cdot w(2)) + \dots</math> <math display="block">A3 \cdot \cos(2 \cdot \pi \cdot t \cdot w(3)) + A4 \cdot \sin(2 \cdot \pi \cdot t \cdot w(4));</math> где:</p>

Вариант	Задание
	$A_1 = 50, A_2 = 10, A_3 = 10, A_4 = 1,$ $w(1) = 1,5 \text{ Гц}, w(2) = 2,5 \text{ Гц}, w(3) = 10 \text{ Гц}, w(4) = 20 \text{ Гц}.$ Отфильтровать сигнал от частот ниже 5 Гц
27	Задан полигармонический сигнал: $Y = A_1 \cos(2\pi t w(1)) + A_2 \sin(2\pi t w(2)) + \dots$ $A_3 \cos(2\pi t w(3)) + A_4 \sin(2\pi t w(4));$ где: $A_1 = 20, A_2 = 15, A_3 = 10, A_4 = 1,$ $w(1) = 2 \text{ Гц}, w(2) = 8 \text{ Гц}, w(3) = 10 \text{ Гц}, w(4) = 20 \text{ Гц}.$ Отфильтровать сигнал от частот выше 8 Гц
28	Задан полигармонический сигнал: $Y = A_1 \cos(2\pi t w(1)) + A_2 \sin(2\pi t w(2)) + \dots$ $A_3 \cos(2\pi t w(3)) + A_4 \sin(2\pi t w(4));$ где: $A_1 = 2, A_2 = 5, A_3 = 10, A_4 = 1,$ $w(1) = 1 \text{ Гц}, w(2) = 10 \text{ Гц}, w(3) = 50 \text{ Гц}, w(4) = 90 \text{ Гц}.$ Отфильтровать сигнал от частот от 0 до 10 Гц и от частот от 51 Гц до 100 Гц
29	Задан полигармонический сигнал: $Y = A_1 \cos(2\pi t w(1)) + A_2 \sin(2\pi t w(2)) + \dots$ $A_3 \cos(2\pi t w(3)) + A_4 \sin(2\pi t w(4));$ где: $A_1 = 10, A_2 = 8, A_3 = 4, A_4 = 1,$ $w(1) = 0,5 \text{ Гц}, w(2) = 1 \text{ Гц}, w(3) = 10 \text{ Гц}, w(4) = 20 \text{ Гц}.$ Выделить сигнал с частотами от 0,9 Гц до 1 Гц
30	Задан полигармонический сигнал: $Y = A_1 \cos(2\pi t w(1)) + A_2 \sin(2\pi t w(2)) + \dots$ $A_3 \cos(2\pi t w(3)) + A_4 \sin(2\pi t w(4));$ где: $A_1 = 8, A_2 = 3, A_3 = 2, A_4 = 1,$ $w(1) = 2 \text{ Гц}, w(2) = 2,5 \text{ Гц}, w(3) = 4 \text{ Гц}, w(4) = 10 \text{ Гц}.$ Выделить сигнал с частотами от 3 до 4 Гц

## П.2.2 Пример дифференцирования сигнала

Составим в среде МАТНСАД цифровой треугольный сигнал, кусочно задав функцию U:

$$\begin{aligned}
 U(t, t_1, t_2, t_3, t_4, U_1, U_2) := & \left\{ \begin{array}{l}
 V \leftarrow \frac{2 \cdot U_1}{5} \text{ if mod } \left[ t \cdot \left( \frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[ \frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 V \leftarrow \frac{3 \cdot U_1}{5} \text{ if mod } \left[ t \cdot \left( \frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[ \frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 V \leftarrow \frac{4 \cdot U_1}{5} \text{ if mod } \left[ t \cdot \left( \frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[ \frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 V \leftarrow U_1 \text{ if mod } \left[ t \cdot \left( \frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[ \frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 V \leftarrow U_1 \text{ if mod } \left[ t \cdot \left( \frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[ \frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 V \leftarrow \frac{4 \cdot U_1}{5} \text{ if mod } \left[ t \cdot \left( \frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[ \frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 V \leftarrow \frac{3 \cdot U_1}{5} \text{ if mod } \left[ t \cdot \left( \frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[ \frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 V \leftarrow \frac{2 \cdot U_1}{5} \text{ if mod } \left[ t \cdot \left( \frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[ \frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 V \leftarrow \frac{U_1}{5} \text{ if mod } \left[ t \cdot \left( \frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[ \frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 V \leftarrow \frac{U_2}{5} \text{ if mod } \left[ t \cdot \left( \frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[ \frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 V \leftarrow \frac{2 \cdot U_2}{5} \text{ if mod } \left[ t \cdot \left( \frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[ \frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 V \leftarrow \frac{3 \cdot U_2}{5} \text{ if mod } \left[ t \cdot \left( \frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[ \frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 V \leftarrow \frac{4 \cdot U_2}{5} \text{ if mod } \left[ t \cdot \left( \frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[ \frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 V \leftarrow U_2 \text{ if mod } \left[ t \cdot \left( \frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[ \frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 V \leftarrow U_2 \text{ if mod } \left[ t \cdot \left( \frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[ \frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 V \leftarrow \frac{4 \cdot U_2}{5} \text{ if mod } \left[ t \cdot \left( \frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[ \frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 V \leftarrow \frac{3 \cdot U_2}{5} \text{ if mod } \left[ t \cdot \left( \frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[ \frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 V \leftarrow \frac{2 \cdot U_2}{5} \text{ if mod } \left[ t \cdot \left( \frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[ \frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 V \leftarrow \frac{U_2}{5} \text{ if mod } \left[ t \cdot \left( \frac{t_1 + t_2 + t_3 + t_4}{5} \right), \left[ \frac{(t_1 + t_2 + t_3 + t_4)^2}{5} \right] \right] > (t_1 + t_2 + t_3 + t_4) \\
 V \leftarrow \frac{U_1}{5} \text{ otherwise}
 \end{array} \right.
 \end{aligned}$$



$T_1$  – длительность первой четверти сигнала  $U$  в секундах:

$$T_1 := 0.040$$

$U_1$  – амплитуда сигнала  $U$  в первом полупериоде:

$$U_1 := 12$$

$T_2$  – длительность второй четверти сигнала  $U$  в секундах:

$$T_2 := 0.040$$

$U_2$  – амплитуда сигнала  $U$  во втором полупериоде:

$$U_2 := -12$$

$T_3$  – длительность третьей четверти сигнала  $U$  в секундах:

$$T_3 := 0.040$$

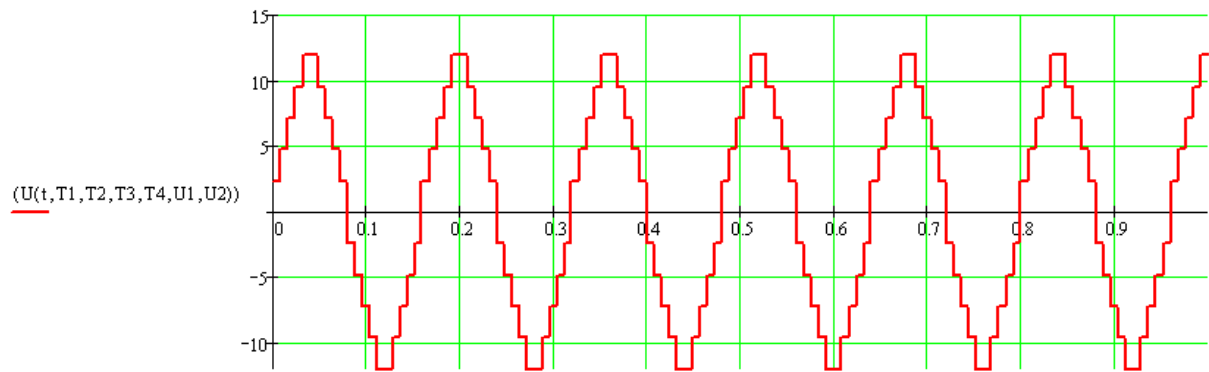
$T_4$  – длительность четвертой четверти сигнала  $U$  в секундах:

$$T_4 := 0.040$$

Общее время периода составит:

$$T_{\text{sum}} := T_1 + T_2 + T_3 + T_4$$

Далее построим изображение сигнала  $U$  на временной диаграмме.



(t)

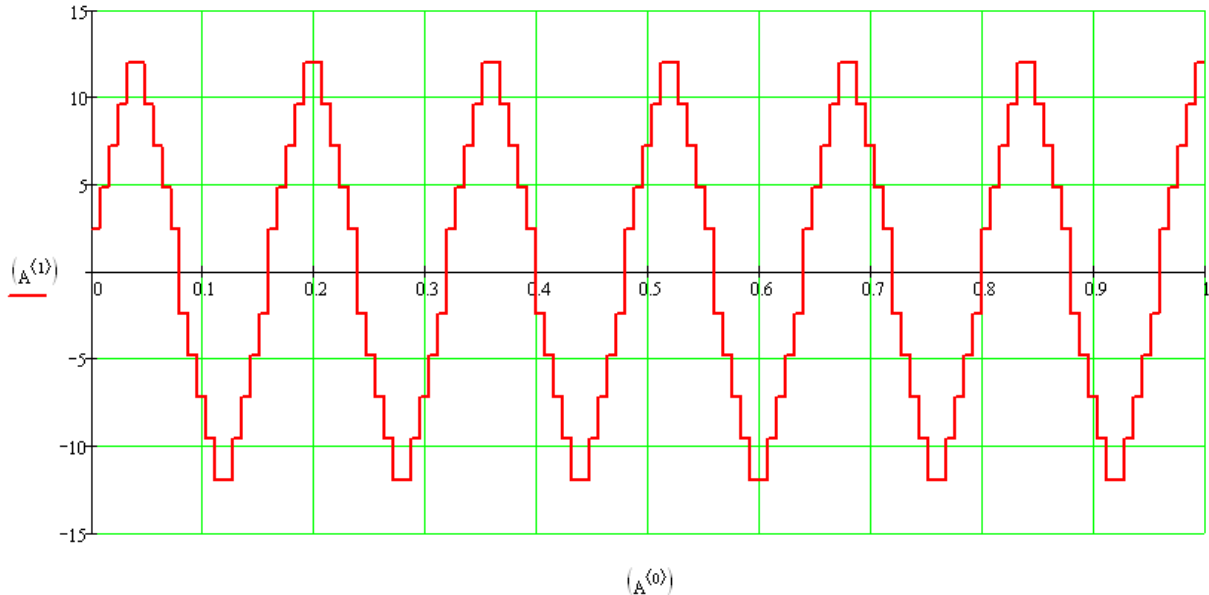
Зададим функцию  $F_{toA}$  для перевода функции сигнала  $U$  в двумерный массив, состоящий из двух столбцов, где первый столбец – столбец значений времени, второй – столбец значений дискретного сигнала. Аргументы функции  $F_{toA}$ :  $totaltime$  – общее время моделирования сигнала, сек.,  $points$  – количество отсчетов (точек).  $Step$  – шаг времени, сек.

```
FtoA(totaltime, points) :=  
  Z ← 0  
  n ← 0  
  step ← totaltime / points  
  while n ≤ points  
    Zn,0 ← step · n  
    Zn,1 ← U[Zn,0, T1, T2, T3, T4, U1, U2]  
    n ← n + 1  
  return Z
```

Сформируем массив A со значениями сигнала U:

```
A := FtoA(1,10000)
```

Далее построим изображение дискретного сигнала на временной диаграмме.



К полученному дискретному сигналу применим метод сглаживания, использующий алгоритм нахождения скользящего среднего значения сигнала:

```
w := 250
```

```
n := rows(A)
```

```
n = 1 × 104
```

```
i := 0..n - 1
```

$$z1_i := \begin{cases} \frac{\sum_{j=0}^i (A^{(1)})_j}{i+1} & \text{if } i < w \\ \frac{\sum_{j=i-w+1}^i (A^{(1)})_j}{w} & \text{otherwise} \end{cases}$$

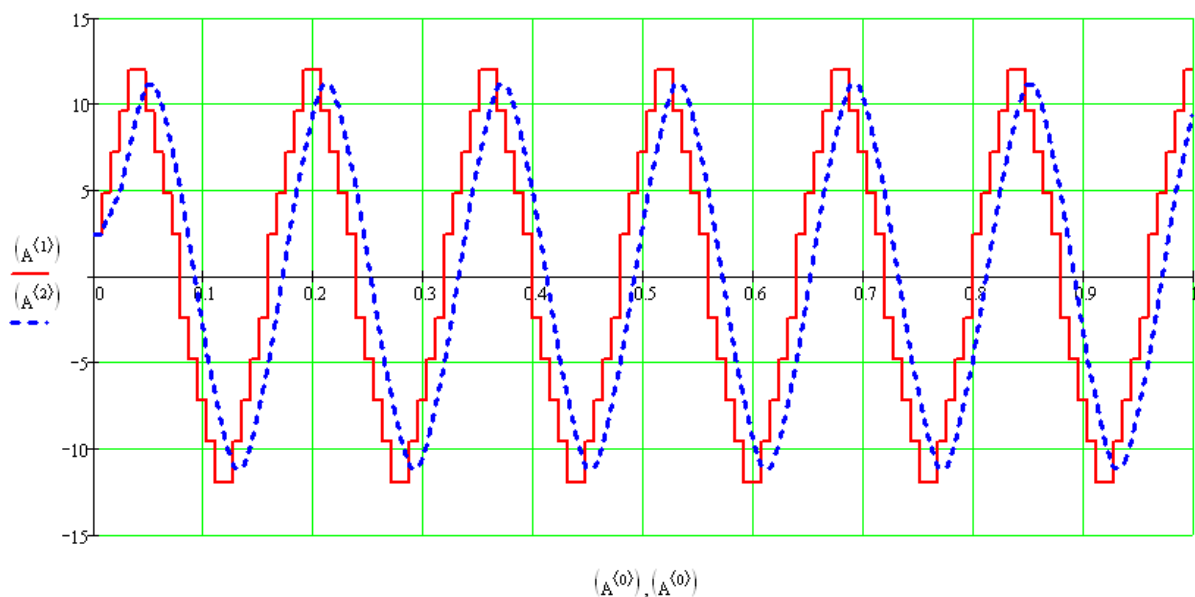
где:  $A_j$  – массив A исходного (несглаженного) сигнала со значением сигнала в точке времени  $j$ ,  $i$  – текущая обрабатываемая точка времени,  $w$  – константа, определяющая при сглаживании пропуск точек времени в количестве  $w$  ( $0 < w < n$ ,  $n$  – общее количество точек времени (элементов массива A)),  $z1$  – массив со сглаженным уровнем сигнала.

Полученный одномерный массив (матрицу-столбец) z1 присвоим третьему столбцу  $A^{(2)}$ :

$$A^{(2)} := z1$$

	0	1	2
0	0	2.4	2.4
1	$1 \cdot 10^{-4}$	2.4	2.4
2	$2 \cdot 10^{-4}$	2.4	2.4
3	$3 \cdot 10^{-4}$	2.4	2.4
4	$4 \cdot 10^{-4}$	2.4	2.4
5	$5 \cdot 10^{-4}$	2.4	2.4
6	$6 \cdot 10^{-4}$	2.4	2.4
7	$7 \cdot 10^{-4}$	2.4	2.4
8	$8 \cdot 10^{-4}$	2.4	2.4
9	$9 \cdot 10^{-4}$	2.4	2.4
10	$1 \cdot 10^{-3}$	2.4	2.4
11	$1.1 \cdot 10^{-3}$	2.4	2.4
12	$1.2 \cdot 10^{-3}$	2.4	2.4
13	$1.3 \cdot 10^{-3}$	2.4	2.4
14	$1.4 \cdot 10^{-3}$	2.4	2.4
15	$1.5 \cdot 10^{-3}$	2.4	2.4

Далее построим на одной временной диаграмме изображения несглаженного и сглаженного дискретных сигналов.



Зададим функцию Diff для получения дискретного продифференцированного по времени сигнала U, где: S – двумерный массив с исходным сигналом, N – номер столбца в массиве S с данными для дифференцирования ( $0 < N$ ):

```

Diff(S,N) :=
  Y ← 0
  t ← 0
  a ← 0
  a0 ← 0
  for Y ∈ 0..last(S<0>)
    dt ← S<Y,0> - t
    t ← S<Y,0>
    a0 ← a
    a ← S<Y,N>
    dt ← 0.000000000000001 if dt ≤ 0
    S<Y,N+1> ← (a - a0) / dt
  S

```

Продифференцируем данные в третьем столбце  $A^{<2>}$  и получим в том же массиве  $A$  четвертый столбец  $A^{<3>}$  с продифференцированным дискретным сигналом  $U$ :

```
A := Diff(A,2)
```

К полученному дискретному сигналу в четвертом столбце  $A^{<3>}$  применим метод сглаживания:

```
w := 250
```

```
n := rows(A)
```

```
n = 1 × 104
```

```
i := 0..n - 1
```

$$z1_i := \begin{cases} \frac{\sum_{j=0}^i (A^{<3>})_j}{i+1} & \text{if } i < w \\ \frac{\sum_{j=i-w+1}^i (A^{<3>})_j}{w} & \text{otherwise} \end{cases}$$

```
A<3> := z1
```

Продифференцируем данные в четвертом столбце  $A^{<3>}$  и получим в том же массиве  $A$  пятый столбец  $A^{<4>}$  с дважды продифференцированным дискретным сигналом  $U$ :

```
A := Diff(A,3)
```

К полученному дискретному сигналу в пятом столбце  $A^{(4)}$  применим метод сглаживания:

```
w := 300
```

```
n := rows(A)
```

```
n = 1 × 104
```

```
i := 0..n - 1
```

$$z1_i := \begin{cases} \frac{\sum_{j=0}^i (A^{(4)})_j}{i+1} & \text{if } i < w \\ \frac{\sum_{j=i-w+1}^i (A^{(4)})_j}{w} & \text{otherwise} \end{cases}$$

```
A(4) := z1
```

Обрезаем левый край сигнала, удаляя первые (верхние) 500 строк из массива A (тем самым удаляя первые 500 точек дискретного сигнала):

```
CUT := 550
```

```
TIMESHIFT := ACUT,0
```

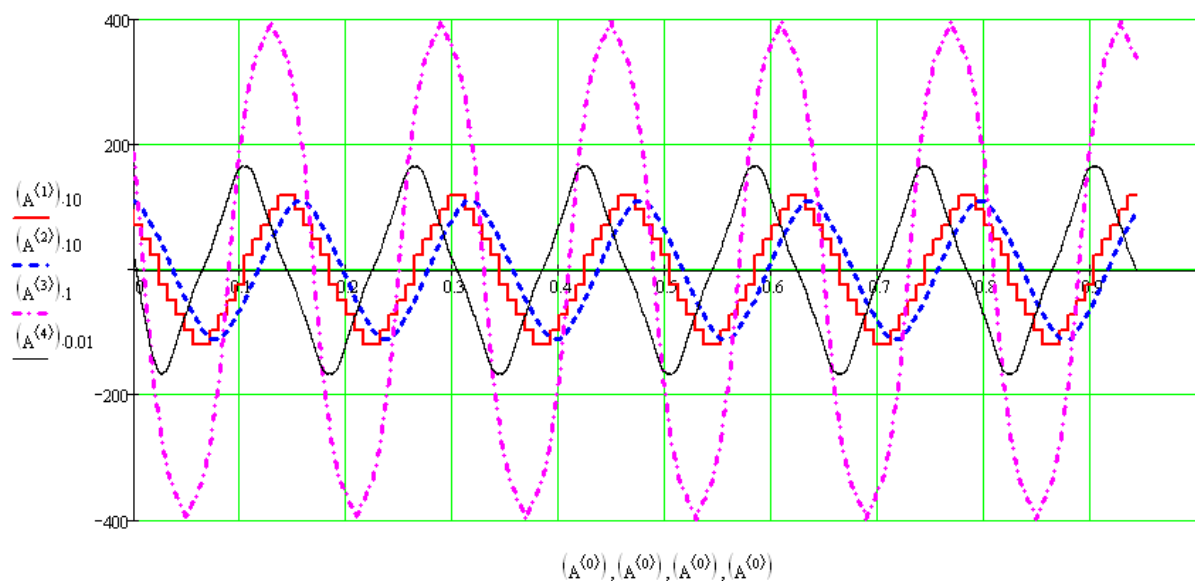
```
A := submatrix[A,CUT,(last(A(0))),0,4]
```

Смещаем влево шкалу времени после обрезки дискретного сигнала:

```
A(0) := (A(0)) - TIMESHIFT
```

	0	1	2	3	4
0	0	9.6	11.136	188.16	2.816·10 <sup>3</sup>
1	1·10 <sup>-4</sup>	9.6	11.136	187.008	2.739·10 <sup>3</sup>
2	2·10 <sup>-4</sup>	9.6	11.136	185.856	2.662·10 <sup>3</sup>
3	3·10 <sup>-4</sup>	9.6	11.136	184.704	2.586·10 <sup>3</sup>
4	4·10 <sup>-4</sup>	9.6	11.136	183.552	2.509·10 <sup>3</sup>
5	5·10 <sup>-4</sup>	9.6	11.136	182.4	2.432·10 <sup>3</sup>
6	6·10 <sup>-4</sup>	9.6	11.136	181.248	2.355·10 <sup>3</sup>
7	7·10 <sup>-4</sup>	9.6	11.136	180.096	2.278·10 <sup>3</sup>
8	8·10 <sup>-4</sup>	9.6	11.136	178.944	2.202·10 <sup>3</sup>
9	9·10 <sup>-4</sup>	9.6	11.136	177.792	2.125·10 <sup>3</sup>
10	1·10 <sup>-3</sup>	9.6	11.136	176.64	2.048·10 <sup>3</sup>
11	1.1·10 <sup>-3</sup>	7.2	11.126	175.104	1.958·10 <sup>3</sup>
12	1.2·10 <sup>-3</sup>	7.2	11.117	173.568	1.869·10 <sup>3</sup>
13	1.3·10 <sup>-3</sup>	7.2	11.107	172.032	1.779·10 <sup>3</sup>
14	1.4·10 <sup>-3</sup>	7.2	11.098	170.496	1.69·10 <sup>3</sup>
15	1.5·10 <sup>-3</sup>	7.2	11.088	168.96	1.6·10 <sup>3</sup>

Построим на одной временной диаграмме изображения несглаженного и сглаженного дискретных сигналов  $U$ , а также одно- и двукратно проинтегрированные сигналы  $U$ :



Для сглаженного дискретного сигнала  $U$  в столбце  $A^{(2)}$  применим метод сокращения точек, используя функцию `rdots` с аргументами:  $S$  - двумерный массив с исходным сигналом,  $d$  - выборка (в новом массиве присутствует каждая  $d$ -точка исходного массива  $S$ ).

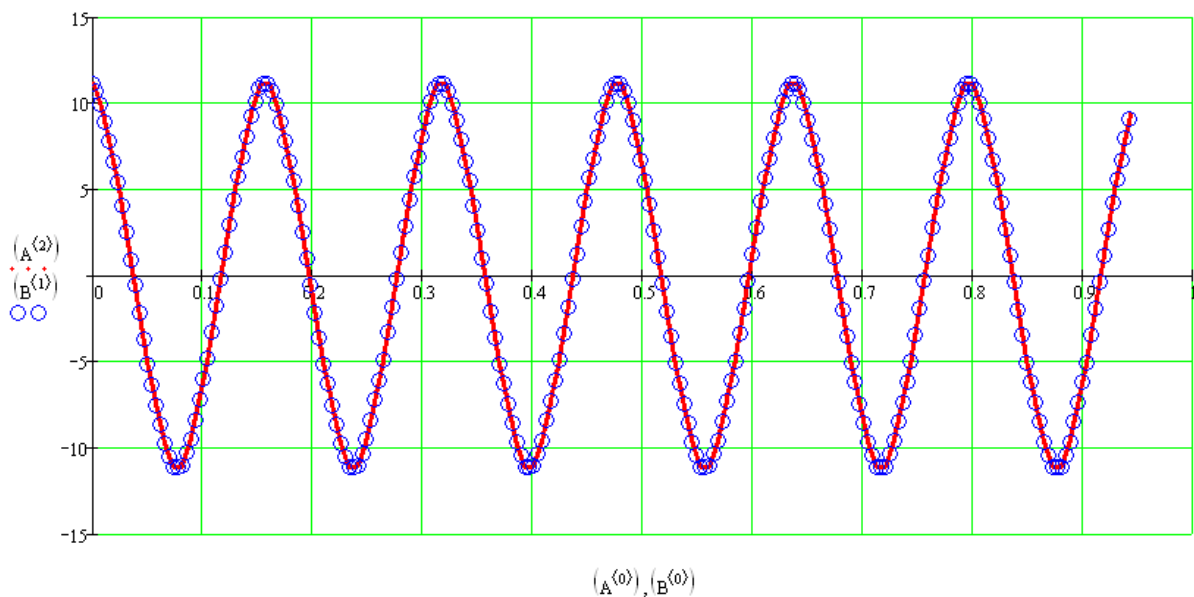
```

rdots(S,d) :=
  Y ← 0
  N ← 0
  while N ≤ last(S(0))
    XY,0 ← SN,0
    XY,1 ← SN,2
    N ← N + (d - 1)
    Y ← Y + 1
  X

```

Сформируем новый массив В:

$B := \text{rdots}(A, 40)$



### Задания

1. Составить в среде Mathcad цифровой (квантованный) сигнал, кусочно задав функцию  $U$  заданной формы в соответствии с номером варианта задания (таблица 2).
2. Сформировать дискретный сигнал.
3. К полученному дискретному сигналу применить метод сглаживания, использующий алгоритм нахождения скользящего среднего значения сигнала.
4. Дважды продифференцировать полученный сигнал.
5. Выполнить построение исходного цифрового сигнала, дискретного сигнала, сглаженного дискретного сигнала, сигнала после однократного дифференцирования, сигнала после двукратного дифференцирования.
6. Оформить отчет о работе.

Таблица 2 – Варианты заданий

Вариант	Форма сигнала
1	двухполярный сигнал с положительной пилообразной полуволной и отрицательной прямоугольной полуволной
2	двухполярный треугольный сигнал
3	двухполярный сигнал с положительной синусоидальной полуволной и отрицательной прямоугольной полуволной
4	двухполярный пилообразный сигнал
5	синусоидальный сигнал без отрицательной полуволны
6	синусоидальный сигнал без положительной полуволны
7	однополярный пилообразный сигнал
8	однополярный треугольный сигнал
9	двухполярный сигнал с положительной прямоугольной полуволной и отрицательной синусоидальной полуволной
10	двухполярный сигнал с положительной прямоугольной полуволной и отрицательной пилообразной полуволной



### П.2.3 Пример интегрирования сигнала

Составим в среде MATHCAD прямоугольный сигнал, кусочно задав функцию  $U$  ( $T1$  – длительность первого полупериода сигнала, сек.,  $U1$  – значение сигнала в первом полупериоде,  $T2$  – длительность второго полупериода сигнала, сек.,  $U2$  – значение сигнала во втором полупериоде,  $T$  – длительность периода сигнала, сек.):

$T1 := 0.080$

$U1 := 5$

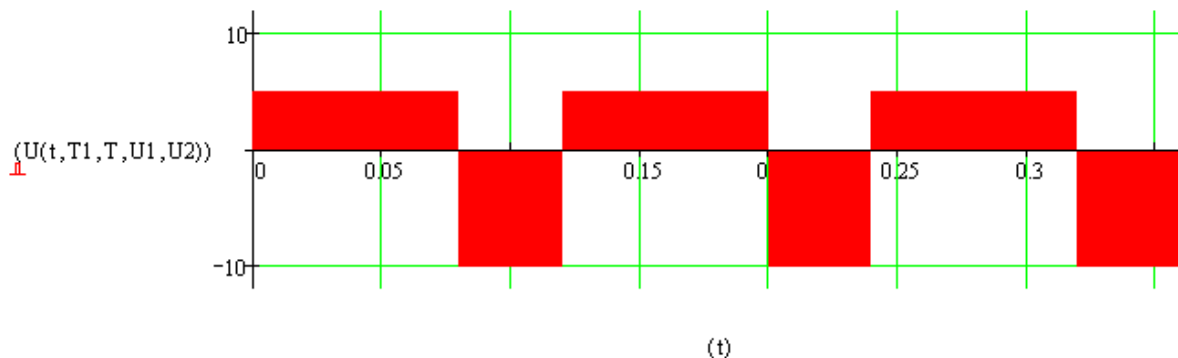
$T2 := 0.040$

$U2 := -10$

$T := T1 + T2$

$$U(t, t1, T, Umax, Umin) := \begin{cases} V \leftarrow Umax & \text{if } \text{mod}\left[t \cdot \left(\frac{T}{t1}\right), (t1) \cdot \left(\frac{T}{t1}\right)\right] \leq T \cdot \left(\frac{T}{t1}\right) \\ V \leftarrow Umin & \text{if } \text{mod}\left[t \cdot \left(\frac{T}{t1}\right), (T) \cdot \left(\frac{T}{t1}\right)\right] \geq t1 \cdot \left(\frac{T}{t1}\right) \\ \text{return } V \end{cases}$$

Далее построим изображение сигнала  $U$  на временной диаграмме.



Зададим функцию  $FtoA$  для перевода функции сигнала  $U$  в двумерный массив, состоящий из двух столбцов, где первый столбец – столбец значений времени, второй – столбец значений дискретного сигнала. Аргументы функции  $FtoA$ :  $totaltime$  – общее время моделирования сигнала, сек.,  $points$  – количество отсчетов (точек).  $Step$  – шаг времени, сек.

```

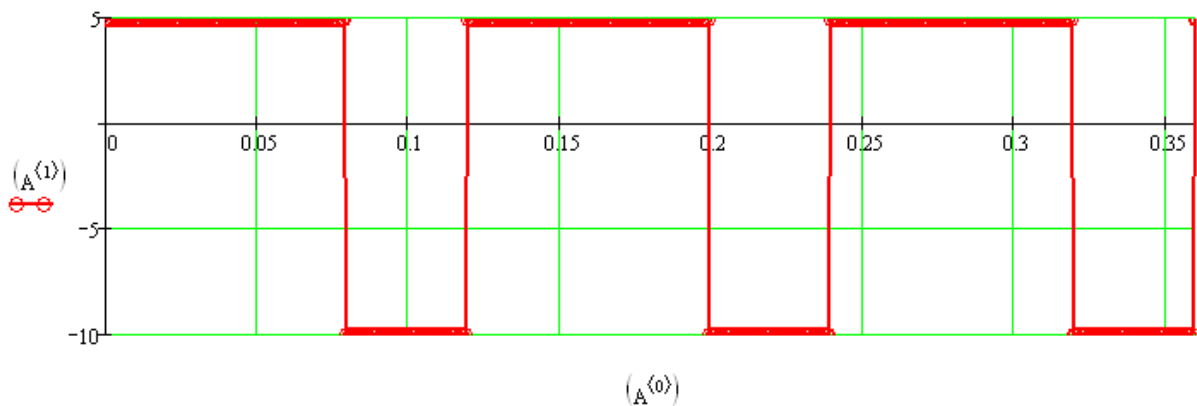
FtoA(totaltime, points) :=
  Z ← 0
  n ← 0
  step ← totaltime / points
  while n ≤ points
    Zn,0 ← step · n
    Zn,1 ← U[Zn,0, T1, T, U1, U2]
    n ← n + 1
  return Z

```

Сформируем массив A со значениями сигнала U:

```
A := FtoA(1, 1000)
```

Далее построим изображение дискретного сигнала на временной диаграмме.



Зададим функцию Integ для получения дискретного проинтегрированного по времени сигнала U, где: S – двумерный массив с исходным сигналом, N – номер столбца в массиве S с данными для интегрирования (0 < N):

```

Integ(S, N) :=
  Y ← 1
  INT ← 0
  L ← last(S<sup>0</sup>)
  SY, N+1 ← 0
  while Y ≤ L
    INT ← [(SY, N) · [(SY, 0) - (SY-1, 0))] + INT
    SY, N+1 ← INT
    Y ← Y + 1
  S

```

Проинтегрируем данные во втором столбце A<sup><1></sup> и получим в том же массиве A третий столбец A<sup><2></sup> с проинтегрированным дискретным сигналом U:

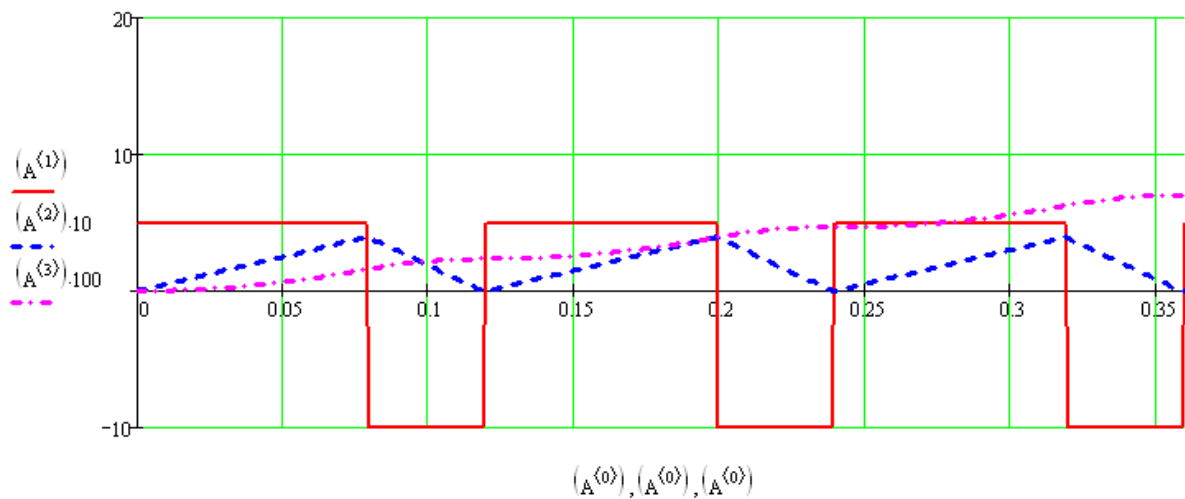
$A := \text{Integ}(A,1)$

Проинтегрируем данные в третьем столбце  $A^{(2)}$  и получим в том же массиве  $A$  четвертый столбец  $A^{(3)}$  с дважды проинтегрированным дискретным сигналом  $U$ :

$A := \text{Integ}(A,2)$

	0	1	2	3
985	0.985	5	0.125	0.189
986	0.986	5	0.13	0.189
987	0.987	5	0.135	0.189
988	0.988	5	0.14	0.189
989	0.989	5	0.145	0.189
990	0.99	5	0.15	0.19
991	0.991	5	0.155	0.19
992	0.992	5	0.16	0.19
993	0.993	5	0.165	0.19
994	0.994	5	0.17	0.19
995	0.995	5	0.175	0.19
996	0.996	5	0.18	0.191
997	0.997	5	0.185	0.191
998	0.998	5	0.19	0.191
999	0.999	5	0.195	0.191
1000	1	5	0.2	0.191

Построим на одной временной диаграмме изображение дискретного сигнала  $U$ , а также одно- и двукратно проинтегрированные сигналы  $U$ :



### Задания

1. Составить в среде Mathcad прямоугольный сигнал, кусочно задав функцию  $U$ , используя следующие параметры:  $T1$  – длительность первого полупериода сигнала, сек.,  $U1$  – значение сигнала в первом полупериоде,  $T2$  – длительность второго полупериода сигнала, сек.,  $U2$  – значение

сигнала во втором полупериоде, при этом  $T1 = N \cdot 0,01$ ,  $U1 = N \cdot 2$ ,  $T2 = (T1 \cdot U1)/5$ ,

$U2 = -(T1 \cdot U1)/T2$ , где  $N$  – номер варианта задания.

2. Сформировать дискретный сигнал.
3. Дважды проинтегрировать полученный сигнал.
4. Выполнить построение исходного цифрового сигнала, дискретного сигнала, сигнала после однократного интегрирования, сигнала после двукратного интегрирования.
5. Дважды продифференцировать сигнал, полученный после двукратного интегрирования.
6. Выполнить построение исходного цифрового сигнала, дискретного сигнала, полученного после двукратного дифференцирования. Получить на временной диаграмме разность этих сигналов и объяснить полученный результат.
7. Оформить отчет о работе.

## П.2.4 Пример работы с изображениями

Сформируем двумерный массив  $M$ , содержащий данные графического файла CAM.BMP в цветовой модели RGB:

```
M := READRGB("CAM.BMP")
```

	0	1	2	3	4	5	6	7	8	9
0	64	64	64	64	96	96	96	96	96	96
1	64	96	64	64	96	96	96	96	96	96
2	64	64	96	96	96	96	96	96	96	96
3	64	64	96	96	96	96	64	64	96	96
4	96	96	96	96	96	96	96	96	96	96
5	96	96	96	64	96	96	96	96	96	64
6	96	96	96	64	96	96	96	96	64	96
7	96	96	96	96	96	96	96	64	64	64
8	96	96	96	96	96	96	64	64	64	32
9	64	64	96	96	64	64	64	64	32	32
10	64	64	64	96	64	64	64	64	32	32
11	64	64	64	64	64	32	32	32	32	32
12	64	64	64	64	32	32	32	0	32	32
13	32	32	32	32	32	32	32	32	32	32
14	32	32	32	32	32	32	32	32	32	64
15	32	32	0	0	32	32	0	64	64	64

Выделим из массива  $M$  массивы  $RM$ ,  $GM$  и  $BM$ , соответствующие красному, зеленому и синему оттенкам графического изображения.

```
rows(M) = 480
```

```
cols(M) = 1.92 × 103
```

```
w :=  $\frac{\text{cols}(M)}{3}$ 
```

```
h := rows(M)
```

Для изменения порядка индексирования массивов присвоим встроенной переменной  $ORIGIN$  значение «1»:

```
ORIGIN := 1
```

```
RM := submatrix(M, 1, h, 1, w)
```

```
GM := submatrix(M, 1, h, w + 1, 2 · w)
```

```
BM := submatrix(M, 1, h, 2 · w + 1, 3 · w)
```

Уменьшим изображение в 2 раза, сократив количество точек:

```
CUT := 2
```

$$\text{newr} := 1.. \left( \frac{\text{rows}(\text{RM})}{\text{CUT}} \right)$$

$$\text{newc} := 1.. \left( \frac{\text{cols}(\text{RM})}{\text{CUT}} \right)$$

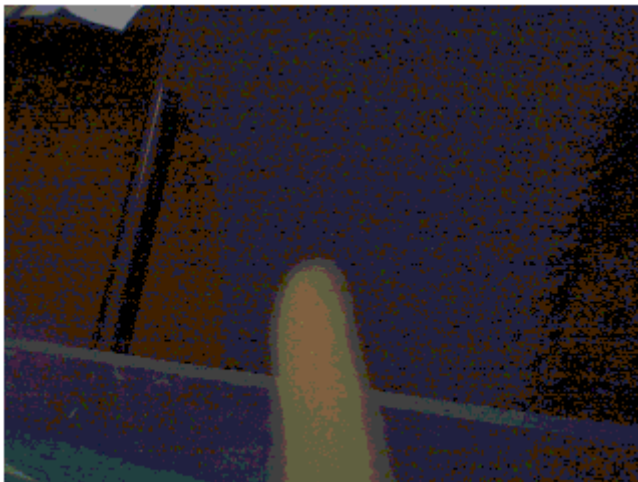
Создадим массивы оттенков NRM, NGM и NBM с уменьшенным изображением:

$$\text{NRM}_{\text{newr}, \text{newc}} := \text{RM}_{\text{CUT} \cdot \text{newr}, \text{CUT} \cdot \text{newc}}$$

$$\text{NGM}_{\text{newr}, \text{newc}} := \text{GM}_{\text{CUT} \cdot \text{newr}, \text{CUT} \cdot \text{newc}}$$

$$\text{NBM}_{\text{newr}, \text{newc}} := \text{BM}_{\text{CUT} \cdot \text{newr}, \text{CUT} \cdot \text{newc}}$$

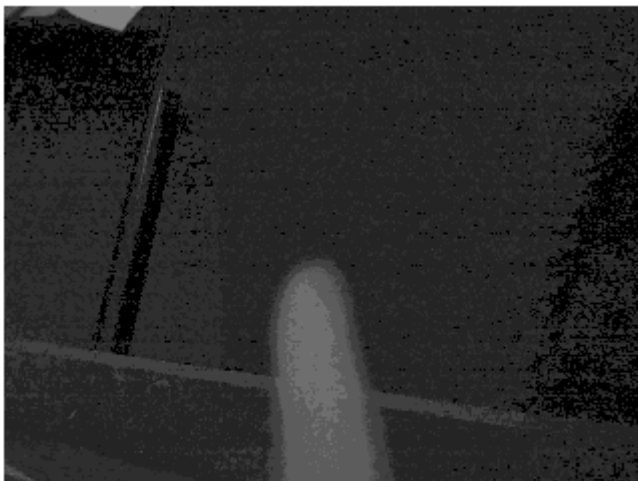
Создадим наложение оттенков NRM, NGM и NBM в компоненте Picture:



NRM, NGM, NBM

Получим графическое изображение в оттенках серого цвета:

$$\text{GRAY} := \text{NRM} \cdot 0.56 + \text{NGM} \cdot 0.33 + \text{NBM} \cdot 0.11$$



GRAY

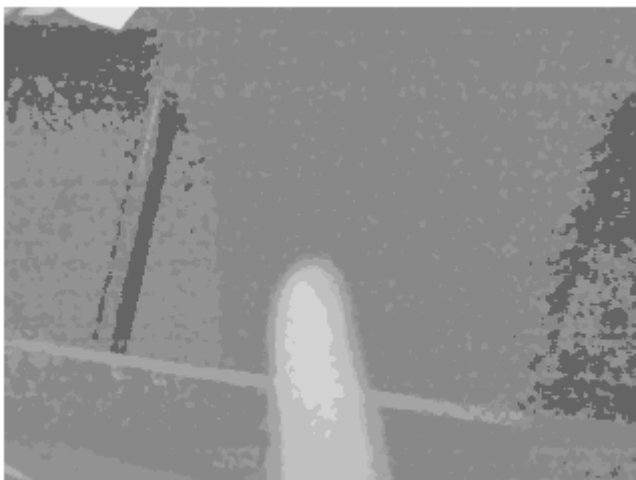
Применим функцию `medflt` (медианный фильтр изображения) для уменьшения уровня шума:

$$\text{medflt}(M, k) := \left| \begin{array}{l} \text{for } i \in \frac{k-1}{2} \dots \text{rows}(M) - \frac{k+1}{2} \\ \quad \text{for } j \in \frac{k-1}{2} \dots \text{cols}(M) - \frac{k+1}{2} \\ \qquad \left| \begin{array}{l} \text{for } r \in 1 \dots k \\ \qquad \text{for } s \in 1 \dots k \\ \qquad \qquad A_{r,s} \leftarrow M_{i+r-\frac{k-1}{2}, j+s-\frac{k-1}{2}} \\ \qquad \qquad N_{i,j} \leftarrow \text{median}(A) \end{array} \right. \\ \qquad \qquad \qquad N \end{array} \right.$$

`GRAY := medflt(GRAY, 3)`

Осветлим полученное изображение:

`GRAY := GRAY + 100`



GRAY

Применим фильтр для выделения некоторого диапазона цветов изображения (цвета изображения меняются от абсолютно черного до абсолютно белого, т.е. от 0 до 255):

$$\text{Filter}(x, x_{\min}, x_{\max}) := \left| \begin{array}{l} x \text{ if } x_{\min} < x \leq x_{\max} \\ 255 \text{ otherwise} \end{array} \right.$$

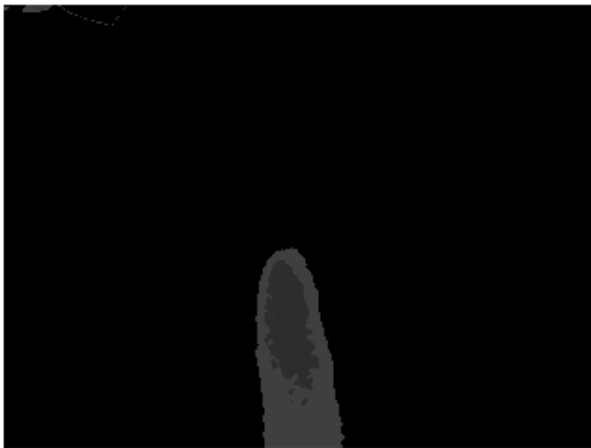
`GRAY2 := Filter(GRAY, 190, 220)`



GRAY2

Получим инверсное изображение:

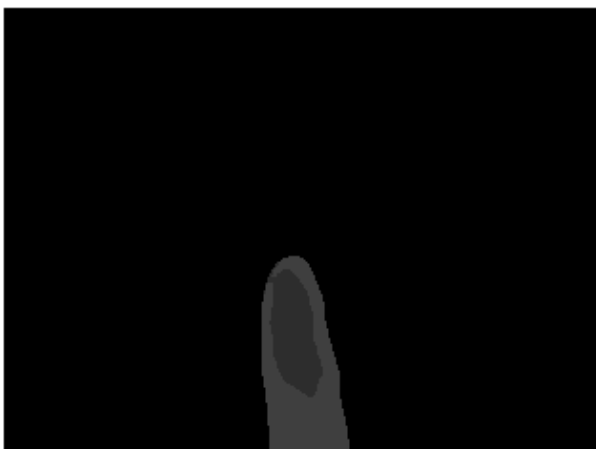
```
INVERSION := 255 - GRAY2
```



INVERSION

Применим функцию `medflt` к инверсному изображению:

```
INVERSION2 := medflt(INVERSION,9)
```



INVERSION2



## Варианты заданий

1. Сформировать в среде Mathcad двумерный массив  $M$ , содержащий данные некоторого графического файла в цветовой модели RGB.
2. Получить графическое изображение в оттенках серого цвета.
3. Применить функцию `medflt`.
4. Применить фильтр для выделения диапазона цветов изображения: от  $N \cdot 10$  до  $N \cdot 25$ , где  $N$  – номер варианта задания. Остальная область изображения закрашивается абсолютно белым цветом.
5. Оформить отчет о работе.

## ЛИТЕРАТУРА

1. Аллен Дж. Архитектура процессоров для цифровой обработки сигналов//ТИИЭР. -1986. -т.73.-N 5.-с.3-37.
2. Блаттер К. Вейвлет-анализ. Основы теории// – М. – Техносфера. - 2006, -279 с.
3. Брейсуэлл Р. Преобразование Хартли. Теория и приложения// -М.-Мир.-1990. -217 с.
4. Гонсалес Р., Вудс Р. Цифровая обработка изображений// Пер. с англ.-Москва.- Техносфера. – 2006. -1072 с.
5. Дагман Э.Е.,Кухарев Г.А. Быстрые дискретные ортогональные преобразования//Новосибирск. – Наука. -1983. -232 с.
6. Губин А.В., Шарыпин Е.В. Архитектура персональных систем обработки изображений//Системы и средства информатики. Ежегодник.-Вып.1.-1989. -с.74-81.
7. Дымков В.И., Сеницын Е.В. Элементы концепции персональных систем обработки изображений//В сб. Системы и средства информатики. Вып. 1.-1989.-с.74-81.
8. Кухарев Г.А.,Тропченко А.Ю.,Шмерко В.П. Системные процессоры для обработки сигналов//Минск. – Беларусь. -1988. -127 с.
9. Кухарев Г.А.,Тропченко А.Ю. Системный процессор для обращения матриц//Известия вузов СССР. Приборостроение.-1990. -т.33. -№ 11 -с.29-32.
10. Очин Е.Ф. Вычислительные системы обработки изображений//Л. - Энергоатомиздат. Ленингр. отд-ние. -1989. -136 с.
11. Прэт У. Цифровая обработка изображений//М. –Мир.-1982. -т.1. - 495 с.-т.2. - 480с.
12. Рабинер Л.,Гоулд Б. Теория и применения цифровой обработки сигналов//М. – Мир. -1978. - 545 с.
13. Уидроу Б., Стирнз С. Адаптивная обработка сигналов//М. - Радио и связь. - 1989, - 440 с.
14. Graps A.L. An introduction to Wavelets//IEEE Computational Sciences and Engineering. -v.2. - №.2. - Summer 1995. - p.p. 50-61.