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 №2 |
| Восстановление непрерывного сигнала по дискретным отсчётам. Теорема Котельникова |
| 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

Практическое исследование этапов аналого-цифрового преобразования сигналов с использованием современных средств имитационного моделирования. Сравнительный анализ аналогового, дискретного и цифрового сигналов. Приобретение практических навыков применения программных средств имитационного моделирования цифровых сигналов.

1. Задачи

Для выполнения этой лабораторной работы необходимо:

1. Выполнить имитационное моделирование аналогового гармонического сигнала одной частоты, описываемого функцией:

𝑥(𝑡) = 𝐴0 + 𝐴cos(2𝜋𝑓𝑡+ 𝜙) (1)

на временном интервале 𝑡 ∈ [𝑡𝑚𝑖𝑛; 𝑡𝑚𝑎𝑥] с использованием символьных переменных;

2. Построить график функции, описывающей аналоговый сигнал;

3. Выполнить моделирование аналого-цифрового преобразования с частотой дискретизации 𝑓𝑑 и разрядностью 𝑏. Кодирование сигнала реализовать с помощью прямого, обратного или дополнительного кода;

4. Построить графики соответствующих функций для дискретного, квантованного и цифрового сигналов;

5. Оценить параметры шума квантования сигнала, построить гистограмму статистического распределения абсолютной погрешности квантования и сопоставить полученные результаты с теоретическими значениями.

После выполнения экспериментальной части необходимо ответить на предложенные контрольные вопросы для закрепления пройденного материала и установления взаимосвязи между полученными результатами практических работ и теоретическими знаниями.

Результаты работы рекомендуется оформить в виде отчета, в котором должна содержаться следующая информация: цель работы; решённые в процессе её достижения задачи; основные математические выражения, использованные при решении задач; текст программы или схема моделирования, результаты моделирования в виде графиков и заключение, позволяющее сделать вывод о сопоставимости результатов практической работы с теоретическими сведениями.

Для моего варианта параметры: А0 = 6 В, А = 5 В,= 3 Гц, φ=π/3.

1. Теоретические сведения

В данной лабораторной работе исследуются процессы аналого-цифрового преобразования (АЦП), включающие дискретизацию, квантование и цифровое кодирование сигнала.

Исследуемый сигнал имеет гармоническую форму, которая задается по формуле (1) .

Дискретизация по времени заключается в замене непрерывного времени t на дискретные моменты: tn=nTd где Td =1/𝑓𝑑 – период дискретизации, а 𝑓𝑑 – частота дискретизации. Согласно теореме Котельникова, для точного восстановления сигнала из его отсчётов частота дискретизации должна быть более чем в два раза выше максимальной частоты спектра:

(2)

При нарушении этого условия возникает эффект наложения спектров, который приводит к искажению сигнала после восстановления.

Следующий этап – квантование по уровню, то есть замена непрерывного множества возможных значений сигнала конечным числом уровней. Для АЦП с разрядностью b количество уровней квантования определяется выражением:

N = 2b. (3)

Шаг квантования q задаётся шириной диапазона измеряемых значений:

(4)

Каждое значение сигнала округляется до ближайшего уровня квантования, что приводит к появлению ошибки квантования, определяемой как:

ε(n) = xk(n) − xd(n) (5)

где xd(n) – дискретные значения исходного сигнала, xk(n) – квантованные значения.

Максимальная по модулю ошибка квантования равна:

(6)

Если предположить, что ошибка распределена равномерно на интервале [−q/2, q/2] её теоретическая дисперсия вычисляется по формуле:

(7)

Среднее значение ошибки при симметричном распределении равно нулю.

На заключительном этапе выполняется цифровое кодирование квантованных уровней. В работе рассматриваются три основных метода кодирования: прямой, обратный и дополнительный коды. Для прямого кода старший разряд представляет знак числа, а оставшиеся b−1 разрядов – его абсолютное значение. В обратном коде отрицательные значения формируются инверсией всех разрядов положительного числа. В дополнительном коде к обратному представлению прибавляется единица, что позволяет исключить двойственность нуля и упрощает реализацию арифметических операций.

Таким образом, в ходе лабораторной работы выполняются три взаимосвязанных этапа: дискретизация аналогового сигнала по времени, квантование его значений по уровню и кодирование полученных отсчётов в цифровую форму. В процессе анализа рассчитываются параметры шага квантования, погрешности и дисперсии, строятся графики аналогового, дискретного, квантованного и цифрового сигналов. Полученные зависимости позволяют экспериментально подтвердить влияние частоты дискретизации и разрядности АЦП на точность представления сигнала и соответствие результатов теоретическим оценкам.

1. Ход выполнения лабораторной работы

Сначала был задан аналоговый гармонический сигнал в соответствии с индивидуальным вариантом по формуле (1).

На первом этапе была выполнена дискретизация по времени. Период дискретизации рассчитывался по формуле (2). Где частота дискретизации = 64. Проверка теоремы Котельникова (2) показала, что условие выполняется, так как 64 > 6. Следовательно, искажения, связанные с наложением спектров, отсутствуют. График аналогового сигнала представлен на рисунке 1, а дискретного – на рисунке 2. На втором рисунке наглядно показаны равномерные отсчёты сигнала с шагом Td =0.015625 с, что подтверждает корректность временной выборки.

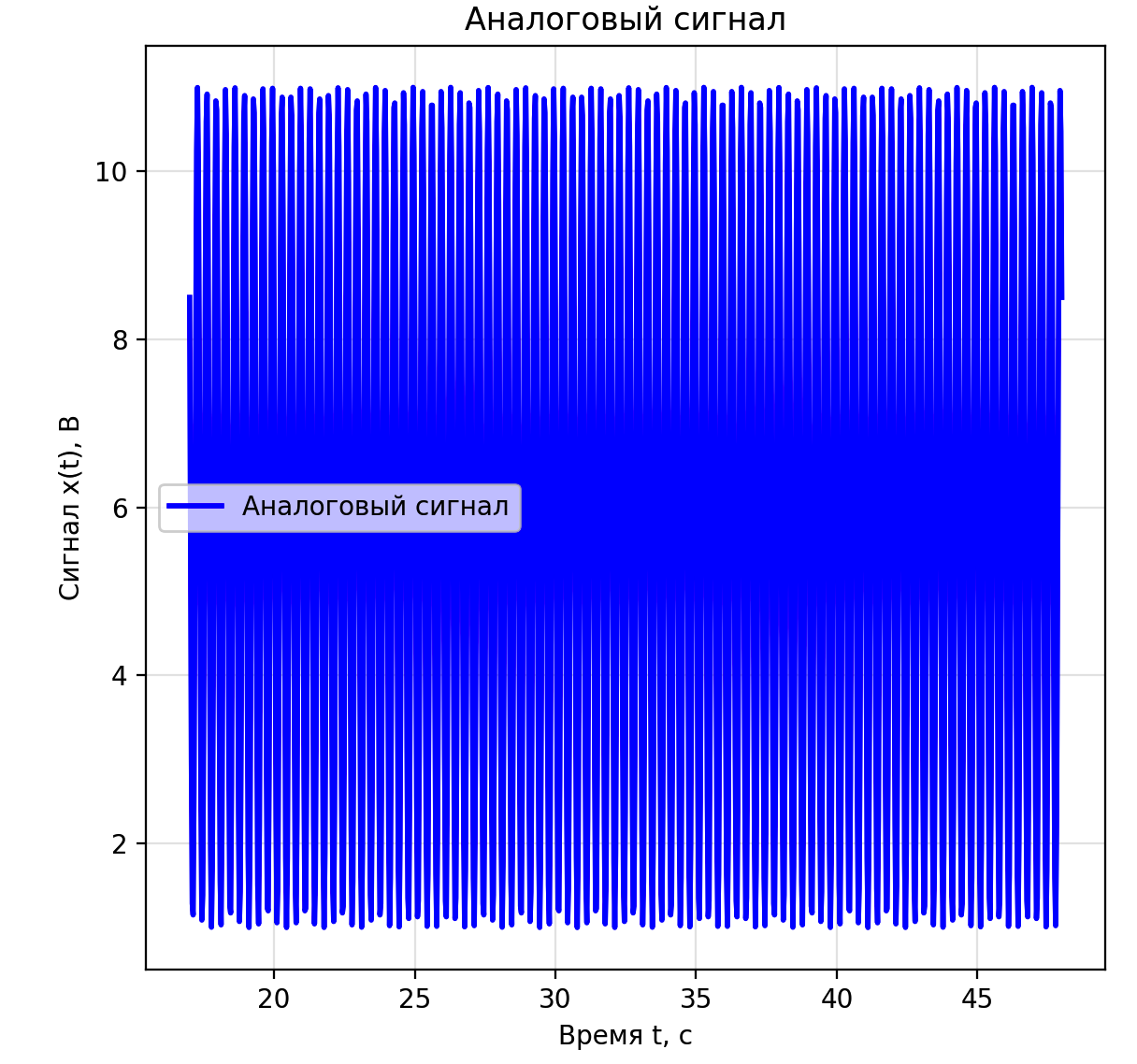


Рисунок 1 – График аналогового сигнала

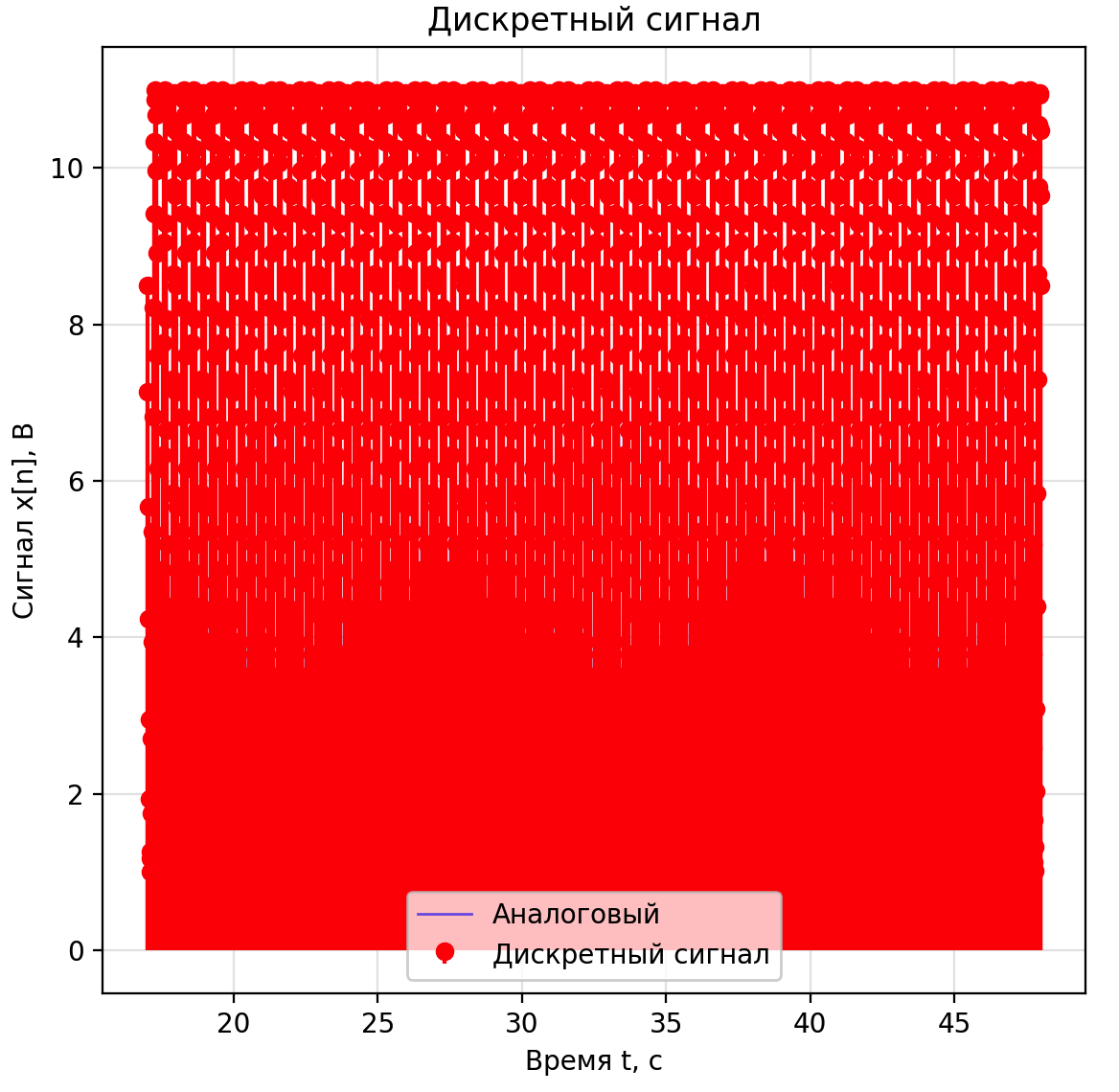


Рисунок 2 – График дискретного сигнала

На рисунках 3 – 4 представлены графики с меньшим временным промежутком для наглядности:

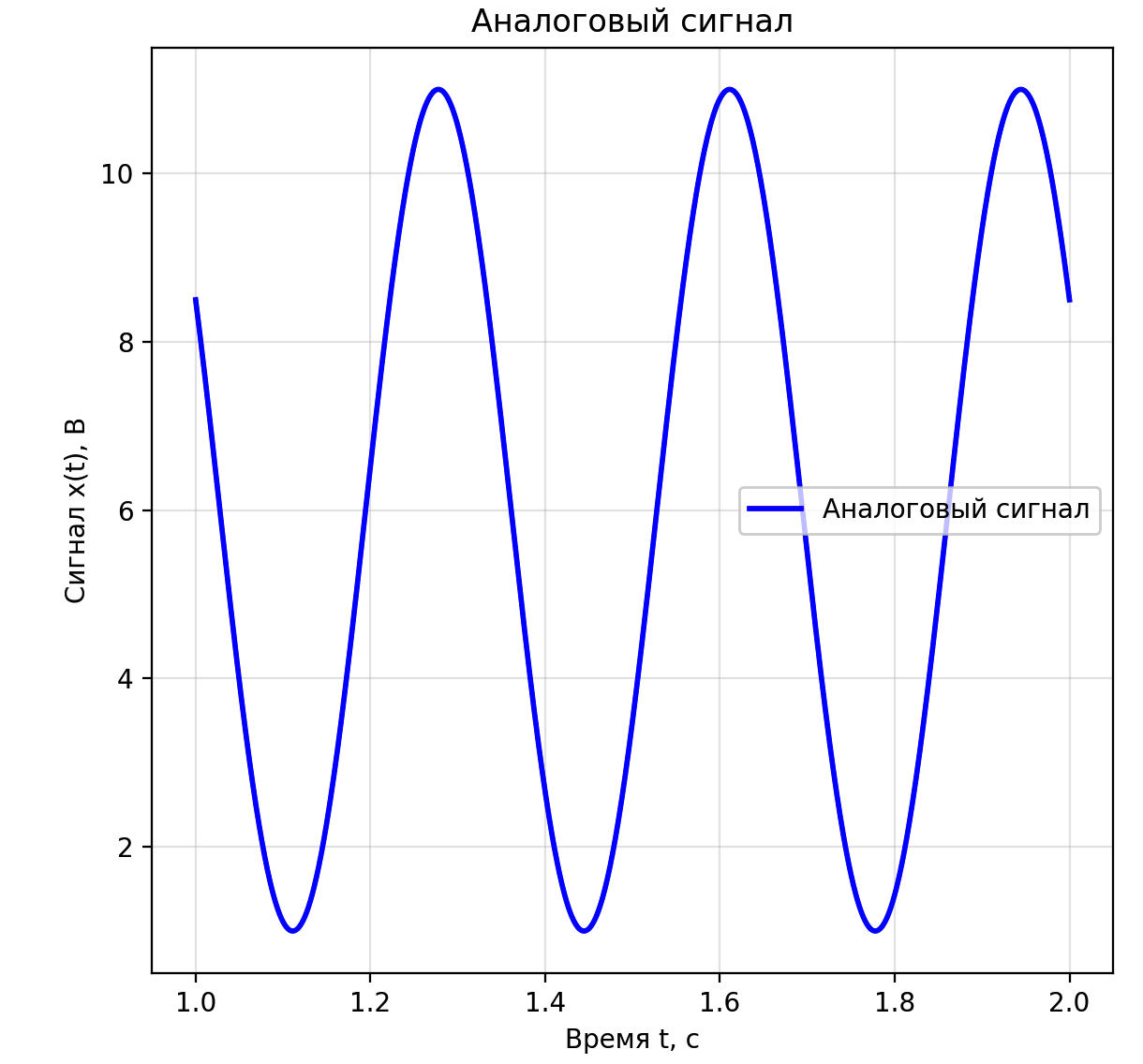
****

Рисунок 3 – График аналогового сигнала на меньшем промежутке

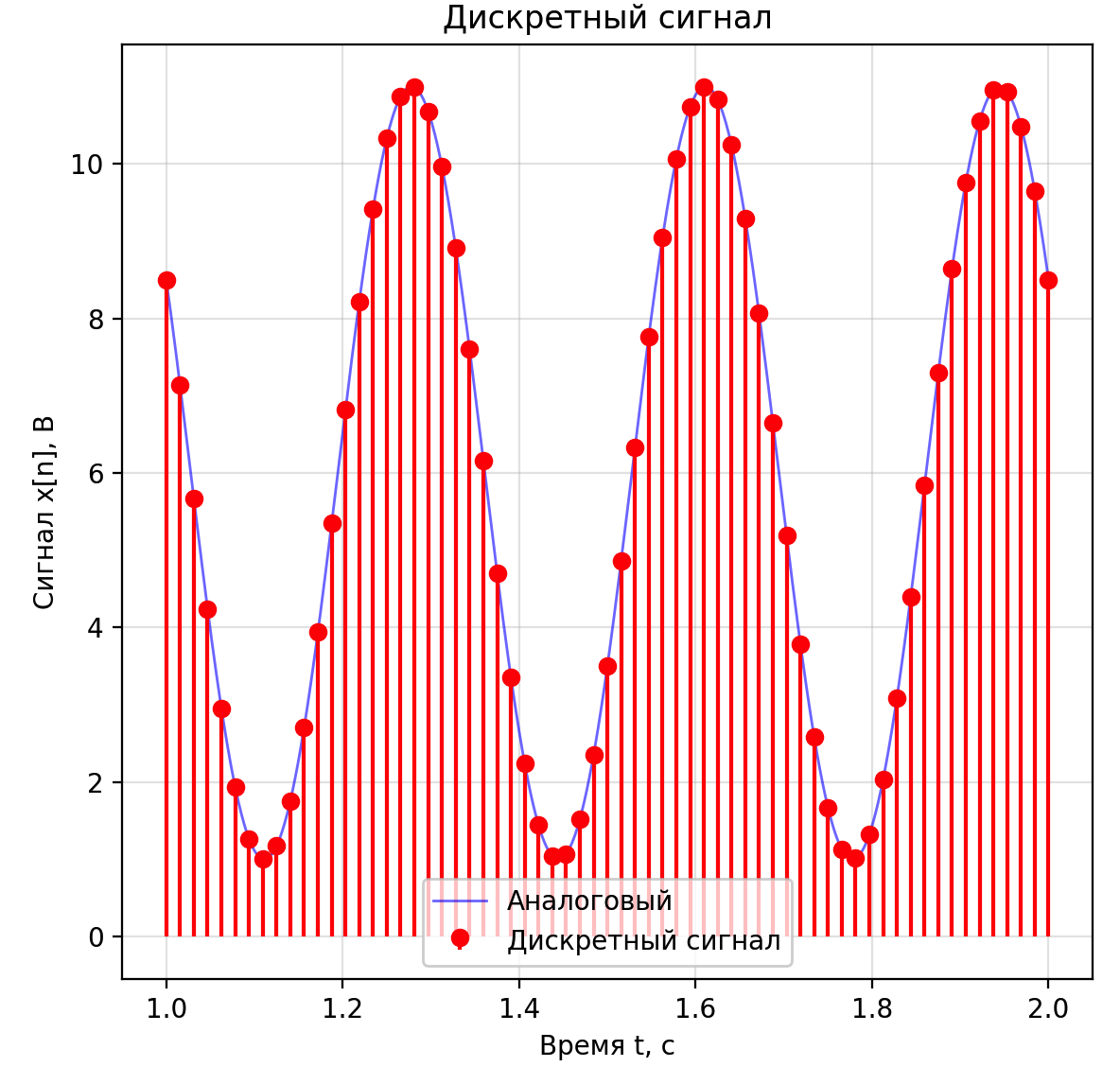
****

Рисунок 4 – График дискретного сигнала на меньшем промежутке

На втором этапе было проведено квантование сигнала по уровню. Количество уровней квантования определялось по формуле (3). Где b=4 бит, что даёт N=16 уровней квантования. Шаг квантования вычислялся по формуле (4) и был равен q = 0.6667В

Квантование производилось методом округления до ближайшего уровня. В результате для каждого отсчёта было найдено ближайшее значение, что породило ошибку квантования, определяемую формулой (5).

Максимальная теоретическая ошибка квантования вычислялось по формуле (6) и была равна = 0.3333. На практике полученная максимальная ошибка составила ±0.3205 В, что близко к теоретическому значению. График квантованного сигнала приведён на рисунке 5. На нём показано, как дискретные значения выравниваются по уровням квантования, отмеченным горизонтальными линиями. На рисунке 6 представлен график на меньшем промежутке времени.

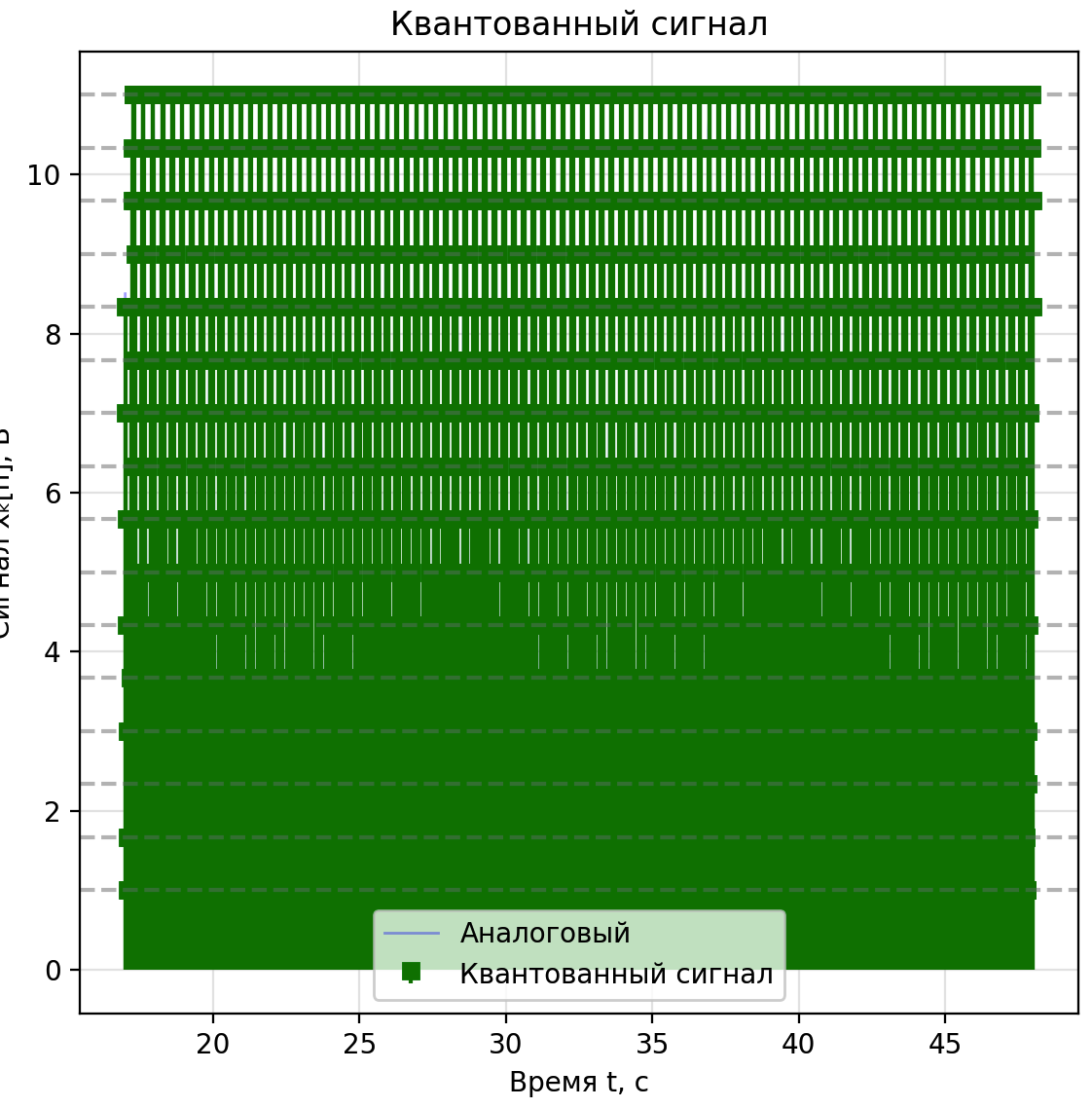


Рисунок 5 – График квантованного сигнала

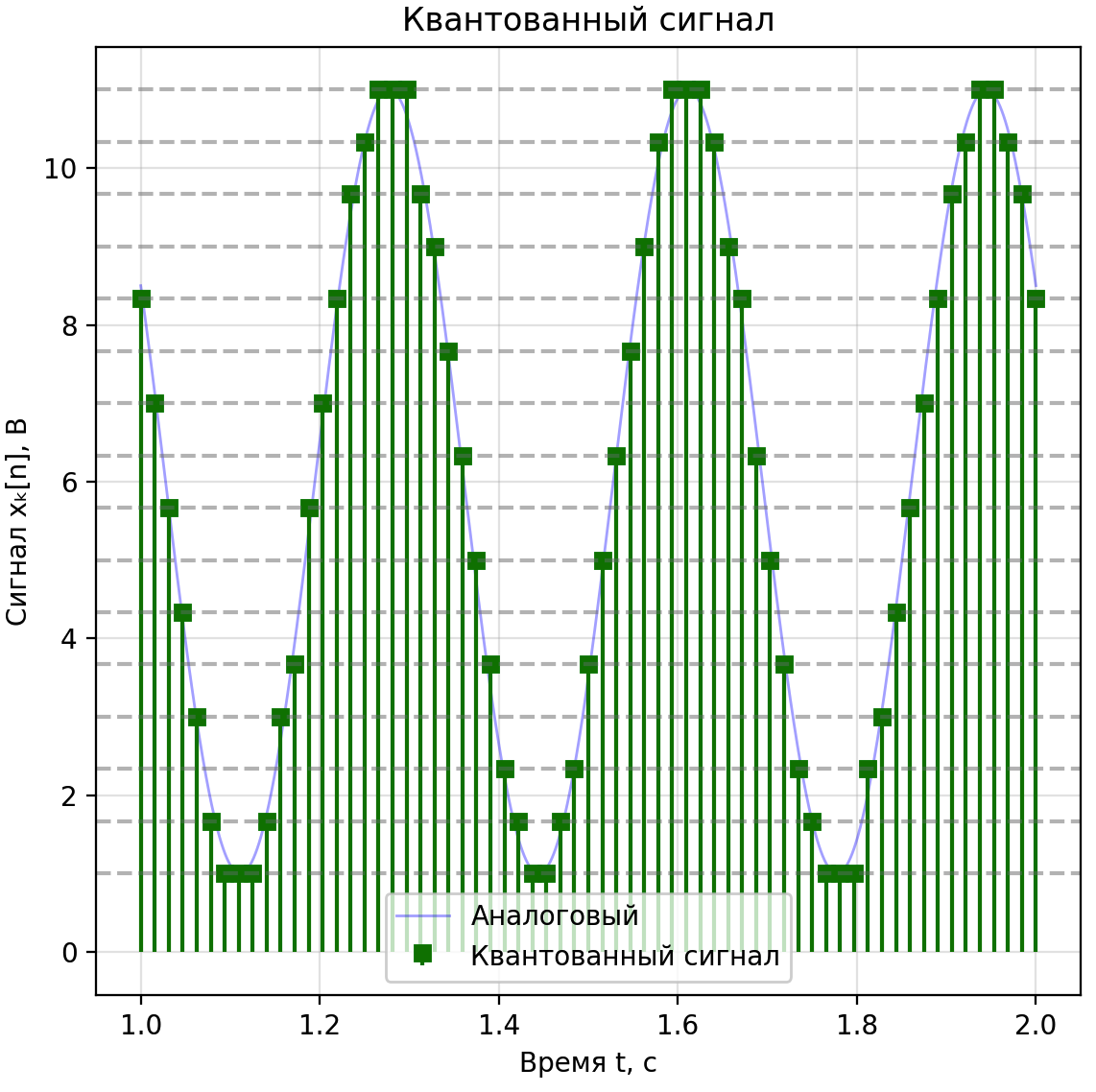


Рисунок 6 – График квантованного сигнала на меньшем промежутке

На третьем этапе было реализовано цифровое кодирование квантованных уровней. Использовался прямой код, при котором старший бит обозначает знак, а оставшиеся b-1 разрядов – модуль числа. Пример двоичных кодов уровней приведён на рисунке 7, на рисунке 8 показан на меньшем промежутке, где рядом с уровнями отображаются соответствующие коды вида «код (индекс)». Такое представление используется в цифровых АЦП для передачи сигналов в цифровую форму.

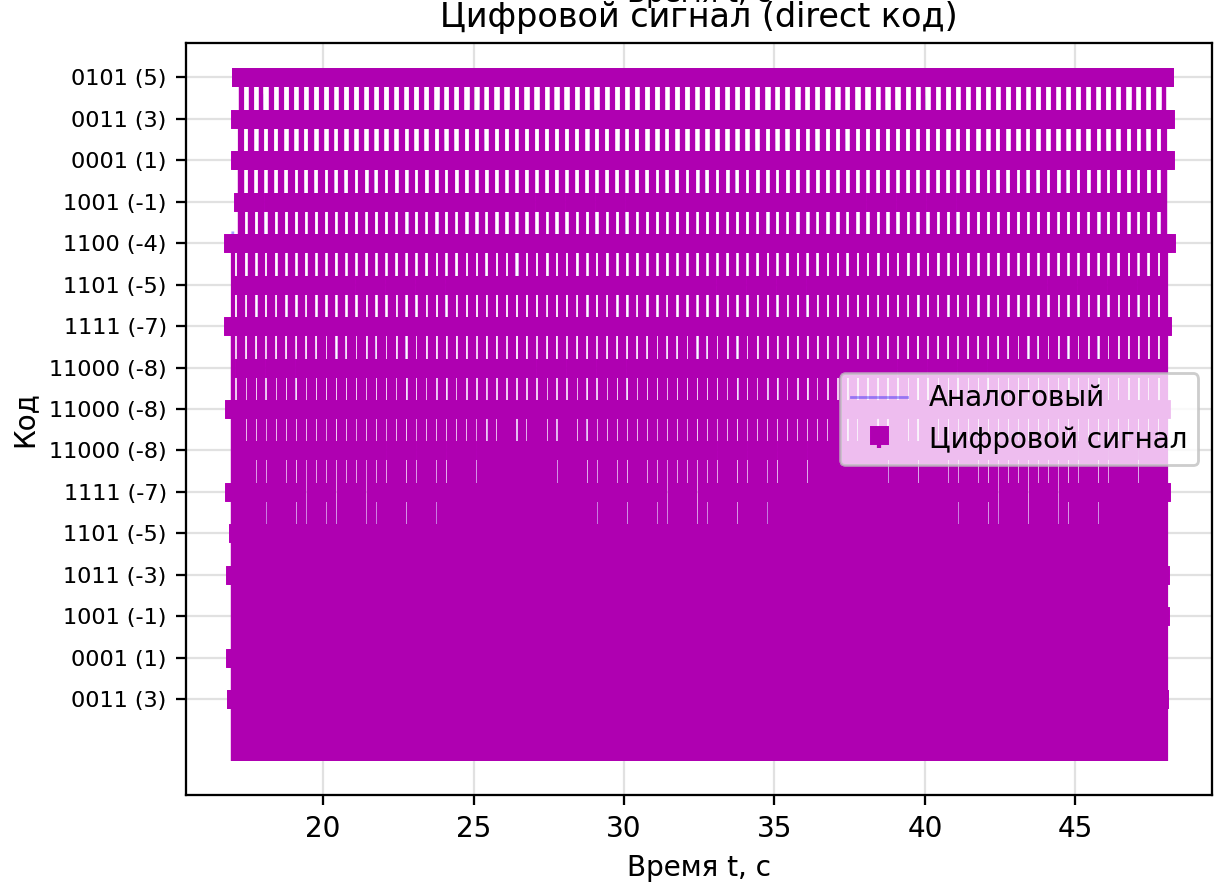


Рисунок 7 – График цифрового сигнала

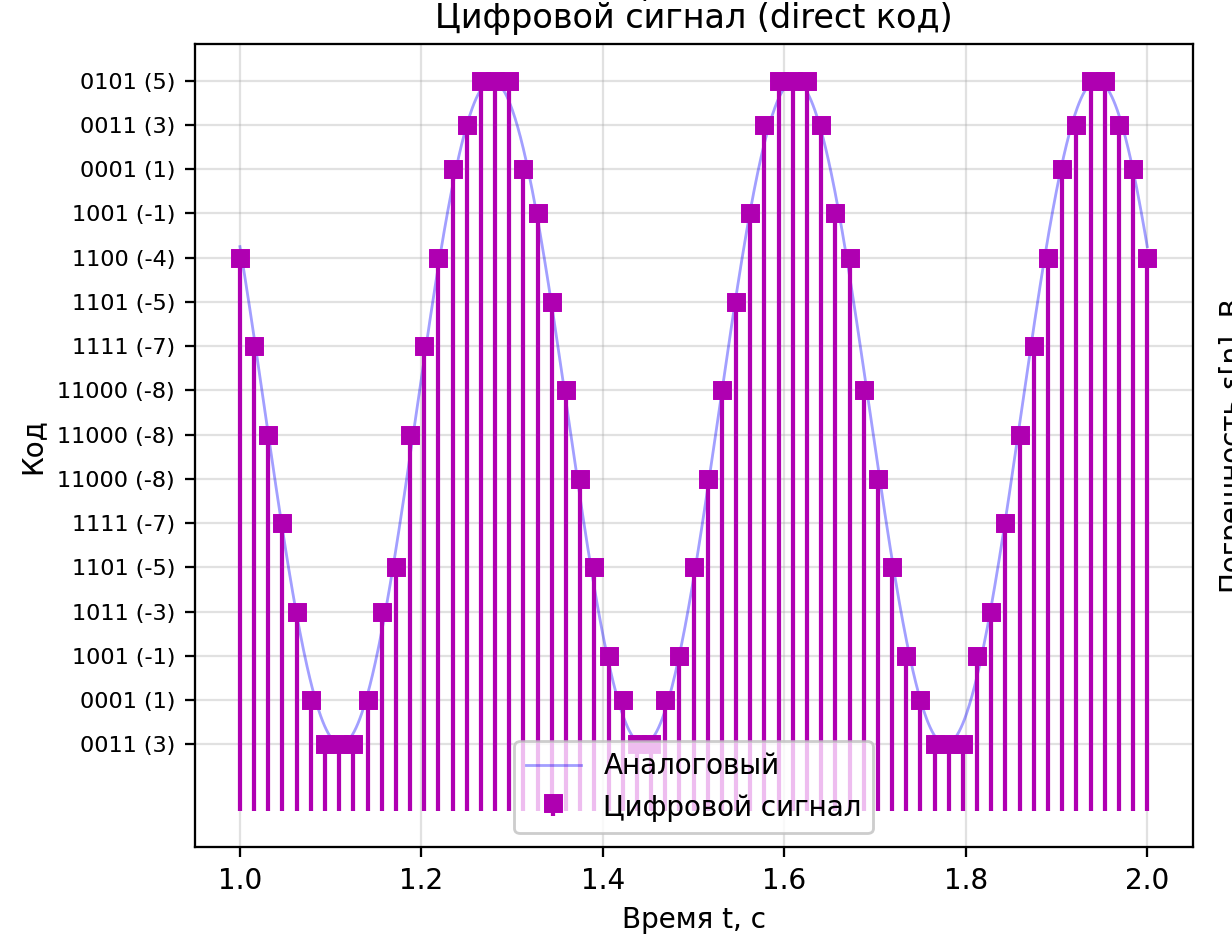


Рисунок 8 – График цифрового сигнала на меньшем промежутке

На четвёртом этапе была выполнена оценка погрешности квантования. Абсолютная ошибка для каждого отсчёта рассчитывалась по формуле (5), после чего построен график зависимости ε(n) от времени, представленный на рисунках 9 и 10. Из графика видно, что значения ошибки находятся в пределах ±q/2±q/2, что подтверждает корректность процесса квантования.

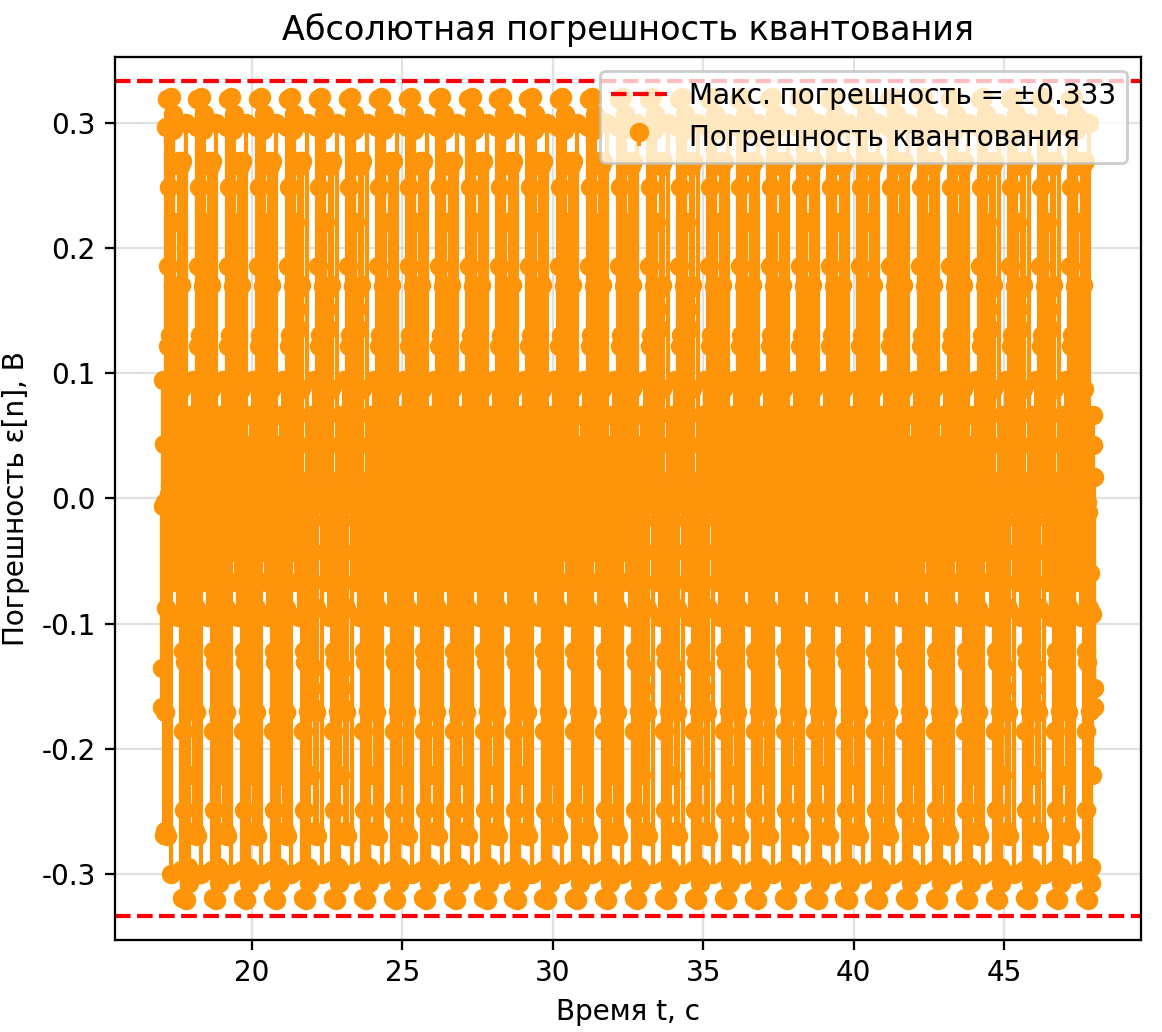


Рисунок 9 – График погрешности квантования

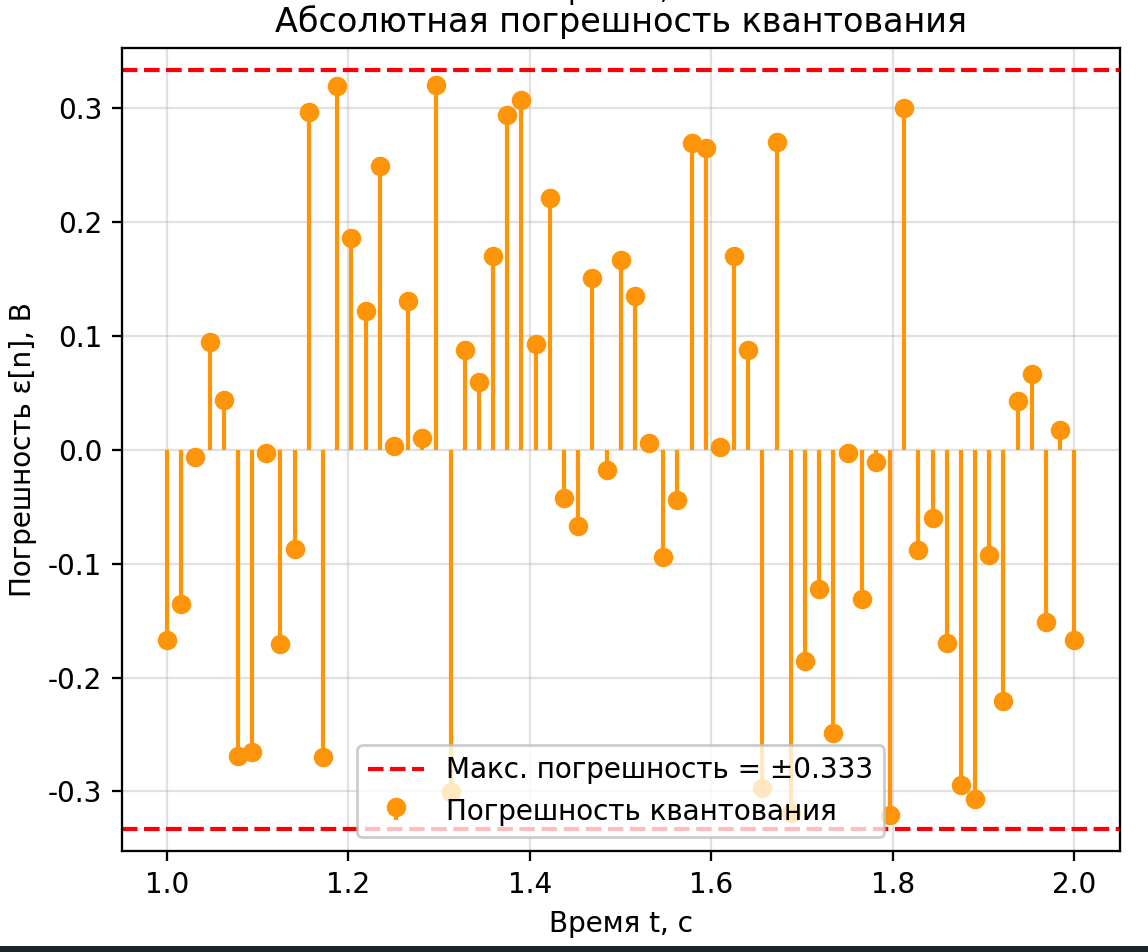


Рисунок 10 – График погрешности квантования на меньшем промежутке

Теоретическая дисперсия шума квантования вычислялась по формуле (7) равняется σε2 = 0.0370 В2.

Экспериментальная дисперсия составила 0.0351 В2, что практически совпадает с теоретическим расчётом, а среднее значение ошибки оказалось близким к нулю (-0.0025 В). Это свидетельствует о равномерном распределении погрешности и отсутствии систематического смещения.

Для подтверждения этого вывода была построена гистограмма распределения погрешности квантования, представлена на рисунке 11.

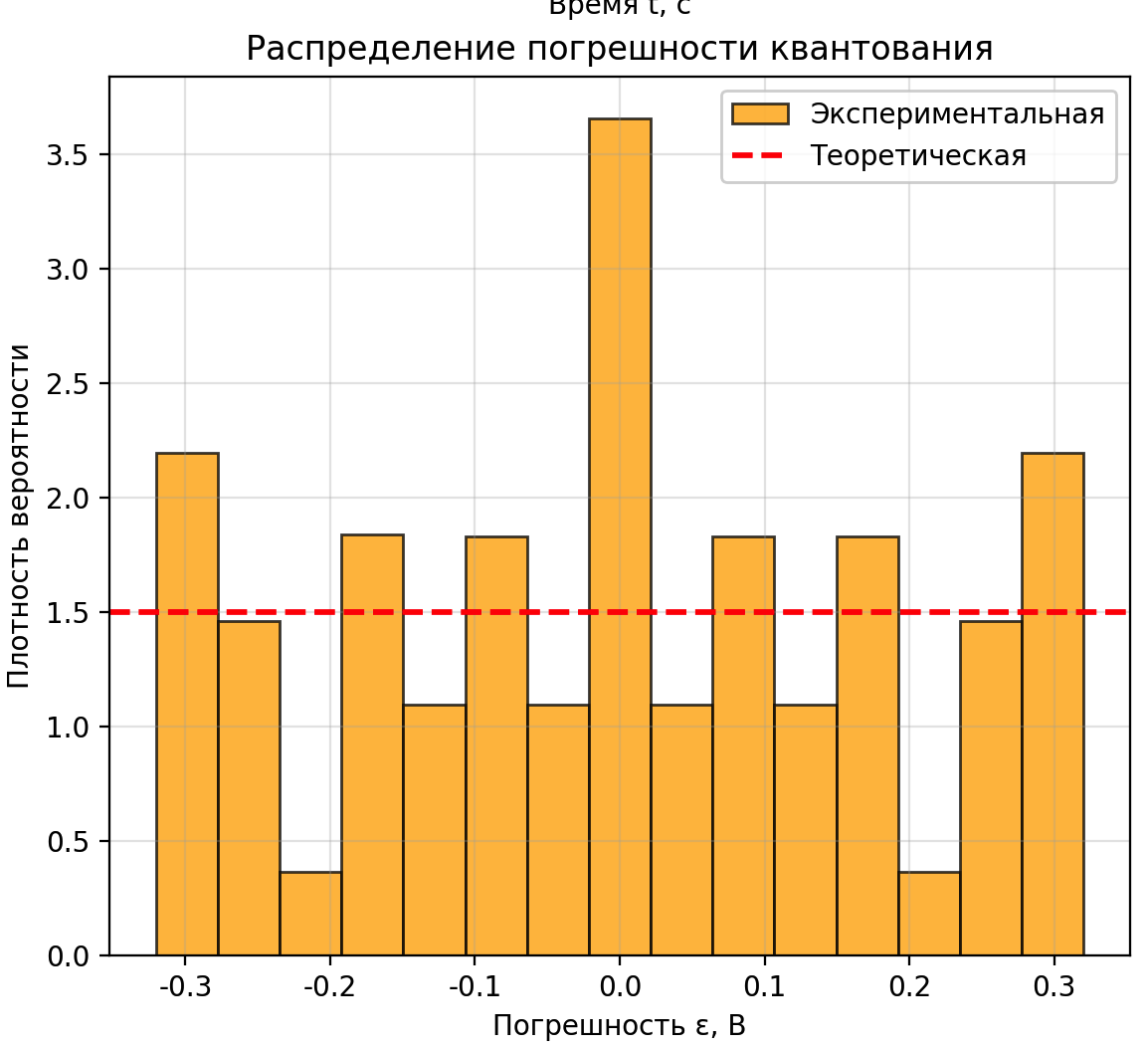


Рисунок 11 – Распределение погрешности

1. Листинг программы

Ниже представлен код программы на языке Python:

import numpy as np

import matplotlib.pyplot as plt

from matplotlib.patches import Rectangle

import matplotlib.patches as patches

# Настройка русских шрифтов для matplotlib

plt.rcParams['font.family'] = ['DejaVu Sans', 'Arial Unicode MS', 'Tahoma']

plt.rcParams['axes.unicode\_minus'] = False

class SignalProcessor:

"""Класс для обработки и преобразования сигналов"""

def \_\_init\_\_(self, A, A0, f, phi, t\_min, t\_max, fd, b, code\_type='direct'):

"""

Инициализация параметров сигнала

Параметры:

A - амплитуда косинуса, В

A0 - постоянная составляющая, В

f - частота сигнала, Гц

phi - начальная фаза, рад

t\_min - начальное время, с

t\_max - конечное время, с

fd - частота дискретизации, Гц

b - разрядность АЦП, бит

code\_type - тип кодирования ('direct', 'inverse', 'complement')

"""

self.A = A

self.A0 = A0

self.f = f

self.phi = phi

self.t\_min = t\_min

self.t\_max = t\_max

self.fd = fd

self.b = b

self.code\_type = code\_type

# Вычисляемые параметры

self.Td = 1 / fd # Период дискретизации

self.N = 2\*\*b # Количество уровней квантования

if fd <= 2 \* f:

print(f"ВНИМАНИЕ: Частота дискретизации {fd} Гц не удовлетворяет теореме Найквиста!")

print(f"Минимальная частота дискретизации должна быть > {2\*f} Гц")

def analog\_signal(self, t):

"""

Генерация аналогового сигнала

x(t) = A0 + A\*cos(2πft + φ)

"""

return self.A0 + self.A \* np.cos(2 \* np.pi \* self.f \* t + self.phi)

def generate\_time\_arrays(self):

"""Генерация массивов времени для аналогового и дискретного сигналов"""

# Аналоговый сигнал - высокое разрешение для плавной кривой

t\_analog = np.linspace(self.t\_min, self.t\_max, 1000)

# Дискретный сигнал - отсчеты через период дискретизации

num\_samples = int((self.t\_max - self.t\_min) \* self.fd) + 1

t\_discrete = np.linspace(self.t\_min, self.t\_max, num\_samples)

return t\_analog, t\_discrete

def discretize\_signal(self, t\_discrete):

"""Дискретизация сигнала по времени"""

return self.analog\_signal(t\_discrete)

def quantize\_signal(self, x\_discrete):

"""

Квантование сигнала по уровню с округлением

"""

# Определение диапазона сигнала

x\_max = max(np.max(x\_discrete), self.A0 + self.A)

x\_min = min(np.min(x\_discrete), self.A0 - self.A)

# Шаг квантования

q = (x\_max - x\_min) / (self.N - 1)

# Уровни квантования

levels = np.linspace(x\_min, x\_max, self.N)

# Квантование с округлением

x\_quantized = np.zeros\_like(x\_discrete)

for i, x in enumerate(x\_discrete):

# Находим ближайший уровень квантования

idx = np.argmin(np.abs(levels - x))

x\_quantized[i] = levels[idx]

return x\_quantized, levels, q

def encode\_signal(self, x\_quantized, levels):

"""

Цифровое кодирование квантованного сигнала

"""

# Создание массива индексов уровней

codes = np.arange(-self.N//2, self.N//2)

# Поиск соответствующих кодов для каждого квантованного значения

digital\_codes = np.zeros(len(x\_quantized), dtype=int)

for i, x in enumerate(x\_quantized):

# Находим индекс ближайшего уровня

level\_idx = np.argmin(np.abs(levels - x))

digital\_codes[i] = codes[level\_idx]

# Преобразование в выбранный тип кода

if self.code\_type == 'direct':

encoded = self.\_direct\_code(digital\_codes)

elif self.code\_type == 'inverse':

encoded = self.\_inverse\_code(digital\_codes)

elif self.code\_type == 'complement':

encoded = self.\_complement\_code(digital\_codes)

else:

raise ValueError("Неизвестный тип кодирования")

return digital\_codes, encoded

def \_direct\_code(self, codes):

"""Прямой код"""

result = []

for code in codes:

if code >= 0:

binary = format(code, f'0{self.b}b')

else:

binary = '1' + format(abs(code), f'0{self.b-1}b')

result.append(binary)

return result

def \_inverse\_code(self, codes):

"""Обратный код"""

result = []

for code in codes:

if code >= 0:

binary = format(code, f'0{self.b}b')

else:

# Инвертируем все биты положительного числа

positive\_binary = format(abs(code), f'0{self.b-1}b')

inverted = ''.join('1' if bit == '0' else '0' for bit in positive\_binary)

binary = '1' + inverted

result.append(binary)

return result

def \_complement\_code(self, codes):

"""Дополнительный код"""

result = []

for code in codes:

if code >= 0:

binary = format(code, f'0{self.b}b')

else:

# Дополнительный код = 2^b - |code|

complement\_value = 2\*\*self.b + code

binary = format(complement\_value, f'0{self.b}b')

result.append(binary)

return result

def calculate\_quantization\_error(self, x\_discrete, x\_quantized):

"""Вычисление абсолютной погрешности квантования"""

return x\_quantized - x\_discrete

def plot\_all\_signals(self):

"""Построение всех графиков сигналов"""

# Генерация временных массивов

t\_analog, t\_discrete = self.generate\_time\_arrays()

# Генерация сигналов

x\_analog = self.analog\_signal(t\_analog)

x\_discrete = self.discretize\_signal(t\_discrete)

x\_quantized, levels, q = self.quantize\_signal(x\_discrete)

digital\_codes, encoded\_signal = self.encode\_signal(x\_quantized, levels)

# Вычисление погрешности

error = self.calculate\_quantization\_error(x\_discrete, x\_quantized)

# Создание фигуры с подграфиками

fig, axes = plt.subplots(2, 3, figsize=(18, 12))

fig.suptitle('Лабораторная работа №3: Непрерывные, дискретные и цифровые сигналы',

fontsize=16, fontweight='bold')

# 1. Аналоговый сигнал

axes[0, 0].plot(t\_analog, x\_analog, 'b-', linewidth=2, label='Аналоговый сигнал')

axes[0, 0].set\_title('Аналоговый сигнал')

axes[0, 0].set\_xlabel('Время t, с')

axes[0, 0].set\_ylabel('Сигнал x(t), В')

axes[0, 0].grid(True, alpha=0.3)

axes[0, 0].legend()

# 2. Дискретный сигнал

axes[0, 1].plot(t\_analog, x\_analog, 'b-', linewidth=1, alpha=0.5, label='Аналоговый')

axes[0, 1].stem(t\_discrete, x\_discrete, linefmt='r-', markerfmt='ro',

basefmt=' ', label='Дискретный сигнал')

axes[0, 1].set\_title('Дискретный сигнал')

axes[0, 1].set\_xlabel('Время t, с')

axes[0, 1].set\_ylabel('Сигнал x[n], В')

axes[0, 1].grid(True, alpha=0.3)

axes[0, 1].legend()

# 3. Квантованный сигнал

axes[0, 2].plot(t\_analog, x\_analog, 'b-', linewidth=1, alpha=0.3, label='Аналоговый')

axes[0, 2].stem(t\_discrete, x\_quantized, linefmt='g-', markerfmt='gs',

basefmt=' ', label='Квантованный сигнал')

# Добавляем уровни квантования

for level in levels:

axes[0, 2].axhline(y=level, color='gray', linestyle='--', alpha=0.5)

axes[0, 2].set\_title('Квантованный сигнал')

axes[0, 2].set\_xlabel('Время t, с')

axes[0, 2].set\_ylabel('Сигнал xₖ[n], В')

axes[0, 2].grid(True, alpha=0.3)

axes[0, 2].legend()

# 4. Цифровой сигнал

axes[1, 0].plot(t\_analog, x\_analog, 'b-', linewidth=1, alpha=0.3, label='Аналоговый')

axes[1, 0].stem(t\_discrete, x\_quantized, linefmt='m-', markerfmt='ms',

basefmt=' ', label='Цифровой сигнал')

# Добавляем коды на оси Y

y\_ticks = levels

y\_labels = [f'{encoded\_signal[i]} ({digital\_codes[i]})'

for i in range(len(levels))]

axes[1, 0].set\_yticks(y\_ticks)

axes[1, 0].set\_yticklabels(y\_labels, fontsize=8)

axes[1, 0].set\_title(f'Цифровой сигнал ({self.code\_type} код)')

axes[1, 0].set\_xlabel('Время t, с')

axes[1, 0].set\_ylabel('Код')

axes[1, 0].grid(True, alpha=0.3)

axes[1, 0].legend()

# 5. Погрешность квантования

axes[1, 1].stem(t\_discrete, error, linefmt='orange', markerfmt='o',

basefmt=' ', label='Погрешность квантования')

axes[1, 1].axhline(y=q/2, color='red', linestyle='--',

label=f'Макс. погрешность = ±{q/2:.3f}')

axes[1, 1].axhline(y=-q/2, color='red', linestyle='--')

axes[1, 1].set\_title('Абсолютная погрешность квантования')

axes[1, 1].set\_xlabel('Время t, с')

axes[1, 1].set\_ylabel('Погрешность ε[n], В')

axes[1, 1].grid(True, alpha=0.3)

axes[1, 1].legend()

# 6. Гистограмма распределения погрешности

axes[1, 2].hist(error, bins=15, density=True, alpha=0.7, color='orange',

edgecolor='black', label='Экспериментальная')

# Теоретическое равномерное распределение

theoretical\_height = 1 / q

axes[1, 2].axhline(y=theoretical\_height, color='red', linestyle='--',

linewidth=2, label='Теоретическая')

axes[1, 2].set\_title('Распределение погрешности квантования')

axes[1, 2].set\_xlabel('Погрешность ε, В')

axes[1, 2].set\_ylabel('Плотность вероятности')

axes[1, 2].grid(True, alpha=0.3)

axes[1, 2].legend()

plt.tight\_layout()

plt.show()

# Вывод статистики

self.print\_statistics(q, error)

def print\_statistics(self, q, error):

"""Вывод статистических характеристик"""

print("\n" + "="\*60)

print("СТАТИСТИЧЕСКИЕ ХАРАКТЕРИСТИКИ СИГНАЛА")

print("="\*60)

print(f"Параметры сигнала:")

print(f" Амплитуда A = {self.A} В")

print(f" Постоянная составляющая A₀ = {self.A0} В")

print(f" Частота f = {self.f} Гц")

print(f" Начальная фаза φ = {self.phi} рад = {self.phi\*180/np.pi:.1f}°")

print(f" Временной интервал: [{self.t\_min}, {self.t\_max}] с")

print(f"\nПараметры дискретизации и квантования:")

print(f" Частота дискретизации fₐ = {self.fd} Гц")

print(f" Период дискретизации Tₐ = {self.Td} с")

print(f" Разрядность АЦП b = {self.b} бит")

print(f" Количество уровней квантования N = {self.N}")

print(f" Шаг квантования q = {q:.4f} В")

print(f" Тип кодирования: {self.code\_type}")

print(f"\nСтатистика погрешности квантования:")

print(f" Теоретическая максимальная погрешность = ±{q/2:.4f} В")

print(f" Экспериментальная максимальная погрешность = ±{np.max(np.abs(error)):.4f} В")

print(f" Теоретическая дисперсия = {q\*\*2/12:.6f} В²")

print(f" Экспериментальная дисперсия = {np.var(error):.6f} В²")

print(f" Среднее значение погрешности = {np.mean(error):.6f} В")

print(f" СКО погрешности = {np.std(error):.6f} В")

# Проверка теоремы Найквиста-Котельникова

nyquist\_freq = 2 \* self.f

print(f"\nПроверка теоремы Найквиста-Котельникова:")

print(f" Минимальная частота дискретизации = {nyquist\_freq} Гц")

print(f" Используемая частота дискретизации = {self.fd} Гц")

if self.fd > nyquist\_freq:

print(f" Условие выполнено (fₐ > 2fₘₐₓ)")

else:

print(f" УСЛОВИЕ НЕ ВЫПОЛНЕНО! Возможны искажения сигнала.")

def main():

"""Основная функция программы"""

print("Лабораторная работа №1")

print("Непрерывные, дискретные и цифровые сигналы")

print("="\*60)

# Параметры моего варианта

# t\_min = 17 с; t\_max = 48 с; A = 5 В; A0 = 6 В; f = 3 Гц; φ = π/3; Код: прямой

A = 5 # Амплитуда, В

A0 = 6 # Постоянная составляющая, В

f = 3 # Частота, Гц

phi = np.pi/3 # Начальная фаза, рад

t\_min = 17 # Начальное время, с

t\_max = 48 # Конечное время, с

fd = 64 # Частота дискретизации, Гц

b = 4 # Разрядность АЦП, бит

code\_type = 'direct' # Тип кодирования

# Создание экземпляра обработчика сигналов

processor = SignalProcessor(A, A0, f, phi, t\_min, t\_max, fd, b, code\_type)

# Построение всех графиков и вывод статистики

processor.plot\_all\_signals()

if \_\_name\_\_ == "\_\_main\_\_":

main()

1. Выводы по лабораторной работе

В ходе выполнения лабораторной работы был исследован процесс аналого-цифрового преобразования гармонического сигнала, включающий три основных этапа: дискретизацию по времени, квантование по уровню и цифровое кодирование отсчётов. Исследуемый сигнал имел форму косинусоиды, задаваемой выражением по формуле (1), с параметрами А0 = 6 В, А = 5 В,= 3 Гц, φ=π/3.

На первом этапе была выполнена дискретизация сигнала с частотой = 64, что значительно превышает удвоенную максимальную частоту спектра. Тем самым было выполнено условие теоремы Котельникова, гарантирующее отсутствие наложения спектров и возможность точного восстановления исходного сигнала из его отсчётов. Графики аналогового и дискретного сигналов подтвердили корректность временной выборки и равномерность распределения отсчётов.

На втором этапе проведено квантование по уровню при разрядности b=4 бита, что соответствует 16 уровням квантования. Шаг квантования составил q=0,6667 В. Анализ результатов показал, что максимальная экспериментальная погрешность квантования =0,3205 В не превышает теоретическое значение = 0,3333 В. Это свидетельствует о корректной реализации алгоритма квантования и равномерном распределении уровней по диапазону изменения сигнала.

На третьем этапе было реализовано цифровое кодирование квантованных уровней в прямом двоичном коде. Для каждого квантованного значения был получен соответствующий цифровой код, отображённый на графике цифрового сигнала. Такое представление соответствует формату работы реальных АЦП и обеспечивает дальнейшую цифровую обработку.

Результаты статистического анализа показали, что теоретическая дисперсия ошибки квантования = 0,0370 В² практически совпадает с экспериментальной = 0,0351 В², а среднее значение ошибки близко к нулю (-0,0025В). Это указывает на отсутствие систематических искажений и подтверждает равномерный характер распределения ошибки, что также видно на гистограмме распределения погрешности.

Таким образом, проведённое исследование подтвердило теоретические положения. Было экспериментально показано, что при соблюдении условий теоремы Котельникова и достаточной разрядности АЦП аналоговый гармонический сигнал может быть точно представлен и передан в цифровой форме. Полученные результаты демонстрируют правильность этапов дискретизации, квантования и кодирования, а также соответствие практических данных теоретическим оценкам.

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: 09.10.2025).
2. Matplotlib Development Team. Matplotlib: Visualization with Python [Электронный ресурс]. – Режим доступа: <https://matplotlib.org/stable/index.html> (дата обращения: 11.10.2025).
3. Аграновский А. В. Методические указания к лабораторной работе № 3 «Непрерывные, дискретные и цифровые сигналы» по дисциплине «Цифровая обработка и передача сигналов». – Санкт-Петербург: ГУАП, 2025.
4. SkyPro. MSE и MAE: ключевые метрики для оценки точности прогнозирования [Электронный ресурс]. – Режим доступа: <https://sky.pro/wiki/analytics/mse-i-mae-klyuchevye-metriki-dlya-otsenki-tochnosti-prognozirovaniya/> (дата обращения: 10.10.2025).