Федеральное государственное образовательное бюджетное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики» (ФГОБУ ВО «СибГУТИ»)

Кафедра РТС

Задание на практическое занятие

Новосибирск 2024

1 Формирование сигнала с ортогональным частотным уплотнением (мультиплексированием)

В системе с ортогональным частотным мультиплексированием сигнал формируется в виде суммы N_c модулированных комплексных колебаний (поднесущих). Частоты соседних поднесущих разделены на минимальное расстояние $\Delta f, df$ для обеспечения ортогональности между поднесущими. Свойство ортогональности колебаний необходимо для разделения колебаний при приеме сигнала OFDM.

Каждая поднесущая с номером k модулируется QAM или QPSK, т.е. переносит выбранный комплексный символ $s(k)$ из сигнальной диаграммы QAM. После суммирования колебаний они передаются параллельно в виде OFDM символа длительностью T.

Для обеспечения ортогональности между поднесущими необходимое расстояние между поднесущими (шаг сетки частот сигнала) определяется как

$$\Delta f = \frac{1}{T}.$$

Сформированный сигнал из N_c поднесущих занимает полосу частот $F = (N_c - 1)\Delta f$ и записывается в виде

$$x(t) = \frac{1}{T}\sum_{k=0}^{N_c-1} s(k) exp(j2\pi k\Delta f t)$$

```
from scipy.fftpack import fft, ifft, fftshift
import numpy as np
import matplotlib.pyplot as plt
T=1e-4 # Длительность символа
Nc=16 # Количество поднесущих
df=1/T # Частотный интервал между поднесущими
ts=T/Nc # Интервал дискретизации

k=1
t=ts*np.arange(0,Nc);
s=1/np.sqrt(T)*np.exp(1j*2*np.pi*k*df*t) # Формирование
    одной поднесущей с частотой f*df
plt.figure(1)
plt.plot(t,s.real) #реальная часть поднесущей
sc_matr = np.zeros((Nc, len(t)), dtype=complex)
sd = np.zeros((1,Nc), dtype=complex)

# Матрица из поднесуших
for k in range(Nc):
    sk_k=1/np.sqrt(T)*np.exp(1j*2*np.pi*k*df*t)
```

```
    sc_matr[k,:]=sk_k

    #sd - вектор Nc передаваемых комплексных символов
sd=np.sign(np.random.rand(1,Nc)-0.5)+1j*np.sign(np.random.
   rand(1,Nc)-0.5)
sd=sd.reshape(Nc)

xt=np.zeros((1,len(t)), dtype=complex)
# формирование суммы модулированных поднесущих
for k in range (Nc):
    sc=sc_matr[k,:]
    xt=xt+sd[k]*sc

xt=xt.reshape(Nc)
# реальная часть сформированного OFDM символа
plt.figure(2)
plt.plot(t,xt.real)
```

Для формирования сигнала в виде суммы модулированных колебаний на практике используется обратное ДПФ. Количество точек ОДПФ в частотной области (вектор на входе ОДПФ) равно количеству отсчетов символа OFDM во временой области.

Символ, сформированный суммированием ортогональных модулированных колебаний и символ, полученный при помощи ОДПФ связаны между собой с учетом нормирующего множителя

$$x(t)_{IFFT} = \frac{\sqrt{T}}{N_c} x(t).$$

Для приема сигнала OFDM необходимо разделить все поднесущие, это выполняется с использованием свойства ортогональности. Символ на k поднесущей выделяется в виде

$$s(k) = \frac{1}{\sqrt{T}} \int_0^T x(t) exp(-j2\pi k \Delta f t) dt.$$

Прямое ДПФ параллельно вычисляет набор таких корреляционных интегралов одновременно в виде набора коррелляторов.

```
 #формирование OFDM символа при помощи ОДПФ
xt2=np.fft.ifft(sd,16);

# реальная часть сформированного OFDM символа
plt.figure(3)
plt.plot(t,xt2.real)

n=3
```

```
#прием символа на n поднесущей в виде интеграла от
    произведения принятого символа на опорное колебание на n
    поднесущей
sr=ts*np.sum(xt*np.conjugate(sc_matr[n,:]));
#прием символа на Nc поднесущих при помощи ДПФ принятого
    символа
sr2=np.sqrt(T)/Nc*np.fft.fft(xt);
```

Вектор sr2 содержит Nc комплексных символов, соответствующих переданным символам на всех поднесущих.