ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

Федеральное государственное образовательное бюджетное учреждение

высшего профессионального образования

«Санкт-Петербургский государственный университет телекоммуникаций

им. проф. М.А. Бонч-Бруевича»

Факультет Инфокоммуникационных сетей и систем

Кафедра Программной инженерии и вычислительной техники

ОТЧЕТ

по дисциплине:

**«Программирование»**

Курсовая работа

**Анализ сигнала на выходе электрической цепи**

Выполнил студент:

Андреев Артур Алексеевич, ИКПИ-32

Дата выполнения:

Проверил:

Воронцова Ирина Олеговна \_\_\_\_\_\_\_\_\_\_

*(подпись)*

Санкт-Петербург

2024

Оглавление

[Введение 3](#_Toc167796020)

[Контрольный расчёт 4](#_Toc167796021)

[Таблица индентификаторов 4](#_Toc167796022)

[Блок-схемы файлов 5](#_Toc167796023)

[Текст программы 7](#_Toc167796024)

[Графики 12](#_Toc167796025)

[Пример вывода 13](#_Toc167796026)

[Используемая литература 14](#_Toc167796027)

# Введение

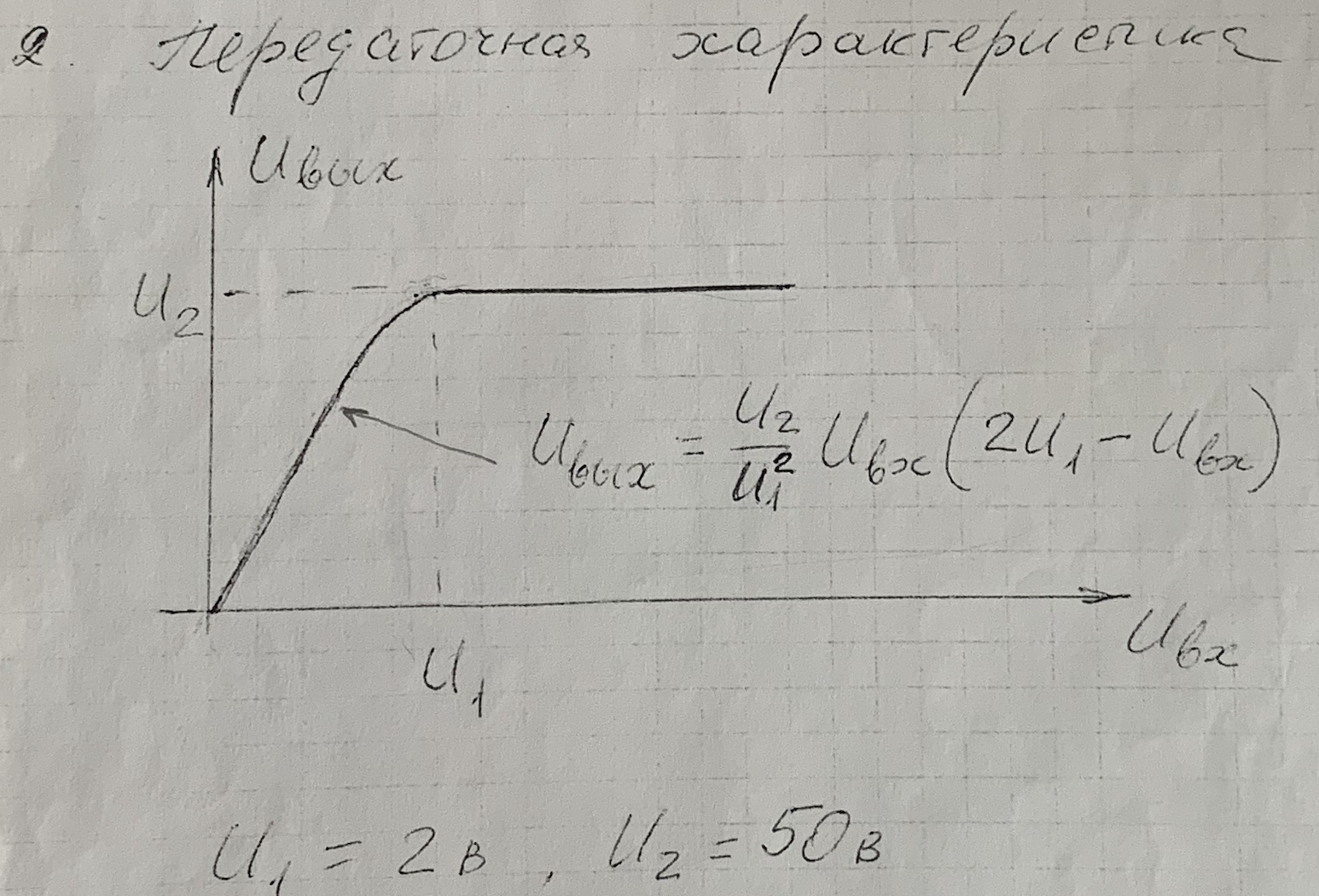
Работа посвящена решению задач машинного анализа электрических цепей. В курсовой работе необходимо для заданной электрической цепи по известному входному сигналу определить выходной сигнал для N равностоящих моментов времени, а затем определить некоторые его характеристики с погрешностью не более 1%. Варианты параметров входного сигнала (код А) и передаточной характеристики (код Б) электрической цепи приведены в приложении. Номер варианта определяется преподавателем индивидуально для каждого студента.

**Постановка задачи**

**A piece of paper with writing on it

Description automatically generated**

**Передаточная характеристика**

****

# Контрольный расчёт

|  |  |
| --- | --- |
| Напряжение входящего сигнала | Напряжение выходящего сигнала |
| Uvh(0)=0.0  Uvh(1.0·10^-6)=1.0  Uvh(2.0·10^-6)=2.0  Uvh(3.0·10^-6)=3.0  Uvh(4.0·10^-6)=4.0  Uvh(5.0·10^-6)=5.0  Uvh(6.0·10^-6)=4.0  Uvh(7.0·10^-6)=3.0  Uvh(8.0·10^-6)=2.0  Uvh(9.0·10^-6)=1.0  Uvh(1.0·10^-5)=-3.3881·10^-15 | Uvix(0)=0.0  Uvix(1.0·10^-6)=37.5  Uvix(2.0·10^-6)=50  Uvix(3.0·10^-6)=50  Uvix(4.0·10^-6)=50  Uvix(5.0·10^-6)=50  Uvix(6.0·10^-6)=50  Uvix(7.0·10^-6)=50  Uvix(8.0·10^-6)=50.0  Uvix(9.0·10^-6)=37.5  Uvix(1.0·10^-5)=-1.6941·10^-13 |

# Таблица идентификаторов

|  |  |  |  |
| --- | --- | --- | --- |
| № | Идентификатор | Тип | Описание |
| 1 | calculate\_Uvh | Функция | Вычисляет Uvh на основе времени t |
| 2 | calculate\_Uvix | Функция | Вычисляет Uvix на основе Uvh |
| 3 | initialize\_time\_array | Функция | Инициализирует массив времени t |
| 4 | calculate\_Uvx | Функция | Вычисляет Uvx для каждого времени t |
| 5 | calculate\_Uvix\_array | Функция | Вычисляет Uvix для каждого Uvx |
| 6 | find\_Umax\_Umin | Функция | Находит максимальное и минимальное значения |
| 7 | calculate\_pulse\_duration | Функция | Вычисляет длительность импульса |
| 8 | print\_results | Функция | Выводит результаты |
| 9 | write\_results\_to\_files | Функция | Записывает результаты в файлы |
| 10 | read\_and\_print\_file | Функция | Читает и выводит содержимое файла |
| 11 | calculate\_with\_precision | Функция | Вычисляет параметры с точностью |
| 12 | display\_menu | Функция | Отображает меню |
| 13 | main | Функция | Главная функция |
| 14 | t | Переменная | Массив времени |
| 15 | tn | Переменная | Начальное время |
| 16 | dt | Переменная | Шаг времени |
| 17 | n | Переменная | Количество точек |
| 18 | U | Переменная | Начальное напряжение |
| 19 | T | Переменная | Период |
| 20 | U1 | Переменная | Пороговое напряжение 1 |
| 21 | U2 | Переменная | Пороговое напряжение 2 |
| 22 | Uvx | Переменная | Массив Uvx |
| 23 | Uvix | Переменная | Массив Uvix |
| 24 | Umax | Переменная | Максимальное напряжение |
| 25 | Umin | Переменная | Минимальное напряжение |
| 26 | Uimp | Переменная | Импульсное напряжение |
| 27 | pulse\_duration | Переменная | Длительность импульса |
| 28 | par | Переменная | Параметр для расчета точности |
| 29 | par1 | Переменная | Временный параметр |
| 30 | p | Переменная | Ошибка точности |
| 31 | tk | Переменная | Конечное время |
| 32 | eps | Переменная | Значение точности |
| 33 | choice | Переменная | Выбор в меню |
| 34 | sub\_choice | Переменная | Выбор в подменю |
| 35 | calc\_choice | Переменная | Выбор для расчета |

# Блок-схемы файлов

main.c

A diagram of a diagram

Description automatically generated

utils.c

A diagram of a company

Description automatically generated

file\_io.c

A diagram of a diagram

Description automatically generated

calculations.c

A diagram of a flowchart

Description automatically generated

# Текст программы

**main.c**

#include <stdio.h>

#include "calculations.h"

#include "file\_io.h"

#include "utils.h"

int main() {

read\_and\_print\_file("intro.txt");

display\_menu();

return 0;

}

**falculations.c**

#include <stdio.h>

#include <math.h>

#include "calculations.h"

#define N 1000

float calculate\_Uvh(float t, float U, float T) {

return (t <= T / 2) ? (2 \* U \* t / T) : (2 \* U \* (T - t) / T);

}

float calculate\_Uvix(float Uvh, float U1, float U2) {

return (Uvh < U1) ? ((U2 / (U1 \* U1)) \* Uvh \* (2 \* U1 - Uvh)) : U2;

}

void initialize\_time\_array(float t[], float tn, float dt, int n) {

for (int i = 0; i < n; i++) {

t[i] = tn + i \* dt;

}

}

void calculate\_Uvx(float t[], float Uvx[], float U, float T, int n) {

for (int i = 0; i < n; i++) {

Uvx[i] = calculate\_Uvh(t[i], U, T);

}

}

void calculate\_Uvix\_array(float Uvx[], float Uvix[], float U1, float U2, int n) {

for (int i = 0; i < n; i++) {

Uvix[i] = calculate\_Uvix(Uvx[i], U1, U2);

}

}

void find\_Umax\_Umin(float values[], int n, float \*Umax, float \*Umin) {

\*Umax = \*Umin = values[0];

for (int i = 1; i < n; i++) {

if (values[i] > \*Umax) \*Umax = values[i];

if (values[i] < \*Umin) \*Umin = values[i];

}

}

float calculate\_pulse\_duration(float values[], float threshold, float dt, int n) {

float pulse\_duration = 0;

for (int i = 0; i < n; i++) {

if (values[i] >= threshold) {

pulse\_duration += dt;

}

}

return pulse\_duration;

}

void calculate\_with\_precision(float t[], float Uvx[], float Uvix[], float tn, float tk, float U, float T, float U1, float U2, float eps, int use\_output) {

float dt, Umax, Umin, Uimp, pulse\_duration;

float par = 1e10, par1, p = 1;

int n = 2;

while (p > 0.01) {

dt = (tk - tn) / (n - 1);

initialize\_time\_array(t, tn, dt, n);

calculate\_Uvx(t, Uvx, U, T, n);

calculate\_Uvix\_array(Uvx, Uvix, U1, U2, n);

float \*values = use\_output ? Uvix : Uvx;

find\_Umax\_Umin(values, n, &Umax, &Umin);

Uimp = Umin + 0.5 \* (Umax - Umin);

pulse\_duration = calculate\_pulse\_duration(values, Uimp, dt, n);

par1 = pulse\_duration;

p = fabs(par - par1) / par1;

printf("n=%3d параметр=%9.7f погрешность=%.4f\n", n, par1, p);

par = par1;

n = n \* 2;

}

}

void print\_results(float t[], float Uvx[], float Uvix[], int n) {

printf(" № t Uvx Uvix\n");

for (int i = 0; i < n; i++) {

printf("%3d %.8f %.3f %.3f\n", i, t[i], Uvx[i], Uvix[i]);

}

}

**falculations.h**

#ifndef CALCULATIONS\_H

#define CALCULATIONS\_H

float calculate\_Uvh(float t, float U, float T);

float calculate\_Uvix(float Uvh, float U1, float U2);

void initialize\_time\_array(float t[], float tn, float dt, int n);

void calculate\_Uvx(float t[], float Uvx[], float U, float T, int n);

void calculate\_Uvix\_array(float Uvx[], float Uvix[], float U1, float U2, int n);

void find\_Umax\_Umin(float values[], int n, float \*Umax, float \*Umin);

float calculate\_pulse\_duration(float values[], float threshold, float dt, int n);

void calculate\_with\_precision(float t[], float Uvx[], float Uvix[], float tn, float tk, float U, float T, float U1, float U2, float eps, int use\_output);

void print\_results(float t[], float Uvx[], float Uvix[], int n);

#endif // CALCULATIONS\_H

**file\_io.c**

#include <stdio.h>

#include "file\_io.h"

void write\_results\_to\_files(float t[], float Uvx[], float Uvix[], int n) {

FILE \*file\_t = fopen("massiv\_t.txt", "w");

FILE \*file\_uvx = fopen("massiv\_uvx.txt", "w");

FILE \*file\_uvix = fopen("massiv\_uvix.txt", "w");

if (!file\_t || !file\_uvx || !file\_uvix) {

printf("Ошибка при открытии файлов!\n");

if (file\_t) fclose(file\_t);

if (file\_uvx) fclose(file\_uvx);

if (file\_uvix) fclose(file\_uvix);

return;

}

for (int i = 0; i < n; i++) {

fprintf(file\_t, "%.6f\n", t[i]);

fprintf(file\_uvx, "%.2f\n", Uvx[i]);

fprintf(file\_uvix, "%.2f\n", Uvix[i]);

}

fclose(file\_t);

fclose(file\_uvx);

fclose(file\_uvix);

printf("Данные записаны в файлы: massiv\_t.txt, massiv\_uvx.txt, massiv\_uvix.txt\n");

}

void read\_and\_print\_file(const char \*filename) {

FILE \*file = fopen(filename, "r");

if (!file) {

printf("Не удалось открыть файл %s для чтения.\n", filename);

return;

}

char line[256];

while (fgets(line, sizeof(line), file)) {

printf("%s", line);

}

fclose(file);

}

**file\_io.h**

#ifndef FILE\_IO\_H

#define FILE\_IO\_H

void write\_results\_to\_files(float t[], float Uvx[], float Uvix[], int n);

void read\_and\_print\_file(const char \*filename);

#endif // FILE\_IO\_H

**utils.c**

#include <stdio.h>

#include "calculations.h"

#include "file\_io.h"

#define N 1000

void display\_menu() {

int choice, n = 11;

float t[N], Uvx[N], Uvix[N];

float tn = 0, T = 0.00001, tk = T, dt, U = 5, U1 = 2, U2 = 50;

float eps = 0.01;

while (1) {

printf("\nМеню:\n");

printf("1 - Выполнить контрольный расчет для n точек\n");

printf("2 - Рассчитать параметры с заданной точностью\n");

printf("3 - Записать данные в файл\n");

printf("4 - Рассчитать параметры с погрешностью\n");

printf("0 - Выход\n");

printf("Выберите опцию: ");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Введите количество точек для контрольного расчета: ");

scanf("%d", &n);

dt = (tk - tn) / (n - 1);

initialize\_time\_array(t, tn, dt, n);

calculate\_Uvx(t, Uvx, U, T, n);

calculate\_Uvix\_array(Uvx, Uvix, U1, U2, n);

print\_results(t, Uvx, Uvix, n);

break;

case 2:

printf("Введите точность (количество точек): ");

scanf("%d", &n);

dt = (tk - tn) / (n - 1);

initialize\_time\_array(t, tn, dt, n);

calculate\_Uvx(t, Uvx, U, T, n);

calculate\_Uvix\_array(Uvx, Uvix, U1, U2, n);

printf("Выберите расчет для входа (1) или выхода (2): ");

int sub\_choice;

scanf("%d", &sub\_choice);

float Umax, Umin, Uimp, pulse\_duration;

if (sub\_choice == 1) {

find\_Umax\_Umin(Uvx, n, &Umax, &Umin);

Uimp = Umin + 0.5 \* (Umax - Umin);

pulse\_duration = calculate\_pulse\_duration(Uvx, Uimp, dt, n);

printf("Для входа: Umax: %f, Umin: %f, Uimp: %f, Длительность импульса: %f\n", Umax, Umin, Uimp, pulse\_duration);

} else if (sub\_choice == 2) {

find\_Umax\_Umin(Uvix, n, &Umax, &Umin);

Uimp = Umin + 0.5 \* (Umax - Umin);

pulse\_duration = calculate\_pulse\_duration(Uvix, Uimp, dt, n);

printf("Для выхода: Umax: %f, Umin: %f, Uimp: %f, Длительность импульса: %f\n", Umax, Umin, Uimp, pulse\_duration);

} else {

printf("Неправильный выбор.\n");

}

break;

case 3:

write\_results\_to\_files(t, Uvx, Uvix, n);

break;

case 4:

printf("Выберите расчет для входа (1) или выхода (2): ");

int calc\_choice;

scanf("%d", &calc\_choice);

if (calc\_choice == 1 || calc\_choice == 2) {

calculate\_with\_precision(t, Uvx, Uvix, tn, tk, U, T, U1, U2, eps, calc\_choice == 2);

} else {

printf("Неправильный выбор.\n");

}

break;

case 0:

return;

default:

printf("Неправильный выбор. Попробуйте снова.\n");

break;

}

}

}

**utils.h**

#ifndef UTILS\_H

#define UTILS\_H

void display\_menu();

#endif // UTILS\_H

# Графики

График контрольного расчёта для 11 точек

Напряжение входа

A graph with a blue line

Description automatically generated

Напряжение выхода A graph with a line drawn on it

Description automatically generated

# Пример вывода

1.Таблица для 11 точек

**A screen shot of a computer code

Description automatically generated**

2.Параметр Входа

A black background with white text

Description automatically generated

3.Параметр Выхода

A black background with white text

Description automatically generated

# 

# Используемая литература

1. Л. Н. Бережной, И. О. Воронцова, Д. В. Окунева «Информатика. Анализ сигнала на выходе электрической цепи. Методические указания к курсовой работе», СПб 2012