

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра механики, мехатроники и робототехники



СПЕЦГЛАВЫ ТЕОРИИ УПРАВЛЕНИЯ МЕХАТРОННЫМИ СИСТЕМАМИ

методические указания по выполнению практических и
самостоятельной работ
для студентов направления
15.03.06 «Мехатроника и робототехника»

Курск 2020

УДК 681.323

Составители: П.А. Безмен, В.В. Бартенев

Рецензент:

Кандидат технических наук, доцент Юго-Западного государственного университета *Е.Н. Политов*

Спецглавы теории управления мехатронными системами: методические указания по выполнению практических и самостоятельной работ для студентов направления 15.03.06 «Мехатроника и робототехника» / Юго-Зап. гос. ун-т; сост. П.А. Безмен, В.В. Бартенев. Курск, 2020. 50 с.

Изложены теоретические предпосылки, задания и примеры решения задач по дисциплине «Спецглавы теории управления мехатронными системами».

Методические указания предназначены для студентов направления 15.03.06 «Мехатроника и робототехника» всех форм обучения.

Текст печатается в авторской редакции

Подписано в печать 21.02.20.. Формат 60x84 1/16.

Усл.печ.л. 3,1 Уч.-изд.л. 2,9 Тираж 100 экз. Заказ 134 Бесплатно.

Юго-Западный государственный университет.

305040 Курск, ул. 50 лет Октября, 94.

Содержание

Планируемые результаты обучения по дисциплине «Спецглавы теории управления мехатронными системами»	4
1. Оптимальный синтез САУ с применением модифицированного генетического алгоритма	5
2. Идентификация элементов САУ на базе синтеза искусственной нейронной сети прямого распространения	23
3. Синтез нечеткого регулятора САУ, реализующего алгоритм Мамдани.....	35
4. Идентификация элементов САУ на базе синтеза нечеткой нейронной сети типа ANFIS.....	42
Рекомендуемая литература	50

Планируемые результаты обучения по дисциплине «Спецглавы теории управления мехатронными системами»

Целью является формирование у обучающихся знаний, умений и приобретение опыта оптимального синтеза систем автоматического управления мехатронными системами, а также подготовка студентов к решению конкретных задач научно-исследовательской и производственно-технологической деятельности.

Задачи дисциплины

- получение представлений о назначении, основных особенностях и структурах современных алгоритмов управления,
- изучение математического описания алгоритмов систем управления, методов анализа и синтеза цифровых систем управления,
- изучение современных подходов к формированию программного обеспечения систем управления движением в реальном времени;
- изучение методов построения современных систем адаптивного, оптимального и интеллектуального компьютерного управления.

Процесс изучения дисциплины «Спецглавы теории управления мехатронными системами» направлен на формирование следующих компетенций

- ПК-1 способностью составлять математические модели мехатронных и робототехнических систем, их подсистем и отдельных элементов и модулей, включая информационные, электромеханические, гидравлические, электрогидравлические, электронные устройства и средства вычислительной техники
- ПК-11 способностью производить расчеты и проектирование отдельных устройств и подсистем мехатронных и робототехнических систем с использованием стандартных исполнительных и управляющих устройств, средств автоматики, измерительной и вычислительной техники в соответствии с техническим заданием

1. Оптимальный синтез САУ с применением модифицированного генетического алгоритма

Цель занятия

Изучение способа оптимального синтеза САУ, основанного на применении генетического алгоритма.

Задание

1. Изучить материал теоретической части.
2. Реализовать математическую модель САУ в коде функций цели.
3. Задать два-три критерия оптимизации САУ, учесть их в реализациях функций цели.
4. Создать динамически присоединяемые библиотеки, скомпилировав в них программный код функций цели.
5. Оптимизировать параметры САУ по заданным критериям с использованием программного средства GMR.
6. Оценить результаты оптимизации САУ, представив результаты математического моделирования (в виде графиков, скриншотов и т.п.).

Отчет должен содержать цель работы, задание и практическую часть, включающую:

1. описание моделируемой САУ;
2. описание критериев оптимизации САУ;
3. скриншоты главного окна программного средства GMR, отображающего начальные условия и результаты оптимизации;
4. листинги файлов реализации, содержащих описание целевых функций;
5. результаты моделирования САУ с оптимизированными параметрами.
6. выводы по результатам проведенных исследований.

Теоретическая часть

Генетические алгоритмы (ГА) применяются для решения многих конкретных научных и технических проблем, но самое популярное приложение генетических алгоритмов – оптимизация многопараметрических функций. Эффективность ГА определяется их способностью манипулировать одновременно многими параметрами, что является крайне необходимым в решении многих прикладных задач, включая проектирование интегральных схем, настройку параметров алгоритмов, поиск решений нелинейных дифференциальных уравнений и т. д.

Эффективность применения ГА зависит от характеристик задачи. Если пространство поиска в задаче небольшое, то решение можно найти методом полного перебора, и применение ГА не требуется. Если пространство гладкое и унимодальное, то любой градиентный алгоритм будет эффективнее ГА. Если существует дополнительная информация о пространстве поиска, то методы поиска, использующие эвристики, часто превосходят ГА. Область применения ГА – задачи с большим пространством поиска решений и отсутствием эвристической информации.

Кроме того, эффективность ГА зависит от характеристик метода кодировки решений, выбора генетических операторов и значений инициализирующих параметров.

Результаты анализа возможностей применения ГА для решения задач оптимизации параметров элементов систем автоматического управления (САУ) проведены в [2].

Опыт многих исследователей алгоритмов управления показал, что для простых одноконтурных систем автоматического управления (САУ) с линейными регуляторами задачи оптимизации, как правило, являются одноэкстремальными. Однако для сложных многоконтурных САУ и систем управления с нейроконтроллерами характерно наряду с глобальным наличие большого числа локальных экстремумов. Кроме того, локальные экстремумы появляются и при введении ограничений на пространство поиска.

Для решения одноэкстремальных задач оптимизации существует достаточное число градиентных и численных методов. Одним из них является метод деформируемого многогранника Нелдера-Мида. При оптимизации одноконтурной САУ с ПИ-регулятором примене-

ние метода позволяет устойчиво находить оптимальные значения настроечных параметров K_p и K_i для функции цели вида

$$F(K_p, K_i, \psi, t) = I_{m,y}(1 + a|\psi_0 - \psi|) \rightarrow \min ,$$

где

$$I_{m,y} = \int_0^{t_p} |y(t)| dt$$

– интеграл по модулю регулируемой величины $y(t)$ на интервале времени переходного процесса t_p ; ψ_0, ψ – соответственно, заданная и текущая степень затухания переходного процесса регулирования; a – масштабный коэффициент, учитывающий вес штрафной функции.

Применение метода деформируемого многогранника для оптимизации двухконтурной САУ с дифференциатором и САУ с нейроконтроллерами различной структуры приводит к неоднозначности решения. В каждом случае результаты зависят от выбранных начальных координат. Из этого следует вывод о многоэкстремальности подобных задач, и для их решения требуются методы глобальной оптимизации.

В настоящее время наиболее предпочтительными методами многоэкстремальной оптимизации являются ГА, реализующие постулаты теории эволюции и опыта селекции растений и животных. Стратегия поиска оптимального решения в ГА опирается на гипотезу селекции: чем выше приспособленность особи, тем выше вероятность того, что у потомков, полученных с ее участием, признаки, определяющие приспособленность, будут выражены еще сильнее.

Если принять, что каждая особь популяции является точкой в координатном пространстве оптимизационной задачи $X_i[x_{1i}, x_{2i}, \dots, x_{ni}]$, а приспособленность особи – соответствующим значением функции цели $f(X_i)$, то популяцию особей можно рассматривать как множество координатных точек в пространстве, а процесс эволюции – как движение этих точек в сторону оптимальных значений целевой функции.

Следует отметить, что классический ГА находит глобальный экстремум в вероятностном смысле. И эта вероятность прямо зависит от числа особей в популяции. При оптимизации сложных многоконтурных и многосвязных систем регулирования и аналогичных систем с нейроконтроллерами ГА (в частности, диплоидная версия ГА) с достаточно высокой вероятностью находят глобальный экстремум. Однако вычисление функции цели, как правило, требует значитель-

ных вычислительных ресурсов, что существенно сказывается на общем времени расчета решения.

Особенностью ГА является то, что ни один из генетических операторов (кроссовер, мутация, инверсия) в процессе генерирования потомков не опирается на знание локального рельефа поверхности отклика функции цели. Формирование потомков генетическими операторами происходит случайным образом, и поэтому нет гарантии, что найденные решения будут лучше родительских. Следовательно, в процессе эволюции встречается немалое число «неудачных» потомков, которые увеличивают число обращений к функции цели и, тем самым, увеличивают время поиска глобального экстремума.

Кроме того, ГА находят оптимальное решение только внутри заданного диапазона поиска. Поэтому диапазоны поиска и число особей в популяции требуется задавать с большим запасом, что также увеличивает время решения.

Перечисленные особенности сдерживают широкое применение ГА в инженерной практике. Однако потребность в таких алгоритмах для решения прикладных задач сравнительно небольшой размерности постоянно растет, особенно в связи с намечающейся тенденцией внедрения в системы управления нейросетевых технологий.

В [2] предложен модифицированный ГА (МГА), применение которого обеспечивает возможность нахождения глобального экстремума с требуемой вероятностью при существенно меньшем, по сравнению с диплоидным ГА, числом обращений к функции цели.

МГА содержит в себе генетические качества статистической селекции популяции поисковых точек. Для исключения «неудачных» потомков в нем реализована процедура регулярного поиска локальных экстремумов с использованием метода деформируемого многогранника. При смене поколений в алгоритме заложена рекомендуемая во многих источниках 10-процентная замена «неперспективных» особей (элиминирование).

Для инициализации алгоритма необходимо установить: размерность задачи оптимизации N , численность особей в популяции μ , точность решения задачи ε и значения матрицы D , определяющие границы интервалов значений координат оптимизируемых векторов популяции.

В общем случае, МГА почти в два раза меньше обращается к функции цели по сравнению с диплоидной версией ГА, что вдвое повышает оперативность получения решения. Кроме того, в процессе

регулярного поиска МГА способен найти глобальный экстремум за пределами заданного диапазона. Перечисленные особенности позволяют заключить, что применение модифицированного ГА в задачах многоэкстремальной оптимизации (в частности, параметров элементов САУ) является более эффективным, чем использование его классической версии.

Практическая часть

МГА может быть реализован в программном средстве Global Minimum Researcher (GMR), предназначенной для решения многопараметрических многоэкстремальных задач оптимизации посредством поиска глобального минимума функции цели. Скриншот основного окна GMR представлен на рис. 1.

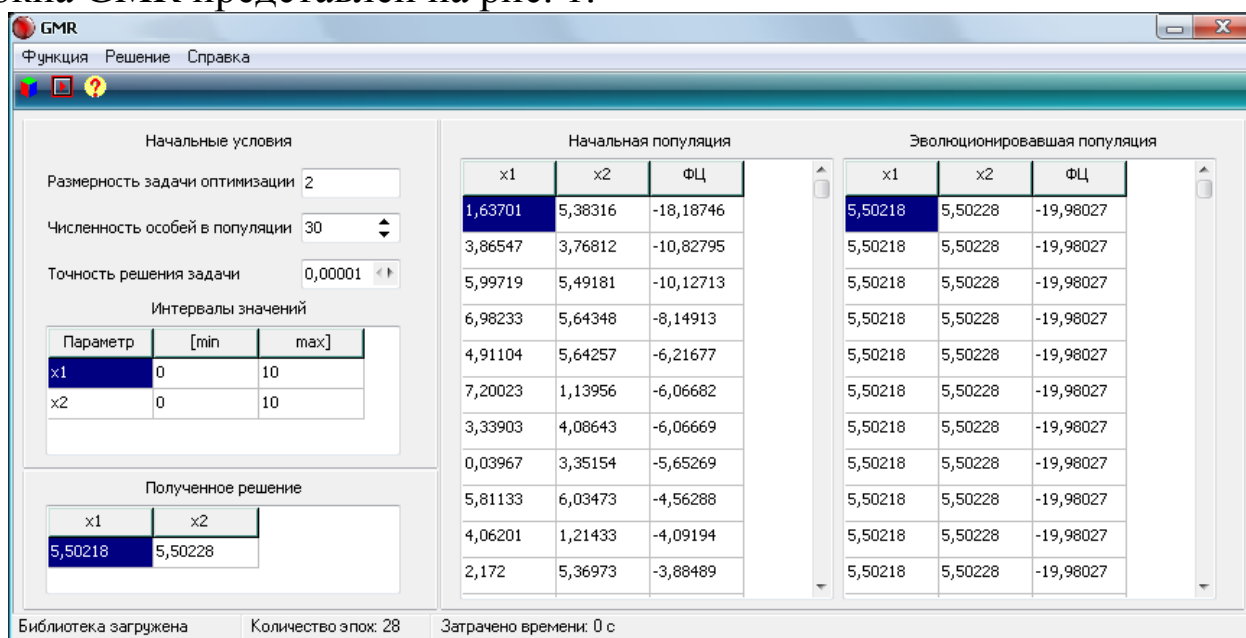


Рисунок 1 – Основное окно приложения «GMR»

На панели «Начальные условия» расположены:

1. окно «Размерность задачи оптимизации», которое служит для отображения числа оптимизируемых параметров;
2. окно «Численность особей в популяции», через которое вводится и отображается количество картежей типа <Решение | Значение функции цели>, обрабатываемых в процессе расчета решения;
3. окно «Точность решения задачи», необходимое для ввода и отображения величины, определяющей максимально допустимое рассогласование значений целевой функции, соответствующих полученным в результате работы программы решениям;

4. таблица «Интервалы значений», посредством которой вводятся и отображаются идентификаторы и априори определяемые границы интервалов значений (область допустимых значений) оптимизируемых параметров.

Размерность задачи оптимизации определяется GMR автоматически, исходя из размера массива, представляющего собой параметр функции, экспортируемой из динамически присоединяемой библиотеки (DLL).

Численность особей в популяции (ЧОП) влияет на расчет решения неоднозначно. В случае если оптимизируемая функция является унимодальной, то ЧОП влияет только на скорость сходимости. Если оптимизируется многоэкстремальная функция, то, помимо скорости сходимости, в зависимости от ЧОП находится еще и вероятность нахождения глобального экстремума.

Точность решения задачи определяет размер окрестности глобального минимума, в которой по окончании работы программы должны находиться все точки n -мерного пространства задачи оптимизации, соответствующие особям эволюционировавшей популяции.

Параметр, определяющий область допустимых значений оптимизируемых переменных, влияет на расчет решения в комплексе с ЧОП и определяет границы области, в которой строится некоторая поверхность при формировании начальной популяции. При этом ЧОП определяет уровень разреженности точек поверхности. Чем больше размер области допустимых значений и количество точек, образующих в ней поверхность, тем более полная информация о характере целевой функции передается программе до непосредственного начала поиска.

Важно отметить, что программа GMR способна находить глобальный минимум, даже в том случае, если он находится за пределами области, определенной пользователем.

На панели «Полученное решение» расположена таблица отображения оптимальных значений параметров.

На панелях «Начальная популяция» и «Эволюционировавшая популяция» расположены таблицы отображения решений с соответствующими значениями функции цели.

Порядок работы с приложением GMR следующий. Изначально необходимо создать DLL, в которую должен быть скомпилирован код функции цели, представляющий собой алгоритмическое описание модели оптимизируемой САУ. При этом функция цели должна воз-

вращать значение характеристики качества САУ, по которой осуществляется оптимизация. Нужно учитывать, что алгоритм GMR предполагает минимизацию значения этой характеристики.

Для создания библиотеки в интегрированной среде разработки Visual C++ (или C++ Builder) необходимо выполнить следующие действия.

1. Выполнить команду File | New | Other и выбрать пиктограмму Dynamic-link Library (либо DLL Wizard). Сделать в диалоговом окне установки, показанные на рис. 2. Откроется окно Редактора Кода.

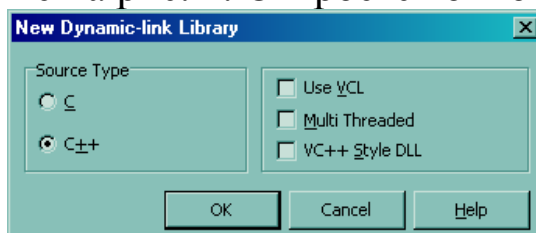


Рисунок 2 – Окно задания опций создания DLL

2. Удалить из текста комментариев.

3. Вставить перед описанием DllEntryPoint директиву включения головного файла:

```
#include "MGA.h"
```

Имя файла может быть другим. Файла MGA.h пока нет. Он будет создан позже. Также, при необходимости, нужно добавить в файл реализации директивы включения заголовочных файлов стандартных библиотек.

4. Вставить после описания DllEntryPoint описание функции цели Opt. В теле функции должна быть описана модель САУ, при этом функция должна возвращать значение характеристики качества САУ, по которой осуществляется оптимизация, то есть программа будет выполнять попытку найти такие значения оптимизируемых параметров САУ, при которых эта характеристика принимает минимальное значение.

5. Сохранить модуль и проект.

В результате рассмотренных действий файл MGA.cpp (имя файла может быть другим, здесь для примера операторы тела функции возвращают значение функции $y = x_0^2 + x_1^2$) примет следующий вид:

```
#include <windows.h>
#include <Math.h>
#include <Math.hpp>
#pragma argsused
#include "MGA.h"
```

```

int WINAPI DllEntryPoint(HINSTANCE hinst, unsigned
    long reason, void* lpReserved)
{
    return 1;
}
//-----
double Opt (double * x)
{
    double y=pow(x[0],2)+pow(x[1],2);
    return y;
}

```

6. Выполнить команду File | New | Other и выбрать пиктограмму Header File. Откроется окно Редактора Кода.

7. Записать в файл следующий код:

```

extern "C" double __declspec(dllexport) Opt
    (double * x);

```

8. Сохранить файл с именем MGA.h.

9. Выполнить команду Project | Build MGA. В результате будет создан файл MGA.dll.

Перед компиляцией рекомендуется выключить опции «Build with runtime packages» и «Dynamic RTL» на вкладках Project | Options | Packages и Project | Options | Linker | Linking соответственно. Это обеспечит возможность использования библиотеки в системе, где не установлены стандартные библиотеки (пакеты) C++ Builder.

Скомпилированную библиотеку нужно подключить к GMR, выполнив команду Функция | Загрузить DLL.

Затем нужно определить идентификаторы и диапазоны допустимых значений оптимизируемых параметров, вызвав двойным щелчком мыши по таблице «Интервалы значений» окно, представленное на рис. 3.

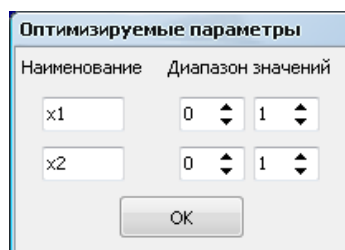


Рисунок 3 – Окно настройки свойств оптимизируемых параметров

После этого становится активной команда Решение | Рассчитать. Далее, настроив оставшиеся параметры, необходимо рассчитать решение.

Следует отметить, что скорость получения решения оценивается количеством итераций (эпох), обеспечивающим сходимость точек, соответствующих особям популяции, в окрестности минимума.

Пример

Качество управления движением мобильного объекта в общем случае определяется скоростью его перемещения, точностью обработки заданной траектории и возможностью осуществления движения в условиях действия возмущений, то есть устойчивостью движения. В этой связи рассматриваемая задача заключается в синтезе ПИД-регулятора системы управления движением мобильного объекта, обеспечивающего оптимальную обработку заданной траектории по критерию, комплексно учитывающему быстродействие и точность движения.

Постановка задачи

Введём неподвижную систему координат $OXYZ$ так, чтобы плоскость OXY совпадала с горизонтальной шероховатой плоскостью, по которой в вязкой среде движется материальная точка массой m под действием силы \vec{F} (рис. 4).

Материальную точку будем рассматривать как систему, положение которой в системе $OXYZ$ определяется вектором обобщенных координат $|x \ y|^T$, где x, y – координаты материальной точки.

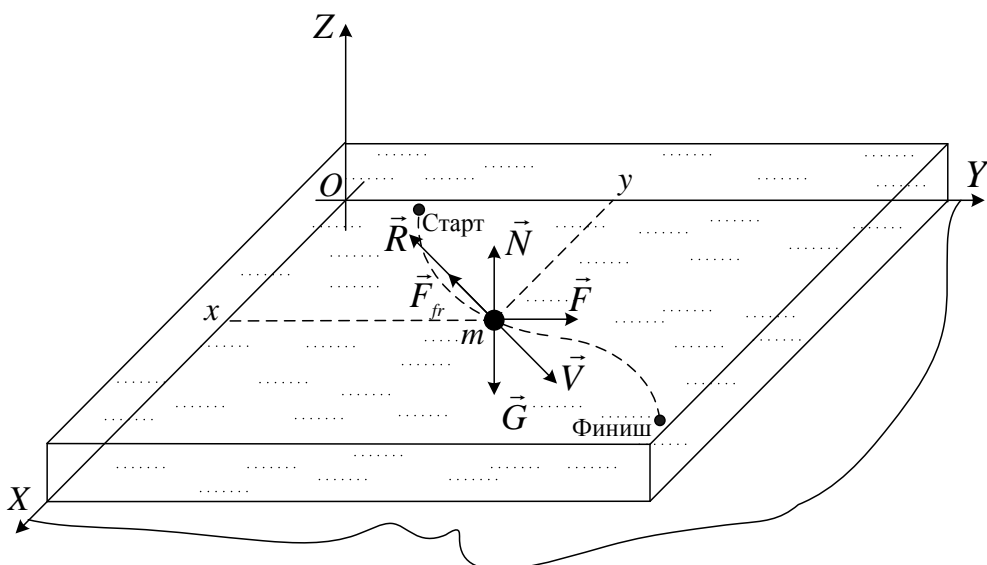


Рисунок 4 – Схема движения материальной точки по шероховатой плоскости в вязкой среде

Запишем уравнение движения материальной точки в векторной форме

$$m\vec{a} = \vec{F} + \vec{F}_{fr} + \vec{R}, \quad (1)$$

где \vec{a} – ускорение материальной точки; \vec{F}_{fr} – сила сухого трения; \vec{R} – сила вязкого трения.

В проекциях на оси системы координат $OXYZ$ уравнение (1) запишем в виде системы дифференциальных уравнений

$$\begin{cases} m\ddot{x} = F_x - F_{fr_x} - R_x, \\ m\ddot{y} = F_y - F_{fr_y} - R_y, \end{cases} \quad (2)$$

где \ddot{x}, \ddot{y} – проекции ускорения \vec{a} материальной точки на оси OX и OY соответственно (две точки обозначают двойное дифференцирование по времени); F_x, F_y – проекции движущей материальную точку силы \vec{F} на оси OX и OY соответственно; F_{fr_x}, F_{fr_y} – проекции силы сухого трения \vec{F}_{fr} на оси OX и OY соответственно; R_x, R_y – проекции силы вязкого сопротивления \vec{R} на оси OX и OY соответственно.

Исследование управляемого движения материальной точки удобно проводить, используя безразмерные переменные. Осуществив тривиальные преобразования, систему уравнений (2) можно привести к форме

$$\begin{cases} \tilde{m}\ddot{\tilde{x}} = \tilde{F}_x - \tilde{F}_{fr_x} - \tilde{R}_x, \\ \tilde{m}\ddot{\tilde{y}} = \tilde{F}_y - \tilde{F}_{fr_y} - \tilde{R}_y. \end{cases} \quad (3)$$

где \tilde{m} – безразмерная масса; \tilde{x}, \tilde{y} – безразмерные координаты; $\dot{\tilde{x}}, \dot{\tilde{y}}$ – безразмерные проекции скорости \vec{V} материальной точки на оси OX и OY соответственно; $\ddot{\tilde{x}}, \ddot{\tilde{y}}$ – безразмерные проекции ускорения \vec{a} материальной точки на оси OX и OY соответственно; \tilde{F}_x, \tilde{F}_y – безразмерные проекции движущей материальную точку силы \vec{F} на оси OX и OY соответственно; $\tilde{F}_{fr_x}, \tilde{F}_{fr_y}$ – безразмерные проекции силы сухого трения \vec{F}_{fr} на оси OX и OY соответственно; \tilde{R}_x, \tilde{R}_y – безразмерные проекции силы вязкого сопротивления \vec{R} на оси OX и OY соответственно.

Примем безразмерную массу материальной точки $\tilde{m} = 1$, тогда система уравнений (3) примет вид

$$\begin{cases} \ddot{\tilde{x}} = \tilde{F}_x - \tilde{F}_{fr_x} - \tilde{R}_x, \\ \ddot{\tilde{y}} = \tilde{F}_y - \tilde{F}_{fr_y} - \tilde{R}_y. \end{cases} \quad (4)$$

Для определения $\tilde{F}_{fr_x}, \tilde{F}_{fr_y}$ из системы уравнений (4) будем использовать модель Кулона (модель симметричного сухого трения). В соответствии с этой моделью запишем в нотации безразмерных переменных, переход к которым осуществлен с учетом того, что коэффициент трения является безразмерным, следующую систему уравнений

$$\begin{cases} \tilde{F}_{fr_x} = \tilde{N}f_x, \\ \tilde{F}_{fr_y} = \tilde{N}f_y, \end{cases} \quad (5)$$

где \tilde{N} – безразмерная проекция силы реакции шероховатой поверхности \vec{N} на ось OZ ; f_x, f_y – коэффициенты трения, определяемые следующим образом:

$$f_x = \begin{cases} f_0 \text{sign}(\dot{\tilde{x}}), \dot{\tilde{x}} \neq 0, \\ \frac{\tilde{F}_x}{\tilde{N}}, \dot{\tilde{x}} = 0, |\tilde{F}_x| \leq f_0 \tilde{N}, \\ f_0 \text{sign}(\tilde{F}_x), \dot{\tilde{x}} = 0, |\tilde{F}_x| > f_0 \tilde{N}, \end{cases} \quad f_y = \begin{cases} f_0 \text{sign}(\dot{\tilde{y}}), \dot{\tilde{y}} \neq 0, \\ \frac{\tilde{F}_y}{\tilde{N}}, \dot{\tilde{y}} = 0, |\tilde{F}_y| \leq f_0 \tilde{N}, \\ f_0 \text{sign}(\tilde{F}_y), \dot{\tilde{y}} = 0, |\tilde{F}_y| > f_0 \tilde{N}, \end{cases} \quad (6)$$

где f_0 – коэффициент сухого трения.

\tilde{R}_x, \tilde{R}_y из системы уравнений (4) определим с использованием линейной модели вязкого трения. В соответствии с этой моделью запишем следующую систему уравнений

$$\begin{cases} \tilde{R}_x = \tilde{\mu}\dot{\tilde{x}}, \\ \tilde{R}_y = \tilde{\mu}\dot{\tilde{y}}. \end{cases} \quad (7)$$

где $\tilde{\mu}$ – безразмерный коэффициент вязкого трения.

С учетом (5) и (7) систему дифференциальных уравнений динамики движения материальной точки (4) можно записать в виде

$$\begin{cases} \ddot{\tilde{x}} = \tilde{F}_x - \tilde{N}f_x - \tilde{\mu}\dot{\tilde{x}}, \\ \ddot{\tilde{y}} = \tilde{F}_y - \tilde{N}f_y - \tilde{\mu}\dot{\tilde{y}}. \end{cases} \quad (8)$$

Сила \vec{F} возникает в результате преобразования управляющего значения u :

$$\vec{F} = \vec{f}(u). \quad (9)$$

В скалярной форме в нотации безразмерных переменных уравнение (9) можно записать в виде

$$\begin{cases} \tilde{F}_x = \begin{cases} \tilde{k}u, u > 0, \\ 0, u < 0, \end{cases} \\ \tilde{F}_y = \begin{cases} 0, u > 0, \\ -\tilde{k}u, u < 0, \end{cases} \end{cases} \quad (10)$$

где \tilde{k} – безразмерный коэффициент пропорциональности.

Пусть $\tilde{k} = 1$, тогда на основе (10) можно записать следующую систему уравнений

$$\begin{cases} \tilde{F}_x = \begin{cases} u, u > 0, \\ 0, u < 0, \end{cases} \\ \tilde{F}_y = \begin{cases} 0, u > 0, \\ -u, u < 0. \end{cases} \end{cases} \quad (11)$$

Пусть $\tilde{F}_x = u_x$, $\tilde{F}_y = u_y$, тогда систему уравнений (11) можно записать в виде

$$\begin{cases} \ddot{\tilde{x}} = u_x - \tilde{N}f_x - \tilde{\mu}\dot{\tilde{x}}, \\ \ddot{\tilde{y}} = u_y - \tilde{N}f_y - \tilde{\mu}\dot{\tilde{y}}, \end{cases} \quad (12)$$

где

$$\begin{cases} u_x = \begin{cases} u, u > 0, \\ 0, u < 0, \end{cases} \\ u_y = \begin{cases} 0, u > 0, \\ -u, u < 0. \end{cases} \end{cases} \quad (13)$$

Система (12) представляет собой систему дифференциальных уравнений динамики управляемого движения материальной точки по шероховатой плоскости в вязкой среде с безразмерными переменными.

В рассматриваемом случае заданной траекторией является s -образная гладкая кривая — сигмоида (рис. 5), задаваемая функцией

$$y^*(x^*) = \frac{a}{1 + e^{kx^* + b}},$$

где a, b, k – параметры сигмоиды.

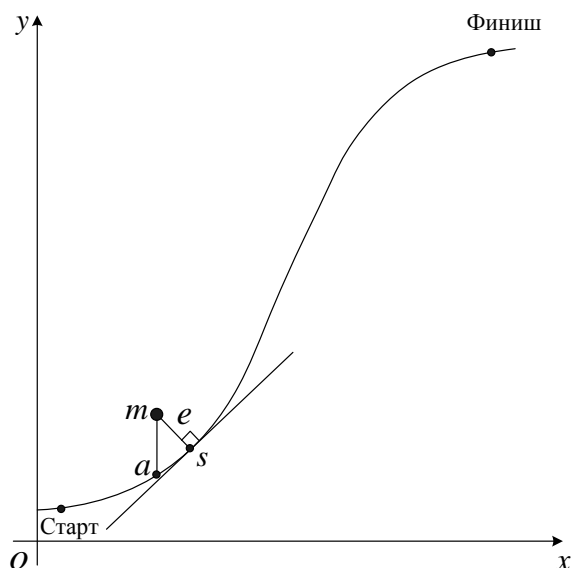


Рисунок 5 – Заданная траектория движения материальной точки

Задача управления заключается в обеспечении оптимального движения материальной точки по заданной s -образной траектории от точки «Старт» до точки «Финиш» (рис. 4, 5). При этом оптимальность движения характеризуется значением показателя, комплексно учитывающего точность и быстродействие САУ при обработке заданной траектории.

Система управления движением

Структурная схема системы управления движением материальной точки представлена на рис. 6. Система включает регулятор (Р); преобразователь управляющего сигнала (П); объект управления (ОУ); блок расчета ускорения, сообщаемого материальной точке силой вязкого сопротивления (ВТ); блок расчета ускорения, сообщаемого материальной точке силой сухого трения (СТ); обзреватель рабочего пространства (ОРП).

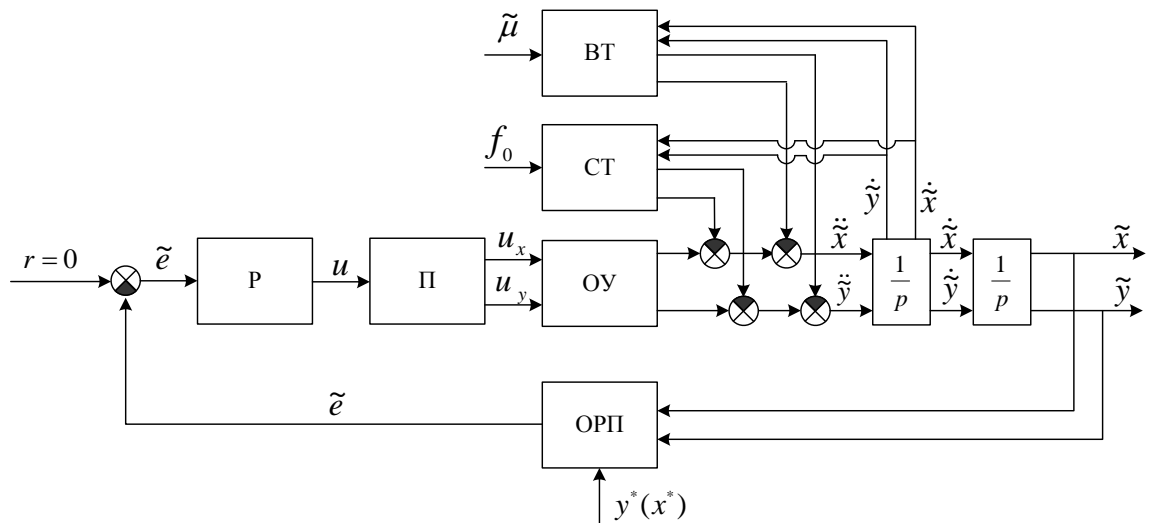


Рисунок 6 – Структурная схема системы управления движением материальной точки

Объектом управления является материальная точка, движущаяся по шероховатой плоскости в вязкой среде. Влияние окружающей среды на объект управления реализуется блоками ВТ и СТ. Функционирование блоков ОУ, ВТ и СТ в системе описывается системой уравнений (12).

Блок ОРП реализует расчет отклонения \tilde{e} положения материальной точки от заданной траектории (рис. 5).

Модуль отклонения $|\tilde{e}|$ определяется минимальным расстоянием между точкой m и кривой. Это расстояние равно безразмерной длине отрезка, соединяющего точку m и точку s кривой, в которой касательная к кривой перпендикулярна этому отрезку. В рамках предложенного алгоритма расчета модуля отклонения \tilde{e} точка s определяется без учета указанного условия следующим образом. Определяются координаты $(x_a, y^*(x_a))$ точки кривой a . На первой итерации поиска $x_a = x$, где x – текущая координата точки m по оси OX . На последующих итерациях x_a увеличивается на 1, если $y > y^*(x)$, либо уменьшается на 1, если $y < y^*(x)$. На каждой итерации рассчитывается длина отрезка ma следующим образом: $ma = \sqrt{(x - x_a)^2 + (y - y^*(x_a))^2}$. Если значение ma увеличивается, то цикл прерывается и возвращается значение ma из предшествующей итерации.

Значение \tilde{e} определяется следующим образом:

$$\begin{cases} \tilde{e} = |\tilde{e}|, & y > y^*(x), \\ \tilde{e} = -|\tilde{e}|, & y < y^*(x). \end{cases}$$

Преобразователь управляющего сигнала функционирует в соответствии с соотношениями (13).

На вход системы подается уставка $r = 0$, что предполагает отсутствие отклонения траектории движения материальной точки от заданной траектории.

Система функционирует в дискретном режиме с безразмерным периодом $\tilde{T} = 1$.

Следует отметить, что прямая, на которой лежит отрезок ms , определяющий модуль отклонения положения материальной точки от заданной траектории, в общем случае неколлинеарна вектору движущей силы \vec{F} , поэтому определение этой силы по отклонению e является задачей, сопряженной с неточностью.

Качество управления движением q определяется временем перемещения материальной точки от точки «Старт» до точки «Финиш» и степенью идентичности траектории ее движения заданной. Значение q определяется следующим образом:

$$q = \sum_{i=1}^n |\tilde{e}_i|, \quad (14)$$

где \tilde{e}_i – безразмерное отклонение от заданной траектории на i -ой итерации управления; n – число итераций управления, осуществленных при движении точки от «Старта» до «Финиша».

Синтез оптимального ПИД-регулятора и результаты моделирования

В качестве регулятора системы управления движением используется ПИД-регулятор.

Уравнение ПИД-регулятора в настоящем случае в дискретной форме имеет следующий вид:

$$u_{i+1} = k_{\text{П}} \tilde{e}_i + k_{\text{Д}} (\tilde{e}_i - \tilde{e}_{i-1}) + k_{\text{И}} \sum_{p=1}^i \tilde{e}_p, \quad (15)$$

где u_{i+1} – управляющий сигнал на $i+1$ -ой итерации управления, $k_{\text{П}}, k_{\text{И}}, k_{\text{Д}}$ – соответственно коэффициенты пропорциональности, интегрирования и дифференцирования; $\tilde{e}_i, \tilde{e}_{i-1}$ – безразмерные отклонения соответственно на i -ой и на $i-1$ -ой итерации управления; \tilde{e}_p – безразмерное отклонение на p -ой итерации управления.

Оптимальные значения коэффициентов $k_{\text{П}}, k_{\text{И}}, k_{\text{Д}}$ были определены с использованием приложения GMR. Скриншот основного окна

приложения GMR в режиме расчета коэффициентов усиления ПИД-регулятора системы управления движением приведен рис. 7. При этом «хромосомы» формировались из значений коэффициентов усиления, $f_0 = 0,05$, $\mu = 0,1$, начальными условиями задачи перемещения были: $x = 10$, $y = 20$, $\dot{x} = 0$, $\dot{y} = 0$.

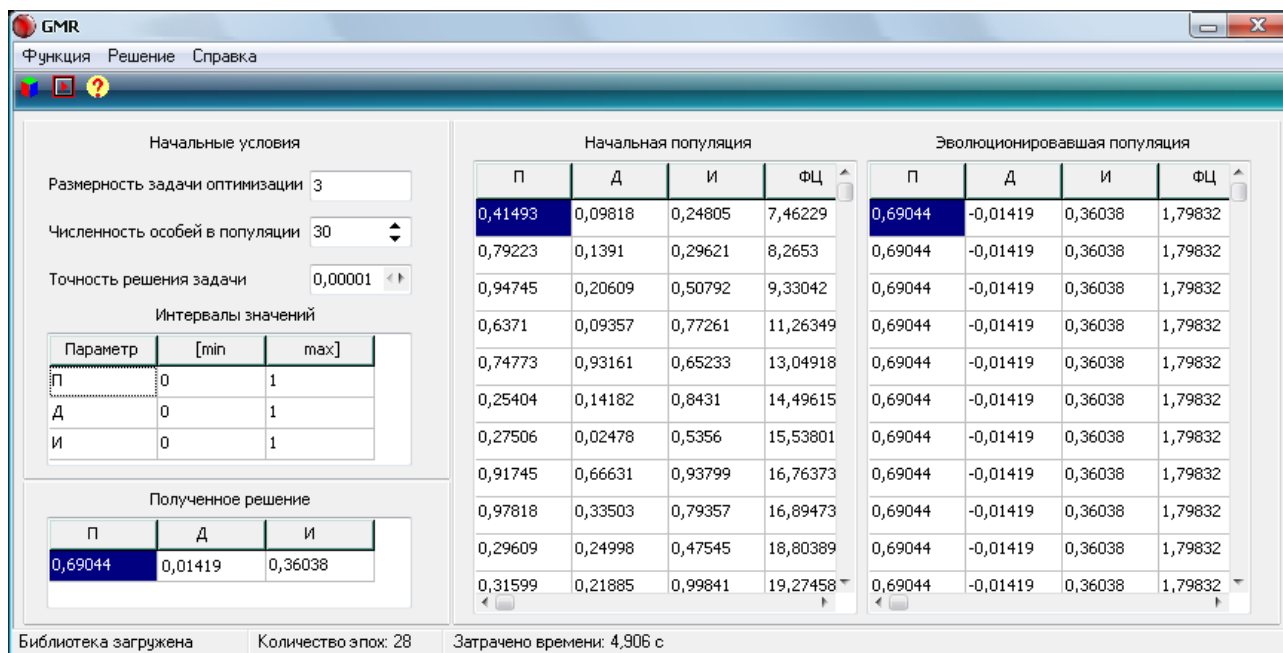


Рисунок 7 – Скриншот программного приложения, реализующего модифицированный генетический алгоритм

Были получены следующие значения коэффициентов усиления: $k_{\text{П}} = 6,9045$, $k_{\text{И}} = 3,60295$, $k_{\text{Д}} = 0,141$.

Управляемое движение материальной точки моделировалось с использованием разработанного программного приложения, интерфейс которого представлен на рис. 8.

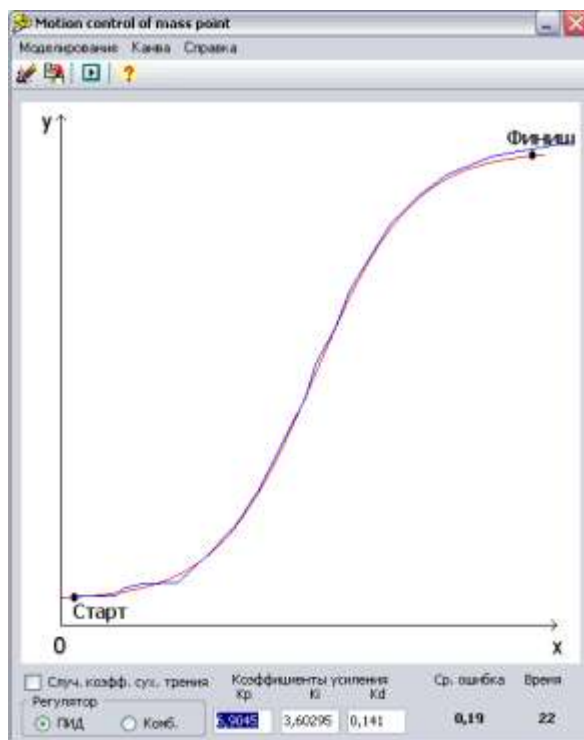


Рисунок 8 – Скриншот программного приложения моделирования управляемого движения материальной точки

Вид траектории движения материальной точки, управляемого с применением оптимального ПИД-регулятора, также показан на рисунке 9.

Аппаратные средства и программное обеспечение

Рекомендуется использование интегрированной среды разработки Visual C++.

Задание на самостоятельную работу

1. Изучить материал теоретической части.
2. Реализовать математическую модель САУ в коде функций цели.
3. Задать два-три критерия оптимизации САУ, учесть их в реализациях функций цели.
4. Создать динамически присоединяемые библиотеки, скомпилировав в них программный код функций цели.
5. Оптимизировать параметры САУ по заданным критериям с использованием программного средства GMR.
6. Оценить результаты оптимизации САУ, представив результаты математического моделирования (в виде графиков, скриншотов и т.п.).

Отчет должен содержать цель работы, задание и практическую часть, включающую:

1. описание моделируемой САУ;
2. описание критериев оптимизации САУ;
3. скриншоты главного окна программного средства GMR, отображающего начальные условия и результаты оптимизации;
4. листинги файлов реализации, содержащих описание целевых функций;
5. результаты моделирования САУ с оптимизированными параметрами.
6. выводы по результатам проведенных исследований.

Вопросы для самостоятельной подготовки

1. Что такое генетический алгоритм?
2. Каковы области применения генетических алгоритмов?
3. Каковы основные классы задач, решаемых с помощью генетических алгоритмов?
4. Какими параметрами определяется качество управления движением мобильным объектом?
5. В чём заключается задача управления движением мобильным объектом?

2. Идентификация элементов САУ на базе синтеза искусственной нейронной сети прямого распространения

Цель: изучение способа идентификации элементов САУ, основанного на синтезе искусственной нейронной сети прямого распространения.

Теоретическая часть

Искусственные нейронные сети (ИНС) – сети с конечным числом слоев из однотипных элементов – искусственных (формальных) нейронов с различными типами связей между слоями. При этом количество нейронов в слоях выбирается необходимым для обеспечения заданного качества решения задачи, а число слоев нейронов как можно меньшим с целью уменьшения времени решения задачи.

Каждый нейрон характеризуется своим текущим состоянием по аналогии с нервными клетками живого организма, которые могут быть возбуждены или заторможены. Он обладает группой синапсов – однонаправленных входных связей, соединенных с выходами других нейронов, а также имеет аксон – выходную связь данного нейрона, с которой сигнал (возбуждения или торможения) поступает на синапсы следующих нейронов. Структурно-функциональная организация формального нейрона представлена на рис. 9.

Каждый синапс характеризуется величиной синаптической связи или ее весом w_i , который по физическому смыслу эквивалентен электрической проводимости.

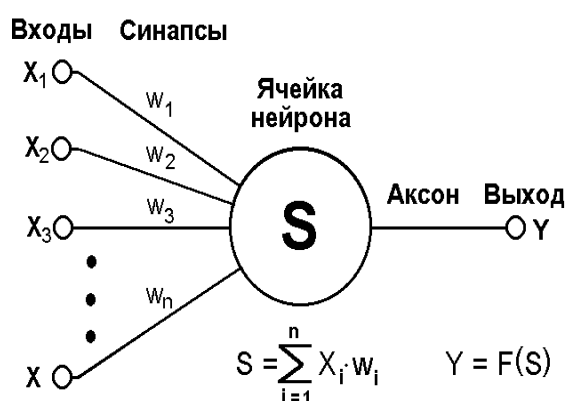


Рисунок 9 – Структурно-функциональная организация искусственного нейрона

Текущее состояние нейрона определяется как взвешенная сумма его входов:

$$S = \sum_{i=1}^n x_i \cdot w_i, \quad (16)$$

где n – число входов нейрона; x_i – значение i -го входа нейрона; w_i – вес i -го синапса.

Выход нейрона есть функция его состояния

$$Y = f(S + w_0) = f(S), \quad (17)$$

где f – некоторая функция, называемая активационной; w_0 – «нейронное смещение», вводимое для инициализации сети, – подключается к неизменяемому входу «+1» (на рис. 9 такой вход не показан).

Активационная функция может иметь различный вид, как показано на рис. 10.

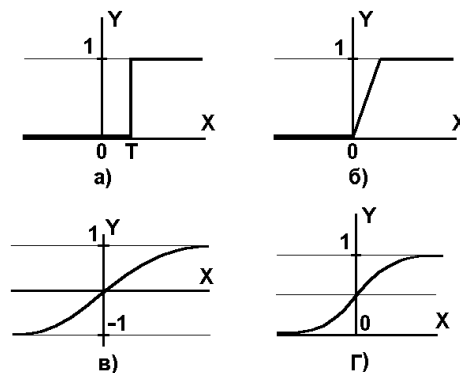


Рисунок 10 – а) ступенчатая функция; б) линейный порог; в) гиперболический тангенс; г) сигмоида

Одной из наиболее распространенных является нелинейная функция с насыщением, так называемая логистическая функция или сигмоида (т.е. функция s -образного вида):

$$f(x) = \frac{1}{1 + e^{-\alpha x}}. \quad (18)$$

При уменьшении α сигмоида становится более пологой, при $\alpha=0$ вырождаясь в горизонтальную линию на уровне 0.5, при увеличении α сигмоида приближается по внешнему виду к функции единичного скачка с порогом T в точке $x=0$. Из выражения для сигмоиды (18) очевидно, что выходное значение нейрона лежит в диапазоне (0,1). Одно из ценных свойств логистической функции – простое выражение для ее производной:

$$f'(x) = \alpha f(x)(1 - f(x)). \quad (19)$$

Следует отметить, что логистическая функция является гладкой, что используется в некоторых алгоритмах обучения. Кроме того, она обладает свойством усиливать слабые сигналы лучше, чем сильные, и

предотвращает насыщение от сильных сигналов, так как они соответствуют областям аргументов, где сигмоида имеет пологий наклон.

ИНС может рассматриваться как направленный граф со взвешенными дугами, в котором искусственные нейроны являются узлами. По архитектуре связей ИНС могут быть сгруппированы в два класса: сети прямого распространения, в которых графы не имеют петель, и рекуррентные сети, или сети с обратными связями.

В наиболее распространенном семействе сетей первого класса, называемых многослойными персептронами, нейроны расположены слоями, связанными однонаправлено. Синтез сети именно такого типа рассматривается в данной работе.

Сети прямого распространения являются статическими в том смысле, что на заданный вход они вырабатывают одну совокупность выходных значений, не зависящих от предыдущего состояния сети.

Функциональные особенности нейронов и способ их объединения в сетевую структуру обуславливают ту или иную парадигму ИНС. Для решения задач идентификации и управления наиболее адекватными, без сомнения, являются многослойные ИНС (МНС) прямого действия или многослойные персептроны (МСП). При проектировании МНС нейроны объединяются в слои, каждый из которых обрабатывает вектор сигналов от предыдущего слоя (или входной вектор). Минимальной реализацией является двухслойная ИНС, состоящая из входного (распределительного), промежуточного (скрытого) и выходного слоя. При подсчете числа слоев входной слой обычно не учитывается, так как служит лишь для распределения входных сигналов по нейронам последующего слоя. На рис. 11 представлена структурная схема двухслойной ИНС прямого распространения. Сигналы в сети распространяются от входа к выходу, связи между нейронами одного слоя и обратные связи отсутствуют.

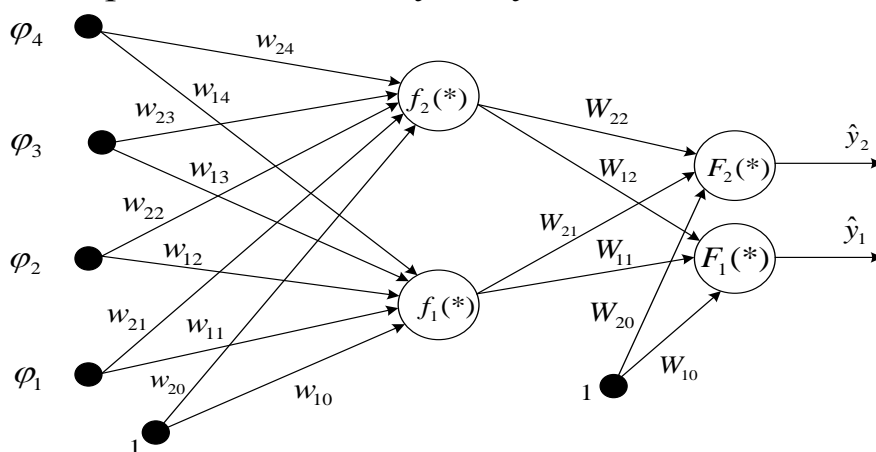


Рисунок 11 – Структурная схема двухслойной нейронной сети

Реализация модели двухслойной ИНС прямого распространения имеет следующее математическое представление:

$$\hat{y}_i(\boldsymbol{\theta}) = \hat{y}_i(\mathbf{w}, \mathbf{W}) = F_i\left(\sum_{j=1}^{n_h} W_{ij} f_j\left(\sum_{l=1}^{n_\varphi} w_{jl} \varphi_l + w_{j0}\right) + W_{i0}\right), \quad (20)$$

где n_φ – размерность вектора входов φ нейронной сети; n_h – число нейронов в скрытом слое; $\boldsymbol{\theta}$ – вектор настраиваемых параметров нейронной сети, включающий весовые коэффициенты и нейронные смещения (w_{jl}, W_{ij}); $f_j(x)$ – активационная функция нейронов скрытого слоя; $F_i(x)$ – активационная функция нейронов выходного слоя.

Рассмотрим функционирование МНС как совокупности взаимосвязанных элементов (нейронов) с математической точки зрения. Каждый структурный элемент МНС получает на входе вектор сигналов $\boldsymbol{\varphi}$, вычисляет его скалярное произведение на вектор весовых коэффициентов нейрона \mathbf{w} и некоторую функцию F в выходной сигнал y . Результат поступает на входы других нейронов или на выход. Таким образом, нейронные сети вычисляют суперпозиции функций одного переменного и их линейные комбинации. Для обоснования возможности использования МНС в качестве моделей динамических систем нужно получить ответ на вопрос: можно ли произвольную непрерывную функцию n переменных получить с помощью операций сложения, умножения и суперпозиции функций одного переменного?

В серии работ А.Н. Колмогорова и В.И. Арнольда решена следующая математическая проблема (составляющая существо тринадцатой проблемы Гильберта): любую непрерывную функцию n переменных можно получить с помощью операций сложения, умножения и суперпозиции из непрерывных функций одного переменного. На основе этих работ доказан ряд теорем об аппроксимации непрерывных функций многих переменных ИНС с использованием практически произвольной функции одного переменного. Помимо подтверждения общих аппроксимирующих свойств МНС необходимо получить ответы на ряд частных вопросов, касающихся структуры сети:

Сколько скрытых слоев должна содержать нейронная сеть?

Сколько нейронов должно быть включено в каждый слой?

Какой тип активационной функции должен быть выбран?

Как отмечено в [1], по мнению некоторых исследователей, любая непрерывная нелинейная функция может быть аппроксимирована

с достаточной точностью ИНС с одним скрытым слоем, содержащим нейроны с сигмоидальными (или типа «гиперболический тангенс») функциями активации, и выходным слоем, содержащим нейроны с линейной активационной функцией. Сделаны попытки исследования влияния числа нейронов в скрытом слое на аппроксимирующие свойства сети, однако полученный результат практически невозможно применить на практике.

Возможно, репрезентативные способности МНС могут быть улучшены путем введения дополнительных скрытых слоев, особенно в случае моделирования сложных взаимосвязей. Однако усложнение структуры МНС приводит к значительным трудностям при практической реализации, обучении и последующем анализе. Это объясняет факт использования именно минимальной реализации МНС в большинстве технических приложений.

Предположим, что в результате проведения эксперимента и предварительной обработки данных получено некоторое множество

$$\mathbf{Z}^N = \{[u(t), y(t)], t = \overline{1, N}\}, \quad (21)$$

где $u(t), y(t)$ – соответственно входы и выходы системы, N – число дискретных отсчетов. Допустим также, что выбрана некоторая модельная структура

$$y(t) = \hat{y}(t | \boldsymbol{\theta}) + e(t). \quad (22)$$

В соответствии с общей схемой реализации процедуры идентификации следующим этапом является оценка параметров выбранной модельной структуры. При использовании нейросетевых модельных структур этот этап представляет собой настройку весовых коэффициентов сети в результате реализации процедуры обучения на множестве примеров. Обучение представляет собой отображение множества экспериментальных данных на множество параметров нейросетевой модели

$$\mathbf{Z}^N \rightarrow \hat{\boldsymbol{\theta}} \quad (23)$$

с целью получения оптимального, в силу некоторого критерия, прогноза выходного сигнала \hat{y} . Традиционно используемым критерием является среднеквадратичная ошибка прогнозирования

$$V_N(\boldsymbol{\theta}, \mathbf{Z}^N) = \sqrt{\frac{1}{N} \sum_{t=1}^N (y(t) - \hat{y}(t | \boldsymbol{\theta}))^2} = \sqrt{\frac{1}{N} \sum_{t=1}^N e^2(t, \boldsymbol{\theta})}. \quad (24)$$

Данный подход относится к классу методов ошибки прогнозирования (МОП), так как основной задачей является минимизация суммарной нормы ошибки прогнозирования. В некоторых случаях

рассматриваются нормы, отличные от квадратичной, которые являются оптимальными при негауссовом распределении возмущений $e(t)$. При использовании критерия в виде (24) МОП соответствует оценке методом максимального правдоподобия при условии нормального распределения возмущений $e(t)$.

Наиболее привлекательной чертой метода является достаточно простой алгоритм оценки параметров (весовых коэффициентов) ИНС и независимость от возмущений (при условии их нормального распределения). В ряде случаев данный критерий не является абсолютно оптимальным, но в практических приложениях приводит к хорошим результатам.

Для обучения ИНС в большинстве случаев применяется градиентный метод. В основе градиентного метода, или метода наискорейшего спуска, лежит определение направления поиска как противоположного направлению градиента, т.е.

$$\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)} - \mu^{(i)} G(\boldsymbol{\theta}^{(i)}). \quad (25)$$

Сходимость метода существенно зависит от выбора шага $\mu^{(i)}$: при достаточно малом шаге обеспечивается уменьшение критерия на каждой итерации: $V_N(\boldsymbol{\theta}^{(i+1)}, \mathbf{Z}^N) \leq V_N(\boldsymbol{\theta}^{(i)}, \mathbf{Z}^N)$. Применение метода к обучению ИНС дает возможность организовать вычисления таким образом, чтобы рационально использовалась структура конкретной ИНС. В этом случае метод называется методом обратного распространения (ошибки), или обобщенным дельта-правилом.

Для выбора шага алгоритма, определяющего скорость сходимости, могут применяться различные методы, в том числе и адаптивные, хотя во многих приложениях используются методы с постоянным шагом $\mu^{(i)}$.

Независимо от выбора шага, градиентный метод может обеспечить только линейную сходимость, т.е. $|\boldsymbol{\theta}^{(i+1)} - \boldsymbol{\theta}^*| \leq c |\boldsymbol{\theta}^{(i)} - \boldsymbol{\theta}^*|$, $c \in [0,1)$.

Недостаточно высокая скорость сходимости алгоритма делает невозможным применение метода для решения задач в режиме реального времени. Тем не менее, метод может быть эффективно использован в нейросетевых приложениях благодаря значительной простоте реализации, скромным требованиям к оперативной памяти и возможности использования естественной параллельности алгоритма при наличии специализированного аппаратного обеспечения.

Практическая часть

Данные, полученные в результате экспериментов с элементами САУ (объектами управления, регуляторами) на стенде, записаны в текстовые файлы Lab_2_n.txt (два входа, один выход). Информация, содержащаяся в этих файлах, должна использоваться для обучения ИНС.

Изначально необходимо открыть txt-файл (рис. 12).

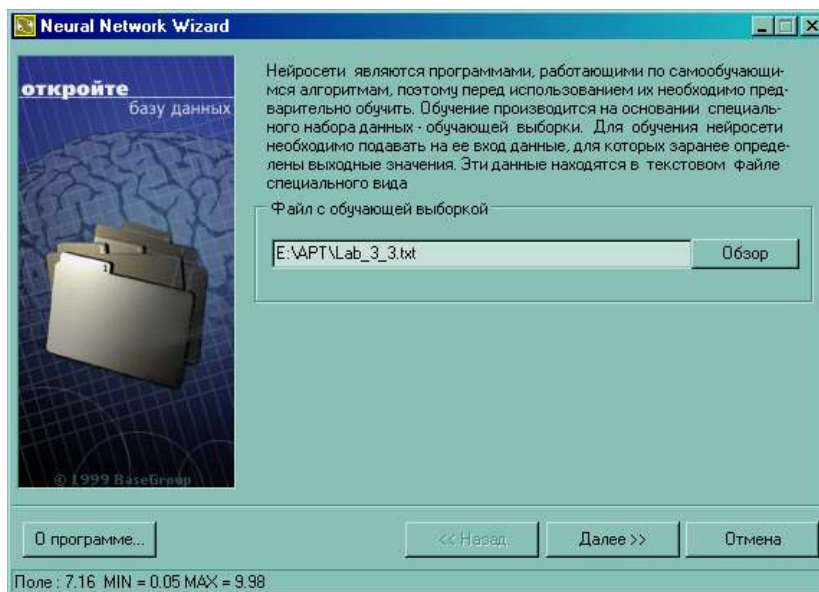


Рисунок 12 – Окно подключения к источнику данных

Затем необходимо задать входные и выходные поля, а также их свойства. При этом нужно учитывать следующее.

Синтезируемая ИНС состоит из входного, выходного и скрытых слоев. Количество нейронов в первом и последнем слое зависит от того, сколько полей определено как входные и выходные. Поля, отмеченные пометкой «не использовать», в обучении и тестировании ИНС применяться не будут.

На вход ИНС должна подаваться информация в нормализованном виде, т.е. числа в интервале $[0,1]$. Данный интервал называется интервалом допустимых значений и определяется функцией активации нейронов. Нормализация необходима для эффективного использования интервала максимальной чувствительности функции активации.

В Neural Network Wizard (NNW) реализована возможность выбора метода нормализации. Возможными вариантами являются: линейная нормализация, экспоненциальная нормализация и авто-

нормализация, основанная на статистических характеристиках выборки. При необходимости можно изменять параметры конкретного метода нормализации. Окно настройки свойств полей представлено на рис. 13.

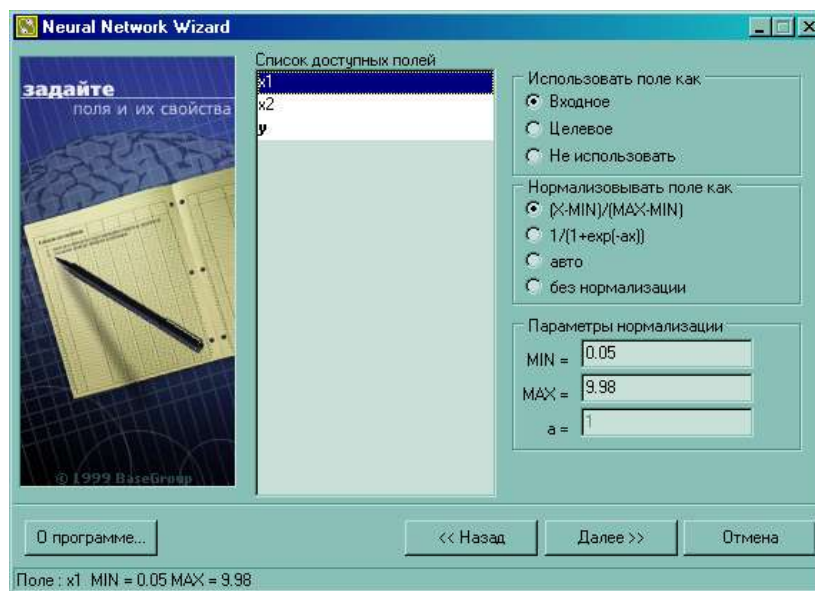


Рисунок 13 – Окно настройки свойств полей

Далее нужно указать количество скрытых слоев, количество нейронов в них и параметр активационной функции.

Общего правила, позволяющего определить количество скрытых слоев, нет. Обычно задается 1-3 скрытых слоев. Можно утверждать, что чем более нелинейная задача, тем больше скрытых слоев должно быть.

В NNW все элементы предыдущего слоя связаны со всеми элементами последующего. Количество нейронов в каждом скрытом слое также необходимо задать. Общих правил определения количества нейронов нет, но необходимо, чтобы число связей между нейронами было меньше количества примеров в обучающей выборке. Иначе ИНС потеряет способность к обобщению, а просто «запомнит» все примеры из обучающей выборки. Тогда при тестировании на примерах, присутствующих в обучающей выборке, ИНС будет демонстрировать прекрасные результаты, а на реальных данных – весьма плохие.

Сигмоида применяется для обеспечения нелинейного преобразования данных. В противном случае ИНС сможет выделить только линейно разделимые множества. Чем больше значение параметра, тем больше активационная функция походит на пороговую. Параметр

сигмоиды подбирается эмпирически. Окно настройки параметров ИНС представлено на рис. 14.

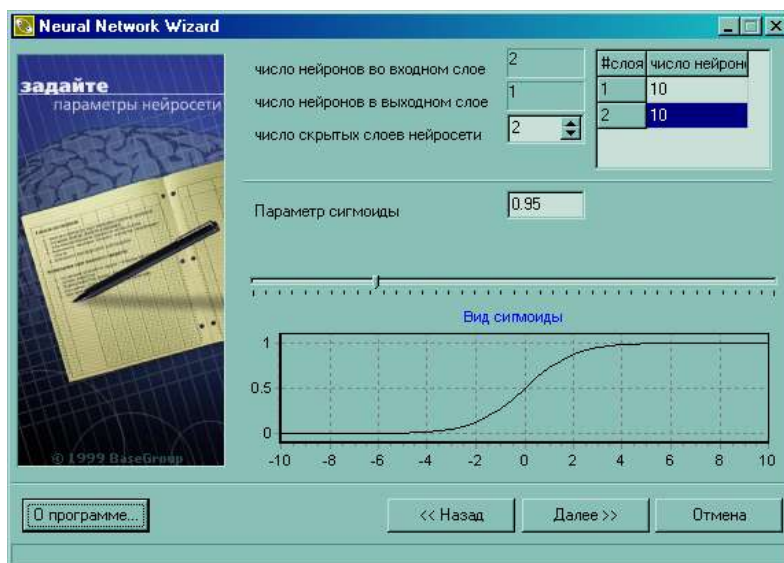


Рисунок 14 – Окно настройки параметров ИНС

После настройки параметров ИНС необходимо настроить параметры обучения, учитывая следующие сведения.

Все примеры, подаваемые на вход ИНС, делятся на два множества – обучающее и тестовое, поэтому требуется задать, сколько процентов примеров будут использоваться в обучающей выборке. Записи, используемые для тестирования, выбираются случайно, но пропорции сохраняются.

Параметр «Скорость обучения» определяет амплитуду коррекции весов на каждом шаге обучения.

Параметр «Момент» определяет степень воздействия i -ой коррекции весов на $i+1$ -ую.

Параметр «Распознана, если ошибка по примеру <» определяет максимальную величину ошибки, при которой пример считается распознанным.

При установке флага напротив «Использовать тестовое множество как валидационное» обучение будет прекращено с выдачей сообщения, как только ошибка на тестовом множестве начнет увеличиваться. Это помогает избежать ситуации переобучения нейросети.

Критерии остановки обучения определяют момент, когда обучение будет закончено.

Окно настройки параметров обучения представлено на рис. 15.

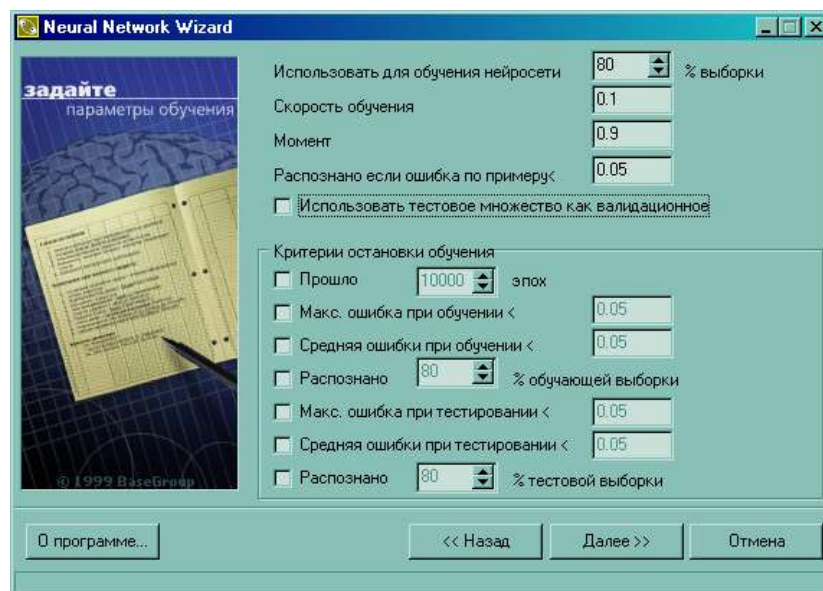


Рисунок 15 – Окно настройки параметров обучения

После завершения настройки параметров обучения сети необходимо запустить процесс обучение. На рис. 16 представлено окно запуска обучения ИНС и отображения его динамики.

В таблице над кнопкой можно наблюдать, как меняется ошибка обучения. В диаграмме отображается распределение ошибки. Зеленые столбцы – ошибка на обучающей выборке, красные – на тестовой выборке. Чем правее столбец, тем выше значение ошибки. Шкала – от 0 до 1. Чем выше столбец, тем больше примеров с указанной ошибкой.

На графиках «Распределение примеров в обучающей/тестовой выборке» можно отслеживать насколько результаты, предсказанные ИНС, совпадают со значениями в обучающей (слева) и тестовой (справа) выборках. Каждый пример обозначен на графике точкой. Если точка попадает на выделенную линию (диагональ), то ИНС предсказала результат с достаточно высокой точностью. Если точка находится выше диагонали, то ИНС недооценила его, ниже – переоценила. Необходимо добиваться, чтобы точки располагались как можно ближе к диагонали.

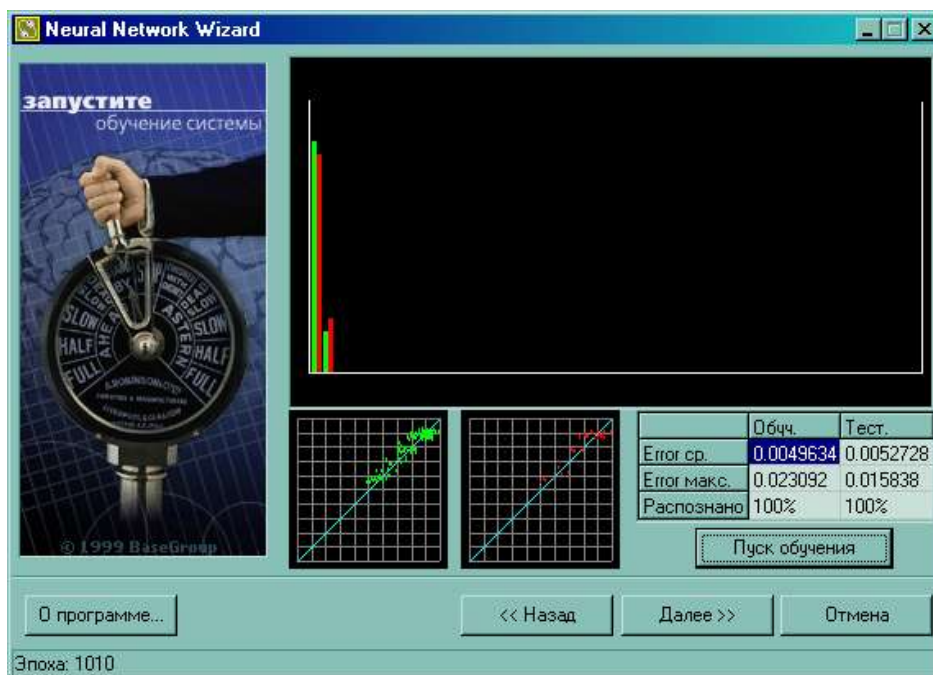


Рисунок 16 – Окно отображения динамики обучения ИНС

По завершению всех рассмотренных выше этапов можно использовать синтезированную ИНС как модель элемента САУ.

На рис. 17 представлено окно, посредством которого осуществляется расчет на базе синтезированной ИНС. В наборе входных параметров нужно ввести требуемые значения и нажать на кнопку «Расчет». В таблице «Рассчитанные параметры» появится результат.

Необходимо иметь в виду, что проверять ИНС на числах, выходящих за границы обучающей и тестовой выборки, некорректно. Например, если ИНС обучена складывать числа от 0 до 10, то необходимо и тестировать ее на числах из этого диапазона.

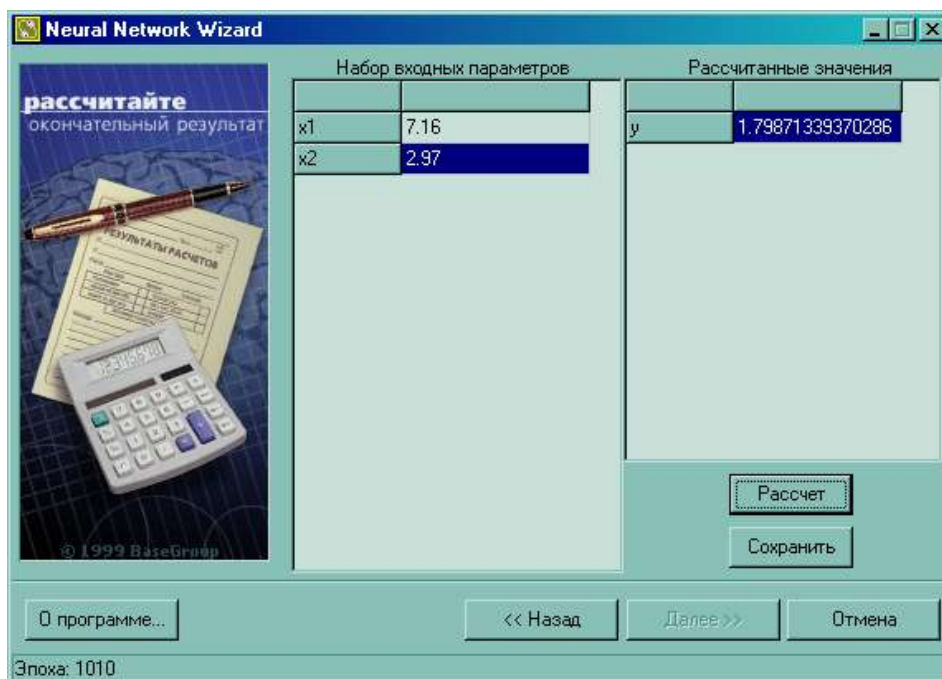


Рисунок 17 – Окно расчета на базе синтезированной ИНС

1.1 Задание

1. Изучить материал теоретической части.
2. Осуществить идентификацию элемента САУ на базе синтеза ИНС прямого распространения, следуя указаниям из практической части.
3. Изменяя параметры ИНС и ее обучения, определить характер влияния этих изменений на качество идентификации. Дать качественную оценку установленным зависимостям.

Отчет должен содержать цель работы, задание и практическую часть, включающую:

1. не менее 5 скриншотов типа рис. 16, отражающих результаты обучения при разных параметрах ИНС и обучения;
2. пояснения к скриншотам;
3. качественные оценки установленных зависимостей между параметрами ИНС (и ее обучения) и качеством идентификации;
4. выводы по результатам проведенных исследований.

3. Синтез нечеткого регулятора САУ, реализующего алгоритм Мамдани

Цель: изучение способа синтеза нечеткого регулятора САУ, реализующего алгоритм Мамдани.

Теоретическая часть

В алгоритме Мамдани используется минимаксная композиция нечетких множеств. Алгоритм включает в себя следующую последовательность этапов:

1. фаззификацию (приведение к нечеткости);
2. нечеткий вывод;
3. дефаззификацию (приведение к четкости).

Сущность процедуры фаззификации заключается в определении степени истинности антецедентов, т.е. значений функций принадлежности для антецедентов каждого правила. Для базы с m правилами и n посылками в антецеденте степень истинности обозначается как $\mu_{ik}(x_k), i = 1, \dots, m, k = 1, \dots, n$.

Формирование логического решения осуществляется следующим образом.

Изначально определяются уровни «отсечения» для антецедента каждого из правил:

$$\text{alfa}_i = \min_k(\mu_{ik}(x_k)) \quad (26)$$

Далее находятся «усеченные» функции принадлежности консеквента:

$$\mu'_i(y) = \min(\text{alfa}_i, \mu_i(y)) \quad (27)$$

Затем осуществляется композиция полученных усеченных функций, для чего используется максимальная композиция нечетких множеств:

$$\mu(y) = \max_i(\mu'_i(y)), \quad (28)$$

где $\mu(y)$ – функция принадлежности итогового нечеткого множества.

Последним этапом является дефаззификация, которая заключается в приведении к четкости результата композиции множеств на предыдущем этапе. В алгоритме Мамдани применяется метод центра тяжести (центроидный метод):

$$y^* = \frac{\int y \cdot \mu(y) dy}{\int \mu(y) dy} \quad (29)$$

На рис. 18 представлена графическая интерпретация нечеткого вывода по Мамдани для двух входных переменных x_1' и x_2' и двух нечетких правил R_1 и R_2 .

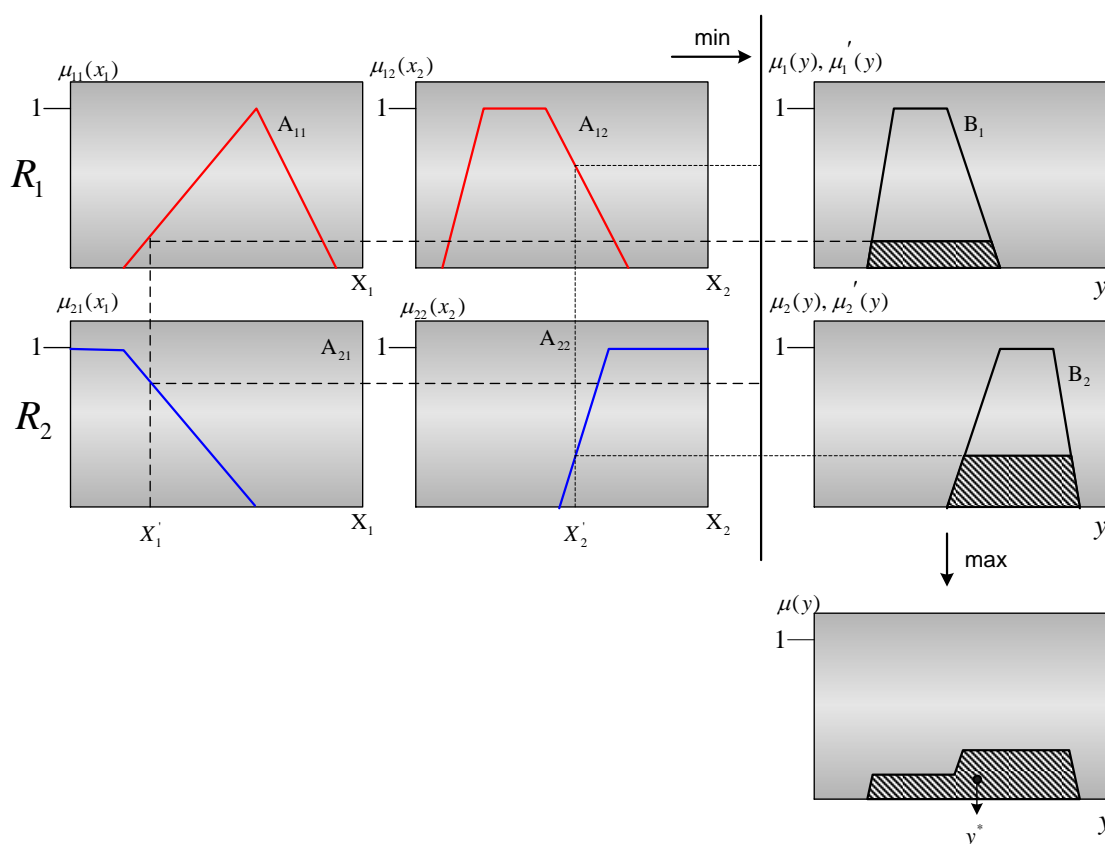


Рисунок 18 – Графическая интерпретация нечеткого вывода по Мамдани

Также существуют другие алгоритмы нечеткого вывода (Сугено, Ларсена, Цукамото). От алгоритма Мамдани они отличаются:

1. видом используемых правил;
2. типом логических операций;
3. методом дефаззификации.

Практическая часть

Синтезируемый регулятор САУ представляет собой нелинейный преобразователь с несколькими входами и одним или несколькими выходами. Для примера предположим, что посредством экспериментов на стендовом оборудовании определено, что аппроксимация зависимости, реализуемой регулятором, имеет вид, приведенный на рис. 19.

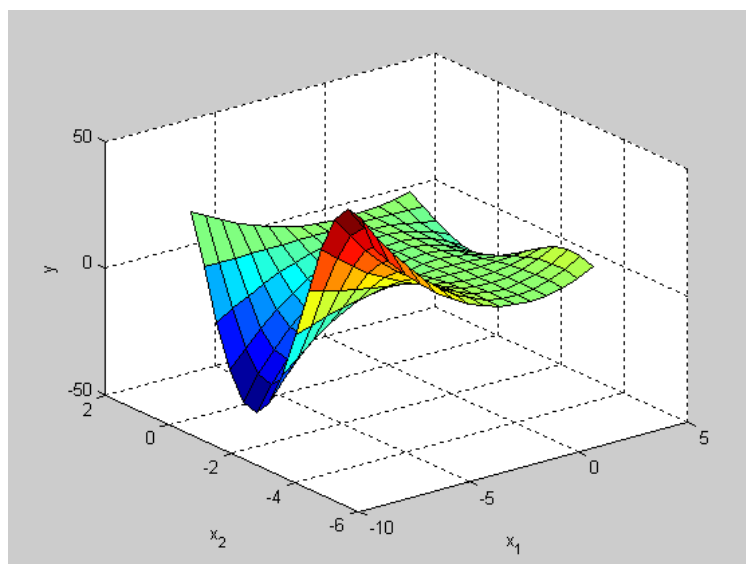


Рисунок 19 – График аппроксимации зависимости, реализуемой регулятором

Поверхности на рис. 19 ставится в соответствие следующие семь нечетких правил:

1. ЕСЛИ $x_1 = \text{«Низкий»}$ И $x_2 = \text{«Низкий»}$, ТО $y = \text{«Высокий»}$;
2. ЕСЛИ $x_1 = \text{«Низкий»}$ И $x_2 = \text{«Средний»}$, ТО $y = \text{«Низкий»}$;
3. ЕСЛИ $x_1 = \text{«Низкий»}$ И $x_2 = \text{«Высокий»}$, ТО $y = \text{«Высокий»}$;
4. ЕСЛИ $x_1 = \text{«Средний»}$, ТО $y = \text{«Средний»}$;
5. ЕСЛИ $x_1 = \text{«Высокий»}$ И $x_2 = \text{«Низкий»}$, ТО $y = \text{«Выше среднего»}$;
6. ЕСЛИ $x_1 = \text{«Высокий»}$ И $x_2 = \text{«Средний»}$, ТО $y = \text{«Ниже среднего»}$;
7. ЕСЛИ $x_1 = \text{«Высокий»}$ И $x_2 = \text{«Высокий»}$, ТО $y = \text{«Выше среднего»}$.

Синтез нечеткого регулятора для рассматриваемого примера состоит в соответствии с [3] в выполнении следующей последовательности шагов в вычислительной среде MATLAB.

1. Открыть FIS-редактор, напечатав слово fuzzy в командной строке. После этого появится новое окно, показанное на рис. 20.

2. Добавить вторую входную переменную (Edit->Add Variable->Input).

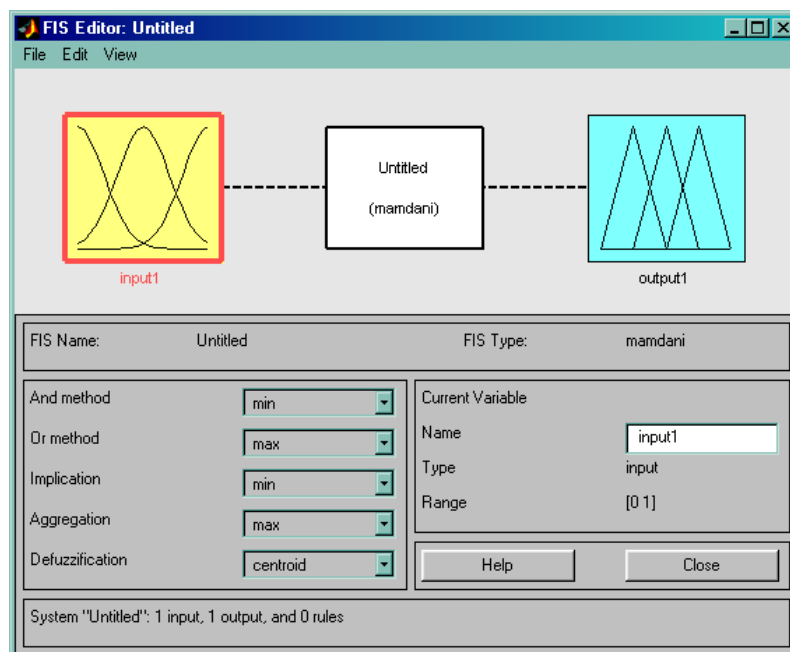


Рисунок 20 – Окно FIS-редактора

3. Переименовать входные и выходную лингвистические переменные в поле Name FIS-редактора.

4. Сохранить проект (File->Export->To Disk).

5. Перейти в редактор функций принадлежности двойным щелчком левой кнопки мыши на блоке x_1 .

6. Задать диапазон изменения переменной x_1 , напечатав -7 3 в поле Range и нажав Enter.

7. Задать функции принадлежности термов лингвистической переменной x_1 . Для лингвистической оценки этой переменной предлагается использовать три терма с треугольными функциями принадлежности. Задать наименования термов переменной («Низкий», «Средний», «Высокий»), выделив их функции принадлежности и отредактировав поле Name (рис. 21.).

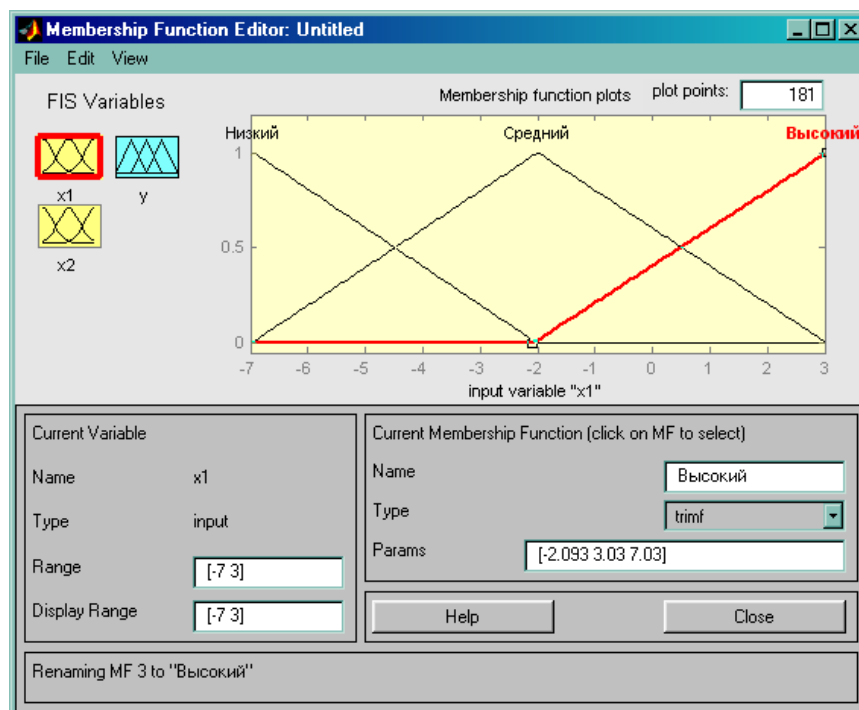


Рисунок 21 – Окно редактора функций принадлежности

8. Перейти в редакторе функций принадлежности к переменной x_2 . Прodelать шаги 6-7 для переменной x_2 . Единственным отличием здесь будет диапазон.

9. Перейти в редакторе функций принадлежности к блоку y . Для лингвистической оценки этой переменной предлагается использовать пять термов с гауссовыми функциями принадлежности.

10. Задать диапазон изменения переменной y , напечатав -50 50 в поле Range и нажав Enter.

11. Удалить функции принадлежности, установленные по умолчанию, выполнив Edit->Remove All MFs.

12. Добавить гауссовы функции принадлежности, выполнив Edit->Add MFs и выбрав в появившемся диалоговом окне MF type = gaussmf, Number of MF's = 5. Задать следующие наименования термов переменной y : «Низкий», «Ниже среднего», «Средний», «Выше среднего», «Высокий».

13. Перейти в редактор базы знаний, выполнив Edit->Rules.

14. Для ввода правила необходимо выбрать в меню соответствующую комбинацию термов и нажать Add rule. Окно редактора правил представлено на рис. 22. В конце каждого правила в скобках указан весовой коэффициент.

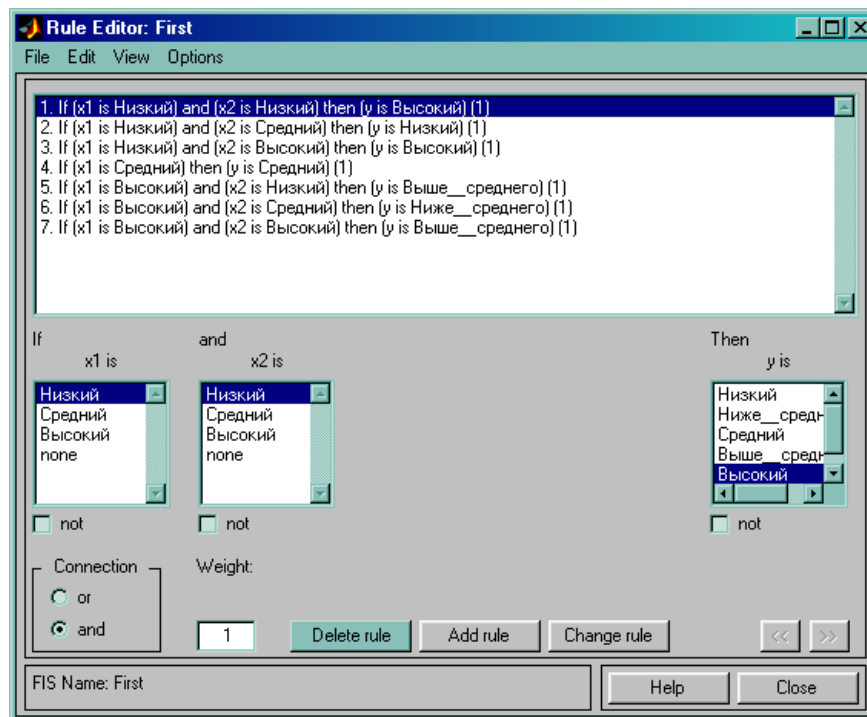


Рисунок 22 – Окно редактора правил

15. Сохранить проект (File->Export->To Disk).
16. Выполнить View->Rules. Появится окно визуализации нечеткого вывода (рис. 23).

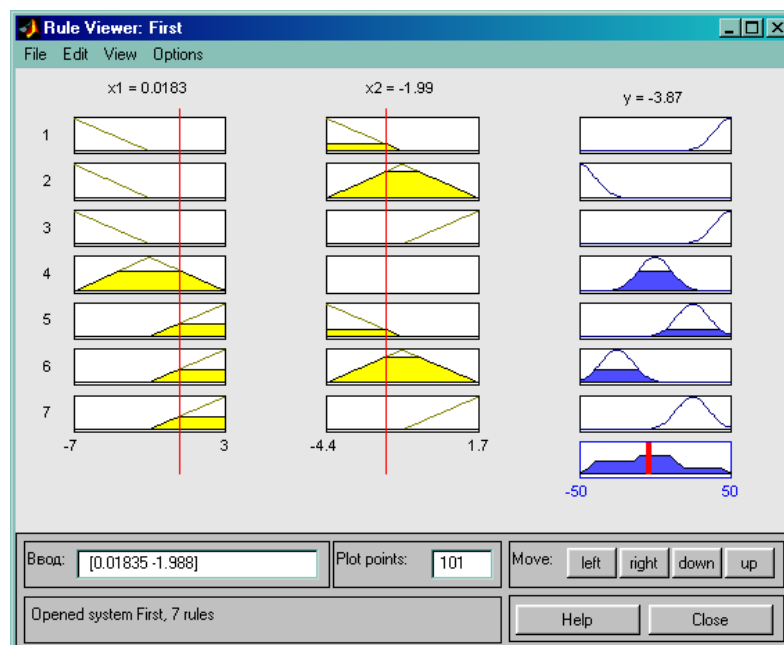


Рисунок 23 – Окно визуализации нечеткого вывода Мамдани

17. Выполнить View->Surface. Появится окно с изображением поверхности «входы – выход», соответствующей синтезированному нечеткому регулятору (рис. 24).

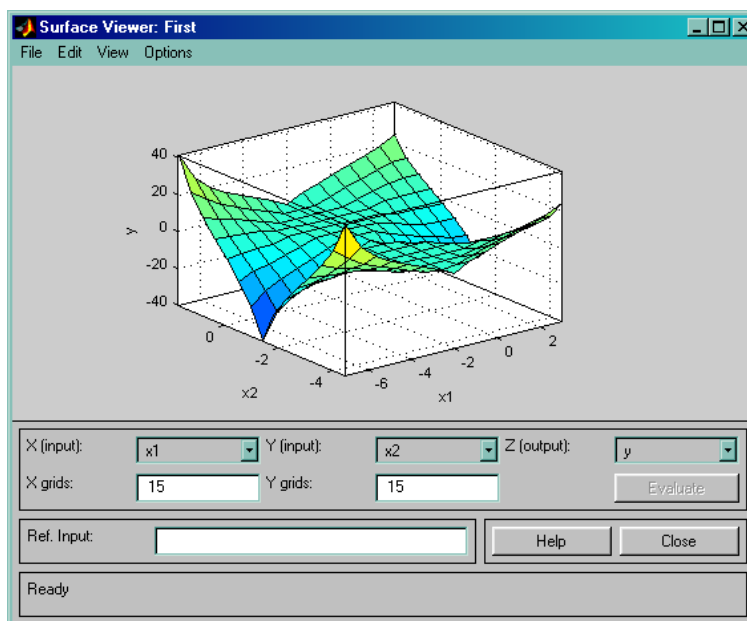


Рисунок 24 – Окно просмотра поверхности

1.2 Задание на самостоятельную работу

1. Изучить материал теоретической части.
2. Самостоятельно выбрать и представить графически нелинейную зависимость, которую должен реализовать синтезируемый регулятор САУ. Входов должно быть не менее двух, может быть более одного выхода.

3. Синтезировать нечеткий регулятор САУ, реализующий алгоритм Мамдани.

4. Сравнить вид выбранной зависимости «входы – выходы» с видом зависимости «входы – выходы», которая соответствует синтезированному нечеткому регулятору САУ. Сделать вывод.

Отчет должен содержать цель работы, задание и практическую часть, включающую:

1. скриншоты типа рис. 19, 21-24 с пояснениями;
2. качественную оценку сравнения выбранной вида зависимости «входы – выходы» с видом зависимости «входы – выходы», которая соответствует синтезированному нечеткому регулятору САУ;
3. выводы по результатам проведенных исследований.

4. Идентификация элементов САУ на базе синтеза нечеткой нейронной сети типа ANFIS

Цель: изучение способа идентификации элементов САУ, основанного на синтезе нечеткой нейронной сети типа ANFIS.

Теоретическая часть

ANFIS (Adaptive Network based Fuzzy Inference System) представляет собой пятислойную нейро-нечеткую сеть прямого распространения сигнала, реализующую нечеткий вывод Сугено. Архитектура ANFIS-сети представлена на рис. 25.

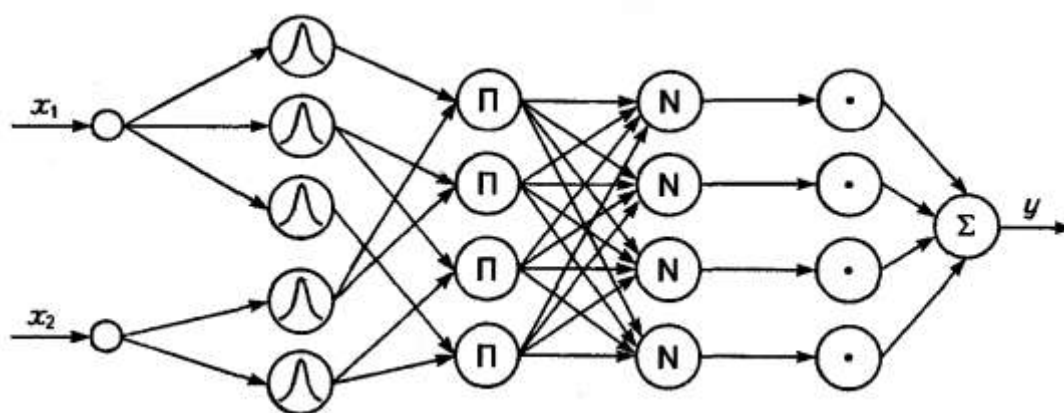


Рисунок 25 – Архитектура ANFIS-сети

На рис. 25 изображена ANFIS-сеть с двумя входными переменными (x_1 и x_2) и четырьмя нечеткими правилами. Для лингвистической оценки входной переменной x_1 используется три терма, для x_2 – два терма.

Введем обозначения, необходимые для дальнейшего изложения:

1. x_1, x_2, \dots, x_n – входы сети;
 2. y – выход сети;
 3. R_r : Если $x_1 = a_{1,r}$ И...И $x_n = a_{n,r}$, то $y = b_{0,r} + b_{1,r}x_1 + \dots + b_{n,r}x_n$ – нечеткое правило с порядковым номером r ;
 4. m – количество правил, $r = \overline{1, m}$;
 5. $a_{i,r}$ – нечеткий терм с функцией принадлежности $\mu_r(x_i)$, применяемый для лингвистической оценки переменной x_i в r -ом правиле ($r = \overline{1, m}, i = \overline{1, n}$);
 6. $b_{q,r}$ – коэффициенты в заключении r -го правила ($r = \overline{1, m}, q = \overline{0, n}$).
- ANFIS-сеть функционирует следующим образом.

Каждый узел первого слоя представляет один терм s , как правило, колоколообразной функцией принадлежности (могут использоваться и другие типы функций). Входы сети x_1, x_2, \dots, x_n соединены только со своими термами. Количество узлов первого слоя равно сумме мощностей терм-множеств входных переменных. На выход узла подается степень принадлежности значения входной переменной соответствующему нечеткому терму:

$$\mu_r(x_i) = \frac{1}{1 + \left| \frac{x_i - c}{a} \right|^{2b}}, \quad (30)$$

где a, b, c – настраиваемые параметры функции принадлежности.

Количество узлов второго слоя равно m . Каждый узел этого слоя соответствует одному нечеткому правилу. Узел второго слоя соединен с теми узлами первого слоя, которые формируют antecedentes соответствующего правила. Следовательно, каждый узел второго слоя может принимать от 1 до n сигналов. Выходом узла является степень выполнения правила, которая рассчитывается как произведение входных сигналов. Обозначим выходы этого слоя через τ_r , $r = \overline{1, m}$.

Количество узлов третьего слоя также равно m . Каждый узел этого слоя рассчитывает относительную степень выполнения нечеткого правила по формуле:

$$\tau_r^* = \frac{\tau_r}{\sum_{j=1, m} \tau_j}. \quad (31)$$

Количество узлов четвертого слоя также равно m . Каждый узел соединен с одним узлом третьего слоя, а также со всеми входами сети (на рис. 25 связи с входами не показаны). Узел четвертого слоя рассчитывает вклад одного нечеткого правила в выход сети по такой формуле:

$$y_r = \tau_r^* (b_{0,r} + b_{1,r}x_1 + \dots + b_{n,r}x_n). \quad (32)$$

Единственный узел пятого слоя суммирует вклады всех правил:

$$y = \sum_{r=1}^m y_r. \quad (33)$$

Типовые процедуры обучения нейронных сетей могут быть применены для настройки ANFIS-сети, так как в ней используются только дифференцируемые функции. Обычно применяется комбинация градиентного спуска в виде алгоритма обратного распространения ошибки (АОРО) и метода наименьших квадратов (МНК). АОРО настраивает параметры antecedentes правил, т. е. функций принадлежности. Посредством применения МНК оцениваются коэффициен-

ты заключений правил, так как они линейно связаны с выходом сети. Каждая итерация процедуры настройки выполняется в два этапа. На первом этапе на входы подается обучающая выборка и по невязке между желаемым и действительным поведением сети посредством применения МНК находятся оптимальные параметры узлов четвертого слоя. На втором этапе остаточная невязка передается с выхода сети на входы и на базе АОРО модифицируются параметры узлов первого слоя. При этом найденные на предыдущем этапе коэффициенты заключений правил не изменяются. Итерационная процедура настройки продолжается, пока невязка превышает заранее установленное значение.

Практическая часть

Данные, полученные в результате экспериментов с элементами САУ (объектами управления, регуляторами) на стенде, записаны в текстовые файлы Lab_3_n.txt (150 записей, два входа, один выход). В обучающую выборку включается 120 записей, в тестовую – 30. Обучающая (tr_set) и тестовая (test_set) выборки формируются в рабочей области среды MATLAB следующим образом:

```
samp=load('Путь к файлу\Lab_3_n.txt');  
samp_number=length(samp);  
test_index=[6:10:samp_number 7:10:samp_number];  
tr_index=setdiff(1:samp_number, test_index);  
tr_set=samp(tr_index, [1 2 3]);  
test_set=samp(test_index, [1 2 3]);
```

Далее командой `anfisedit` необходимо запустить ANFIS-редактор, в котором будет сгенерирована и обучена нечеткая система в соответствии с [3]. Для загрузки обучающей выборки в окне ANFIS-редактора (рис. 26) в поле Load data нужно выбрать опции Training (меню Type) и worksp. (меню From) и нажать кнопку Load Data....

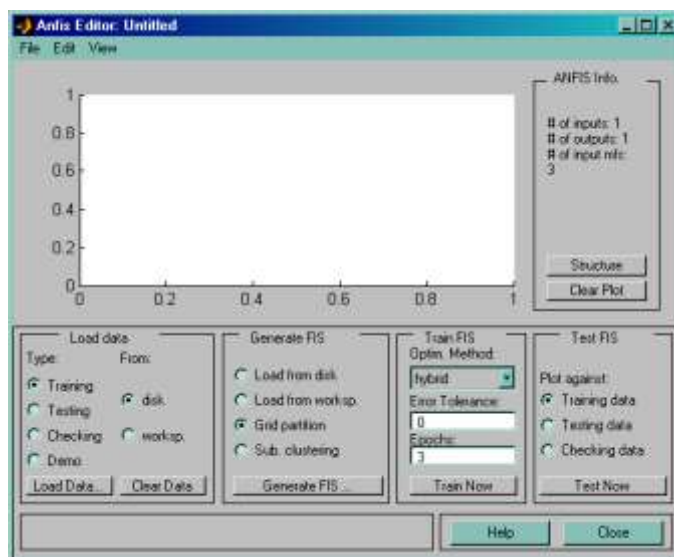


Рисунок 26 – Основное окно ANFIS-редактора

В результате появиться окно (рис. 27), в котором требуется ввести идентификатор обучающей выборки `tr_set`. Для загрузки тестовой выборки нужно установить опцию `Testing` в меню `Type`, нажать `Load Data...` и в появившемся окне набрать `test_set`. После этих действий ANFIS-редактор будет выглядеть так, как показано на рис. 28.



Рисунок 27 – Окно ввода имени переменной в ANFIS-редакторе

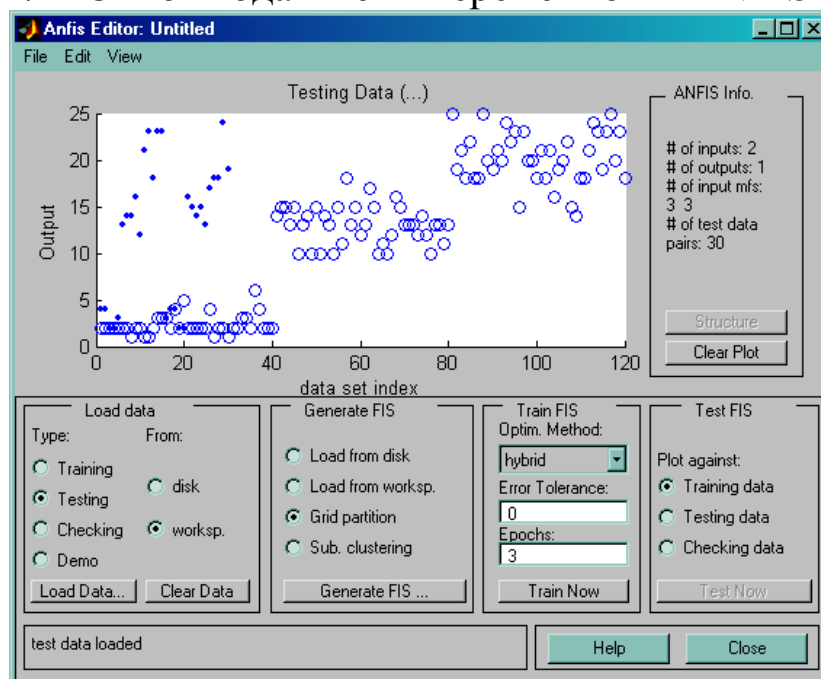


Рисунок 28 – Окно ANFIS-редактора после загрузки обучающей и тестовой выборки

Данные обучающей выборки показаны окружностями, а тестовой выборки – точками.

Для создания исходной системы нечеткого вывода требуется нажать кнопку Generate FIS.... По умолчанию используется простой метод решетчатого разбиения (Grid partition), согласно которому функции принадлежности нечетких термов равномерно распределяются внутри диапазона изменения данных. База знаний содержит все возможные варианты правил. Коэффициенты в заключениях правил принимаются равными нулю.

Теперь с нечеткой системой можно работать, используя GUI-модули пакета Fuzzy Logic Toolbox. Вызов этих модулей осуществляется через меню Edit и View. Для изображения синтезированной системы в виде нейро-нечеткой сети (рис. 29) необходимо нажать кнопку Structure в поле ANFIS Info.

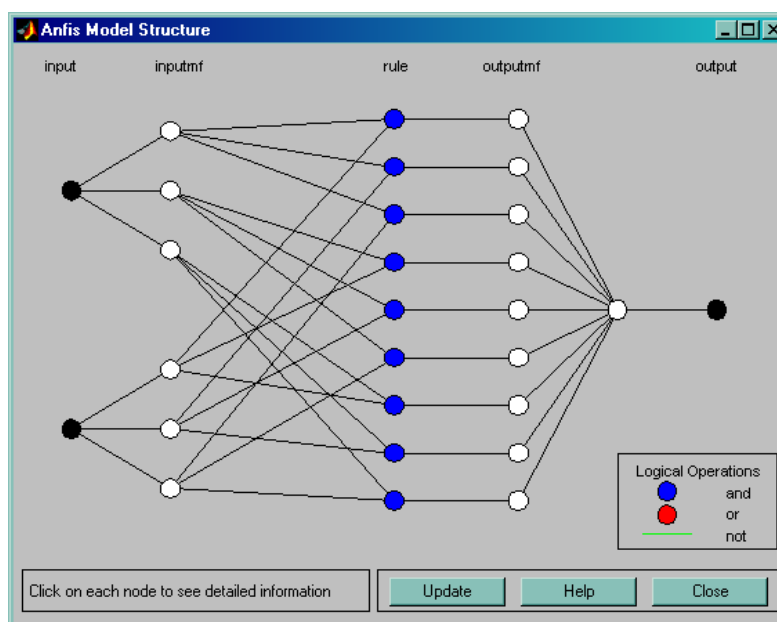


Рисунок 29 – Архитектура синтезированной нейро-нечеткой сети

Обучение нечеткой системы запускается после задания критерия его остановки. Критерием остановки может быть число итераций либо величина ошибки.

Для запуска обучения в поле Epochs меню Train FIS нужно ввести число итераций и нажать Train Now. Динамика обучения выводится в основном окне ANFIS-редактора (рис. 30): ось абсцисс соответствует итерациям алгоритма обучения, а ось ординат – значениям среднеквадратичной ошибки.

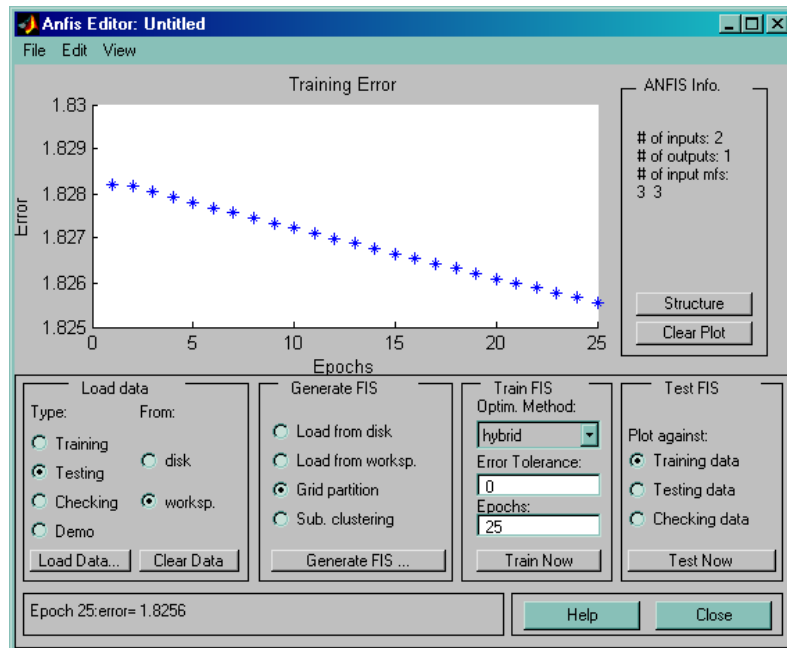


Рисунок 30 – Динамика обучения нечеткой системы

Для проверки модели на тестовой выборке требуется установить в поле Test FIS опцию Testing data и нажать Test Now. Результаты тестирования выводятся в основном в окне ANFIS-редактора (рис. 31). Экспериментальные данные показаны точками, а результаты моделирования – звездочками.

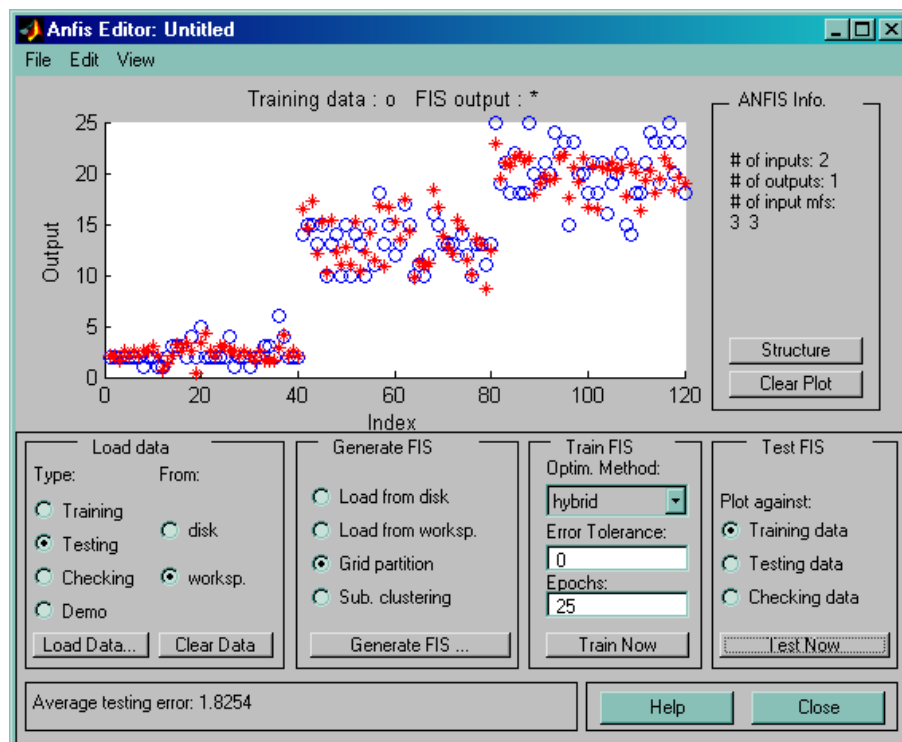


Рисунок 31 – Окно с результатами тестирования нечеткой системы

Далее выполняется идентификация параметров линейной и квадратичной модели.

Идентификация параметров линейной модели $y = a_0 + a_1x_1 + a_2x_2$ и расчет ее ошибок на обучающей и тестовой выборках осуществляются на базе следующего программного кода:

```
N=length(tr_set);
NN=length(test_set);
X=[ones(N,1) tr_set(:,1) tr_set(:,2)];
A=X\tr_set(:,3);
Out_model=X*A;
rmg_tr=norm(tr_set(:,3)-Out_model)/sqrt(N);
X_test = [ones(NN,1) test_set(:,1) test_set(:,2)];
Out_model = X_test*A;
rmg_ch=norm(test_set(:,3)-Out_model)/sqrt(NN);
```

Идентификация параметров квадратичной модели $y = a_0 + a_1x_1 + a_2x_2 + a_3x_1^2 + a_4x_2^2 + a_5x_1x_2$ и расчет ее ошибок на обучающей и тестовой выборках осуществляются на базе следующего программного кода:

```
X=[ones(N,1) tr_set(:,1) tr_set(:,2) tr_set(:,1).^2 tr_set(:,2).^2
tr_set(:,1).*tr_set(:,2)];
A=X\tr_set(:,3);
Out_model = X*A;
rmg_tr=norm(tr_set(:,3)-Out_model)/sqrt(N);
X_test=[ones(NN,1) test_set(:,1) test_set(:,2) test_set(:,1).^2
test_set(:,2).^2 test_set(:,1).*test_set(:,2)];
Out_model=X_test*A;
rmg_ch=norm(test_set(:,3)-Out_model)/sqrt(NN);
```

Задание на самостоятельную работу

1. Изучить материал теоретической части.
2. Осуществить идентификацию элемента САУ на базе синтеза нечеткой нейронной сети типа ANFIS, следуя указаниям из практической части. Изменяя параметры ANFIS и ее обучения, определить характер влияния этих изменений на качество идентификации. Дать качественную оценку установленным закономерностям.
3. Осуществить идентификацию параметров линейной и квадратичной моделей, используя программный код, представленный в практической части.

4. Сравнить результаты идентификации на базе синтеза ANFIS, линейной и квадратичной регрессий.

Отчет должен содержать цель работы, задание и практическую часть, включающую:

1. все скриншоты, представленные в практической части (за исключением рис. 26 и 27), отражающие ход решения задачи идентификации на базе синтеза ANFIS;

2. скриншот с изображением поверхности «входы – выход», соответствующей нечеткой модели элемента САУ;

3. значения среднеквадратичных ошибок полученных моделей;

4. выводы по результатам проведенных исследований.

Рекомендуемая литература

1. Методы робастного, нейро-нечёткого и адаптивного управления: Учебник / Под ред. Н.Д. Егупова, 2-е изд. – М.: Изд-во МГТУ им. Баумана, 2002. – 744 с.

2. Сабанин В.Р., Смирнов А.И., Репин А.И. Модифицированный генетический алгоритм для задач оптимизации в управлении // Exponenta Pro. Математика в приложениях. – 2004. – №3-4. – С. 78-85.

3. Штовба С.Д. Проектирование нечетких систем в среде МАТ-LAB. – М.: Горячая линия – Телеком, 2007. – 284 с.