Министерство образования и науки РФ
Федеральное государственное бюджетное образовательное учреждение
Новосибирский государственный технический университет
Кафедра Автоматизированных систем управления

Расчетно-графическая работа

По дисциплине: Основы теории управления

Вариант 10

Группа: АВТ-412

Студент: Лазаревич М.М.

Преподаватель: Достовалов Д.Н.

Новосибирск, 2016

Задание:

Даны уравнения, описывающие процессы в системе автоматического управления:

$$\begin{cases} \dot{x}_1 = k_4(x_2 - Z) \\ \dot{x}_1 = \frac{k_2}{T}(x_3 - k_3 x_1) - \frac{1}{T}x_2 \\ \dot{x}_3 = \frac{k_1}{T_1}x_{\text{РБ}} - \frac{1}{T_1}x_3 \\ \dot{x}_{\text{РБ}} = \frac{k_{\text{РБ}}}{T_0}e - \frac{1}{T_0}x_{\text{РБ}} \\ e = V - k_{\text{ОС}}x_1 \end{cases} \qquad (1)$$

где $Y = x_1$ – выходная (регулируемая) координата системы;

V – входное воздействие;

Z – возмущающее воздействие;

x_1, x_2, x_3 – переменные состояния системы;

$k_{РБ}$, k_{OC} – передаточный коэффициенты решающего блока и ветви обратной связи системы;

k_1, k_2, k_3, k_4 – передаточные коэффициенты;

T_0, T_1, T – постоянные времени, рассчитываемые в секундах.

Первые два уравнения в (1) описывают объект управления (рис. 1). Третье уравнение в (1) соответствует усилителю мощности. Четвертое уравнение описывает решающий блок. Пятое уравнение – уравнение замыкания (обратной связи) системы.

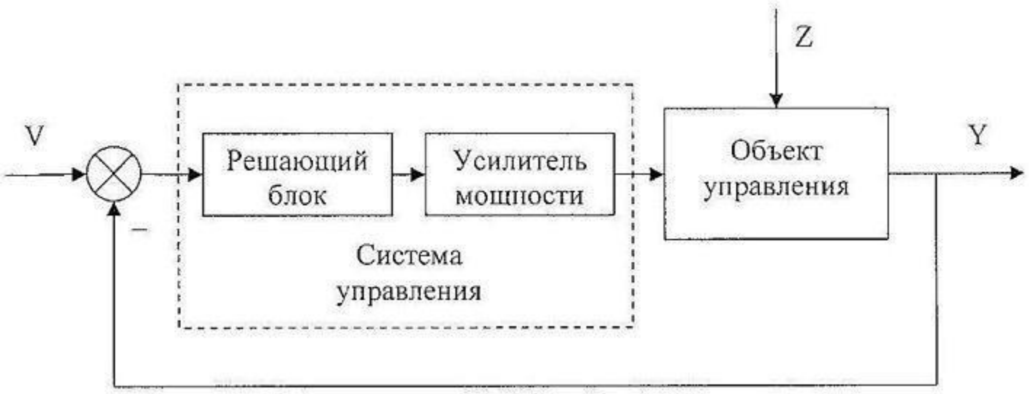

Рис.1. Обобщенная структура системы

Таблица 1 – Значения параметров звеньев исходной системы

Номер варианта	k_1	T_1	k_2	k_3	k_4	T	T_0	k_{OC}	$k_{\text{РБ}}$	Z_0	Δx_1^c
10	50	0.55	0.4	2.5	1	0.09	20	1	1	20	0.5

Этапы выполнения работы

1. Построение структурной схемы исходной системы по заданному математическому описанию.

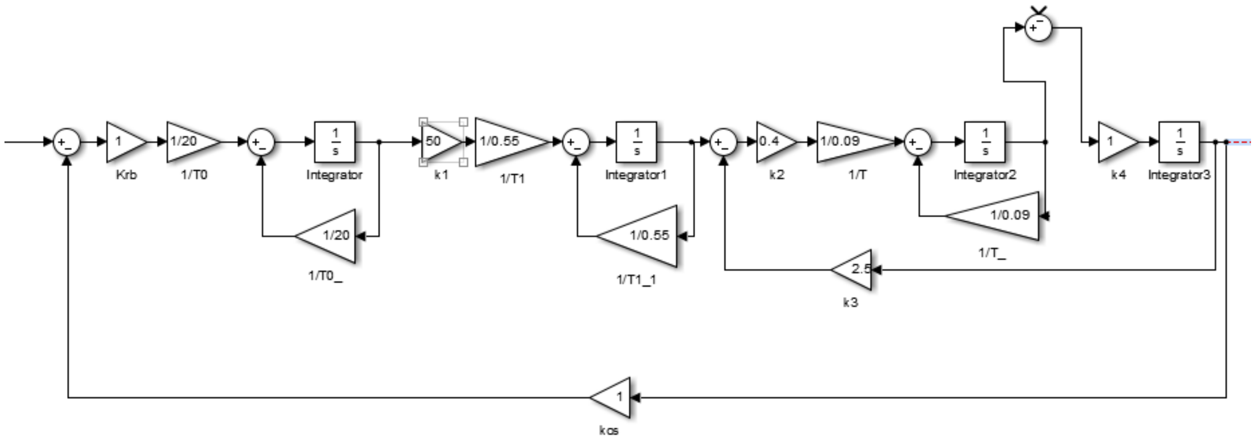

Рис.2. Структурная схема системы с заданными параметрами

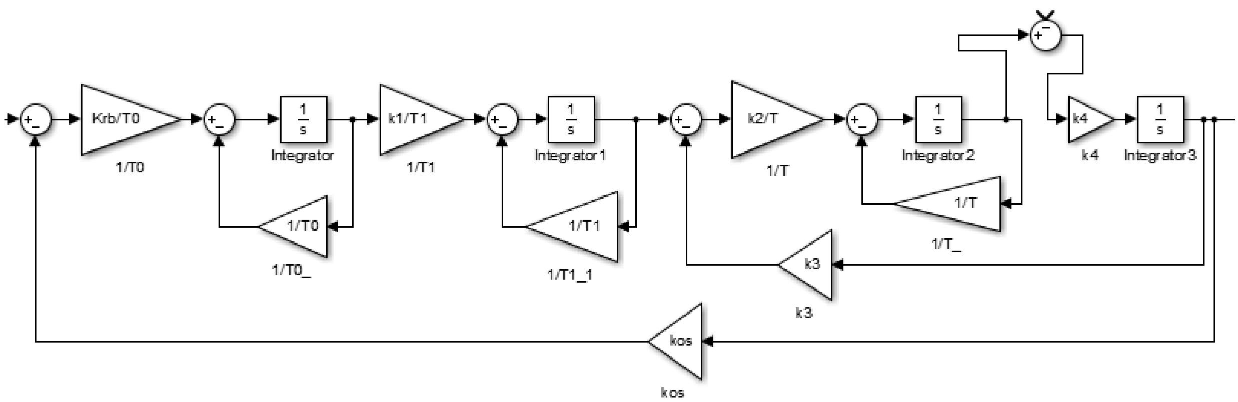

Рис.3. Структурная схема системы в общем виде

2. Определение передаточных функций отдельных структурных частей и системы в целом.

Передаточная функция решающего блока:

$$W_{рб}(p) = \frac{Kрб}{T0p + 1} = \frac{1}{20p + 1}$$

Передаточная функция усилителя мощности:

$$W_{ус}(p) = \frac{k1}{T1p + 1} = \frac{50}{0.55p + 1}$$

Передаточная функция объекта управления:

$$W_{оу}(p) = \frac{k_2 k_4}{Tp^2 + p + k_2 k_3 k_4} = \frac{0.4}{0.09p^2 + p + 1}$$

Передаточная функция всей системы:

$$W_{\text{системы}}(p) = \frac{k_{\text{рб}} k1k2k4}{(T0p+1)(T1p+1)\big((Tp+1)p + k2k3k4\big) + k_{\text{ос}} k2k4k1k_{\text{рб}}} =$$

$$= \frac{20}{0.99p^4 + 12.8495p^3 + 31.64p^2 + 21.55p + 21}$$

3. Анализ устойчивости всей системы и ОУ критериями Гурвица и Рауса.

Критерий Гурвица:

1) ОУ

Характеристическое уравнение: $0.09p^2 + p + 1 = 0$

Все коэффициенты положительные, соблюдается необходимое условие устойчивости.

Матрица Гурвица:

$$\begin{bmatrix} 1 & 0 \\ 0.09 & 1 \end{bmatrix}$$

$\Delta_1 = 1, \Delta_2 = 1$

Все главные определители матрицы Гурвица положительные, значит объект управления устойчив.

2) Вся система

Характеристическое уравнение: $0.99p^4 + 12.8495p^3 + 31.64p^2 + 21.55p + 21 = 0$

Необходимое условие также соблюдается.

Матрица Гурвица:

$$\begin{bmatrix} 12.8495 & 21.55 & 0 & 0 \\ 0.99 & 31.64 & 21 & 0 \\ 0 & 12.8495 & 21.55 & 0 \\ 0 & 0.99 & 31.64 & 21 \end{bmatrix}$$

$\Delta_1 = 12.8495, \Delta_2 = 385.22368, \Delta_3 = 4834.26764875, \Delta_4 = 101519.62062375$

Главные определители матрицы Гурвица положительные, система устойчивая.

4. Анализ устойчивости системы по критерию Рауса.

1) Объект управления

Построим таблицу Рауса по коэффициентам характеристического уравнения объекта управления:

Таблица 2 – Таблица Рауса для объекта управления

Cij	1	2	r
1	0.09	1	-
2	1	-	-
3	1	-	0.09

Коэффициенты первого столбца таблицы положительны, следовательно, объект управления устойчив.

2) Вся система

Построим таблицу Рауса по коэффициентам характеристического уравнения системы:

Таблица 3 – Таблица Рауса для всей системы

Cij	1	2	3	r
1	0.99	31.64	21	-
2	12.8495	21.55	-	-
3	29.98	21	-	0.077
4	9.2665	-	-	0.43
5	21	-	-	3.24

Коэффициенты первого столбца таблицы положительны, следовательно, система устойчива.

5. Анализ устойчивости системы по критерию Найквиста.

Передаточная функция разомкнутой системы:

$$W\text{раз}(p) = \frac{20}{0.99p^4 + 12.8495p^3 + 31.64p^2 + 21.55p + 1}$$

Характеристическое уравнение: $0.99p^4 + 12.8495p^3 + 31.64p^2 + 21.55p + 1$

Проверка устойчивости методом Гурвица:

$$\begin{bmatrix} 12.8495 & 21.55 & 0 & 0 \\ 0.99 & 31.64 & 1 & 0 \\ 0 & 12.8495 & 21.55 & 0 \\ 0 & 0.99 & 31.64 & 1 \end{bmatrix}$$

$\Delta_1 = 12.8495, \Delta_2 = 385.22368, \Delta_3 = 8136{,}46065375, \Delta_4 = 8136{,}46065375$

Главные определители положительные, значит разомкнутая система устойчивая.

Амплитудно-фазовая характеристика разомкнутой системы:

$$\mathrm{W_p(jw)} = \frac{20}{0.98\omega^8 + 102.46\omega^6 + 449.26\omega^4 + 401\omega^2 + 1}(1 - 31.64\omega^2 + 0.99\omega^4 - j\omega(21.55 - 12.85\omega^2))$$

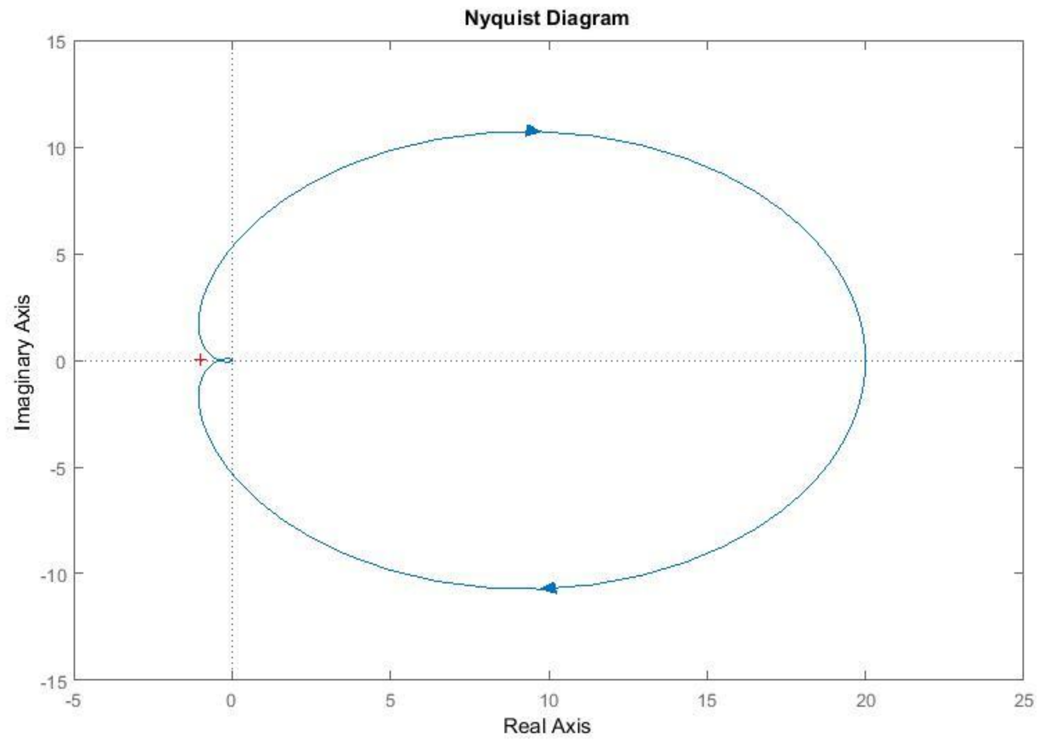

Рис.4. Амплитудно-фазовая характеристика разомкнутой системы

Амплитудно-фазовая характеристика не охватывает точку (-1,0), соответственно замкнутая система устойчивая.

6. Расчёт статического режима системы.

$$\Delta(p) = V(p) - Y(p) = V(p) - k_{\text{ос}}W_{\text{оу}}(W_{\text{рб}}W_{\text{ус}}\Delta - Z(p)\frac{Tp+1}{k_2})$$

$$\Delta(p) = \frac{1}{1 + k_{\text{ос}}W_{\text{рб}}W_{\text{ус}}W_{\text{оу}}}V(p) + \frac{Tp+1}{k2(1 + k_{\text{ос}}W_{\text{рб}}W_{\text{ус}}W_{\text{оу}})}Z(p)$$

$$t \to \infty, p \to 0$$

$$\mathrm{W}_{\text{рб}}(0) = k\text{рб} = 1$$

$$\mathrm{W}_{\text{ус}}(0) = k_1 = 50$$

$$\mathrm{W}_{\text{оу}}(0) = \frac{k_2k_4}{k_2k_3k_4} = 0.4$$

$$\delta_0 = \frac{v}{21} + \frac{z}{8.4}$$

7. Переходные характеристики системы и объекта управления.

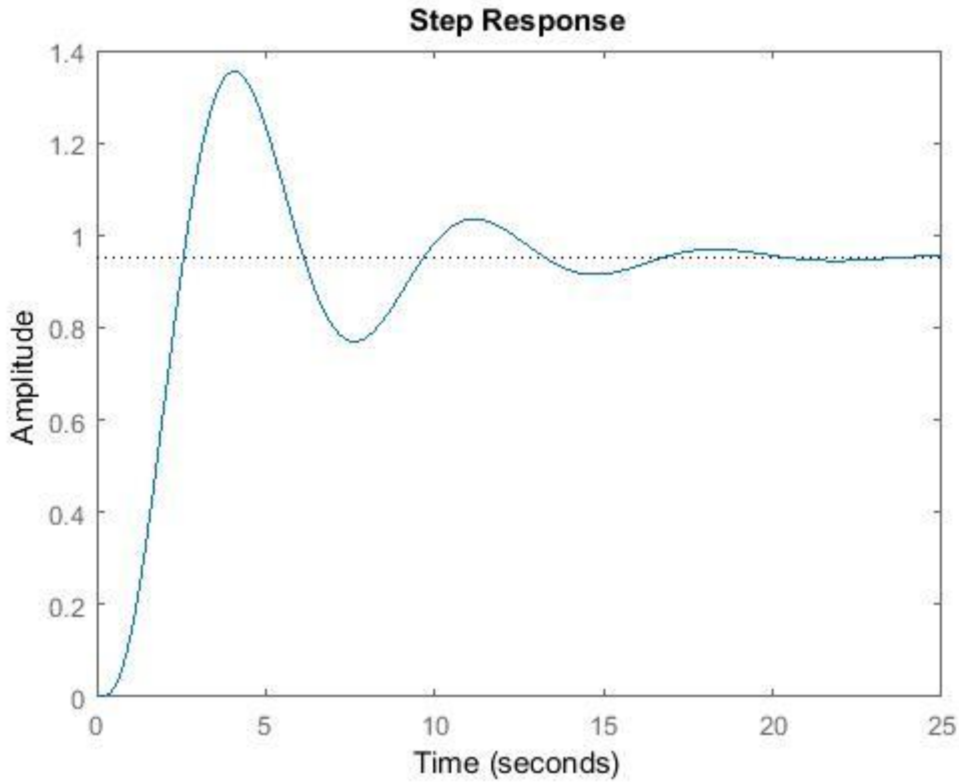

Рис.5 Переходная характеристика всей системы

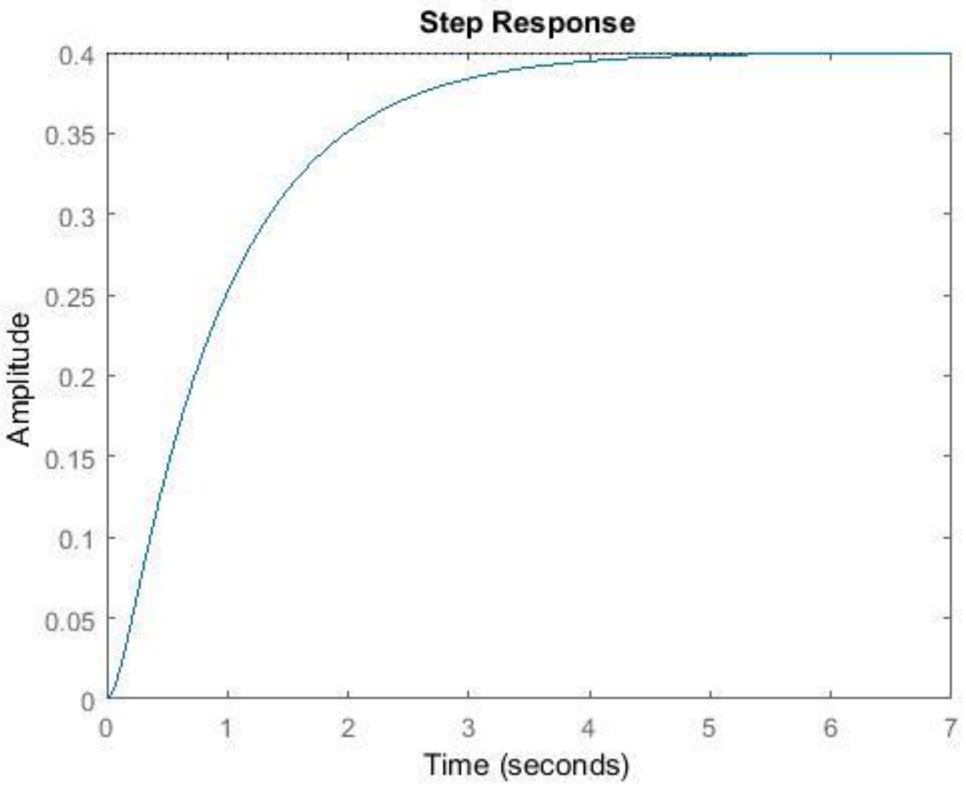

Рис.6. Переходная характеристика объекта управления

8. Временные диаграммы для переменных $x_1, x_2, x_3, x_{РБ}$, при условии, что входное воздействие изменяется по закону $V(t) = 0.3/(t + 0.6)$, а возмущающее – $Z(t) = 0.2\sin(5t)$.

Схема:

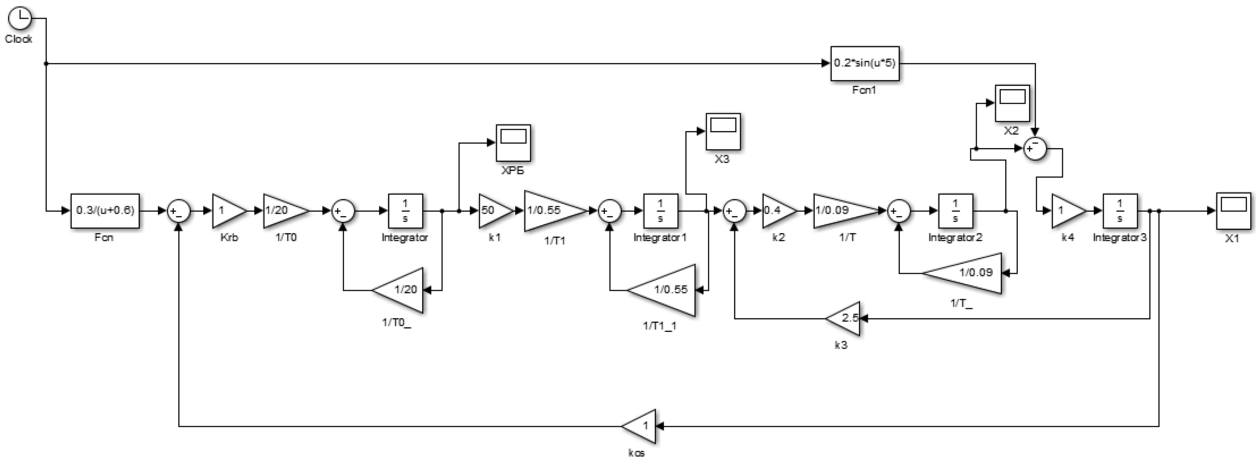

Рис.7. Схема в Simulink

Временные диаграммы:

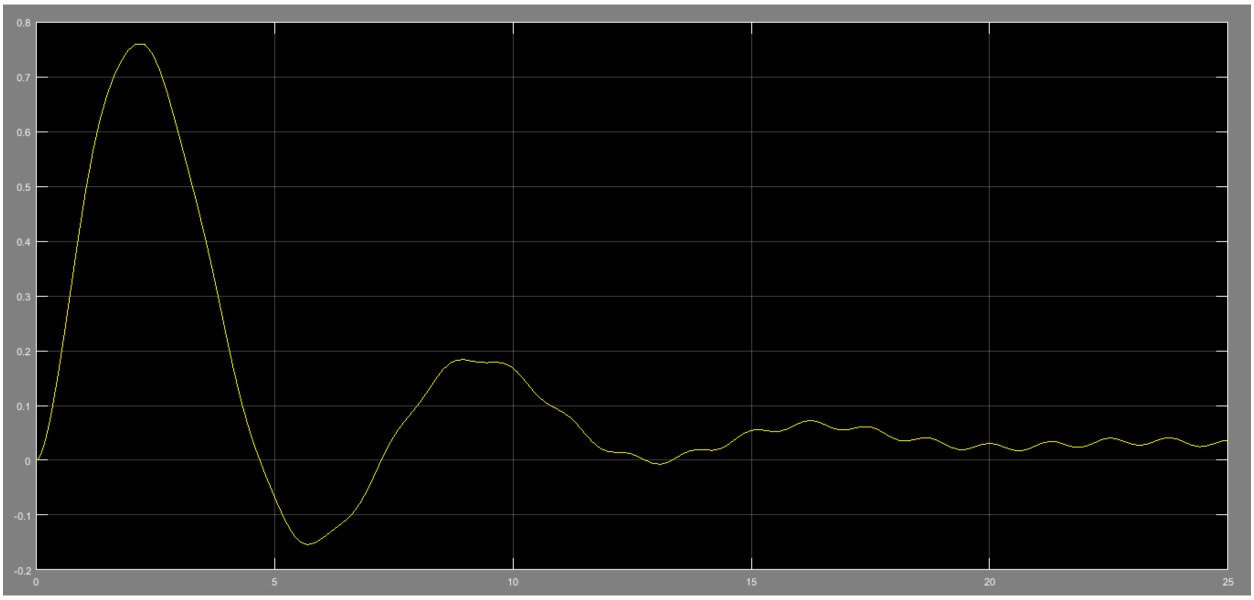

Рис.8. Временная диаграмма для переменной x3

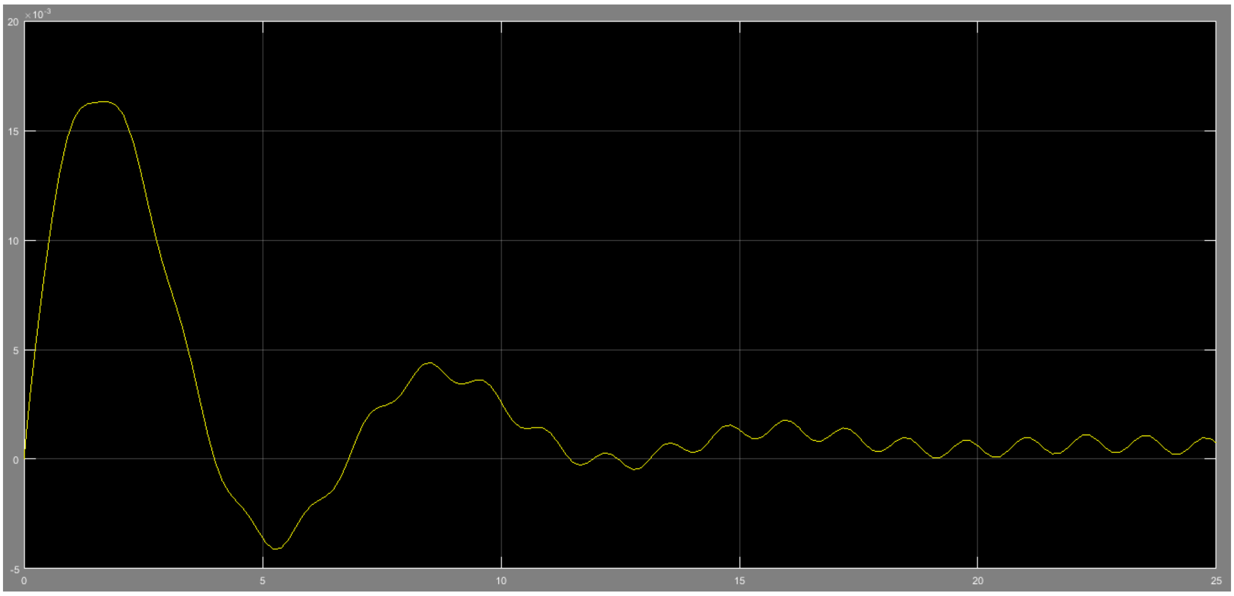

Рис.9. Временная диаграмма для переменной хрб

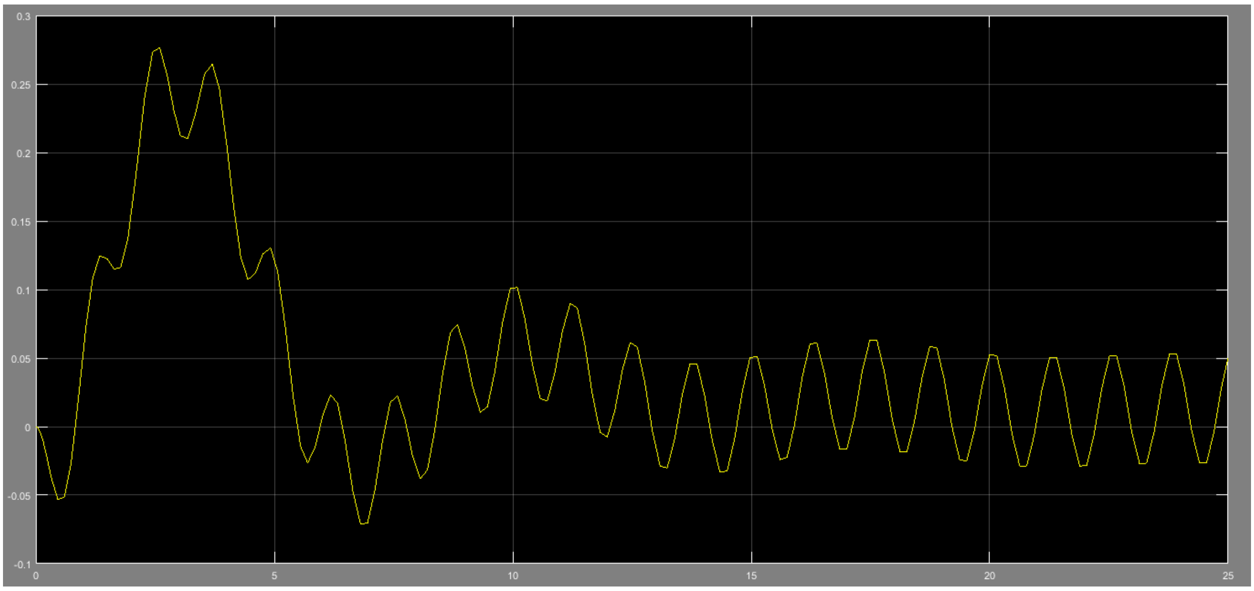

Рис.10. Временная диаграмма для переменной х1

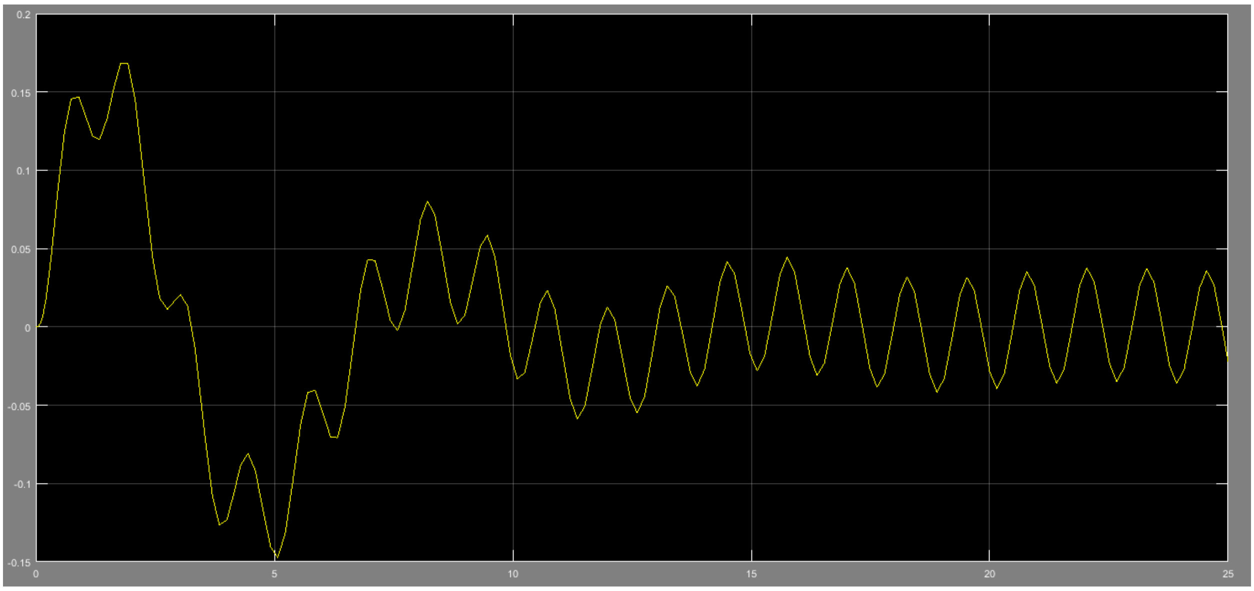

Рис.11. Временная диаграмма для переменной x2

9. Временные диаграммы для переменных $x_1, x_2, x_3, x_{РБ}$ при тех же воздействиях, полученные в собственной программе.

Временные диаграммы:

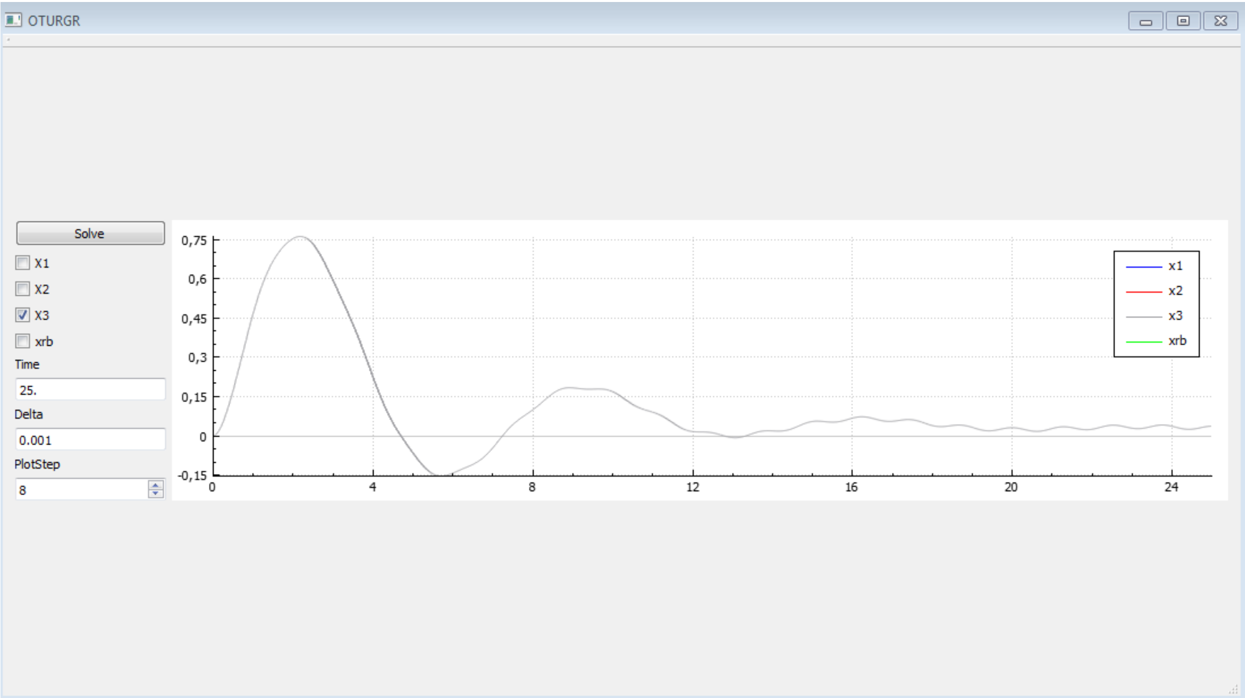

Рис.12. Временная диаграмма для переменной x3

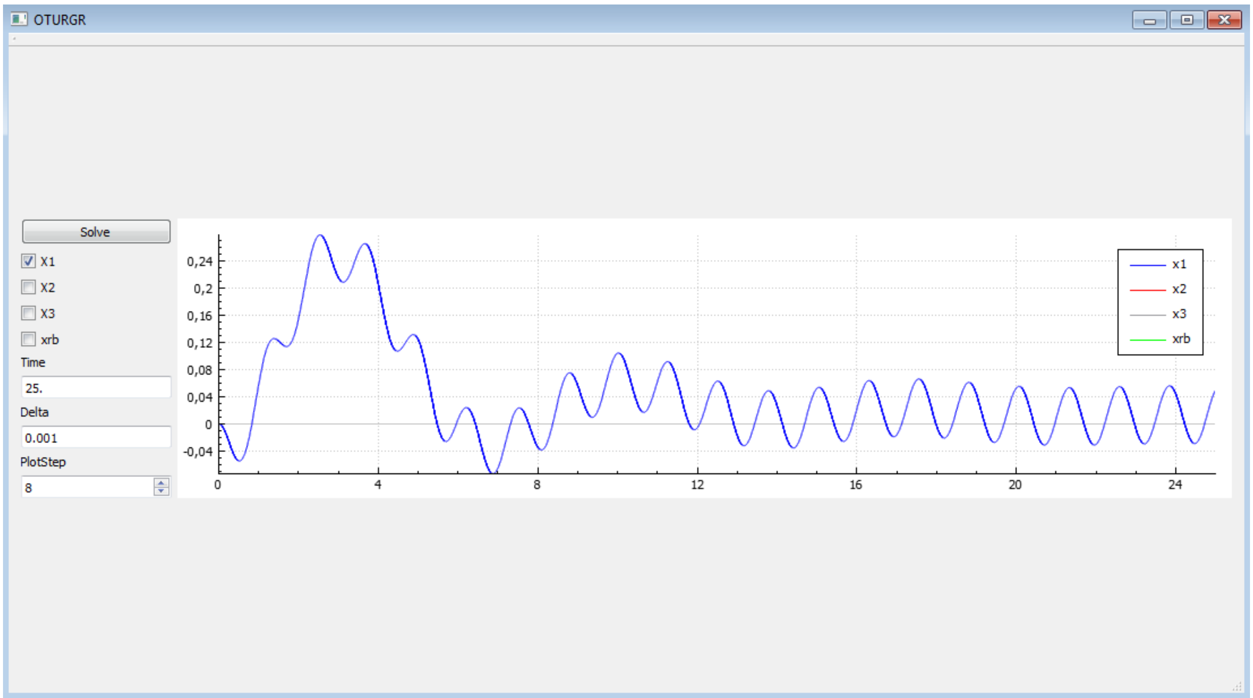

Рис.13. Временная диаграмма для переменной x1

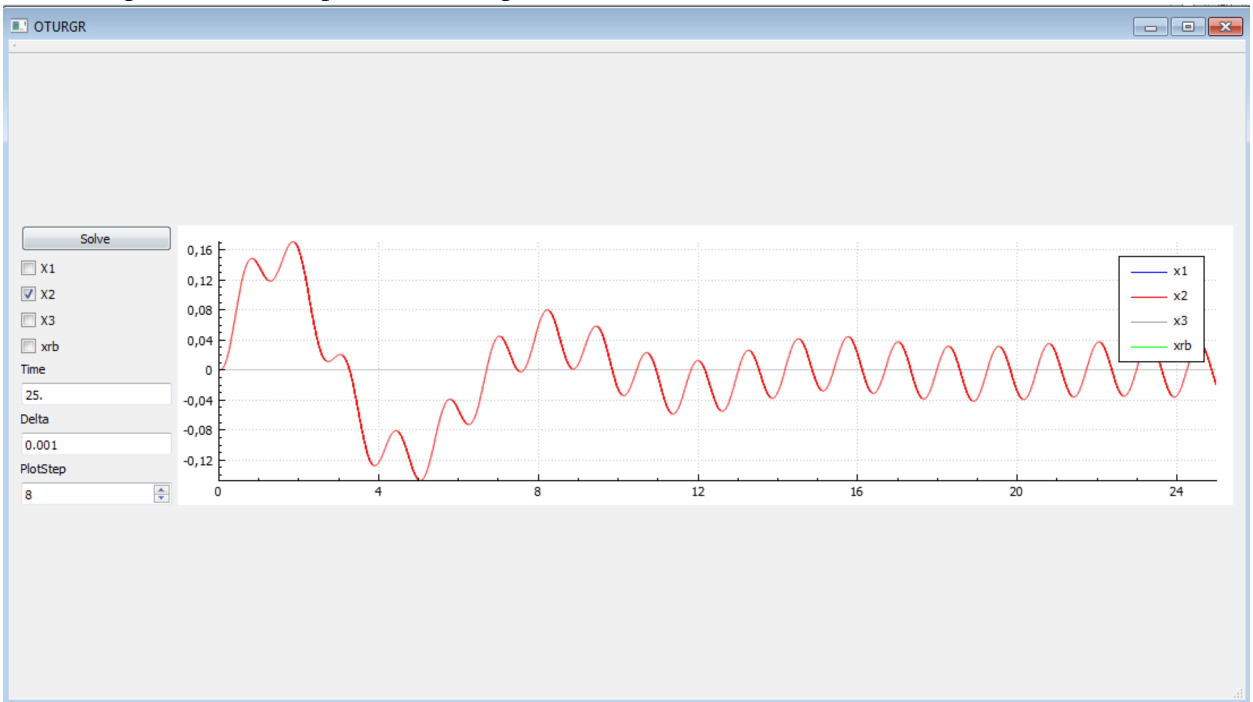

Рис.14. Временная диаграмма для переменной x2

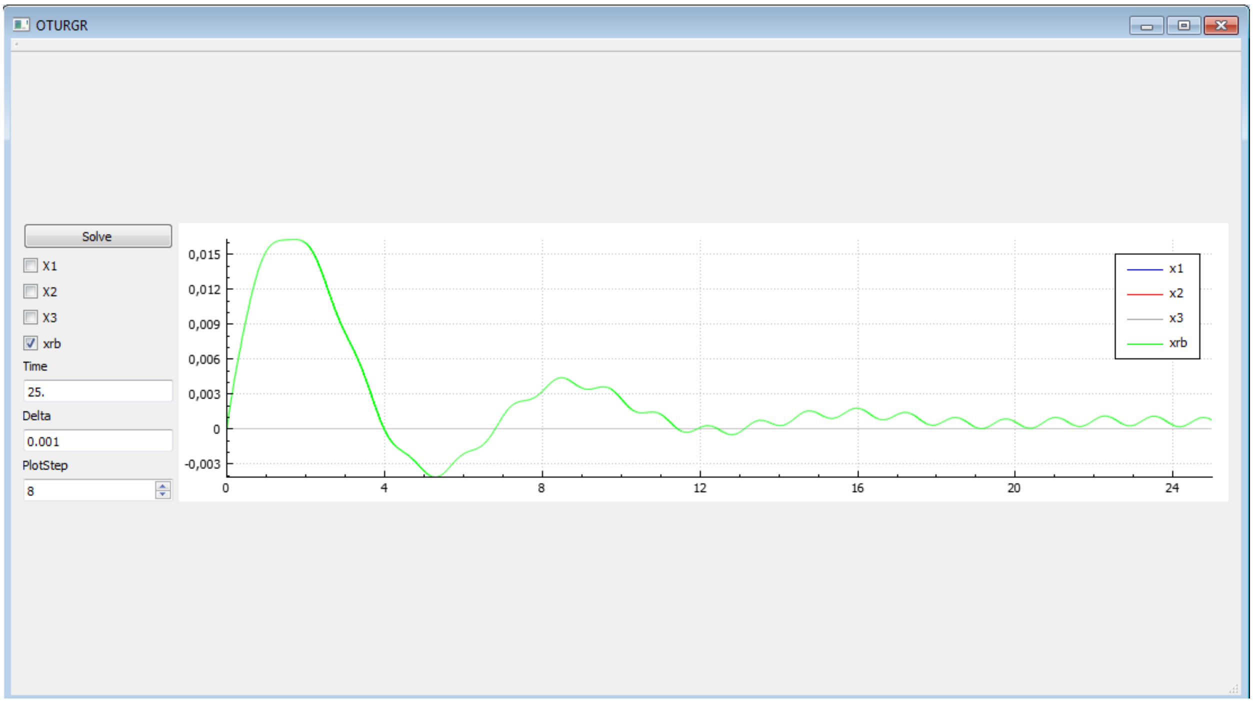

Рис.15. Временная диаграмма для переменной xpб

10. Выводы.

Целью работы был анализ системы автоматического управления и исследование реакций системы на различные входные и возмущающие воздействия методом компьютерного моделирования. В ходе работы были получены передаточные характеристики элементов системы и системы в целом, переходные характеристики объекта управления и всей системы, проверена устойчивость системы управления и объекта управления, как оказалось, они устойчивы, это подтверждается видом переходных характеристик. Получены временные диаграммы для переменных $x_1, x_2, x_3, x_{\text{РБ}}$ при заданных входном и возмущающем воздействиях, написана программа, для решения системы уравнения (1), в ней также были получены временные диаграммы для переменных, которые подтверждают корректность работы данной программы.

Листинг программы, используемой в пункте 9:

Модуль, генерирующий выходные данные

```
#ifndef OTURGR_H
#define OTURGR_H
typedef double(*func)(double t);
typedef double(*FUNC)(double* x,double t);
template <func V>
double XRB(double* x,double t)//x1,x2,x3,xrb,t count=4
{
    const double krbt0=1./20., t0=krbt0,kos=1.;
    return krbt0*(V(t) - kos*x[0])-t0*x[3];//x1
}
template<func Z>
double X1(double* x, double t)
{
    const double k4=1.f;
    return k4*(x[1] - Z(t));
}
double X2(double* x, double t)
{
    const double k2t=0.4/0.09, k3=2.5, T=1./0.09;//k2/T k3 1/T
    return k2t*(x[2] - k3*x[0]) - T*x[1];
}
double X3(double* x, double t)
{
    const double k1t1=50/0.55, t1=1.f/0.55;
    return k1t1*x[3] - t1*x[2];//xrb x3
}

void rungekutta(double *x, double t, double delta, FUNC* funcs, int count)
{
    double *k = new double[4 * count];
    double *ktemp = new double[count];
    for (int i = 0; i < count; i++)
    {
        k[i] = delta*funcs[i](x, t);
        ktemp[i] = x[i] + k[i] * 0.5;
    }
    for (int i = 0; i < count; i++)
    {
        k[count + i] = delta*funcs[i](ktemp, t + delta*0.5);
    }
    for(int i=0;i<count;i++)
       ktemp[i] = x[i] + k[count + i] * 0.5;
    int count2 = count +count;
    for (int i = 0; i < count; i++)
    {  k[count2 + i] = delta*funcs[i](ktemp, t + delta*0.5);
    }
    for(int i=0;i<count;i++)
       ktemp[i] = x[i] + k[count2 + i];

    int count3 = count + count2;
    for (int i = 0; i < count; i++)
        k[count3 + i] = delta*funcs[i](ktemp, t + delta);
    const double koeff = 1. / 6.;
    for (int i = 0; i < count; i++)
        x[i] += koeff*(k[i] + 2.*(k[i + count] + k[i + count2]) + k[i + count3]);
    delete[]ktemp;
    delete[]k;
}

double z(double t)
{
```

```
    return 0.2*sin(5.*t);
}
double v(double t)
{
    return 0.3/(t + 0.6);
}
#include <memory>
#define maxsize 20000
void solve(int count,double tc,double delta,FUNC*funcs,int skip,double*& data,int&
datasize)
{
    double t = 0;
    if (skip < 1)
        skip = 1;
    int sk = skip;
    datasize=(count+1)*(unsigned int((tc/delta))/skip);
    if(datasize<count||datasize>maxsize)
        return;
    data=new double[datasize];
    double *x = new double[count];
    memset(x, 0, count*sizeof(double));
    int dataiter=0;
    while (t <= tc&& dataiter<datasize)
    {
        if(sk==skip)
        {
           memcpy(&data[dataiter],x,int64_t(count*sizeof(double)));
           //for(int i=0;i<count;i++)
             //  data[dataiter+i]=x[i];
           dataiter+=count;
           data[dataiter]=t;
           dataiter++;
           sk=0;
         }
        rungekutta(x,t+delta,delta,funcs,count);
        t += delta;
        sk++;
    }
    delete[]x;
    datasize=dataiter;
}

#endif // OTURGR_H
```