Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Институт информационных технологий

Кафедра ИС

# ОТЧЁТ

по лабораторной работе №3

по дисциплине «Технологии обработки информации»

на тему «Дискретные фильтры»

Выполнил:

ст. гр. ИС/б-21-2-о

Мовенко К. М.

Проверил:

Карлусов В.Ю.

Севастополь

2024

# ЦЕЛЬ РАБОТЫ

Получить практические навыки расчета и анализа временных (импульсной и переходной) характеристик и частотных (АЧХ, ФЧХ, фазовой и групповой задержки) характеристик дискретных фильтров. Познакомиться с функциями среды MATLAB для дискретной фильтрации, преобразования форм представления дискретных фильтров, расчёта и построения графиков временных и частотных характеристик дискретных систем.

# ВАРИАНТ ЗАДАНИЯ

Вариант 2. Система задана формой представления «функция передачи»:

# ХОД РАБОТЫ

## Вычисление дискретной свёртки

Заданные коэффициенты полиномов числителя и знаменателя и были подставлены в формулу функции передачи. Фильтр является рекурсивным.

При отбросе знаменателя была получена формула нерекурсивного фильтра.

Для нерекурсивного фильтра импульсная характеристика представляет собой массив коэффициентов ().

Был написан программный код, который с помощью функции conv (convolve) вычисляет выходные значения при прохождении сигнала через фильтр и строит график выходного сигнала (листинг 1). Входной сигнал был задан вектором .

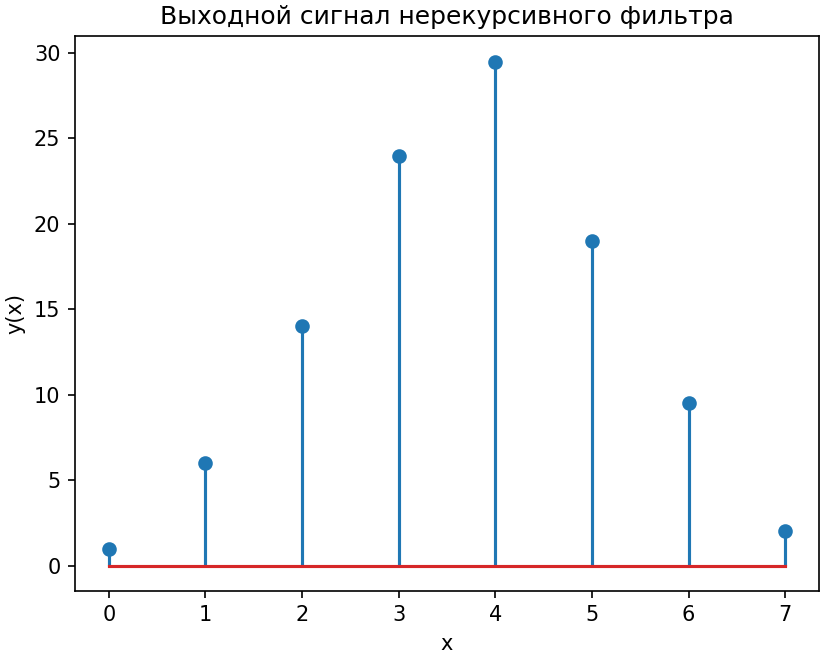


Рисунок 1 – График выходного сигнала нерекурсивного фильтра

Затем с помощью функции filter были вычислены значения выходного сигнала рекурсивного фильтра (листинг 2). При вычислениях была рассмотрена вторая порция сигнала .

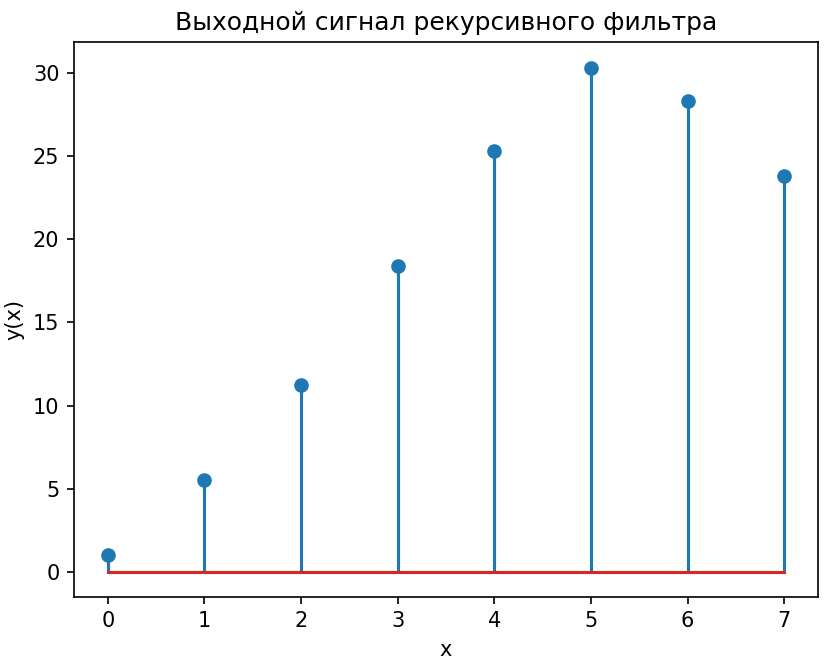


Рисунок 2 – График выходного сигнала рекурсивного фильтра

## Расчёт временных характеристик системы

Был написан код, который рассчитывает импульсную характеристику рекурсивного фильтра с помощью функции impulse (листинг 3).

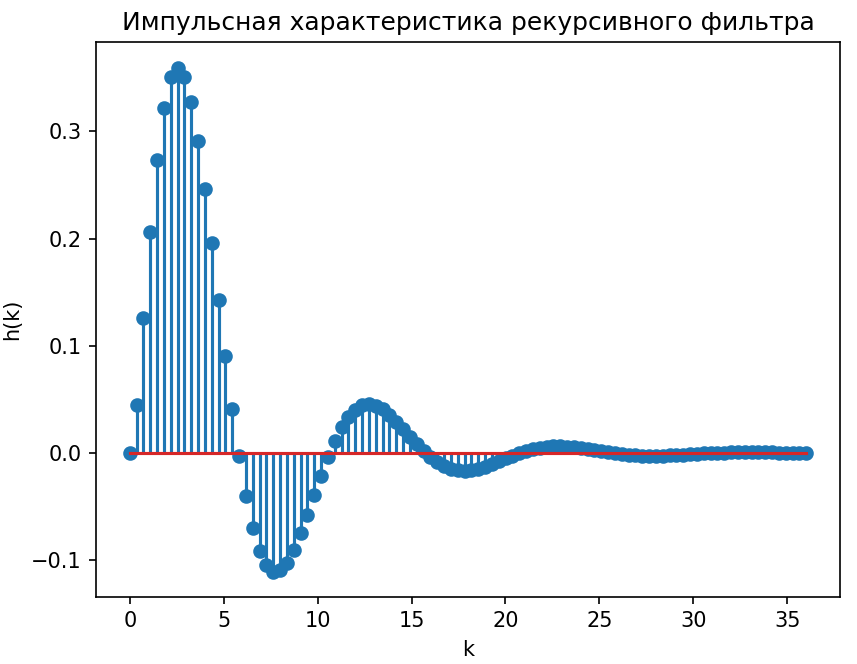


Рисунок 3 – График импульсной характеристики рекурсивного фильтра

Затем с помощью функции filter при подаче единичных импульсов была получена переходная характеристика рекурсивного фильтра (листинг 4).



Рисунок 4 – График переходной характеристики рекурсивного фильтра

## Нули и полюсы фильтра

Была написана программа, которая для заданных параметров рекурсивного фильтра вычисляет нули и полюсы и выводит их на экран (листинг 5).

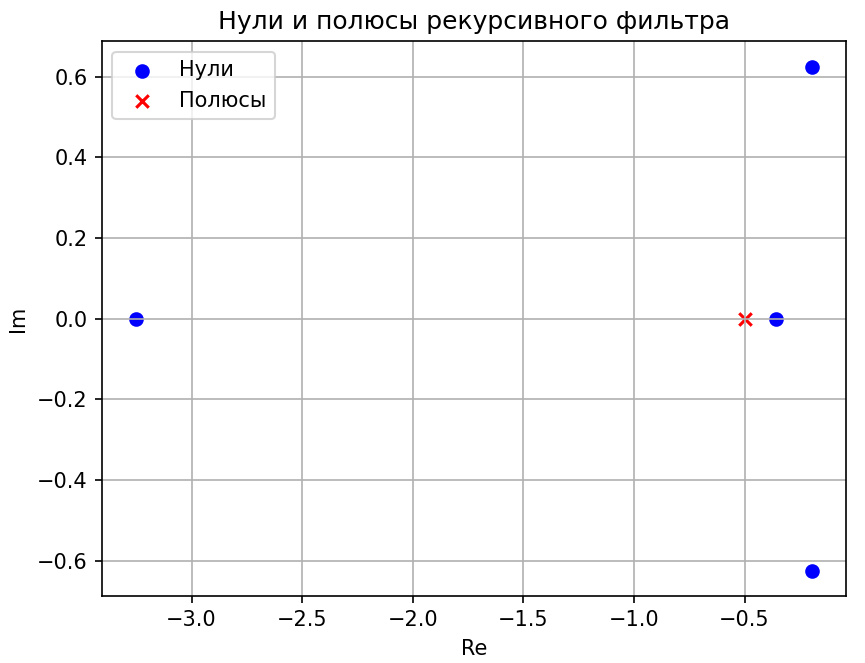


Рисунок 5 – Нули и полюсы рекурсивного фильтра

## Расчёт частотных характеристик системы

Был написан код для построения амплитудно- и фазочастотной характеристик фильтра (листинг 6). АЧХ была построена в децибелах, а ФЧХ – в радианах.

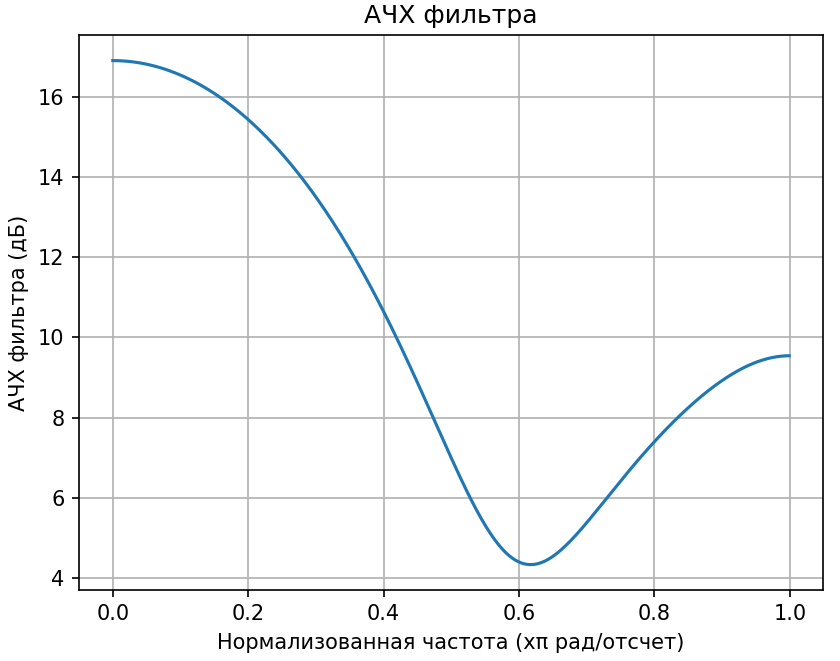


Рисунок 6 – АЧХ фильтра

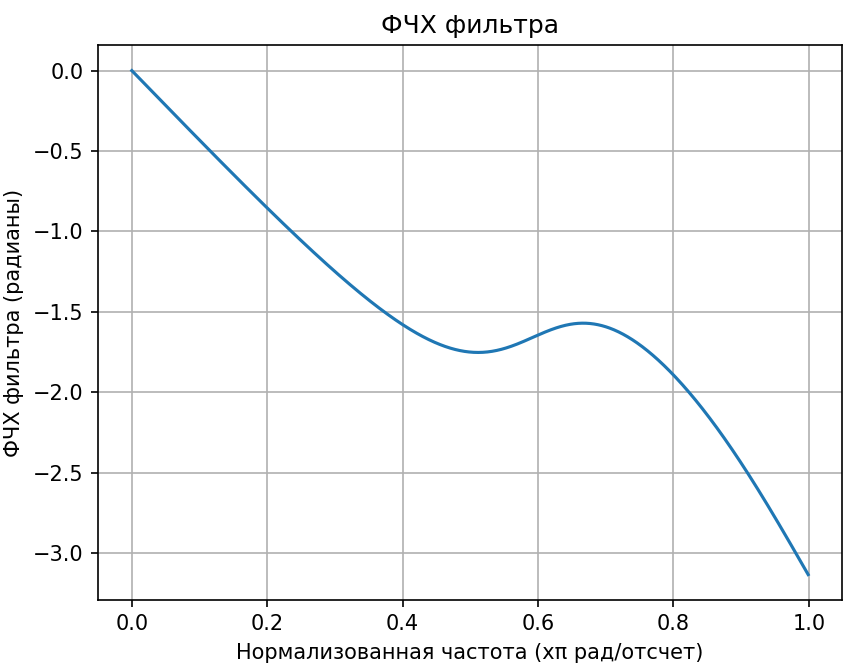


Рисунок 7 – ФЧХ фильтра

Затем был написан код для расчёта фазовой и групповой задержек с помощью функций phasedelay и grpdelay (листинг 7).

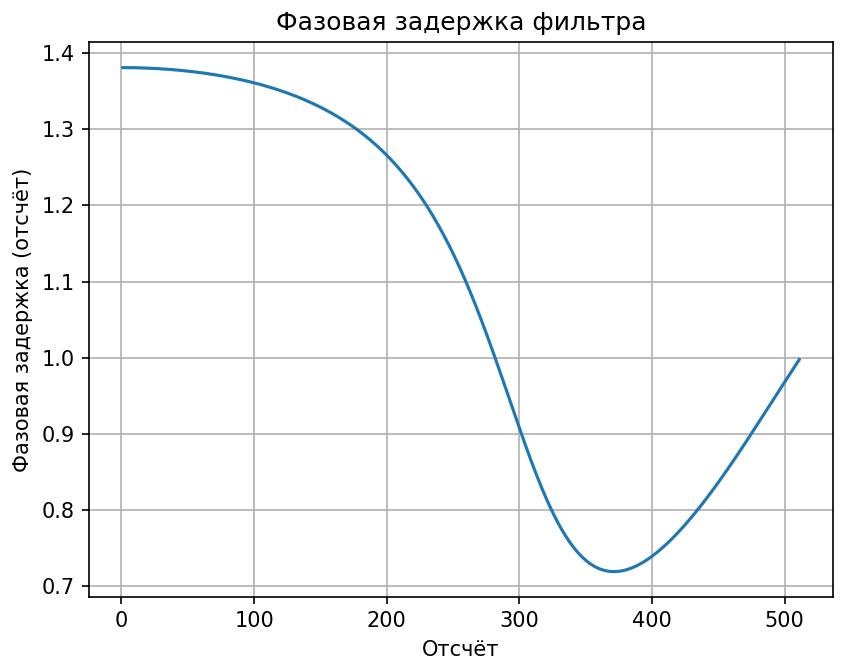


Рисунок 8 – Фазовая задержка фильтра

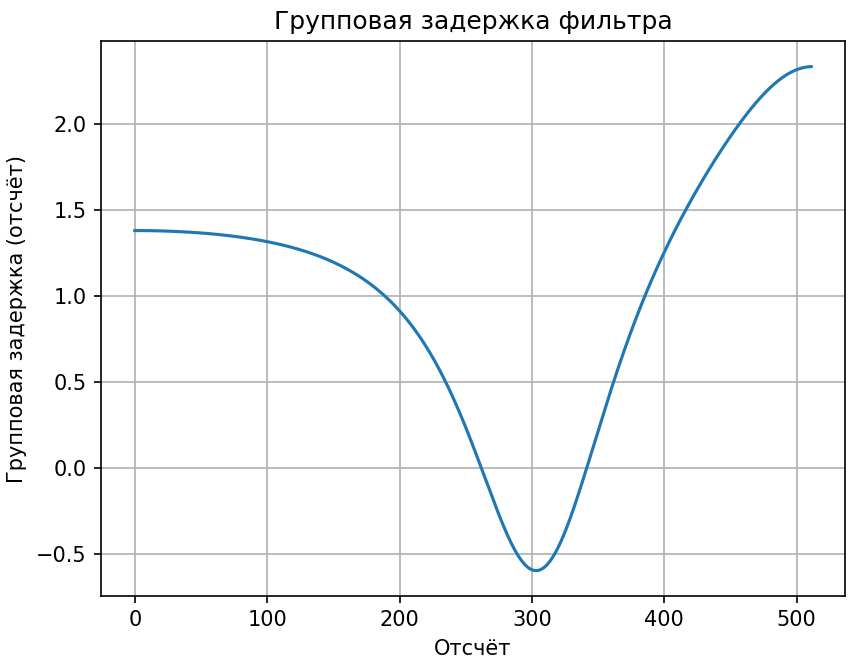


Рисунок 9 – Групповая задержка фильтра

## Представление системы в виде секций второго порядка

Был написан код для представления системы в виде секций второго порядка (листинг 8). В результате была получена матрица sos.

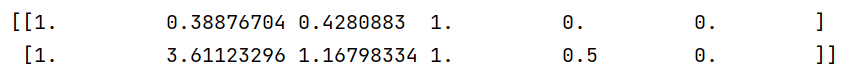


Рисунок 10 – Матрица sos

По её строкам были определены коэффициенты функции передачи.

# программный код

Листинг 1 – Выходной сигнал нерекурсивного фильтра

import numpy as np  
import scipy.signal as signal  
import matplotlib.pyplot as plt  
  
*# дискретная свёртка входного сигнала*x = [1, 2, 3, 4]  
  
*# импульсная характеристика нерекурсивного фильтра*h = [1, 4, 3, 2, 0.5]  
  
*# отсчёты выходного сигнала*y = signal.convolve(x, h)  
  
*# график выходного сигнала*plt.stem(y)  
plt.title('Выходной сигнал нерекурсивного фильтра')  
plt.xlabel('x')  
plt.ylabel('y(x)')  
plt.show()

Листинг 2 – Выходной сигнал рекурсивного фильтра

import numpy as np  
import scipy.signal as signal  
import matplotlib.pyplot as plt  
  
*# параметры фильтра*b = [1, 4, 3, 2, 0.5]  
a = [1, 0.5]  
  
*# дискретная свёртка входного сигнала*x1 = [1, 2, 3, 4]  
x2 = [5, 4, 3, 2]  
  
*# расчет значений выходного сигнала для рекурсивного фильтра*zi = np.zeros(max(len(a), len(b)) - 1)  
y1, s = signal.lfilter(b, a, x1, zi=zi)  
y2, \_ = signal.lfilter(b, a, x2, zi=s)  
  
*# Объединяем порции выходного сигнала*y = np.concatenate((y1, y2))  
  
*# график выходного сигнала*plt.stem(y)  
plt.title('Выходной сигнал рекурсивного фильтра')  
plt.xlabel('x')  
plt.ylabel('y(x)')  
plt.show()

Листинг 3 – Импульсная характеристика рекурсивного фильтра

import scipy.signal as signal  
import matplotlib.pyplot as plt  
  
*# параметры фильтра*b = [1, 4, 3, 2, 0.5]  
a = [1, 0.5]  
  
*# расчет импульсной характеристики*t, h = signal.impulse((a, b))  
  
*# график импульсной характеристики*plt.figure()  
plt.stem(t, h)  
plt.title('Импульсная характеристика рекурсивного фильтра')  
plt.xlabel('k')  
plt.ylabel('h(k)')  
plt.show()

Листинг 4 – Переходная характеристика рекурсивного фильтра

import numpy as np  
import scipy.signal as signal  
import matplotlib.pyplot as plt  
  
*# параметры фильтра*b = [1, 4, 3, 2, 0.5]  
a = [1, 0.5]  
  
*# переходная характеристика рекурсивного фильтра*g = signal.lfilter(b, a, np.ones(20))  
  
*# график переходной характеристики*plt.stem(g)  
plt.title('Переходная характеристика рекурсивного фильтра')  
plt.xlabel('k')  
plt.ylabel('g(k)')  
plt.grid(True)  
plt.show()

Листинг 5 – Нули и полюсы рекурсивного фильтра

import numpy as np  
import scipy.signal as signal  
import matplotlib.pyplot as plt  
  
*# параметры фильтра*b = [1, 4, 3, 2, 0.5]  
a = [1, 0.5]  
  
*# нули и полюса рекурсивного фильтра*z, p, k = signal.tf2zpk(b, a)  
  
*# Постройте график нулей и полюсов*plt.figure()  
plt.scatter(np.real(z), np.imag(z), marker='o', color='b', label='Нули')  
plt.scatter(np.real(p), np.imag(p), marker='x', color='r', label='Полюсы')  
plt.xlabel('Re')  
plt.ylabel('Im')  
plt.title('Нули и полюсы рекурсивного фильтра')  
plt.legend()  
plt.grid(True)  
plt.show()

Листинг 6 – АЧХ и ФЧХ фильтра

import numpy as np  
import scipy.signal as signal  
import matplotlib.pyplot as plt  
  
*# параметры фильтра*b = [1, 4, 3, 2, 0.5]  
a = [1, 0.5]  
  
*# вектор частот*w, h = signal.freqz(b, a, worN=2\*512)  
  
*# АЧХ в децибелах*K\_amp = 20 \* np.log10(np.abs(h))  
  
*# график АЧХ фильтра*plt.figure()  
plt.plot(w / np.pi, K\_amp)  
plt.title('АЧХ фильтра')  
plt.xlabel('Нормализованная частота (xπ рад/отсчет)')  
plt.ylabel('АЧХ фильтра (дБ)')  
plt.grid(True)  
  
*# график ФЧХ фильтра*plt.figure()  
plt.plot(w / np.pi, np.angle(h))  
plt.title('ФЧХ фильтра')  
plt.xlabel('Нормализованная частота (xπ рад/отсчет)')  
plt.ylabel('ФЧХ фильтра (радианы)')  
plt.grid(True)  
plt.show()

Листинг 7 – Фазовая и групповая задержка

import numpy as np  
import scipy.signal as signal  
import matplotlib.pyplot as plt  
  
*# параметры фильтра*b = [1, 4, 3, 2, 0.5]  
a = [1, 0.5]  
  
*# расчет фазовой задержки*w, H = signal.freqz(b, a)  
phase\_delay = -np.unwrap(np.angle(H)) / w  
  
*# расчет групповой задержки*\_, group\_delay = signal.group\_delay((b, a))  
  
*# график фазовой задержки*plt.figure()  
plt.plot(phase\_delay)  
plt.title('Фазовая задержка фильтра')  
plt.xlabel('Отсчёт')  
plt.ylabel('Фазовая задержка (отсчёт)')  
plt.grid(True)  
  
*# график групповой задержки*plt.figure()  
plt.plot(group\_delay)  
plt.title('Групповая задержка фильтра')  
plt.xlabel('Отсчёт')  
plt.ylabel('Групповая задержка (отсчёт)')  
plt.grid(True)  
plt.show()

Листинг 8 – Преобразование в форму sos

import scipy.signal as signal  
  
*# параметры фильтра*b = [1, 4, 3, 2, 0.5]  
a = [1, 0.5]  
  
*# нули и полюса рекурсивного фильтра*z, p, k = signal.tf2zpk(b, a)  
  
*# преобразование в форму SOS*sos = signal.zpk2sos(z, p, k)  
  
*# вывод матрицы SOS*print(sos)

# ВЫВОД

В ходе работы были получены навыки расчёта и анализа временных (импульсная, переходная) и частотных (АЧХ, ФЧХ, фазовая и групповая задержки) характеристик дискретных фильтров.