Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение
высшего образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

Э. Дмитриев, Н.Г. Калашникова

ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ

Методические указания для выполнения практических и
лабораторных работ для студентов технических направлений
подготовки и специальностей

Томск
2025

УДК 621.371, 537.87 (075.8)
ББК 32.841-01я73, 32.811.3я73
Д53

Рецензент:
Рогожников Е.В., заведующий кафедрой телекоммуникаций и основ радиотехники ТУСУРа, кандидат технических наук

Д 53 Цифровая обработка сигналов: Методические указания для выполнения практических и лабораторных работ для студентов технических направлений подготовки и специальностей / Э. Дмитриев, Н.Г. Калашникова – Томск: Томск. гос. ун-т систем управления и радиоэлектроники, 2025. – 48 с.

Методическое пособие предназначено для студентов, обучающихся по направлениям, связанным с цифровой обработкой сигналов и телекоммуникационными технологиями. В пособии содержатся подробные указания для выполнения практических и лабораторных работ, направленных на закрепление и углубление теоретических знаний в области цифровой обработки сигналов.

Практикум охватывает основные алгоритмы и методы обработки сигналов, которые применяются в современных цифровых системах передачи информации. Студенты познакомятся с процессами формирования, кодирования, фильтрации и восстановления сигналов, а также с различными техниками анализа и синтеза сигналов, включая дискретное преобразование и использование математических моделей.

Особое внимание в пособии уделено практическому применению полученных знаний, что позволяет студентам не только разобраться в теории, но и овладеть необходимыми навыками работы с современными цифровыми системами обработки информации. Практические задания, включающие как теоретические расчеты, так и использование специализированных программных средств, помогают студентам развить умения анализа и оптимизации процессов обработки сигналов в реальных условиях.

Одобрено на заседании кафедры ТОР, протокол № 5 от 19 февраля 2025 г.

УДК 621.371, 537.87 (075.8)
ББК 32.841-01я73, 32.811.3я73

© Дмитриев Э., Калашникова Н.Г., 2025
© Томск. гос. ун-т систем управления и радиоэлектроники, 2025

Оглавление

ВВЕДЕНИЕ 4
Практическая работа № 1 5
Практическая работа № 2 8
Практическая работа № 3 9
Практическая работа № 4 11
Практическая работа № 5 13
Практическая работа № 6 15
Практическая работа № 7 18
Практическая работа № 8 20
Практическая работа № 9 24
Лабораторная работа № 1 27
Лабораторная работа № 2 32
Лабораторная работа № 3 37
Лабораторная работа № 4 41
ЗАКЛЮЧЕНИЕ 46
СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ 47
ПРИЛОЖЕНИЕ А (справочное) Параметры функции plot() 48

ВВЕДЕНИЕ

Цель данного методического пособия по дисциплине "Цифровая обработка сигналов" заключается в углублении теоретических знаний студентов в области методов и алгоритмов цифровой обработки сигналов, а также в формировании практических навыков применения этих методов для решения реальных задач. В ходе практикума студенты ознакомятся с основными инструментами и технологиями, используемыми для обработки цифровых сигналов, научатся разрабатывать и реализовывать алгоритмы фильтрации, анализа и преобразования сигналов, а также работать с современными программными средствами для обработки и визуализации сигналов.

Методическое пособие содержит девять практических и четыре лабораторных работ. Практические работы представлены следующим перечнем названий:

1. Знакомство с языком программирования Octave/MATLAB. Основные математические операторы.
2. Знакомство с языком программирования Octave/MATLAB. Работа с массивами.
3. Знакомство с языком программирования Octave/MATLAB. Построение графиков.
4. Ключевые операции ЦОС. Корреляция, линейная и циклическая свёртка.
5. Ключевые операции ЦОС. Двумерная свёртка. Работа с изображениями.
6. Прямое дискретное преобразование Фурье.
7. Прямое дискретное быстрое преобразование Фурье.
8. Расчет коэффициентов КИХ-фильтром методом взвешивания.
9. Цифровая манипуляция.

Лабораторные работы затрагивают следующие темы:

1. Начало работы с Octave/MATLAB.
2. Однородное квантование в АЦП.
3. Многоскоростная обработка сигналов.
4. Фильтр скользящего среднего.

Для выполнения лабораторных и практических работ по дисциплине "Цифровая обработка сигналов" используются современные программные средства, Octave и MATLAB.

MATLAB – это среда для численных вычислений и разработки алгоритмов, широко применяемая в научных исследованиях и инженерной практике для обработки и анализа сигналов. Данная среда предоставляет обширные возможности для реализации методов цифровой обработки сигналов, включая фильтрацию, спектральный анализ, преобразования Фурье, а также другие алгоритмы для работы с временными и частотными рядами. MATLAB включает в себя удобные инструменты для визуализации данных, что позволяет студентам наглядно анализировать результаты обработки сигналов.

Octave, являющийся открытым аналогом MATLAB, также используется для реализации и тестирования алгоритмов цифровой обработки сигналов. Несмотря на наличие некоторых отличий, Octave поддерживает множество функций и инструментов, схожих с MATLAB, что делает его доступным и удобным для студентов, желающих освоить основные принципы работы с цифровыми сигналами.

Практическая работа № 1

«Знакомство с языком программирования Octave/MATLAB. Основные математические операторы»

Цель практической работы: ознакомление студентов с основами работы в языке программирования Octave/MATLAB, изучение базовых операторов и функций.

Задачи практической работы:

– ознакомиться с основными арифметическими, логическими и специальными операторами в языке Octave/MATLAB;

– изучить методы работы с векторами и матрицами, а также освоить создание и модификацию массивов данных.

1. Вводный материал

Операторы – это неотъемлемая часть математических выражений, вычисление которых является одной из основных задач Octave/MATLAB как системы, созданной для выполнения численных расчетов. В таблицах 1.1-1.4 перечислены наиболее часто используемые операторы языка Octave/MATLAB.

Таблица 1.1 – Арифметические операторы

Символ	Роль
+	Сложение
+	Унарный плюс
–	Вычитание
–	Унарный минус
.*	Поэлементное умножение
*	Умножение матриц
./	Поэлементное правое деление
/	Матричное правое деление
.\	Поэлементное левое деление
\	Матричное левое деление
.^	Поэлементная степень
^	Матричная степень
.'	Перемещение
'	Комплексное сопряженное транспонирование

Таблица 1.2 – Операторы отношения

Символ	Роль
==	Равно
~=	Не равно
>	Больше
>=	Больше или равно
<	Меньше
<=	Меньше или равно

Таблица 1.3 – Логические операторы

Символ	Роль
&	Равно

Продолжение таблицы 1.3

|	Не равно
&&	Больше
||	Больше или равно
~	Меньше

Таблица 1.4 – Некоторые специальные символы

Символ	Роль

:	Создание вектора с помощью диапазона чисел от 1 до 10:
	`x = 1 : 10`
	Создание вектора с равномерным шагом, равным 3:
	`x = 1 : 3 : 19`
	Обращение к элементам массива со второго по пятый:
	`A(2 : 5)`
	Обращение ко всем строкам матрицы третьего столбца:
	`A(:, 3)`
	Задание границ цикла:
	`x = 1;`
	`for k = 1 : 25`
	`  x = x + x ^ 2;`
	`end`

;	Задать конец строки при перечислении элементов матрицы:
	`A = [12, 13; 14, 15]`
	Запретить вывод значений массива в командную строку:
	`A = [12, 13; 14, 15];`
	Примечание: если переменной присваивается какое-либо значение, после запуска программы или отдельной строки результат расчета будет показан в командной строке. Знак «;» в конце строки позволяет отменить данную функцию.

()	Приоритет операций:
	`(A. * (B. / C) ) – D`
	Задание аргумента функции:
	`b = abs(x)`
	`plot(t, s)`
	Обращение к элементам массива:
	`A(8)`
	`b(3, 10)`
	`c = d(:, 1)`

[]	Создание вектора или матрицы:
	`X = [10 12 -3]`
	Создание пустой матрицы:
	`A = []`
	Удаление столбца матрицы:
	`A(:, 1) = []`

{}	Создание массива ячеек:
	`C = {[2.6 4.7 3.9], rand(8) * 6, 'This is Sparta!'}`
	Обращение к элементу массива ячеек:
	`A = C{4, 7, 2}`

Любую программу рекомендуется начинать со следующими строками:

```
clc
close all
clear all (или clear variables)
```

Первая команда используется для очищения командной строки, вторая закрывает все открытые окна. Третья команда очищает все переменные в рабочем пространстве (Workspace). Таким образом, при каждом запуске программы можно быть уверенным в полном пересчете всех данных.

2. Задание на практику

Задача 1.

Петя каждый день ложится спать ровно в полночь и недавно узнал, что оптимальное время для его сна составляет X минут. Петя хочет поставить себе будильник так, чтобы он прозвенел ровно через X минут после полуночи, однако для этого необходимо указать время сигнала в формате часы, минуты. Помогите Пете определить, на какое время завести будильник. Результат необходимо вывести в командную строку. Часы и минуты в выводе программы должны располагаться на разных строках.

Выводить выражение в командную строку можно по следующему примеру:

```
a = 5;
disp(['Мой психологический возраст - ',...
    num2str(a),' лет.'])
```

Символ «...» означает продолжение текста команды в следующей строке. Для решения задачи потребуется использование дополнительных функций, например, `fix()`, `round()` или `floor()`. Все функции на языке Octave используются в данной форме:

```
a = fix(b);
```

где `a` – переменная, которой присваивается значение функции, `b` – аргумент функции. Для вызова справки по функции `fix` можно в командной строке написать

```
help fix
```

Задача 2.

Аня узнала, что ей для сна надо X минут. В отличие от Пети, Аня ложится спать после полуночи в H часов и M минут. Помогите Ане определить, на какое время ей поставить будильник, чтобы он прозвенел ровно через X минут после того, как она ляжет спать.

В качестве входных данных задаются значения X, H и M. Гарантируется, что Аня должна проснуться в тот же день, что и заснуть. Программа должна выводить время, на которое нужно поставить будильник: в первой строке часы, во второй – минуты.

3. Контрольные вопросы к работе

1. Какие арифметические операторы доступны в языке Octave/MATLAB, и как они применяются для работы с числами?
2. Как в языке Octave/MATLAB можно работать с векторами и матрицами, и что такое поэлементные операции?
3. Какие функции языка Octave/MATLAB позволяют округлять числа, и в чем разница между ними?

Практическая работа № 2

«Знакомство с языком программирования Octave/MATLAB. Работа с массивами»

Цель практической работы: приобретение навыков работы с массивами и матрицами в языке Octave/MATLAB.

Задачи практической работы:

– освоить этапы создания и работы с массивами на языке Octave/MATLAB;

– изучить функции генерации случайных чисел;

– изучить функции преобразования и перестановки матриц.

1. Задание на практику

Задание 1.

1) Создайте массив размерностью 1x7.

2) Инвертируйте массив, полученный в п.1.

3) Создайте ещё один массив такой же размерности при помощифункции `randi()`.

4) Выполните с полученными массивами: сложение, вычитание, умножение, деление (левое и правое).

5) Объедините массивы, полученные при сложении и умножении.

6) Выведите с первого по четвертое значение массива, полученного в п.5.

7) Умножьте второй символ массива из п.6 на 7.

8) Из массива п.6 выведите предпоследний символ с помощью ключевого слова.

Задание 2.

1) Создайте матрицу размерностью 3x4, в которой любой из элементов будет NaN.

2) Найдите в матрице из п.1 два элемента, которые будут меньше десяти.

3) Создайте матрицу, соразмерную матрице из п.1, используя функцию `randi()`.

4) Выполните вычитание и деление (любое) полученных матриц.

5) Выполните конкатенацию матриц из п.1 и п.3.

6) Транспонируйте полученную матрицу.

7) Найдите максимальное значение матрицы из п.6.

8) Выполните переформирование матрицы из п.3, изменив ее размерность на 4x3, используя функцию `reshape()`.

Задание 3.

1) Вычислите выражение, используя в качестве корня возведение в степень $1/2$:

$$x = \frac{\sqrt{\left(\pi^2 - a + \frac{34 - a}{a + 3}\right) \cdot \left(\frac{15}{a} + 7\right)}}{180 + (a^2 \cdot a - 40)}.$$

Переменная a является номером варианта.

2) Вычислите выражение из п.1, используя функцию вычисления квадратного корня. Сравните результаты.

2. Контрольные вопросы к работе:

1. Что такое инвертирование массива и как оно выполняется в языке Octave/MATLAB?

2. Как используется функция `randi()` для создания случайных массивов в Octave/MATLAB?

3. Что такое транспонирование матрицы и как его выполнить в Octave/MATLAB?

4. Как изменить размерность матрицы с использованием функции `reshape()` в Octave/MATLAB?

Практическая работа № 3

«Знакомство с языком программирования Octave/MATLAB. Построение графиков»

Цель практической работы: ознакомление с основами построения анимаций и графиков в языке программирования Octave/MATLAB.

Задачи практической работы:

– математическое моделирование движения брошенного тела;

– освоение операторов графического построения в Octave/MATLAB.

1. Задание на практику

Задача 1.

Выведите анимацию тела, брошенного под углом к горизонту, в режиме реального времени. В качестве исходных параметров используйте скорость движения в начальный момент времени u_0, а также угол наклона a_0. Частота обновления должна быть в пределах 10-30 Гц. На графике необходимо выводить только положение тела, траекторию строить не нужно. Для выполнения практического задания можно воспользоваться математическими выражениями, представленными ниже.

– время полета:

$$t = \frac{2u_0 \sin a_0}{g}.$$

– проекция скорости тела:

$$u_x(t) = u_0 \cos a_0 ;$$

$$u_y(t) = u_0 \sin a_0 - gt .$$

– дальность полета:

$$L = \frac{u_0^2 \sin 2a_0}{g} .$$

– максимальная высота полета:

$$H_{max} = \frac{u_0^2 \sin^2 a}{2g} .$$

– уравнение движения тела:

$$x = u_x * t ;$$

$$y = u_0 \sin a_0 * t - \frac{gt^2}{2} .$$

Пример построения графика представлен на рисунке 1.1.

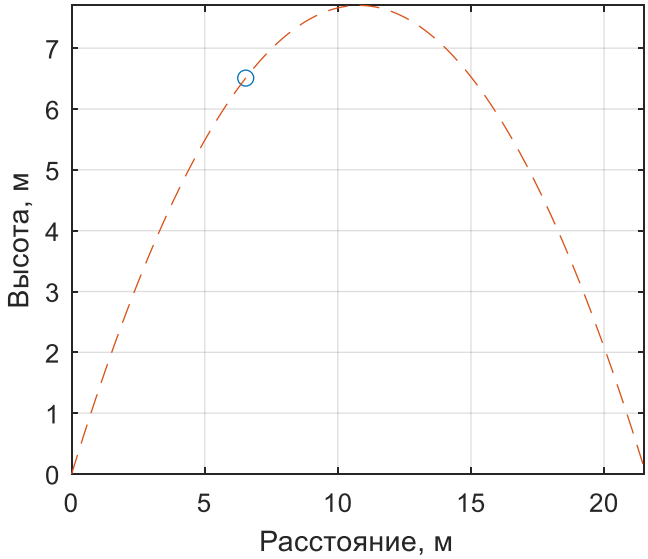

Рисунок 1.1 – Пример построения графика

Пример построения графика:

```
figure
p = plot (x,y,'o','Color','r'); # построение круглыми точками
                                          # красного цвета
axis equal # выравнивание масштаба по осям
axis ([0 30 0 30]) # пределы осей
grid on # включение сетки
xlabel ('Расстояние, м') # подпись оси x
ylabel ('Высота, м') # подпись оси y
title ('Название графика') # подпись окна
hold on # удержание окна, если требуется построение
             # следующих линий поверх текущего графика
delete (p) # удаление линии
```

Команда «`figure`» необходима для создания нового графического окна. Если уже были выведены другие окна, то пользователь, не прописав данную команду, увидит построение графика в предыдущем окне.

Аргументы «`x`» и «`y`» могут быть как переменными, так и двумерными векторами.

Для выполнения задания также можно использовать следующие функции: `pause (x)` – выдерживание паузы при исполнении программы в секундах; `tic`, `toc` – измерение времени выполнения строк кода программы между командами.

2. Контрольные вопросы к работе

1. Для чего используется функция `plot()`?
2. Как изменить цвет линии на графике в Octave/MATLAB?
3. Каким образом задаётся название осей на графиках в Octave/MATLAB?

Практическая работа № 4

«Ключевые операции ЦОС. Корреляция, линейная и циклическая свёртка»

Цель практической работы: получить навыки расчёта корреляции и свёртки.

Задачи практической работы:

– расчет корреляции и свертки «вручную»;

– расчет корреляции и свертки в Octave/MATLAB.

1. Задание на практику

Задание 1. Расчет корреляции и свертки «вручную».

1. Рассчитайте корреляцию и свертку (линейную и циклическую) следующих последовательностей: a = [2 -4 1 3 8] и b.

2. Рассчитайте свертку (линейную и циклическую) и корреляцию следующих последовательностей: c = [4 -3 5] и d.

Значения последовательностей b и d представлены в таблице 1.1.

Задание 2. Расчет корреляции и свертки в Octave/MATLAB.

1. Рассчитайте свертку (линейную и циклическую) и корреляцию заданных последовательностей в Octave/MATLAB. Сравните результат с расчетами из з.1. Постройте результат свертки и взаимнокорреляционную функцию (ВКФ).

2. Рассчитайте автокорреляционную функцию (АКФ) заданных последовательностей в Octave/MATLAB. Отобразите результат на графике.

Таблица 1.1 – Перечень вариантов

Вариант	b	d
1	0 -4 -2 -3 -3	-2 0 -3 4 4 0
2	0 -1 5 -1 -3	3 0 -2 0 -3 -2
3	5 5 1 -4 -2	-1 3 -3 -3 -2 2
4	3 2 0 1 -2	2 -2 2 -2 -1 2
5	3 -4 5 3 0	-1 4 4 1 1 1
6	-2 -1 0 -2 4	-2 -2 -2 -2 0 -1
7	5 0 -3 5 5	0 -3 -1 0 1 -1
8	2 3 -2 -3 -2	-1 0 1 -3 -1 3
9	-4 5 3 0 1	-2 0 4 1 1 -2
10	-2 -1 2 -3 3	-3 2 0 3 2 4
11	4 -1 2 -3 -4	2 1 0 4 1 1
12	4 4 1 -3 -2	4 -3 0 -2 4 2
13	1 0 -4 2 -4	-3 1 -3 3 3 2
14	-3 2 1 5 2	3 0 0 3 -3 -2
15	-4 3 -2 0 1	4 0 4 -1 2 2
16	1 2 2 -3 -3	4 -2 -3 1 4 2
17	-3 1 -2 -1 1	-1 -1 1 -1 3 4
18	3 -1 1 -3 5	1 0 2 2 2 -3
19	2 3 0 -4 -2	-2 -1 0 1 0 4
20	1 5 2 5 -2	2 -1 2 2 -3 -1

2. Контрольные вопросы

1. В чём отличие линейной и циклической свёртки?
2. Что такое корреляция?
3. Чем отличается АКФ от ВКФ?

Практическая работа № 5

«Ключевые операции ЦОС. Двумерная свёртка. Работа с изображениями»

Цель практической работы:
Задачи практической работы:
– расчёт линейной и циклической свёртки;
– расчёт ВКФ и АКФ;
– двумерная свёртка. Обработка изображений.

1. Задание на практику

Задание 1.

Рассчитайте корреляцию и свертку (линейную и циклическую) следующих последовательностей: a = [-3 2 1 -3 8] и b.

Рассчитайте свертку (линейную и циклическую) и корреляцию следующих последовательностей: c = [2 5 3] и d.

Значения последовательностей b и d представлены в таблице 1.1.

Постройте результат свертки и ВКФ. Сравните результаты, полученные в пунктах 1 и 2 с графическим расчетом.

Рассчитайте АКФ последовательностей a и b. Отобразите результат на графике. Сформируйте вектор в виде повторяющихся значений массивов: a b a c a b a. Рассчитайте АКФ полученной последовательности и постройте результат на графике.

Таблица 1.1 – Перечень вариантов

Вариант	b	d
1	0 -4 -2 -3 -3	-2 0 -3 4 4 0
2	0 -1 5 -1 -3	3 0 -2 0 -3 -2
3	5 5 1 -4 -2	-1 3 -3 -3 -2 2
4	3 2 0 1 -2	2 -2 2 -2 -1 2
5	3 -4 5 3 0	-1 4 4 1 1 1
6	-2 -1 0 -2 4	-2 -2 -2 -2 0 -1
7	5 0 -3 5 5	0 -3 -1 0 1 -1
8	2 3 -2 -3 -2	-1 0 1 -3 -1 3
9	-4 5 3 0 1	-2 0 4 1 1 -2
10	-2 -1 2 -3 3	-3 2 0 3 2 4
11	4 -1 2 -3 -4	2 1 0 4 1 1
12	4 4 1 -3 -2	4 -3 0 -2 4 2
13	1 0 -4 2 -4	-3 1 -3 3 3 2
14	-3 2 1 5 2	3 0 0 3 -3 -2
15	-4 3 -2 0 1	4 0 4 -1 2 2
16	1 2 2 -3 -3	4 -2 -3 1 4 2
17	-3 1 -2 -1 1	-1 -1 1 -1 3 4
18	3 -1 1 -3 5	1 0 2 2 2 -3
19	2 3 0 -4 -2	-2 -1 0 1 0 4
20	1 5 2 5 -2	2 -1 2 2 -3 -1

Задание 2.

1. Загрузите любое изображение на свой выбор размером не менее 1024x1024 пикселей в среду Octave/MATLAB. Проведите обработку с помощью двумерной свертки. Ядра для обработки изображения представлены в таблице 1.2. Представьте графические результаты.

2. Найдите способ изменить тональность изображения (в холодную/теплую сторону). Продемонстрируйте его в действии, отобразив изображение до и после обработки.

Таблица 1.2 – Ядра для обработки изображения

Операция	Ядро
Оригинальное отображение	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
Выделение границ	$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$
Повышение резкости	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$
Размытие по Гауссу (3 x 3)	$\frac{1}{9} \cdot \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$
Тиснение	$\begin{bmatrix} -2 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix}$

2. Контрольные вопросы

1. Дайте определения АКФ.
2. Дайте определения процедуре свёртки сигнала.
3. Что такое ядро свёртки?
4. С какой целью используется двумерная свёртка?

Практическая работа № 6

«Прямое дискретное преобразование Фурье»

Цель практической работы: приобретение навыков расчёта ДПФ-образа сигнала.
Задачи практической работы:
– рассчитать значение прямого ДПФ-образа заданной последовательности;
– рассчитать значение обратного ДПФ-образа заданной последовательности.

1. Пример расчета прямого ДПФ

Необходимо рассчитать ДПФ следующей дискретной числовой последовательности:

$$x(nT) = [1, 0, 0, 1].$$

Для большей ясности, тоже самое мы можем записать следующим образом:

$$x(0) = 1, \qquad x(T) = 0, \qquad x(2T) = 0, \qquad x(3T) = 1.$$

Прямое дискретное преобразование Фурье рассчитывается согласно следующей формуле:

$$X(k) = \sum_{n=0}^{N-1} x(nT) e^{-jk\omega nT}, k = 0, 1, ..., N-1.$$

Таким образом ДПФ для заданной последовательности при $k = 0$ рассчитывается следующим образом:

$$X(0) = \sum_{n=0}^{3} x(nT) e^{-j0} = \sum_{n=0}^{3} x(nT) = x(0) + x(T) + x(2T) + x(3T) =$$

$$= 1 + 0 + 0 + 1 = 2.$$

При $k = 1$:

$$X(1) = \sum_{n=0}^{3} x(nT) e^{-j\omega nT}.$$

В данном случае степень экспоненциальной компоненты не будет обнуляться для всех слагаемых, при этом период дискретизации T нам не известен, однако известно, что циклическая частота обратно пропорцианальна частоте дискретизации согласно следующей формуле:

$$\omega = \frac{2\pi}{NT}.$$

Выполним замену и рассчитаем X для $k = 1$:

$$X(1) = \sum_{n=0}^{3} x(nT) e^{-j2\pi n/N} = 1 + 0 + 0 + 1 e^{-j2\pi 3/4} = 1 + e^{-j3\pi/2} =$$

$$= 1 + \cos(\frac{3\pi}{2}) - j\sin(\frac{3\pi}{2}) = 1 + j.$$

При $k = 2$:

$$X(2) = \sum_{n=0}^{3} x(nT) e^{-j2\omega nT} =$$

$$= \sum_{n=0}^{3} x(nT) e^{-j2\pi 2n/N} = \sum_{n=0}^{3} x(nT) e^{-j4\pi n/N} = 1 + 0 + 0 + 1 e^{-j4\pi 3/4} = 1 + e^{-j3\pi} =$$

$$= 1 - 1 = 0.$$

При $k = 3$:

$$X(3)=\sum_{n=0}^{3}x(nT)e^{-j2\omega 3n/T}=\sum_{n=0}^{3}x(nT)e^{-j2\pi 2n/N}=1+0+0+1e^{-j9\pi/2}=1+i\,.$$

Таким образом результат прямого преобразования Фурье

$$X(k)=[2,\ 1+j,\ 0,\ 1-j]\,.$$

2. Пример обратного ДПФ

Для иллюстрации обратного дискретного преобразования Фурье выполним расчёт временного ряда из уже известного нам ДПФ-образа:

$$X(k)=[2,\ 1+j,\ 0,\ 1-j]\,.$$

При $n=0$:

$$x(nT)=x(0)=\frac{1}{N}\sum_{k=0}^{N-1}X(k)=\frac{1}{4}[X(0)+X(1)+X(2)+X(3)]=$$

$$=\frac{1}{4}[2+(1+j)+0+(1-j)]=1\,.$$

При $n=1$:

$$x(T)=\frac{1}{N}\sum_{k=0}^{N-1}X(k)e^{jk\omega T}=\frac{1}{N}\sum_{k=0}^{N-1}X(k)e^{jk2\pi/N}=$$

$$=\frac{1}{4}\sum_{k=0}^{3}X(k)e^{jk\pi/2}=\frac{1}{4}[2+(1+j)e^{j\pi/2}+0+(1-j)e^{j3\pi/2}]=$$

$$=\frac{1}{4}[2+(1+j)j+0+(1-j)(-j)]=\frac{1}{4}[2+j-1-j-1]=0\,.$$

При $n=2$:

$$x(2T)=\frac{1}{N}\sum_{k=0}^{N-1}X(k)e^{jk2\pi/N}=\frac{1}{4}\sum_{k=0}^{3}X(k)e^{jk\pi}=$$

$$=\frac{1}{4}[2+(1+j)e^{j\pi}+(1-j)e^{j3\pi}]=\frac{1}{4}[2-(1+j)-(1-j)]=0\,.$$

При $n=3$:

$$x(3T)=\frac{1}{N}\sum_{k=0}^{N-1}X(k)e^{jk2\pi/N}=\frac{1}{4}\sum_{k=0}^{3}X(k)e^{jk3\pi/2}=$$

$$=\frac{1}{4}[2+(1+j)e^{j3\pi/2}+(1-j)e^{j9\pi/2}]=\frac{1}{4}[2+(1+j)(-j)+(1-j)j]=$$

$$=\frac{1}{4}[2-j+1+j+1]=1.$$

Таким образом результат вычисления обратного ДПФ равен:

$$x(nT)=[1,0,0,1]\,.$$

3. Задание на практику

Задание 1.

Рассчитайте прямой ДПФ-образ для следующих выражений:

Таблица 3.1 – Варианты последовательностей

Вариант	Последовательность
1	[1 0 1 0 1 0]
2	[1 -1 0 0 0 -1]
3	[0 1 1 1 0 0]
4	[1 0 -1 0 0 -1]
5	[1 0 0 2 0 -1]
6	[1 2 0 0 1 0]
7	[1 1 0 0 0 1]
8	[2 0 0 -1 1 0]
9	[1 0 0 1 1 0]
10	[2 0 0 1 0 -2]

Задания 2.

Рассчитайте обратное ДПФ для следующих выражений:

Таблица 3.2 – Варианты последовательностей

Вариант	Последовательность
1	[3, –1, 3, –1]
2	[0, 1–j, 2, 1+j]
3	[3, –j, 5, j]
4	[1, 1–2j, 1, 1+2j]
5	[0, 1+5j, 2, 1–5j]
6	[1, 3–2j, 5, 3+2j]
7	[0, 3+3j, 6, 3–3j]
8	[–1, –3j, 5, 3j]
9	[3, j, 1, –j]
10	[3, 2+j, 1, 2–j]

4. Контрольные вопросы

1. Для чего используется прямое ДПФ преобразование?
2. Для чего используется обратное ДПФ преобразование?

Практическая работа № 7

«Прямое дискретное быстрое преобразование Фурье»

Цель практической работы: приобретение навыков расчёта быстрого БПФ-образа сигнала.

Задачи практической работы:

– рассчитать значение прямого быстрого БПФ-образа заданной последовательности.

1. Пример расчета быстрого ДПФ

Необходимо рассчитать БПФ следующей дискретной числовой последовательности:

$$x(nT) = [1, 0, 0, 1].$$

Для большей ясности тоже самое мы можем записать следующим образом:

$$x(0) = 1, \qquad x(T) = 0, \qquad x(2T) = 0, \quad x(3T) = 1.$$

Для расчёта четырёхточечного БПФ алгоритм «бабочка» будет выглядеть так, как это показано на рисунке 1.1.

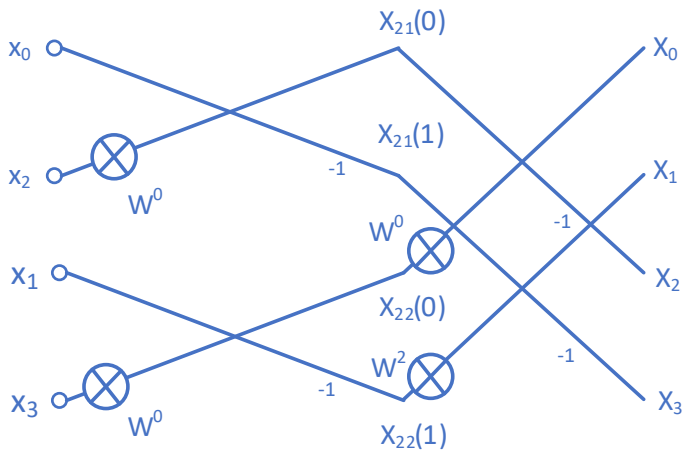

Рисунок 1.1 – Алгоритм для четырехточечного БПФ

Тогда согласно представленному алгоритму, выполним расчет:

$$X_{21}(0) = x_0 + x_2 = 1,$$

$$X_{21}(1) = x_0 - x_2 = 1,$$

$$X_{22}(0) = x_1 + x_3 = 1,$$

$$X_{22}(1) = x_1 - x_3 = -1,$$

$$X_{11}(0) = X_{21}(0) + W_4^0 X_{22}(0) = 1 + 1 = 2,$$

$$X_{11}(1) = X_{21}(1) + W_4^2 X_{22}(1) = 1 + e^{-\pi/2}(-1) = 1 + j,$$

$$X_{11}(2) = X_{21}(0) - W_4^0 X_{22}(0) = 1 - 1 = 0,$$

$$X_{11}(3) = X_{21}(1) - W_4^2 X_{22}(1) = 1 - j.$$

2. Задание на практику

Задание 1.

Рассчитайте прямой БПФ-образ для следующих выражений:

Таблица 2.1 – Варианты последовательностей

Вариант	Последовательность
1	[1 0 1 0 1 0]
2	[1 -1 0 0 0 -1]
3	[0 1 1 1 0 0]
4	[1 0 -1 0 0 -1]
5	[1 0 0 2 0 -1]
6	[1 2 0 0 1 0]
7	[1 1 0 0 0 1]
8	[2 0 0 -1 1 0]
9	[1 0 0 1 1 0]
10	[2 0 0 1 0 -2]

3. Контрольные вопросы к работе

1. В чём отличие ДПФ преобразования от БПФ преобразования?
2. Что такое алгоритм «бабочка»?
3. Согласно какой формуле определяется вычислительная сложность БПФ?

Практическая работа № 8

«Расчет коэффициентов КИХ-фильтром методом взвешивания»

Цель практической работы: освоение методов расчета коэффициентов КИХ-фильтров методом взвешивания, а также применение оконных функций для получения желаемых частотных характеристик фильтра.

Задачи практической работы:

– ознакомиться с процессом расчета коэффициентов КИХ-фильтров с использованием метода взвешивания;

– изучить и применить оконные функции (например, Хэмминга, Блэкмена, Кайзера) для получения оптимальных характеристик фильтра;

– научиться рассчитывать параметры фильтра, включая количество коэффициентов, используя заданные частотные характеристики и спецификации;

– рассчитать коэффициенты КИХ-фильтров для различных типов фильтров (например, фильтры низких, высоких, полосовых и заграждающих частот) в соответствии с вариантом.

1. Вводный материал

Расчёт коэффициентов КИХ-фильтров методом взвешивания включает следующие этапы:

Этап 1. Задать «идеальную» или желаемую частотную характеристику фильтра $H_D(\omega)$.

Этап 2. Получить импульсную характеристику (ИХ) $h_D(n)$ желаемого фильтра.

Этап 3. Выбрать весовую функцию, которая удовлетворяет требованиям к полосе пропускания или затуханию, а затем определить число коэффициентов фильтра, использовав выражение для связи длины фильтра с шириной перехода.

Этап 4. Получить значения выбранной весовой функции $w(n)$ и значения коэффициентов реального КИХ-фильтра $h(n)$.

В зависимости от типа КИХ фильтра по частотной избирательности идеальная частотная характеристика будет иметь математическое описание согласно таблице 1.1.

Таблица 1.1 – Примеры идеальных ИХ

	Идеальная импульсная характеристика, $h_D(n)$	
Тип фильтра	$h_D(n), n \neq 0$	$h_D(0)$
Фильтр нижних частот	$2f_c \dfrac{\sin(n\omega_c)}{n\omega_c}$	$2f_c$
Фильтр верхних частот	$-2f_c \dfrac{\sin(n\omega_c)}{n\omega_c}$	$1-2f_c$
Полосовой фильтр	$2f_2 \dfrac{\sin(n\omega_2)}{n\omega_2} - 2f_1 \dfrac{\sin(n\omega_1)}{n\omega_1}$	$2(f_2 - f_1)$
Заграждающий фильтр	$2f_1 \dfrac{\sin(n\omega_1)}{n\omega_1} - 2f_2 \dfrac{\sin(n\omega_2)}{n\omega_2}$	$1-2(f_2 - f_1)$

Итоговые коэффициенты фильтра представляют собой результат перемножение идеальной импульсной характеристики с весовой функцией:

$$h(n) = h_D(n)w(n). \qquad (1.1)$$

Весовая функция для некоторых окон представлена в таблице 1.2.

Таблица 1.2 – Оконные функции

Функция	Ширина перехода (нормированная) (Гц)	Неравномерность в полосе пропускания (дБ)	Главный лепесток относительно бокового лепестка	Затухание в полосе подавления	Формула
Прямоугольная	$0,9/N$	0,7416	13	21	1
Хеннинга	$3,1/N$	0,0546	31	44	$0,5+0,5\cos\left(\frac{2\pi n}{N}\right)$
Хэмминга	$3,3/N$	0,0194	41	53	$0,54+0,46\cos\left(\frac{2\pi n}{N}\right)$
Блэкмена	$5,5/N$	0,0017	57	75	$0,42+0,5\cos\left(\frac{2\pi n}{N-1}\right)+0,8\cos\left(\frac{4\pi n}{N-1}\right)$
Кайзера	$2,93/N$ $(\beta=4,54)$	0,0274	-	50	$\frac{I_0\left(\beta\sqrt{1-\left[\frac{2n}{N-1}\right]^2}\right)}{I_0(\beta)}$
	$4,32/N$ $(\beta=6,76)$	0,00275		70	
	$5,71/N$ $(\beta=8,96)$	0,000275		90	

2. Пример расчета

Необходимо реализовать фильтр согласно спецификации, представленной в таблице 2.1.

Таблица 2.1 – Спецификация фильтра

Тип	Граничные частоты f_c		Неравномерность в полосе пропускания	Ширина полосы перехода Δf	Затухание в полосе подавления	Уровень боковых лепестков	Оконная функция	Частота дискретизации $f_д$
	нижняя	верхняя						
ФНЧ	1,5 кГц	-	-	0,5 кГц	>50 дБ	-	-	8 кГц

Согласно таблице 1.1 зададим идеальную импульсную характеристику следующей зависимостью:

$$h_D(n) = 2f_c\frac{\sin(n\omega_c)}{n\omega_c}, \text{ при } n \neq 0,$$

$$h_D(n) = 2f_c, \text{ при } n = 0.$$

Из таблицы 1.2 следует, что заданной спецификации удовлетворяет оконные функции Хэмминга, Блэкмена или Кайзера. Выберем функция Хэмминга:

$$w(n) = 0,54 + 0,46\cos(2\pi n/N).$$

Для определения необходимого количества коэффициентов N переведём фактическую ширину переходной полосы в нормированному виду:

$$\Delta\tilde{f} = \frac{\Delta f}{f_\partial} = \frac{0,5}{8} = 0,0625 .$$

Выполним расчёт N:

$$N = \frac{3,3}{\Delta\tilde{f}} = \frac{3,3}{0,0625} = 52,8 \approx 53 .$$

Вследствие эффекта смазывания характеристик фильтра, частота среза будет отличаться от представленной в спецификации. Учтём это, добавив к полосе среза половину диапазона переходной полосы и перейдем к нормированным частотам:

$$f_c' = f_c + \Delta f / 2 = 1,5 + 0,25 = 1,75 \to 1,75/8 = 0,21875 .$$

Рассчитаем коэффициенты фильтра.

$n = 0$:

$$h_D(0) = 2 f_c = 2 \cdot 0,21875 = 0,4375 ,$$
$$w(0) = 0,54 + 0,46\cos(0) = 1 ,$$
$$h(0) = h_D(0) w(0) = 0,4375 .$$

$n = 1$:

$$h_D(1) = \frac{2 \cdot 0,21875}{2\pi \cdot 0,21875} \sin(2\pi \cdot 0,21875) = 0,31219 ,$$
$$w(1) = 0,54 + 0,46\cos(2\pi / 53) = 0,99677 ,$$
$$h(1) = h_D(1) w(1) = 0,31118 .$$

$n = 2$:

$$h_D(2) = \frac{2 \cdot 0,21875}{2 \cdot 2\pi \cdot 0,21875} \sin(2 \cdot 2\pi \cdot 0,21875) = 0,06013 ,$$
$$w(2) = 0,54 + 0,46\cos(2\pi \cdot 2 / 53) = 0,98713 ,$$
$$h(2) = h_D(2) w(2) = 0,06012 .$$

Подобный расчёт выполняется для остальных 25 значений половины импульсной характеристики фильтра.

3. Задание на практику

Рассчитайте коэффициенты фильтров согласно параметрам, заданным в таблице 3.1.

Таблица 3.1 – Таблица заданных параметров фильтров

№	Тип	Граничные частоты f_c		Неравномерность в полосе пропускания	Ширина полосы перехода Δf	Затухание в полосе подавления	Уровень боковых лепестков	Оконная функция	Частота дискретизации f_∂
		нижняя	верхняя						
1	ПФ	150 Гц	250 Гц	<0,1 дБ	150 Гц	>60 дБ		Кайзера	1 кГц
2	ПФ	150 Гц	250 Гц	<0,1 дБ	200 Гц	>60 дБ		Блэкмена	1 кГц
3	ФНЧ	4 МГЦ	-	-	1,7 МГц	>50 дБ	-	-	20 МГц
4	ФНЧ	3 кГц	-	-	1,2 кГц	>50 дБ	-	-	12 кГц
5	ФНЧ	80 кГц	-	-	30 кГц	>70	-	-	250 кГц
6	ФНЧ	67 кГц	-	-	33 кГц	-	-	Хэмминга	200 кГц
7	ФВЧ	10 МГц	-	-	7,33 МГц	-	-	Блэкмена	30 МГц
8	ФВЧ	33 кГц	-	-	15 кГц	>40	-	-	100 кГц
9	ФВЧ	500 Гц	-	-	250 Гц	>60	-	-	1,5 кГц
10	ФВЧ	1 МГц	-	-	500 кГц	>75	-	-	3 МГц
11	ЗФ	3 кГц	6 кГц	-	2 кГц	-	-	-	15 кГц
12	ЗФ	0,2 МГц	0,5 МГц	-	0,3 МГц	-	-	-	1,5 МГц

4. Контрольные вопросы

1. Для чего используется метод взвешивания при расчете коэффициентов КИХ-фильтров?
2. Чем отличаются оконные функции Хэмминга, Блэкмена и Кайзера в контексте проектирования цифровых фильтров?
3. Возможно ли получить точные частотные характеристики фильтра при заданных спецификациях, и как оконная функция влияет на эти характеристики?

Практическая работа № 9

«Цифровая манипуляция»

Цель практической работы: формирование навыков работы с цифровым манипулированным сигналом.

Задачи практической работы:

– сформировать цифровой манипулированный сигнал;

– добавить шум, построить созвездие;

– провести декодирование сигнала и оценить коэффициент битовых ошибок.

1. Вводный материал

В данной работе рассматривается процесс формирования битового потока и кодирование его в синфазную и квадратурную составляющую. Этапы выполнения данной работы показаны на рисунке 1.1.

Рисунок 1.1 – Этапы выполнения работы

2. Квадратурная амплитудная манипуляция

Цифровая манипуляция осуществляется с помощью кодера синфазной и квадратурной составляющей сигнала. Этот процесс подразумевает преобразование битового сигнала в уровни модуляции, число которых зависит от порядка манипуляции (M). Порядок задается кодером синфазной и квадратурной составляющей. Порядок манипуляции соответствует количеству информационных бит в символе манипуляции, поэтому на выходе кодера количество символов сокращается в M раз. В случае BPSK один символ переносит один информационный бит, в QPSK – 2, в 8PSK – 3, в QAM-16 – 4. Например, сигнал QPSK (или QAM-4, созвездие которого представлено на рисунке 2.1) имеет 2 уровня амплитуды по двум

координатным осям, соответствующим квадратурам I и Q. Независимо от порядка манипуляции после кодера применяется одна и та же схема квадратурного модулятора.

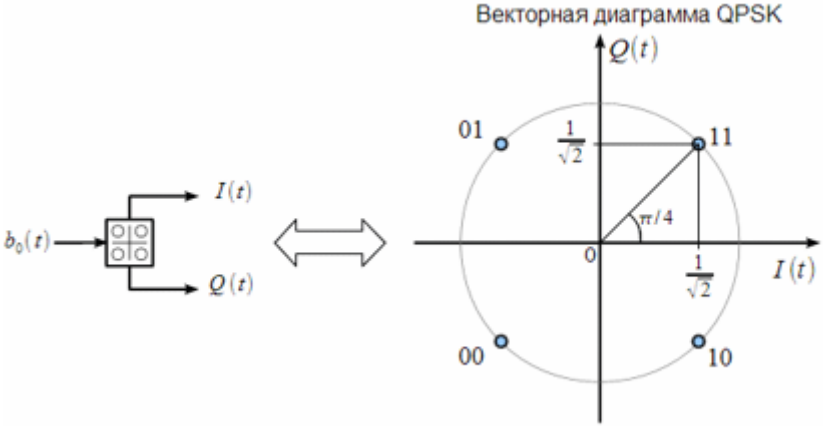

Рисунок 2.1 – Кодирование синфазной и квадратурной составляющей

3. Ход выполнения работы

1) В начале работы требуется задать исходные параметры. Количество передаваемых бит N = 12000, порядок манипуляции – M. Рассматриваемые манипуляции: QPSK, QAM16, QAM64.

Битовый поток моделируется функцией `randi()`. Требуется задать матрицу размерностью 1xN, диапазон значений от 0 до 1. Поскольку кодер I/Q принимает на вход числовые данные от 0 до M, битовый поток необходимо преобразовать в соответствующие уровни с помощью функции `bi2de()` (преобразователь двоичных чисел в десятичные). Строка преобразуется в два столбца, затем каждая пара битов одной строки преобразуется в десятичное число. В случае манипуляции с индексом 4 входные биты будут преобразованы в 512 уровней, то есть в log2(M) раз меньше. Опция «`left-msb`» (младший значащий бит расположен слева) включает привычное представление двоичных сигналов по столбцам, то есть «10» соответствует значению 2, а «01» – 1.

```
%% Формирование битового потока
bits = randi(??);


% преобразование битового потока в символы (от 0 до M-1)
sym = bi2de(reshape(bits,log2(M),…
length(bits)/log2(M)).','left-msb');
```

2) Далее полученные данные подаются на кодер I/Q. Кодер сопоставляет каждому входному десятичному значению точку в координатной комплексной плоскости.

```
%% Кодирование синфазной и квадратурной составляющих
sqam = qammod(sym, M);
```

3) После этого с помощью функции `awgn()` в канале добавляется аддитивный белый гауссовский шум. Для построения созвездия используется функция `scatter()`, на вход которой подается два массива – реальная и мнимая часть полученных символов. Параметр «`filled`» включает закрашивание точек на графике, поскольку по умолчанию они

отображаются в виде пустых кружков. В данном пункте сформируйте 3 массива с отношением сигнал/шум 20, 15 и 10 дБ.

```
% Построение созвездия
subplot(2,2,1)
scatter(real(sqam), imag(sqam), 'filled')
xlabel('Квадратура I')
ylabel('Квадратура Q')
title('Сигнал без шума')
subplot(2,2,2)…
```

4) На приемной стороне обработка происходит в обратном порядке. Подайте на вход декодера один из сформированных ранее массивов. Сначала символы демодулируются функцией `qamdemod()`, на входе которой указывается входной сигнал и порядок манипуляции. Затем полученные данные подаются на преобразователь десятичных чисел в двоичные. Последним этапом остается выстраивание полученных битовых значений в одну строку.

```
%% Декодер синфазной и квадратурной составляющей
sym_rx = qamdemod(??, ??);
% преобразование символов (от 0 до M-1) в битовый поток
bits_rx = de2bi(??,'left-msb');
bits_rx = reshape(bits_rx.',1,prod(size(bits_rx)));
```

5) Теперь можно оценивать коэффициент ошибок. Запишите выражение для его расчета последующей формуле:

$$BER = \frac{N_{err}}{N}, \tag{3.1}$$

где N_{err} – количество ошибочно принятых бит;

N – общее количество бит.

N_{err} можно вычислить с помощью функции biterr(), на входе которой указывается два массива (битовый поток на входе и выходе системы).

Далее измените порядок манипуляции на QAM-16 и QAM-64, постройте графики созвездия и оцените коэффициент ошибок.

4. Контрольные вопросы

1. Что такое порядок модуляции?
2. Как помехоустойчивость системы зависит от используемого порядка модуляции?
3. Для чего используется функция `awgn()`?

Лабораторная работа № 1

«Начало работы с Octave/MATLAB»

Цель лабораторной работы: научиться создавать программы построения графиков функций и сигналов (от простых к более сложным).

Задачи лабораторной работы:

– изучить основные функции и команды построения графиков Octave/MATLAB;

– построить графики заданных функций;

– построить графики гармонических сигналов и их спектров.

1. Построение графиков функций

Пример 1. Построить график косинусоиды y=cos(x), $0 \le x \le 2\pi$.

Ниже даны пояснения, комментарии и код программы.

```
      % - это символ комментария.
      %Замечание: для разделения целого и десятичной дроби %используется не запятая, а точка.
      x = 0:0.01:2*pi; %pi – число п; x определяется в виде %массива точек равномерного разбиения сегмента %[0;2*pi] с %шагом 0.01(или вектора с компонентами %0,0.01,0.02,…,6.28). В конце команды ставим «;», %иначе произойдет вывод в командном окне всех %компонент вектора x
      plot (x,cos(x))%строим простой график
```

Данную программу можно набрать в командном окне и запустить выполнение нажатием клавиши Enter. Появится графическое окно с графиком косинусоиды. После написания скрипта его можно запустить нажатием кнопки F5 (или кнопкой «Сохранить и выполнить/продолжить» на панели инструментов). Это окно удобно использовать для сохранения программы в виде отдельного файла с каким-либо именем и расширением .m. Чтобы перед запуском написанного кода автоматически удалять все открытые графические окна и старые переменные, необходимо в начале кода написать строки:

```
clc %очищаем командную строку
clear all %очищаем переменные
close all %закрываем все графические окна
```

Пример 2. Построить графики функций y=cos(x) и y=sin(2x) на одном рисунке.

```
clc
clear all
close all
x = 0:0.01:2*pi;
figure %создаем новое графическое окно
hold on %удерживаем окно для построения нескольких %линий
plot(x,cos(x))
plot(x,sin(2*x))
```

Пример 3. Формирование графического окна.

```
      clc
      close all
      clear all
      set(0,'DefaultAxesFontSize',14,...
      'DefaultAxesFontName','Times New Roman'); %управление %шрифтом по
умолчанию
      x = 0:0.01:2*pi;
      figure('Position',[100 100 360 240]) %создание %графического окна
с параметрами начала координат, %высотой и шириной окна
      hold on
      plot(x,cos(x))
      plot(x,sin(2*x))
      grid on %включение сетки
      title('Временная форма сигнала') %подпись рисунка
      xlabel('Время, с') %подпись оси x
      ylabel('Амплитуда, В') %подпись оси y
      xlim([0 3]) %ограничение отображения по оси x
      ylim([-1.5 1.5]) %ограничение отображения по оси y
      legend('Косинус','Синус') %добавление легенды
```

Индивидуальное задание №1

Построить графики функции $y=y(x)$, если $y_1=|x|$, $y_2=x^3$, $y_3=\mathrm{arctg}(x)$, $y_4=\cos(x^2)$, $y_5=\sin(x+2)$, $y_6=\lg(5-x)$, $-4\le x\le 4$. График каждой функции построить в отдельном окне, при этом если $y=y_1+y_2$, то y_1– первая функция, y_2 – вторая функция, y – третья функция. Использовать шрифт по умолчанию 14 Times New Roman, размер окна 360x240 пикселей. Добавить все необходимые подписи.

Таблица 1.1 – Индивидуальное задание №1 по вариантам

Вариант	1	2	3	4	5	6	7	8	9
Y	y_1+y_2	y_1+y_3	y_1+y_4	y_1+y_5	y_2+y_3	y_2+y_4	y_2+y_5	y_3+y_4	y_3+y_5
Вариант	10	11	12	13	14	15	16	17	18
Y	$y_1\times y_2$	$y_1\times y_3$	$y_1\times y_4$	$y_1\times y_5$	$y_2\times y_3$	$y_2\times y_4$	$y_2\times y_5$	$y_3\times y_4$	$y_3\times y_5$
Вариант	19	20	21	22	23	24	25	26	27
y	y_4+y_5	$y_4\times y_5$	y_1+y_6	y_2+y_6	y_3+y_6	y_4+y_6	y_5+y_6	$y_1\times y_6$	$y_2\times y_6$

2. Построение усложненных графиков функций

Пример 4. Формирование кривой графика функции определенного вида.

```
clc
clear all
close all
x=1:10; %x представлен вектором с компонентами %1,2,3,…,10
figure
plot(x,x.*x,'*c--')
```

Здесь .* – знак поэлементного способа умножения x на x, в результате получаем вектор x^2 с компонентами 1, 4, 9,…, 100. '*c--' – эта запись относится к параметрам plot и задает вид кривой графика. Первый символ записи * внутри кавычек характеризует тип маркера кривой (описание маркеров находится в соответствующей таблице приложения 1). Запись «c» определяет цвет кривой (цвет выбирается в соответствующей таблице приложения 1), символ «– –» задает тип линии.

Пример 5. Задание толщины всех линий на графике.

```
clc
clear all
close all
x = 0:0.01:2*pi;
figure
plot(x,cos(x),'Linewidth',2) %установка толщины линии
```

Индивидуальное задание №2.

Построить графики трех функций в одном окне, выбирая для них оригинальные маркеры, цвета, типы и толщины линий. Функции брать из задания №1 лабораторной работы.

3. Построение гармонических сигналов и их функций

В следующем примере для построения нескольких графических окон в одном используется функция `subplot()`. Первый и второй аргументы указывают размер сетки для построения новых окон. Последний аргумент указывает номер окна, в котором будет построен текущий график. Например, запись `subplot(3, 2, 4)` позволяет построить 6 графических окон (в виде матрицы 3x2, где 3 – количество строк, 2 – количество столбцов) и выделить четвертое из них в качестве текущего. То есть следующее построение графика с помощью инструментов `plot()`, `stem()`, `surf()` или других будет выполнено в четвертом окне из шести.

Пример 6. Представление сигналов и их спектров с помощью функции `subplot()`.

```
clc
clear all
close all
fs = 32e3; %частота дискретизации
T = 0.5; %длительность сигнала
f = 115; %частота гармоники
t = 1/fs:1/fs:T; %отсчеты времени
s = sin(2*pi*f*t); %отсчеты сигнала
%построение временной формы сигнала
figure('Position',[100 100 360 480])
subplot(2,1,1) %построение графика в окне с индексами
%2, 1, 1, где 2 - количество окон по строкам, 1 -
%количество окон по столбцам, 1 - номер текущего окна
plot(t,s)
grid on
xlim([0 0.5])
ylim([-1.5 1.5])
spec = abs(fft(s)); %расчет спектральных составляющих
spec = spec*2/length(spec);
fscale = 0:fs/length(spec):fs-fs/length(spec); %частотная шкала
%построение спектра сигнала
subplot(2,1,2)
```

```
plot(fscale,spec)
grid on
xlim([0 250])
ylim([0 1.5])
```

Индивидуальное задание №3.

Построить графики трех различных гармонических сигналов и их спектров. Исходные данные взять из таблицы. В таблице показаны параметры двух сигналов. Необходимо построить временную и спектральную форму первого, второго и суммарного сигнала. Подписать названия графиков и все оси, привести шрифт к формату из предыдущих частей задания, выбрать подходящие параметры ограничений по осям.

Таблица 3.1 – Индивидуальное задание №3 по вариантам

Вариант	fs, кГц	f, Гц	A	T, c	Вариант	fs	f	T	A
1	16	850, 1950	1 0,8	1,2	16	64	850, 1950	0,5	1,2 0,6
2	64	1150, 1800	0,8 1	0,5	17	64	450, 1950	0,5	1 0,4
3	16	600, 1250	1 0,9	0,8	18	64	950, 2450	0,4	1,2 0,6
4	16	1100, 1550	0,9 1,7	1	19	32	900, 1800	0,8	0,6 1
5	32	850, 1950	0,6 1	0,5	20	16	1100, 1550	1	0,6 1
6	64	700, 2250	1,2 0,6	0,4	21	32	700, 2250	1	0,6 1
7	32	450, 1950	1,2 0,6	1	22	16	750, 2200	1	0,8 1
8	32	650, 2100	0,6 1	0,25	23	64	1100, 1550	1,2	1,4 0.4
9	32	1050, 1900	0,6 1,2	0,6	24	32	800, 1400	0,8	1 0,4
10	64	900, 1800	1 0,9	0,5	25	16	1150, 1800	0,8	0,9 1,7
11	64	800, 1400	1,5 0,6	1,2	26	64	600, 1250	0,5	0,6 1
12	32	500, 1200	1 0,5	0,8	27	32	1150, 1800	1,2	1 0,8
13	16	950, 2450	1,4 0,4	0,8	28	16	650, 2100	0,6	1 0,9
14	64	800, 1650	0,6 1	1	29	64	800, 1650	0,25	1 0,9
15	16	800, 1500	0, 1	1	30	32	950, 2450	1	0,6

4. Контрольные вопросы

1. Какие функции используются в Octave/MATLAB для построения графиков?
2. Как вывести рисунок на экран?
3. Как построить спектр сигнала?
4. Как вывести несколько графических окон в одном?

Лабораторная работа № 2

«Однородное квантование в АЦП»

Цель лабораторной работы: формирование навыков работы с дискретными сигналами в среде Octave/MATLAB.

Задачи лабораторной работы:

– формирование дискретных сигналов;

– построение временной формы сигнала и его спектра;

– квантование дискретных сигналов.

1. Теоретический материал

Для формирования дискретного сигнала в первую очередь необходимо задать вектор с временными отсчетами. Он определяет моменты взятия выборок непрерывного сигнала. Одна из наиболее существенных составляющих ошибки при измерениях с помощью АЦП – погрешность квантования – является результатом самого процесса преобразования. Погрешность квантования – это погрешность, вызванная значением шага квантования и определяемая как 1/2 величины наименьшего значащего разряда (LSB). Она не может быть исключена в АЦП, так как является неотъемлемой частью процесса преобразования, определяется разрешающей способностью АЦП и не меняется от АЦП к АЦП с равным разрешением. Передаточная характеристика АЦП – это функция зависимости кода на выходе АЦП от напряжения на его входе. Такой график представляет собой кусочно-линейную функцию из 2N «ступеней», где N – разрядность АЦП. Каждый горизонтальный отрезок этой функции соответствует одному из значений выходного кода АЦП, как показано на рисунке 1.1. Если соединить линиями начала этих горизонтальных отрезков (на границах перехода от одного значения кода к другому), то идеальная передаточная характеристика будет представлять собой прямую линию, проходящую через начало координат.

Рисунок 1.2 иллюстрирует идеальную передаточную характеристику для 3-х разрядного АЦП с контрольными точками на границах перехода кода. Выходной код принимает наименьшее значение (000b) при значении входного сигнала от 0 до 1/8 полной шкалы (максимального значения кода этого АЦП).

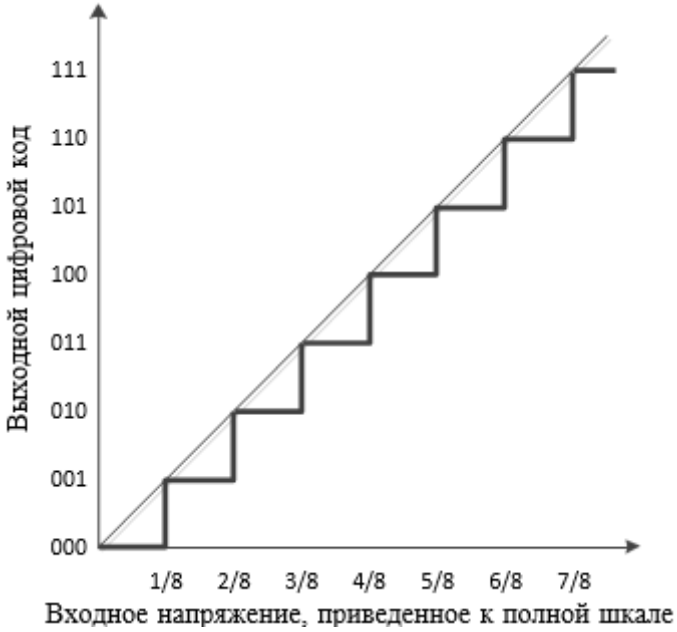

Рисунок 1.1 – Идеальная передаточная характеристика 3-разрядного АЦП

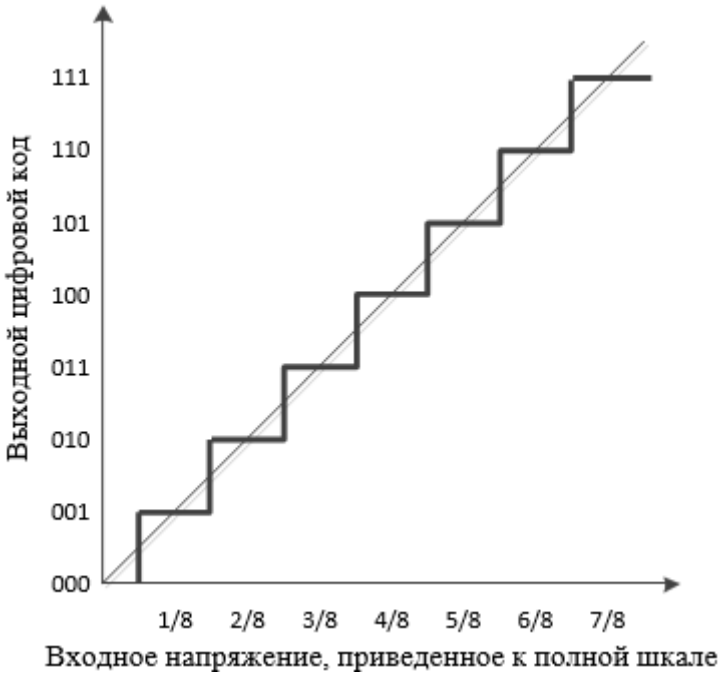

Рисунок 1.2 – Передаточная характеристика 3-разрядного АЦП со смещением на -1/2 LSB для уменьшения погрешности квантования на 1/2 LSB

Также следует отметить, что АЦП достигнет значения кода полной шкалы (111b) при 7/8 полной шкалы, а не при значении полной шкалы. Таким образом, переход в максимальное значение на выходе происходит не при напряжении полной шкалы, а при значении, меньшем на наименьший значащий разряд (Least Significant Bit, LSB), чем входное напряжение полной шкалы. Передаточная характеристика может быть реализована со смещением -1/2 LSB. Это достигается смещением передаточной характеристики влево, что смещает погрешность квантования из диапазона -1... 0 LSB в диапазон -1/2 ... +1/2 LSB.

При квантовании с разрядностью 2 бита со смещением используется 4 уровня, поэтому диапазон амплитуд на входе квантователя можно разделить на 4 зоны. В каждой из них амплитуда сигнала должна устремиться к среднему значению. Данный процесс изображен на рисунке 1.3.

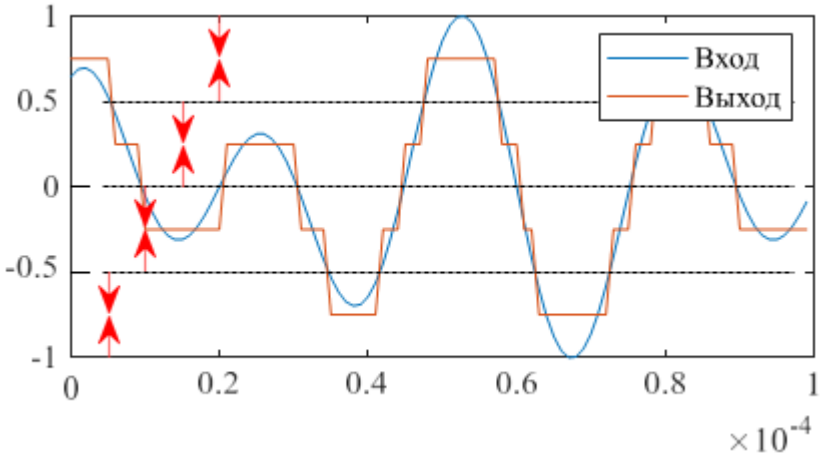

Рисунок 1.3 – Двухбитное квантование сигнала со смещением

Рассмотрим процесс дискретизации на примере формирования гармонического сигнала с линейной частотной модуляцией (ЛЧМ). Ход работы представлен на рисунке 1.4.

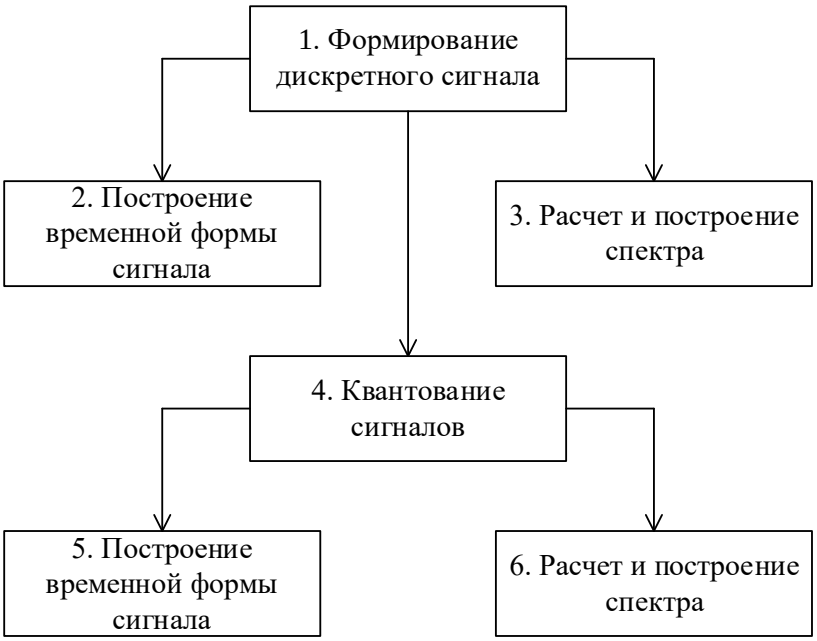

Рисунок 1.4 – Этапы выполнения работы

2. Задание на работу

2.1 Формирование опорных сигналов. Построение временных и спектральных форм

Индивидуальное задание №1.

Сформировать опорный гармонический сигнал $s_1(t)$ с частотой N кГц, где N – номер варианта по списку. Частота дискретизации $f_S = 1$ МГц, длительность символа $T = (f_S/10)/f_S$. Нормировать амплитуду сигнала к пределам [-1 1].

Сформировать опорный ЛЧМ сигнал $s_2(t)$, используя функцию chirp(). Синтаксис функции подразумевает указание четырех основных параметров (t, f_0, T, f_1): t – это временной массив, f_0 – частота в начале символа, T и f_1 – значение времени и частоты в конце символа.

Постройте временные и спектральные формы для двух сигналов. Отобразить все четыре рисунка в одном графическом окне. Подписать названия всех осей и графиков, а также единицы измерения. На осях подобрать пределы для корректного отображения сигналов.

Пример №1.

Чтобы построить спектральное представление сигнала, нужно также рассчитать частотную шкалу. Для отображения амплитуды гармоник в соответствии с временным сигналом (привести шкалу в Вольты) рассчитанные спектральные отсчеты нормируются путем умножения на два и деления на длину массива. Чтобы шкала полученных графиков соответствовала дБВ, от полученных значений необходимо взять $20 \cdot \log 10$.

```
%Расчет спектра
m = 4;
fscale = -fs/2:fs/(length(t)*m):fs/2-fs/(length(t)*m); %частотная шкала
spec1 = fftshift(abs(fft(s1,length(t)*m)));
spec1 = spec1*2/length(t);
```

2.2 Квантование

Пример №2. При квантовании с разрядностью 2 бита используется 4 уровня, поэтому диапазон амплитуд на входе квантователя можно разделить на 4 зоны. В каждой из них амплитуда сигнала должна устремиться к среднему значению.

```
s2 = zeros(1,1024); %нулевой массив
for i = 1:length(s1) %начало цикла
   if s1(i) >= -1 & s1(i) < -0.5
% условие зоны 1 [-1 -0.5]
      s2(i) = -0.75; %выходной сигнал
   elseif s1(i) >= -0.5 & s1(i) < 0
% условие зоны 2 [-0.5 0]
      s2(i) = -0.25;
   elseif s1(i) >= 0 & s1(i) < 0.5
% условие зоны 3 [0 0.25]
      s2(i) = 0.25;
   else % иначе (если не сработали все предыдущие условия)
      s2(i) = 0.75;
   end % конец конструкции if... elseif... end
end % конец цикла
```

Индивидуальное задание №2.

В примере №1 использовался двухбитный квантователь, в котором выполнялась проверка условия каждой из четырех зон. Если данную проверку оформить в цикле, она будет продолжаться во всех четырех зонах, даже если совпадение было найдено в первой. При использовании вложенных циклов для оптимизации расчета следует использовать функцию break. Она завершает текущий цикл (в тексте он выделен серым цветом). Таким образом, проверка условия будет прекращаться после обнаружения необходимого уровня квантования. После вложенного цикла выполняется проверка единичного уровня, поскольку верхняя граница каждой зоны не включается в диапазон, и единичный уровень выпадает из области проверки.

```
s3 = zeros(1, length(s1));
for i = 1:length(s1)
   for k = 1:2^B
      if s1(i) >= -1+2*k/2^B-2/2^B & s1(i) < _____
         s3(i) = _______;
         break
      end
   end
   if s1(i) == 1
      s3(i) = 1-1/2^B;
   end
end
```

Реализуйте квантователь с разрядностью 2, используя код выше, заполняя прочерки соответствующими выражениями. Также реализуйте квантователи с разрядностью 4, 6 и 8 бит.

Постройте временную и спектральную форму квантованных сигналов всех разрядностей (для одной разрядности в одном графическим окне четыре рисунка).

Найдите и постройте сигнал ошибки между исходным и квантованным сигналами для всех разрядностей.

3. Контрольные вопросы

1. Какие функции используются в Octave/MATLAB для построения графиков?
2. Как вывести рисунок на экран?
3. Как построить спектр сигнала?
4. Как вывести несколько графических окон в одном?

Лабораторная работа № 3

«Многоскоростная обработка сигналов»

Цель работы: знакомство с преобразованием частоты дискретизации в среде Octave/MATLAB.

Задачи лабораторной работы:

– децимация цифрового сигнала;

– фильтрация с помощью ФНЧ;

– интерполяция цифрового сигнала.

1. Теоретический материал

Децимацией (прореживанием, сокращением) цифровых данных принято называть уплотнение данных с удалением избыточной информации. Последнее имеет место, если шаг дискретизации данных был установлен с большим запасом, и обработку сигнала можно провести на более низкойчастоте. Информация высокочастотной части сигнала может быть ненужной, если основная энергия полезной части сигнала заключена в низкочастотной области. Децимация может потребоваться и в том случае, если массивы данных представлены с разным шагом дискретизации.

На рисунке 1.1а показан процесс децимации в 2 раза без использования ФНЧ, в результате этой операции в децимированном сигнале проявляются высокочастотные гармоники. Использование ФНЧ позволяет устранить гармоники до децимации, как это показано на рисунке 1.1б. Чем выше коэффициент подавления ФНЧ в полосе заграждения, тем меньше будет уровень искажения от наложения.

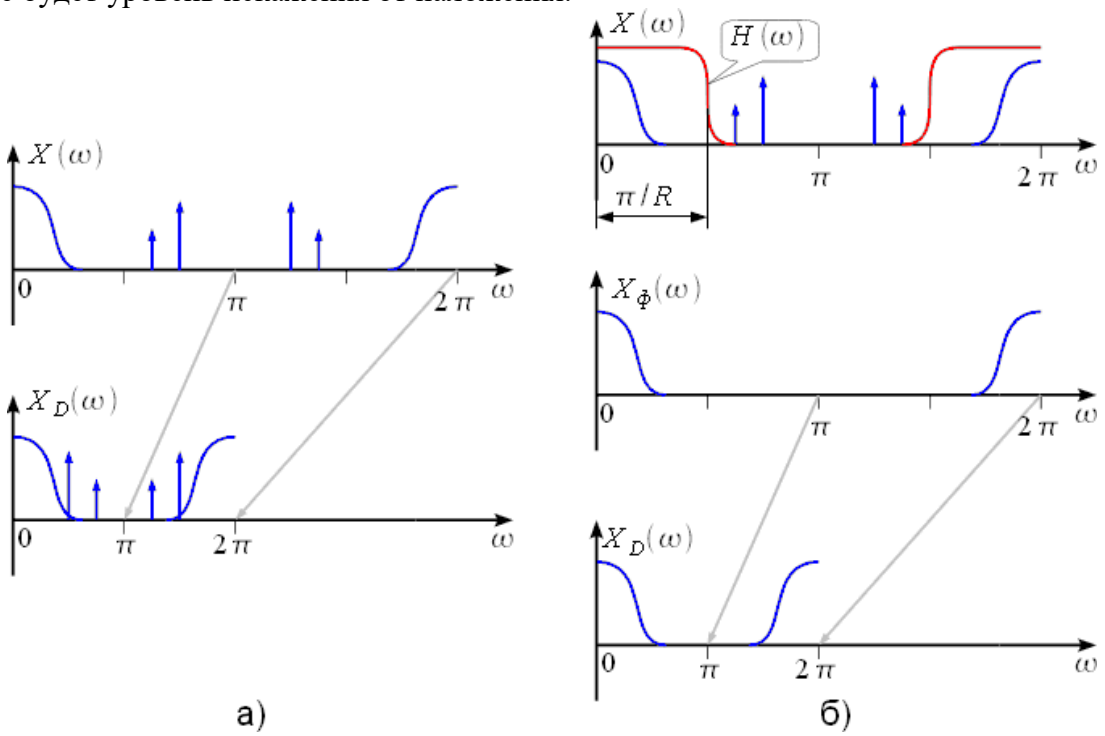

Рисунок 1.1 – Наложение гармоник и их устранение при помощи ФНЧ

Рассмотрим использование фильтров в задаче интерполяции сигналов. Фильтр интерполятор обеспечивает увеличение частоты дискретизации сигнала в R раз, где R – коэффициент интерполяции. Схема фильтра интерполятора и пояснение принципа его работы представлена на рисунке 1.2.

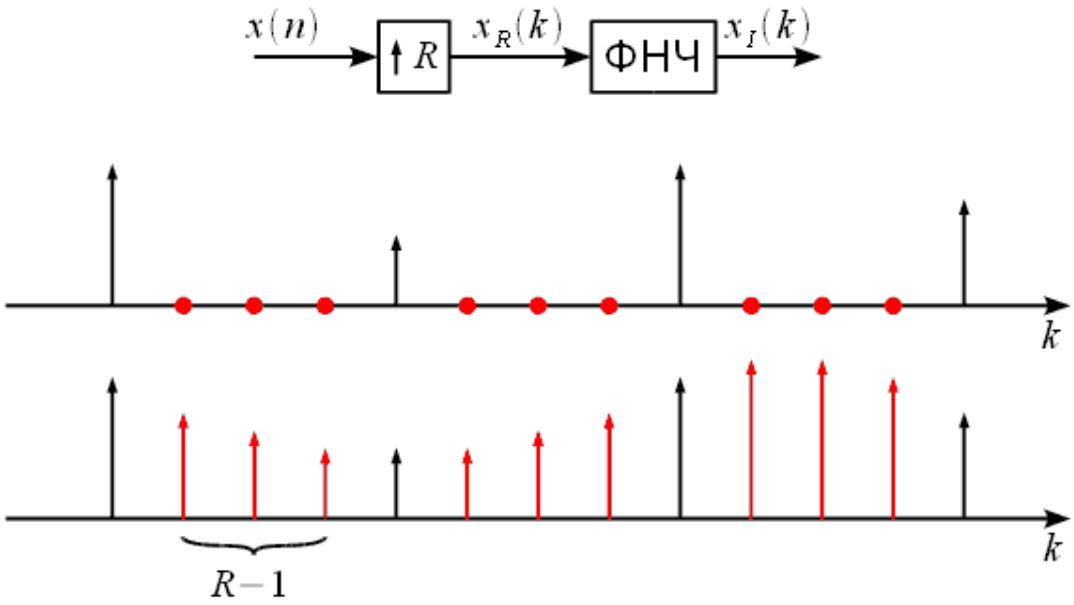

Рисунок 1.2 – Фильтр интерполятор и принцип его работы

Исходный сигнал $x(n)$ (на рисунке 1.2 показан черным) поступает на интерполятор, который увеличивает частоту дискретизации в R раз путем вставки $R-1$ нулей между отсчетами исходного сигнала (на рисунке 1.2 показан красными точками). В результате получается сигнал $x_R(k)$, дополненный нулями, который поступает на ФНЧ. В частотной области данный процесс можно представить так, как показано на рисунке 1.3.

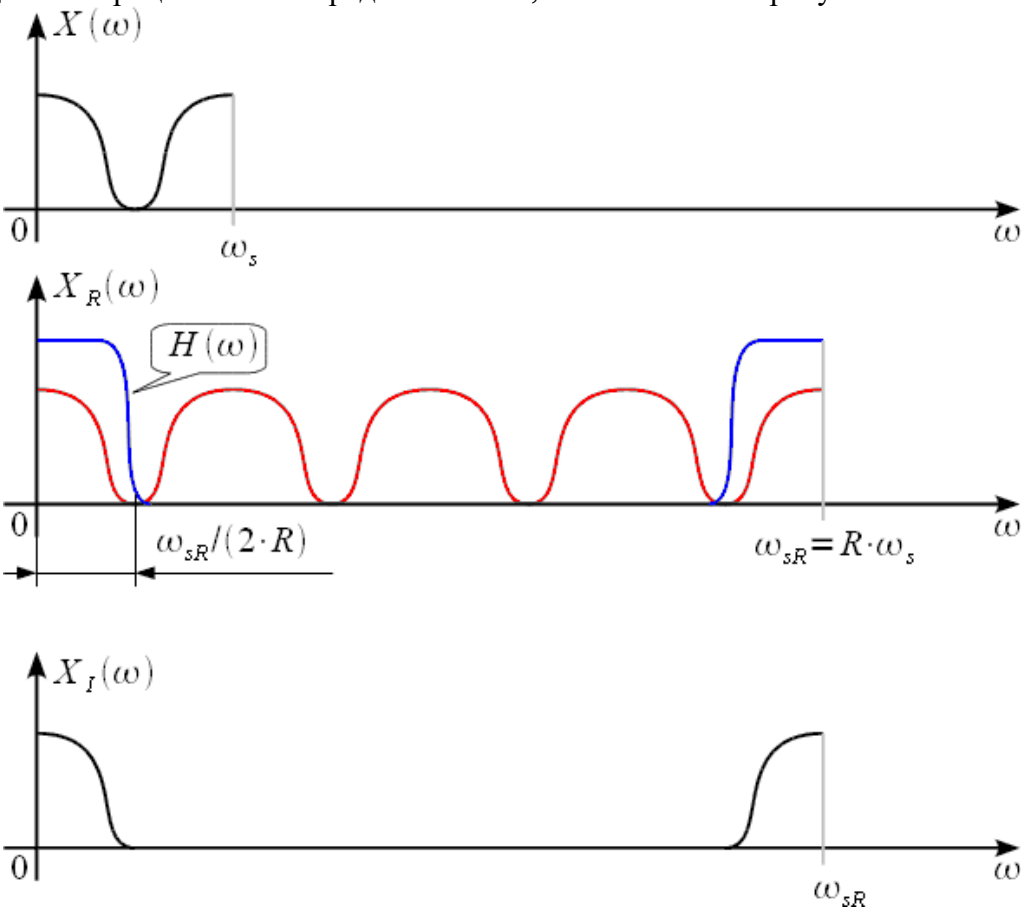

Рисунок 1.3 – Процесс интерполяции в частотной области

2. Задание на работу

На рисунке 2.1 представлен порядок выполнения работы.

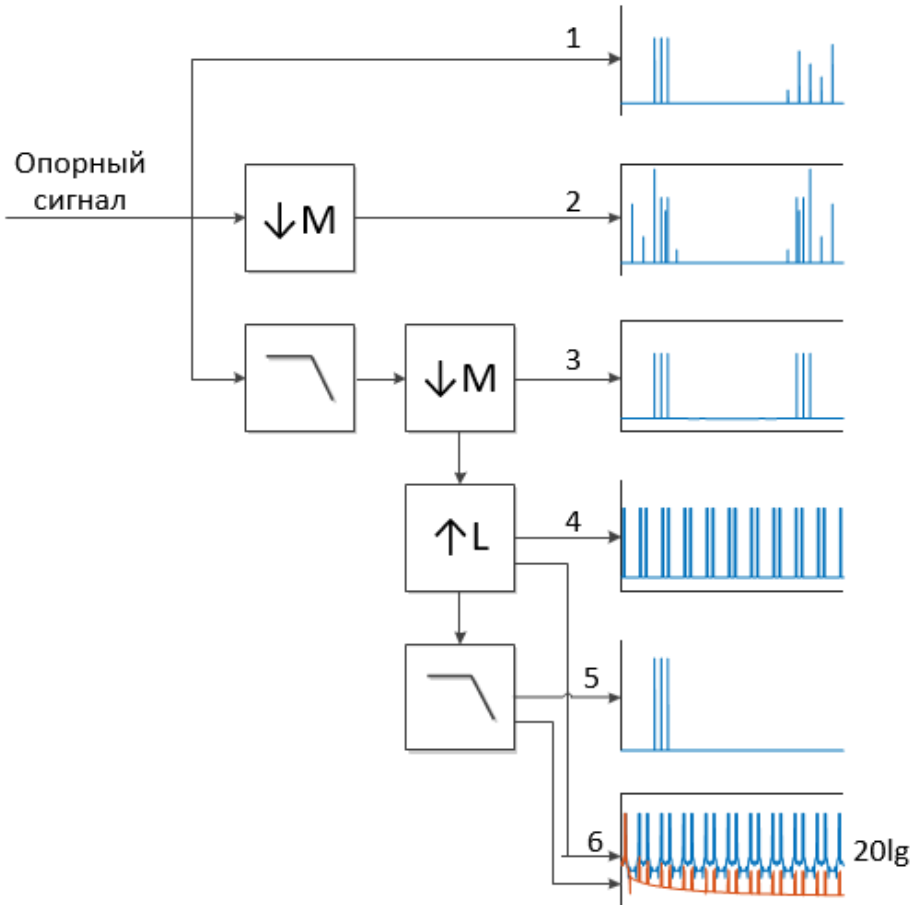

Рисунок 2.1 – Порядок выполнения работы

2.1 Формирование опорного сигнала

Индивидуальное задание №1.

Сформировать опорный сигнал $s(t)$, представляющий собой сумму гармонических сигналов с частотами N кГц, $N+4$ кГц и $N+8$ кГц (N – номер варианта) и амплитудой 1 В. Частота дискретизации равна 1 МГц.

К опорному сигналу добавить сигнал помехи, состоящий из гармонических сигналов с частотами 40, 45, 50, 55 и 60 кГц и различной амплитудой от 0,1 до 1 В. Построить временную и спектральную формы сигнала (линия 1 на рисунке 2.1).

Для построения сигнала во временной области использовать функцию `stem()`. Для наглядности ограничить ось времени диапазоном 0-0.0005 секунд.

2.2 Децимация

Индивидуальное задание №2.

Провести децимацию сформированного сигнала с шагом 10. Показать на графике спектр децимированного сигнала (2). Для построения децимированного сигнала во временной области использовать функцию `stem()`. Для наглядности ограничить ось времени диапазоном 0-0.0005 секунд.

Внимание! Функции `decimate()` и `interp1()` не использовать.

Индивидуальное задание №3.

Пропустить опорный сигнал через цифровой фильтр. Провести децимацию сигнала с шагом 10. Показать на графике спектр децимированного сигнала после фильтрации (3). Для синтеза фильтра использовать функцию `fir1()`. По умолчанию данная функция имеет два аргумента, это порядок фильтра и частота среза ФНЧ, задаваемая в виде отношения частоты в герцах к частоте Найквиста. Частоту среза задать вдвое больше, чем максимальная частота полезного сигнала, порядок фильтра – 400. Фильтрация выполняется командой `filter(h,1,s)`, где h – полученная импульсная характеристика фильтра, а s – входной сигнал.

2.3 Интерполяции

Индивидуальное задание №4.

Провести интерполяцию полученного сигнала (индивидуальное задание №3) с повышением частоты дискретизации до 2 МГц. Построить спектр (4).

Индивидуальное задание №5.

Пропустить сигнал через цифровой ФНЧ, при этом выбрать необходимую частоту среза. Показать на графике спектр полученного сигналапосле фильтрации (5). Построить в одном графическом окне сигнал до фильтрации и после в логарифмическом масштабе (6).

3. Контрольные вопросы

1. Как выполняется децимация с целым шагом?
2. Как выполняется интерполяция с целым шагом?
3. Для чего конвертер частоты дискретизации содержит цифровой фильтр?

Лабораторная работа № 4

«Фильтр скользящего среднего»

Цель работы: формирование навыков синтезирования фильтра в среде Octave/MATLAB.
Задачи лабораторной работы:
– формирование опорного сигнала с шумом;
– создание цифрового фильтра;
– построение временных и спектральных характеристик;
– проверка реакции фильтра на ступенчатое воздействие.

1. Теоретический материал

Элементарной формой КИХ-фильтра является фильтр скользящего среднего (moving average), представленный на рисунке 1.1. Фильтры скользящего среднего популярны для сглаживания данных, например, для анализа стоимости акций и т.д. Входные отсчеты $x(n)$ пропускаются через ряд регистров памяти (помеченных z^{-1} в соответствии с представлением элемента задержки при z-преобразовании). В приведенном примере имеется четыре каскада, соответствующих 4-точечному фильтру скользящего среднего. Каждый отсчет умножается на 0,25, и результаты умножения суммируются для получения значения скользящего среднего, которое подается на выход $y(n)$. На рисунке 1.1 также представлено общее уравнение фильтра скользящего среднего на N точек.

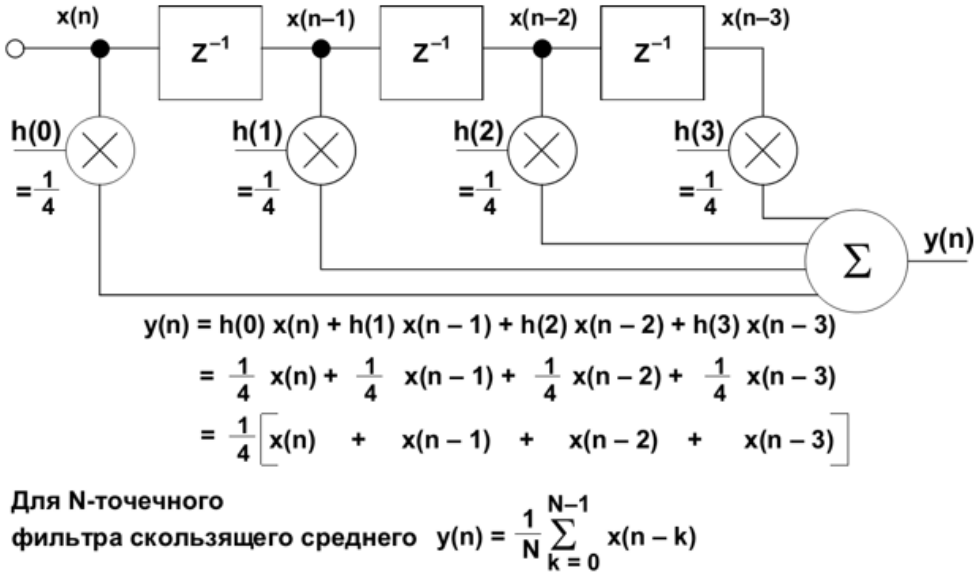

Рисунок 1.1 – Точечный фильтр скользящего среднего

С учетом равенства коэффициентов, наиболее простой путь исполнения фильтра скользящего среднего представлен на рисунке 1.2. Обратите внимание, что первым шагом является запоминание первых четырех отсчетов $x(0)$, $x(1)$, $x(2)$, $x(3)$ в регистрах. Эти величины суммируются и затем умножаются на 0,25 для получения первого выхода $y(3)$.

Поступающий на вход фильтра отсчет $x(4)$ суммируется с результатом, а отсчет $x(0)$ вычитается из результата. Затем новый результат должен быть умножен на 0,25. Поэтому вычисления, требуемые для получения нового значения на выходе, состоят из одного суммирования, одного вычитания и одного умножения, независимо от длины фильтра скользящего среднего.

Рисунок 1.2 – Схема исполнения фильтра скользящего среднего

Реакция 4-точечного фильтра скользящего среднего на ступенчатое воздействие представлена на рисунке 1.3. Обратите внимание, что фильтр скользящего среднего не имеет выброса по фронту входного сигнала. Это делает его полезным в приложениях обработки сигналов, где требуется фильтрация случайного белого шума при сохранении характера входного импульса. На рисунке 1.4 показано, что уровень шума снижается по мере увеличения числа точек. Существенно, что время реакции фильтра на ступенчатое воздействие от 0 % до 100 % равно произведению общего количества точек фильтра на период дискретизации.

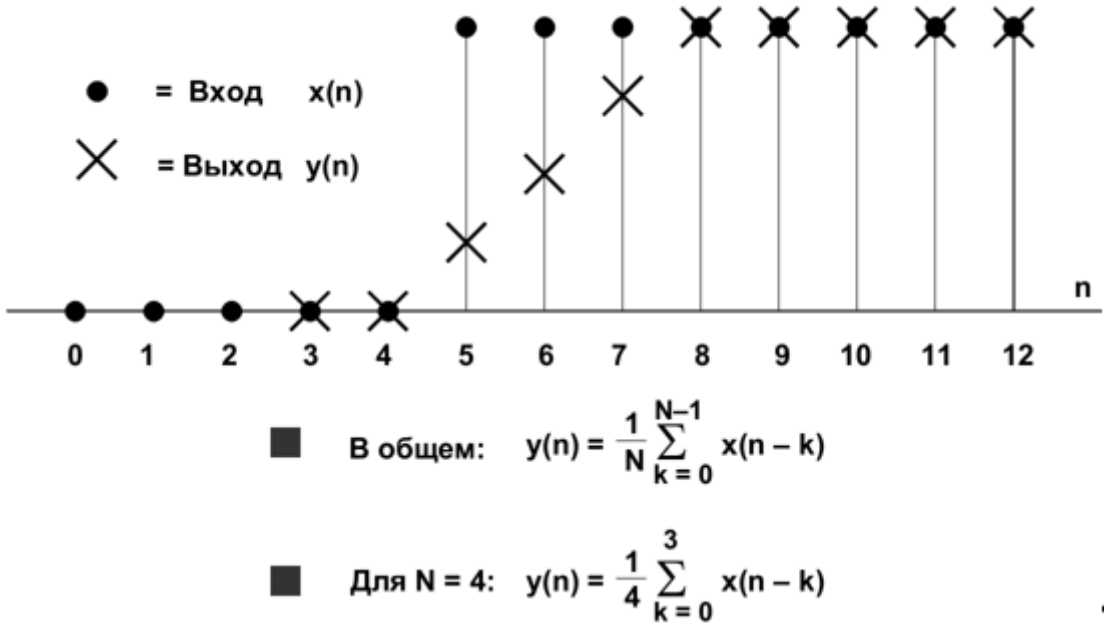

Рисунок 1.3 – Реакция 4-точечного фильтра скользящего среднего на ступенчатое воздействие

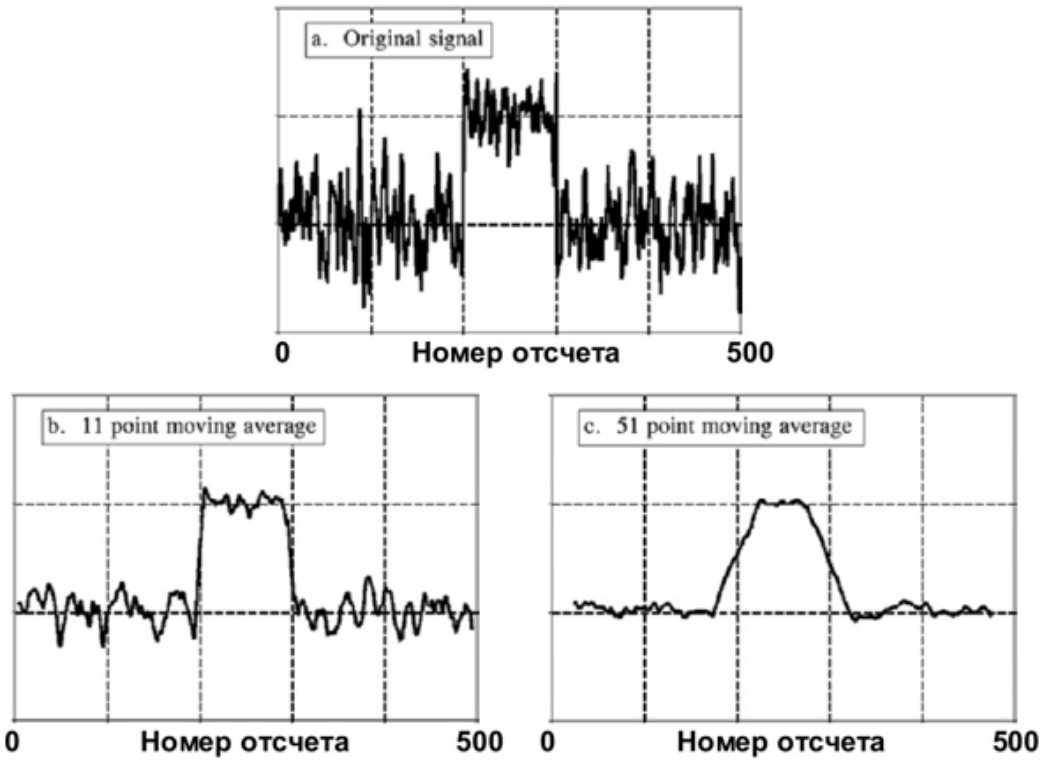

Рисунок 1.4 – Реакция фильтра скользящего среднего на воздействие в виде смеси шума и ступенчатого сигнала

Частотная характеристика фильтра скользящего среднего выражается функцией $\sin(x)/x$. Она представлена в линейном масштабе на рисунке 1.5. Увеличение числа точек при реализации фильтра сужает основной лепесток, но существенно не уменьшает амплитуду боковых лепестков частотной характеристики, которая равна приблизительно -14 дБ для фильтра с 11 и с 31 отводами (длиной буфера). Естественно, эти фильтры не подходят в том случае, где требуется большое ослабление в полосе задержания.

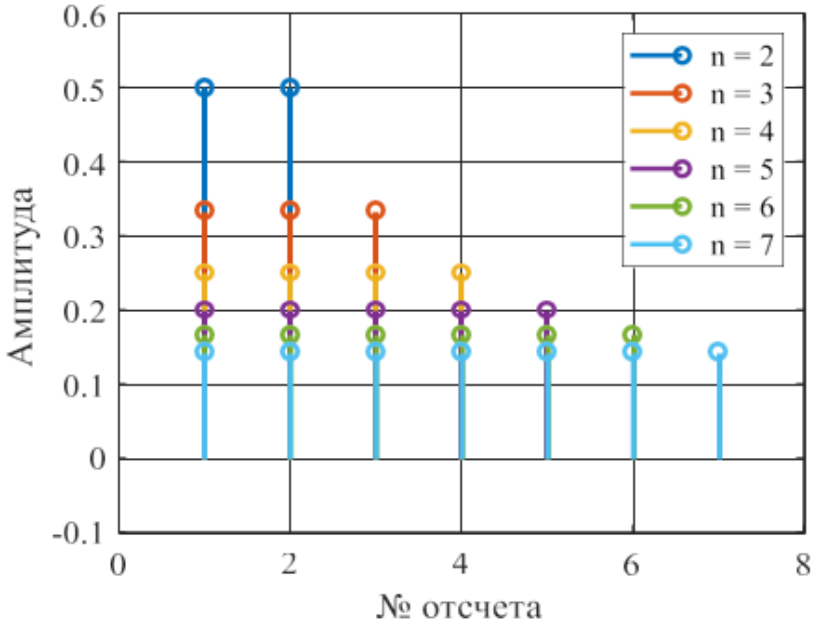

Рисунок 1.5 – Отсчеты импульсной характеристики фильтра скользящего среднего

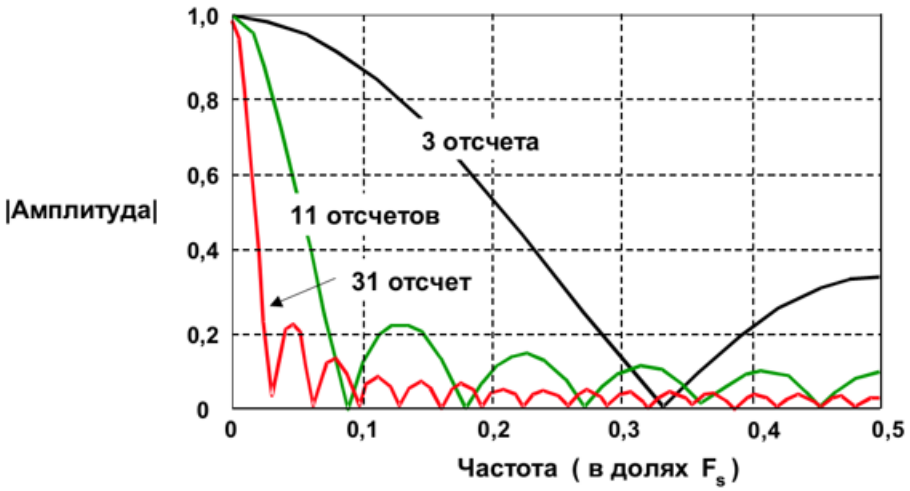

Рисунок 1.6 – Частотная характеристика фильтра скользящего среднего

Можно существенно улучшить эффективность простого КИХ-фильтра скользящего среднего, выбирая разные веса или значения коэффициентов вместо равных значений. Крутизна спада может быть увеличена добавлением большего количества звеньев в фильтр, а характеристики полосы затухания улучшаются выбором надлежащих коэффициентов фильтра. Обратите внимание, что, в отличие от фильтра скользящего среднего, для реализации каждой ступени обобщенного КИХ-фильтра требуется цикл умножения с накоплением. Сущность проектирования КИХ-фильтра сводится к выбору соответствующих коэффициентов и необходимого числа звеньев при формировании желаемой частотной характеристики фильтра $H(f)$. Ключевой теоремой проектирования КИХ-фильтра является утверждение, что коэффициенты $h(n)$ КИХ-фильтра являются квантованными значениями импульсной характеристики этого фильтра. Соответственно, импульсная характеристика является обратным дискретным преобразованием Фурье от $H(f)$.

2. Задание на работу

2.1 Формирование опорного сигнала

Индивидуальное задание №1.

Сформировать опорный сигнал $s(t)$, представляющий собой сумму гармонических сигналов с частотами N кГц, $N+3$ кГц и $N+6$ кГц (N – номер варианта), амплитудой 1 В. Частота дискретизации равна 1 МГц.

К опорному сигналу добавить аддитивный белый гаусовский шум (АБГШ), как показано ниже. Построить временную форму сигнала и спектр в логарифмическом масштабе.

Пример №1.

АБГШ добавляется функцией `awgn()`, которая может иметь два или три аргумента. Первый – входной сигнал, второй – отношение сигнал/шум в дБ, третий аргумент указывает на автоматическое измерение мощности сигнала для корректного расчета уровня шума.

```
sn = awgn(s,10,'measured');
```

2.2 Фильтрация

Индивидуальное задание №2.

Напишите на языке Octave/MATLAB программу для цифровой фильтрации с помощью фильтра скользящего среднего, руководствуясь следующими алгоритмом:

1. Задать переменную n для длины фильтра.
2. Задать переменную-буфер для хранения текущего сигнала с нулевым значением.
3. Добавить цикл с изменяемой переменной k в диапазоне от 1 до суммы длины входного сигнала и длины фильтра без одного отсчета.
4. Далее добавить следующее условие:

Если k меньше или равно n, то к числу в буфере добавить текущий отсчет сигнала.

Иначе, если k больше длины входного сигнала, то от буфера отнять входной отсчет сигнала с индексом $(k-n)$.

Иначе к буферу прибавить текущий отсчет сигнала и отнять отсчет с индексом $(k-n)$.

Конец условия.

5. Выходному отсчету присвоить значение буфера, поделенное на длину фильтра.
6. Конец цикла.

Проверить работу фильтра с различной длиной (2-50), построить временные и спектральные характеристики. Проверить реакцию фильтра на ступенчатое воздействие.

3. Контрольные вопросы

1. Как выполняется децимация с целым шагом?
2. Как выполняется интерполяция с целым шагом?
3. Для чего конвертер частоты дискретизации содержит цифровой фильтр?

ЗАКЛЮЧЕНИЕ

Освоение данного методического пособия обеспечивает приобретение необходимых знаний и компетенций в области цифровой обработки сигналов, позволяя студентам освоить как основные методы, так и сложные алгоритмы обработки и анализа сигналов, применяемых в современных инженерных системах.

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Björnson E. et al. Introduction to Multiple Antenna Communications and Reconfigurable Surfaces. – 2024.

2. Бакулин М. Г., Варукина Л. А., Крейнделин В. Б. Технология MIMO: принципы и алгоритмы. – 2014.

3. Clerckx B., Oestges C. MIMO wireless networks: channels, techniques and standards for multi-antenna, multi-user and multi-cell systems. – Academic Press, 2013.

4. Paulraj A., Nabar R., Gore D. Introduction to space-time wireless communications. – Cambridge university press, 2003.

5. Brown T., Kyritsi P., De Carvalho E. Practical guide to MIMO radio channel: With MATLAB examples. – John Wiley & Sons, 2012.

6. Björnson E. et al. Massive MIMO networks: Spectral, energy, and hardware efficiency //Foundations and Trends® in Signal Processing. – 2017. – Т. 11. – №. 3-4. – С. 154-655.

7. Marzetta T. L., Larsson E. G., Yang H. Fundamentals of massive MIMO. – Cambridge University Press, 2016.

ПРИЛОЖЕНИЕ А

(справочное)

Параметры функции plot()

Color:

Specifier	Color
r	Red
g	Green
b	Blue
c	Cyan
m	Magenta
y	Yellow
k	Black
w	White

LineStyle:

Specifier	Line Style
-	Solid line (default)
--	Dashed line
:	Dotted line
-.	Dash-dotted line

Marker type:

Specifier	Marker Type
+	Crosshair
o	Circle
*	Star
.	Point
x	Cross
's'	Square
'd'	Diamond
^	Upward-facing triangle
>	Right-pointing triangle
<	Left-pointing triangle
'p'	Pentagram (default)
'h'	Hexagram