Федеральное государственное автономное образовательное учреждение высшего образования
«Национальный исследовательский университет Высшая школа экономики»

Московский институт электроники и математики им. А.Н. Тихонова
Департамент электронной инженерии

ОТЧЕТ

по домашней работе № 2
по дисциплине
«Основы построения инфокоммуникационных систем и сетей»

Выполнила студентка БИТ231
Байздренко В.М.

Преподаватель:
Саматов М.Р.

Москва, 2025

Оглавление

Задание на выполнение домашней работы

Вариант 4:

Провести имитационное моделирование двухканальной системы, с частотным разделением каналов, предназначенной для одновременной передачи речевого сигнала (1 канал) и сигнала передачи данных (2 канал).

До поступления во 2 канал на передающем участке цифровой сигнал данных должен быть преобразован в аналоговый сигнал. На выходе 2 канала приемного участка также должны получить цифровой сигнал.

Схема реализуемой системы передачи

Рисунок 1. Схема реализуемой системы передачи.

Краткие теоретические сведения

Частотное разделение каналов (ЧРК) - один из ключевых методов реализации многоканальных систем передачи (МСП). В основе этого подхода лежит следующий принцип:

1. Преобразование первичных сигналов: каждый исходный сигнал (голос, данные и т.д.) преобразуется в канальный сигнал через модуляцию индивидуальных несущих частот.
2. Формирование группового сигнала: канальные сигналы $U_1(t)$, $U_2(t)$, …, $U_n(t)$ объединяются в суммарный сигнал. Частоты $f_1 \ldots f_n$ выбираются так, чтобы спектры канальных сигналов не перекрывались.
3. Передача по линии связи: групповой сигнал $U_\Sigma(t)$ передается через линейный тракт, где подвергается искажениям и воздействию помех.

Генерация сигналов заключается в создании аналогового (речевого) и цифрового сигналов, как суммы гармоник, а также несущих.

Аналогово-цифровое преобразование заключается в преобразовании непрерывного аналогового сигнала в дискретный цифровой код. АЦП состоит из нескольких этапов, таких как дискретизация по времени, квантование по амплитуде и кодирование.

Дискретизация – это процесс преобразования непрерывного (аналогового) сигнала в дискретный, путем выборки его значений в определенные моменты времени. Этот процесс описывается теоремой Котельникова: «Для корректной реконструкции исходного аналогового сигнала из его дискретных отсчетов частота дискретизации fs должна быть не менее двух максимальных частот спектра исходного сигнала: $fs \geq 2 * fmax$, где fmax – максимальная частота спектра сигнала.»

Квантование – это ключевой этап преобразования аналогового или дискретного сигнала в цифровой формат. Он заключается в ограничении значений сигнала конечным набором уровней, что позволяет представить сигнал в виде фиксированных чисел, например, бит. Эти уровни называются квантовыми уровнями. Шаг квантования – величина интервала между соседними уровнями квантования.

Преобразование многоуровневого сигнала в код с низким основанием называется кодированием. В нашем случае при кодировании происходит преобразование в двоичный код, то есть каждый квантовый уровень отображается в виде двоичный последовательности (кодовой цифры).

Цифро-аналоговое преобразование используется для преобразования дискретного цифрового сигнала обратно в аналоговую

форму перед модуляцией и передачей. Для этого используется кубическая интерполяция, то есть для каждого интервала $[t_k, t_{k+1}]$ строится полином 3-й степени: $P_k(t) = a_k + b_{k(t-t_k)} + c_k(t-t_k)^2 + d_k(t-t_k)^3$. В коде для этого используется функция interp1d из библиотеки scipy.interpolate.

Амплитудная модуляция заключается в процессе изменения амплитуды несущего сигнала в зависимости от модулирующего сигнала, который как раз содержит в себе передаваемую информацию. Математическая модель АМ: $s_{AM}(t) = \left[1 + m\, s_{\text{инф}}(t)\right] * cos(\omega t)$. При гармоническом модулирующем сигнале спектр АМ-сигнала содержит несущую f_c, и боковые $f_c - f_m$ и $f_c + f_m$ частоты. При глубине модуляции m >1 происходит перемодуляция, при m <3 мощность боковых полос может быть недостаточной, поэтому в коде m = 0.5, в качестве оптимального значения. После этого происходит формирование группового сигнала (сумма сигналов).

Фильтрация в системе передачи играет роль при разделении каналов, подавлении помех и восстановлении сигналов. В системе передачи используется три типа фильтров:

- Полосовые – для формирования группового сигнала и разделения каналов;
- Низкочастотные – для восстановления сигналов после демодуляции;
- Частотно-зависимые – в модели каналов.

Линия передачи – это совокупность проводников, предназначенных для передачи электромагнитной энергии или электрических сигналов от одного устройства к другому.

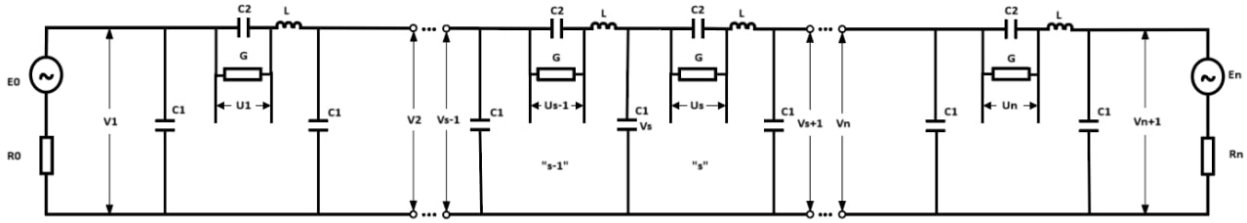

Рисунок 2.1. Общая эквивалентная схема дискретной модели линии передачи

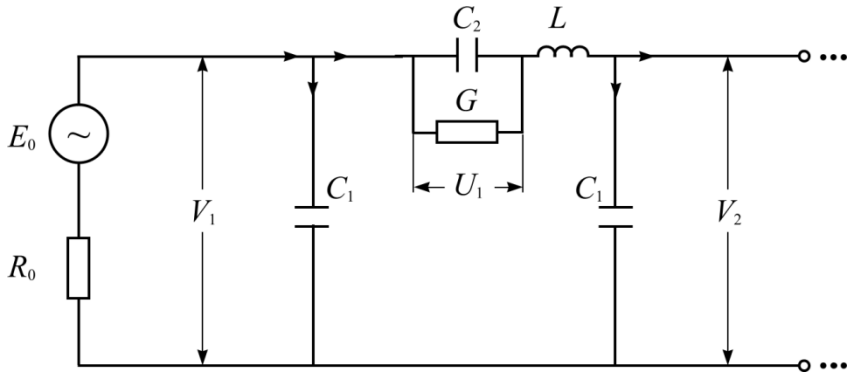

Рисунок 2.2. Эквивалентная схема входного звена дискретной модели линии передачи.

Расчет граничных условий входного звена (рисунок 1.2):

$$\frac{d^2V_1}{dt^2} = \frac{1}{LC_1}(V_2 - V_1 + U_1) + \frac{1}{R_0C_1}\frac{d}{dt}(E_0 - V_1)$$

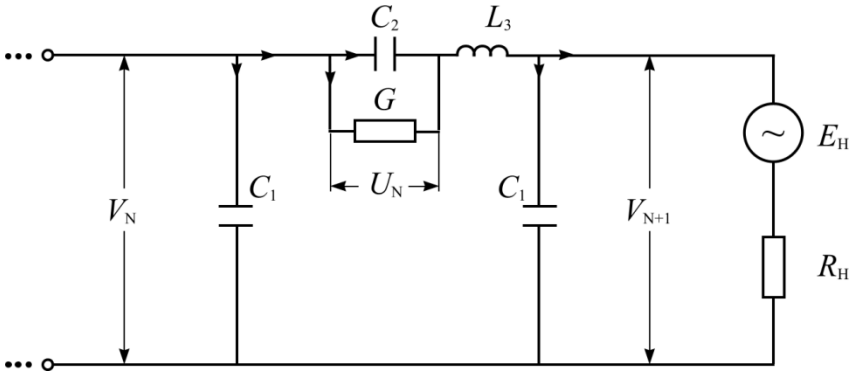

Рисунок 2.3. Эквивалентная схема выходного звена дискретной модели линии передачи.

Расчет граничных условий выходного звена (рисунок 1.3):

$$\frac{d^2V_{N+1}}{dt^2} = \frac{1}{LC_1}(V_N - V_{N+1} - U_N) + \frac{1}{R_нC_1}\frac{d}{dt}(E_н - V_{N+1})$$

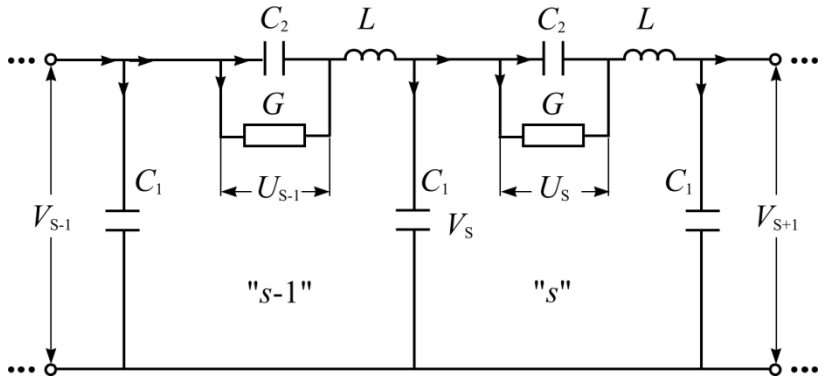

Рисунок 2.4. Эквивалентная схема двух соседних звеньев дискретной модели линии передачи.

Уравнения, связывающие напряжения в соседних ячейках:

$$\frac{d^2V_s}{dt^2} = \frac{1}{LC_1}(V_{s-1} - 2V_s + V_{s+1} + U_s - U_{s-1})$$

$$\frac{d^2U_s}{dt^2} = \frac{1}{LC_2}(V_s - V_{s+1} - U_s) - \frac{G}{C_2}\frac{dU_s}{dt}$$

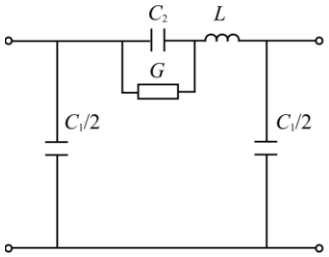

Рисунок 2.5. Эквивалентная схема одной ячейки дискретной модели линии передачи.

Формулы для расчета параметров эквивалентной схемы одной ячейки:

$$L = \sqrt{\frac{Z_0^2(f_0)\omega_0^2(2\omega_в^2 - \omega_н^2 - \omega_0^2)}{(\omega_в^2 - \omega_н^2)^2 \cdot (\omega_0^2 - \omega_н^2)}}$$

$$C_1 = \frac{2}{L(\omega_в^2 - \omega_н^2)}$$

$$C_2 = \frac{1}{\omega_н^2 L}$$

$$Z_0(f) = \sqrt{\frac{L^2(\omega_в^2 - \omega_н^2)^2 \cdot (\omega^2 - \omega_н^2)}{\omega^2(2\omega_в^2 - \omega_н^2 - \omega^2)}}$$

Демодуляция - метод восстановления информационного сигнала путём умножения принятого сигнала на опорное колебание с последующей фильтрацией. Математическая модель: $s_{\text{демод}}(t) = \frac{2\, s_{AM}(t)\, cos(\omega t)}{m}$.

Листинг программы

```
import numpy as np
import math
import matplotlib.pyplot as plt
from scipy.interpolate import interp1d


def f2w(f):
    return 2.0 * math.pi * f

def Z1(f, C1):
    return 2.0 / (1j * f2w(f) * C1)

def Z2(f, C2):
    return 1.0 / (1j * f2w(f) * C2)

def Z3(f, L):
    return 1.0j * f2w(f) * L

def Gam(f, L, C1, C2):
    ZY = (Z2(f, C2) + Z3(f, L)) / Z1(f, C1)
    return 2.0 * np.arcsinh(np.sqrt(ZY))

def Zw(f, L, C1, C2):
    num = Z1(f, C1)**2 * (Z2(f, C2) + Z3(f, L))
    den = 2 * Z1(f, C1) + Z2(f, C2) + Z3(f, L)
    return np.sqrt(num / den)

def bandpass_filter(time, signal, fl, fh):
    n = len(signal)
    dt = time[1] - time[0]
    freqs = np.fft.fftfreq(n, dt)
    spectrum = np.fft.fft(signal)

    for i in range(n):
        if not (fl < abs(freqs[i]) < fh):
            spectrum[i] = 0j

    return np.fft.ifft(spectrum).real


fs = 10       # Базовая частота сигнала
T = 0.2       # Длительность сигнала
n = 10000     # Число точек дискретизации
mod_depth = 0.5  # Глубина модуляции
Z0 = 10       # Характеристическое сопротивление линии
Nc = 100      # Число ячеек в линии

# Инициализация массивов
time = np.linspace(0, T, n)
freqs = np.fft.fftfreq(n, T/n)

sig1 = np.zeros(n)  # Аналоговый (речевой) сигнал
sig2 = np.zeros(n)  # Цифровой сигнал
sam1 = np.zeros(n)  # Несущая 10*fs
sam2 = np.zeros(n)  # Несущая 50*fs

```

```
# Генерация сигналов
for i, t in enumerate(time):
    sig1[i] = 5.0 * math.sin(1*f2w(fs)*t) + 7.0 * math.sin(4*f2w(fs)*t) + 9.0 * math.sin(7*f2w(fs)*t)

    sig2[i] = 6.5 * math.sin(2*f2w(fs)*t) + 8.5 * math.sin(3*f2w(fs)*t) + 10.0 * math.sin(5*f2w(fs)*t)

    sam1[i] = math.sin(f2w(10*fs)*t)
    sam2[i] = math.sin(f2w(50*fs)*t)


# Дискретизация по времени
d_t = 1 / (2 * 10 * fs)
d_n = int(T / d_t)
d_sig = np.zeros(d_n)
d_time = np.zeros(d_n)

for i in range(d_n):
    idx = i * int(n/d_n)
    d_sig[i] = sig2[min(idx, n-1)]
    d_time[i] = i * d_t

# Квантование по уровню (2 уровня)
min_level = min(sig2)
max_level = max(sig2)
d_U = (max_level - min_level) / 1  # Для 2 уровней

q_sig = np.zeros(d_n)
for i in range(d_n):
    value = d_sig[i]
    if value >= (min_level + max_level)/2:
        q_sig[i] = max_level
    else:
        q_sig[i] = min_level

# Формирование цифрового сигнала
lengthOfBits = int(n / d_n)
sig2d = np.zeros(n)

for i in range(d_n):
    start_idx = i * lengthOfBits
    end_idx = (i+1) * lengthOfBits
    if end_idx > n: end_idx = n
    sig2d[start_idx:end_idx] = q_sig[i]

# Визуализация исходных сигналов
plt.figure(figsize=(16, 12))

plt.subplot(2, 2, 1)
plt.title('Речевой сигнал')
plt.plot(time, sig1)

plt.subplot(2, 2, 2)
plt.title('Спектр речевого сигнала')
plt.xlim(0, 8*fs)
spectrum = np.fft.fft(sig1)
plt.plot(freqs, np.abs(spectrum)/n*2)
```

```
113
114 plt.subplot(2, 2, 3)
115 plt.title('Цифровой сигнал')
116 plt.plot(time, sig2, '--', color='gray')
117 plt.scatter(d_time, q_sig, color='red')
118 plt.plot(time, sig2d)
119
120 plt.subplot(2, 2, 4)
121 plt.title('Спектр цифрового сигнала')
122 plt.xlim(0, 100*fs)
123 spectrum = np.fft.fft(sig2d)
124 plt.plot(freqs, np.abs(spectrum)/n*2)
125
126 plt.tight_layout()
127 plt.show()
128
129
130 interp_func = interp1d(d_time, q_sig, kind='cubic',
131 fill_value="extrapolate")
132 sig2a = interp_func(time)
133
134 plt.figure(figsize=(16, 6))
135
136 plt.subplot(1, 2, 1)
137 plt.title("ЦАП - восстановленный сигнал")
138 plt.plot(time, sig2a)
139
140 plt.subplot(1, 2, 2)
141 plt.title('Спектр после ЦАП')
142 plt.xlim(fs, 12*fs)
143 spectrum = np.fft.fft(sig2a)
144 plt.plot(freqs, np.abs(spectrum)/n*2)
145
146 plt.tight_layout()
147 plt.show()
148
149
150 # Амплитудная модуляция
151 for i in range(n):
152     sam1[i] = sam1[i] * (1 + mod_depth * sig1[i] / 2.0)
153     sam2[i] = sam2[i] * (1 + mod_depth * sig2a[i] / 2.0)
154
155 # Формирование группового сигнала
156 grp = bandpass_filter(time, sam1, 3*fs-5, 17*fs+5) +
    bandpass_filter(time, sam2, (50-13)*fs-5, (50+13)*fs+5)
157
158 # Визуализация модулированных сигналов
159 plt.figure(figsize=(16, 6))
160
161 plt.subplot(1, 2, 1)
162 plt.title("Амплитудная модуляция")
163 plt.plot(time, sam1, label='10fs')
164 plt.plot(time, sam2, label='50fs')
165 plt.legend()
166
167 plt.subplot(1, 2, 2)
168 plt.title('Спектр АМ сигналов')
169 plt.xlim(fs, 70*fs)
```

```
spectrum1 = np.fft.fft(sam1)
spectrum2 = np.fft.fft(sam2)
plt.plot(freqs, np.abs(spectrum1)/n*2)
plt.plot(freqs, np.abs(spectrum2)/n*2)

plt.tight_layout()
plt.show()

# Визуализация группового сигнала
plt.figure(figsize=(16, 6))

plt.subplot(1, 2, 1)
plt.title("Групповой сигнал")
plt.plot(time, grp)

plt.subplot(1, 2, 2)
plt.title('Спектр группового сигнала')
plt.xlim(fs, 70*fs)
spectrum = np.fft.fft(grp)
plt.plot(freqs, np.abs(spectrum)/n*2)

plt.tight_layout()
plt.show()


# Расчет параметров линии
fl = 0.5 * fs
fh = 70.5 * fs
fc = (fl + fh) / 2
wl = f2w(fl)
wc = f2w(fc)
wh = f2w(fh)

L = math.sqrt(Z0**2 * wc**2 * (2*wh**2 - wl**2 - wc**2) /
              ((wh**2 - wl**2)**2 * (wc**2 - wl**2)))
C1 = 2.0 / (L * (wh**2 - wl**2))
C2 = 1.0 / (wl**2 * L)
G = 0

print(f'Параметры ячейки ЛП:\nC1 = {C1:.6f}\nC2 = {C2:.6f}\nL = {L:.6f}')

# Расчет характеристик линии
freq_lp = np.linspace(0.8*fl, fh*1.2, int(T*fh))
Gama_val = Gam(freq_lp, L, C1, C2)
Zw_val = Zw(freq_lp, L, C1, C2)
dF = (Gam(freq_lp+0.1, L, C1, C2).imag - Gam(freq_lp-0.1, L, C1,
C2).imag)/0.2

# Визуализация характеристик линии
plt.figure(figsize=(16, 10))

plt.subplot(2, 2, 1)
plt.title("Постоянная распространения")
plt.plot(freq_lp, Gama_val.real, 'b', label=r'$\alpha(f)$')
plt.plot(freq_lp, Gama_val.imag, 'r', label=r'$\phi(f)$')
plt.legend()

plt.subplot(2, 2, 2)
```

```
plt.title("Характеристическое сопротивление")
plt.plot(freq_lp, np.abs(Zw_val), label='|Z|')
plt.plot(freq_lp, Zw_val.real, '--', label='Re')
plt.plot(freq_lp, Zw_val.imag, '--', label='Im')
plt.legend()

plt.subplot(2, 2, 3)
plt.title("Производная фазовой постоянной")
plt.plot(freq_lp, dF)

plt.tight_layout()
plt.show()

# Моделирование линии передачи
dt = T / n
aU = np.zeros(Nc)       # Напряжения на C2
dU = np.zeros(Nc)       # Производные напряжений на C2
aV = np.zeros(Nc+1)     # Напряжения на C1
dV = np.zeros(Nc+1)     # Производные напряжений на C1

lp_in = np.zeros(n)     # Входные напряжения
lp_out = np.zeros(n)    # Выходные напряжения

dpp = 20  # Число итераций на шаг

def d_signal(t):
    idx = min(n-1, int(t/dt))
    return (grp[idx] - grp[idx-1])/dt if idx > 0 else 0

for it in range(n):
    t = dt * it
    for _ in range(dpp):
        # Граничные условия
        dV[0] += (1.0/(L*C1)*(aV[1]-aV[0]+aU[0]) +
                  1.0/(Z0*C1)*(d_signal(t) - dV[0])) * dt/dpp

        # Уравнения для внутренних ячеек
        for ic in range(Nc):
            dU[ic] += (1.0/(L*C2)*(aV[ic]-aV[ic+1]-aU[ic]) -
                       G/C2*dU[ic]) * dt/dpp
            if ic > 0:
                dV[ic] += (0.5/(L*C1)*(aV[ic-1]-2*aV[ic]+aV[ic+1]+aU[ic]-
aU[ic-1])) * dt/dpp

        dV[Nc] += (1.0/(L*C1)*(aV[Nc-1]-aV[Nc]-aU[Nc-1]) +
                   1.0/(Z0*C1)*(-dV[Nc])) * dt/dpp

        # Обновление напряжений
        for ic in range(Nc):
            aV[ic] += dV[ic] * dt/dpp
            aU[ic] += dU[ic] * dt/dpp
        aV[Nc] += dV[Nc] * dt/dpp

    lp_in[it] = aV[0]
    lp_out[it] = aV[Nc]

# Визуализация результатов прохождения через линию
plt.figure(figsize=(16, 6))
```

```
282
283 plt.subplot(1, 2, 1)
284 plt.title("Входной и выходной сигналы на ЛП")
285 plt.plot(time, lp_in, label='Вход')
286 plt.plot(time, lp_out, label='Выход')
287 plt.legend()
288
289 plt.subplot(1, 2, 2)
290 plt.title('Спектр после ЛП')
291 plt.xlim(0, 70*fs)
292 spectrum = np.fft.fft(lp_out)
293 plt.plot(freqs, np.abs(spectrum)/n*2)
294
295 plt.tight_layout()
296 plt.show()
297
298
299 # Выделение канальных сигналов
300 rch1 = bandpass_filter(time, lp_out, 3*fs-5, 17*fs+5)
301 rch2 = bandpass_filter(time, lp_out, (50-13)*fs-5, (50+13)*fs+5)
302
303 plt.figure(figsize=(16, 6))
304
305 plt.subplot(1, 2, 1)
306 plt.title("Выделение канальных сигналов")
307 plt.plot(time, rch1, label='Канал 1')
308 plt.plot(time, rch2, label='Канал 2')
309 plt.legend()
310
311 plt.subplot(1, 2, 2)
312 plt.title('Спектры канальных сигналов')
313 plt.xlim(0, 70*fs)
314 spectrum1 = np.fft.fft(rch1)
315 spectrum2 = np.fft.fft(rch2)
316 plt.plot(freqs, np.abs(spectrum1)/n*2)
317 plt.plot(freqs, np.abs(spectrum2)/n*2)
318
319 plt.tight_layout()
320 plt.show()
321
322 # Демодуляция
323 mch1 = np.zeros(n)
324 mch2 = np.zeros(n)
325
326 for i in range(n):
327     mch1[i] = rch1[i] * math.cos(f2w(10*fs)*time[i])
328     mch2[i] = rch2[i] * math.cos(f2w(50*fs)*time[i])
329
330 norm_factor1 = 2.0 / mod_depth
331 norm_factor2 = 2.0 / mod_depth
332 mch1 *= norm_factor1
333 mch2 *= norm_factor2
334
335 plt.figure(figsize=(16, 6))
336
337 plt.subplot(1, 2, 1)
338 plt.title('Амплитудная демодуляция')
339 plt.plot(time, mch1, label='Канал 1')
```

```
plt.plot(time, mch2, label='Канал 2')
plt.legend()

plt.subplot(1, 2, 2)
plt.title('Спектры демодуляции')
plt.xlim(0, 15*fs)
spectrum1 = np.fft.fft(mch1)
spectrum2 = np.fft.fft(mch2)
plt.plot(freqs, np.abs(spectrum1)/n*2)
plt.plot(freqs, np.abs(spectrum2)/n*2)

plt.tight_layout()
plt.show()

# Восстановление сигналов
rsig1 = bandpass_filter(time, mch1, 1, 7*fs+1)
rsig2 = bandpass_filter(time, mch2, 1, 15*fs+1)

plt.figure(figsize=(16, 6))

plt.subplot(1, 2, 1)
plt.title('Восстановленные сигналы')
plt.plot(time, rsig1, label='Аналоговый')
plt.plot(time, rsig2, label='Цифровой')
plt.legend()

plt.subplot(1, 2, 2)
plt.title('Спектры после ФНЧ')
plt.xlim(0, 13*fs)
spectrum1 = np.fft.fft(rsig1)
spectrum2 = np.fft.fft(rsig2)
plt.plot(freqs, np.abs(spectrum1)/n*2)
plt.plot(freqs, np.abs(spectrum2)/n*2)

plt.tight_layout()
plt.show()

# АЦП для восстановленного цифрового сигнала
d_sig_restored = np.zeros(d_n)
for i in range(d_n):
    idx = i * int(n/d_n)
    d_sig_restored[i] = rsig2[min(idx, n-1)]

q_sig_restored = np.zeros(d_n)
for i in range(d_n):
    value = d_sig_restored[i]
    if value >= (min_level + max_level)/2:
        q_sig_restored[i] = max_level
    else:
        q_sig_restored[i] = min_level

rsig2d = np.zeros(n)
for i in range(d_n):
    start_idx = i * lengthOfBits
    end_idx = (i+1) * lengthOfBits
    if end_idx > n: end_idx = n
    rsig2d[start_idx:end_idx] = q_sig_restored[i]

```

```
398 # Финальная визуализация
399 plt.figure(figsize=(16, 6))
400
401 plt.subplot(1, 2, 1)
402 plt.title('Восстановленный цифровой сигнал')
403 plt.plot(time, rsig2, '--', color='gray')
404 plt.scatter(d_time, q_sig_restored, color='red')
405 plt.plot(time, rsig2d)
406
407 plt.subplot(1, 2, 2)
408 plt.title('Спектр восстановленного сигнала')
409 plt.xlim(0, 100*fs)
410 spectrum = np.fft.fft(rsig2d)
411 plt.plot(freqs, np.abs(spectrum)/n*2)
412
413 plt.tight_layout()
414 plt.show()
415
```

Результаты моделирования:

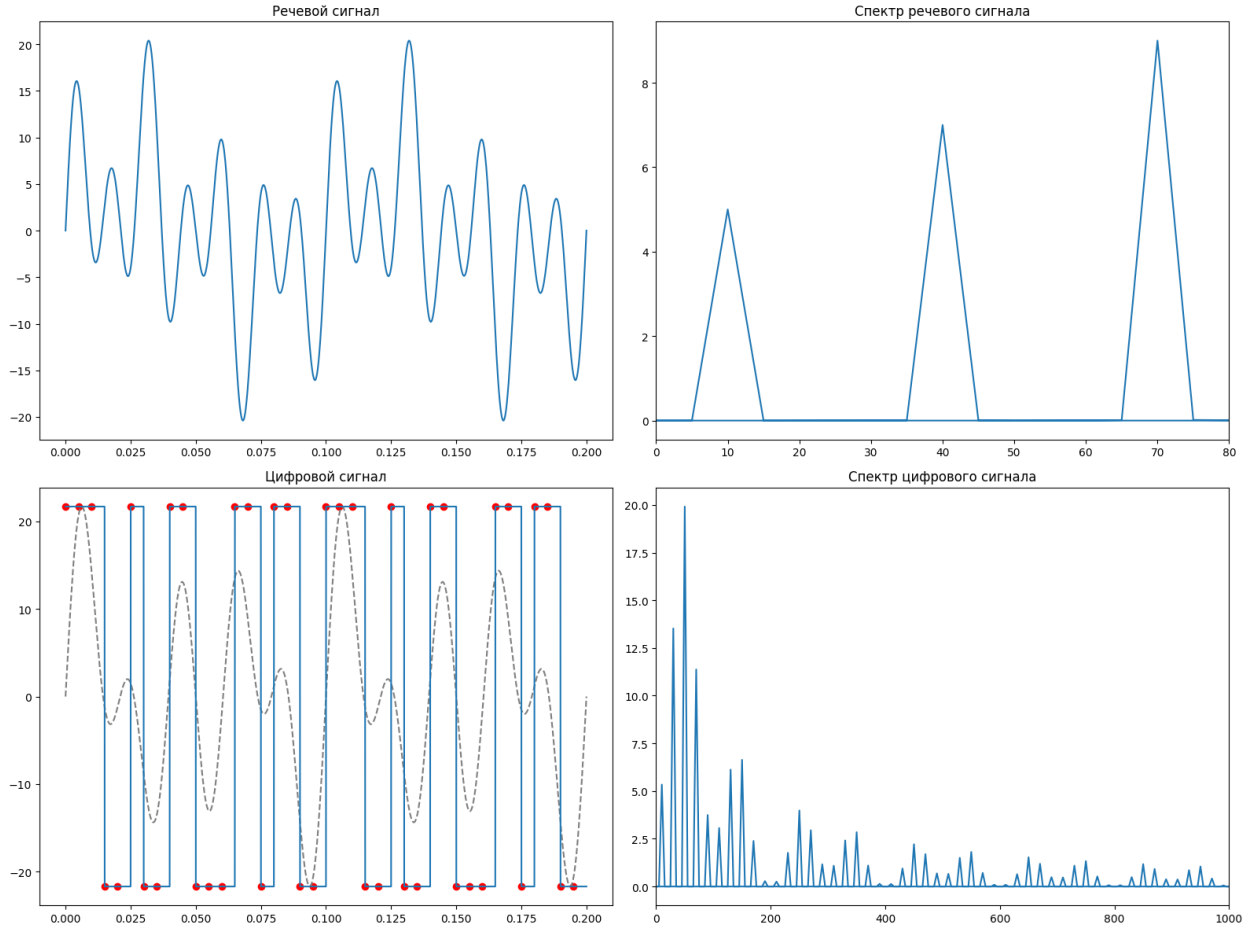

Рисунок 3.1. Исходные сигналы.

Левый верхний график содержит исходный аналоговый речевой сигнал (сумма гармоник 1fs, 4fs, 7fs). Правый верхний график содержит спектр речевого сигнала с пиками на 1 Гц, 4 Гц, 7 Гц. Левый нижний график содержит исходный цифровой сигнал (серый пунктир), его дискретные отсчеты (красные точки) и восстановленный после ЦАП сигнал (синяя линия). Правый нижний график содержит спектр цифрового сигнала с гармониками на 2 Гц, 3 Гц, 5 Гц.

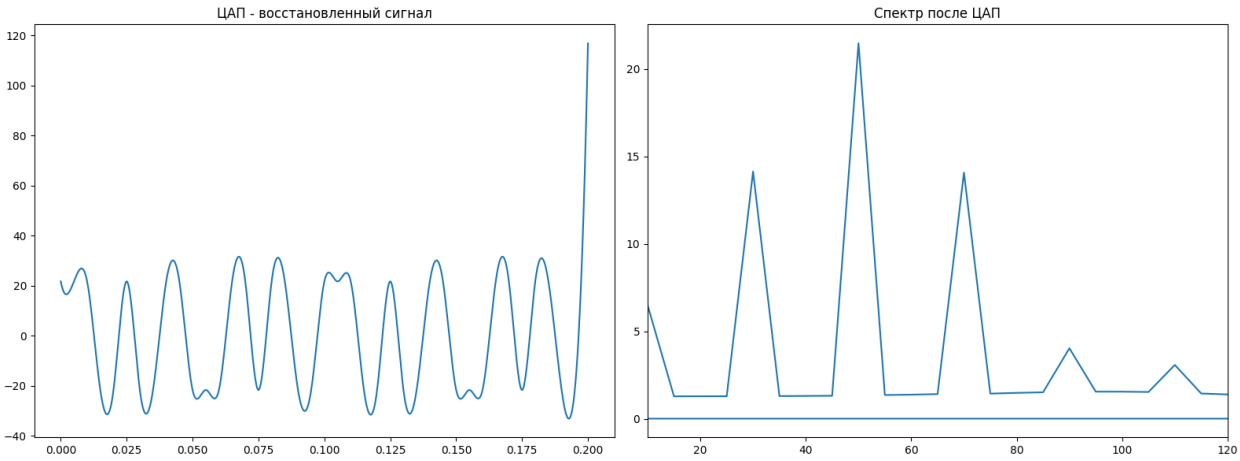

Рисунок 3.2. Результат ЦАП.

Левый график содержит восстановленный аналоговый сигнал после цифро-аналогового преобразования (кубическая интерполяция). Правый график содержит спектр сигнала после ЦАП. Видны основные гармоники (10-70 Гц) и высокочастотные компоненты, возникшие при интерполяции.

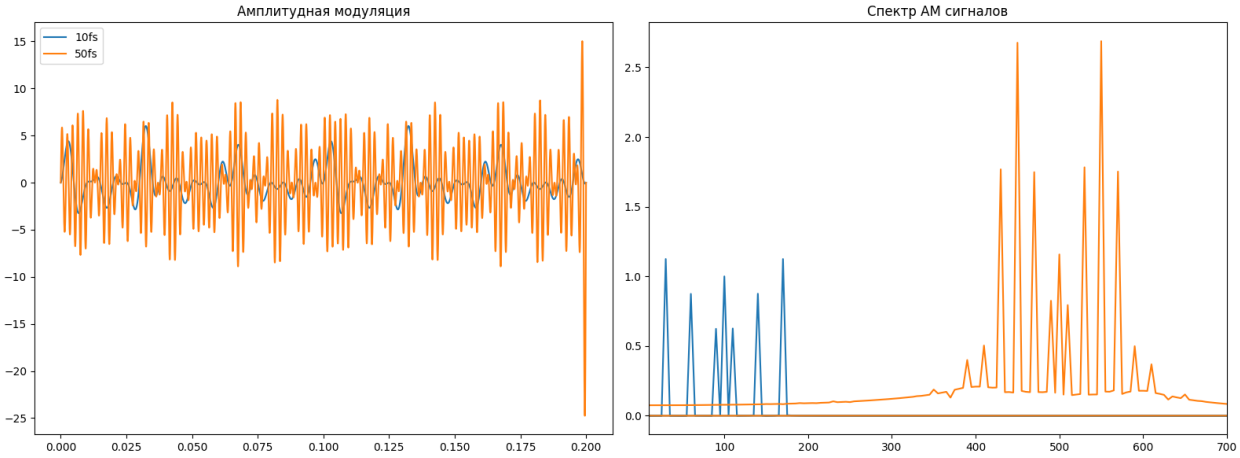

Рисунок 3.3. Модулированные сигналы.

Слева: АМ-сигналы: канал 1 (несущая 100 Гц, синий), канал 2 (несущая 500 Гц, оранжевый). Справа: спектры АМ-сигналов. Пики на несущих частотах (100 Гц и 500 Гц) с боковыми полосами.

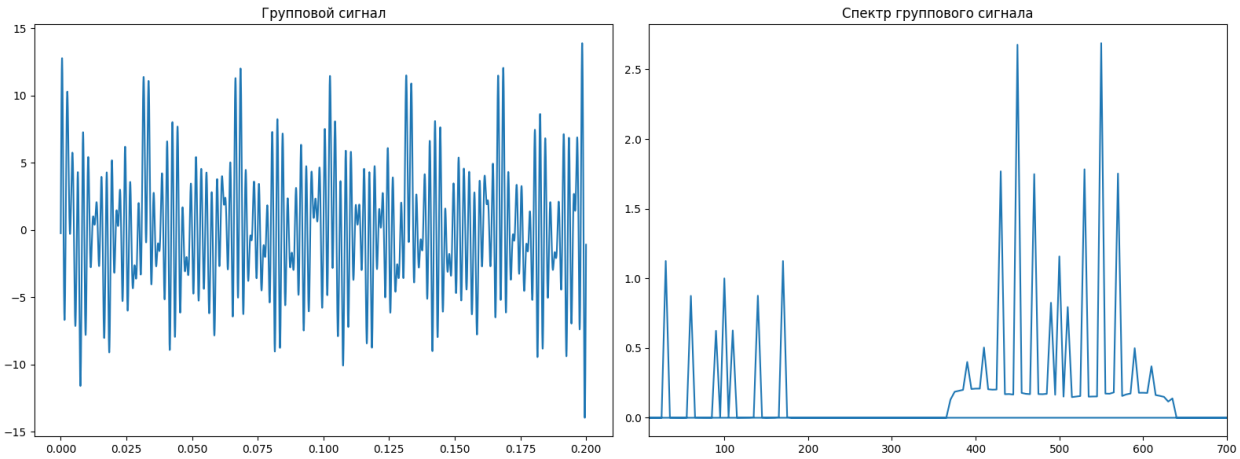

Рисунок 3.4. Групповой сигнал.

Слева: суммарный сигнал после объединения двух каналов. Справа: спектр группового сигнала. Четко видны два неперекрывающихся диапазона канала 1 и канала 2.

Параметры ячейки ЛП:
C1 = 0.000034
C2 = 0.339573
L = 0.002984

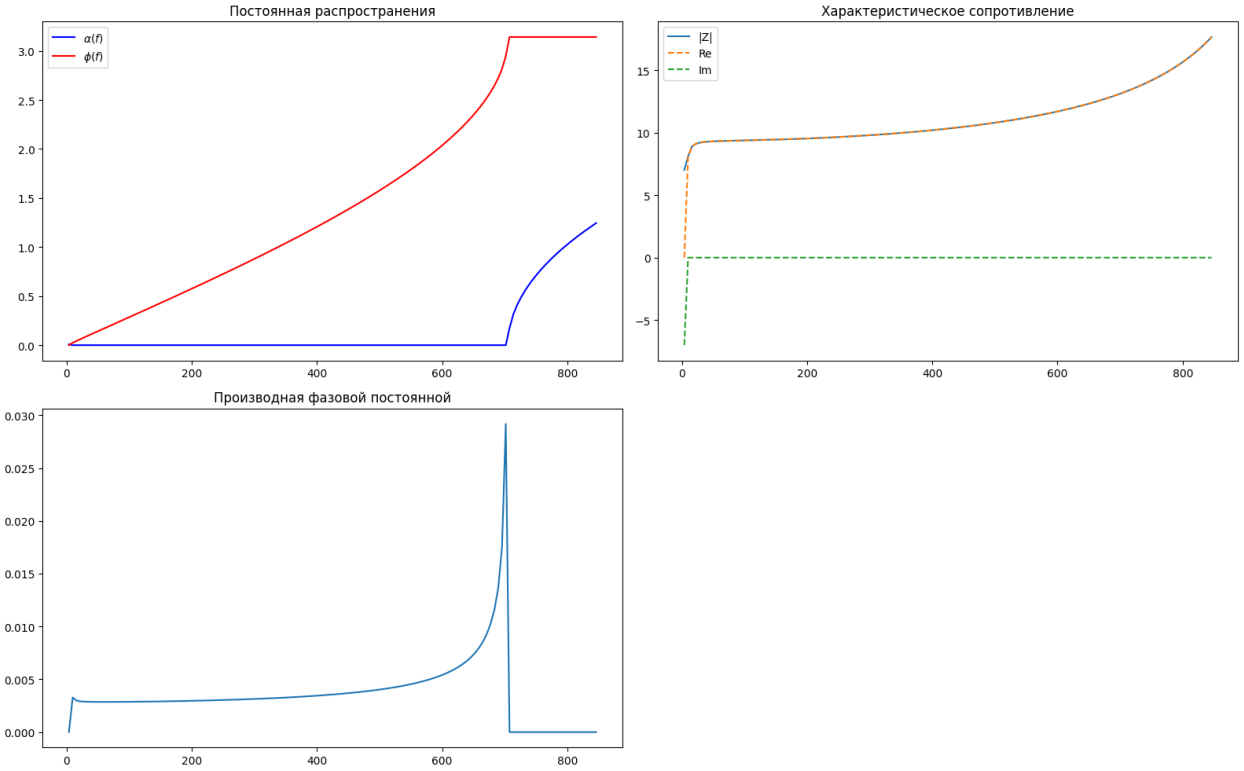

Рисунок 3.5. Характеристики линии передачи.

Верхний левый график: постоянная распространения (действительная часть α - затухание, мнимая φ - фаза). Верхний правый: характеристическое

сопротивление (|Z|, Re, Im). Нижний левый график: производная фазовой постоянной (dφ/df), определяющая задержку сигнала.

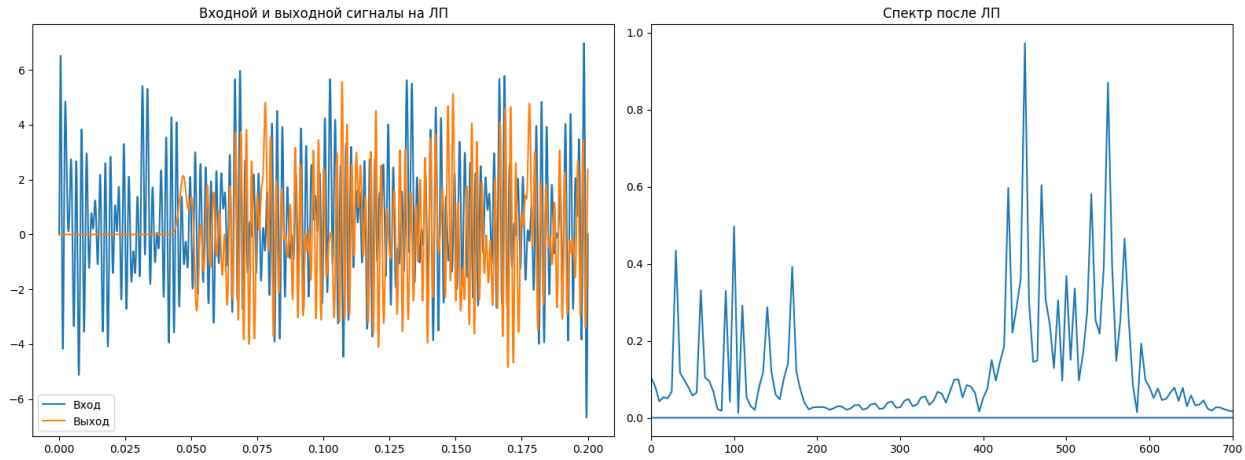

Рисунок 3.6. Сигналы на входе/выходе ЛП.

Слева: входной (синий) и выходной (оранжевый) сигналы линии передачи. Видна задержка и незначительные искажения. Справа: спектр выходного сигнала ЛП. Сохраняются основные компоненты группового сигнала.

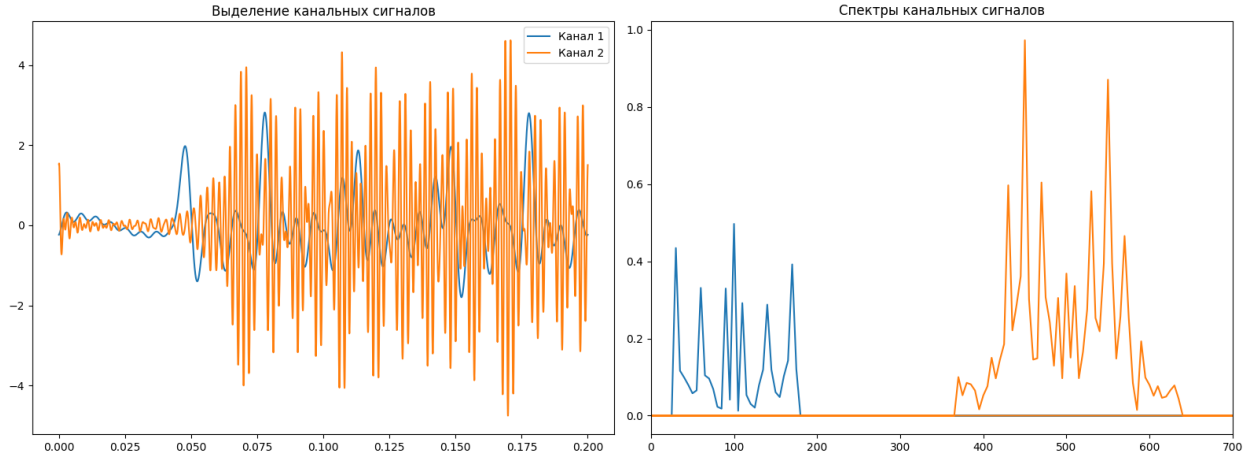

Рисунок 3.7. Разделение каналов.

Слева: выделенные сигналы после полосовой фильтрации. Справа: спектры канальных сигналов. Отсутствие перекрестных помех подтверждает корректность ЧРК.

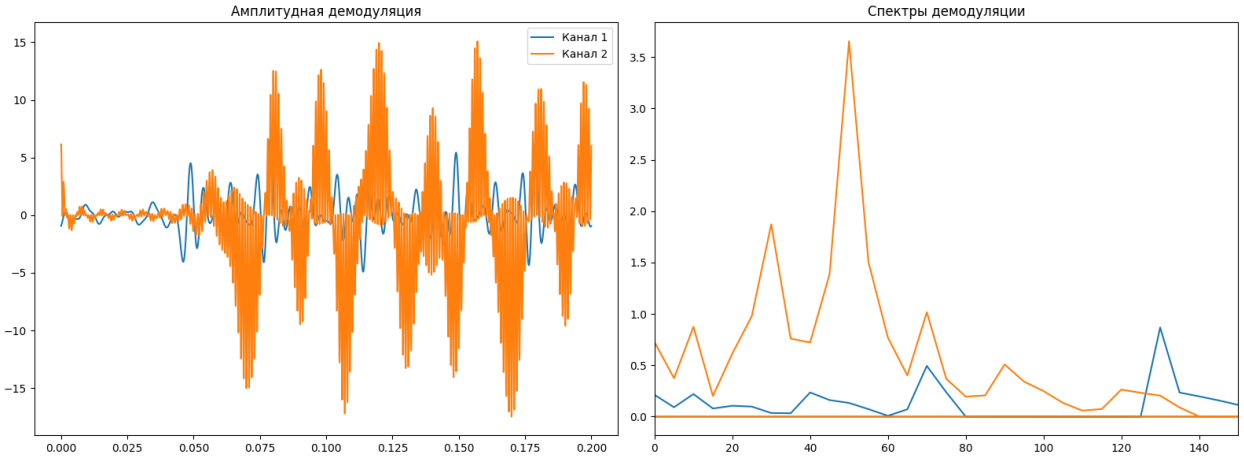

Рисунок 3.8. Демодуляция.

Слева: демодулированные сигналы после умножения на опорные несущие. Справа: спектры демодулированных сигналов. Низкочастотные компоненты (0-70 Гц) соответствуют исходным сигналам, высокочастотные (90-150 Гц) - артефакты удвоенных несущих.

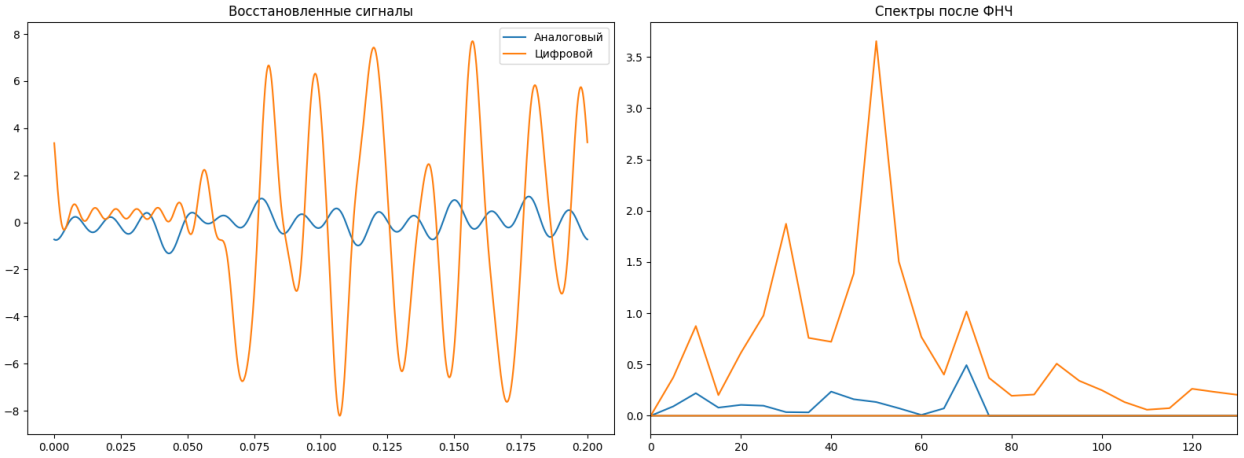

Рисунок 3.9. Восстановленные сигналы после ФНЧ.

Слева расположен график с фильтрацией ФНЧ: восстановленный речевой сигнал (синий) и цифровой сигнал (оранжевый). Справа: спектры после ФНЧ. Устранены высокочастотные компоненты, сохранены исходные гармоники (10–70 Гц).

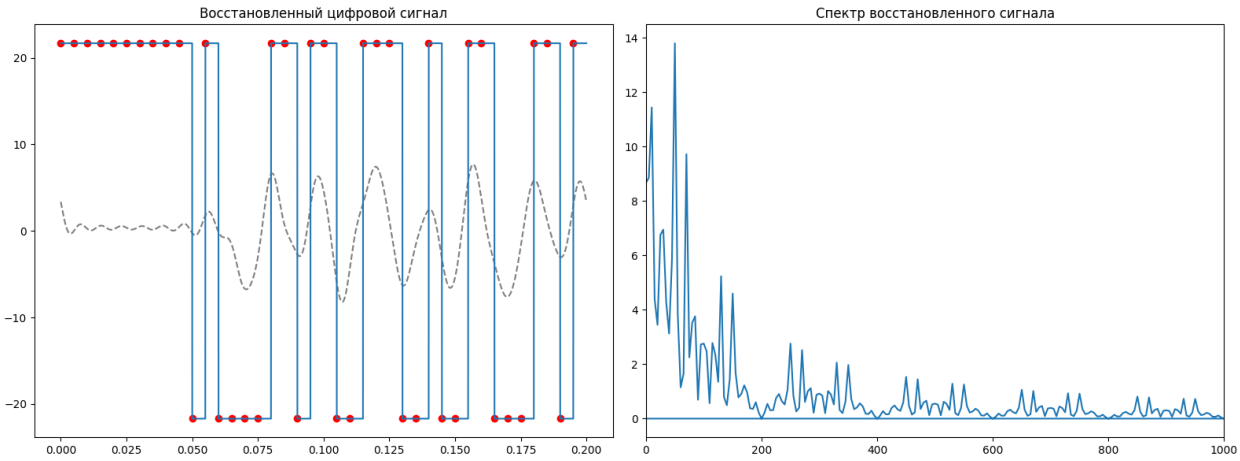

Рисунок 3.10. Цифровой сигнал на выходе.

Слева показан восстановленный цифровой сигнал: демодулированный (серый пунктир), дискретные отсчеты после квантования (красные точки), итоговый цифровой сигнал (синяя линия). Справа: спектр цифрового сигнала. Пики на 20 Гц, 30 Гц, 50 Гц соответствуют исходному сигналу.

Выводы:

Моделирование подтвердило работоспособность двухканальной системы с частотным разделением каналов (ЧРК). Графики подтвердили корректность моделирования одновременной передачи аналогового речевого сигнала (канал 1: 1-7 Гц) и цифрового сигнала данных (канал 2: 2-5 Гц) с последующим восстановлением исходных сигналов на приемной стороне.

Амплитудная модуляция с глубиной m=0.5 обеспечила оптимальный баланс между мощностью несущей и боковых полос. Демодуляция с ФНЧ корректно восстановила полученные сигналы.

Квантование при АЦП на 2 уровня успешно оцифровало сигнал, а кубическая интерполяция при ЦАП минимизировала ошибки при восстановлении сигнала.

Также были рассчитаны характеристики линии передачи.

Таким образом моделирование двухканальной системы, с частотным разделением каналов, предназначенной для одновременной передачи речевого сигнала (1 канал) и сигнала передачи данных (2 канал) выполнено корректно.