ГУАП

КАФЕДРА № 42

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| канд. техн. наук, доцент |  |  |  | А. В. Аграновский |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1 |
| Сигналы и фильтры |
| по курсу: ЦИФРОВАЯ ОБРАБОКА И ПЕРЕДАЧА СИГНАЛОВ |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4329 |  |  |  | С.Т. Лисицин |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2025

1. Цель работы

Познакомиться с основными видами сигналов и изучить методы анализа цифровых сигналов.

1. Задачи

Для выполнения следующих задач использовался язык программирования высокого уровня Python:

Задание 1

1. Сформируйте единичный импульс длиной 20 отсчетов.

2. Получите импульсные характеристики рассмотренных фильтров.

3. Постройте соответствующие графики.

4. Сравните с результатами расчета по представленным формулам.

Задание 2

1. Определите единичный скачок длиной 20 отсчетов.

2. Получите импульсные характеристики рассмотренных фильтров.

3. Постройте соответствующие графики.

4. Сравните с результатами расчета по представленным формулам.

Задание 3

1. Используя синусоидальные сигналы постоянной амплитуды и различной частоты,

получите амплитудно-частотную характеристику фильтров.

2. Постройте соответствующие графики.

3. Сравните с результатами расчета по представленным формулам.

Задание 4

1. Используя синусоидальные сигналы постоянной амплитуды с нулевой начальной фазой

при различных частотах, получите фазо-частотную характеристику фильтров.

2. Постройте соответствующие графики.

3. Сравните с результатами расчета по представленным формулам.

1. Теоретические сведения

Сигналы, которые используются для анализа фильтров, бывают нескольких типов. Наиболее важные из них: единичный импульс, единичный скачок и гармонический сигнал.

Единичный импульс δ[n] равен единице при n=0 и нулю при всех остальных n. Он используется для нахождения импульсной характеристики фильтра. Импульсная характеристика h[n] полностью определяет поведение линейного дискретного фильтра.

Единичный скачок u[n] равен единице при n ≥ 0 и нулю при n < 0. Он применяется для нахождения переходной характеристики фильтра.

Для анализа частотных свойств фильтра используется синусоидальный сигнал

x[n] = A·sin(ωn+φ), (1)

где A – амплитуда, ω – частота, φ – начальная фаза.

При подаче на вход фильтра синусоидального сигнала на выходе также получается синусоида той же частоты, но с изменённой амплитудой и сдвигом по фазе.

Амплитудно-частотная характеристика (АЧХ) описывает зависимость коэффициента усиления фильтра от частоты входного сигнала. Она показывает, какие частоты фильтр пропускает, а какие подавляет.

Фазо-частотная характеристика (ФЧХ) описывает зависимость сдвига по фазе на выходе фильтра от частоты входного сигнала. Сдвиг по фазе возникает из-за задержки сигнала фильтром и является важным параметром при обработке.

В данной работе рассматриваются два фильтра первого порядка: нерекурсивный и рекурсивный.

Нерекурсивный фильтр (FIR) имеет вид:

, (2)

где a – коэффициент фильтра. Его импульсная характеристика конечна, так как он не содержит обратной связи.

Рекурсивный фильтр (IIR) имеет вид:

y[n] = x[n] + a·y[n-1], (3)

где a – коэффициент обратной связи. Его импульсная характеристика бесконечна, так как он содержит обратную связь и выход зависит от предыдущих выходных значений.

Импульсные и переходные характеристики этих фильтров, а также их АЧХ и ФЧХ можно получить как аналитически, так и экспериментально через моделирование.

Формулы вычисления характеристик для данных фильтров представлены на рисунках 1-2:

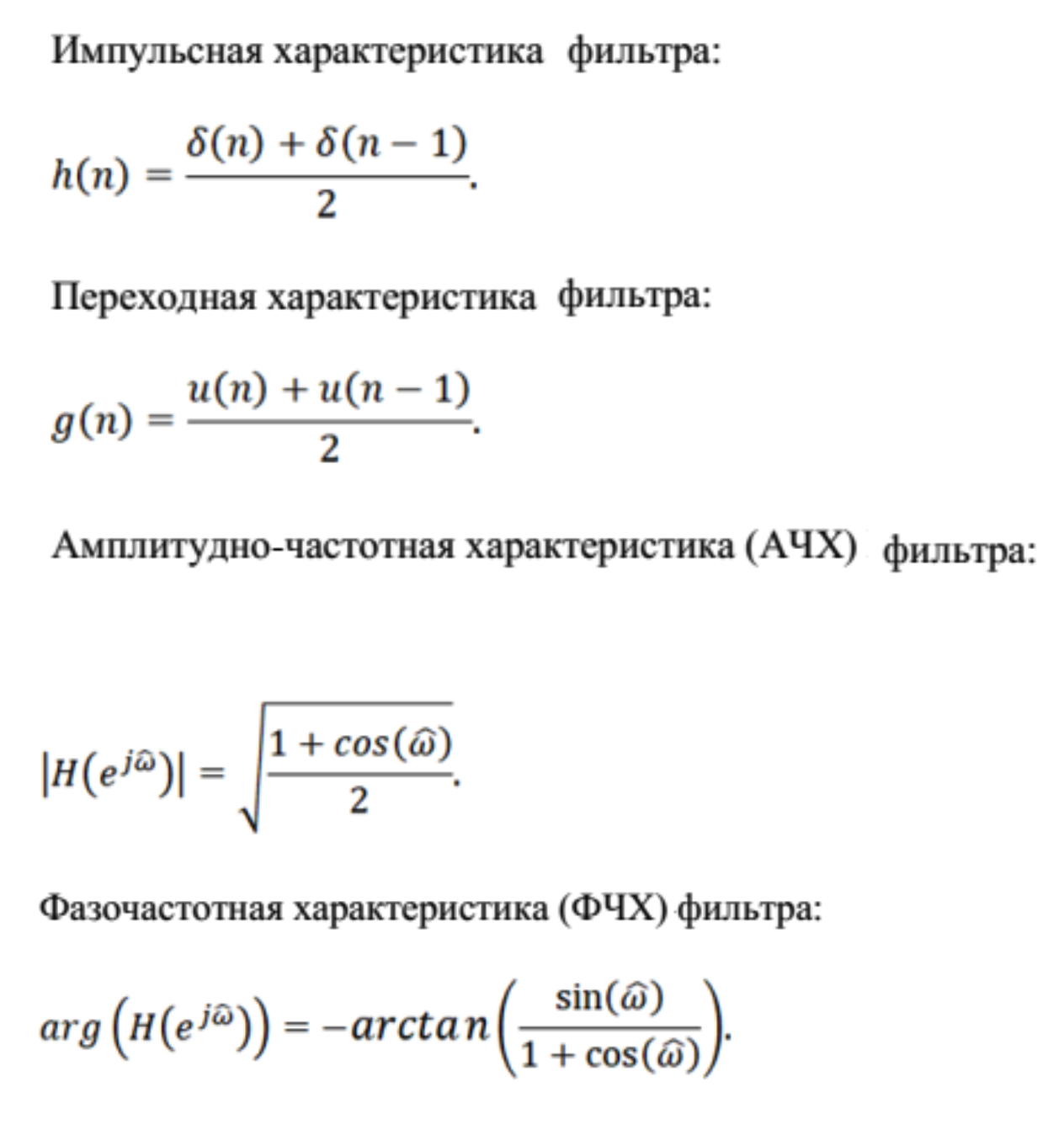


Рисунок 1 – Формулы вычисления характеристик для нерекурсивного фильтра(FIR)

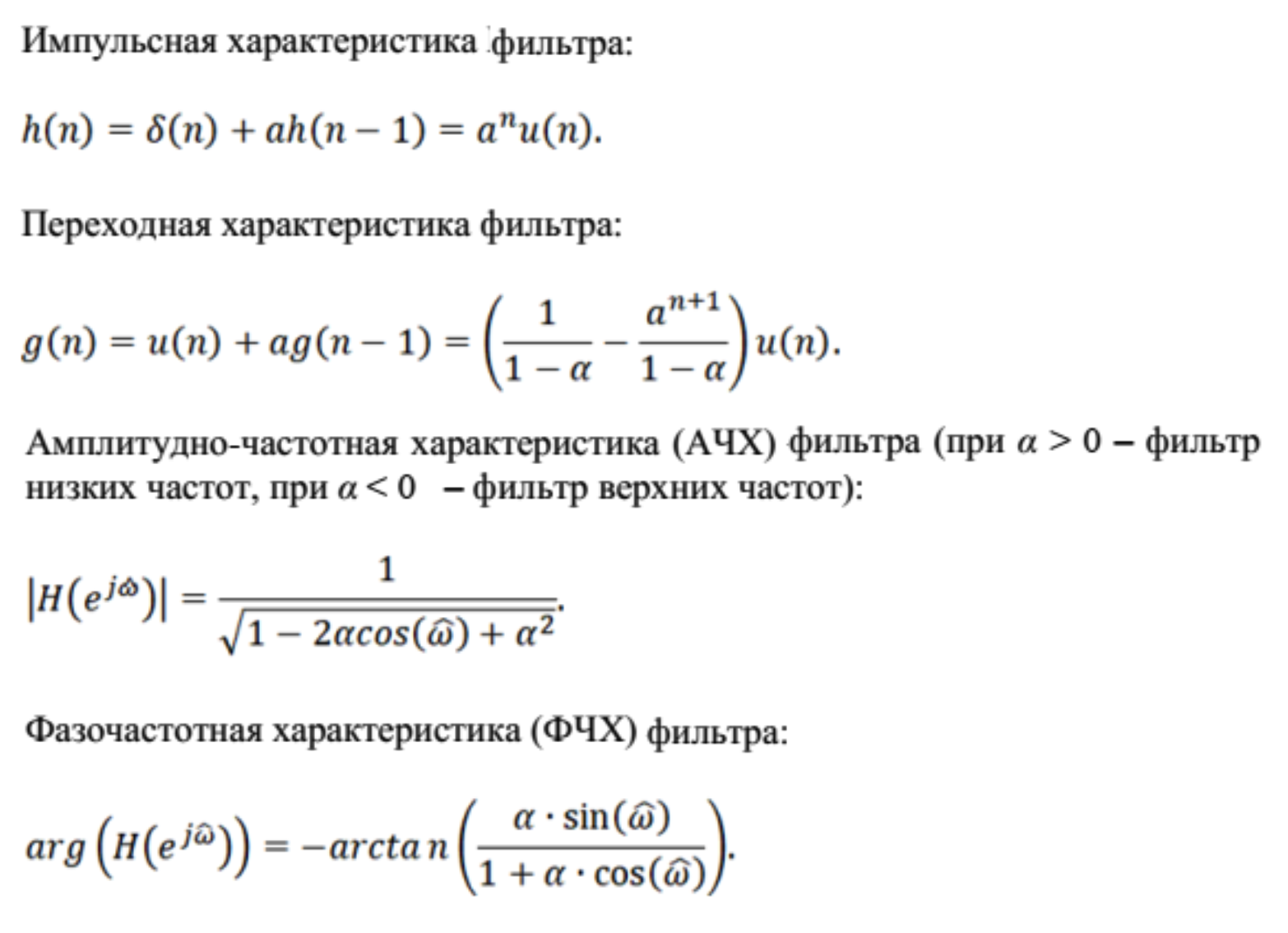


Рисунок 2 – Формулы вычисления характеристик для рекурсивного фильтра(IIR)

1. Ход выполнения лабораторной работы

Сначала был сформирован единичный импульс длиной 20 отсчетов. Этот сигнал использовался для получения импульсных характеристик двух фильтров: нерекурсивного FIR и рекурсивного IIR. В обоих случаях характеристики были рассчитаны как экспериментально на основе программного кода, так и теоретически по аналитическим формулам. На рисунке 3 представлены результаты для обоих фильтров. Хорошо видно, что импульсная характеристика FIR фильтра совпадает с коэффициентами фильтра и имеет конечную длину. Импульсная характеристика IIR фильтра представляет собой убывающую последовательность. Экспериментальные и теоретические графики совпадают, максимальная ошибка равна нулю.

Далее был сформирован единичный скачок длиной 20 отсчетов. Этот сигнал использовался для анализа переходных характеристик фильтров. На рисунке 4 показаны результаты моделирования и расчета. Для FIR фильтра характеристика мгновенно достигает установившегося уровня, начиная с первого отсчета после импульса, что совпадает с аналитическим выражением. Для IIR фильтра наблюдается постепенный рост к пределу. Экспериментальные и теоретические данные совпадают, что подтверждается полным наложением графиков.

Третий этап работы заключался в исследовании амплитудно-частотных характеристик. На вход фильтров подавались гармонические сигналы постоянной амплитуды и различной частоты. На основе установившейся части выходного сигнала определялась амплитуда. Полученные экспериментальные зависимости представлены на рисунке 5 вместе с теоретическими кривыми. Для FIR фильтра измеренные значения совпадают с функцией АЧХ, представленной на рисунке 1. Для IIR фильтра экспериментальные точки совпали с кривой, построенной по формуле, представленной на рисунке 2. Отчетливо видно, что IIR фильтр усиливает низкочастотные сигналы (в два раза при нулевой частоте) и подавляет высокочастотные, в то время как FIR действует как усредняющий оператор и плавно сглаживает сигнал.

Заключительный этап касался фазо-частотных характеристик. Для синусоидальных сигналов с нулевой начальной фазой была определена разность фаз на входе и выходе фильтра. На рисунке 6 представлены результаты моделирования и аналитических расчетов. Экспериментальные кривые, полученные с использованием функции freqz, совпали с теоретическими расчетами. Из графиков видно, что FIR фильтр вызывает задержку примерно на пол-отсчета, а у IIR наблюдается плавный фазовый сдвиг, увеличивающийся с частотой. Это соответствует известным свойствам рекурсивных фильтров, для которых выход зависит от прошлых значений.

Таким образом, все этапы работы выполнены последовательно: сначала формировались тестовые сигналы, затем они пропускались через фильтры, результаты сравнивались с аналитическими расчетами, после чего строились графики. Экспериментальные и теоретические данные совпали, что подтверждает корректность как программной реализации, так и теоретических формул.

1. Графики характеристик исследуемых фильтров

На рисунках 3-7 представлены графики для сравнения фильтров с теоретическими значениями и между собой.

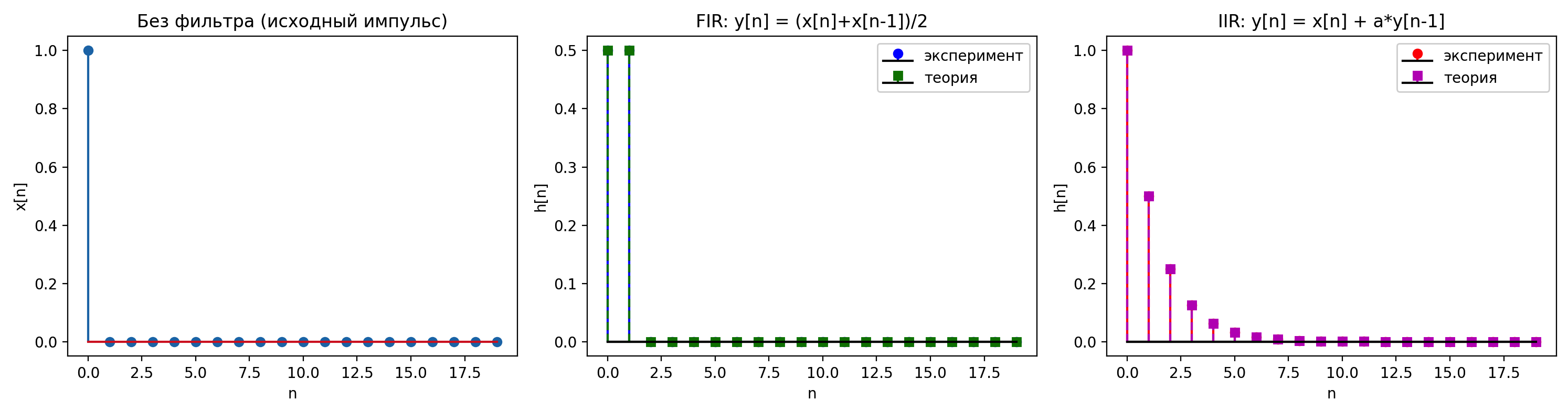


Рисунок 3 – Сравнение фильтров для единичного импульса

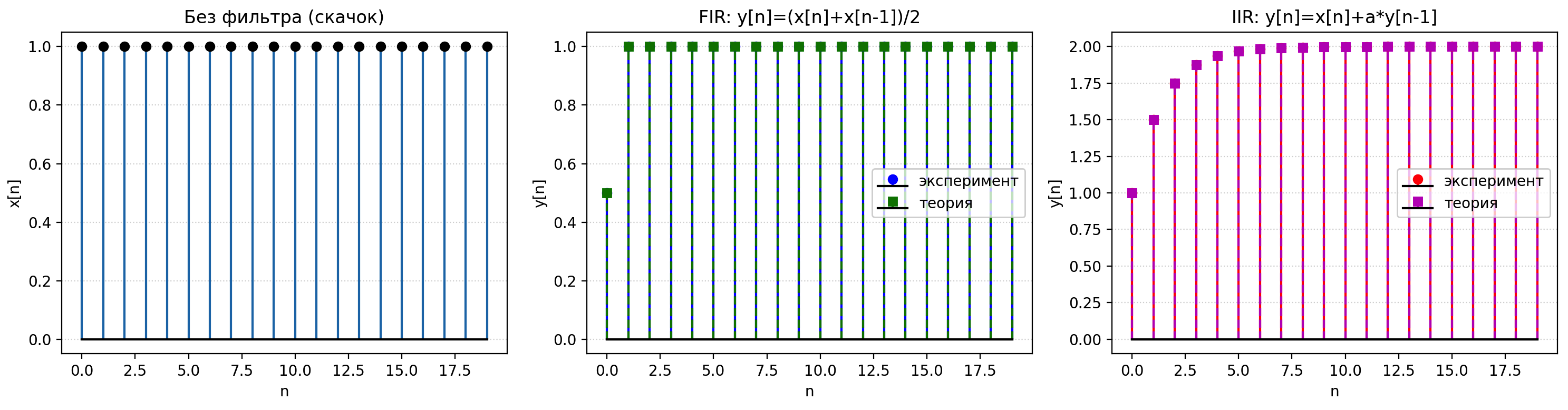


Рисунок 4 – Сравнение фильтров для единичного скачка

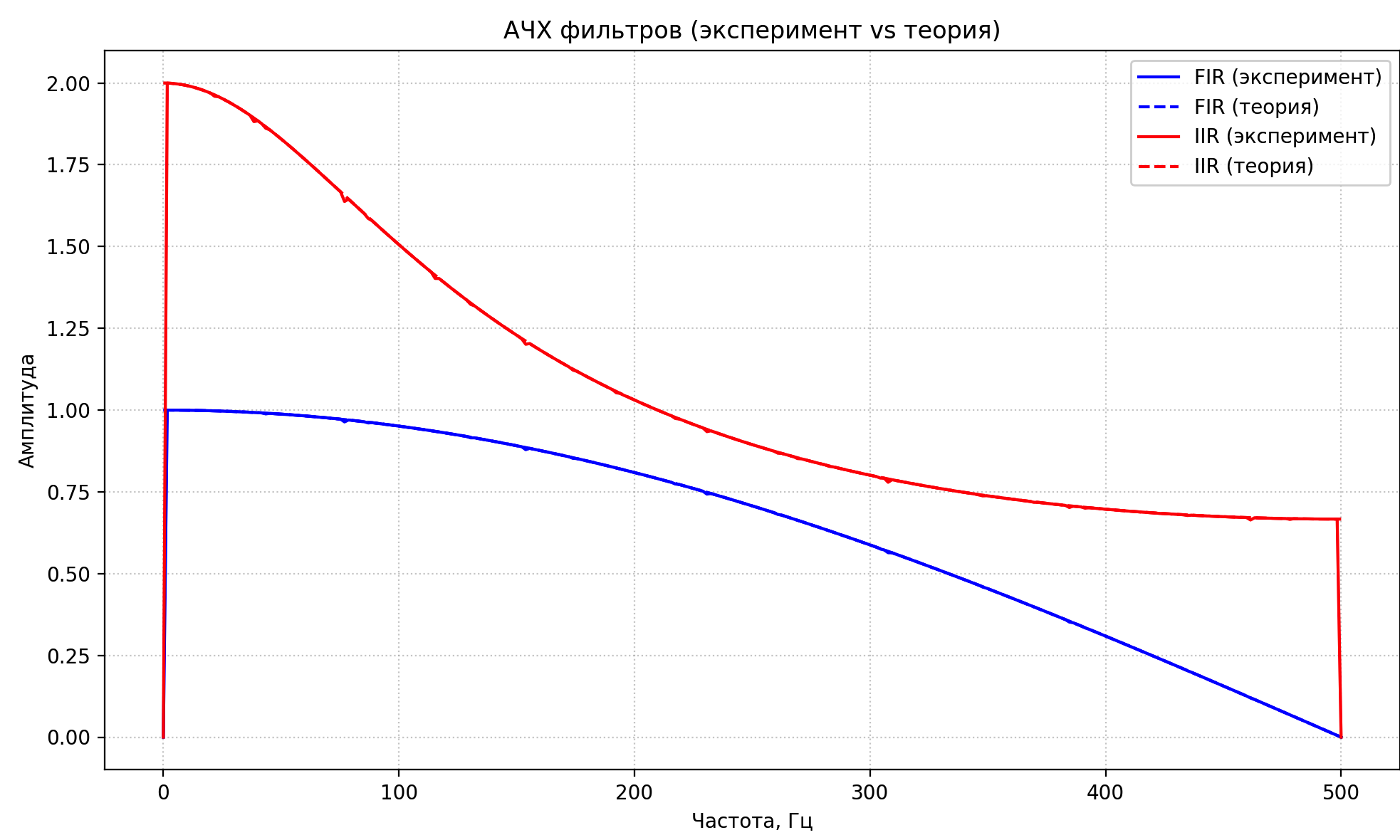


Рисунок 5 – Сравнение АЧХ фильтров для синусоиды

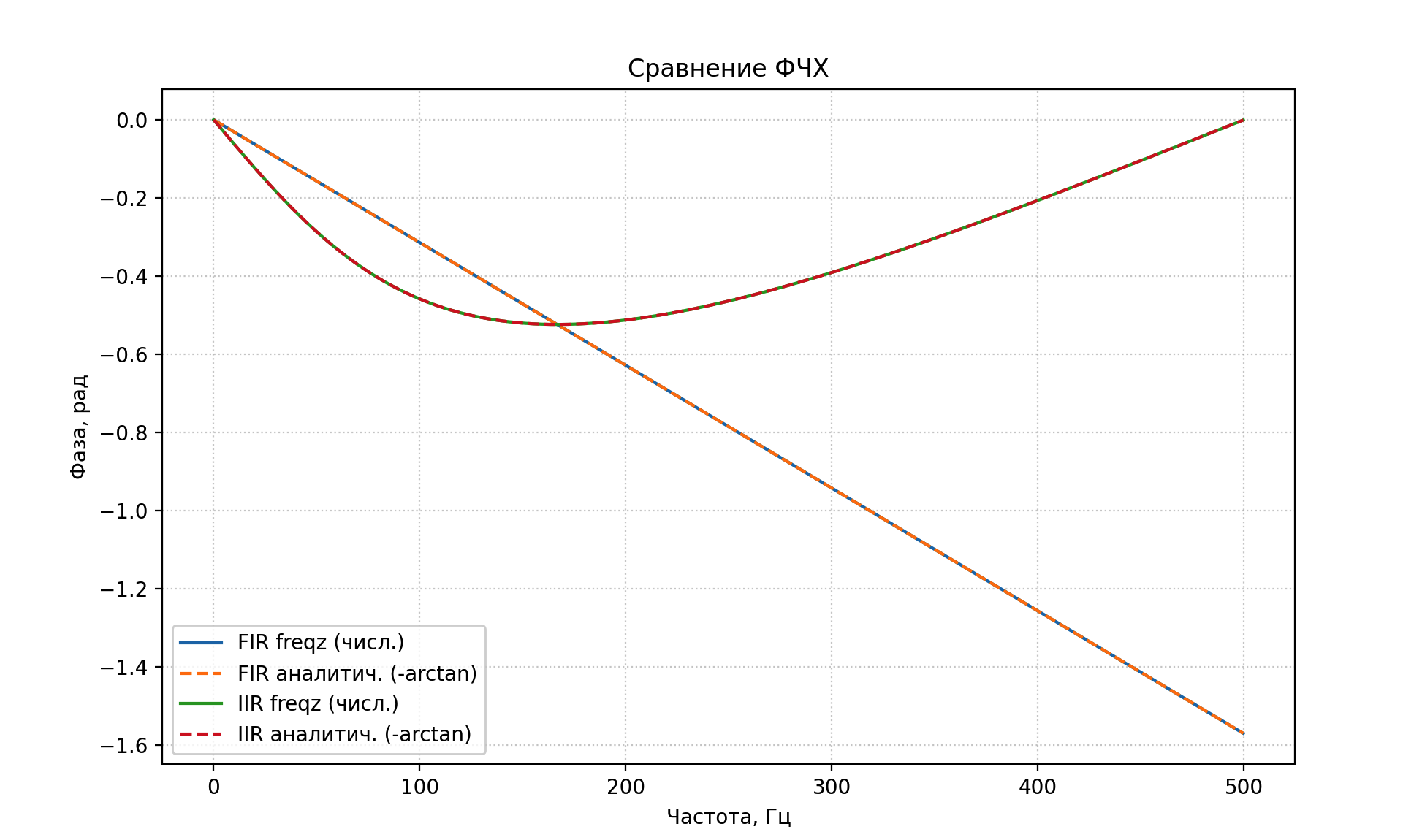


Рисунок 6 – Сравнение ФЧХ фильтров для синусоиды с нулевой начальной фазой

1. Листинг программы

Ниже представлен код программы на языке Python:

import numpy as np

import matplotlib.pyplot as plt

N = 20

a = 0.5

# единичный импульс

x = np.zeros(N)

x[0] = 1.0

# экспериментальный FIR: y[n] = (x[n] + x[n-1]) / 2

h\_fir = np.zeros(N)

for n in range(N):

xm1 = x[n-1] if n-1 >= 0 else 0.0

h\_fir[n] = 0.5 \* (x[n] + xm1)

# теоретический FIR

h\_fir\_theory = np.zeros(N)

h\_fir\_theory[0] = 0.5

if N > 1:

h\_fir\_theory[1] = 0.5

# экспериментальный IIR: y[n] = x[n] + a \* y[n-1]

h\_iir = np.zeros(N)

for n in range(N):

if n == 0:

h\_iir[n] = x[n]

else:

h\_iir[n] = x[n] + a \* h\_iir[n-1]

# теоретический IIR: h[n] = a^n

n\_arr = np.arange(N)

h\_iir\_theory = a \*\* n\_arr

# вычисление ошибок

err\_fir = h\_fir - h\_fir\_theory

err\_iir = h\_iir - h\_iir\_theory

print("FIR (эксперимент):", np.round(h\_fir, 6))

print("FIR (теория) :", np.round(h\_fir\_theory, 6))

print("max |err| FIR :", np.max(np.abs(err\_fir)))

print()

print("IIR (эксперимент):", np.round(h\_iir, 6))

print("IIR (теория) :", np.round(h\_iir\_theory, 6))

print("max |err| IIR :", np.max(np.abs(err\_iir)))

# построение графиков (без фильтра, FIR, IIR)

plt.figure(figsize=(15,4))

plt.subplot(1,3,1)

plt.stem(x)

plt.title("Без фильтра (исходный импульс)")

plt.xlabel("n"); plt.ylabel("x[n]")

plt.subplot(1,3,2)

plt.stem(h\_fir, linefmt="b-", markerfmt="bo", basefmt="k-", label="эксперимент")

plt.stem(h\_fir\_theory, linefmt="g--", markerfmt="gs", basefmt="k-", label="теория")

plt.title("FIR: y[n] = (x[n]+x[n-1])/2")

plt.xlabel("n"); plt.ylabel("h[n]")

plt.legend()

plt.subplot(1,3,3)

plt.stem(h\_iir, linefmt="r-", markerfmt="ro", basefmt="k-", label="эксперимент")

plt.stem(h\_iir\_theory, linefmt="m--", markerfmt="ms", basefmt="k-", label="теория")

plt.title("IIR: y[n] = x[n] + a\*y[n-1]")

plt.xlabel("n"); plt.ylabel("h[n]")

plt.legend()

plt.tight\_layout()

plt.show()

import numpy as np

import matplotlib.pyplot as plt

# параметры

N = 20 # длина сигнала

a = 0.5 # коэффициент рекурсивного фильтра

# 1) единичный скачок

x = np.ones(N)

# 2) экспериментальные отклики

# FIR: y[n] = (x[n] + x[n-1]) / 2

y\_fir = np.zeros(N)

for n in range(N):

xm1 = x[n-1] if n-1 >= 0 else 0.0

y\_fir[n] = 0.5 \* (x[n] + xm1)

# IIR: y[n] = x[n] + a\*y[n-1]

y\_iir = np.zeros(N)

for n in range(N):

if n == 0:

y\_iir[n] = x[n]

else:

y\_iir[n] = x[n] + a \* y\_iir[n-1]

# 3) теоретические отклики

# FIR: для шага u[n]: y[0] = 0.5, y[n>=1] = 1

y\_fir\_theory = np.zeros(N)

y\_fir\_theory[0] = 0.5

y\_fir\_theory[1:] = 1.0

# IIR: решение рекуррентного уравнения для шага u[n]:

# y[n] = (1 - a^(n+1)) / (1 - a), n >= 0

n = np.arange(N)

y\_iir\_theory = (1 - a\*\*(n+1)) / (1 - a)

# 4) построение графиков

plt.figure(figsize=(15,4))

# (а) без фильтра

plt.subplot(1,3,1)

plt.stem(x, basefmt="k-", markerfmt="ko")

plt.title("Без фильтра (скачок)")

plt.xlabel("n"); plt.ylabel("x[n]")

plt.grid(axis='y', linestyle=':', alpha=0.5)

# (б) FIR

plt.subplot(1,3,2)

plt.stem(y\_fir, linefmt="b-", markerfmt="bo", basefmt="k-", label="эксперимент")

plt.stem(y\_fir\_theory, linefmt="g--", markerfmt="gs", basefmt="k-", label="теория")

plt.title("FIR: y[n]=(x[n]+x[n-1])/2")

plt.xlabel("n"); plt.ylabel("y[n]")

plt.legend(); plt.grid(axis='y', linestyle=':', alpha=0.5)

# (в) IIR

plt.subplot(1,3,3)

plt.stem(y\_iir, linefmt="r-", markerfmt="ro", basefmt="k-", label="эксперимент")

plt.stem(y\_iir\_theory, linefmt="m--", markerfmt="ms", basefmt="k-", label="теория")

plt.title("IIR: y[n]=x[n]+a\*y[n-1]")

plt.xlabel("n"); plt.ylabel("y[n]")

plt.legend(); plt.grid(axis='y', linestyle=':', alpha=0.5)

plt.tight\_layout()

plt.show()

# 5) печать для проверки

print("n x[n] FIR\_exp FIR\_th IIR\_exp IIR\_th")

for i in range(N):

print(f"{i:2d} {x[i]:4.1f} {y\_fir[i]:7.4f} {y\_fir\_theory[i]:7.4f} {y\_iir[i]:7.4f} {y\_iir\_theory[i]:7.4f}")

import numpy as np

import matplotlib.pyplot as plt

# параметры

Fs = 1000 # частота дискретизации, Гц

N\_per = 4096 # длина синусоиды

transient = 500 # сколько отсчетов отбросить

freqs\_count = 300 # количество точек

a = 0.5 # коэффициент для IIR

# частоты (рад/отсчет и Гц)

omega = np.linspace(0, np.pi, freqs\_count)

freqs\_hz = omega \* Fs / (2\*np.pi)

# массивы амплитуд

amp\_fir\_meas = np.zeros(freqs\_count)

amp\_iir\_meas = np.zeros(freqs\_count)

n = np.arange(N\_per)

for k, w in enumerate(omega):

# вход: синус с амплитудой 1

x = np.sin(w \* n)

# FIR: y[n] = (x[n]+x[n-1])/2

y\_fir = np.zeros\_like(x)

for i in range(len(x)):

xm1 = x[i-1] if i-1 >= 0 else 0.0

y\_fir[i] = 0.5\*(x[i] + xm1)

# IIR: y[n] = x[n] + a\*y[n-1]

y\_iir = np.zeros\_like(x)

for i in range(len(x)):

if i == 0:

y\_iir[i] = x[i]

else:

y\_iir[i] = x[i] + a\*y\_iir[i-1]

# установившаяся часть

ys = y\_fir[transient:]

yi = y\_iir[transient:]

# амплитуда (половина размаха)

amp\_fir\_meas[k] = (np.max(ys) - np.min(ys)) / 2

amp\_iir\_meas[k] = (np.max(yi) - np.min(yi)) / 2

# теоретические АЧХ

H\_fir\_theory = np.sqrt((1 + np.cos(omega)) / 2)

H\_iir\_theory = 1 / np.sqrt(1 + a\*\*2 - 2\*a\*np.cos(omega))

# построение

plt.figure(figsize=(10,6))

plt.plot(freqs\_hz, amp\_fir\_meas, 'b-', label="FIR (эксперимент)")

plt.plot(freqs\_hz, H\_fir\_theory, 'b--', label="FIR (теория)")

plt.plot(freqs\_hz, amp\_iir\_meas, 'r-', label="IIR (эксперимент)")

plt.plot(freqs\_hz, H\_iir\_theory, 'r--', label="IIR (теория)")

plt.xlabel("Частота, Гц")

plt.ylabel("Амплитуда")

plt.title("АЧХ фильтров (эксперимент vs теория)")

plt.legend()

plt.grid(True, linestyle=":", alpha=0.6)

plt.tight\_layout()

plt.show()

import numpy as np

import matplotlib.pyplot as plt

from scipy.signal import freqz

# параметры

Fs = 1000

N\_per = 2048

freqs\_count = 5

a = 0.5

# частоты (рад/отсчет и Гц)

omega = np.linspace(0.1, np.pi, freqs\_count) # не начинаем с 0, иначе будет константа

freqs\_hz = omega \* Fs / (2\*np.pi)

n = np.arange(N\_per)

signals = []

firs = []

iirs = []

# формирование сигналов и фильтрация

for i, w in enumerate(omega):

x = np.sin(w \* n)

signals.append(x)

# FIR: y[n] = (x[n]+x[n-1])/2

y\_fir = np.zeros(N\_per)

for k in range(N\_per):

xm1 = x[k-1] if k-1 >= 0 else 0.0

y\_fir[k] = 0.5 \* (x[k] + xm1)

firs.append(y\_fir)

# IIR: y[n] = x[n] + a\*y[n-1]

y\_iir = np.zeros(N\_per)

for k in range(N\_per):

if k == 0:

y\_iir[k] = x[k]

else:

y\_iir[k] = x[k] + a\*y\_iir[k-1]

iirs.append(y\_iir)

b\_fir = [0.5, 0.5]

a\_fir = [1]

b\_iir = [1]

a\_iir = [1, -a]

# Частотный отклик (вернёт omega в радиан/сэмпл, т.е. [0, pi])

w, h\_fir = freqz(b\_fir, a\_fir, worN=2048) # w — рад/сэмпл

\_, h\_iir = freqz(b\_iir, a\_iir, worN=2048)

# Перевод частоты в Гц для осей

freqs = w \* Fs / (2\*np.pi)

# ФЧХ от freqz (распаковываем)

phi\_fir\_freqz = np.unwrap(np.angle(h\_fir))

phi\_iir\_freqz = np.unwrap(np.angle(h\_iir))

# Для FIR: phi = atan2(Imag, Real) с Real = 0.5\*(1+cos w), Imag = -0.5\*sin w

phi\_fir\_analytic = np.unwrap(np.arctan2(-np.sin(w), 1 + np.cos(w)))

# Для IIR: phi = -atan2(a\*sin w, 1 - a\*cos w)

phi\_iir\_analytic = np.unwrap(-np.arctan2(a \* np.sin(w), 1 - a \* np.cos(w)))

plt.figure(figsize=(10,6))

plt.plot(freqs, phi\_fir\_freqz, label="FIR freqz (числ.)")

plt.plot(freqs, phi\_fir\_analytic, '--', label="FIR аналитич. (-arctan)")

plt.title("Сравнение ФЧХ")

plt.xlabel("Частота, Гц")

plt.ylabel("Фаза, рад")

plt.plot(freqs, phi\_iir\_freqz, label="IIR freqz (числ.)")

plt.plot(freqs, phi\_iir\_analytic, '--', label="IIR аналитич. (-arctan)")

plt.legend()

plt.grid(True, linestyle=":", alpha=0.6)

plt.show()

1. Выводы по лабораторной работе

В ходе выполнения лабораторной работы были исследованы основные характеристики дискретных фильтров первого порядка – нерекурсивного (FIR) и рекурсивного (IIR). Для анализа использовались три типа сигналов: единичный импульс, единичный скачок и гармонический сигнал. Это позволило проследить, как фильтры реагируют на разные входные воздействия и как их поведение связано с аналитическими формулами.

Импульсная характеристика, полученная экспериментально, совпала с теоретической. Для FIR фильтра импульсная характеристика оказалась конечной длины и совпала с коэффициентами фильтра. Для IIR фильтра импульсная характеристика оказалась бесконечной, так как выход зависит от предыдущих выходных значений, и это хорошо видно на затухающей последовательности при подаче единичного импульса.

При подаче единичного скачка стало заметно различие переходных характеристик. FIR фильтр быстро выходит на постоянное значение, тогда как IIR фильтр постепенно накапливает отклик и стремится к установившемуся значению, зависящему от коэффициента обратной связи. Это подтверждает, что рекурсивные фильтры обладают «памятью» и реагируют более инерционно.

Наиболее интересной частью работы оказалось исследование частотных характеристик. При подаче гармонического сигнала на выходе сохранялась та же частота, но изменялись амплитуда и фаза. Это позволило построить амплитудно-частотные и фазо-частотные характеристики как экспериментально, так и теоретически. АЧХ показала, что фильтры по-разному усиливают или подавляют частоты: рекурсивный фильтр при нулевой частоте усиливает сигнал в 2 раза, а на высоких частотах его амплитуда снижается, в то время как нерекурсивный ведет себя более предсказуемо, усредняя соседние отсчёты.

При анализе ФЧХ стало понятно, что сдвиг по фазе возникает из-за того, что фильтр задерживает сигнал относительно входа. Это особенно хорошо видно на IIR фильтре, где фаза изменяется плавно по мере роста частоты.

В итоге можно сказать, что работа позволила связать формулы из теории с результатами моделирования. Импульсная и переходная характеристики, АЧХ и ФЧХ, полученные экспериментально, совпали с теоретическими выражениями.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Python Software Foundation. Python 3. Стандартная библиотека [Электронный ресурс]. – Режим доступа: <https://docs.python.org/3/library/math.html> (дата обращения: 11.09.2025).
2. Matplotlib Development Team. Matplotlib: Visualization with Python [Электронный ресурс]. – Режим доступа: <https://matplotlib.org/stable/index.html> (дата обращения: 11.09.2025).
3. Аграновский А. В. Методические указания к лабораторной работе № 1 «Сигналы и фильтры» по дисциплине «Цифровая обработка и передача сигналов». – Санкт-Петербург: ГУАП, 2025.