

Министерство науки и высшего образования Российской Федерации
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИТМО»**

Факультет Систем Управления и Робототехники

Дисциплина: Методы машинного обучения в робототехнике

ОТЧЁТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ №4

Генетические алгоритмы

Выполнил студент

Топольницкий А.А.

Группа

R4134с

Преподаватель

Маргун А.А.

Санкт-Петербург

2023 г.

Задание 1.

1. Создать функцию 'PID_func.m', в которой будет проходить тестирование различных наборов коэффициентов ПИД-регулятора и вычисляться целевая функция J для каждого набора

`function J = PID_func(G,dt,params),`

где G – передаточная функция объекта управления в символическом виде, dt – интервал дискретизации, params – вектор из трех параметров регулятора K_p , K_i , K_d .

2. Задать передаточную функцию K для ПИД-регулятора в символическом виде. Для этого может быть использовано символическое задание оператора Лапласа: `s = tf('s')`.

3. Используя функции `series()` и `feedback` создать замкнутую систему с единичной отрицательной обратной связью для передаточных функций G и K.

4. Задать вектор значений времени t от 0 до 20 с в соответствии с интервалом дискретизации.

5. Вычислить реакцию выхода замкнутой системы y и значения управляющего сигнала u для задающего воздействия, равного 1. Для этого могут быть использованы функции `step()` и `lsim()`.

6. Рассчитать значение целевой функции на всем промежутке моделирования, просуммировав все взвешенные квадратичные значения ошибок и управления с учетом интервала дискретизации dt. Принять весовые коэффициенты: по ошибке регулирования $Q=1$, по управлению (penalty) $R = 0.001$.

7. Графики моделирования с итеративным обновлением при вызове функции PID_func могут быть получены с помощью кода:

```
step(ClosedLoop,t)
```

```
h = findobj(gcf,'type','line');
```

```
set(h,'linewidth',2);
```

```
drawnow
```

Выполнение задания 1.

```
1 function J = PID_func(G,dt,params)
2
3     s = tf('s'); % символическое задание оператора Лапласа
4     %передаточная функция ПИД-регулятора
5     K = params(1) + params(2)/s + params(3) * s / (1 + 0.001 * s);
6     % соединяем две передаточные функции последовательно в разомкнутый контур
7     Loop = series(K,G);
8     % замкнутая система с единичной отрицательной обратной связью
9     ClosedLoop = feedback(Loop, 1);
10    t = 0 : dt : 20;
11    [y, t] = step(ClosedLoop, t);
12
13    % моделирование отклика системы для задающего сигнала r=1
14    u = lsim(K, 1 - y, t);
15
16    Q = 1;
17    R = 0.001;
18    % вычисляем целевую функцию, суммируя все взвешенные квадраты ошибок и управления
19    J = dt * sum(Q * (1 - y(:)) .^ 2 + R * u(:) .^ 2);
20
21    % обновляемые графики для замкнутой системы
22    step(ClosedLoop,t)
23    h = findobj(gcf,'type','line');
24    set(h,'linewidth',2);
25    drawnow
```

Рисунок 1. Создание функции PID_func

Задание 2.

Создать функцию 'history_func.m' для сохранения результатов работы генетического алгоритма, используя следующий листинг:

```
function [state, options,optchanged] = history_func(options,state,flag)
```

```
persistent history
```

```
persistent cost
```

```
optchanged = false;
```

```
switch flag
```

```
    case 'init'
```

```
        history(:,1) = state.Population;
```

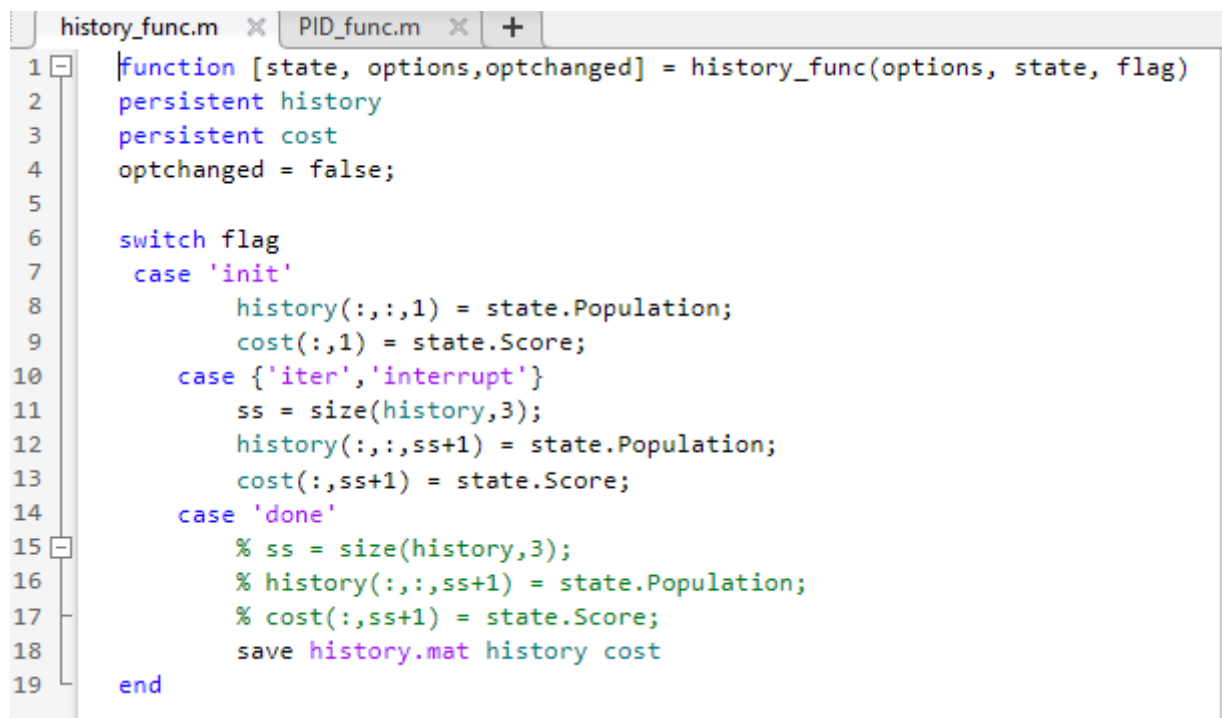
```
        cost(:,1) = state.Score;
```

```

    case {'iter','interrupt'}
        ss = size(history,3);
        history(:,ss+1) = state.Population;
        cost(:,ss+1) = state.Score;
    case 'done'
        ss = size(history,3);
        history(:,ss+1) = state.Population;
        cost(:,ss+1) = state.Score;
        save history.mat history cost
end

```

Выполнение задания 2.



```

history_func.m  PID_func.m  +
1  function [state, options,optchanged] = history_func(options, state, flag)
2  persistent history
3  persistent cost
4  optchanged = false;
5
6  switch flag
7  case 'init'
8      history(:,1) = state.Population;
9      cost(:,1) = state.Score;
10 case {'iter','interrupt'}
11     ss = size(history,3);
12     history(:,ss+1) = state.Population;
13     cost(:,ss+1) = state.Score;
14 case 'done'
15     % ss = size(history,3);
16     % history(:,ss+1) = state.Population;
17     % cost(:,ss+1) = state.Score;
18     save history.mat history cost
19 end

```

Рисунок 2. Создание функции history_func

Задание 3.

1. Создать файл 'GA_PID.m' для моделирования генетического алгоритма. Задать интервал дискретизации 0.001 с.
2. Количество индивидуумов в популяции (PopSize) и число поколений (MaxGenerations) принять равными 30 и 15, соответственно.
3. Сформировать символически передаточную функцию объекта управления

$$G(s) = \frac{1}{s(s^2 + s + 1)} \quad (1)$$

4. Определить оптимальные значения параметров регулятора для наименьшего значения целевой функции с помощью Matlab функции для вызова генетического алгоритма 'ga()'.

Выполнение задания 3.

Лабораторная №4

Выполнил: Топольницкий А.

Проверил: Маргун А.А.

```
clear all; close all; clc;
```

Задание интервала дискретизации, количества индивидуумов в популяции и число поколений. Задание передаточной функции

```
dt = 0.001; % интервал дискретизации
PopSize = 30;
MaxGenerations = 15;
s = tf('s'); % символическое задание оператора Лапласа
G = 1 / (s * (s * s + s + 1)); % передаточная функция объекта управления
```

Задание оптимальных значений параметров регулятора и вызов функции ga()

```
options = optimoptions(@ga,'PopulationSize',PopSize,'MaxGenerations',MaxGenerations,'OutputFcn',@history_func);
[x,fval] = ga(@(K)PID_func(G,dt,K),3,-eye(3),zeros(3,1),[],[],[],[],[],options);
```

Рисунок 3. Выполнение третьего задания

Построение графиков.

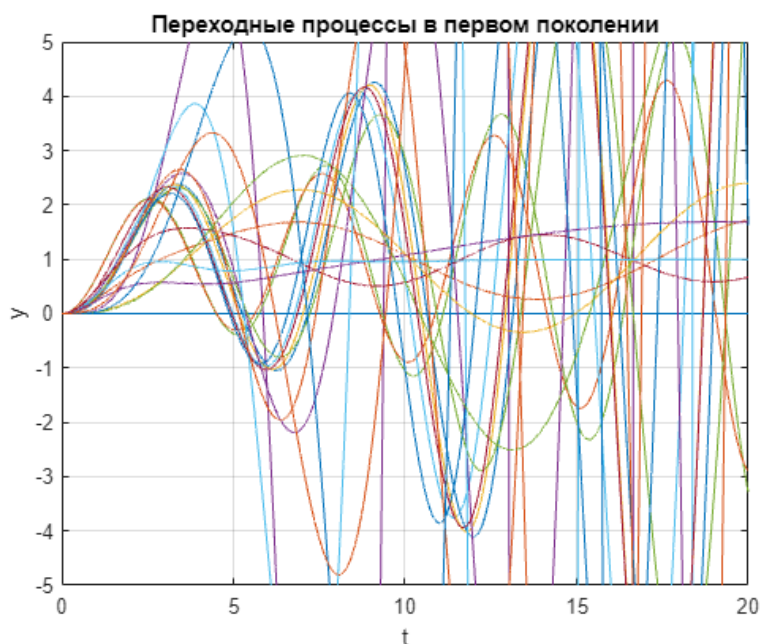


Рисунок 4. Переходные процессы в первом поколении

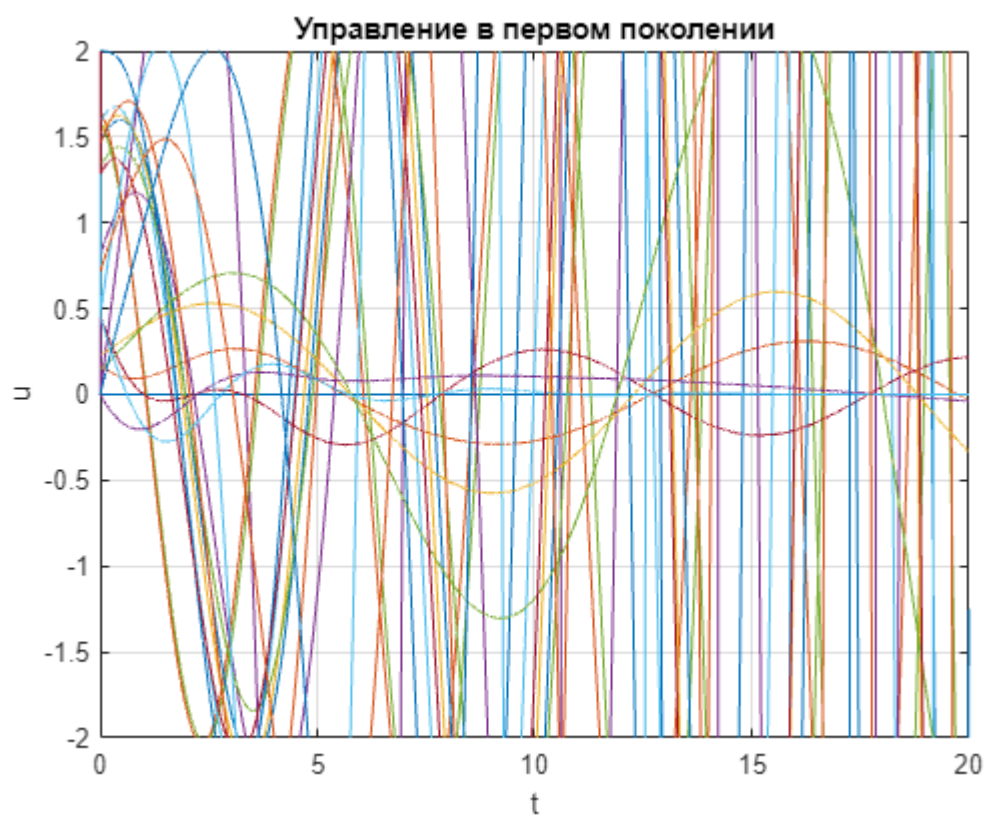


Рисунок 5. Управление в первом поколении

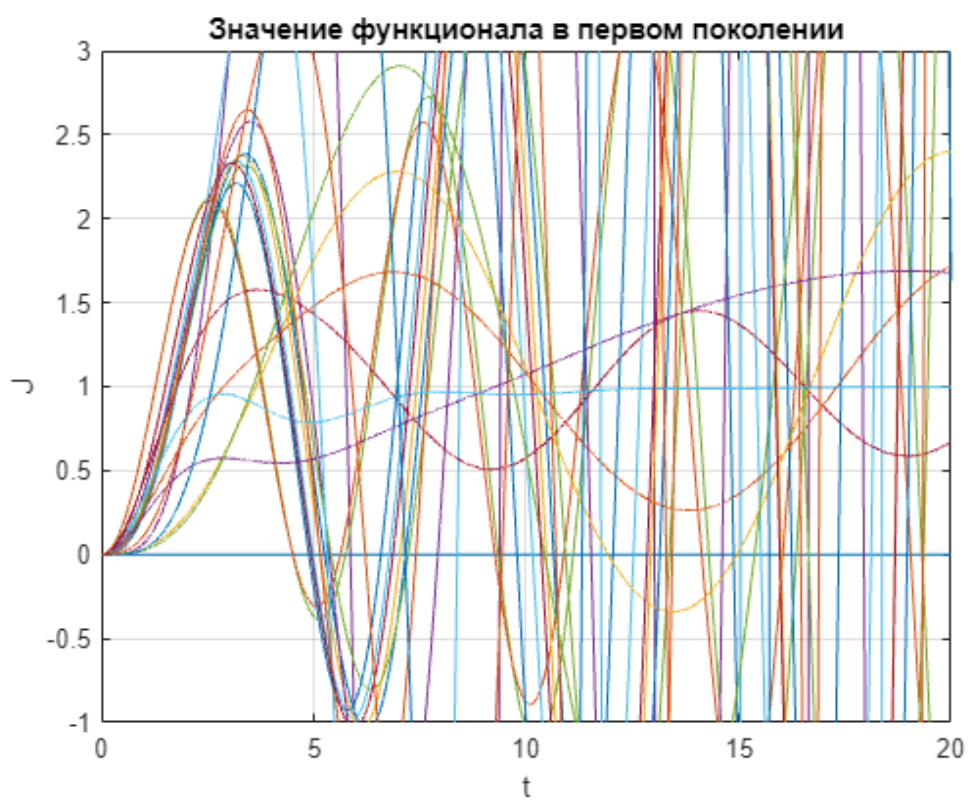


Рисунок 6. Поведение функционала в первом поколении

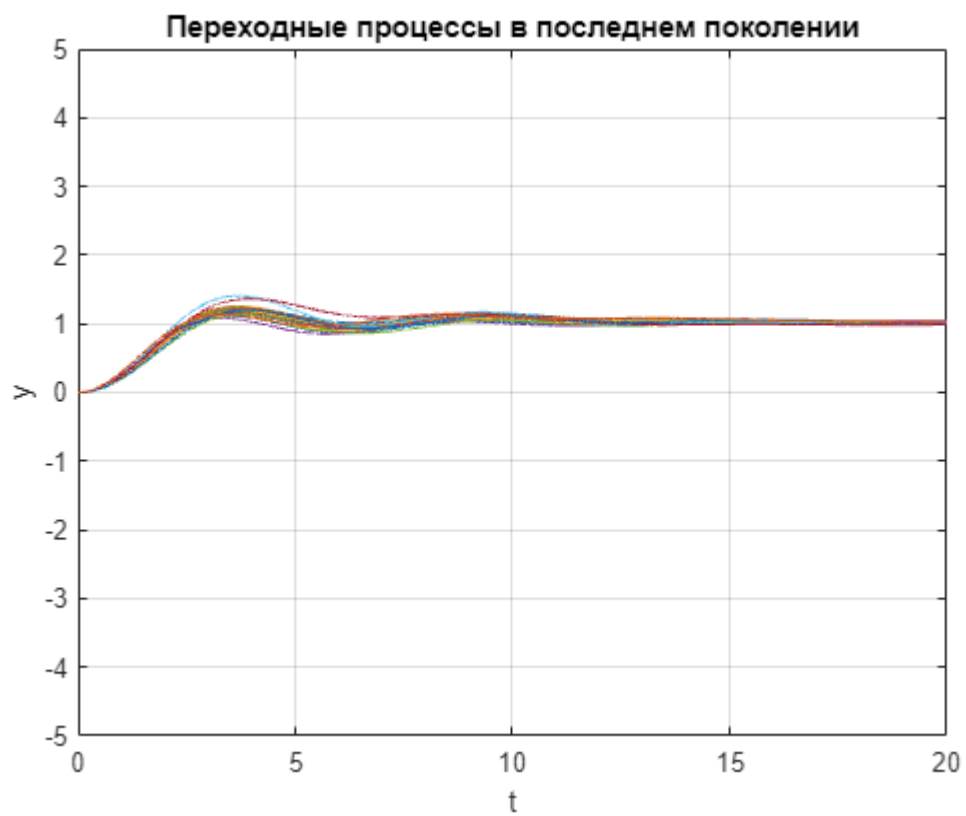


Рисунок 7. Переходные процессы в последнем поколении

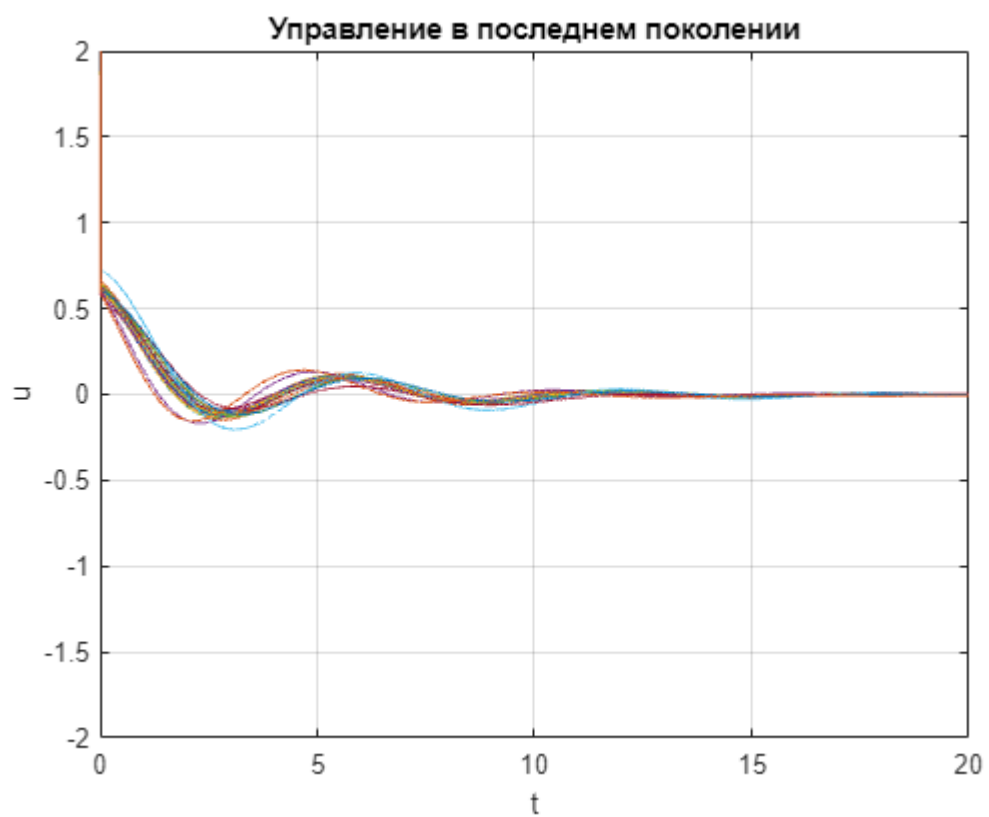


Рисунок 8. Управление в последнем поколении

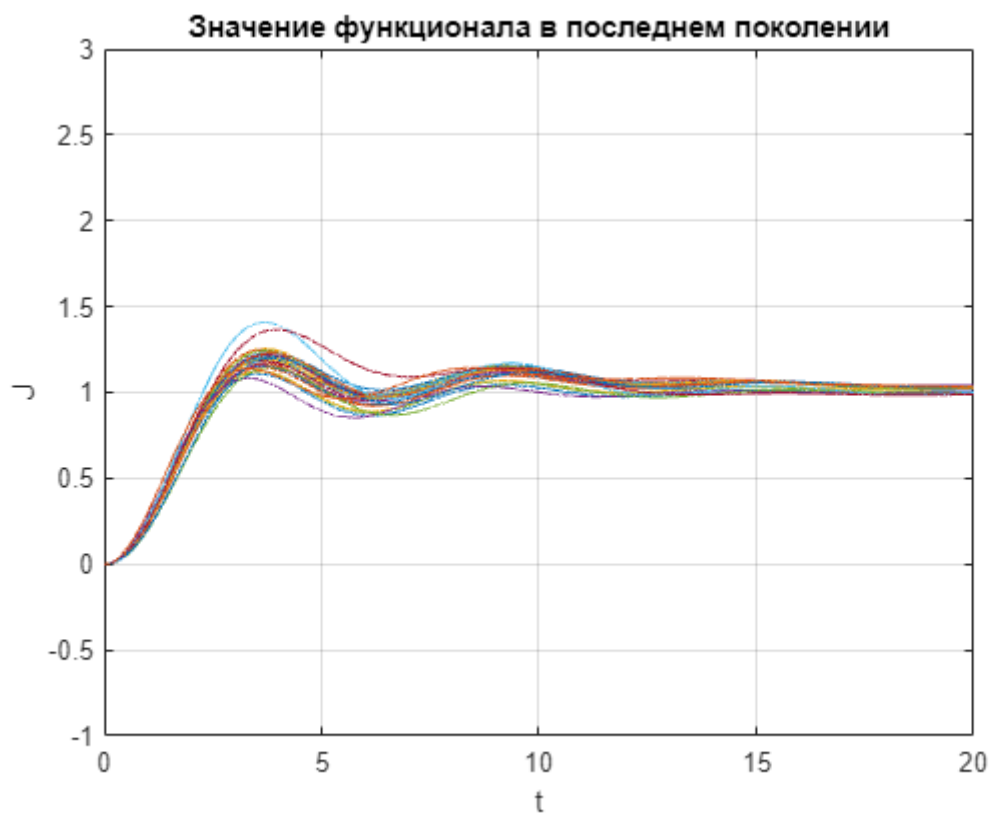


Рисунок 9. Поведение функционала в последнем поколении

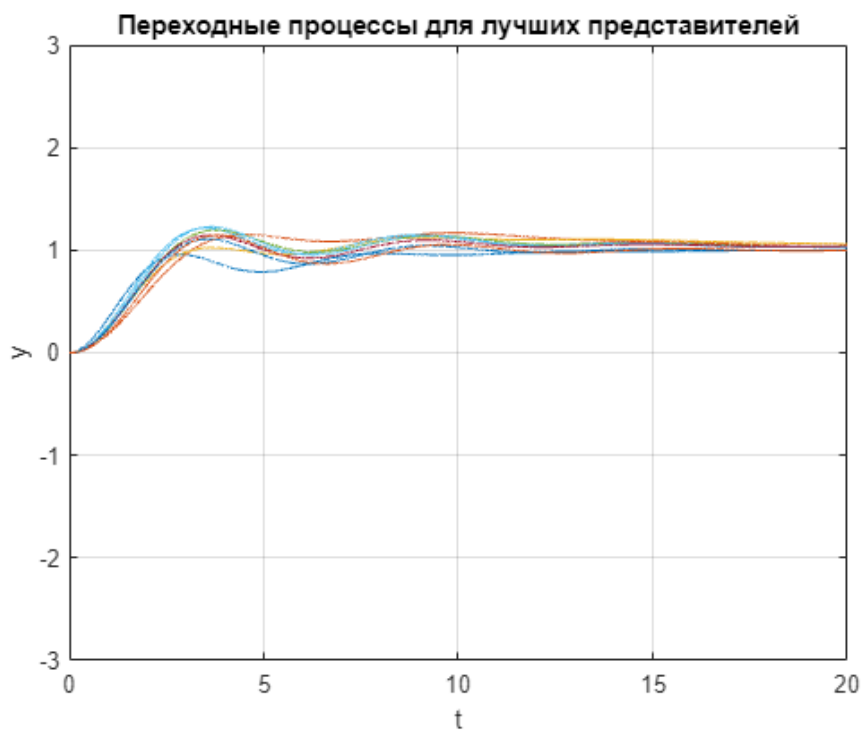


Рисунок 10. Переходные процессы для лучших представителей каждого поколения

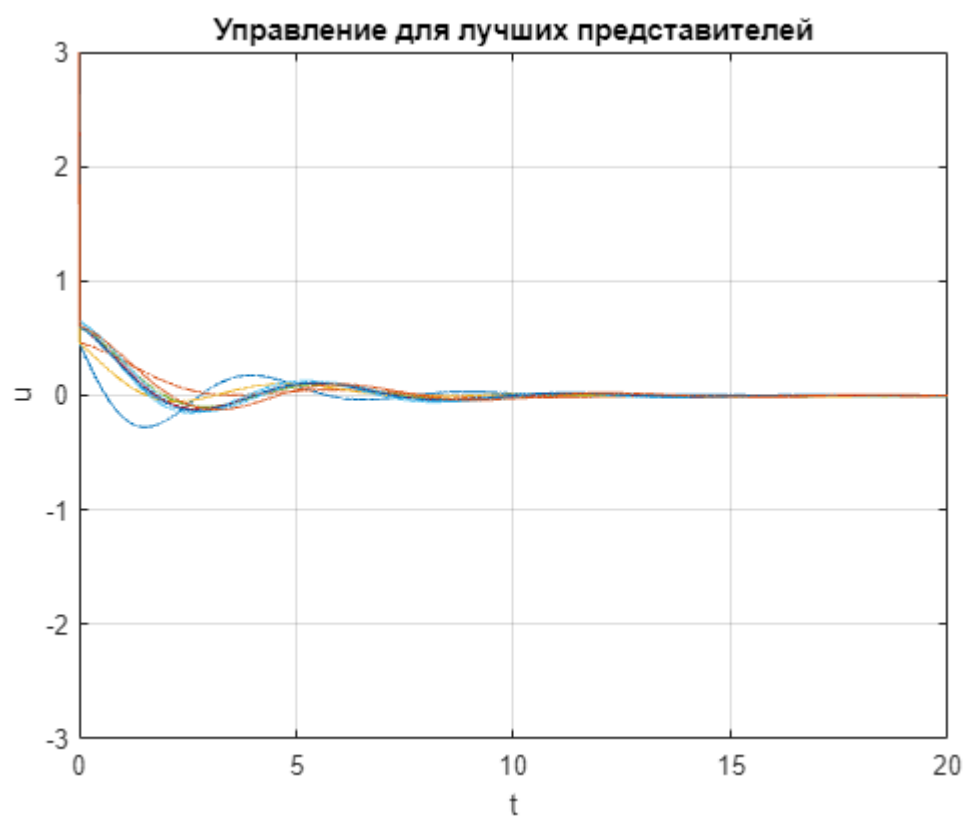


Рисунок 11. Управление для лучших представителей каждого поколения

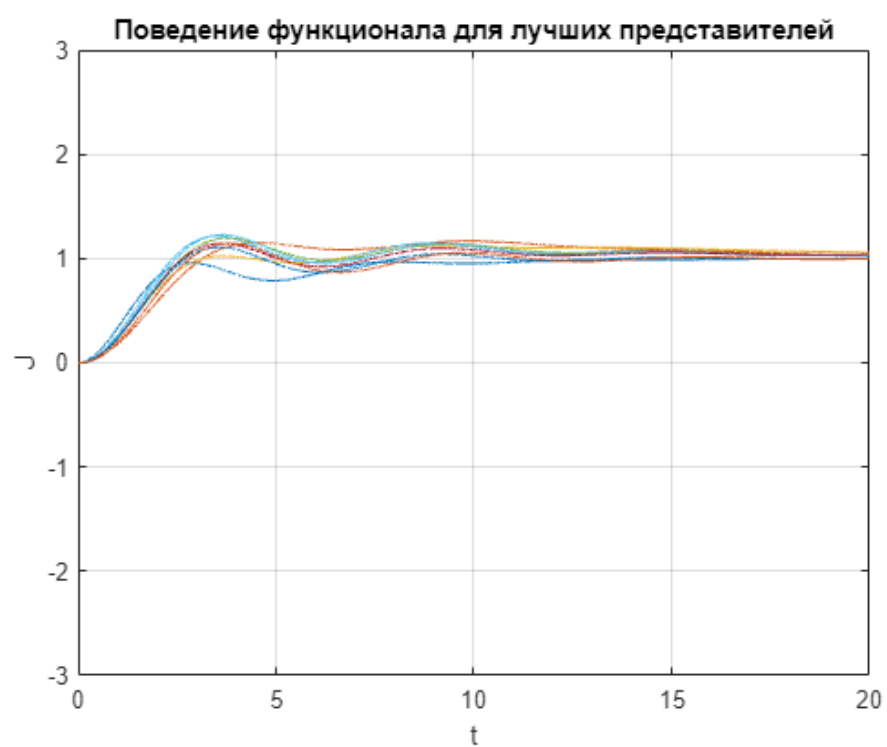


Рисунок 12. Поведение функционала для лучших представителей каждого поколения

Выводы по работе:

В ходе выполнения лабораторной работы была создана функция PID_func, в которой проходило тестирование различных наборов коэффициентов ПИД-регулятора и вычислялся функционал. Была также написана функцию history_func для сохранения результатов.

Было проведено моделирование генетического алгоритма с помощью встроенной в MATLAB функции ga() для заданной передаточной функции объекта управления. На основе моделирования были получены следующие графики:

- Управление, переходные процессы и поведение функционала для первого поколения. По ним можно сделать вывод, что всего несколько наборов ПИД-регулятора обеспечивают устойчивость системы, а остальные нет;
- Управление, переходные процессы и поведение функционала для последнего поколения. В последнем поколении больше индивидуумов обеспечивают устойчивость системы, однако можно заметить, что даже за 15 поколений не все индивидуумы из популяции смогли обеспечить устойчивость;
- Управление, переходные процессы и поведение функционала для лучших представителей каждого поколения. Можно заметить, что в каждом поколении был хотя бы один представитель, обеспечивающий устойчивость. При этом с ростом номера поколения нельзя сказать, что произошло кратное изменение диапазона, в котором происходит процесс сходимости. Минимальное значение функционала $J = 1.481$ при векторе параметров ПИД-регулятора $x = [0.6083 \quad 2.8843e - 4 \quad 0.3758]$.