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 2

Группа: АВТ-412

Преподаватель: Достовалов Д. Н.

Студентка: Васенкова Ю. В.

Новосибирск, 2016

Цель работы

Анализ системы автоматического управления и исследование реакций системы на различные входные и возмущающие воздействия методом компьютерного моделирования.

Задание

Уравнения, описывающие процессы в системе автоматического управления:

$$\begin{cases} \dot{x}_1 = k_4 (x_2 - Z), \\ \dot{x}_2 = \frac{k_2}{T}(x_3 - k_3 x_1) - \frac{1}{T} x_2, \\ \dot{x}_3 = \frac{k_1}{T_1} x_{РБ} - \frac{1}{T_1} x_3, \\ \dot{x}_{РБ} = \frac{k_{РБ}}{T_0} e - \frac{1}{T_3} x_{РБ}, \\ e = V - k_{ОС} x_1, \end{cases} \qquad (1)$$

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

V – входное воздействие;

Z – возмущающее воздействие;

x_1, x_2, x_3 – переменные состояния системы;

$k_{РБ}, k_{ОС}$ – передаточные 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 (обратно связи (ОС)) системы.

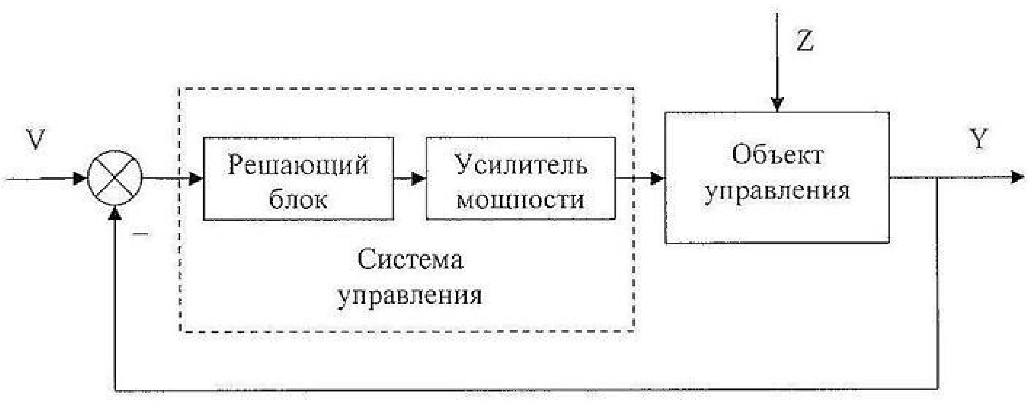

Рис.1. Обобщенная структура системы

Исходные данные

Таблица 1. Значения параметров звеньев исходной системы

Номер варианта	k_1	T_1	k_2	k_3	k_4	T	k_{OC}	Z_0	Δx_1^c
2	40	0.5	0,4	2,5	1	0.082	1	20	0.5

$k_{РБ} = 1$
$T_0 = 2$
$T_3 = 4$

Ход работы

1) Построение структурной схемы исходной системы по заданной системе уравнений.

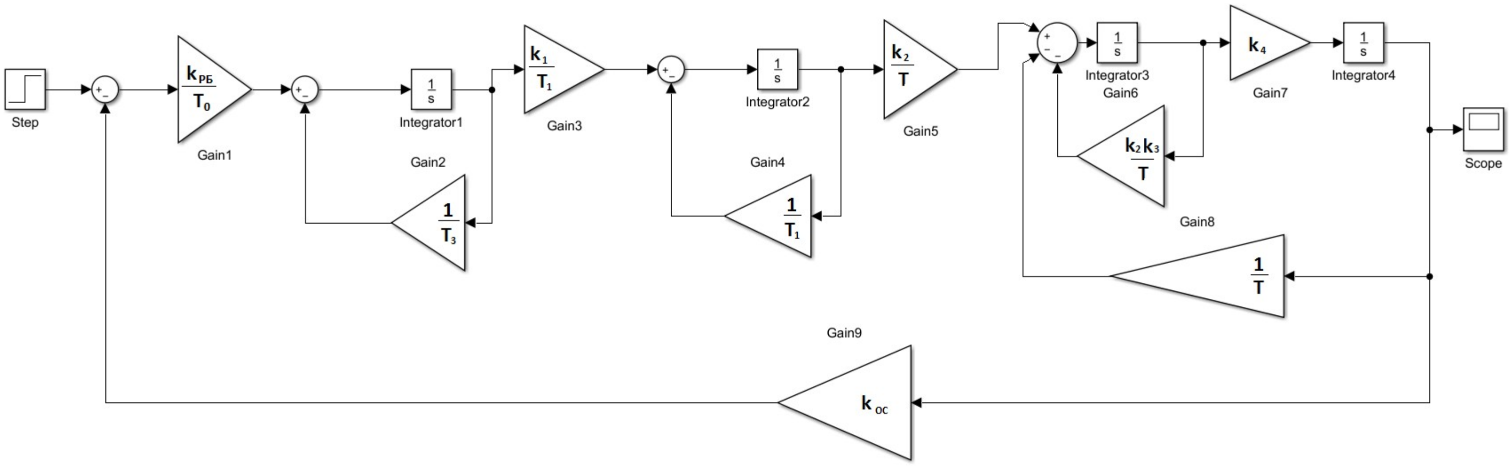

Рис.2. Структурная схема исходной системы

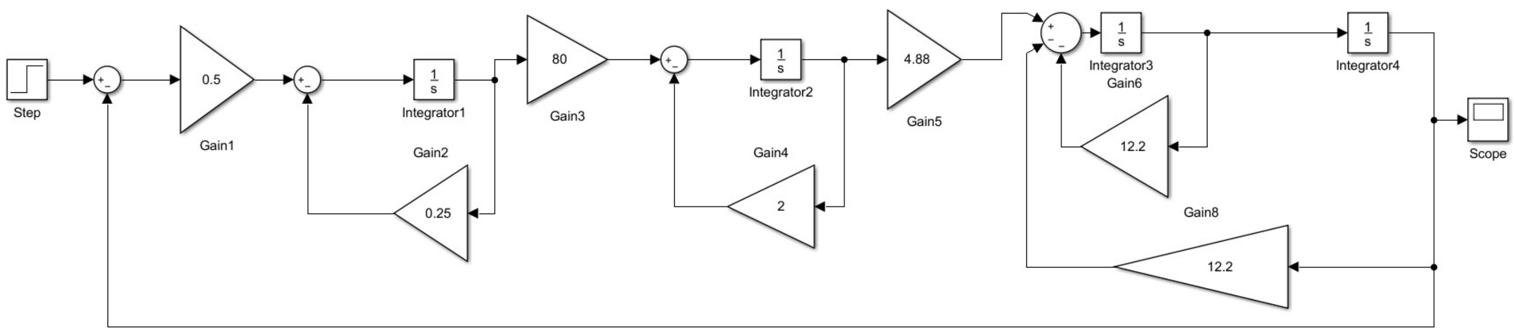

Рис.3. Структурная схема исходной системы (с рассчитанными параметрами)

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

```
Trial>> a1= series(w1,feedback(w2,w3))

a1 =

    0.5
  --------
  s + 0.25

Trial>> a2= series(w4,feedback(w5,w6))

a2 =

   80
  -----
  s + 2

Trial>> a3= series(4.88,feedback(series(feedback(w7,w8),w10),w9))

a3 =

         4.88
  -------------------
  s^2 + 12.2 s + 12.2

a3 =

        0.4
  ----------------
  0.08 s^2 + s + 1
```

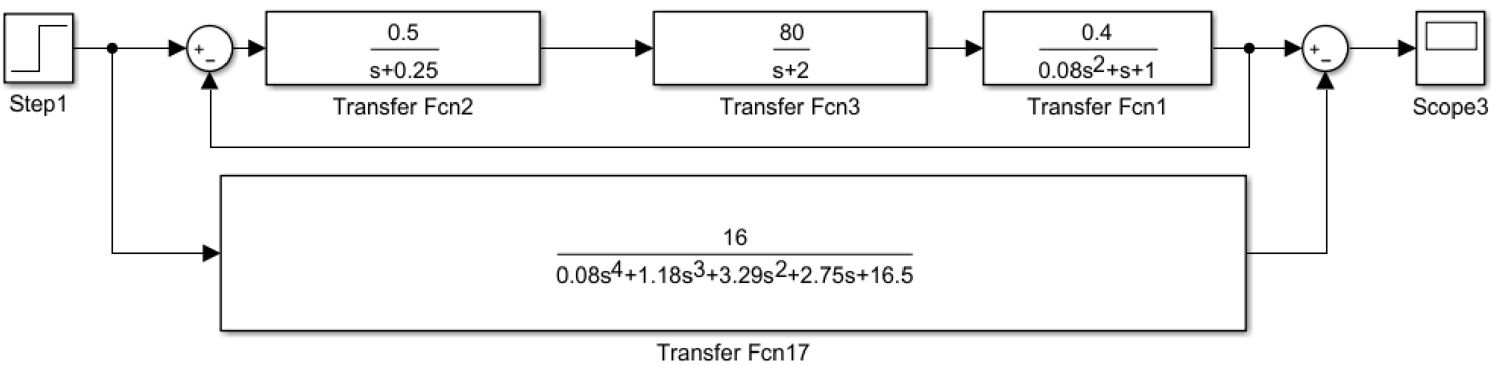

Рис.4. Передаточные функции отдельных частей и системы в целом

3) Анализ устойчивости объекта управления и системы в целом по критерию Гурвица и Рауса.

Для устойчивости системы необходимо, чтобы все коэффициенты характеристического полинома были положительными, но положительность коэффициентов не является достаточным условием устойчивости.

$0,08s^4 + 1,18s^3 + 3,29s^2 + 2,75s + 16,5 = 0$

Необходимое усл. устойчивости выполняется.

Критерий Гурвица (алгебраический критерий)

Составим определитель Гурвица по след. правилам:

1. Главная диагональ содержит коэффициенты с a_{n-1} по a_0;
2. Над гл. диагональю – коэффициенты, индекс которых уменьшается на 1;
3. Под гл. диагональю – коэффициенты, индекс которых увеличивается на 1;

   Для устойчивости линейной системы необходимо и достаточно, чтобы были положительными n гл. определителей матрицы Гурвица.

Проверка устойчивости всей системы:

```
Trial>> matrix = [1.18 2.75 0 0; 0.08 3.29 16.5 0; 0 1.18 2.75 0; 0 0.08 3.29 16.5]

matrix =

    1.1800    2.7500         0         0
    0.0800    3.2900   16.5000         0
         0    1.1800    2.7500         0
         0    0.0800    3.2900   16.5000

Trial>> det(matrix(1:1,1:1))

ans =

    1.1800

Trial>> det(matrix(1:2,1:2))

ans =

    3.6622

Trial>> det(matrix(1:3,1:3))

ans =

  -12.9035
```

Где matrix – матрица Гурвица.

Для устойчивости линейной системы необходимо и достаточно, чтобы n гл. определителей матрицы Гурвица были положительными.

Исходя из вычислений, можно сделать вывод, что система *не устойчивая*, так как определитель 3 порядка является отрицательным.

Проверка устойчивости объекта управления:

$0{,}08s^2 + s + 1 = 0$

```
Trial>> matrix2 = [1 0; 0.08 1]

matrix2 =

    1.0000         0
    0.0800    1.0000

Trial>> det(matrix2(1:1,1:1))

ans =

     1

Trial>> det(matrix2(1:2,1:2))

ans =

     1
```

Где matrix2 – матрица Гурвица.

Исходя из вычислений, можно сделать вывод, что объект управления *устойчивый*, так как все определители матрицы Гурвица положительные.

Критерий Рауса (алгебраический критерий)

Заполним специальную таблицу по алгоритму ниже:

1. В первой строке таблицы записываются коэффициенты характеристического уравнения, начиная с a_n, через один, по убыванию индекса;
2. Во второй строке таблицы записываются коэффициенты характеристического уравнения, начиная с a_{n-1}, через один, по убыванию индекса;
3. Для построения третьей и последующих строк вычисляется величина r_i по формуле: $r_i = c_{i-2,1} / c_{i-1,1}$;
4. Записывается третья и последующие строки, элементы которых вычисляются по формуле: $c_{ij} = c_{i-2,j}+1 - r_i * c_{i-1,j+1}$.
   Число строк в таблице должно быть равно n+1.

	1	**2**	**3**	**r_i**
1	$c_{11} = a_n$	$c_{12} = a_{n-2}$	$c_{13} = a_{n-4}$	-
2	$c_{21} = a_{n-1}$	$c_{22} = a_{n-3}$	$c_{23} = a_{n-5}$	-
3	$c_{31} = c_{12}-r_3*c_{22}$	$c_{32} = c_{13}-r_3*c_{23}$	$c_{33} = c_{14}-r_3*c_{24}$	$r_3 = c_{11} / c_{21}$
4	$c_{41} = c_{22}-r_4*c_{32}$	$c_{42} = c_{23}-r_4*c_{33}$	$c_{43} = c_{24}-r_4*c_{34}$	$r_4 = c_{21} / c_{31}$
5	$c_{51} = c_{32}-r_5*c_{42}$	$c_{52} = c_{33}-r_5*c_{43}$	$c_{53} = c_{34}-r_5*c_{44}$	$r_5 = c_{31} / c_{41}$

Проверка устойчивости всей системы:

	1	**2**	**3**	**r_i**
1	0,08	3,29	16,5	-
2	1,18	2,75	0	-
3	3,103	16,5	-	0,068
4	***-3,52***	-	-	0,38

Для устойчивости линейной системы необходимо и достаточно, чтобы коэффициенты первого столбца таблицы Рауса были одного знака (положительными).

Система *не устойчивая*, т. к. c_{41} = -3,52 (коэффициент отрицательный), а так как мы уже нашли отрицательный коэффициент вычисление в таблице можно прекратить.

Проверка устойчивости объекта управления:

	1	**2**	**r_i**
1	0,08	1	-
2	1	0	-
3	1	-	0,08

Объект управления <u>*устойчивый*</u>, т. к. все элементы первого столбца одного знака (положительные).

4) Анализ устойчивости системы автоматического управления по критерию Найквиста.

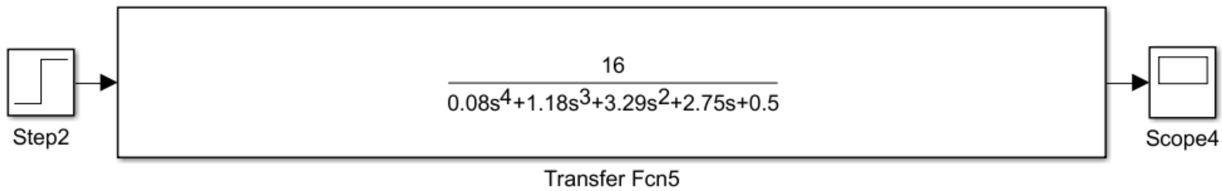

Рис.5. Передаточная функция разомкнутой системы

Критерий применим для систем с отрицательной обратной связью.

Критерий позволяет определить устойчивость замкнутой системы по амплитудно-фазовой характеристике и устойчивости разомкнутой системы.

1. *Если система в разомкнутом состоянии устойчива, то для устойчивости замкнутой системы необходимо и достаточно, чтобы амплитудно-фазовая характеристика разомкнутой системы при изменении частоты от 0 до +∞ не охватывала точку комплексной плоскости с координатами (−1, j0).*
2. *Если система в разомкнутом состоянии неустойчива и её характеристическое уравнение имеет k правых корней, то для устойчивости замкнутой системы необходимо и достаточно, чтобы частотная характеристика разомкнутой системы при изменении частоты от 0 до +∞ охватывала точку комплексной плоскости с координатами (−1, j0) k/2 раз.*

Проверим устойчивость разомкнутой системы методом Гурвица:

```
Trial>> matrix3 = [1.18 2.75 0 0; 0.08 3.29 0.5 0; 0 1.18 2.75 0; 0 0.08 3.29 0.5]

matrix3 =

    1.1800    2.7500         0         0
    0.0800    3.2900    0.5000         0
         0    1.1800    2.7500         0
         0    0.0800    3.2900    0.5000

Trial>> det(matrix3(1:1,1:1))

ans =

    1.1800

Trial>> det(matrix3(1:2,1:2))

ans =

    3.6622

Trial>> det(matrix3(1:3,1:3))

ans =

    9.3748

Trial>> det(matrix3(1:4,1:4))

ans =

    4.6874
```

Система в разомкнутом состоянии *устойчива*.

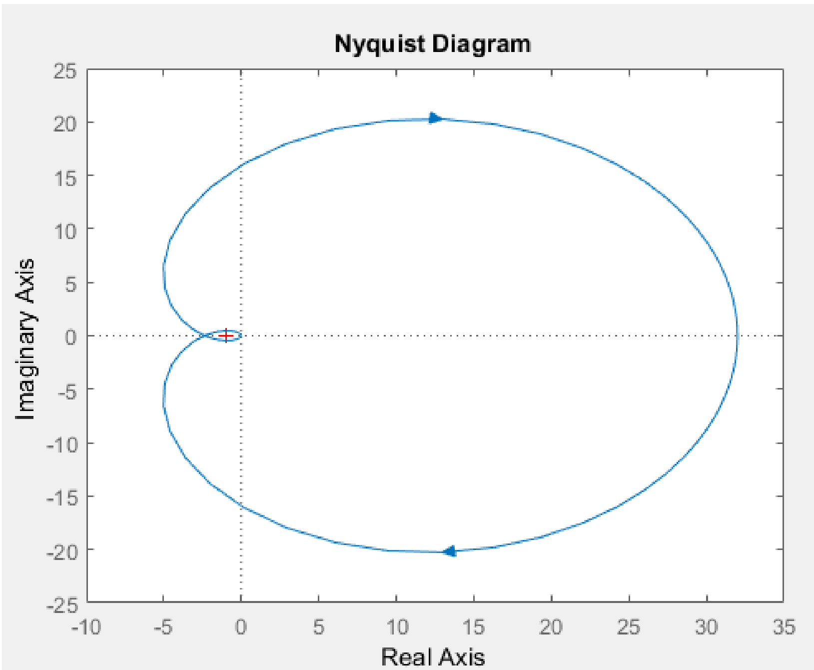

Рис.6. Амплитудно-фазовая характеристика разомкнутой системы

Замкнутая система *не устойчива*, так как годограф охватывает точку (-1; j0).

5) Расчет статического режима системы.

$$\begin{cases} \dot{x}_1 = k_4\left(x_2 - Z\right), \\ \dot{x}_2 = \dfrac{k_2}{T}\left(x_3 - k_3 x_1\right) - \dfrac{1}{T} x_2, \\ \dot{x}_3 = \dfrac{k_1}{T_1} x_{РБ} - \dfrac{1}{T_1} x_3, \\ \dot{x}_{РБ} = \dfrac{k_{РБ}}{T_0} e - \dfrac{1}{T_3} x_{РБ}, \\ e = V - k_{OC} x_1, \end{cases}$$

Уравнения необходимо приравнять к 0, подставить заданные значения и вычислить:

$$\begin{cases} x_2 - 20 = 0 \\ 4.88 * \left(x_3 - 2.5 * x_1\right) - 12.2 * x_2 = 0 \\ 80 * x_{РБ} - 2 * x_3 = 0 \\ 0.5 - 0.25 * x_{РБ} = 0 \end{cases}$$

$$\begin{cases} x_2 = 20 \\ 4.88 * x_3 - 12.2 * x_1 - 12.2 * x_2 = 0 \\ 80 * x_{РБ} - 2 * x_3 = 0 \\ x_{РБ} = 2 \end{cases}$$

$$\begin{cases} x_2 = 20 \\ 4.88 * x_3 - 12.2 * x_1 - 12.2 * x_2 = 0 \\ x_3 = 80 \\ x_{РБ} = 2 \end{cases}$$

В результате значения будут следующими:

$$\begin{cases} x_2 = 20 \\ x_1 = 12 \\ x_3 = 80 \\ x_{РБ} = 2 \end{cases}$$

6) Используя MATLAB/Simulink, получить переходные характеристики объекта управления и системы автоматического управления при отсутствии возмущающего воздействия.

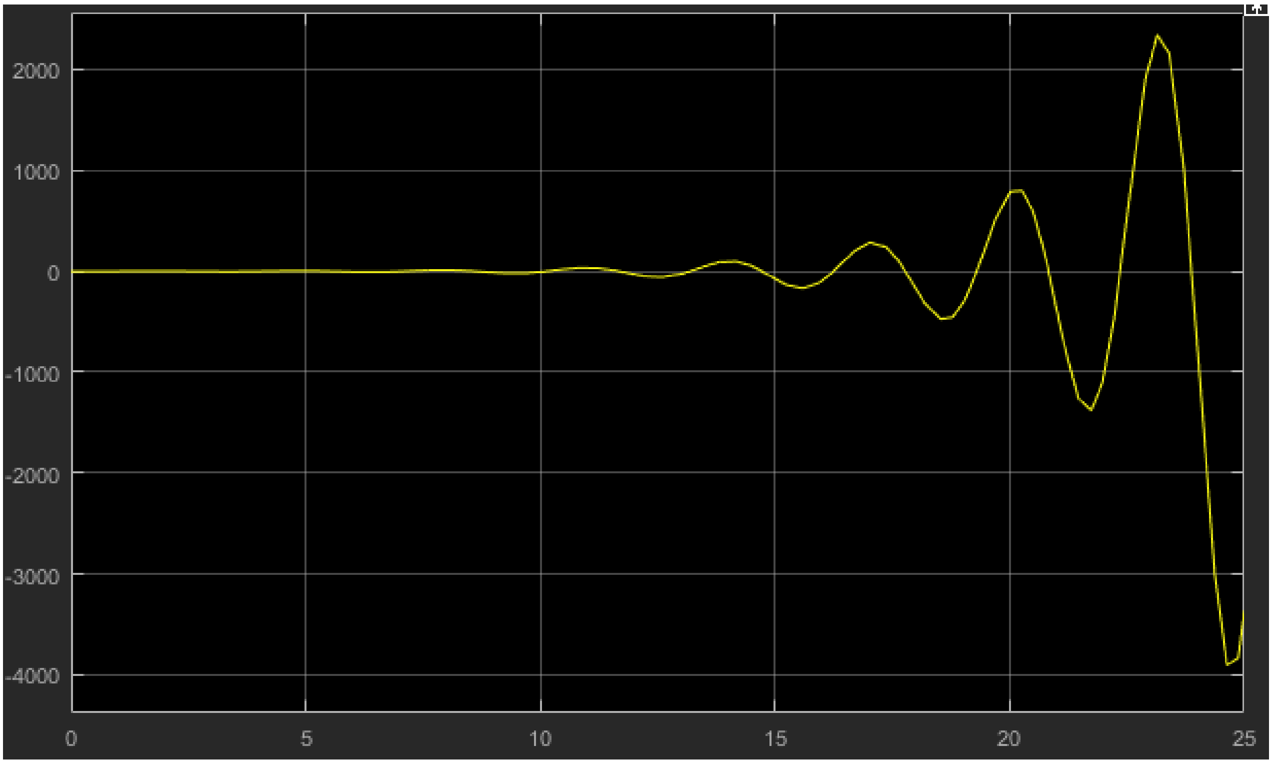

Рис.7. Переходная характеристика системы

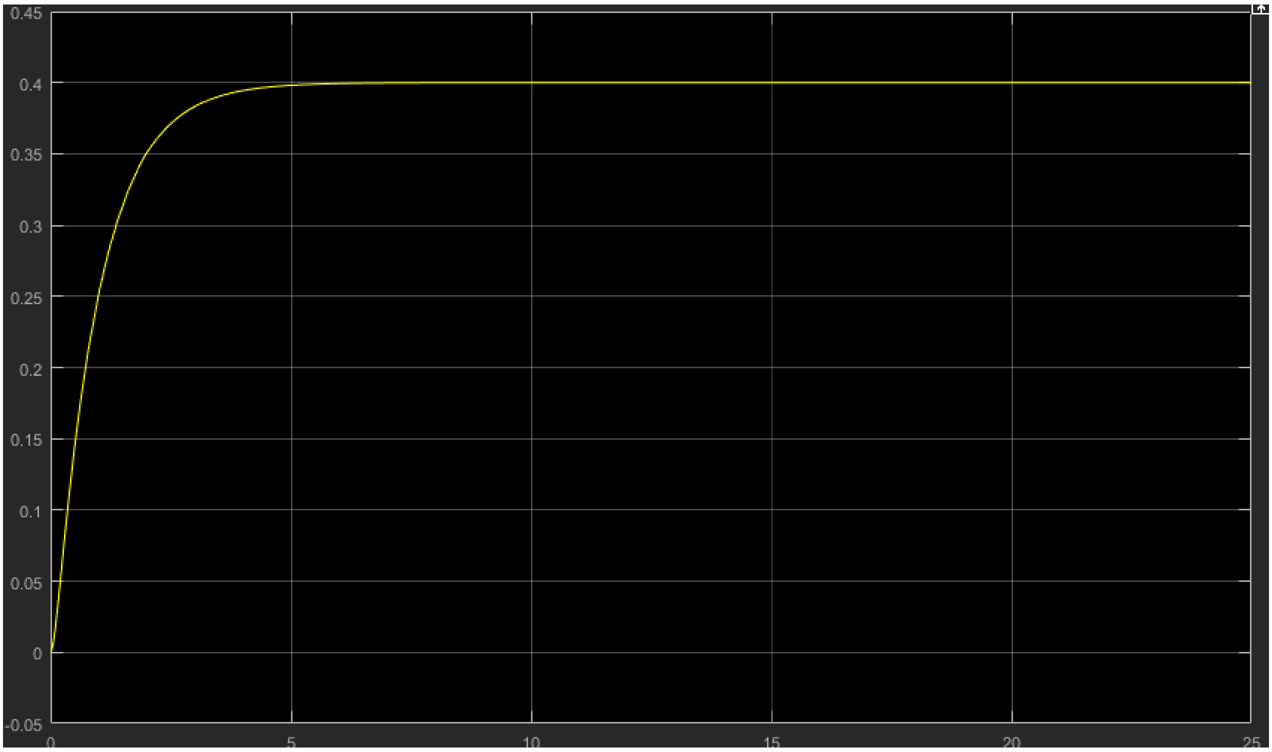

Рис.8. Переходная характеристика объекта управления

7) Используя MATLAB/Simulink, получить 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 – $Z(t)=0.2\sin(5t)$.

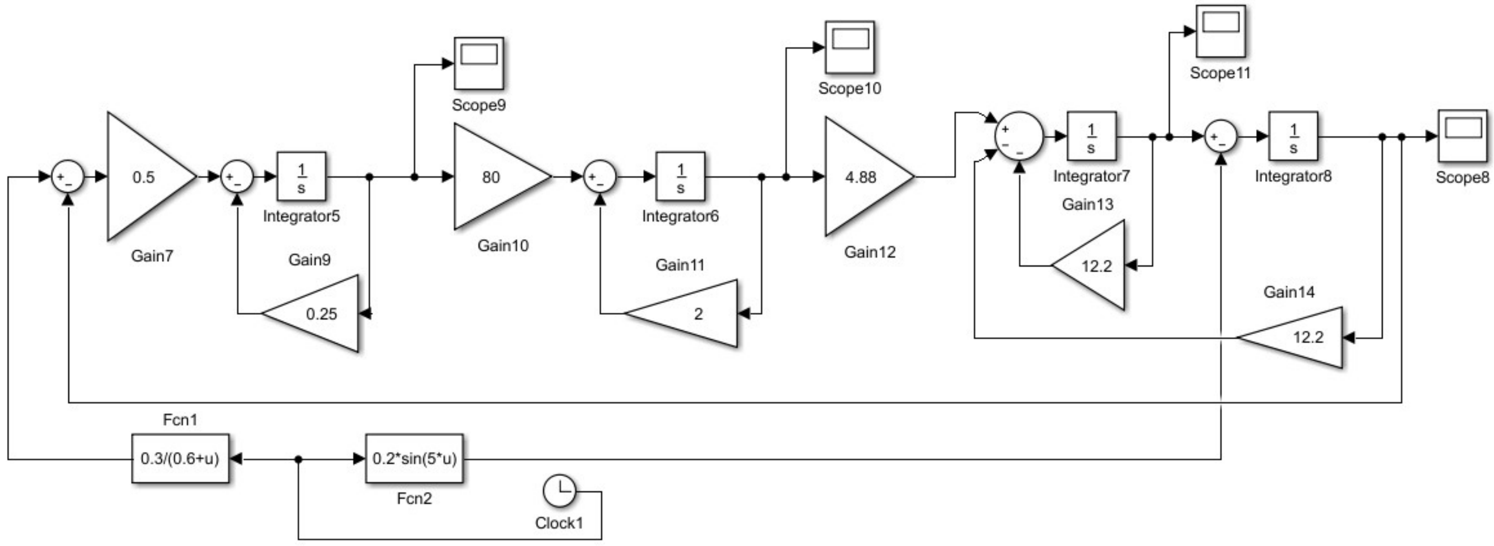

Рис.9. Структурная схема

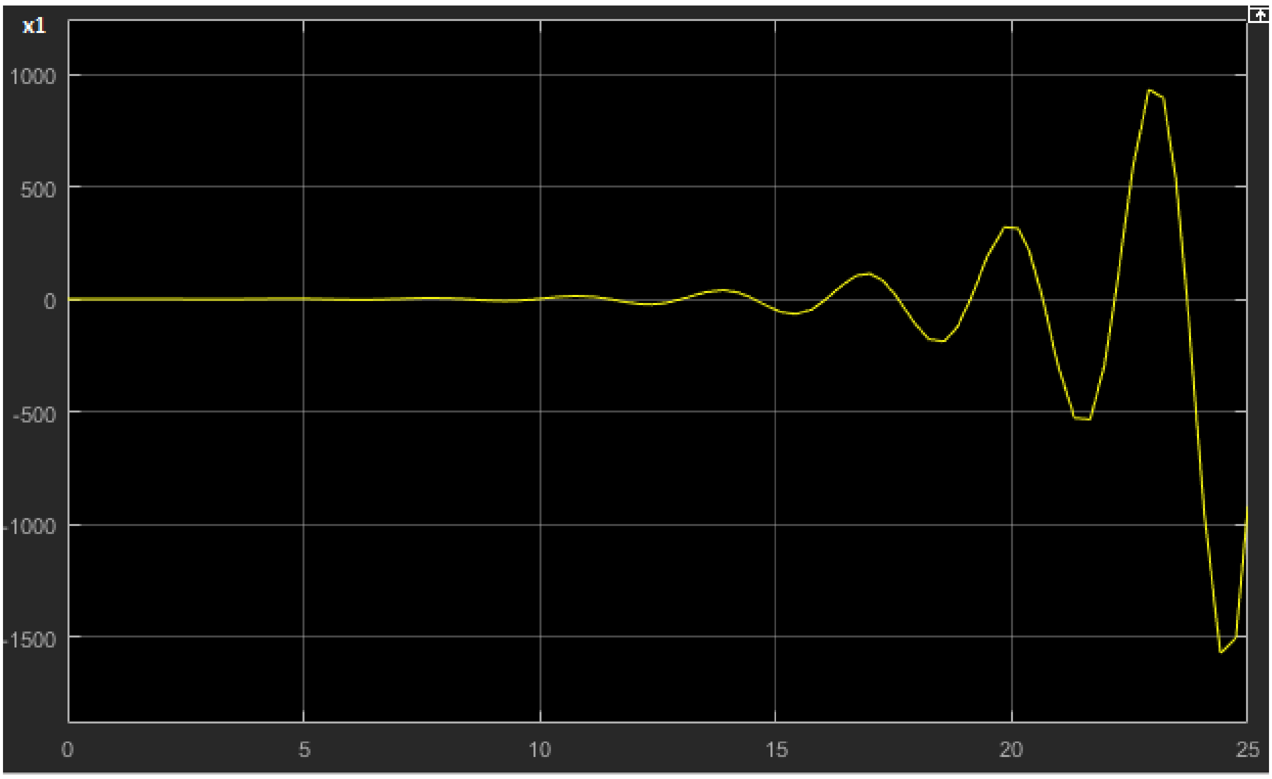

Рис.10. Временная диаграмма x_1

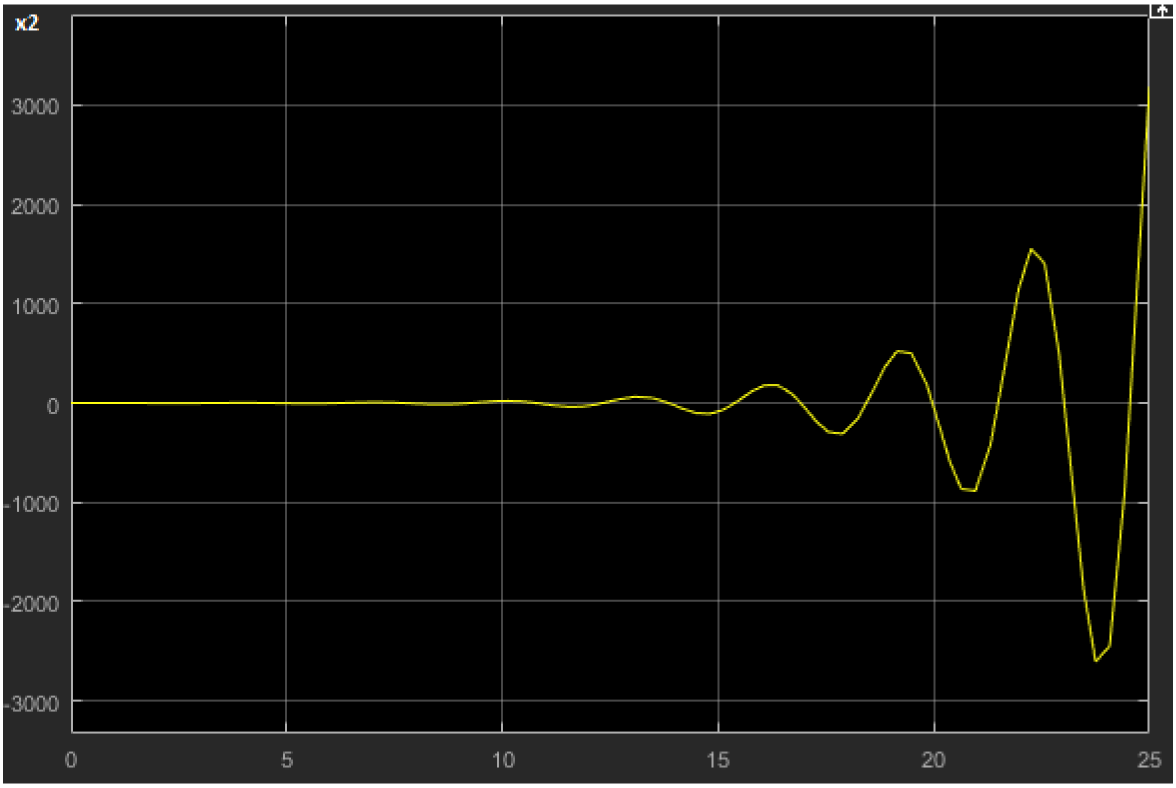

Рис.11. Временная диаграмма x_2

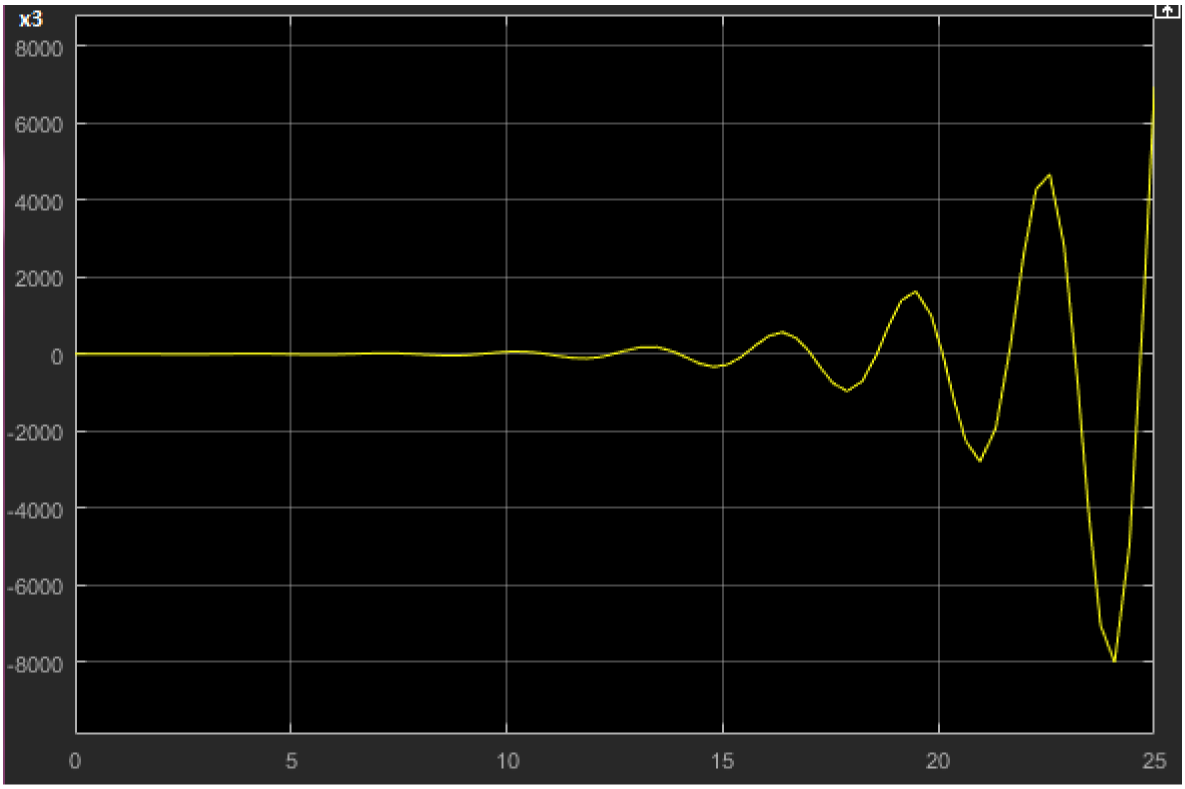

Рис.12. Временная диаграмма x_3

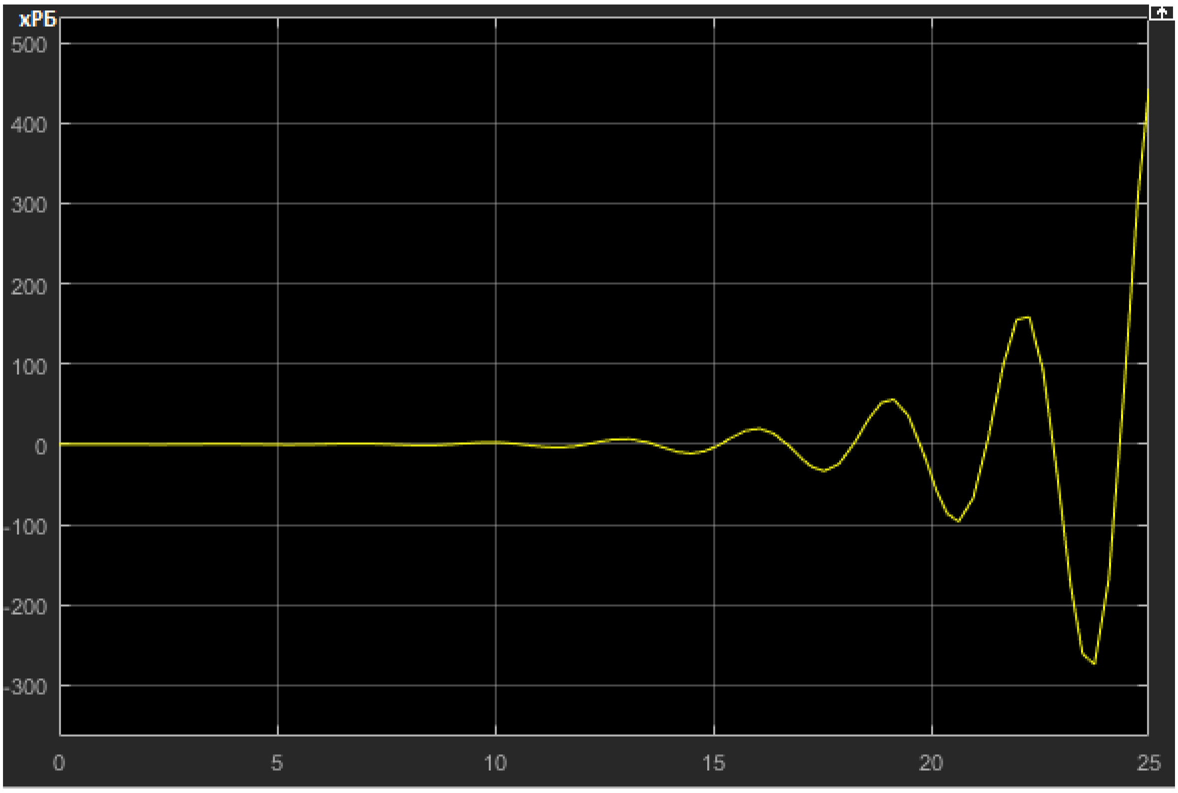

Рис.13. Временная диаграмма $x_{РБ}$

8) Написать программу для численного решения задачи (1) и получить временные диаграммы для переменных $x_1, x_2, x_3, x_{РБ}$ при тех же воздействиях.

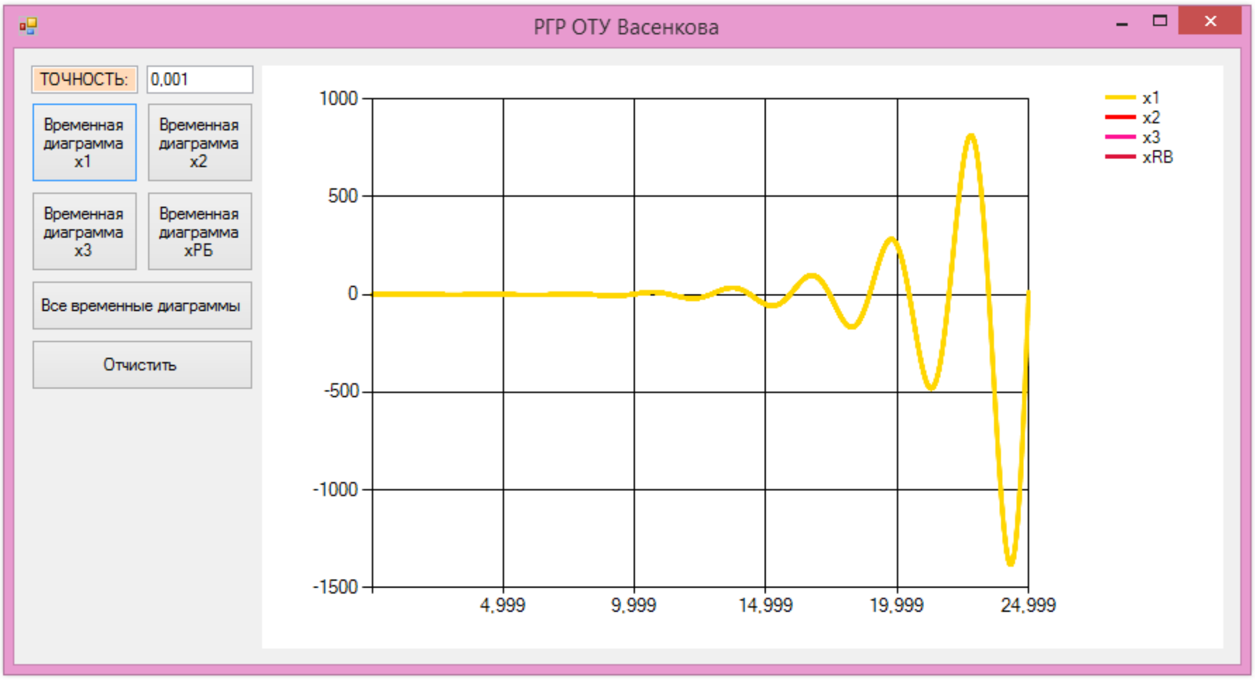

Рис.14. Временная диаграмма x_1

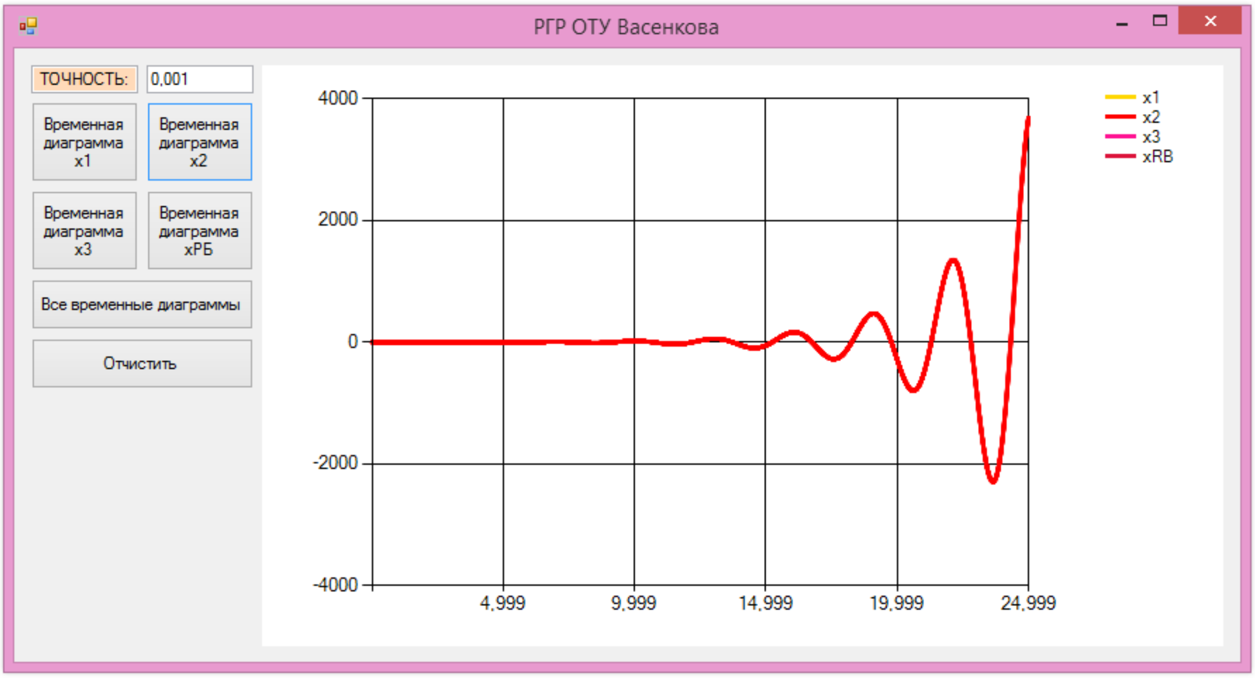

Рис.15. Временная диаграмма x_2

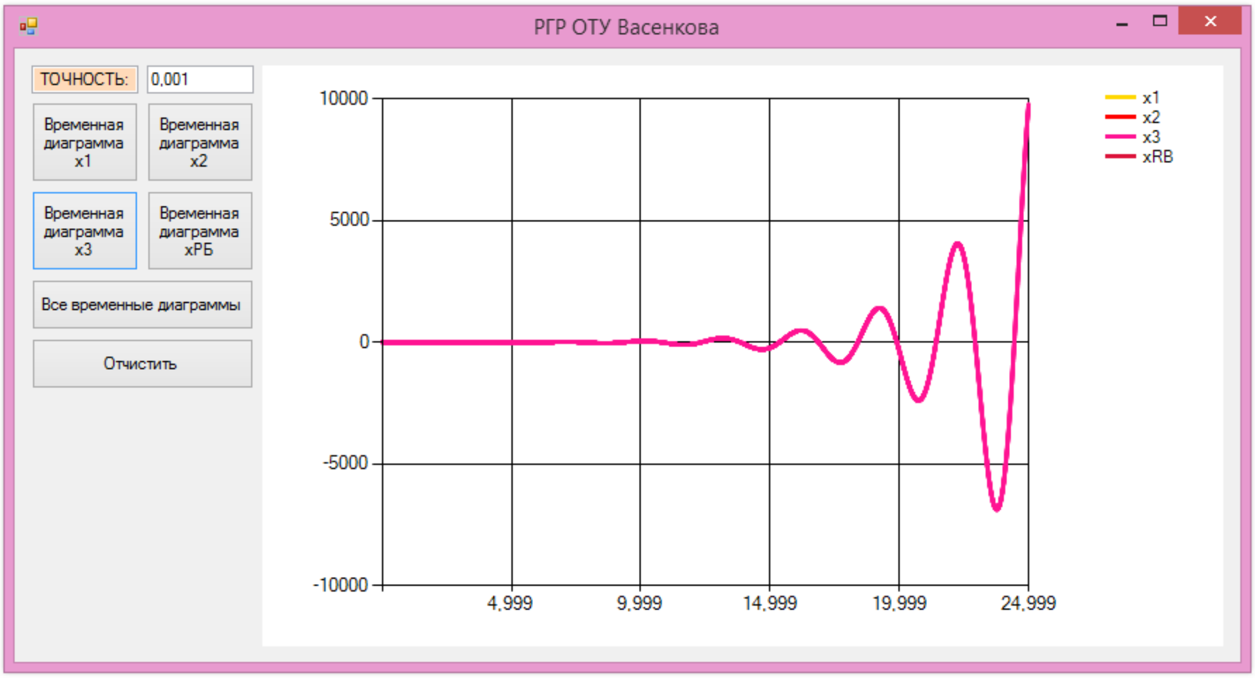

Рис.16. Временная диаграмма x_3

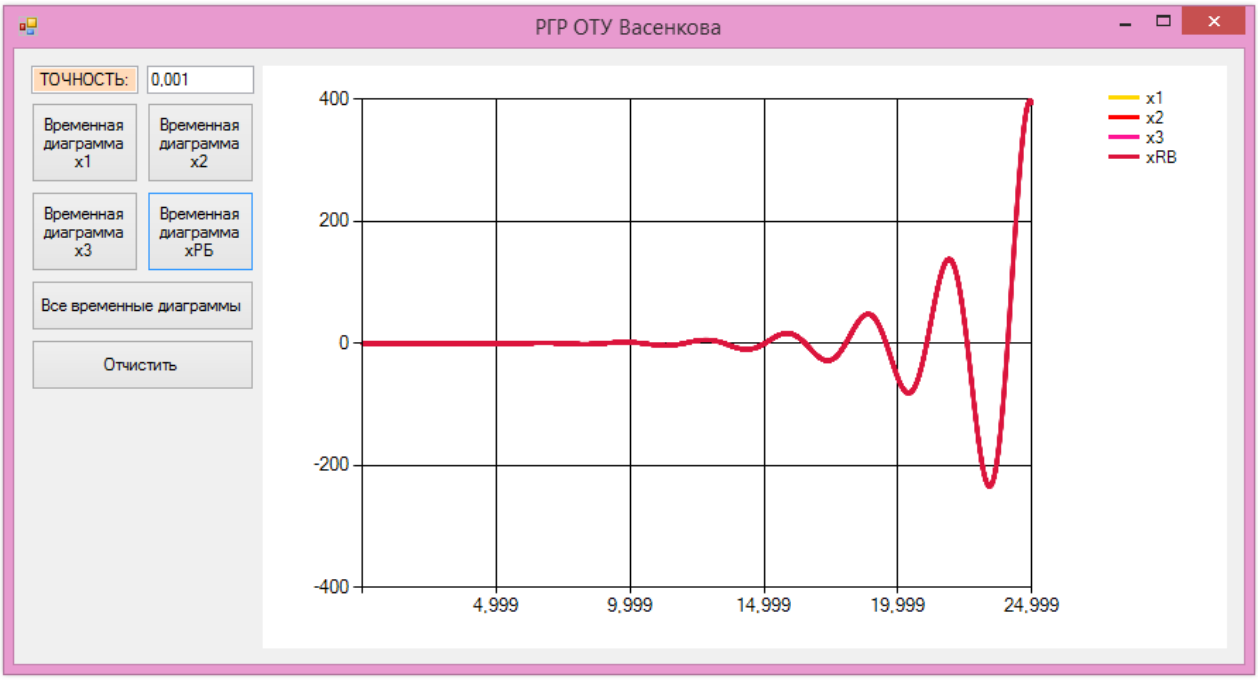

Рис.17. Временная диаграмма $x_{РБ}$

Выводы

Целью расчётно-графической работы был анализ системы автоматического управления и исследование реакций системы на различные входные и возмущающие воздействия методом компьютерного моделирования. В результате проделанной работы можно утверждать, что заданная система автоматического управления неустойчивая. Такой вывод был получен с помощью нескольких критериев: критерий Гурвица, Рауса, Найквиста, а также это можно определить по переходным характеристикам системы. Также была выявлена устойчивость объекта управления, найдены значения переменных x_1,x_2,x_3,$x_{РБ}$ в статическом режиме системы, переходные характеристики объекта управления, системы автоматического управления при отсутствии возмущающего воздействия и при условии, что входное воздействие изменяется по закону $V(t)=0.3/(t+0.6)$, а возмущающее – $Z(t)=0.2\sin(5t)$.

Листинг программы, написанной в п.8.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace RGR
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        List<double> x1 = new
List<double>();
        List<double> x2 = new
List<double>();
        List<double> x3 = new
List<double>();
        List<double> x4 = new
List<double>();
        List<double> t = new
List<double>();

        double x1max = 0;
        double x2max = 0;
        double x3max = 0;
        double x4max = 0;
        double x1min = 0;
        double x2min = 0;
        double x3min = 0;
        double x4min = 0;

        double Z(double t)
        {
            return 0.2 * Math.Sin(5 *
t * 3.14 / 180);
        }

        double V(double t)
        {
            return 0.3 / (t + 0.6);
        }

        double f1(double x2, double z)
        {
            return x2 - z;
        }

        double f2(double x1, double
x2, double x3)
        {
            return (0.4 / 0.082) * (x3
- 2.5 * x1) - x2 / 0.082;
        }

        double f3(double x3, double
x4)
        {
            return 40 * x4 / 0.5 -
x3 / 0.5;
        }

        double f4(double x1, double
x4, double v)
        {
            return 0.5 * (v - x1) - x4
/ 4;
        }

        void Runge_Kutt(double h)
        {
            int j = 1;
            double K1 = 0, K2 = 0, K3
= 0, K4 = 0;
            int N = (int)(10 / h) + 1;


            x1.Clear();
            x2.Clear();
            x3.Clear();
            x4.Clear();
            t.Clear();


            x1.Add(0);
            x2.Add(0);
            x3.Add(0);
            x4.Add(0);
            t.Add(0);
            t[0] = 0;
            x1max = 0;
            x2max = 0;
            x3max = 0;
            x4max = 0;
            x1min = 0;
            x2min = 0;
            x3min = 0;
            x4min = 0;
            for (double i = h; i <=
25; i += h)
            {
                t.Add(i);
                K1 = f1(x2[j - 1],
Z(t[j - 1]));
```

```
                K2 = f1(x2[j - 1] + h
* K1 / 2, Z(t[j - 1] + h / 2));
                K3 = f1(x2[j - 1] + h
* K2 / 2, Z(t[j - 1] + h / 2));
                K4 = f1(x2[j - 1] + h
* K3, Z(t[j - 1] + h));
                x1.Add(x1[j - 1] + h *
(K1 + 2 * K2 + 2 * K3 + K4) / 6);

                K1 = f2(x1[j - 1],
x2[j - 1], x3[j - 1]);
                K2 = f2(x1[j - 1] + h
* K1 / 2, x2[j - 1] + h * K1 / 2, x3[j
- 1] + h * K1 / 2);
                K3 = f2(x1[j - 1] + h
* K2 / 2, x2[j - 1] + h * K2 / 2, x3[j
- 1] + h * K2 / 2);
                K4 = f2(x1[j - 1] + h
* K3, x2[j - 1] + h * K3, x3[j - 1] +
h * K3);
                x2.Add(x2[j - 1] + h *
(K1 + 2 * K2 + 2 * K3 + K4) / 6);

                K1 = f3(x3[j - 1],
x4[j - 1]);
                K2 = f3(x3[j - 1] + h
* K1 / 2, x4[j - 1] + h * K1 / 2);
                K3 = f3(x3[j - 1] + h
* K2 / 2, x4[j - 1] + h * K2 / 2);
                K4 = f3(x3[j - 1] + h
* K3, x4[j - 1] + h * K3);
                x3.Add(x3[j - 1] + h *
(K1 + 2 * K2 + 2 * K3 + K4) / 6);

                K1 = f4(x1[j - 1],
x4[j - 1], V(t[j - 1]));
                K2 = f4(x1[j - 1] + h
* K1 / 2, x4[j - 1] + h * K1 / 2,
V(t[j - 1] + h / 2));
                K3 = f4(x1[j - 1] + h
* K2 / 2, x4[j - 1] + h * K2 / 2,
V(t[j - 1] + h / 2));
                K4 = f4(x1[j - 1] + h
* K3, x4[j - 1] + h * K3, V(t[j - 1] +
h));
                x4.Add(x4[j - 1] + h *
(K1 + 2 * K2 + 2 * K3 + K4) / 6);

            if (j == 1 || x1[j] >
x1max)
                {
                    x1max = x1[j];
                }
            if (j == 1 || x2[j] >
x2max)
                {
                    x2max = x2[j];
                }
            if (j == 1 || x3[j] >
x3max)
                {
                    x3max = x3[j];
                }
            if (j == 1 || x4[j] >
x4max)
                {
                    x4max = x4[j];
                }
            if (j == 1 || x1[j] <
x1min)
                {
                    x1min = x1[j];
                }
            if (j == 1 || x2[j] <
x2min)
                {
                    x2min = x2[j];
                }
           if (j == 1 || x3[j] <
x3min)
                {
                    x3min = x3[j];
                }
           if (j == 1 || x4[j] <
x4min)
                {
                    x4min = x4[j];
                }
                j++;
            }

        }

        private void
button1_Click(object sender, EventArgs
e)
        {

chart1.Series[0].Points.Clear();

chart1.Series[1].Points.Clear();

chart1.Series[2].Points.Clear();

chart1.Series[3].Points.Clear();


Runge_Kutt(Convert.ToDouble(textBox1.T
ext));
      for (int i = 0; i < t.Count; i+
+)
            {
```

```
chart1.Series[0].Points.AddXY(t[i], x1[i]);
            }
        }

        private void button2_Click(object sender, EventArgs e)
        {
chart1.Series[0].Points.Clear();
chart1.Series[1].Points.Clear();
chart1.Series[2].Points.Clear();
chart1.Series[3].Points.Clear();

Runge_Kutt(Convert.ToDouble(textBox1.Text));
      for (int i = 0; i < t.Count; i++)
            {
chart1.Series[1].Points.AddXY(t[i], x2[i]);
            }
        }

        private void button3_Click(object sender, EventArgs e)
        {
chart1.Series[0].Points.Clear();
chart1.Series[1].Points.Clear();
chart1.Series[2].Points.Clear();
chart1.Series[3].Points.Clear();

Runge_Kutt(Convert.ToDouble(textBox1.Text));
       for (int i = 0; i < t.Count; i++)
            {
chart1.Series[2].Points.AddXY(t[i], x3[i]);
            }
        }

        private void button4_Click(object sender, EventArgs e)
        {
chart1.Series[0].Points.Clear();
chart1.Series[1].Points.Clear();
chart1.Series[2].Points.Clear();
chart1.Series[3].Points.Clear();

Runge_Kutt(Convert.ToDouble(textBox1.Text));
       for (int i = 0; i < t.Count; i++)
            {
chart1.Series[3].Points.AddXY(t[i], x4[i]);
            }
        }

        private void button5_Click(object sender, EventArgs e)
        {
chart1.Series[0].Points.Clear();
chart1.Series[1].Points.Clear();
chart1.Series[2].Points.Clear();
chart1.Series[3].Points.Clear();

Runge_Kutt(Convert.ToDouble(textBox1.Text));
       for (int i = 0; i < t.Count; i++)
            {
chart1.Series[0].Points.AddXY(t[i], x1[i]);
chart1.Series[1].Points.AddXY(t[i], x2[i]);
chart1.Series[2].Points.AddXY(t[i], x3[i]);
chart1.Series[3].Points.AddXY(t[i], x4[i]);
```

```
            }
        }

        private void
button6_Click(object sender, EventArgs
e)
        {

chart1.Series[0].Points.Clear();

chart1.Series[1].Points.Clear();

chart1.Series[2].Points.Clear();

chart1.Series[3].Points.Clear();
        }

    }
}
```