



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Крымский федеральный университет имени В.И. Вернадского»

Физико-технический институт

Кафедра компьютерной инженерии и моделирования

Лабораторная работа № 3  
**«Количество информации при неполной достоверности  
сообщений»** по дисциплине  
«Теория информации и кодирование»

Выполнил:  
студент 3 курса  
группа ИВТ-222  
Гоголев В. Г.

Проверил:  
Филиппов Д.М.  
«\_\_\_» \_\_\_\_\_ 20\_\_ г.  
Подпись: \_\_\_\_\_

Симферополь, 2024

**Цель:** рассчитать информационные характеристики дискретных сигналов и каналов при использовании канала без помех и с помехами.

**Техническое задание:** источник информации вырабатывает информационный сигнал с  $N$  различными символами.

Вероятности появления символов на входе задаются в виде счетчика случайных чисел. Длительность каждого символа генерируется случайным образом во временном интервале  $(0 \div N]$  мкс. Источник информации подключен к каналу передачи сигналов. Канал передачи сигналов может работать как с помехами, так и без помех. При работе канала с помехами вероятность ошибки в канале задается случайным образом в интервале  $[0 \div q]$ , где  $q=1/(2 \cdot N)$ .

Необходимо разработать программное обеспечение и провести комплекс численных экспериментов по расчету пропускной способности и скорости передачи информации при использовании канала без помех и канала с помехами.

### **Ход работы:**

#### **Вариант № 4**

**Задание I.** С использованием разработанного программного обеспечения необходимо провести комплекс численных экспериментов (не менее 6), в ходе которого необходимо:

а) сгенерировать массив вероятностей появления совокупности сообщений на входе дискретного канала;

б) сгенерировать длительности каждого символа сообщения;

в) сгенерировать матрицу переходов со входа на выход в канале передачи информации с помехами с учетом технического задания, используя счетчик случайных чисел;

г) рассчитать пропускную способность и скорость передачи при использовании канала без помех;

д) рассчитать пропускную способность и скорость передачи при использовании канала с помехами.

```
#a) сгенерировать массив вероятностей появления совокупности сообщений на входе дискретного канала;

def generate_P_X(N):
    P_X = [random.random() for _ in range(N)]
    P_X = [p / sum(P_X) for p in P_X]
    return P_X

print(pd.DataFrame(generate_P_X(N)))

#б) сгенерировать длительности каждого символа сообщения;

def generate_time(N):
    message_times = [ random.uniform(0, q) for _ in range(N)]

    return message_times

print(pd.DataFrame(generate_time(N)))
```

Рисунок 1 – функции generate\_P\_X и generate\_time

Функции принимают на вход N(кол-во сообщений, по условию варианта N = 11) и возвращают список случайных значений, функция для генерирования входных сообщений так же нормирует список, что бы сумма вероятностей сообщений была равна 1, функция для задания времени сообщений создает значения в диапазоне от 0 до q.

Для генерации значений используется метод random и uniform класса random, которые работают по алгоритму генератора случайных чисел (псевдогенератора), принцип работы которого описан в предыдущих лабораторных работах.

```

'''#в)сгенерировать матрицу переходов со входа на выход в канале передачи информации
с помехами с учетом технического задания, используя счетчик случайных чисел;'''

def generate_matrix_P_X_Y(N, q):
    P_X_Y_noise = [[0] * N for _ in range(N)]
    for i in range(N):
        for j in range(N):
            if i == j:
                P_X_Y_noise[i][j] = 1 - q
            else:
                P_X_Y_noise[i][j] = random.uniform(0, q / (N - 1))

    for i in range(N):
        row_sum = sum(P_X_Y_noise[i])
        for j in range(N):
            P_X_Y_noise[i][j] /= row_sum

    return P_X_Y_noise

```

Рисунок 2 – функция generate\_matrix\_P\_X\_Y

Функция калькулирует матрицу переходов. Создается пустой двумерный массив нулей, далее цикл проходится по строкам и столбцам матрицы, и проверяет элемент на диагональность, если это диагональный элемент, то он считается как:  $1 - q$ , другие элементы считаются как  $q / (N - 1)$

	0	1	2	3	4	5	6	7	8	9	10
0	0.972868	0.001981	0.000692	0.002049	0.004557	0.004460	0.002323	0.003104	0.003455	0.002885	0.001627
1	0.000358	0.983166	0.001478	0.003083	0.001950	0.001833	0.000260	0.001299	0.000745	0.001527	0.004301
2	0.002671	0.001270	0.979128	0.002129	0.002784	0.002808	0.002503	0.000133	0.000445	0.001910	0.004219
3	0.003434	0.001778	0.004173	0.977825	0.000433	0.002097	0.002686	0.001832	0.002548	0.001538	0.001656
4	0.002451	0.000931	0.000228	0.003600	0.980947	0.001442	0.004231	0.002012	0.003163	0.000294	0.000701
5	0.001564	0.000259	0.002693	0.004174	0.003198	0.979545	0.000650	0.000693	0.004113	0.002973	0.000140
6	0.004251	0.003062	0.002937	0.003876	0.002501	0.003944	0.970889	0.000854	0.003060	0.002949	0.001677
7	0.002005	0.002998	0.001250	0.001758	0.002523	0.000066	0.004213	0.979461	0.002038	0.000410	0.003278
8	0.001828	0.002938	0.002895	0.003371	0.001998	0.003456	0.002531	0.003383	0.973999	0.001889	0.001712
9	0.002015	0.001814	0.002467	0.004403	0.000742	0.003043	0.001884	0.003448	0.000466	0.975689	0.004031
10	0.001274	0.000012	0.002322	0.000391	0.002579	0.000402	0.002834	0.000542	0.000953	0.000956	0.987735

Рисунок 3 – матрица переходов

```

def calucalte_data_without_noise(H_X, time):
    capacity_no_noise = math.log2(N) / sum(time) # пропускная способность
    speed_no_noise = H_X / sum(time) # скорость В бит/с

    return {'Скорость': speed_no_noise,
            'П/с': capacity_no_noise}

```

Рисунок 4 – функция расчета параметров для канала без помех

Функция принимает энтропия входных сообщений и массив с длительностью символов, далее используя формулы из методических указаний вычисляется скорость передачи данных и пропускная способность канала.

Скорость передачи для дискретного канала без помех определяется следующим выражением:

$$\bar{I}(Y) = \frac{H(Y)}{\bar{\tau}} \quad (3.1),$$

где  $H(Y) = -\sum_{i=1}^N p(y_i) \log_2 p(y_i)$  - энтропия информационного сигнала на входе в канал передачи;

$\bar{\tau} = \sum_{i=1}^N p(y_i) \cdot \tau_i$  - средняя длительность символа.

Пропускная способность дискретного канала без помех характеризуется максимальной энтропией информационного сигнала:

$$C = \frac{\log_2 N}{\bar{\tau}} \quad (3.2),$$

Рисунок 5 – формулы для расчета параметров канала без помех

```
def calculate_data_with_noise(P_X,P_X_Y_noise,time):
    H_X_Y = 0 #
    for i in range(N):
        for j in range(N):
            H_X_Y -= P_X[i] * P_X_Y_noise[i][j] * math.log2(P_X_Y_noise[i][j])

    speed_noise = (H_X - H_X_Y) / sum(time)
    capacity_noise = (math.log2(N) - H_X_Y) / sum(time)

    return {'Скорость':speed_noise,
            'П/б': capacity_noise}
```

Рисунок 6 – функция расчета параметров для канала с помехами

Функция принимает на вход входящий массив сообщений, матрицу переходов, массив с длительностью каждого сигнала.

Формулы взяты из методических указаний.

Скорость передачи информации по дискретному каналу с помехами определяется следующим соотношением

$$\bar{I}(Z, Y) = \frac{H(Y) - H(Y/Z)}{\bar{T}} \quad (3.3),$$

где  $H(Y/Z)$  - остаточная энтропия сигнала, обусловленная действием помех. Остаточная энтропия сигнала ( $H(Y/Z)$ ) определяется соотношением:

$$H(Y/Z) = - \sum_{i=1}^N \sum_{j=1}^N p(y_i, z_j) \log_2 p(y_i / z_j) \quad (3.4),$$

Пропускная способность дискретного канала с помехами

$$C = \frac{(\log_2 N - H(Y/Z))}{\bar{T}} \quad (3.7).$$

Рисунок 6 - формулы для расчета параметров канала с помехами

```
14.42 - скорость без помех(бит/с)
16.28 - пропускная способность без помех (бит/с)
-----
13.38 - скорость с помехами (бит/с)
15.24 - пропускная способность с помехами (бит/с)
```

Рисунок 7 – вывод результатов работы программы

**Задание II.** Вычислить среднюю пропускную способность и среднюю скорость передачи для канала без помех и для канала с помехами.

В результате работы было сделано 6 численных экспериментов и рассчитана средняя скорость и пропускная способность для каналов с помехами и без.Ц

```
-----
Средняя скорость за 6 тестов без помех:13.44 бит/с
Средняя П/с за 6 тестов без помех:14.56 бит/с
-----
Средняя скорость за 6 тестов с помехами:12.45 бит/с
Средняя П/с за 6 тестов с помехами:13.58 бит/с
```

Рисунок 8 – результаты по средним показателям за 6 тестов

## ЗАКЛЮЧЕНИЕ

В результате выполнения работы были получены навыки по вычислению скорости передачи информации и пропускной способности канала, как для канала с помехами, так и без.

Анализируя результаты работы, сделаны выводы, что пропускная способность и скорость передачи канала без помех выше, чем у канала с помехами.

## ПРИЛОЖЕНИЕ

```
import math
import random
import pandas as pd

N = 11
q = 1 / (2 * N)

#a)сгенерировать массив вероятностей появления совокупности сообщений на входе
дискретного канала;

def generate_P_X(N):
    P_X = [random.random() for _ in range(N)]
    P_X = [p / sum(P_X) for p in P_X]
    return P_X

#б)сгенерировать длительности каждого символа сообщения;

def generate_time(N):
    message_times = [ random.uniform(0, q) for _ in range(N)]

    return message_times

'''#в)сгенерировать матрицу переходов со входа на выход в канале передачи
информации
с помехами с учетом технического задания, используя счетчик случайных чисел;'''

def generate_matrix_P_X_Y(N, q):
    P_X_Y_noise = [[0] * N for _ in range(N)]
    for i in range(N):
        for j in range(N):
            if i == j:
                P_X_Y_noise[i][j] = 1 - q
            else:
                P_X_Y_noise[i][j] = random.uniform(0, q / (N - 1))

    for i in range(N):
        row_sum = sum(P_X_Y_noise[i])
        for j in range(N):
            P_X_Y_noise[i][j] /= row_sum

    return P_X_Y_noise

def create_var(N,q):
```



```

P_X_Y = generate_matrix_P_X_Y(N,q) # матрица переходов
P_X = generate_P_X(N) # матрица входных сообщений
time = generate_time(N) # массив с длительностями сообщений в миллисекундах
H_X = -sum([p * math.log2(p) for p in P_X]) # энтропия на входе в канал

return P_X_Y,P_X,time,H_X

P_X_Y,P_X,time,H_X = create_var(N,q)

#г) рассчитать пропускную способность и скорость передачи при использовании канала
без помех;

def calucalte_data_without_noise(H_X, time):
    capacity_no_noise = math.log2(N)/ sum(time) # пропускная способность
    speed_no_noise = H_X/ sum(time) # скорость В бит/с

    return {'Скорость': speed_no_noise,
            'П/с': capacity_no_noise}

#д) д) рассчитать пропускную способность и скорость передачи при использовании
канала с помехами.
def calculate_data_with_noise(P_X,P_X_Y_noise,time):
    H_X_Y = 0 #
    for i in range(N):
        for j in range(N):
            H_X_Y -= P_X[i] * P_X_Y_noise[i][j] * math.log2(P_X_Y_noise[i][j])

    speed_noise = (H_X - H_X_Y) / sum(time)
    capacity_noise = (math.log2(N) - H_X_Y) / sum(time)

    return {'Скорость':speed_noise,
            'П/с': capacity_noise}

no_noise = calucalte_data_without_noise(H_X,time)
noise = calculate_data_with_noise(P_X, P_X_Y, time)

print(round(no_noise['Скорость'],2),"- скорость без помех(бит/с)")
print(round(no_noise['П/с'],2),"- пропускная способность без помех (бит/с)")
print("-----")
print(round(noise['Скорость'],2),"- скорость с помехами (бит/с)")
print(round(noise['П/с'],2),"- пропускная способность с помехами (бит/с)\n")
print("-----")

avg_speed_noise = 0
avg_speed_no_noise = 0
avg_capacity_noise = 0
avg_capacity_no_noise = 0
k = 6

for i in range(k):

```

```

P_X_Y,P_X,time,H_x = create_var(N,q)
no_noise = calucalte_data_without_noise(H_X,time)
noise = calculate_data_with_noise(P_X, P_X_Y, time)
avg_speed_no_noise += no_noise['Скорость']
avg_capacity_no_noise += no_noise['П/с']
avg_speed_noise += noise['Скорость']
avg_capacity_noise += noise['П/с']

print(f"Средняя скорость за {k} тестов без помех:{(avg_speed_no_noise/k):.2f}
бит/с")
print(f"Средняя П/с за {k} тестов без помех:{(avg_capacity_no_noise/k):.2f}
бит/с")
print('-----')
print(f"Средняя скорость за {k} тестов с помехами:{(avg_speed_noise/k):.2f}
бит/с")
print(f"Средняя П/с за {k} тестов с помехами:{(avg_capacity_noise/k):.2f} бит/с")

```

## Приложение 1 – листинг программного кода