| https://sun6-22.userapi.com/s/v1/if2/6Ht1d-WrFC8wSedTY0j_dTjWhgz0nAoKyCKztOc4O8PXX_UeSTrdZXBDOnRtTC6SrQrvjUT6wDnWOSs08w5wi7yP.jpg?size=933x933&quality=96&crop=78,0,933,933&ava=1 | **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 выполнению домашнего задания № 2

по дисциплине

«Основы построения инфокоммуникационных систем и сетей»

Выполнил студент БИТ221

Круглов А.Е

Преподаватель:

Саматов М.Р.

Москва, 2024

**1)** **Привести и описать модели линии передач с частотным и временным**

**разделением каналов. Описать методы формирования канальных**

**сигналов в СП с ЧРК и ВРК.**



В импульсной модуляции присутствуют разделительные временные промежутки сигналов, в течении которых должны передаваться первые и вторые импульсы сигналов.

На рис 1 показана схема системы передачи с временным разделением, она работает следующим образом: на вход подается некоторое количество начальных сигналов, затем подаем на фильтр низких частот, чтобы ограничить сигнал, так как он находиться в широком спектре. Генерируем некоторые электронные импульсы и в соответствии с их номером подаем тот или иной сигнал на соответствующий модулятор. В результате получаются сигналы, находящиеся на некотором расстоянии друг от друга. сигнал объединяется с помощью объединяющего устройства (сумматора). Для понимания о количестве сигналов передается используется ФПСС, который добавляет синхроимпульс, отделяя наборы сигналов. После передачи сигнала он обрабатывается развязывающим устройством и затем происходит демодуляция в канальных селекторах. В конце каждые сигнал отфильтровывается с помощью фильтра низких частот.

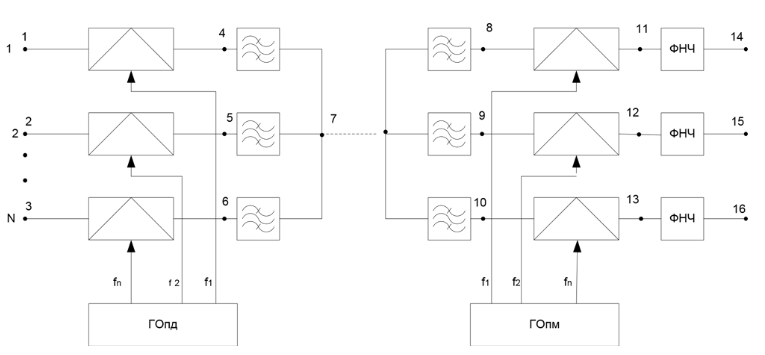


Рис 2 Структурная схема системы передачи с частотным разделением каналов

На вход системы подаются исходные сигналы. Затем они подаются на модуляторы, для каждого модулятора требуется подать несущий сигнал с заданной частотой. После модуляции каждый сигнал проходит через полосовой фильтр, который отсекает ненужные частоты, пропуская только несущую и боковые полосы. Это гарантирует, что каждый канал будет занимать полосу частот без перекрытий. Затем все модулированные сигналы объединяются с помощью сумматора. Далее сигнал подается на полосовой фильтр приемника для разделения частот, соответствующих изначальным сигналам, после эти сигналы попадают на демодулятор, а затем на фильтр низких частот.

Опорная частота: fs = 17\*5 = 85 Гц

**2.1) Реализовать Передачу и прием трех сигналов по системе передачи с частотным разделением каналов с использованием амплитудной модуляции. И параметрами сигнала:**

**а) каждый сигнал состоит из одной гармоники; глубина модуляции =**

**0,5.**

**б) каждый сигнал состоит из двух гармоник (fs, 2fs); глубина**

**модуляции = 0,8.**

import numpy as np

import matplotlib.pyplot as plt

# Опорная частота

fs = 17 \* 5 # 85 Hz

t = np.linspace(0, 1, 900) # временная шкала от 0 до 1 секунды, 1000 точек

n = len(t)

m\_single = 0.5

m\_double = 0.8

# Функция для преобразования частоты в угловую частоту

def f2w(f):

return 2 \* np.pi \* f

# Функция для демодуляции AM-сигнала

def demodulate\_am\_signal(signal, f, t):

carrier = np.cos(2 \* np.pi \* f \* t)

demodulated = signal \* carrier

return np.abs(np.fft.ifft(np.fft.fft(demodulated) \* (np.abs(np.fft.fftfreq(len(t), t[1] - t[0])) < 2 \* f)))

# Генерация первичных сигналов

sig1 = 1.0 \* np.cos(f2w(fs) \* t)

sig2 = 2.0 \* np.cos(f2w(fs) \* t)

sig3 = 1.5 \* np.cos(f2w(fs) \* t)

# Генерация несущих сигналов

fn = fs

sam1 = np.cos(f2w(fn) \* t)

sam2 = np.cos(f2w(fn) \* t)

sam3 = np.cos(f2w(fn) \* t)

# Амплитудная модуляция для случая с одной гармоникой

sam1\_single = sam1.copy()

sam2\_single = sam2.copy()

sam3\_single = sam3.copy()

for i in range(n):

sam1\_single[i] = sam1\_single[i] \* (1 + m\_single \* sig1[i] / 2.0)

sam2\_single[i] = sam2\_single[i] \* (1 + m\_single \* sig2[i] / 2.0)

sam3\_single[i] = sam3\_single[i] \* (1 + m\_single \* sig3[i] / 2.0)

# Амплитудная модуляция для случая с двумя гармониками

sig1\_double = 1.0 \* np.cos(f2w(fs) \* t) + 0.8 \* np.cos(f2w(2 \* fs) \* t)

sig2\_double = 2.0 \* np.cos(f2w(fs) \* t) + 0.8 \* np.cos(f2w(2 \* fs) \* t)

sig3\_double = 1.5 \* np.cos(f2w(fs) \* t) + 0.8 \* np.cos(f2w(2 \* fs) \* t)

sam1\_double = sam1.copy()

sam2\_double = sam2.copy()

sam3\_double = sam3.copy()

for i in range(n):

sam1\_double[i] = sam1\_double[i] \* (1 + m\_double \* sig1\_double[i] / 2.0)

sam2\_double[i] = sam2\_double[i] \* (1 + m\_double \* sig2\_double[i] / 2.0)

sam3\_double[i] = sam3\_double[i] \* (1 + m\_double \* sig3\_double[i] / 2.0)

# Суммирование сигналов для передачи по одному каналу

combined\_signal\_single = sam1\_single + sam2\_single + sam3\_single

combined\_signal\_double = sam1\_double + sam2\_double + sam3\_double

# Демодуляция сигналов с одной гармоникой

demodulated\_signals\_single = []

for f in range(2):

demodulated\_signal = demodulate\_am\_signal(combined\_signal\_single, fn, t)

demodulated\_signals\_single.append(demodulated\_signal)

# Демодуляция сигналов с двумя гармониками

demodulated\_signals\_double = []

for f in range(2):

demodulated\_signal = demodulate\_am\_signal(combined\_signal\_double, fn, t)

demodulated\_signals\_double.append(demodulated\_signal)

# Визуализация сигналов

plt.figure(figsize=(24, 16))

# Визуализация исходных сигналов с одной гармоникой

plt.suptitle('AM-сигналы с одиночной и двойной гармоникой', fontsize=16)

for i, signal in enumerate([sig1, sig2, sig3]):

plt.subplot(6, 3, i + 1)

plt.plot(t, signal)

plt.title(f'Исходный сигнал {i+1} (одиночная гармоника)')

plt.xlabel('Time [s]')

plt.ylabel('Amplitude')

# Визуализация суммарного сигнала с одной гармоникой

plt.subplot(6, 3, 4)

plt.plot(t, combined\_signal\_single)

plt.title('Комбинированный сигнал (одиночная гармоника)')

plt.xlabel('Time [s]')

plt.ylabel('Amplitude')

# Визуализация демодулированных сигналов с одной гармоникой

for i, signal in enumerate(demodulated\_signals\_single):

plt.subplot(6, 3, 5 + i)

plt.plot(t, signal)

plt.title(f'Демодулированный сигнал {i+1} (одиночная гармоника)')

plt.xlabel('Time [s]')

plt.ylabel('Amplitude')

# Визуализация исходных сигналов с двумя гармониками

for i, signal in enumerate([sig1\_double, sig2\_double, sig3\_double]):

plt.subplot(6, 3, 9 + i)

plt.plot(t, signal)

plt.title(f'Исходный сигнал {i+1} (двойная гармоника)')

plt.xlabel('Time [s]')

plt.ylabel('Amplitude')

# Визуализация суммарного сигнала с двумя гармониками

plt.subplot(6, 3, 13)

plt.plot(t, combined\_signal\_double)

plt.title('Комбинированный сигнал (двойная гармоника)')

plt.xlabel('Time [s]')

plt.ylabel('Amplitude')

# Визуализация демодулированных сигналов с двумя гармониками

for i, signal in enumerate(demodulated\_signals\_double):

plt.subplot(6, 3, 14 + i)

plt.plot(t, signal)

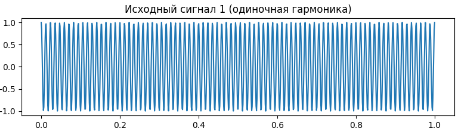
plt.title(f'Демодулированный сигнал {i+1} (двойная гармоника)')

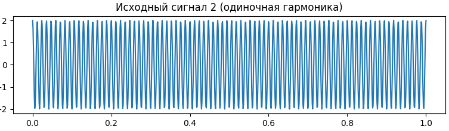
plt.xlabel('Time [s]')

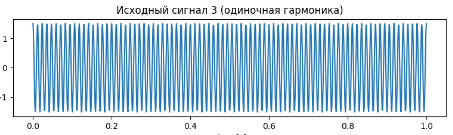
plt.ylabel('Amplitude')

plt.tight\_layout(rect=[0, 0.03, 1, 0.95])

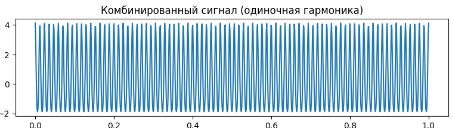
plt.show()



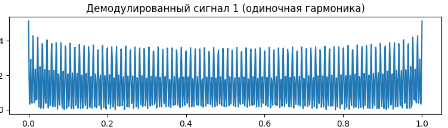
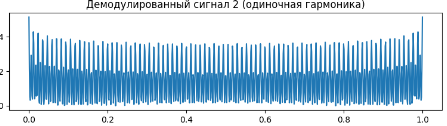




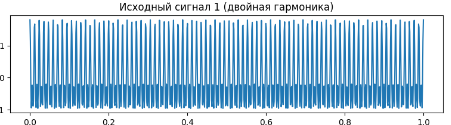
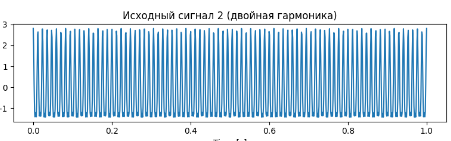
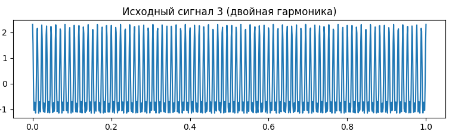
*Рис 1. Исходные сигналы одной гармоники*



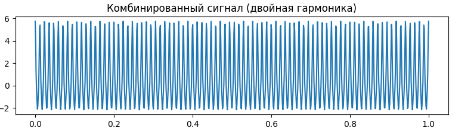
*Рис 2. Суммарный переданный сигнал одной гармоники*

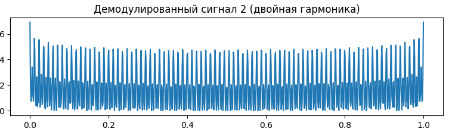
*Рис 3. Демодулированный сигнал одной гармоники*

*Рис 4. Исходные сигналы двойной гармоники*



*Рис 5. Суммарный переданный сигнал двойной гармоники*

*Рис 6. Демодулированный сигнал двойной гармоники*

**2.2) Реализовать передачу и прием трех сигналов по системе передачи с временным разделением каналов с использованием амплитудно-импульсной модуляции. И параметрами сигнала:**

**а) каждый сигнал состоит из одной гармоники; глубина модуляции =**

**0,5; скважность ПППИ = 10.**

**б) каждый сигнал состоит из двух гармоник (fs, 2fs); глубина**

**модуляции = 0,8; скважность ПППИ = 10.**

**import numpy as np**

**import matplotlib.pyplot as plt**

**def f2w(f):**

**return 2 \* np.pi \* f**

**# Параметры сигналов**

**fs = 85**

**Time = 0.2**

**n = 10000**

**As = 1.5**

**Nimp = int(6 \* fs \* Time)**

**Dimp = 1 / 10**

**Aimp = 1**

**Tshift = n / Nimp / 4**

**Asyn = 0.5**

**# Временная шкала**

**time = np.linspace(0, Time, n)**

**# Функция для генерации сигналов и модуляции**

**def generate\_signals(single\_harmonic=True, modulation\_index=0.5):**

**if single\_harmonic:**

**sig1 = 1.0 \* np.cos(f2w(fs) \* time)**

**sig2 = 2.0 \* np.cos(f2w(fs) \* time)**

**sig3 = 1.5 \* np.cos(f2w(fs) \* time)**

**else:**

**sig1 = 1.0 \* (np.cos(f2w(fs) \* time) + np.cos(f2w(2 \* fs) \* time))**

**sig2 = 2.0 \* (np.cos(f2w(fs) \* time) + np.cos(f2w(2 \* fs) \* time))**

**sig3 = 1.5 \* (np.cos(f2w(fs) \* time) + np.cos(f2w(2 \* fs) \* time))**

**simp1 = np.zeros(n)**

**simp2 = np.zeros(n)**

**simp3 = np.zeros(n)**

**ssyn = np.zeros(n)**

**#Импульсный сигнал**

**for ii in range(0, Nimp):**

**for it in range(int(n/Nimp \* ii), int(n/Nimp \* ii + n/Nimp \* Dimp)):**

**simp1[it] = Aimp**

**for it in range(int(n/Nimp \* ii + Tshift), int(n/Nimp \* ii + n/Nimp \* Dimp + Tshift)):**

**simp2[it] = Aimp**

**for it in range(int(n/Nimp \* ii + 2 \* Tshift), int(n/Nimp \* ii + n/Nimp \* Dimp + 2 \* Tshift)):**

**simp3[it] = Aimp**

**for it in range(int(n/Nimp \* ii + 3 \* Tshift), int(n/Nimp \* ii + n/Nimp \* Dimp + 3 \* Tshift)):**

**ssyn[it] = Asyn**

**spam1 = simp1 \* (1 + modulation\_index \* (sig1) / 2.0)**

**spam2 = simp2 \* (1 + modulation\_index \* (sig2) / 2.0)**

**spam3 = simp3 \* (1 + modulation\_index \* (sig3) / 2.0)**

**sgrp = spam1 + spam2 + spam3 + ssyn**

**return time, sig1, sig2, sig3, spam1, spam2, spam3, ssyn, sgrp**

**# Функция для демодуляции сигналов**

**def demodulate\_signal(modulated\_signal, carrier):**

**return modulated\_signal \* carrier**

**# Генерация и визуализация сигналов**

**def plot\_signals(single\_harmonic=True, modulation\_index=0.5):**

**time, sig1, sig2, sig3, spam1, spam2, spam3, ssyn, sgrp = generate\_signals(single\_harmonic, modulation\_index)**

**# Демодуляция**

**carrier1 = np.where(spam1 != 0, 1, 0)**

**carrier2 = np.where(spam2 != 0, 1, 0)**

**carrier3 = np.where(spam3 != 0, 1, 0)**

**demod1 = demodulate\_signal(spam1, carrier1)**

**demod2 = demodulate\_signal(spam2, carrier2)**

**demod3 = demodulate\_signal(spam3, carrier3)**

**plt.figure(figsize=(20, 20))**

**plt.subplot(7, 1, 1)**

**plt.title('Модулируемый сигнал 1')**

**plt.plot(time, sig1)**

**plt.subplot(7, 1, 2)**

**plt.title('Модулируемый сигнал 2')**

**plt.plot(time, sig2)**

**plt.subplot(7, 1, 3)**

**plt.title('Модулируемый сигнал 3')**

**plt.plot(time, sig3)**

**plt.subplot(7, 1, 4)**

**plt.title('Синхронизирующий сигнал')**

**plt.plot(time, ssyn)**

**plt.subplot(7, 1, 5)**

**plt.title('Амплитудно-импульсная модуляция для 1-ого сигнала')**

**plt.plot(time, spam1)**

**plt.subplot(7, 1, 6)**

**plt.title('Амплитудно-импульсная модуляция для 2-ого сигнала')**

**plt.plot(time, spam2)**

**plt.subplot(7, 1, 7)**

**plt.title('Амплитудно-импульсная модуляция для 3-ого сигнала')**

**plt.plot(time, spam3)**

**plt.figure(figsize=(20, 10))**

**plt.title('Групповой сигнал')**

**plt.plot(time, sgrp)**

**plt.show()**

**plt.figure(figsize=(20, 20))**

**plt.subplot(3, 1, 1)**

**plt.title('Демодулированный сигнал 1')**

**plt.plot(time, demod1)**

**plt.subplot(3, 1, 2)**

**plt.title('Демодулированный сигнал 2')**

**plt.plot(time, demod2)**

**plt.subplot(3, 1, 3)**

**plt.title('Демодулированный сигнал 3')**

**plt.plot(time, demod3)**

**plt.tight\_layout()**

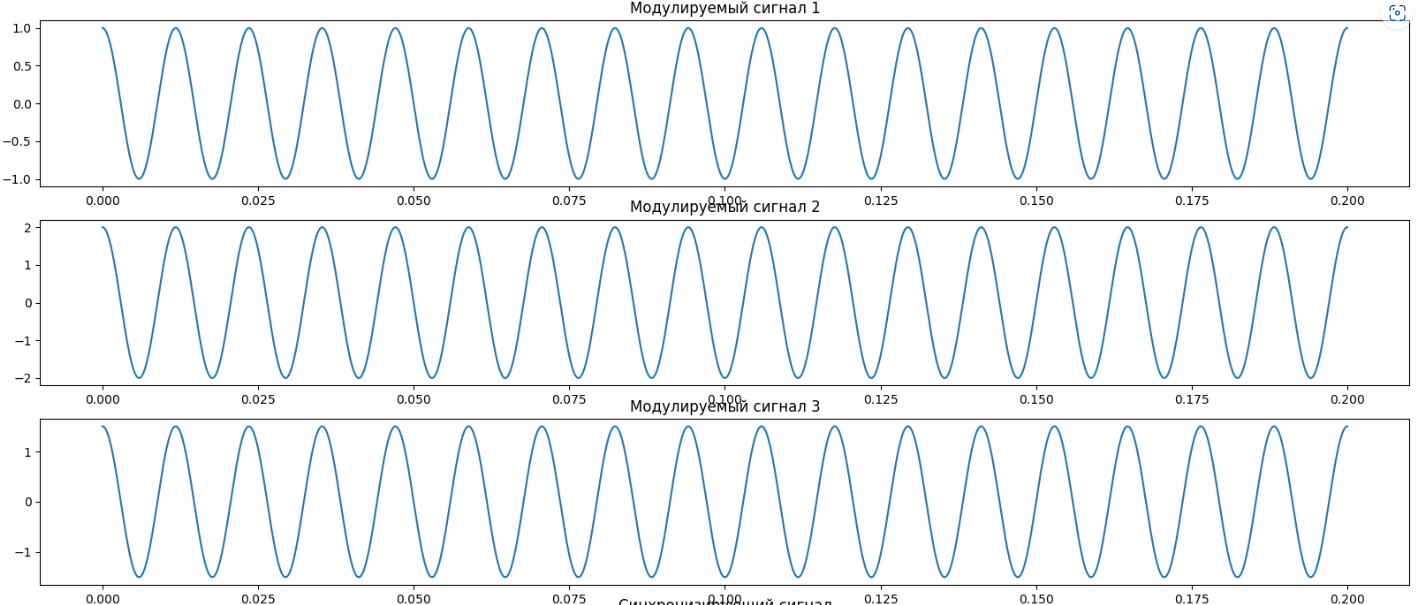
**plt.show()**

**# Вариант а: каждый сигнал состоит из одной гармоники**

**plot\_signals(single\_harmonic=True, modulation\_index=0.5)**

**# Вариант б: каждый сигнал состоит из двух гармоник (fs, 2fs)**

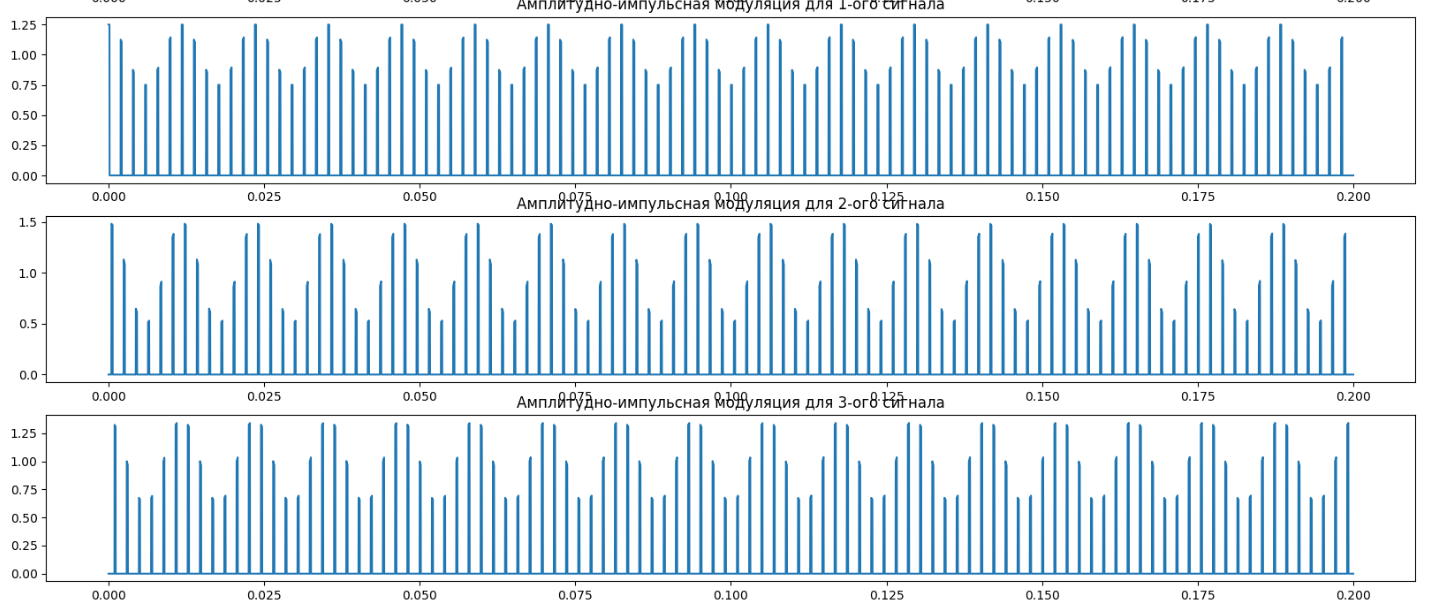
**plot\_signals(single\_harmonic=False, modulation\_index=0.8)**



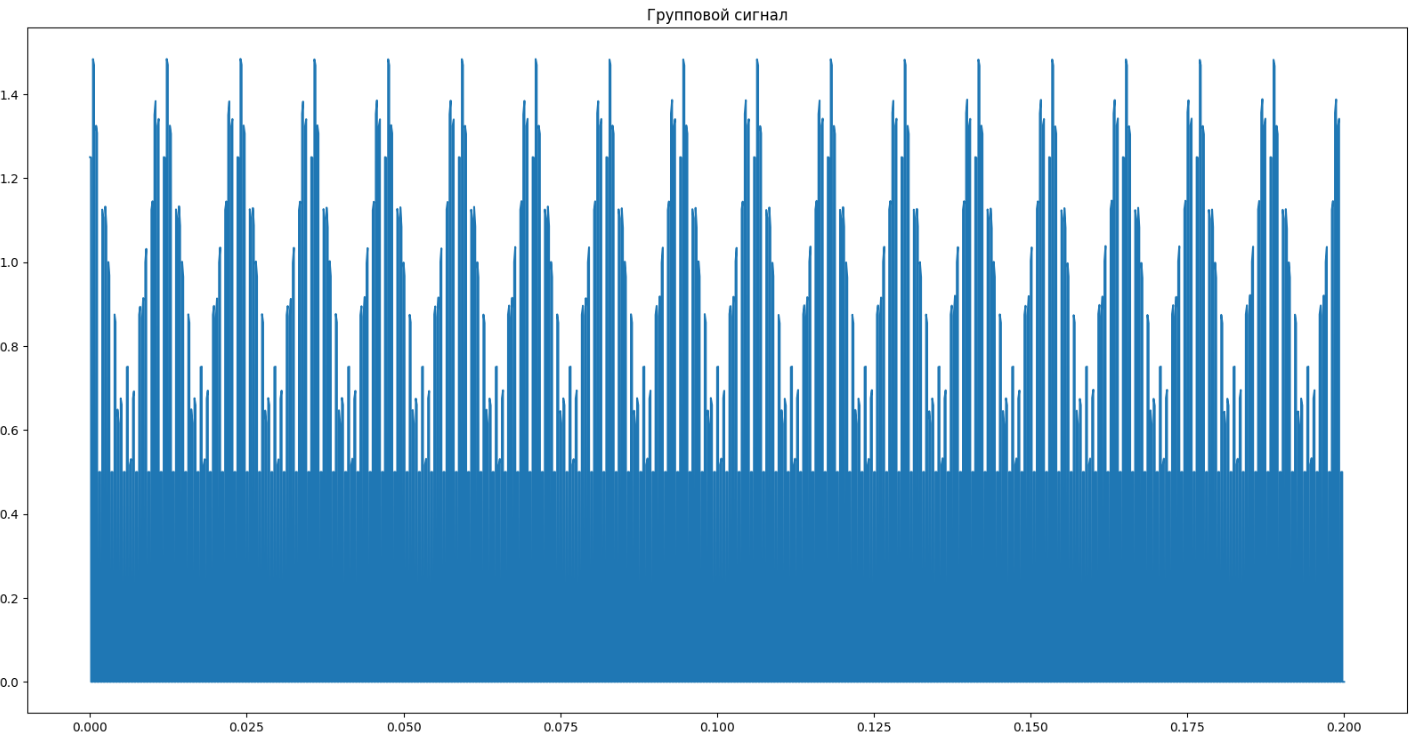
*Рис 7. Моделируемые сигналы одной гармоники*



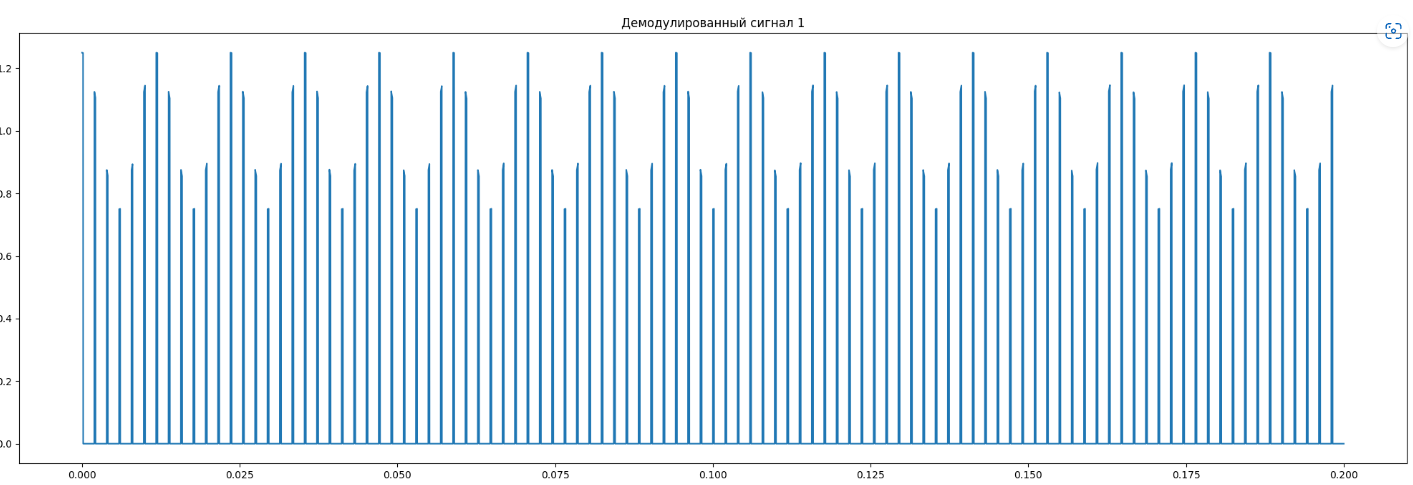
*Рис 8. Синхронизирующий сигнал одной гармоники*

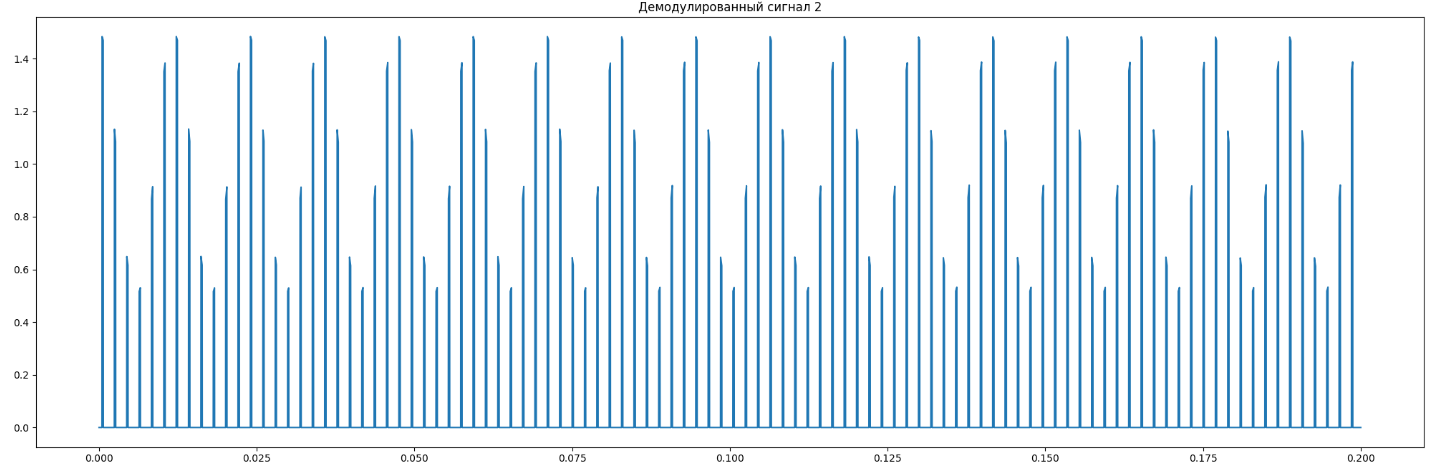
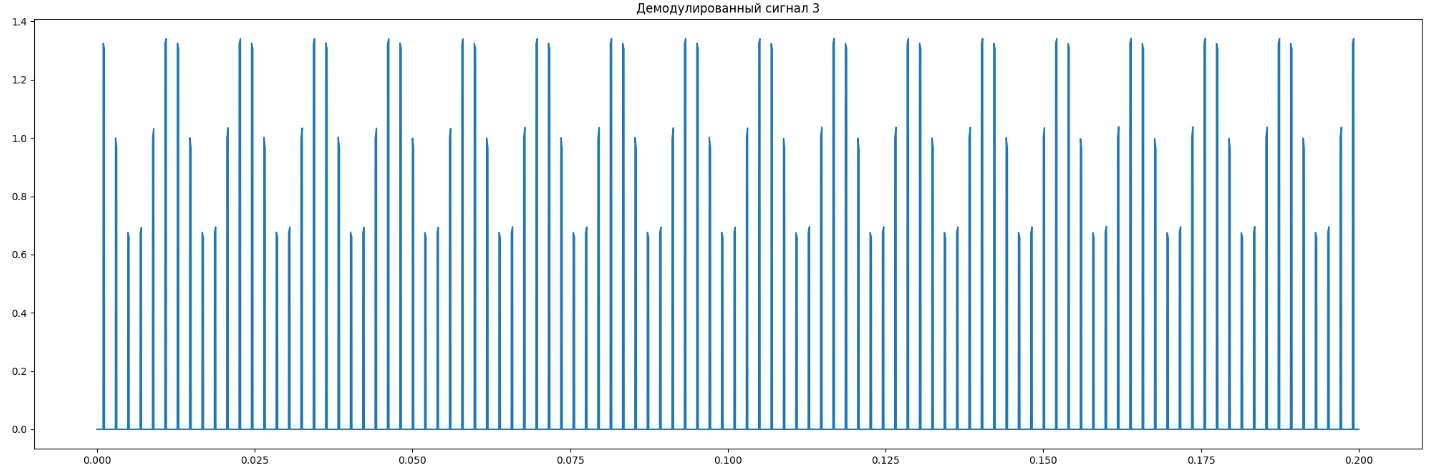


*Рис 9. Амплитудно-импульсные модуляции сигналов одной гармоники*

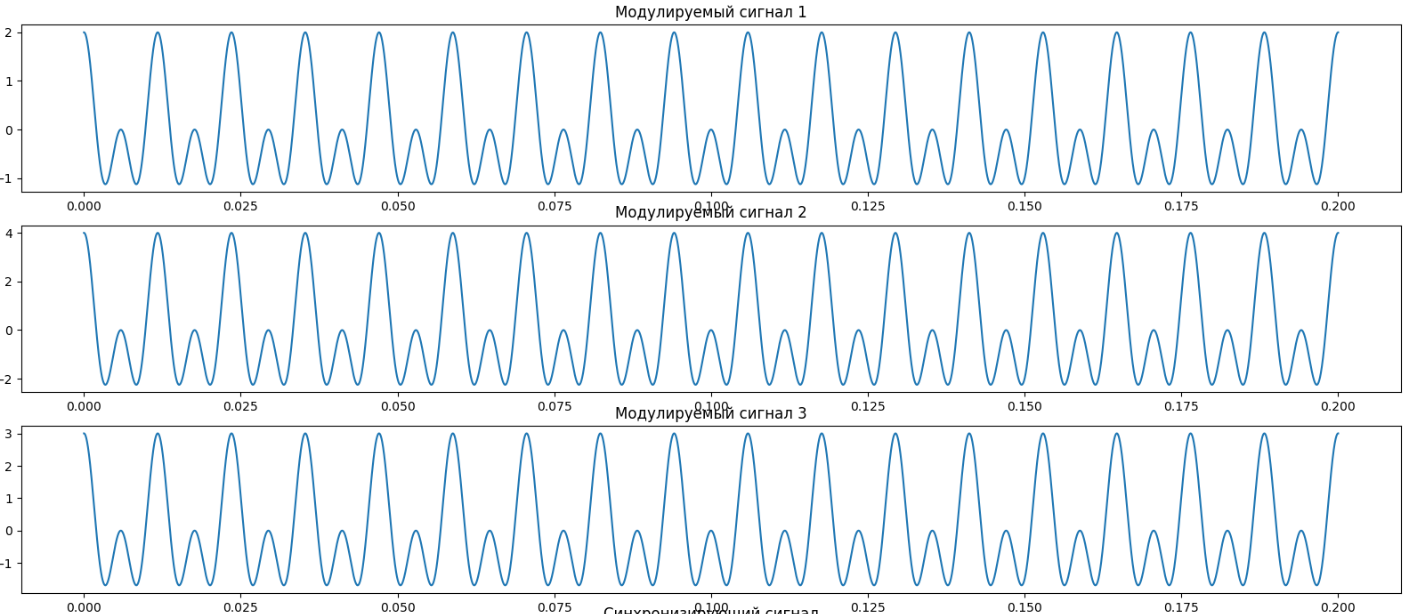


*Рис 10. Групповой сигнал одной гармоники*



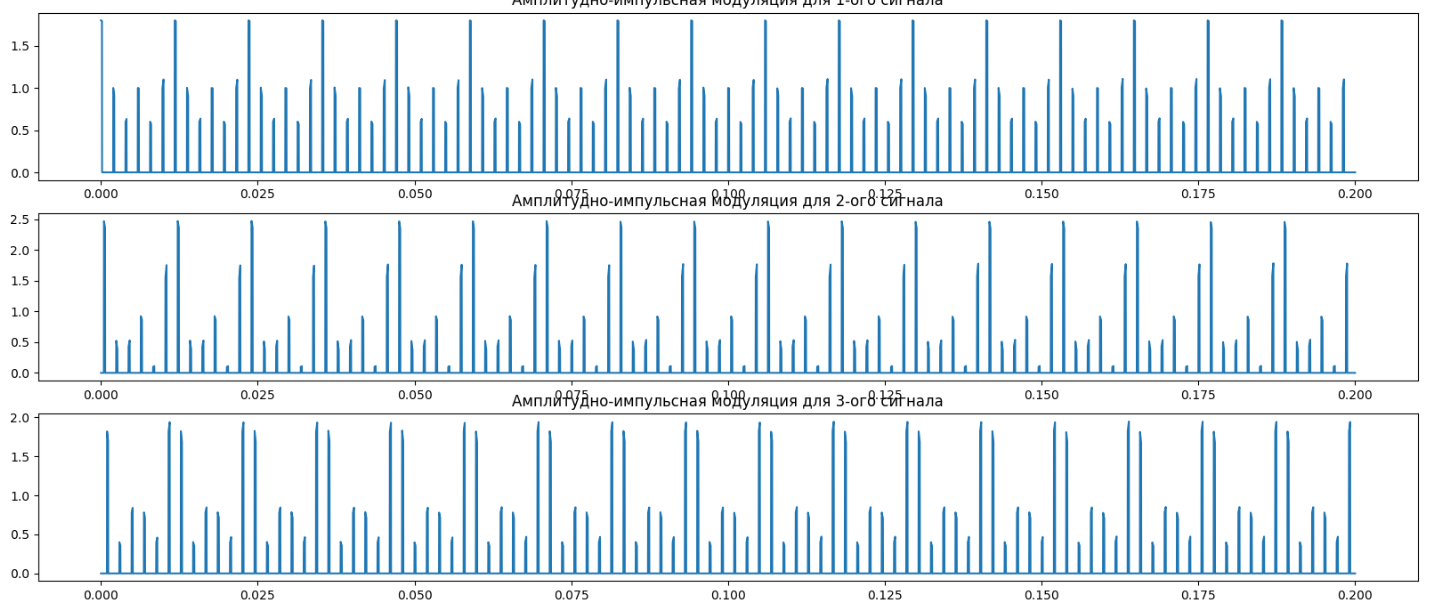
*Рис 11. Восстановленные сигналы одной гармоники*



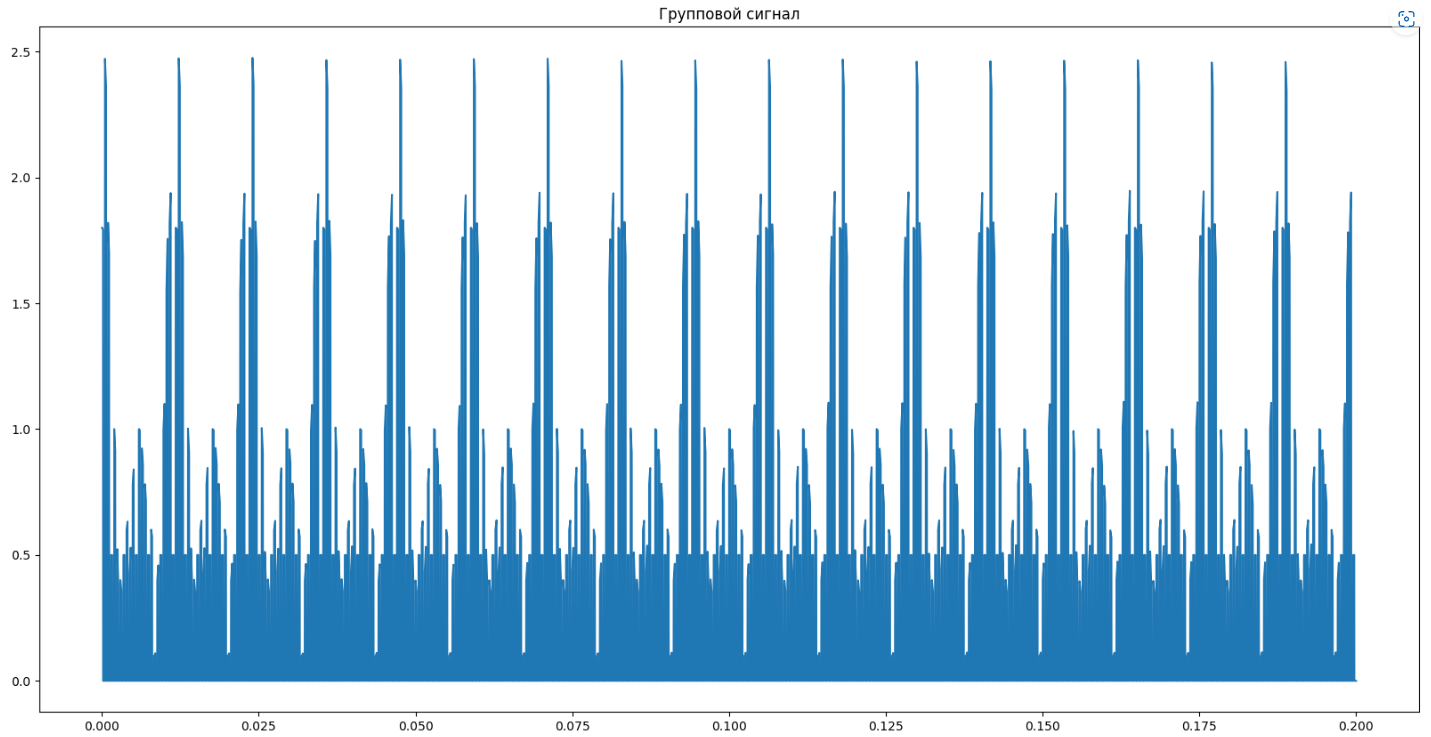
*Рис 12. Моделируемые сигналы двойной гармоники*



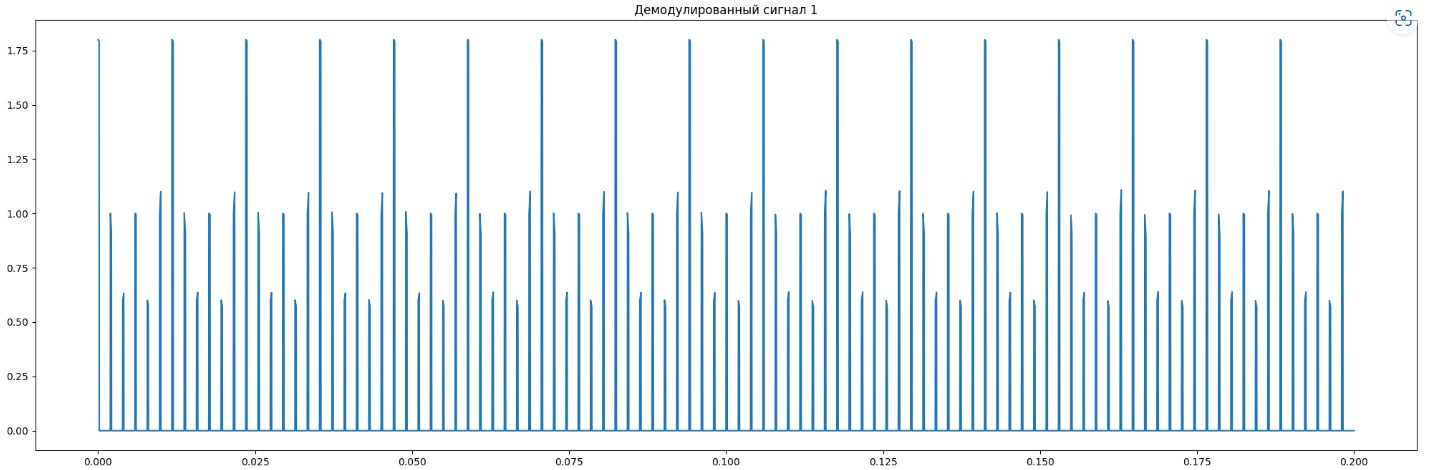
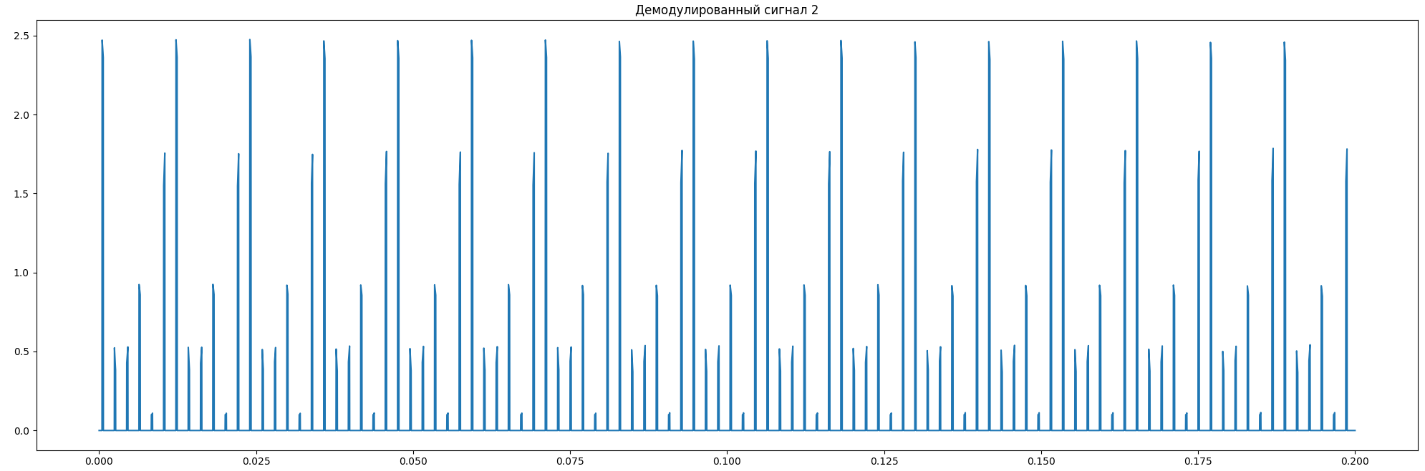
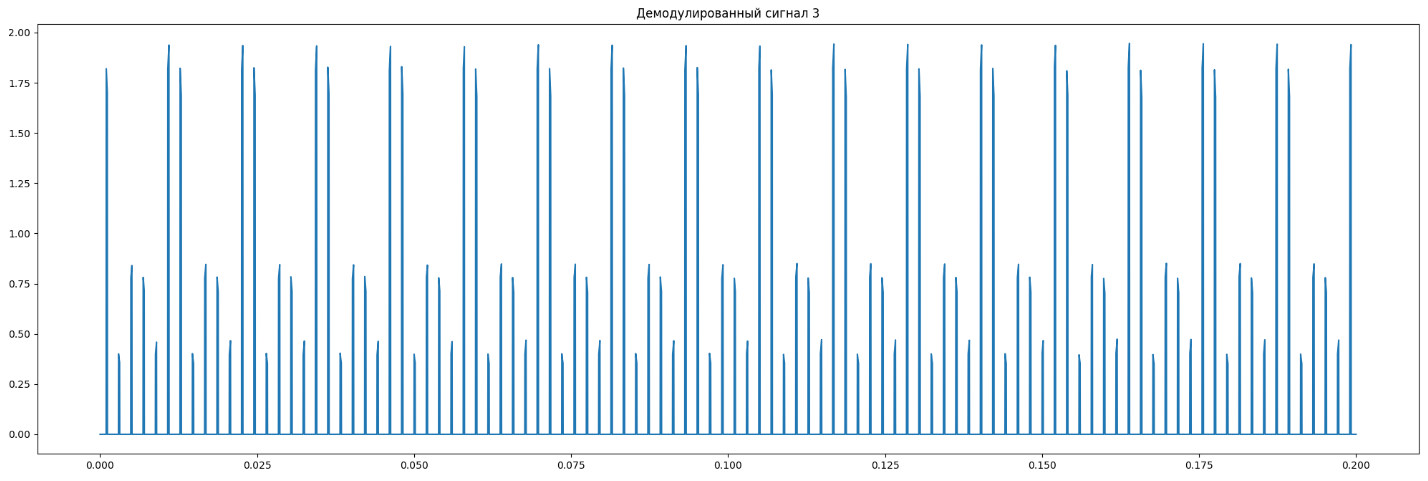
*Рис 13. Синхронизирующий сигнал двойной гармоники*



*Рис 14. Амплитудно-импульсные модуляции сигналов двойной гармоники*



*Рис 15. Групповой сигнал двойной гармоники*

*Рис 16. Восстановленные сигналы двойной гармоники*

**2.3) Реализовать передачу и прием трех сигналов по системе передачи с временным разделением каналов с использованием широтно-импульсной**

**модуляции. И параметрами сигнала:**

**а) каждый сигнал состоит из одной гармоники; глубина модуляции =0,5; скважность ПППИ = 10.**

**б) каждый сигнал состоит из двух гармоник (fs, 2fs); глубина**

**модуляции = 0,8; скважность ПППИ = 10.**

**import numpy as np**

**import matplotlib.pyplot as plt**

**import math**

**def f2w(f):**

**return 2.0 \* math.pi \* f**

**def harm\_signal1(x, fs):**

**return np.sin(f2w(fs) \* x)**

**def harm\_signal2(x, fs):**

**return 0.5 \* (np.sin(f2w(fs) \* x) + np.sin(f2w(2 \* fs) \* x))**

**def triangle(x, Ttr, Atr):**

**return 2 \* Atr / math.pi \* np.arcsin(np.sin(f2w(1 / Ttr) \* (x + Ttr / 4)))**

**fs = 85**

**time = 0.2**

**n = 10000**

**Nimp = int(6 \* fs \* time)**

**pimp = 0.1**

**Aimp = 1**

**# Параметры для одного гармонического сигнала**

**depth\_a = 0.5**

**As1 = 1.0**

**# Параметры для двух гармонических сигналов**

**depth\_b = 0.8**

**As2 = 1.0**

**t = np.linspace(0, time, n)**

**Vin = np.zeros((6, n))**

**Vimp = np.zeros((6, n))**

**# Генерация сигналов**

**for it in range(n):**

**Vin[0, it] = harm\_signal1(t[it], fs)**

**Vin[1, it] = harm\_signal1(t[it], fs)**

**Vin[2, it] = harm\_signal1(t[it], fs)**

**Vin[3, it] = harm\_signal2(t[it], fs)**

**Vin[4, it] = harm\_signal2(t[it], fs)**

**Vin[5, it] = harm\_signal2(t[it], fs)**

**# Моделирующий импульсный сигнал**

**for ii in range(Nimp):**

**start = int(n / Nimp \* ii)**

**for j in range(6):**

**end = start + int(n / Nimp \* pimp)**

**Vimp[j, start:end] = Aimp**

**start = end**

**# Широтно-импульсная модуляция**

**Vpwm = np.zeros((6, n))**

**Vtr = np.zeros(n)**

**Atr = As1 \* 1.5**

**Ttr = 1 / fs / 3**

**# Генерация треугольной волны**

**for it in range(n):**

**Vtr[it] = triangle(t[it], Ttr, Atr)**

**# Генерация ШИМ сигналов**

**indices = [np.argwhere(np.diff(np.sign(Vtr - Vin[i]))).flatten() for i in range(6)]**

**for i in range(6):**

**index = indices[i]**

**j = 0**

**while j < len(index) - 1:**

**Vpwm[i, index[j]:index[j + 1]] = Aimp**

**j += 2**

**# Мультиплексирование сигналов**

**TDM\_signal = np.zeros(n)**

**for i in range(0, n, 6):**

**for j in range(6):**

**if i + j < n:**

**TDM\_signal[i + j] = Vpwm[j, i + j]**

**# Демодуляция сигналов**

**demod = [Vpwm[i] \* (Vin[i] / (Atr / 2)) for i in range(6)]**

**# Визуализация результатов**

**plt.figure(figsize=(25, 30))**

**titles = ['Сигнал 1 (одна гармоника)', 'Сигнал 2 (одна гармоника)', 'Сигнал 3 (одна гармоника)',**

**'Сигнал 4 (две гармоники)', 'Сигнал 5 (две гармоники)', 'Сигнал 6 (две гармоники)']**

**for i in range(6):**

**plt.subplot(7, 3, i \* 3 + 1)**

**plt.title(titles[i])**

**plt.plot(t, Vin[i])**

**plt.subplot(7, 3, i \* 3 + 2)**

**plt.title('Импульсный сигнал ' + str(i + 1))**

**plt.plot(t, Vimp[i])**

**plt.subplot(7, 3, i \* 3 + 3)**

**plt.title('Широтно-импульсная модуляция (сигнал ' + str(i + 1) + ')')**

**plt.plot(t, Vtr)**

**plt.plot(t, Vin[i], '--')**

**plt.plot(t, Vpwm[i])**

**for i in range(2):**

**plt.subplot(7, 4, 19 + i)**

**plt.title('Демодулированный сигнал ' + str(i + 1))**

**plt.plot(t, demod[i])**

**for i in range(2, 6):**

**plt.subplot(7, 3, 17 + i)**

**plt.title('Демодулированный сигнал ' + str(i + 1))**

**plt.plot(t, demod[i])**

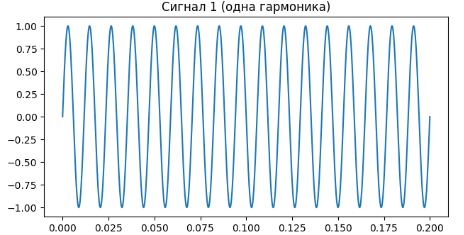
**plt.subplot(7, 3, 21)**

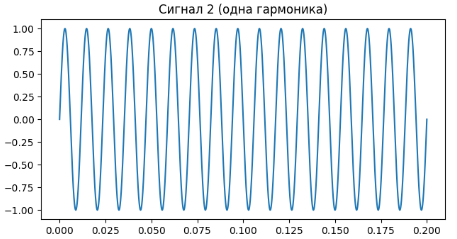
**plt.title('Групповой сигнал')**

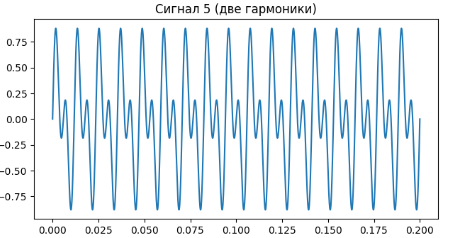
**plt.plot(t, TDM\_signal)**

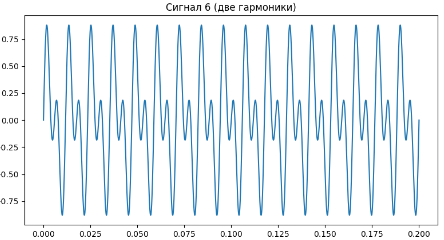
**plt.tight\_layout()**

**plt.show()**

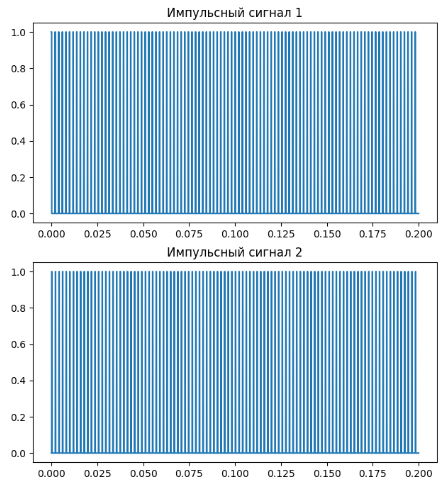


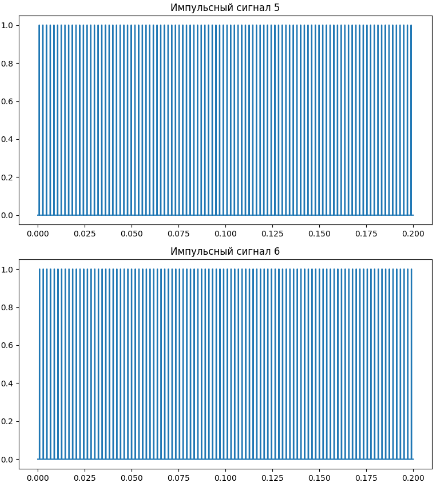




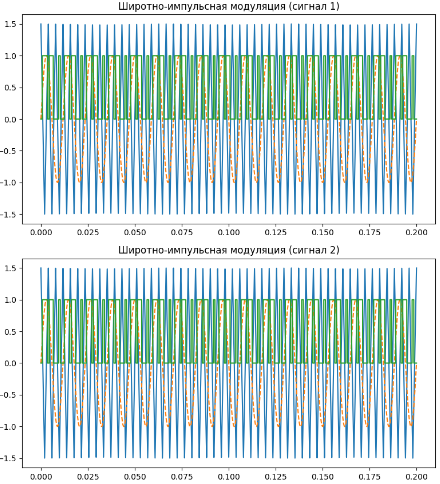


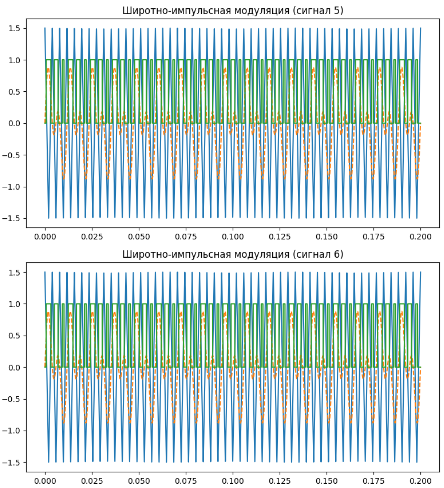
*Рис 17. Исходные сигналы*

**

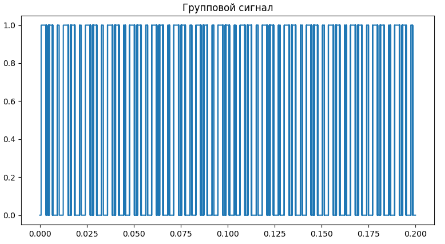


*Рис 18. Импульсные сигналы*

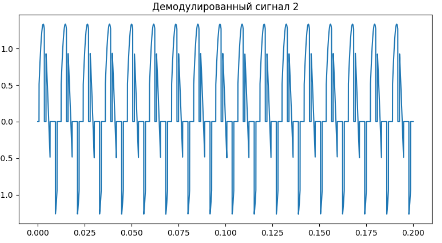
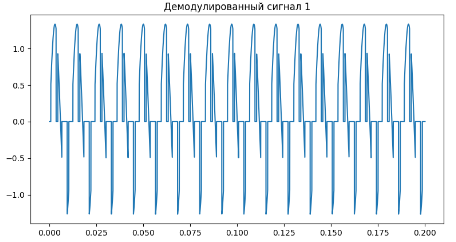
**

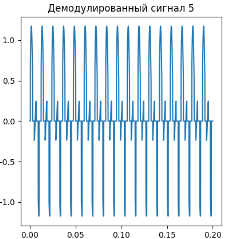
**

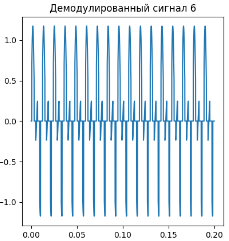
*Рис 19. Широтно-импульсная модуляция*



*Рис 20. Групповой сигнал*







*Рис 21. Демодулированные сигналы*

**2.4) Реализовать передачу и прием трех сигналов по системе передачи с временным разделением каналов с использованием фазово-импульсной**

**модуляции. И параметрами сигнала:**

**а) каждый сигнал состоит из одной гармоники; глубина модуляции =**

**0,5; скважность ПППИ = 10.**

**б) каждый сигнал состоит из двух гармоник (fs, 2fs); глубина**

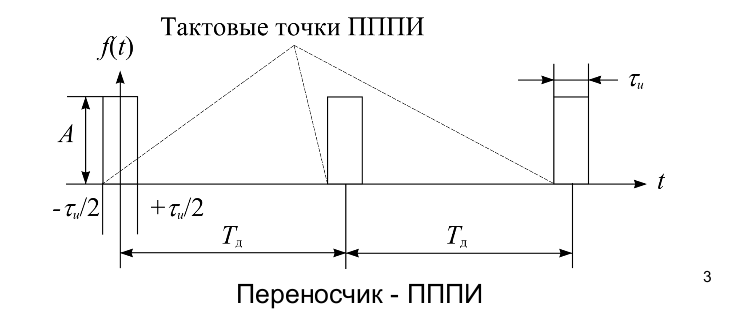
**модуляции = 0,8; скважность ПППИ = 10.**

**Вывод**

В рамках данной работы были смоделированы процессы приема и передачи сигналов по СП с временным разделением каналов с использованием амплитудной, амплитудно-импульсной и широтно-импульсной модуляциями.

**Вопросы**

1. Что такое тактовые точки? Как определяется их положение?



Тактовые точки – временные точки, в которых происходит синхронизация передачи и приема сигнала. Каждая новая точка определяется началом нового прямоугольного импульса

1. Почему в спектре ПППИ присутствуют частоты следования?

ПППИ представляет собой периодический сигнал в виде прямоугольных импульсов и промежутков между ними. В спектре такой сигнал показывает наличие частоты импульсов и гармоник основной частоты. Гармоники формируются из-за того что частоты сигнала периодично повторяются с импульсами.

1. Каким условиям должна удовлетворять ПППИ для использования в СП с ВРК?

Импульс должен быть достаточно длительным чтобы снизить вероятность ошибки, но не превышать выделенный интервал времени для канала.

Частота импульсов должна быть такой чтобы гармоники сигнала не накладывались.

1. Объясните спектральный состав АИМ сигнала.

Спектральный состав АИМ сигнала состоит из несущей частоты w0, соответствующей периоду, принимая значения А0 и двух боковых частот (w0+W); (w0-W), где W – частота модулируемого сигнала и имеет значение ½\*m\*A0, где m – глубина модуляции.

1. Почему в СП с ВРК при использовании ФИМ обязательно будет присутствовать ШИМ?

В ВРК используются фиксированные временные интервалы, при этом изменение фазы сигнала будет влиять на его ширину импульсов.

1. По какому принципу выбираются полосы частот в СП с ЧРК?

Полосы частот выбираются на основе частот первичного сигнала. Они могут быть либо равны исходной полосе, либо превышать ее.

1. Объясните спектральный состав АМ сигнала. Чем он отличается от спектра АИМ сигнала?

При гармонической модуляции сигнала, спектр АМ-сигнала является линейным и состоит из трех спектральных линий:

* Центральная линия соответствует несущей частоте с амплитудой Umh​ и частотой Fh​.
* Две боковые линии имеют частоты fH​ - F (нижняя боковая) и fH​ + F (верхняя боковая) с амплитудой Umb​=2m⋅Umh​​, где m - коэффициент модуляции.

Отличие от спектра АИМ сигнала заключается в наличии двух боковых полос, которые возникают в результате модуляции амплитуды несущего сигнала.

1. Объясните наличие фильтров низких частот в системах передачи.

В СП с ЧРК ФНЧ используются для того, чтобы убрать лишние частоты в спектре.

В СП с ВРК ФНЧ используются для ограничения спектра частот первичных сигналов.

1. Если скважность ПППИ < 1, возможно ли использование ее в СП с ВРК? Если можно, то как?

Если скважность будет меньше 1, то длительность импульса будет больше периода, из этого следует, что импульсы будут перекрываться. СП с ВРК разделяет время на интервалы, в которых передаются данные одного канала. Следовательно в данном случае можно использовать ПППИ < 1, выделяя импульсы из перекрывающихся сигналов с помощью фильтров.

1. По какому принципу производится выбор импульсной модуляции (ИМ) в СП с ВРК? Какую ИМ мы можем считать более эффективной и надежной и почему?

Амплитудно-импульсная модуляции является

помехоустойчивой. Для ее оценки используется отношение мощности полезного Pc сигнала с полосой пропускания dFнч к мощности помехи Pп на выходе ФНЧ. Повысить помехоустойчивость АИМ можно увеличением амплитуды импульсов, но это приводит к снижению эффективности СП. При ШИМ и ФИМ действие помехи проявляется в изменении длительности импульсов и их сдвиге на оси времени относительно их значений в отсутствие помех. Следовательно, происходит смещение тактовых точек.