

Санкт-Петербургский Политехнический Университет
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Отчёт по лабораторной работам №4 и №5

Дисциплина: Телекоммуникационные технологии

Тема: Аналоговая модуляция. Фазовая и частотная модуляция.

Работу выполнил студент

гр. 33501/4 Леженин Ю.И.

Преподаватель

Богач Н.В.

Санкт-Петербург
29 апреля 2018 г.

1 Цель работы.

Изучение амплитудной модуляции/демодуляции сигнала. Изучение частотной и фазовой модуляции/демодуляции сигнала.

2 Постановка задачи.

Сгенерировать однотоновый сигнал низкой частоты. Выполнить амплитудную модуляцию/демодуляцию для различных значений глубины модуляции, амплитудную модуляцию/демодуляцию с подавлением несущей, однополосную модуляцию и синхронное детектирование. Выполнить фазовую и частотную модуляцию/демодуляцию. Получить спектры модулированных сигналов.

3 Ход работы.

3.1 Амплитудная модуляция.

Амплитудно-модулированный сигнал задается по закону

$$\begin{aligned} u(t) &= U_m [1 + M \cos(\Omega t)] \cos(\omega_0 t) = \\ &= U_m \left[\cos(\omega_0 t) + \frac{M}{2} \cos((\omega_0 + \Omega) t) + \frac{M}{2} \cos((\omega_0 - \Omega) t) \right], \end{aligned}$$

где M – глубина модуляции, Ω – частота модулируемого сигнала, ω_0 – несущая частота.

Для повышения КПД используется балансная модуляция:

$$u(t) = U_m \cos(\Omega t) \cos(\omega_0 t).$$

Однако отсутствие опорного сигнала с несущей частотой значительно усложняет демодуляцию.

Кроме того, ввиду идентичности информации на частотах ниже и выше несущей, может выполняться однополосная амплитудная модуляция:

$$u(t) = U_m M \cos((\omega_w \pm \Omega) t).$$

Для демонстрации амплитудной модуляции был сгенерирован однотоновый сигнал, для которого была выполнена модуляция при $M = 1$, $M = 0.5$, модуляция с подавлением несущей и однополосная модуляция. Результаты представлены на рисунках 3.1, 3.2, 3.3, 3.4.

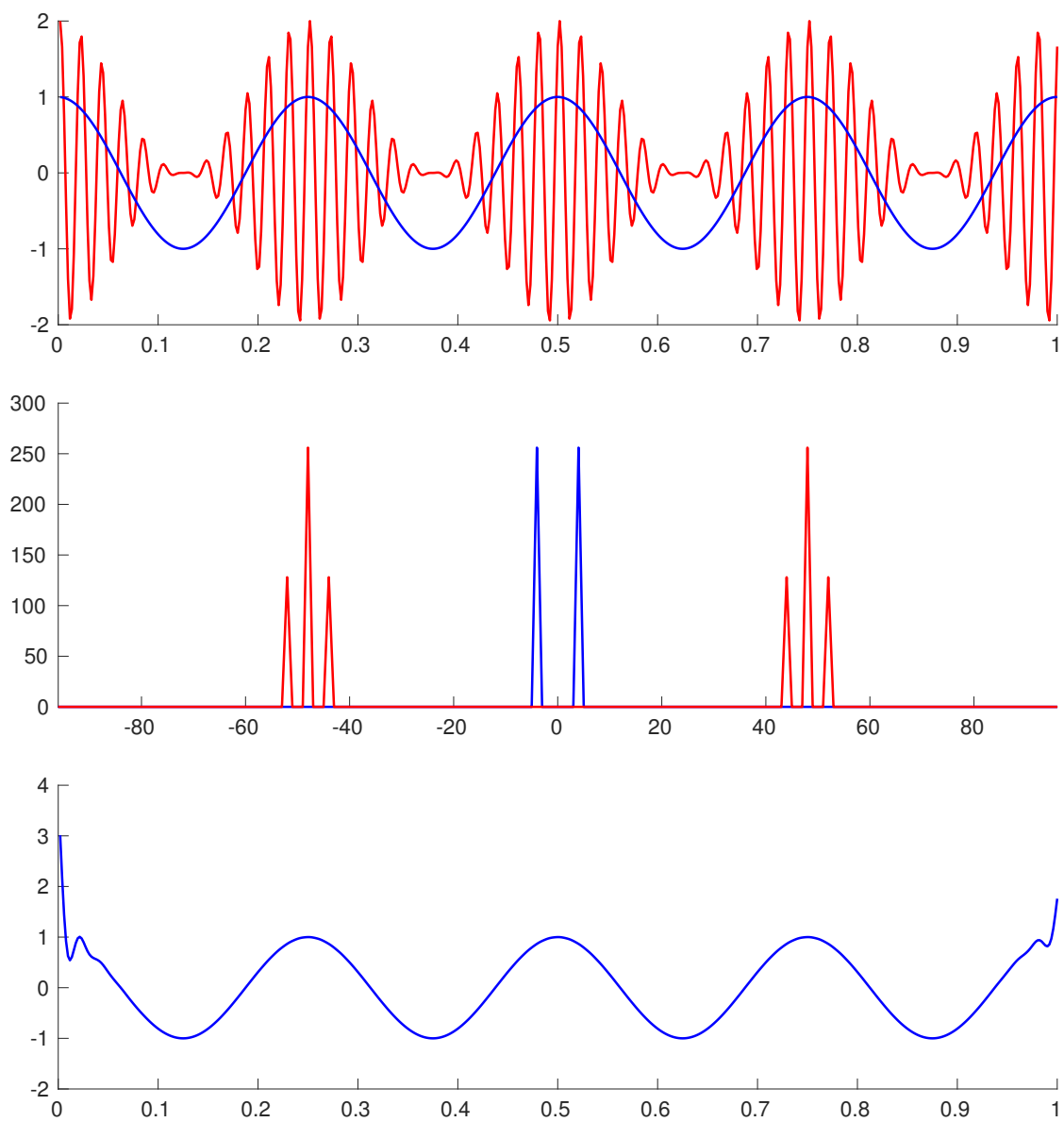


Рис. 3.1: Модулируемый однотоновый сигнал ($\Omega = 4$ Гц), амплитудно-модулированный сигнал ($\omega_0 = 64$ Гц, $M = 1$), их спектры, демодулированный сигнал.

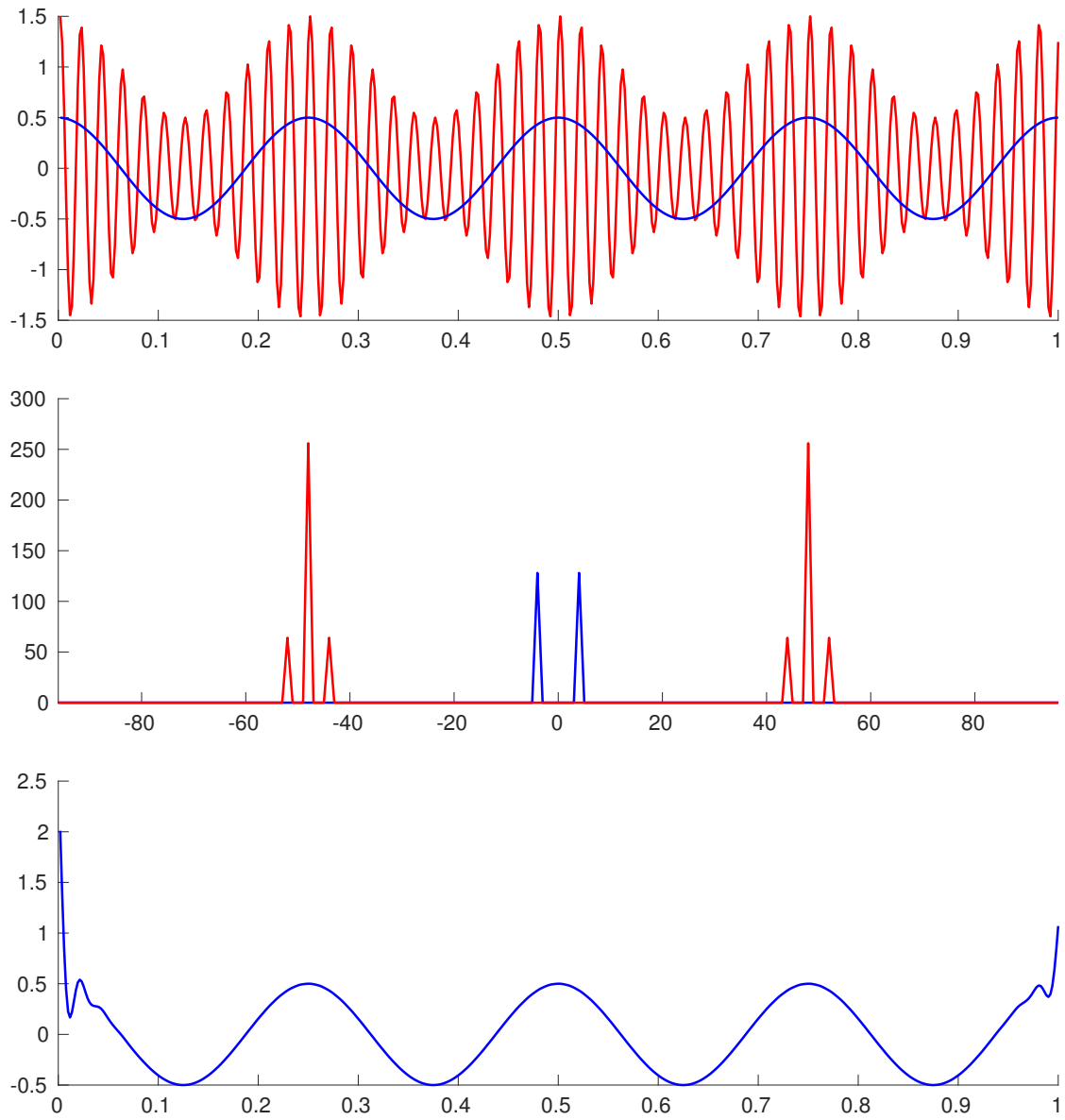


Рис. 3.2: Модулируемый однотоновый сигнал ($\Omega = 4$ Гц), амплитудно-модулированный сигнал ($\omega_0 = 64$ Гц, $M = 0.5$), их спектры, демодулированный сигнал.

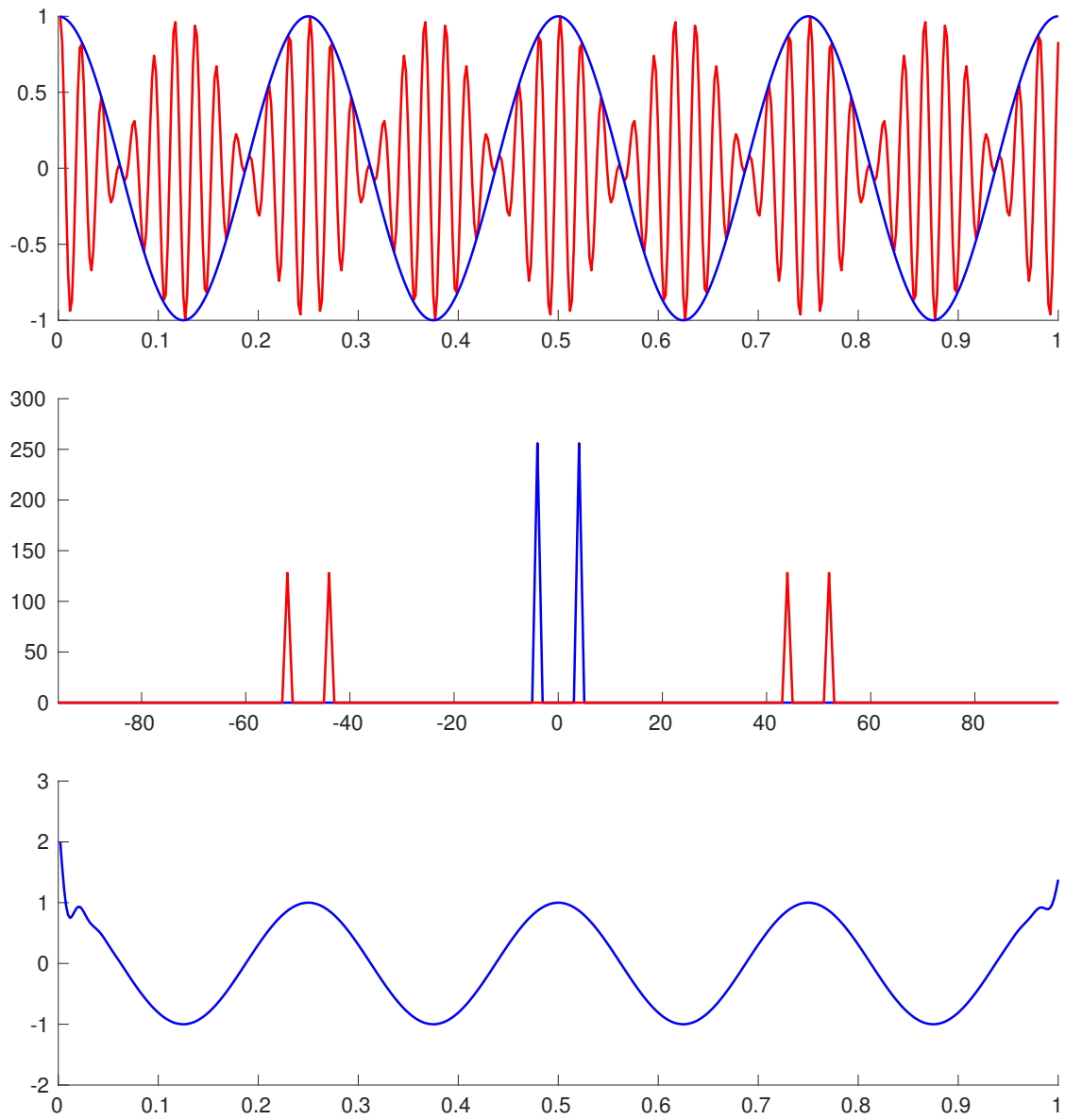


Рис. 3.3: Модулируемый однотональный сигнал ($\Omega = 4$ Гц), амплитудно-модулированный сигнал ($\omega_0 = 64$ Гц) с подавлением несущей, их спектры, демодулированный сигнал.

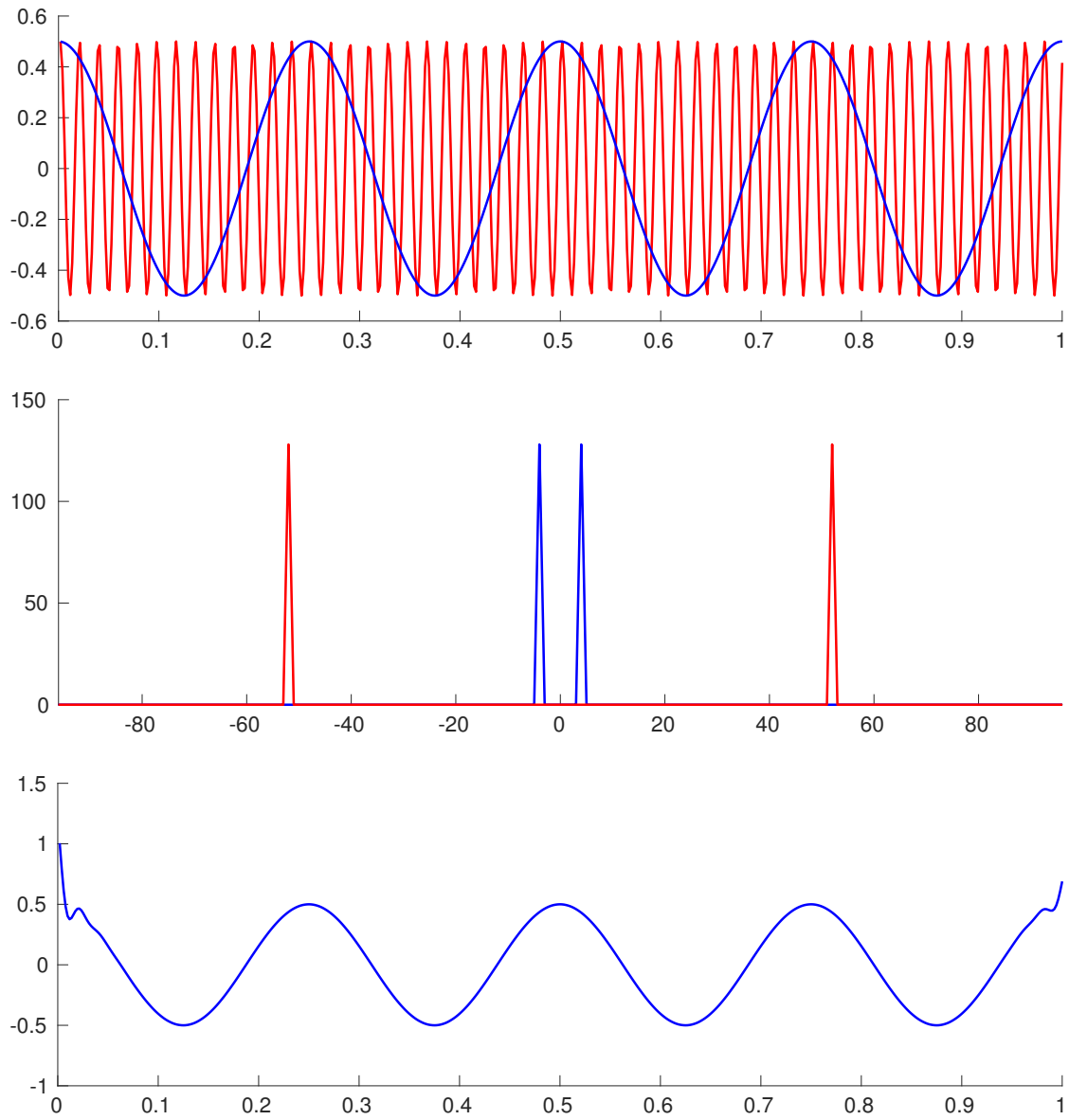


Рис. 3.4: Модулируемый однотональный сигнал ($\Omega = 4$ Гц), однополосный амплитудно-модулированный сигнал ($\omega_0 = 64$ Гц) с подавлением несущей, их спектры, демодулированный сигнал.

3.2 Частотная и фазовая модуляция.

При фазовой модуляции фаза модулированного сигнала пропорциональна амплитуде передаваемого сигнала:

$$u(t) = U_m \cos(\omega_0 t + k s(t)).$$

При этом полная фаза колебаний задается выражением

$$\psi(t) = \omega_0 t + k s(t).$$

Девияция фазы вниз и вверх определяется следующим образом:

$$\Delta\phi_{\text{в}} = k s_{\text{max}}, \quad \Delta\phi_{\text{н}} = k s_{\text{min}}.$$

Мгновенная частота является производной от полной фазы:

$$\omega(t) = \frac{d}{dt}\psi(t) = \omega_0 + k \cdot \frac{d}{dt}s(t).$$

Видно, что при фазовой модуляции частота модулированного сигнала зависит от производной модулируемого сигнала.

При фазовой модуляции фаза модулированного сигнала пропорциональна интегралу передаваемого сигнала:

$$u(t) = U_m \cos(\omega_0 t + k \int_0^t s(t) dt).$$

При этом полная фаза колебаний определяется как

$$\psi(t) = \omega_0 t + k \int_0^t s(t) dt,$$

а мгновенная частота

$$\omega(t) = \frac{d}{dt}\psi(t) = \omega_0 + k \cdot s(t).$$

Девияция частоты вниз и вверх находится следующим образом:

$$\Delta\omega_{\text{в}} = k s_{\text{max}}, \quad \Delta\omega_{\text{н}} = k s_{\text{min}}.$$

Частота модулированного сигнала зависит от амплитуды модулируемого сигнала.

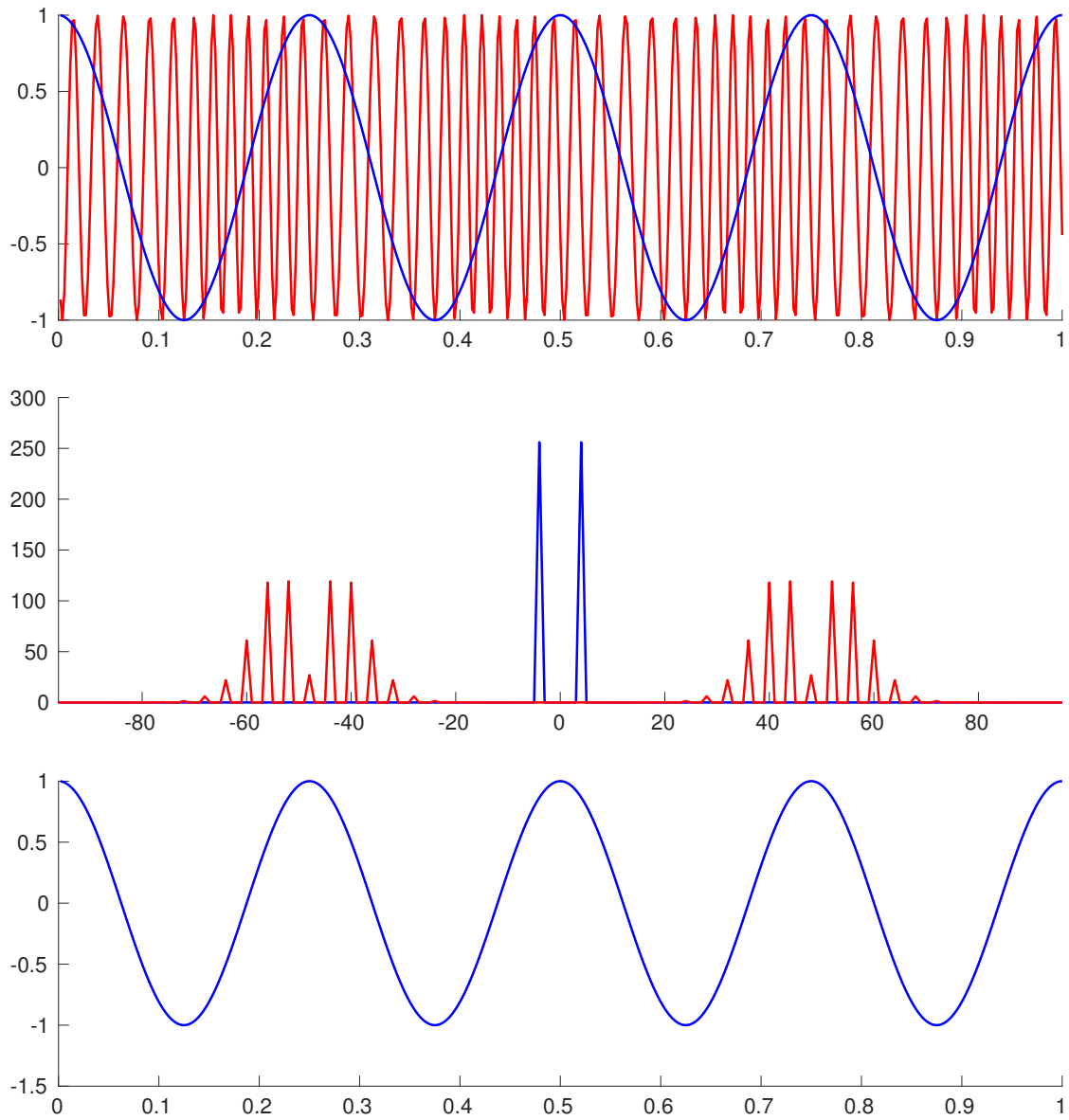


Рис. 3.5: Модулируемый однотоновый сигнал ($\Omega = 4$ Гц), фазово-модулированный сигнал ($\omega_0 = 48$ Гц, $\Delta\phi = \frac{5\pi}{6}$), их спектры, демодулированный сигнал.

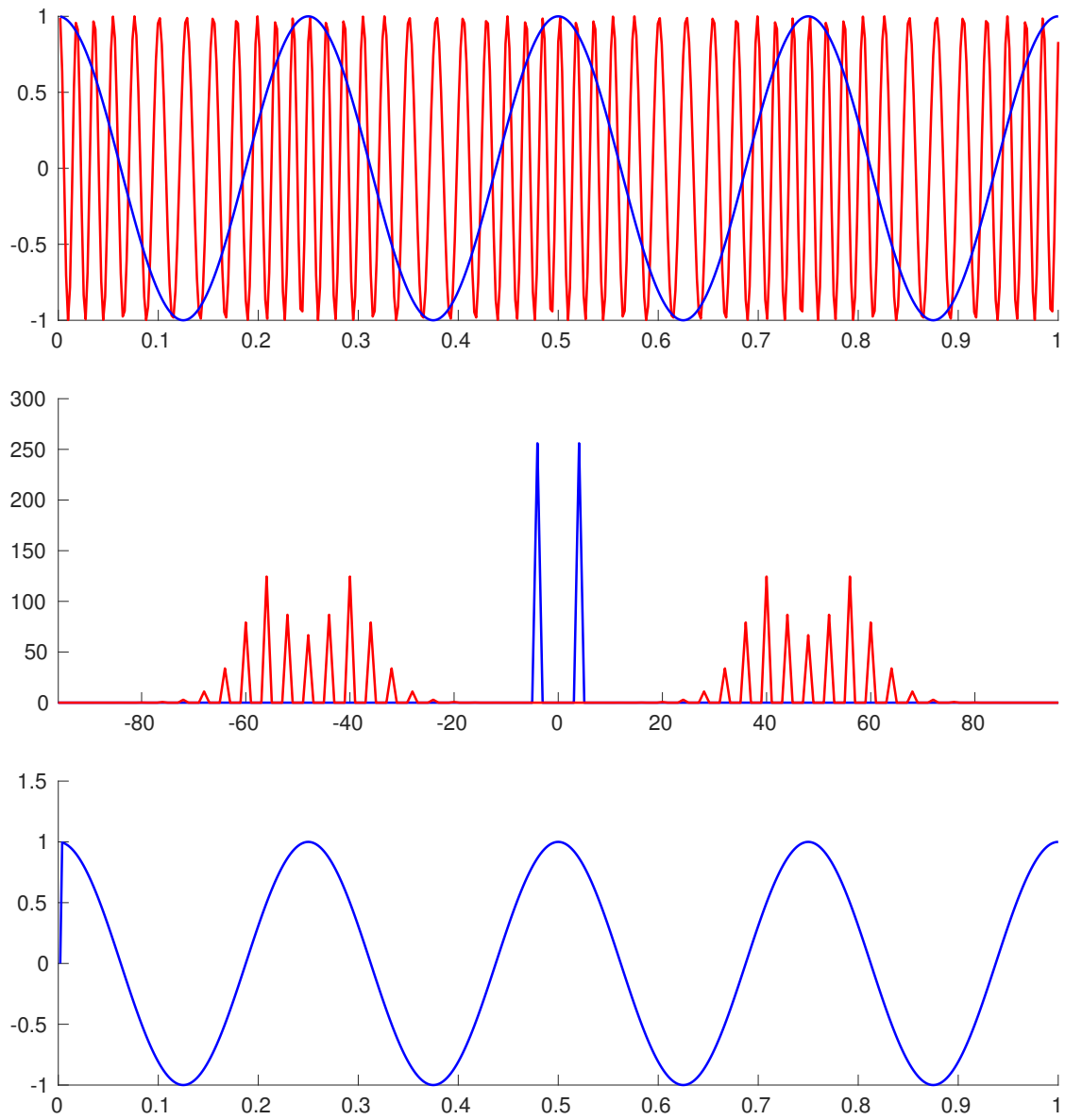


Рис. 3.6: Модулируемый однотоновый сигнал ($\Omega = 4$ Гц), частотно-модулированный сигнал ($\omega_0 = 48$ Гц, $\Delta\omega = 12$ Гц), их спектры, демодулированный сигнал.

4 Выводы.

Модуляция позволяет перенести информацию содержащуюся в одном сигнале на другой сигнал, обладающими необходимыми свойствами. Модуляция дает возможность перенести спектр сигнала на заданный диапазон частот и выполнить частотное разделение каналов, что позволяет более эффективно использовать передающее оборудование.

Как правило, для модуляции используют гармонические колебания. В зависимости от того, на какой из параметров переносится информация, различают амплитудную (АМ), частотную (ЧМ) или фазовую (ФМ) модуляцию несущего сигнала.

Фазовая и частотная модуляции позволяют расширить спектра сигнала и тем самым повысить помехоустойчивость передачи.

5 Приложение.

Листинг 1: Программа для выполнения амплитудной модуляции/демодуляции.

```
1 close all;
2 clear all;
3
4 figure_properties = {'units', 'centimeters', 'position',
5                     , [12, 10, 30, 10], ...
6                     'DefaultAxesPosition', [0.08, 0.17, 0.88, 0.8]};
7 lw = 1.8;
8 fs = 16;
9
10 N = 512;
11 Fs = 512;
12 Fc = 48;
13 F0 = 4;
14 M = 1;
15
16 t = (1:N)/Fs;
17 f = (-N/2:N/2-1) .* (Fs/N);
18
19 x = M * cos(2 * pi * F0 * t); % 0.5 * sin(2 * pi * F0
20                               /2 * t);
21 y_amod = ammod(x, Fc, Fs, 0, 1);
```

```

21 y_ssbm = ssbmod(x, Fc, Fs, 0, 'upper');
22
23 y_ademod = amdemod(y_amod, Fc, Fs, 0, 1);
24 y_ssbdemod = ssbdemod(y_ssbm, Fc, Fs);
25
26 s = fftshift(fft(x));
27 s_amod = fftshift(fft(y_amod));
28 s_ssbm = fftshift(fft(y_ssbm));
29
30 s_ademod = fftshift(fft(y_ademod));
31 s_ssbdemod = fftshift(fft(y_ssbdemod));
32
33 figure (figure_properties{:})
34 hold on
35 plot(t, y_amod, 'r', 'LineWidth', lw)
36 plot(t, x, 'b', 'LineWidth', lw)
37 set(gca, 'FontSize', fs)
38
39 figure (figure_properties{:})
40 hold on
41 plot(f, abs(s), 'b', 'LineWidth', lw)
42 plot(f, abs(s_amod), 'r', 'LineWidth', lw)
43 %xlim([-Fs/2, Fs/2]);
44 xlim([-Fc*2, Fc*2]);
45 set(gca, 'FontSize', fs)
46
47 figure (figure_properties{:})
48 hold on
49 plot(t, y_ssbm, 'r', 'LineWidth', lw)
50 plot(t, x, 'b', 'LineWidth', lw)
51 set(gca, 'FontSize', fs)
52
53 figure (figure_properties{:})
54 hold on
55 plot(f, abs(s), 'b', 'LineWidth', lw)
56 plot(f, abs(s_ssbm), 'r', 'LineWidth', lw)
57 %xlim([-Fs/2, Fs/2]);
58 xlim([-Fc*2, Fc*2]);
59 set(gca, 'FontSize', fs)
60
61 figure (figure_properties{:})

```

```

62 hold on
63 plot(t, y_ademod, 'b', 'LineWidth', lw)
64 % plot(t, x, 'b', 'LineWidth', lw)
65 set(gca, 'FontSize', fs)
66
67 figure (figure_properties{:})
68 hold on
69 plot(f, abs(s_ademod), 'b', 'LineWidth', lw)
70 xlim([-Fs/2, Fs/2]);
71 set(gca, 'FontSize', fs)
72
73 figure (figure_properties{:})
74 hold on
75 plot(t, y_ssbдемод, 'b', 'LineWidth', lw)
76 % plot(t, x, 'b', 'LineWidth', lw)
77 set(gca, 'FontSize', fs)
78
79 figure (figure_properties{:})
80 hold on
81 plot(f, abs(s_ssbдемод), 'b', 'LineWidth', lw)
82 xlim([-Fs/2, Fs/2]);
83 set(gca, 'FontSize', fs)

```

Листинг 2: Программа для выполнения частотной и фазовой модуляции/демодуляции.

```

1 close all;
2 clear all;
3
4 figure_properties = {'units', 'centimeters', 'position',
5     , [12, 10, 30, 10], ...
6     'DefaultAxesPosition', [0.08, 0.17, 0.88, 0.8]};
7 lw = 1.8;
8 fs = 16;
9
10 N = 512;
11 Fs = 512;
12 Fc = 48;
13 F0 = 4;
14
15 t = (1:N)/Fs;
16 f = (-N/2:N/2-1) .* (Fs/N);

```

```

16
17 x = cos(2 * pi * F0 * t);
18
19 y_pmod = pmmmod(x, Fc, Fs, 5 * pi / 6);
20 y_fmod = fmmmod(x, Fc, Fs, 12);
21
22 y_pdemod = pmdemod(y_pmod, Fc, Fs, 5 * pi / 6);
23 y_fdemod = fmdemod(y_fmod, Fc, Fs, 12);
24
25
26 s = fftshift(fft(x));
27 s_pmod = fftshift(fft(y_pmod));
28 s_fmod = fftshift(fft(y_fmod));
29
30 s_pdemod = fftshift(fft(y_pdemod));
31 s_fdemod = fftshift(fft(y_fdemod));
32
33 figure (figure_properties{:})
34 hold on
35 plot(t, y_pmod, 'r', 'LineWidth', lw)
36 plot(t, x, 'b', 'LineWidth', lw)
37 set(gca, 'FontSize', fs)
38
39 figure (figure_properties{:})
40 hold on
41 plot(f, abs(s), 'b', 'LineWidth', lw)
42 plot(f, abs(s_pmod), 'r', 'LineWidth', lw)
43 xlim([-Fc*2, Fc*2]);
44 set(gca, 'FontSize', fs)
45
46 figure (figure_properties{:})
47 hold on
48 plot(t, y_fmod, 'r', 'LineWidth', lw)
49 plot(t, x, 'b', 'LineWidth', lw)
50 set(gca, 'FontSize', fs)
51
52 figure (figure_properties{:})
53 hold on
54 plot(f, abs(s), 'b', 'LineWidth', lw)
55 plot(f, abs(s_fmod), 'r', 'LineWidth', lw)
56 xlim([-Fc*2, Fc*2]);

```

```

57 set(gca, 'FontSize', fs)
58
59 figure (figure_properties{:})
60 hold on
61 plot(t, y_pdemod, 'b', 'LineWidth', lw)
62 %plot(t, x, 'b', 'LineWidth', lw)
63 set(gca, 'FontSize', fs)
64
65 % figure (figure_properties{:})
66 % hold on
67 % plot(f, abs(s_pdemod), 'b', 'LineWidth', lw)
68 % xlim([-Fc*2, Fc*2]);
69
70 figure (figure_properties{:})
71 hold on
72 plot(t, y_fdemod, 'b', 'LineWidth', lw)
73 %plot(t, x, 'b', 'LineWidth', lw)
74 set(gca, 'FontSize', fs)
75
76 % figure (figure_properties{:})
77 % hold on
78 % plot(f, abs(s_fdemod), 'b', 'LineWidth', lw)
79 % xlim([-Fc*2, Fc*2]);

```