**ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

(СПбГУТ)

ФАКУЛЬТЕТ ИНФОКОММУНИКАЦИОННЫХ СЕТЕЙ И СИСТЕМ **(ИКСС)**

КАФЕДРА ПРОГРАММНОЙ ИНЖЕНЕРИИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ **(ПИиВТ)**

Дисциплина: «Программирование»

**Курсовая работа**

**Тема**: «Решение задач машинного анализа электрических цепей»

Вариант 16 (А-16, Б-7, В-3)

Выполнил: Студент 1 курса,

группы ИКПИ-95

Новиков Семен

Принял: Доцент, к.т.н.

Футахи Абдо Ахмед Хасан

2020

**Оглавление**

[Задача: 3](#_Toc40564491)

[Разработка алгоритма: 4](#_Toc40564492)

[Таблица идентификаторов: 5](#_Toc40564493)

[Контрольные расчеты и графики: 6](#_Toc40564494)

[Программа на языке C++: 7](#_Toc40564495)

[Результат выполнения программы: 11](#_Toc40564496)

Задача:

Электрическая цепь задается передаточной характеристикой, которая описывает зависимость напряжения на выходе цепи Uвых (выходного сигнала) от напряжения на входе Uвх (входного сигнала). Напряжение на входе цепи Uвх(t) и передаточная характеристика могут задаваться либо в виде формул, т.е. аналитически, либо графически. Конкретный вид формул и графиков определяются кодами варианта. Код варианта выдает преподаватель.

Результатом курсовой работы являются программа на языке С++ с пояснительной запиской. Для заданного варианта электрической цепи и входного сигнала программа должна позволять:

а) рассчитать значения входного сигнала,

б) рассчитать значения выходного сигнала,

в) рассчитать с заданной точностью значение характеристики W выходного сигнала,

г) построить графики входного и выходного сигналов.

**А16:**

|  |  |
| --- | --- |
| Входной сигнал | Рабочий набор |
|  | *a* = 20В/с; *b* = 0,5В/с;  *c* = 17В/с;  *t*нач = 5с; *t*1 = 10с;  *t*2 = 15с;  *t*3 = 45с;  *t*кон = 50с; |

**Б7:**

|  |  |
| --- | --- |
| Передаточная характеристика | Рабочий набор |
|  | *a* = 5В;  *b* = 0,05(1/В);  *U*вх1 = 10В |

**В3:**

|  |  |
| --- | --- |
| Характеристика | Рабочий набор |
| Разность между минимальным и максимальным значениями выходного сигнала |  |

Разработка алгоритма:

Программа состоит из одного файла: main.cpp.

Программа просит пользователя ввести количество интервалов времени. Происходит проверка корректности ввода, если была найдена ошибка, то программа просит ввести заново.

Далее в функциях происходит расчет входных сигналов, выходных сигналов, поиск максимального выходного сигнала, расчет заданного уровня.

В алгоритме функции расчета входных сигналов была использована вложенная условная конструкция if - else. Изначально вычисляется шаг расчета времени. В цикле это значение прибавляется к текущему значению времени. Также в цикле заполняется массив времени.

В алгоритме функции поиска максимального выходного сигнала в цикле было использовано условие if. Если выходной сигнал больше максимального, то максимальному присваивается его значение.

В алгоритме функции поиска момента времени, при котором выходной сигнал достигает заданного условия, в цикле было использовано условие if. Если выходной сигнал больше или равен значению заданного условия, то моменту времени присваивается значение из массива времени.

Далее идет расчет момента времени, когда выходной сигнал достигает заданного уровня, и определение точности этого расчета.

При первом проходе, т.е. при N = Nнач, величина погрешности не вычисляется, а только запоминается вычисленное значение характеристики. Вводится проверка превышения текущего числа интервалов N некоторого максимального значения Nmax, допустим, 1000. Введение такой проверки предотвращает бесконечно долгое выполнение программы в случае слишком медленного уменьшения относительной погрешности δW. Если количество интервалов превысит максимально допустимое Nmax, то процесс вычисления прекращается и должно быть выдано сообщение “Требуемая точность не достигнута”.

После в файл передаются массивы времени, входного и выходного сигнала. Вызывается программа для построения двух графиков по данным из файла.

Таблица идентификаторов:

|  |  |  |
| --- | --- | --- |
| Обозначение в задаче | Идентификатор | Назначение |
| *t*нач | *t\_start* | Начальный момент времени наблюдения входного напряжения |
| *t*кон | *t\_end* | Конечный момент времени наблюдения выходного напряжения |
| *t* | *t – в getMax* | Текущий момент времени |
| *t1,t2,t3* | *t1,t2,t3* | Точки времени для смены формулы расчета входного сигнала |
| *U*max | *u\_max* | Максимальное выходное напряжение |
| *N*нач | *n\_custom* | Значение количества интервалов времени, задающееся пользователем |
| *N*макс | *n\_max* | Допустимое значение числа интервалов времени |
| *t*пп | *t* | Момент времени, при котором выходной сигнал достигнет заданного уровня |
| *N* | *N* | Текущее значение количества интервалов времени |
| ∆*t* | *dt* | Временной интервал между двумя соседними отсчетами входного сигнала |
| *U*вых(*t*) | results | Массив для хранения отсчетов выходного сигнала |
|  | *time* | Массив для хранения времени |
| *ε* | *eps* | Допустимая погрешность вычисления параметра |
| δ*t* | *delta* | Оценка погрешности вычисления величины *time\_moment* |
| Uвх | *input* | Результат входа из пункта А |

Контрольные расчеты и графики:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Набор контрольных значений | | | | Ручные вычисления | | | | Вычисления на ЭВМ | | | |
| *t*нач | *t*кон | *N* | *i* | | *U*вх | *U*вых | *t*пп | *i* | *U*вх | *U*вых | *t*пп |
| 5 | 50 | 12 | 0 | | 99.37 | 493.7 | 5 | 0 | 99.375 | 493.77 | 5 |
| 1 | | 98.4 | 484.4 | 1 | 98.4375 | 484.497 |
| 2 | | 97.5 | 475.31 | 2 | 97.5 | 475.312 |
| 3 | | 96.5 | 466.21 | 3 | 96.5625 | 466.216 |
| 5 | 0 | | 87.5 | 383.3 | 50 | 0 | 87.5625 | 383.36 | 50 |
| 1 | | 87.28 | 380.9 | 1 | 87.2812 | 380.901 |
| 2 | | 87 | 378.4 | 2 | 87 | 378.45 |
| 3 | | 86.7 | 376 | 3 | 86.7188 | 376.007 |
| 4 | | 86.4 | 373.5 | 4 | 86.4375 | 373.572 |
| 5 | | 86.1 | 371.1 | 5 | 86.1562 | 371.145 |
| 6 | | 85.8 | 368.7 | 6 | 85.875 | 368.726 |
| δ*t* = 0.0 | | | | | δ*t* = 0.0 | | | |

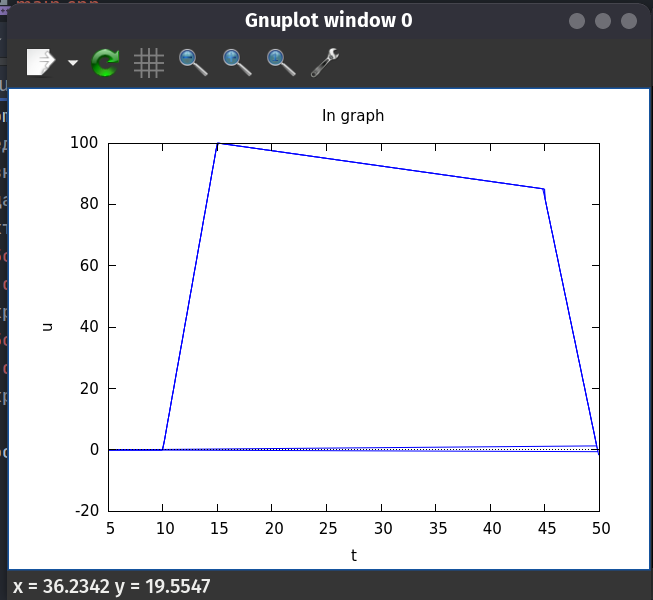


Рис.1. График сигнала

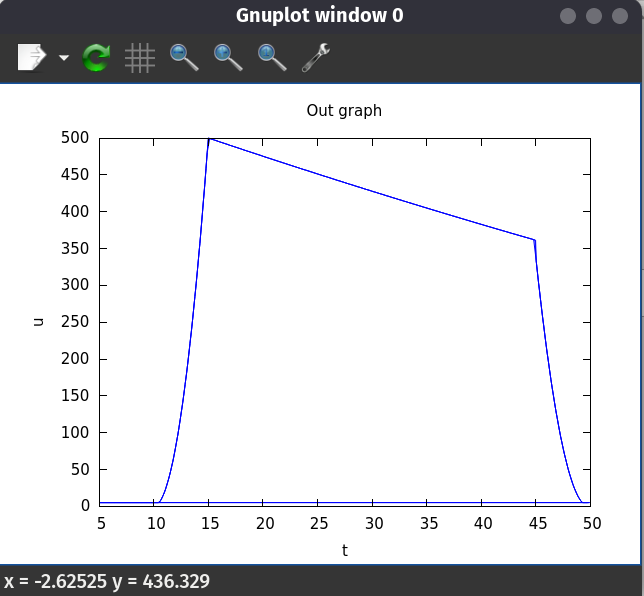


Рис.2. График выходного сигнала

Программа на языке C++:

Файл main.cpp  
#include <iostream>

#include <cmath>

#include <fstream>

#include <climits>

using namespace std;

constexpr char const \*graph\_handl =

"set zero 0.0\n"

"set term pop\n"

"unset colorbox\n"

"set xlabel \"t\"\n"

"set ylabel \"u\"\n"

"set grid front\n"

"unset grid\n"

"set size ratio 0.75\n"

"set xrange [5 : 50 ]\n"

"set zeroaxis\n"

"set datafile missing \"NaN\"\n"

"plot [5 : 50 ] '-' notitle with lines lt rgb \"#0000ff\"\n";

ofstream graph\_in;

ofstream graph\_out;

constexpr double t\_start = 5;

constexpr double t1 = 10;

constexpr double t2 = 15;

constexpr double t3 = 45;

constexpr double t\_end = 50;

// пункт A

double MathFunction(double t) {

static constexpr double a = 20;

static constexpr double b = 0.5;

static constexpr double c = 17;

if (t <= t1) {

return 0;

} else if (t1 < t && t <= t2) {

return a \* (t - t1);

} else if (t2 < t && t <= t3) {

return a \* (t2 - t1) - b \* (t - t2);

} else if (t > t3) {

return a \* (t2 - t1) - b \* (t3 - t1) - c \* (t - t3);

}

return 0;

}

double calc(uint64\_t step, double &t\_middle) {

double dt = (t\_end - t\_start) / step;

double max\_exit = -999999999999999.0;

double min\_exit = 999999999999999.0;

double max\_exit\_t = 0;

double min\_exit\_t = 0;

static constexpr double a = 5;

static constexpr double b = 0.05;

static constexpr double input1 = 10.0;

double exit = 0;

double t = t\_start;

for (uint64\_t i = 0; i < step; i++) {

double input = MathFunction(t);

graph\_in << t << " " << input << endl;

if (input <= input1) {

exit = a;

} else if (input > input1) {

exit = b \* input \* input; // bUвх^2

}

graph\_out << t << " " << exit << endl;

if (exit > max\_exit) {

max\_exit = exit;

max\_exit\_t = t;

}

if (exit < min\_exit) {

min\_exit = exit;

min\_exit\_t = t;

}

t\_middle = max\_exit\_t - min\_exit\_t;

t += dt;

}

// Разность между минимальным и

// максимальным значениями выходного сигнала

return max\_exit - min\_exit;

}

void draw(const string &path\_graph) {

string param = "gnuplot -persist " + path\_graph;

if (system((param).c\_str()) == 0) {

cout << "Открытие графика..." << endl;

} else {

cout << "Ошибка! Не удалось открыть график!" << endl;

}

}

int main() {

uint64\_t n\_custom = 0; // колличество интервалов времени

cout << "Введите колличество интервалов времени: ";

cin.exceptions(istream::failbit | istream::badbit);

input\_reguest:

try {

cin >> n\_custom;

if (n\_custom <= 1)

throw n\_custom;

}

catch (...) {

cout << "Неверный формат ввода. Попробуйте еще раз: ";

cin.clear();

cin.ignore(INT\_MAX, '\n');

goto input\_reguest;

}

string path\_graph\_in = "gnuplot\_graph\_in.txt"; // путь до графика

string path\_graph\_out = "gnuplot\_graph\_out.txt"; // путь до графика

graph\_in.open(path\_graph\_in);

graph\_out.open(path\_graph\_out);

if (!graph\_in.is\_open())

cout << "Ошибка при создании file graph in: " << path\_graph\_in << endl;

if (!graph\_out.is\_open())

cout << "Ошибка при создании file graph out: " << path\_graph\_out << endl;

graph\_in << "set title \"In graph\"\n";

graph\_in << graph\_handl;

graph\_out << "set title \"Out graph\"\n";

graph\_out << graph\_handl;

uint64\_t N = n\_custom;

constexpr double eps = 0.01;

double delta = eps + 1; // Задание начального значения погрешности

constexpr uint64\_t n\_max = 1000;

double t0;

double t;

double u\_max;

while (delta >= eps && N <= n\_max) {

u\_max = calc(N, t);

if (N != n\_custom) {

delta = abs((t0 - t) / t);

}

t0 = t;

N \*= 2;

}

if (delta > n\_max) {

cout << "Заданная точность не достигнута" << endl;

cout << "Фактическая погрешнoсть = " << delta << endl;

} else {

N = N / 2;

cout << "Разность между минимальным и максимальным значениями выходного сигнала " << u\_max << endl;

cout << "Заданная точность достигнута при N= " << N << endl;

cout << "Фактическая погрешность " << delta << endl;

}

graph\_in.close();

graph\_out.close();

draw(path\_graph\_in);

draw(path\_graph\_out);

return 0;

}

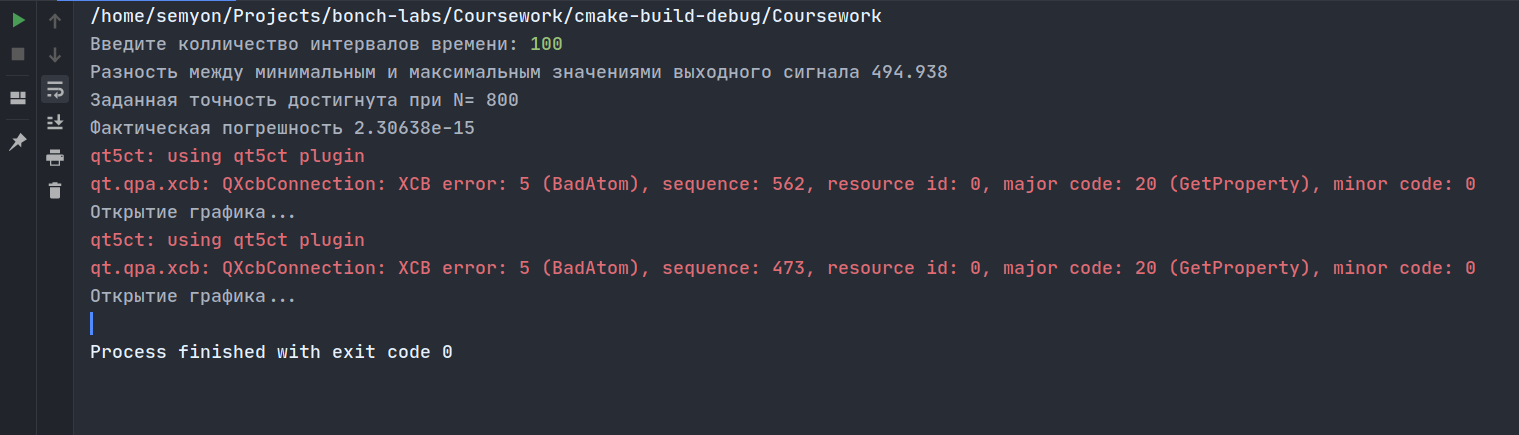
Результат выполнения программы:  


Рис 3. Результат работы программы

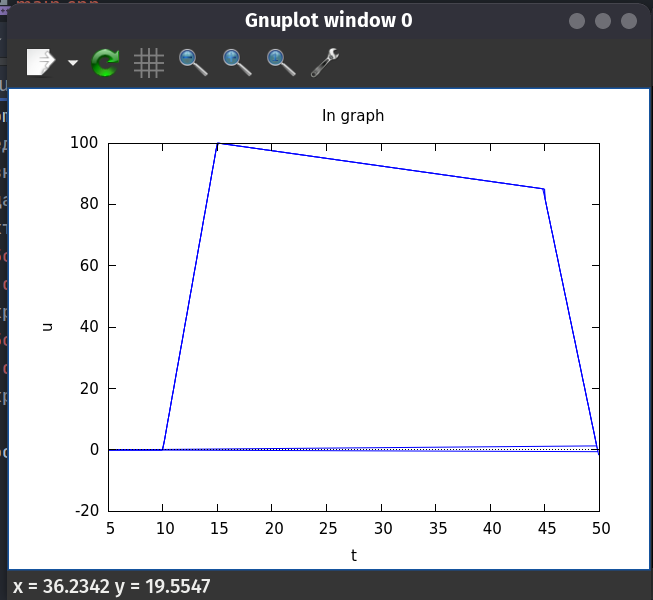


Рис 4. График выходного сигнала, построенный в результате работы программы

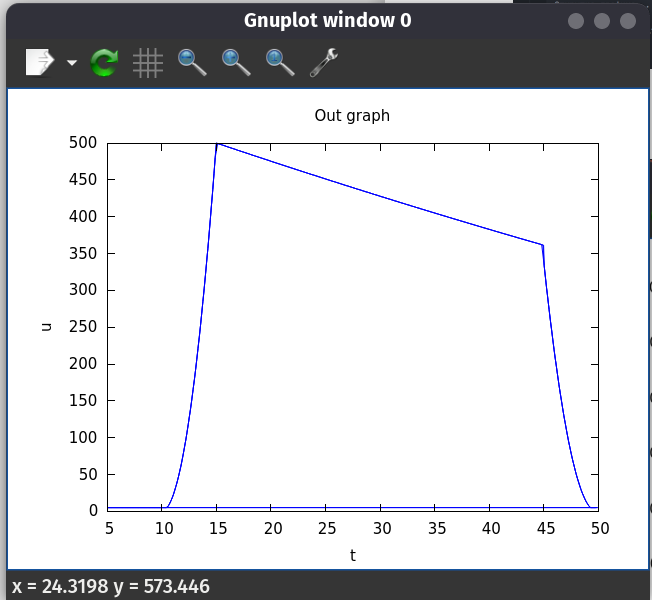


Рис 5. График выходного сигнала, построенный в результате работы программы