

Лабораторная работа 4

Моделирование сигнала с широотно-импульсной модуляцией

Выполнил:
студент гр. Р4106
Игнашов Иван Максимович
Вариант 8

1. Цель работы

Осуществить компьютерное моделирование сигнала с широтно-импульсной модуляцией.

Порядок работы:

1. Сформировать последовательность униполярных импульсов
2. Сформировать сигнал с широтно-импульсной модуляцией
3. Сымитировать передачу модулированного сигнала по каналу связи, где происходит его затухание и искажение помехами
4. Произвести фильтрацию и восстановление сигнала:
 - фильтрацию верхних частот сигнала
 - усиление напряжения сигнала на k - коэффициент полезного затухания сигнала
 - фильтрацию нижних частот сигнала
5. Преобразовать сигнал обратно в последовательность униполярных прямоугольных импульсов

2. Отправка ШИМ-сигнала

2.1 Моделирующая функция

Моделирующая функция:

$$s_t = \frac{A \cdot \tau(t)}{T} + \sum_{k=1}^{nk} \left(\frac{2A}{\pi k} * \sin\left(\frac{\pi k \tau(t)}{T}\right) * \cos\left(\frac{2\pi k t}{T}\right) \right), \text{ где}$$

A - амплитуда колебаний (A = 10)

T - период импульсов (T = 63)

$\tau(t)$ - функция изменения длительности импульсов от времени ($\tau(t) = \frac{t}{16} + 25$)

```
6
7
8 % Параметры Вариант 8
9 function tau = imp_duration(t) % Функция длительностей импульсов
10     tau_0 = 25; % Начальная длительность импульса
11     tau = t / 16 + tau_0;
12
13
14 T = 63; % Период импульса
15 kp = 8; % Количество периодов
16
17 A = 10; % Амплитуда колебаний
18 nk = 32; % число гармонирующих составляющих
19
20 times = 0:(kp*T);
21 signal = zeros(1, T*kp + 1);
22 for time = times
23     sum = 0;
24     for k = 1:nk
25         sum = sum + ((2*A) / (pi*k)) ...
26                 * sin(pi*k*imp_duration(time) / T) ...
27                 * cos(2*pi*k*time / T);
28     end
29     signal(1, time + 1) = (A*imp_duration(time)) / T + sum;
30 end
```

Рис. 1: Код генерации моделирующей функции

2.2 Генерация ШИМ-колебания

Высокочастотные колебания:

$$U_{nt} = A_H \cos(\omega_H t), \text{ где}$$

A_H - амплитуда ВЧ колебания ($A_H = 1$)

ω_H - угловая частота ($\omega_H = \frac{5 \cdot 2\pi}{T}$)

Итоговое колебание:

$$U_{rez_t} = s_t * U_{n_t}$$

```

30
31 % Задание ВЧ колебаний
32 A_n = 1; % амплитуда ВЧ
33 omega_n = 5 * 2*pi / T; % угловая частота ВЧ
34 Un = A_n * cos(omega_n * times); % ВЧ колебание
35
36 % Модулированный сигнал
37 Urez = signal .* Un;
38

```

Рис. 2: Код генерации выходного сигнала

2.3 Этапы отправки сигнала

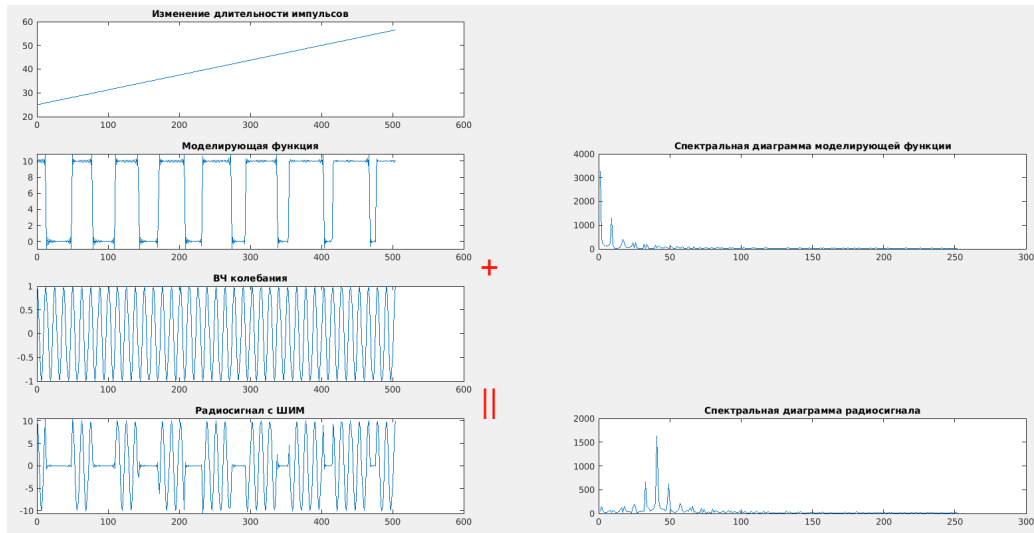


Рис. 3: Генерация сигнала

3. Приём ШИМ-сигнала

3.1 Обфускация сигнала

При передаче сигнала по каналу связи происходит некоторое его затухание и искажение помехами:

$$Ukr_t = (k * Urez_t) + Upom_t, \text{ где}$$

k - коэффициент затухания полезного сигнала

$Upom_t$ - мгновенные значения помех

```
59 % ЭТАП 2 - Обфускация сигнала
60 function Upom = interference(scale, len) % Функция помех
61     Upom = zeros(1, len);
62     for i = 1:len
63         Upom(1, i) = (rand() - 0.5) * scale;
64     end
65 end
66
67 fade_k = 0.75; % коэффициент затухания полезного сигнала
68 Upom = interference(max(Urez)*0.5, kp*T + 1);
69
70 % Моделирование сигнала на приёмнике
71
72 Ukr = (fade_k * Urez) + Upom; % входной сигнал из канала
73 input_frecs = fft(Ukr);
74
```

Рис. 4: Код обфускации сигнала

3.2 Фильтрация сигнала от помех

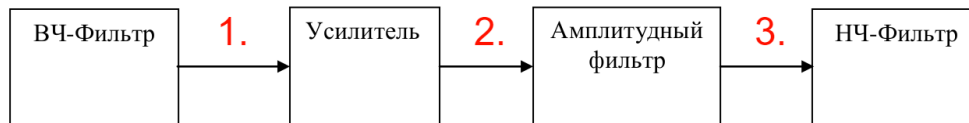


Рис. 5: Схема фильтрации приемного устройства

Для фильтрации принимаемого сигнала последовательно используются:

- ВЧ-фильтр - **частотно**-избирательная фильтрация, в качестве оператора которой используется функция Хевисайда (1 если $x \geq \alpha$ 0 иначе)
- Усилитель - во **временной** области выражение напряжения имеет вид $U_{rez2_t} = \frac{1}{k} U k 2_t$, где k - коэффициент полезного затухания сигнала
- НЧ-фильтр - фильтр нижних частот, частотная характеристика фильтра определяется выражением $k_t = \frac{1}{1+(\frac{t}{f_n})^{2n}}$, где f - верхняя частота среза фильтра; n - порядок фильтра

Для перевода сигнала из временной области в частотную и обратно использовались функция быстрого преобразования Фурье и обратная к ней соответственно

```

75 % ЭТАП 3 - Фильтрация
76 function filtered_freq = filter1(freq) % частотно-избирательная фильтрация
77     alpha = 100;% параметр фильтра
78     if (abs(freq) - alpha >= 0)
79         filtered_freq = freq;
80     else
81         filtered_freq = 0;
82     end
83 end
84
85 filter1_freqs = arrayfun(@(f) filter1(f), input_freqs);
86 Uk2 = ifft(filter1_freqs);
87 Urez2 = Uk2 / fade_k; % усиление сигнала обратно
88
89 function filtered_A = filter2(A) % отсечение отрицательной части амплитуды
90     if (A > 0)
91         filtered_A = A;
92     else
93         filtered_A = 0;
94     end
95 end
96
97 Ud = arrayfun(@(A) filter2(A), Urez2);
98 Ud_freqs = fft(Ud);
99
100 function filtered_freq = filter3(freq, time) % фильтр нижних частот
101     f = 10; % верхняя частота среза фильтра
102     n = 2; % порядок фильтра
103     filtered_freq = freq / (1 + (time / (f * n)) ^ (2 * n));
104 end
105
106 Ufiltered_freqs = arrayfun(@(t) filter3(Ud_freqs(1,t), t - 1), 1:kp*T + 1);
107 Ufiltered = ifft(Ufiltered_freqs);
108

```

Рис. 6: Код этапов фильтрации сигнала

3.3 Этапы приёма сигнала

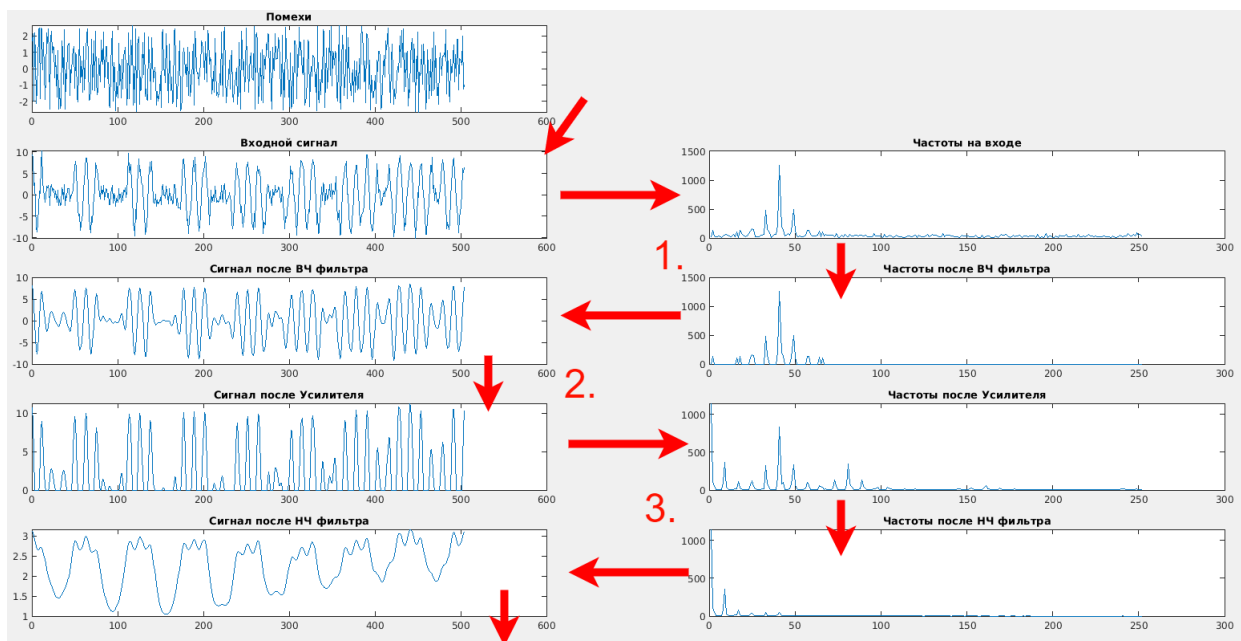


Рис. 7: Обработка сигнала

4. Преобразование в импульсы

После фильтрации была получена огибающая функция сигнала h_t , следующее выражение преобразует её в последовательность униполярных прямоугольных импульсов:

$Udet_t = if(h_t > m, 1, 0)$, где m - эмпирически подобранный параметр, зависящий от формы h_t

```
108
109 % ЭТАП 4 - Получение цифровых импульсов
110 function determ_imp = determ(A) % огибающая функция сигнала
111     m = 2; % параметр для детерминирования
112     if (A > m)
113         determ_imp = 1;
114     else
115         determ_imp = 0;
116     end
117 end
118
119 Udet = arrayfun(@determ(A), Ufiltered);
120
```

Рис. 8: Код преобразования в импульсы

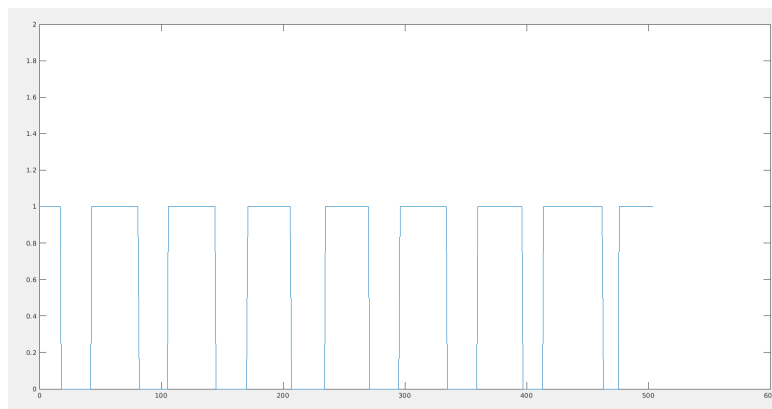


Рис. 9: График полученных импульсов

Полученные импульсы соответствуют изначально сформированной последовательности.