

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

Соболев А.М.

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 Kp, Ki, Kd.

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
```

1. Создать функцию 'PID_func.m'

Листинг 1. Реализация PID_func

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

s = tf('s'); % символическое задание оператора Лапласа
%передаточная функция ПИД-регулятора
K = params(1) + params(2)/s + params(3) * s / (1 + 0.001 * s);
% соединяем две передаточные функции последовательно в разомкнутый контур
Loop = series(K,G);
% замкнутая система с единичной отрицательной обратной связью
ClosedLoop = feedback(Loop, 1);
t = 0 : dt : 20;
[y, t] = step(ClosedLoop, t);

% моделирование отклика системы для задающего сигнала r=1
u = lsim(K, 1 - y, t);

Q = 1;
R = 0.001;
% вычисляем целевую функцию, суммируя все взвешенные квадраты ошибок и
управления
J = dt * sum(Q * (1 - y(:)) .^ 2 + R * u(:) .^ 2);

% обновляемые графики для замкнутой системы
step(ClosedLoop,t)
h = findobj(gcf,'type','line');
set(h,'linewidth',2);
drawnow
```

2. Создать функцию 'history_func.m'

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

```

Листинг 2. Реализация PID_func

```

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'
    save history.mat history cost
end

```

3. Создать файл ‘GA_PID.m’

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

Генетические алгоритмы

Выполнил: Соболев А.М.

Проверил: Маргун А.А.

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

```
% интервал дискретизации
dt = 0.001;
PopSize = 30;
MaxGenerations = 15;

% оператора Лапласа в символьном представлении
s = tf('s');
% передаточная функция объекта управления
G = 1 / (s * (s * s + s + 1));
```

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);
```

Построение графиков.

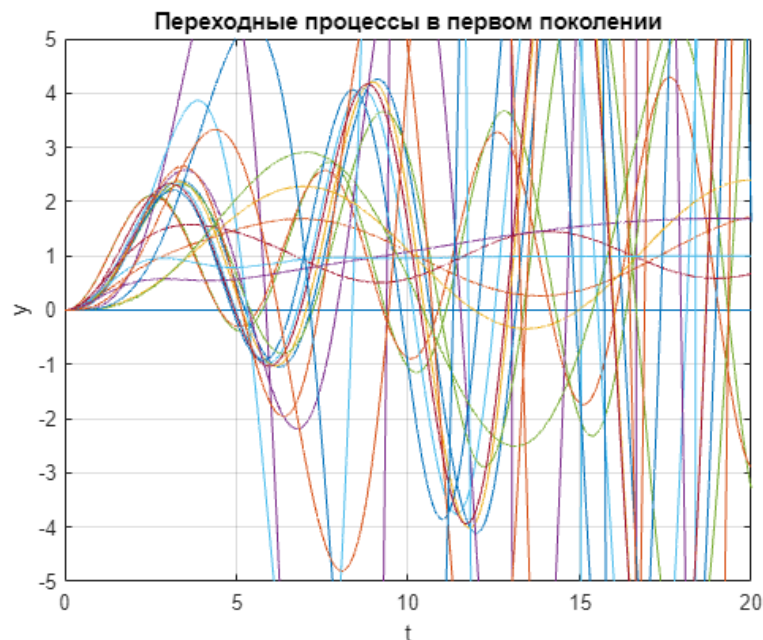


Рисунок 1. Переходные процессы в первом поколении

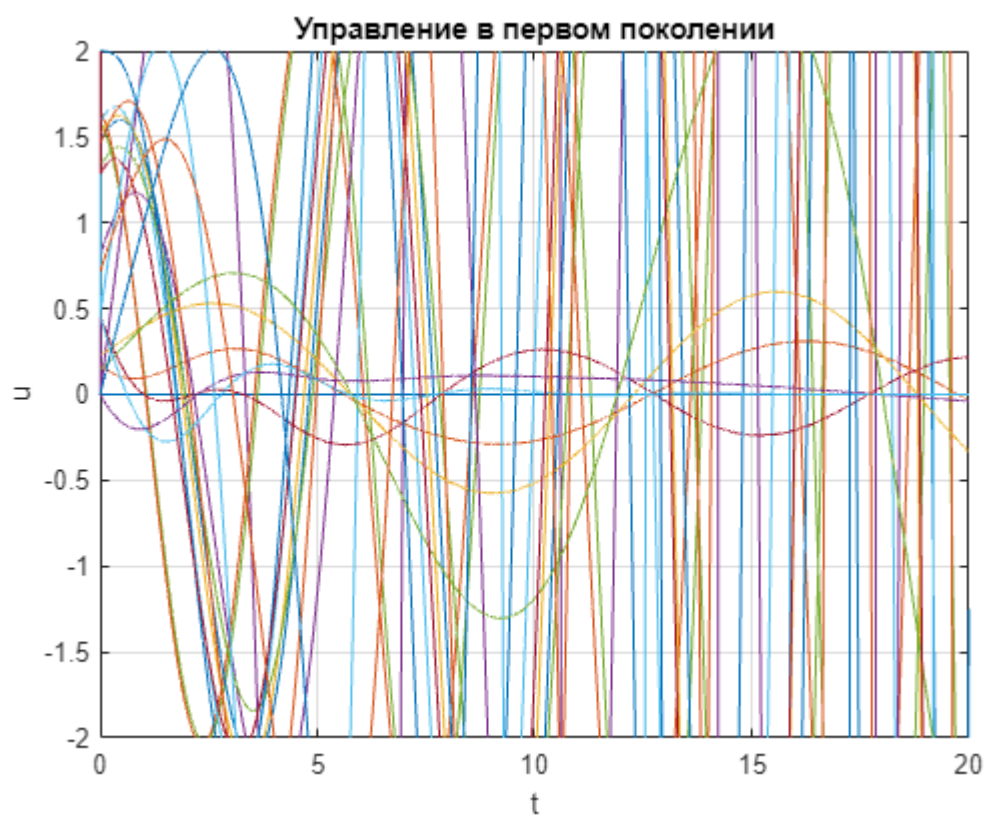


Рисунок 2. Управление в первом поколении

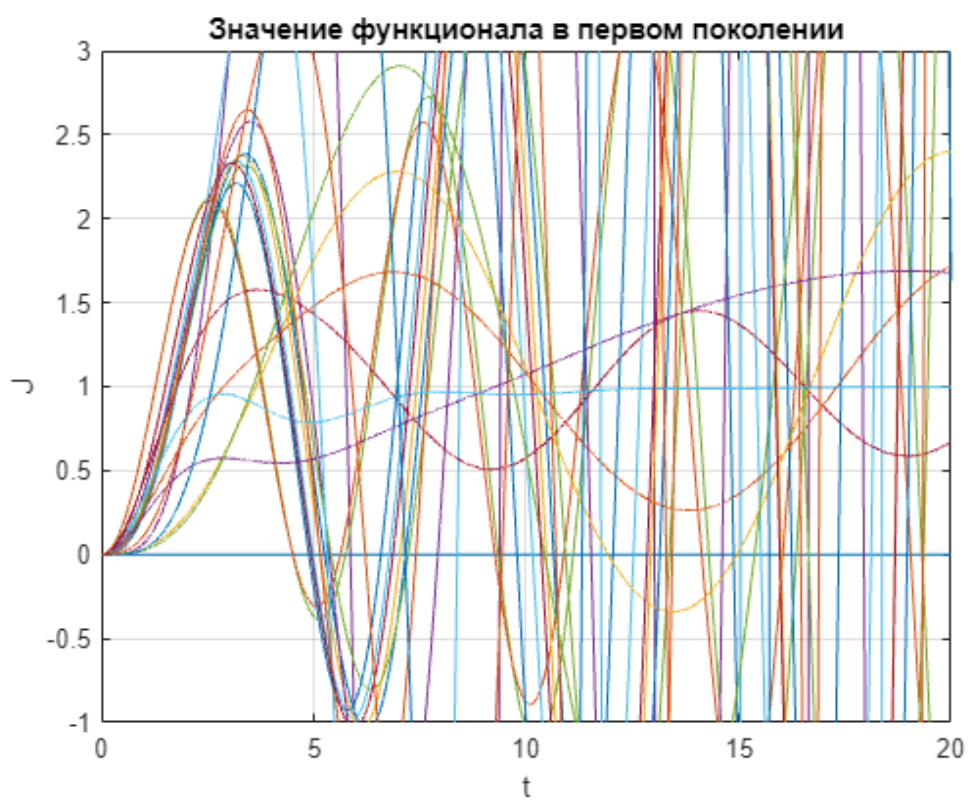


Рисунок 3. Поведение функционала в первом поколении

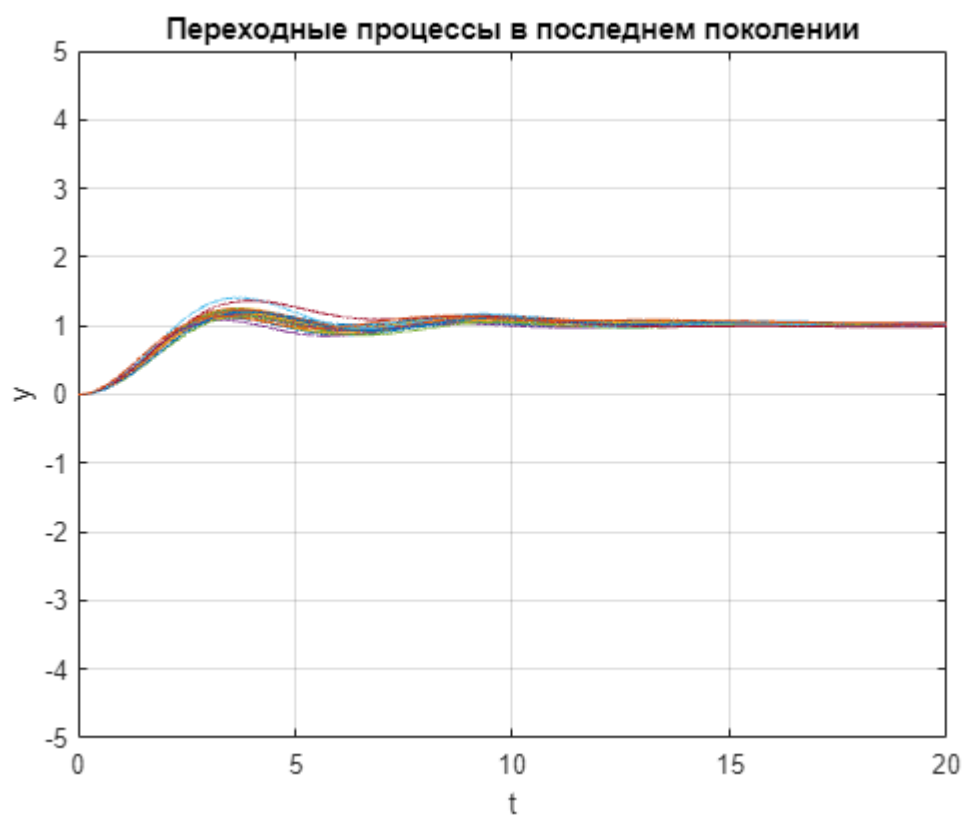


Рисунок 4. Переходные процессы в последнем поколении

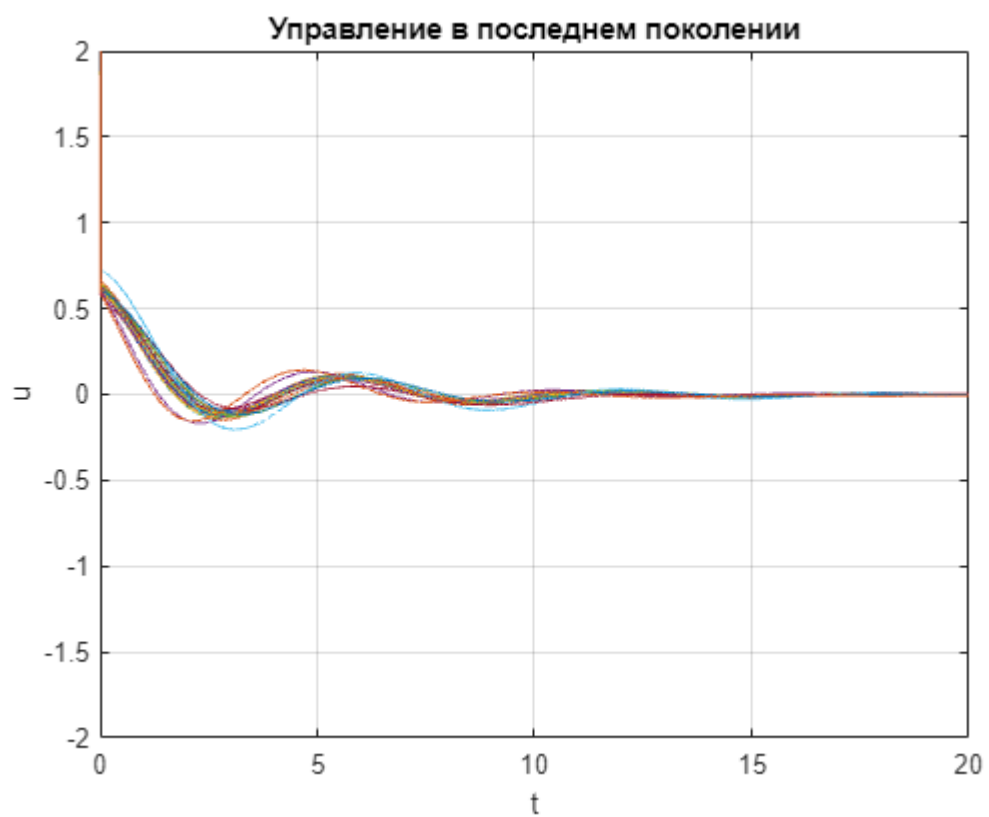


Рисунок 5. Управление в последнем поколении

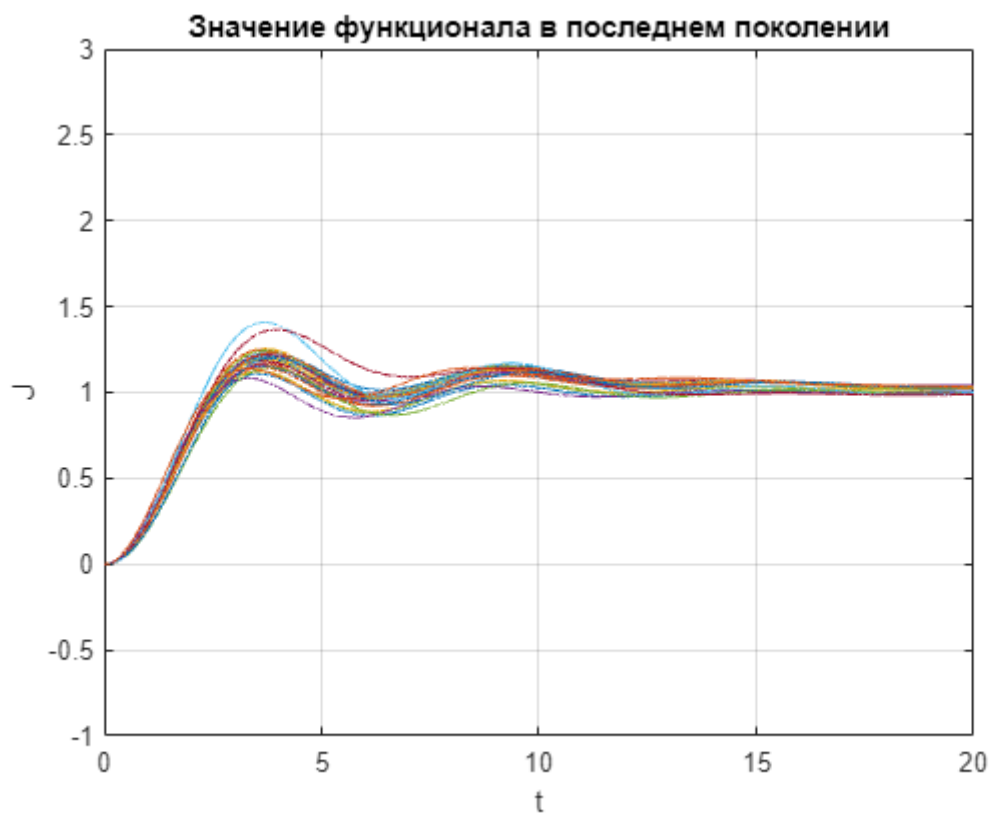


Рисунок 6. Поведение функционала в последнем поколении

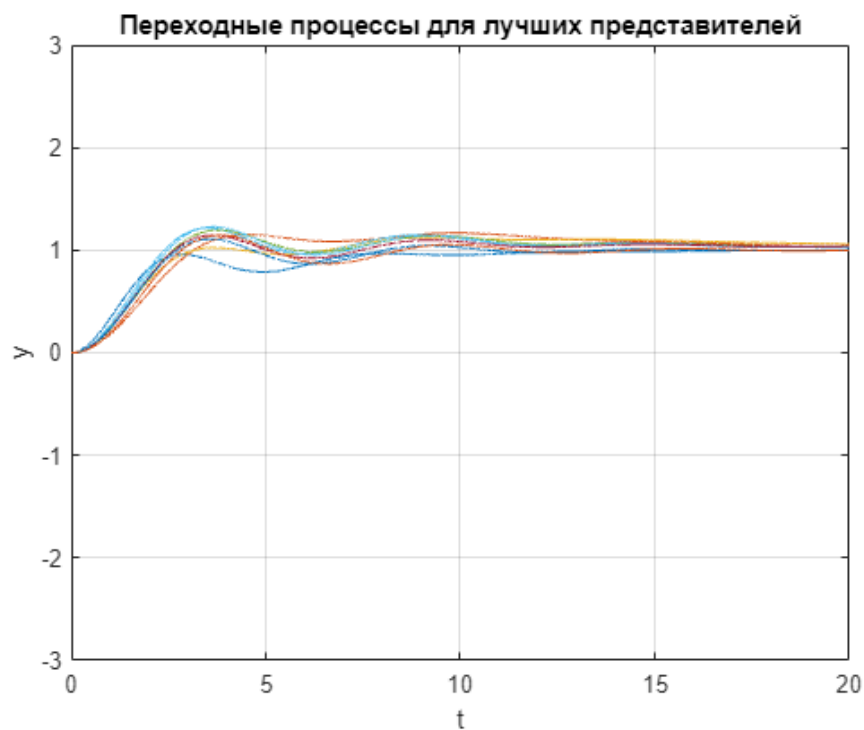


Рисунок 7. Переходные процессы для лучших представителей каждого поколения

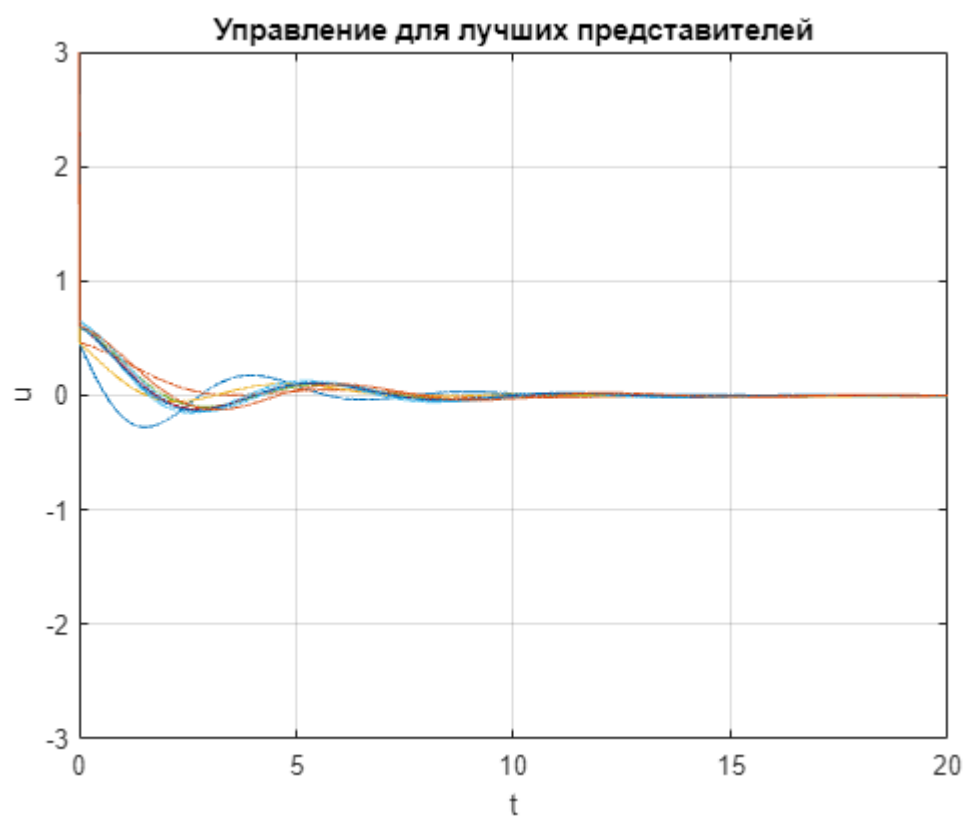


Рисунок 81. Управление для лучших представителей каждого поколения

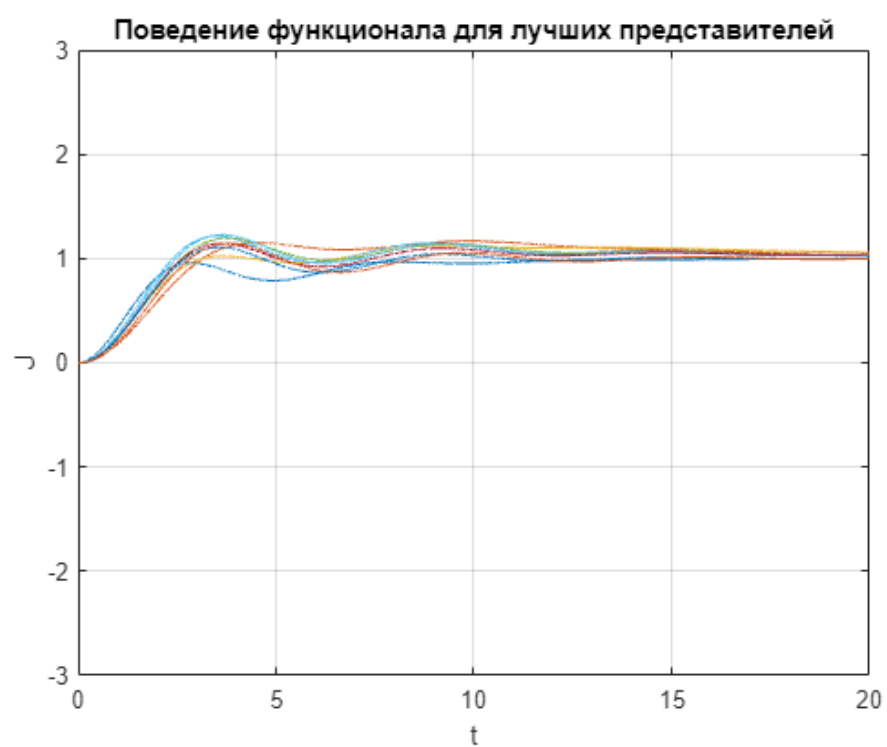


Рисунок 92. Поведение функционала для лучших представителей каждого поколения

Выводы по работе:

В ходе выполнения лабораторной работы была создана функция `PID_func`, которая отвечает за тестирование различных наборов коэффициентов ПИД-регулятора и вычисляет функционал. Написана функция `history_func` для сохранения результатов симуляции.

С помощью встроенной в MATLAB функции `ga()` для заданной передаточной функции объекта было проведено моделирование управления. На основе моделирования были получены следующие графики для разных поколений регуляторов:

Поколение №1: судя по графикам управления, переходных процессов и поведения функционала видно, что большинство регуляторов не способны обеспечить необходимое управление. Некоторые регуляторы способны дать удовлетворительный результат, но таких меньшинство.

Последнее поколение: графики представляют картину обратную ситуации в первом поколении – большинство “особей” обеспечивает устойчивость системы.

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 $x = [0.6083 \quad 2.8843e-4 \quad 0.3758]$.