

Санкт-Петербургский политехнический университет Петра Великого  
Институт компьютерных наук и технологий  
Кафедра компьютерных систем и программных технологий

# Телекоммуникационные технологии

Отчет по лабораторной работе №4

Аналоговая модуляция

**Работу**

**выполнил:**

Графов Д.И.

Группа: 33531/2

**Преподаватель:**

Богач Н.В.

Санкт-Петербург  
2019

# Содержание

1. Цель работы	3
2. Программа работы	3
3. Теоретическая информация	3
4. Ход выполнения работы	4
5. Результаты работы	6
6. Выводы	9

# 1. Цель работы

Изучение амплитудной модуляции/демодуляции сигнала.

# 2. Программа работы

1. Сгенерировать однотоновый сигнал низкой частоты.
2. Выполнить амплитудную модуляцию (АМ) сигнала по закону  $u(t) = (1 + MU_m \cos \Omega t) \cos(\omega_0 t + \phi_0)$  для различных значений глубины модуляции.
3. Получить спектр модулированного сигнала.
4. Выполнить модуляцию с подавлением несущей  $u(t) = MU_m \cos(\Omega t) \cos(\omega_0 t + \phi_0)$ . Получить спектр.
5. Выполнить одностороннюю модуляцию:  $u(t) = U_m \cos(\omega t) \cos(\omega_0 t + \phi_0) + \frac{U_m}{2} \sum_{n=1}^N (M_n \cos(\omega_0 + \Omega_n)t + \phi_0 + \Phi_n)$  положив  $n=1$ .
6. Выполнить синхронное детектирование и получить исходный односторонний сигнал.
7. Рассчитать КПД модуляции.  $\eta_{AM} = \frac{U_m^2 M^2 / 4}{P_U} = \frac{M^2}{M^2 + 2}$

# 3. Теоретическая информация

Модуляция (лат. modulatio — размеренность, ритмичность) — процесс изменения одного или нескольких параметров модулируемого несущего сигнала при помощи модулирующего сигнала.

Амплитудная модуляция — вид модуляции, при которой изменяемым параметром несущего сигнала является его амплитуда.

Разновидности амплитудной модуляции:

- АМ с подавлением несущей (Suppressed carrier) - удаление несущего колебания. Демодуляция выполняется путём синхронного детектирования.
- Односторонняя модуляция (Single sideband) - удаление одной из боковых полос.

Демодуляция АМ-сигнала может быть выполнена несколькими способами. Простейший путь - вычислить модуль входного АМ-сигнала, затем сгладить получившиеся односторонние косинусоидальные импульсы, пропуская их через фильтр низких частот.

## 4. Ход выполнения работы

Данная работа выполнялась на языке Python.

Характеристики сигналов:

- частота модулируемого сигнала: 10 Гц;
- амплитуда модулируемого сигнала: 1;
- частота несущей: 100 Гц;
- амплитуда несущей: 1.

Для начала, по заданным формулам был закодирован сигнал (строки 17-19). Затем, была осуществлено детектирование закодированных сигналов: для первого примера с помощью функции `filfilt()` от модуля закодированного сигнала; и для второго примера, с помощью умножения на несущую.

После этого, был произведён расчёт спектров полученных сигналов.

Листинг 1. `main.py`

---

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 from scipy.signal import butter, filtfilt, hilbert
4
5
6 def sine_wave(amplitude=1, freq=10, show=False, save=False):
7     fs = 2000 # sampling rate
8     ts = 1 / fs # sampling interval
9     n = 2 ** 13 # number of fft points, pick power of 2
10    t = np.arange(0, n * ts, ts) # time vector
11
12    carrier_freq = 100
13    carrier_amplitude = 1
14    carrier = np.sin(2 * np.pi * carrier_freq * t)
15    sig = np.sin(2 * np.pi * freq * t)
16    m = amplitude / carrier_amplitude
17    singleton_modulated = (1 + m * amplitude * sig) * carrier_amplitude * carrier
18    suppressed_carried_mod = amplitude * carrier_amplitude * sig * carrier
19    single_sideband_mod = hilbert(sig) * np.cos(2 * np.pi * carrier_freq * t) - hilbert(sig) * carrier
20
21    order = 5
22    normal_cutoff = freq / carrier_freq
23
24    fnum, fdenom = butter(order, normal_cutoff)
25    singleton_demodulated = filtfilt(fnum, fdenom, abs(singleton_modulated))
26    suppressed_carried_demod = suppressed_carried_mod * carrier_amplitude * carrier
27
28    fft_freq = np.fft.fftfreq(n, ts) # python function to get Hz frequency axis
29
30    singleton_modulated_fft = \
31        abs(np.fft.fft(singleton_modulated)) / n * 2 # discrete Fourier Transform ( / n * 2 - normaliz
```

```

32     singleton_demodulated_fft = abs(np.fft.fft(singleton_demodulated)) / n * 2
33
34     suppressed_carried_mod_fft = abs(np.fft.fft(suppressed_carried_mod)) / n * 2
35     suppressed_carried_demod_fft = abs(np.fft.fft(suppressed_carried_demod)) / n * 2
36
37     single_sideband_mod_fft = abs(np.fft.fft(single_sideband_mod)) / n * 2
38
39     draw_plot(t, (singleton_modulated, singleton_demodulated),
40               ('Singleton Modulated sine wave', 'Singleton Demodulated sine wave'),
41               'time (s)', 'amplitude (V)', show, save)
42     draw_plot(fft_freq[1:], (singleton_modulated_fft[1:], singleton_demodulated_fft[1:]),
43               ('Singleton Modulated sine wave', 'Singleton Demodulated sine wave'),
44               'frequency (hz)', 'amplitude (V)', show, save)
45
46     draw_plot(t, (suppressed_carried_mod, suppressed_carried_demod),
47               ('Suppressed carrier Modulated sine wave', 'Suppressed carrier Demodulated sine wave'),
48               'time (s)', 'amplitude (V)', show, save)
49     draw_plot(fft_freq, (suppressed_carried_mod_fft, suppressed_carried_demod_fft),
50               ('Suppressed carrier Modulated sine wave', 'Suppressed carrier Demodulated sine wave'),
51               'frequency (hz)', 'amplitude (V)', show, save)
52
53     draw_plot(t, (single_sideband_mod,),
54               ('Single sideband Modulated sine wave',),
55               'time (s)', 'amplitude (V)', show, save)
56     draw_plot(fft_freq, (single_sideband_mod_fft,),
57               ('Single sideband carrier Modulated sine wave',),
58               'frequency (hz)', 'amplitude (V)', show, save)
59
60
61     def draw_plot(x, y, labels, xlabel, ylabel, show=False, save=False):
62         size = 500
63         plt.figure()
64         for i in range(0, len(y)):
65             plt.plot(x[:size], y[i][:size], label=labels[i])
66         plt.xlabel(xlabel)
67         plt.ylabel(ylabel)
68         plt.legend()
69         plt.grid(True)
70         if save:
71             plt.savefig('../out/' + labels[0].split(' ')[0] + '_' + xlabel.split(' ')[0] + '.png')
72         if show:
73             plt.show()
74
75
76     if __name__ == '__main__':
77         pic = 0
78         sine_wave(show=True, save=True)

```

---

## 5. Результаты работы

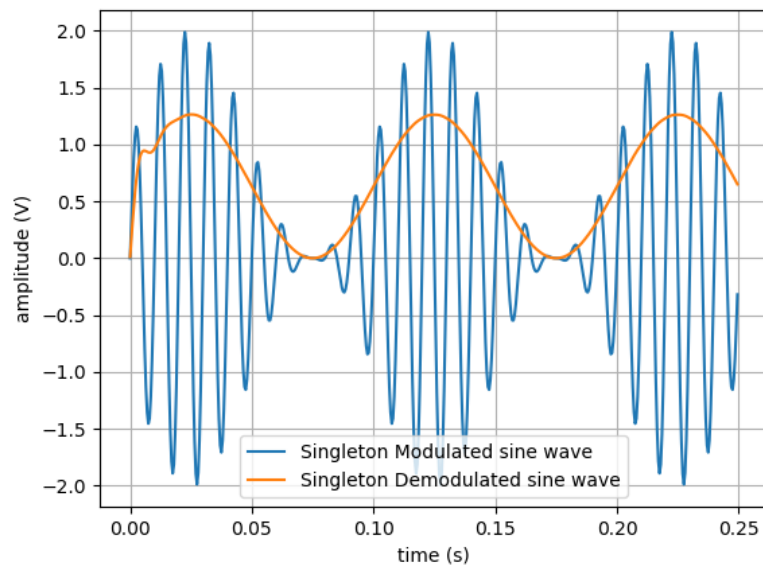


Рисунок 5.1. Амплитудная (однотональная) модуляция и демодуляция

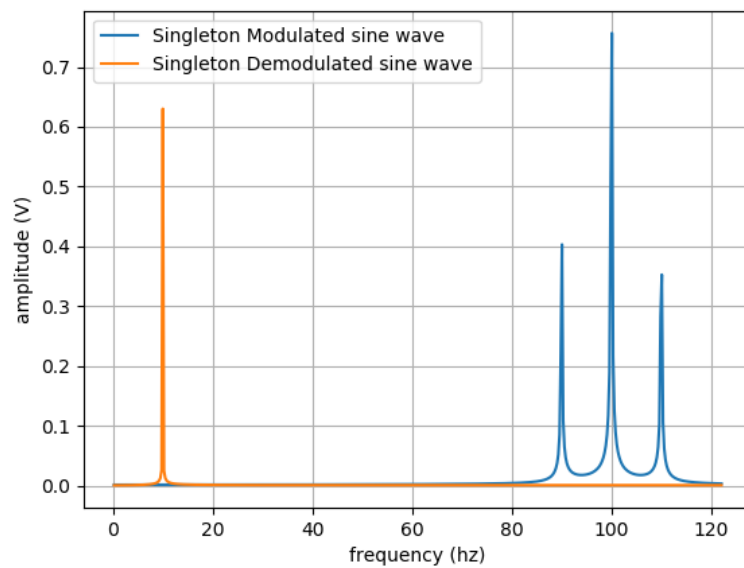


Рисунок 5.2. Спектр промодулированного сигнала

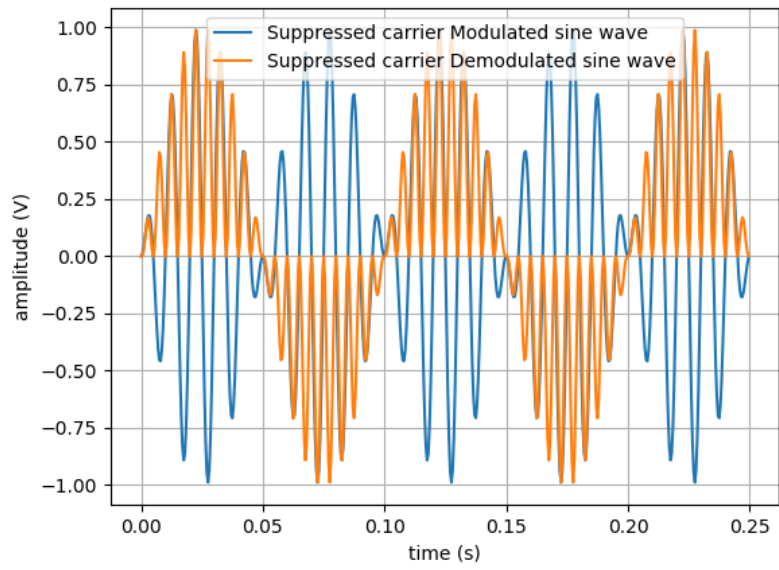


Рисунок 5.3. Модуляция и демодуляция с подавлением несущей

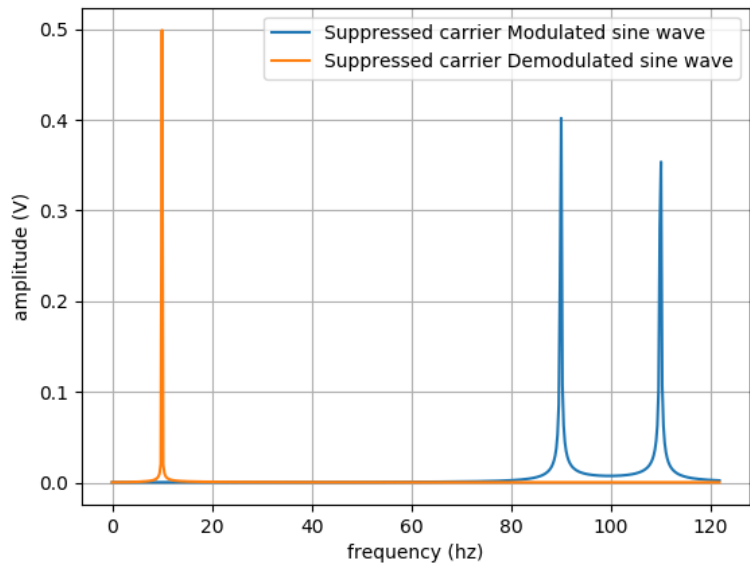


Рисунок 5.4. Спектр промодулированного сигнала

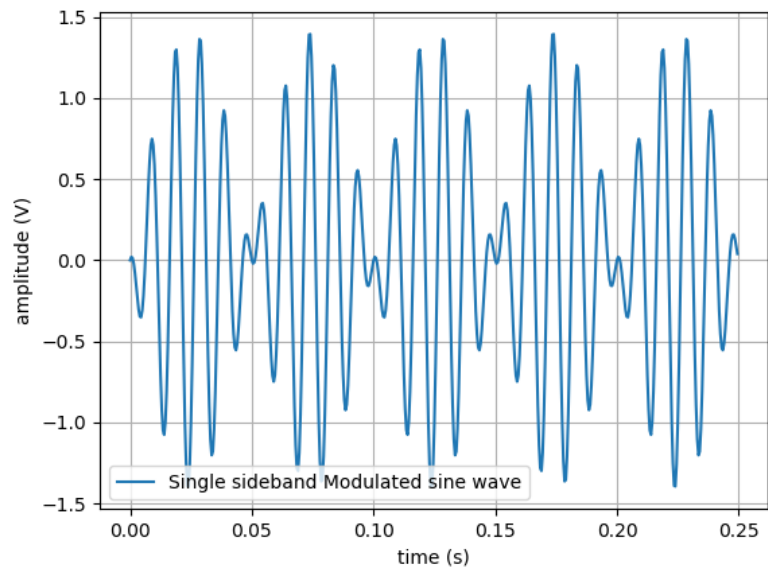


Рисунок 5.5. Однополосная модуляция

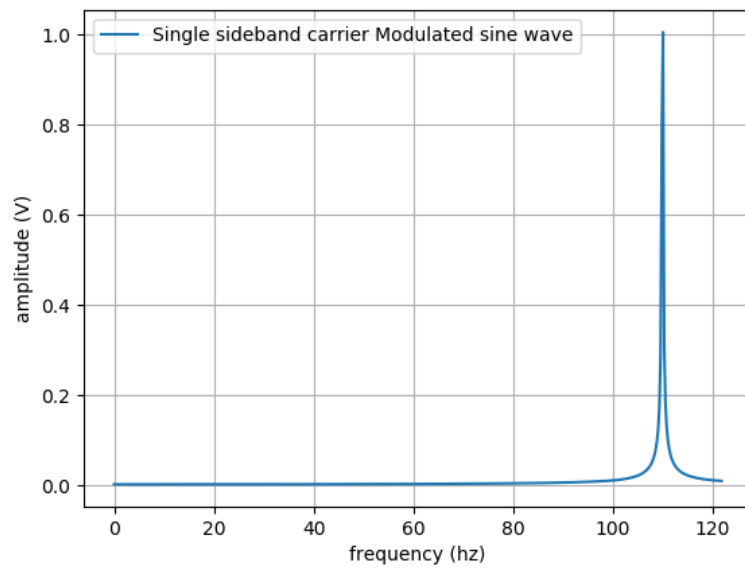


Рисунок 5.6. Спектр промодулированного сигнала



## 6. Выводы

В ходе выполнения работы я ознакомился с амплитудной модуляцией и демодуляцией, их разновидностями. Для однотоновой модуляции и модуляции с подавлением несущей были продемонстрированы частотные и временные характеристики закодированных и декодированных сигналов. Декодировать последний сигнал не удалось.