**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САУ**

отчет

**по лабораторной работе № 6**

**по дисциплине «Интеллектуальные системы управления»**

Тема: **РЕШЕНИЕ ЗАДАЧ АППРОКСИМАЦИИ СРЕДСТВАМИ**

**НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ**

**8 (3) вариант**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 9492 |  | Ливаренко С.С. |
| Преподаватель |  | Порохненко К.А. |

Санкт-Петербург

2024

**Цель работы.**

Ознакомление с пакетом *Neural Networks Toolbox* (MATLAB), построение нейросетевого регулятора. Эффективность управления системы с нейросетевым регулятором проверяется по результатам моделирования в *Toolbox Simulink*.

**Основные сведения**

В состав пакета нейронных сетей (*Neural Networks Toolbox*) включено инструментальное средство организации диалога с пользователем *NNTool*, которое является графическим интерфейсом пользователя GUI (*Graphic User Interface*) системы Matlab.

Вызов GUI-интерфейса *NNTool* осуществляется командой *nntool* из командной строки. После вызова появляется окно *Network/Data Manager* (Управление сетью/данными), которое содержит области:

* *Inputs* - последовательность входов (заданные входы);
* *Targets* - последовательность целей (заданные выходы);
* *Input Delay States* -начальные условия линии задержки входа;
* *Networks* - (список нейронных сетей);
* *Outputs* - последовательность выходов (действительные выходы);
* *Errors* - последовательность ошибок обучения;
* *Layer Delay States* - начальные условия линии задержки слоя.

В поле Network and Data содержатся кнопки:

* *Help* - вызов окна подсказки;
* *New Data* - вызов окна формирования данных;
* *New Network* - вызов окна создания новой нейронной сети;
* *Impor*t - вызов окна для извлечения или загрузки данных;
* *Export* - вызов окна для передачи или загрузки данных в файл.

**Результаты работы**

1. Вызов GUI-интерфейса *NNTool* иформирование последовательностей входа и целей. На рисунке 1 представлен процесс добавления матриц входов и выходов сети.

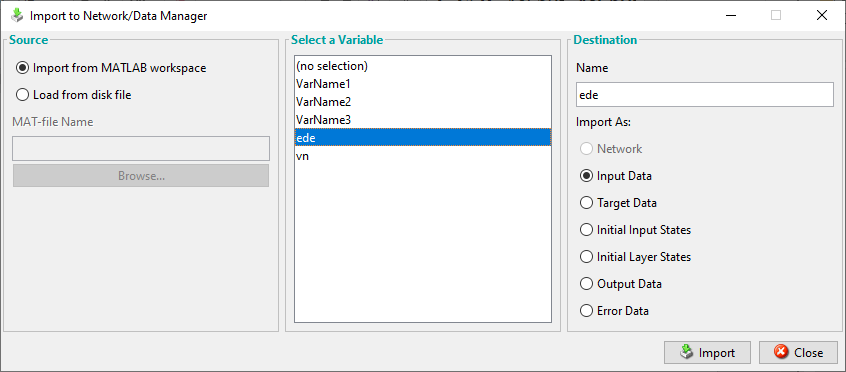


Рисунок 1 – Окно интерфейса *NNToolи (Data Manager)*

1. Создание нейронной сети. На рисунке 2 представлен процесс непосредственно формирования нейронной сети.

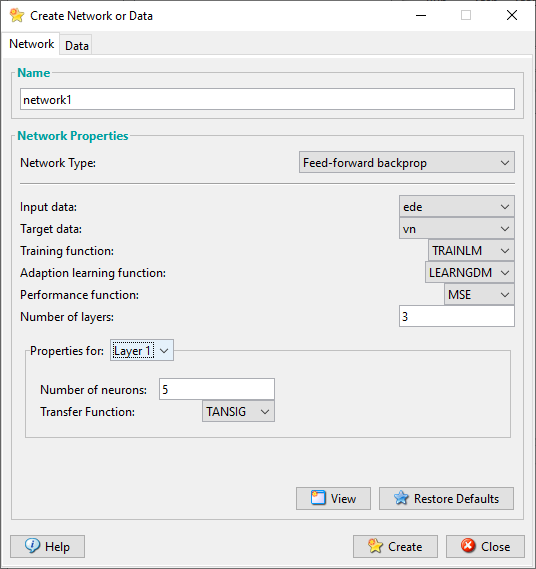


Рисунок 2 – Окно create Network or Data

В поле свойства слоев (*Properties for Layer*) задано количество нейронов (*Number of neurons*) и функции активации (*Transfer Function*) согласно варианту: 1-й слой - 5; *tansig*; 2-й слой - 3; *tansig*; 3-й слой -1 *purelin*.

1. Инициализация сети

В окне Network/Data Manager, в области *Networks* появилось имя новой созданной сети *Network*1 (рисунок 3).

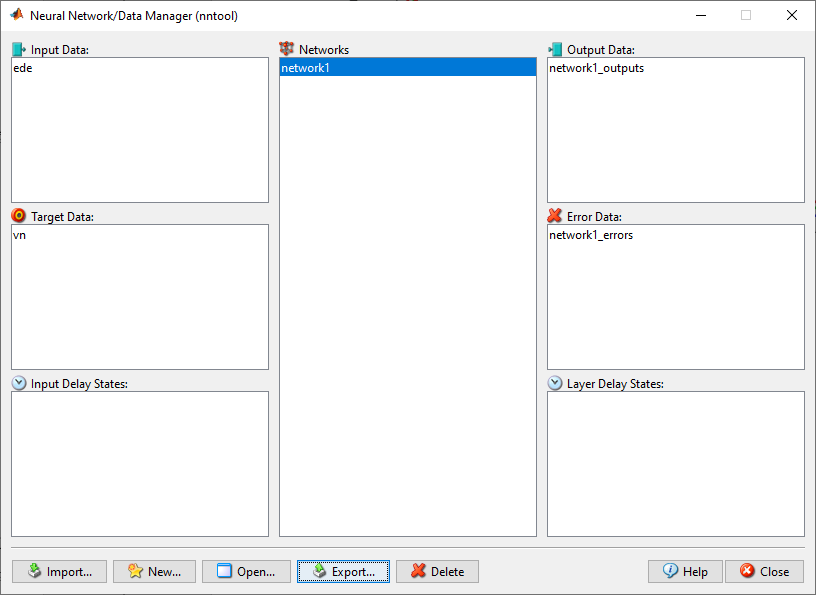


Рисунок 3 – Окно Network/Data Manager

Для инициализации сети была открыта диалоговая панель Network: *network*1 (рисунок 4).

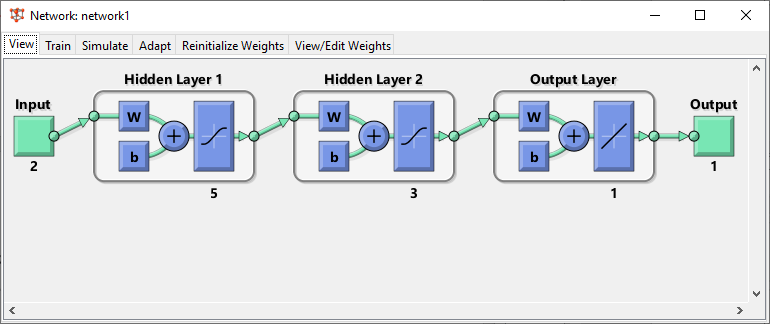


Рисунок 4 – Диалоговая панель сети network1

На рисунке 5 показана установка диапазонов переменных и инициализация весов.

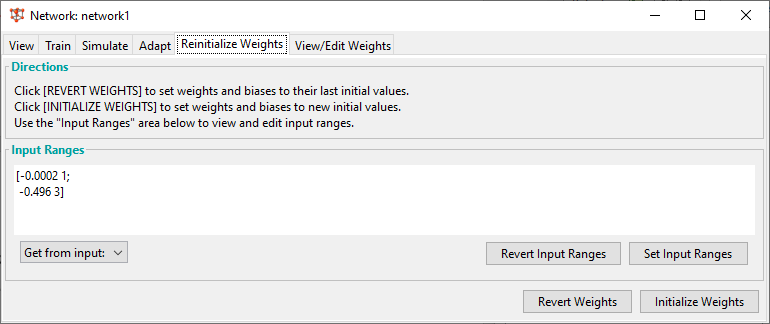


Рисунок 5 – Инициализация весов и диапазонов

1. Обучение сети

Для обучения сети в закладке Train установлены имена последовательностей входа *ede* и цели *vn* (рисунок 6)*,* а также на закладке *Training Parameters* значения параметров процедуры обучения*.* Число циклов обучения (epochs) установлено 500 (рисунок 7).

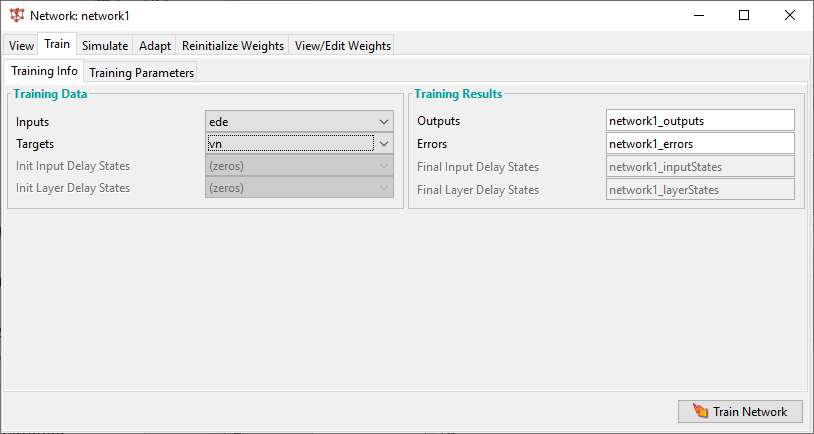


Рисунок 6 – Окно закладки Train

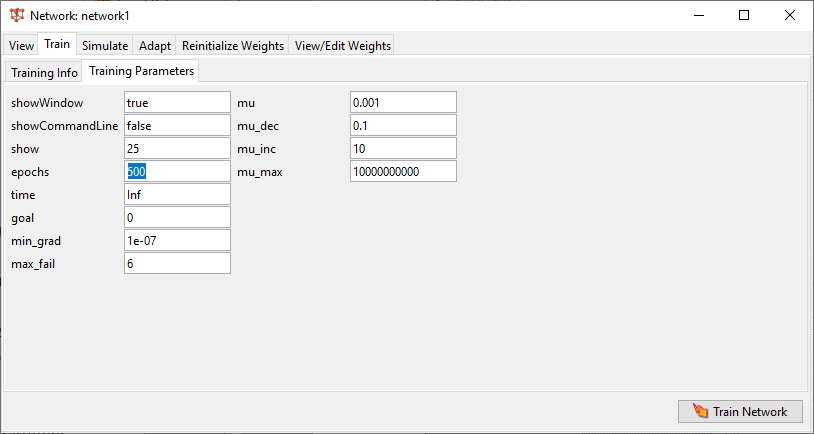


Рисунок 7 – Окно закладки *Training Parameters*

После нажатия соответствующей кнопки происходит обучение нейросети. Его результаты представлены на рисунке 8.

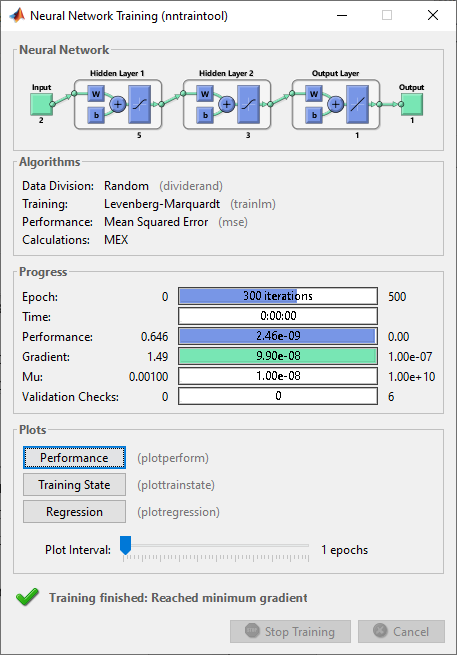


Рисунок 8 – Окно закладки *Training Parameters*

Как видно из рисунка, обучение завершено уже через 300 из 500 итераций.

1. Сравнение качества переходных процессов в системе управления с линейным и нейросетевым регуляторами.

На рисунках 9 - 12 изображены графики переходного процесса двух регуляторов.

1. в номинальном режиме

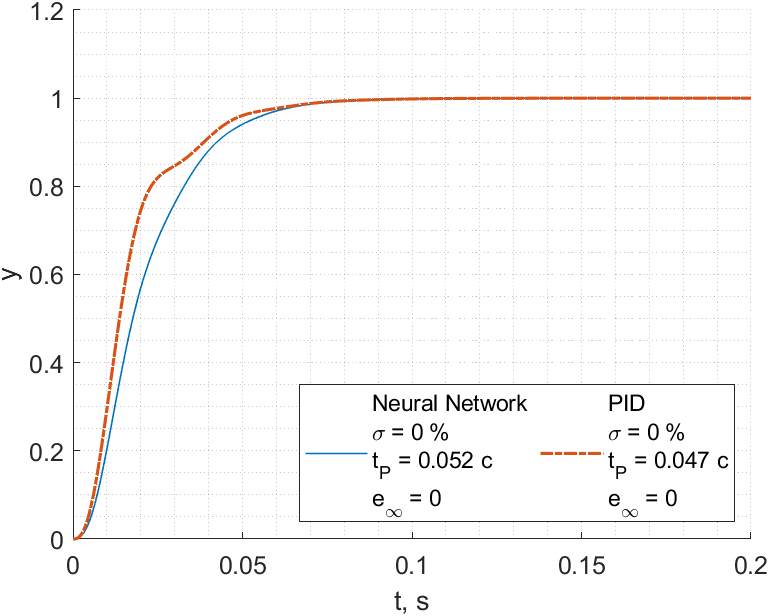


Рисунок 9 – Сравнение переходных процессов нейросетевого и ПД-регулятора в номинальном режиме

1. при наличии ограниченных изменений параметров (k3 =125; k4=200; k5=500);

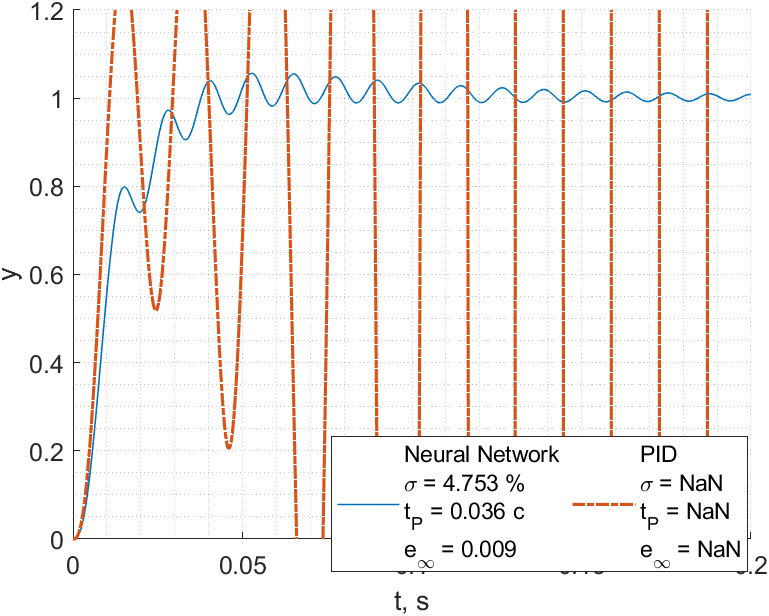


Рисунок 10 – Сравнение переходных процессов нейросетевого и ПД-регулятора при наличии ограниченных изменений параметров

1. при введении звена насыщением ± 0.15 для номинальных параметров;

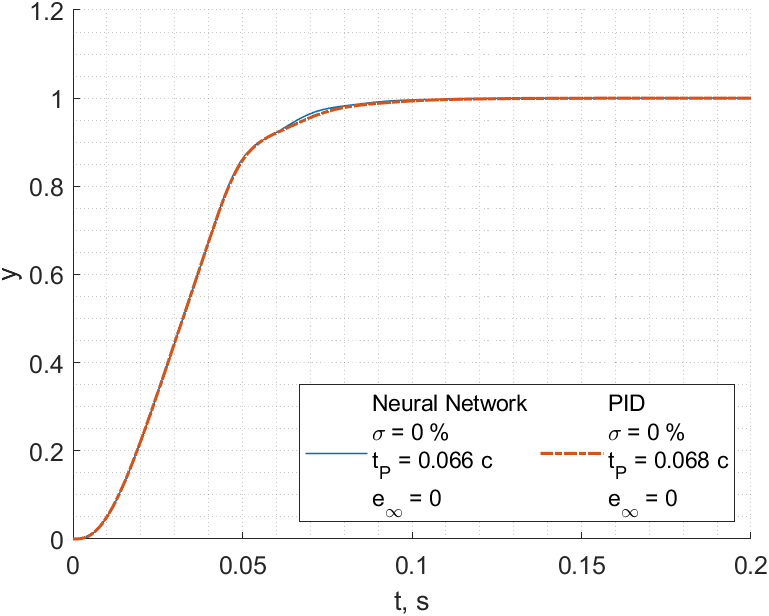


Рисунок 11 – Сравнение переходных процессов нейросетевого и ПД-регулятора при введении звена с насыщением

1. при воздействии внешних возмущений.

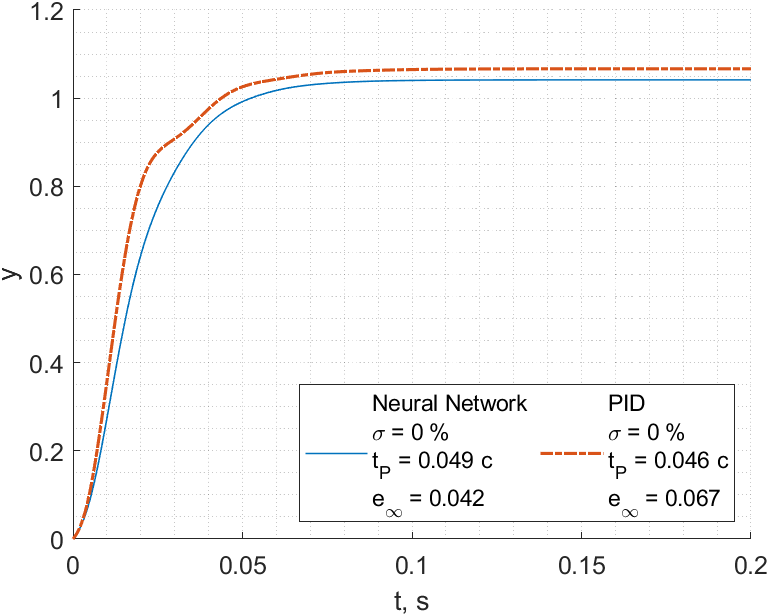


Рисунок 12 – Сравнение переходных процессов нейросетевого и ПД-регулятора при воздействии внешних возмущений

В таблице 1 приведено сравнение показателей качества двух регуляторов.

Таблица 1 – Таблица сравнения показателей качества нейросетевого и ПД-регулятора при различных режимах.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Режим | Нейросетевой регулятор | | | Линейный регулятор ПД | | |
|  |  |  |  |  |  |
| Номинальный режим | 0 | 0.052 | 0 | 0 | 0.047 | 0 |
| Ограниченные изменения параметров | 4.753 | 0.036 | 0.009 | NaN | NaN | NaN |
| Введение звена насыщения | 0 | 0.066 | 0 | 0 | 0.068 | 0 |
| Внешние возмущения | 0 | 0.049 | 0.042 | 0 | 0.046 | 0.067 |

Как видно из таблицы, во всех случаях система с нейросетевым регулятором сохраняет свою устойчивость и во всех экспериментах, кроме номинального режима, имеет лучшие показатели качества.

**Вывод.**

В ходе лабораторной работы был изучен пакет *Neural Networks Toolbox* (MATLAB), построен нейросетевой регулятор с заданными параметрами слоёв. Полученный регулятор, благодаря своей нелинейной природе позволил достичь лучших показателей качества, чем ПД-регулятор во всех экспериментах с изменениями исходной системы.

**Приложение А**

**Скрипт лабораторной работы**

clc, clear, close all

warning('off', 'all')

warning

set(0, 'DefaultAxesFontSize', 12);

set(0, 'fixedWidthFontName', 'Times New Roman');

% set(0, 'defaultLegendInterpreter', 'tex');

% set(0, 'defaulttextinterpreter', 'tex');

mkdir lr2

sim\_t = 0.2;

K3\_(1) = 250; K3\_(2) = 125;

K4\_(1) = 100; K4\_(2) = 200;

K5\_(1) = 150; K5\_(2) = 500;

global fuzzy\_reg, global fis

% fuzzy\_reg = "lab\_2\_1.fis";

model\_name = "lab\_6";

open(model\_name + ".slx")

disturbance = 0;

text1 = ["Neural Network", "PID"];

fun1 = @(i) set\_param(model\_name + "/Switch", 'sw', string(2-i));

%% пункт

K3 = K3\_(1); K4 = K4\_(1); K5 = K5\_(1);

limit = inf;

plot\_graphs(model\_name, fun1, text1)

%%

K3 = K3\_(2); K4 = K4\_(2); K5 = K5\_(2);

limit = inf;

plot\_graphs(model\_name, fun1, text1)

%%

limit = 0.15;

K3 = K3\_(1); K4 = K4\_(1); K5 = K5\_(1);

plot\_graphs(model\_name, fun1, text1)

%%

disturbance = 10;

K3 = K3\_(1); K4 = K4\_(1); K5 = K5\_(1);

limit = inf;

plot\_graphs(model\_name, fun1, text1)

return