

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тульский государственный университет»  
(ТулГУ)

На правах рукописи



**ХАПКИН Дмитрий Леонидович**

**ПРИКЛАДНЫЕ МЕТОДЫ СИНТЕЗА НЕЙРОСЕТЕВЫХ РЕГУЛЯТОРОВ  
ДЛЯ ОБЪЕКТОВ УПРАВЛЕНИЯ С ОГРАНИЧИТЕЛЯМИ**

Специальность 2.3.1. Системный анализ, управление  
и обработка информации, статистика

Диссертация на соискание ученой степени  
кандидата технических наук

Научный руководитель:  
доктор технических наук, доцент  
Феофилов Сергей Владимирович

Тула 2023

## Оглавление

Введение.....	5
1. Обзор искусственных нейронных сетей и их применения в системах автоматического управления.....	11
1.1. Искусственный нейрон.....	11
1.2. Типы и структуры искусственных нейронных сетей.....	12
1.3. Методы обучения искусственной нейронных сетей.....	17
1.4. Методы оптимизации параметров искусственных нейронных сетей в процессе обучения.....	19
Пример расчета метода обратного распространения ошибки для нейронной сети прямого распространения.....	21
Пример расчета метода обратного распространения ошибки во времени для рекуррентной нейронной сети.....	24
1.5. Процесс обучения искусственной нейронной сети.....	27
1.6. Применение нейронных сетей в системах управления.....	29
1.7. Выводы.....	39
2. Методы синтеза нейросетевых регуляторов для объектов с ограничителями....	41
2.1. Выбор рациональной структуры нейросетевого имитатора и нейросетевого регулятора на основе нейронных сетей прямого распространения.....	41
Пример обучения ИНС в качестве нейроимитатора линейного объекта.....	42
Пример синтеза нейрорегулятора для управления линейным объектом методом инверсного нейроуправления.....	44
2.2. Класс нелинейностей.....	46
Пример синтеза нейроимитатора на основе нейронной сети прямого распространения для звена с ограничением типа насыщение.....	53
Пример синтеза нейроимитатора на основе нейронной сети прямого распространения для звена с ограничителем типа жесткий механический упор.....	55

2.3. Структура нейроимитатора и нейрорегулятора на основе рекуррентных нейронных сетей для объектов с ограничителями.....	57
2.4. Алгоритм формирования обучающей выборки для нейронных сетей разных структур с учетом наличия ограничителей в объекте управления.....	58
2.5. Метод синтеза нейрорегулятора на основе схемы обучения инверсного нейрорегулятора для объектов с ограничителями.....	59
Пример синтеза нейрорегулятора методом инверсного нейрорегулятора для звена с ограничителем типа насыщение.....	62
Пример синтеза нейрорегулятора методом инверсного нейрорегулятора для звена с ограничителем типа жесткий механический упор.....	63
Пример синтеза нейрорегулятора на основе рекуррентной нейронной сети методом инверсного нейрорегулятора для звена с ограничителем типа жесткий механический упор.....	65
2.6. Метод синтеза нейросетевого регулятора на основе схемы обучения обратного пропуска ошибки через нейроимитатор.....	67
2.7. Выводы.....	72
3. Синтез устойчивой замкнутой следящей системы с нейросетевым регулятором и объектом управления с ограничителем.....	74
3.1. Преобразование ИНС к задаче целочисленного программирования.....	74
Пример преобразования ИНС с функцией активации leakyReLU к задаче целочисленного программирования.....	76
3.2. Структурный синтез устойчивой нейросетевой системы управления.....	78
3.3. Поиск контр-примеров, нарушающих условия устойчивости.....	84
3.4. Обучение нейронных сетей на контр-примерах.....	85
Пример синтеза устойчивой системы управления обратным маятником.....	87
3.5. Выводы.....	92
4. Методика синтеза устойчивых нейросетевых регуляторов для объектов с ограничителями.....	94

Пример синтеза устойчивой нейросетевой системы управления перевернутым маятником с учетом времени регулирования.....	98
Пример синтеза устойчивой замкнутой системы управления гидроприводом. .....	100
4.1. Выводы.....	126
Заключение.....	127
Список сокращений и условных обозначений.....	129
Список литературы.....	130
Приложение А.....	141
Приложение Б.....	142

## Введение

**Актуальность темы исследования.** Одной из проблем современной теории автоматического управления является сложность синтеза регуляторов для нелинейных объектов управления (ОУ) в условиях неполной информации. Конечно, существуют многочисленные, часто выдающиеся, но отдельные результаты и методы, позволяющие проектировать системы управления для определенных классов нелинейных объектов управления. К сожалению, сегодня они уже не могут удовлетворить запросы разработчиков автоматических систем управления реальными сложными динамическими объектами. В качестве примера можно привести такие распространенные системы, как гидравлические следящие приводы. С точки зрения теории автоматического управления такие объекты являются существенно нелинейными (содержат жесткие механические ограничители, фазовые ограничения, сухое и вязкое трение, нежесткость механической передачи и т.д.), нестационарными (их параметры существенно зависят от внешней окружающей среды, температуры, давления и т.д.), предполагают использование цифрового управления с заданными показателями качества. Таким образом, системы управления должны быть адаптивными и близкими к оптимальным по заданному критерию. В этом случае синтез регуляторов классическими методами предполагает обращение к соответствующим фундаментальным разделам теории автоматического управления и необходимости их одновременного использования, что приводит к громоздким алгоритмам, практическая реализация которых затруднена в связи со сложностью и стоимостью разработки и реализации.

Указанные проблемы являются системными и в рамках классической теории управления сложно решаемы. Одним из перспективных подходов является использование регуляторов, основанных на искусственных нейронных сетях (ИНС). Они могут быть эффективно применены в случае, когда отсутствует

адекватная верифицированная и достаточно точная математическая модель объекта управления, но могут быть получены экспериментальные данные. Достоинством таких регуляторов является их способность к обучению и адаптации под объект на основе полученных данных. Другим аргументом в пользу применения нейрорегуляторов является то, что практически все известные методы синтеза дают на выходе линейный закон управления. Это является существенным ограничением в случае нелинейного объекта управления. Искусственные нейронные сети могут быть представлены нелинейной динамической системой, а значит являются более гибкими и в этом смысле имеют преимущества.

В области ИНС в настоящее время наиболее интенсивно решаются задачи обработки данных (классификация изображений, обработка текстов на естественном языке, распознавание и синтез звука, и т.д.). Также проводятся научные исследования и по нейрорегулированию, но таких публикаций существенно меньше относительно их общего количества. Значительный вклад в это направление как известные зарубежные, так и российские ученые: М. Морари [55–57], Х. Дай [31, 58], С. Чен [56], А. Н. Чернодуб [23], Д. А. Дзюба [23], В. А. Терехов [10], А. И. Галушкин [2], М. В. Бураков [1], И. М. Макаров [3], В. М. Лохин [3] и др. Анализ научных работ позволяет сделать вывод о том, что остается значительный нераскрытый потенциал применения нейрорегуляторов, например, для управления объектами с широко распространенными на практике ограничителями. Кроме того, для замкнутых нейросетевых систем управления отсутствуют теоретические гарантии устойчивости, что существенно снижает возможности их применения в критически важных или опасных объектах. Таким образом, задачи, связанные с разработкой методов синтеза нейрорегуляторов для объектов с наиболее часто встречающимися на практике нелинейностями (ограничители типа насыщение, ограничители типа жесткий механический упор, и т. д.), а также разработка методов, гарантирующих устойчивость замкнутых нейросетевых систем управления являются актуальными.

**Цель работы** является разработка методов синтеза гарантированно устойчивых систем управления с нейросетевыми регуляторами для объектов с ограничителями.

Для достижения указанной цели были поставлены и решены следующие **задачи**.

1. Анализ существующих типов искусственных нейронных сетей и подходов к синтезу на их основе регуляторов в замкнутых системах автоматического управления с целью выбора перспективных направлений для работы с динамическими объектами с ограничителями.
2. Разработка алгоритма формирования структуры нейросетевого регулятора с учетом особенностей объекта управления и типов нелинейностей.
3. Разработка алгоритма формирования обучающей выборки с учетом режимов работы и типа нелинейности объекта управления.
4. Разработка модифицированных методов синтеза нейросетевых регуляторов, обеспечивающих управление объектами с ограничителями и позволяющие использовать различные типы ИНС.
5. Разработка метода синтеза нейросетевого регулятора, обеспечивающего гарантированную устойчивость в заданной локальной области фазового пространства с использованием функции Ляпунова.
6. Формирование прикладной методики синтеза и оптимизации по комплексному критерию нейросетевых регуляторов для объектов с ограничителями, в том числе в случае отсутствия точной математической модели, объединяющей разработанные методы и алгоритмы.

**Объект исследования:** замкнутые динамические системы управления нелинейными объектами с ограничителями и нейросетевыми регуляторами.

**Предмет исследования:** динамические процессы в нейросетевых системах управления с существенно нелинейными объектами.

**Методы исследования.**

При выполнении диссертационной работы использовались методы классической и современной теории управления, включающие методы интеллектуального управления, машинного обучения, конечномерной оптимизации, теории устойчивости Ляпунова и теории искусственных нейронных сетей. Все предложенные алгоритмы были реализованы и на языке программирования python с помощью библиотек Pytorch и Gurobi, а также некоторые элементы реализованы в системе Matlab.

**Положения, выносимые на защиту, обладающие научной новизной:**

1. Алгоритмы выбора структуры и формирования обучающей выборки для нейромитатора и нейрорегулятора, учитывающие наличие в объекте управления ограничителей различных видов.
2. Метод синтеза нейросетевого регулятора, основанный на схеме обучения «инверсное нейроуправление». Позволяет обеспечить управление объектом с ограничителями и использовать различные типы ИНС.
3. Метод синтеза нейросетевого регулятора, основанный на схеме обучения «обратный пропуск ошибки через нейромитатор». Позволяет обеспечить управление объектом с ограничителями и использовать различные типы ИНС.
4. Метод синтеза нейросетевого регулятора для объектов с ограничителями, гарантирующий устойчивость полученной замкнутой системы в заданной локальной области фазового пространства.
5. Методика синтеза нейросетевых регуляторов, объединяющая разработанные методы и алгоритмы, гарантирующая устойчивость полученной замкнутой системы в заданной локальной области и позволяющая оптимизировать качество переходного процесса.

**Реализация результатов работы.** Теоретические исследования, проведённые в диссертационном исследовании, поддержаны грантом РФФИ №20-38-90213 Аспиранты «Прикладные методы синтеза адаптивных робастных

нейросетевых регуляторов для существенно нелинейных объектов управления» и грантом РФФИ № 23-29-00609 «Перспективные методы синтеза и оптимизации нейросетевых систем управления для нелинейных объектов с ограничителями в условиях неполной информации». Отдельные результаты исследования использованы в учебном процессе ТулГУ и применяются в АО «Конструкторское бюро приборостроения им. академика А.Г. Шипунова».

**Практическая значимость.** Предложенная методика синтеза нейросетевых регуляторов позволяет решать задачи синтеза систем управления для нелинейных объектов с ограничителями, точная математическая модель которых не может быть построена. Полученные в диссертации теоретические результаты имеют выраженную практическую направленность. Совместное использование с методами конечномерной оптимизации и простая программная реализация позволяют получать системы с наилучшими показателями качества.

**Апробация работы.** Основные положения диссертационной работы и отдельные ее части докладывались на:

- XVIII Всероссийской научно-технической конференции студентов, магистрантов, аспирантов и молодых ученых «Техника XXI века глазами молодых ученых и специалистов», г. Тула, 2020 г.;
- XI Международной юбилейной научно-технической конференции «Проблемы совершенствования робототехнических и интеллектуальных систем летательных аппаратов», г. Москва, 2020 г.;
- XVI Всероссийской научно-технической конференции «Проблемы проектирования и производства систем и комплексов», Тула, 2020 г.;
- XXI Международная научно-методическая конференция «Информатика: проблемы, методы, технологии», г. Воронеж, 2021 г.;
- 3rd International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA 2021). Russia, Lipetsk, 2021 г.;
- IV Всероссийской научно - технической конференции «Мехатронные системы (теория и проектирование)», Тула, 2021;

- XVIII Всероссийской школе-конференции молодых ученых "Управление большими системами", г. Челябинск, 2022 г.;
- 4rd International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA), г. Липецк, 2022 г.;
- XVII Всероссийской научно-технической конференции «Проблемы проектирования и производства систем и комплексов», г. Тула, 2022 г.;
- 15-я Международная научно-техническая конференция «Приборостроение – 2022», г. Минск, 2022 г.

### **Публикации.**

Результаты исследования по теме диссертации изложены в 11 научно-технических статьях и материалах конференций, три [13, 15, 20] из которых опубликованы в изданиях ВАК, три [38, 39, 48] в изданиях, индексируемых в базе Scopus, 5 в РИНЦ [11, 12, 14, 19, 21]. Также получено два свидетельства о государственной регистрации для ЭВМ [17, 18].

### **Структура и объем работы.**

Диссертационная работы состоит из введения, 4-х глав, заключения, списка литературы. Работа содержит 142 страницы машинописного текста, в том числе 90 рисунков и 7 таблиц. Список литературы включает 90 наименования используемых источников.

## 1. Обзор искусственных нейронных сетей и их применения в системах автоматического управления

### 1.1. Искусственный нейрон

Искусственные нейронные сети (ИНС) состоят из связанных искусственных нейронов и являются математической моделью, идейно повторяющей биологические нейронные сети мозга [9, 43]. Первыми модель нейрона представили У. Маккалок и У. Питтс [60]. Искусственный нейрон (рисунок 1.1) представляет собой упрощенную математическую модель естественного нейрона [9, 10, 16]. Связь между нейронами содержит коэффициент, называемым весом  $w$ . Поступающие на вход нейрона значения умножаются на соответствующие им веса связи  $w_i$ , затем эти значения суммируются и проходят через нелинейную функцию, называемую функцией активации [16, 32, 46, 68]. Полученное значение передается далее другим нейронам или является одним из выходов нейронной сети. Обычно нейроны внутри сети распределены по слоям. У каждого слоя определено количество нейронов и общий тип функции активации.

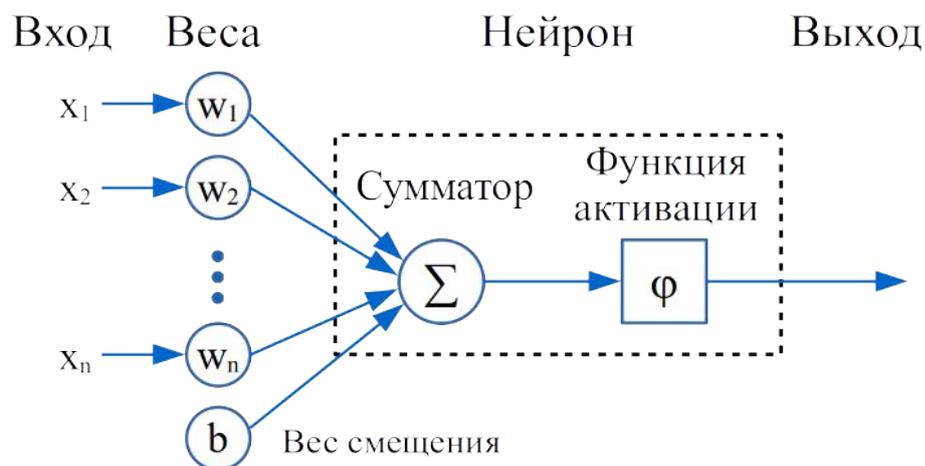


Рисунок 1.1 - Схема нейрона

Такая структура нейронной сети за счет изменения значений весов и значения коэффициентов смещения позволяет формировать взаимосвязи между данными на входе и выходе сети. Этот процесс называется обучением нейронной сети [16, 43]. В результате нейронная сеть может решать поставленную задачу: классифицировать входные данные на требуемые категории, выделять из входной информации только требуемые данные (фильтрация, распознавание), предсказывать следующий шаг (например, рост и падение акций), аппроксимировать нелинейную непрерывную функцию многих переменных (если функция активации является нелинейной, то двухуровневая нейронная сеть будет универсальным аппроксиматором такой функции [32]), хранить и воспроизводить информацию в своей памяти.

## 1.2. Типы и структуры искусственных нейронных сетей

Различают несколько основных типов искусственных нейронных сетей: нейронные сети прямого распространения [72, 73] (рисунок 1.2), сверточные нейронные сети [29, 44] (рисунок 1.3), рекуррентные нейронные сети [36] (рисунок 1.4).

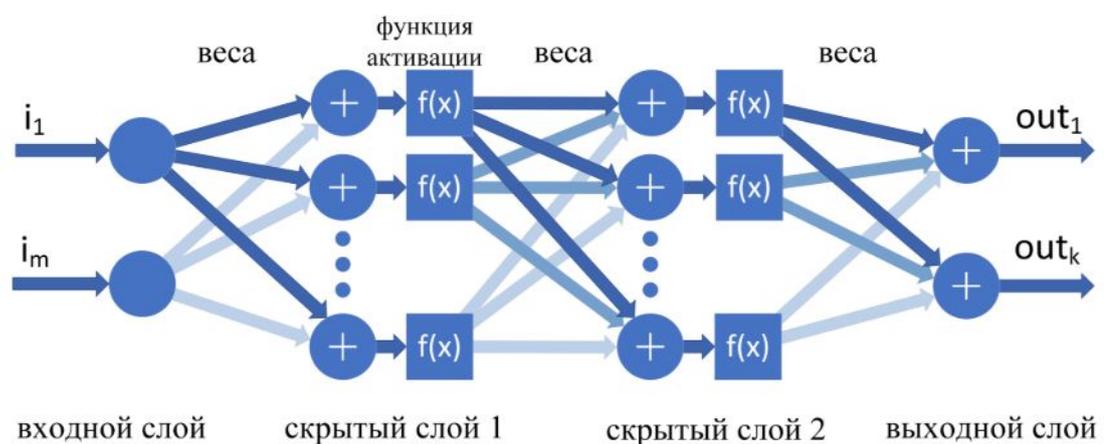


Рисунок 1.2 - Структура многослойной нейронной сети прямого распространения

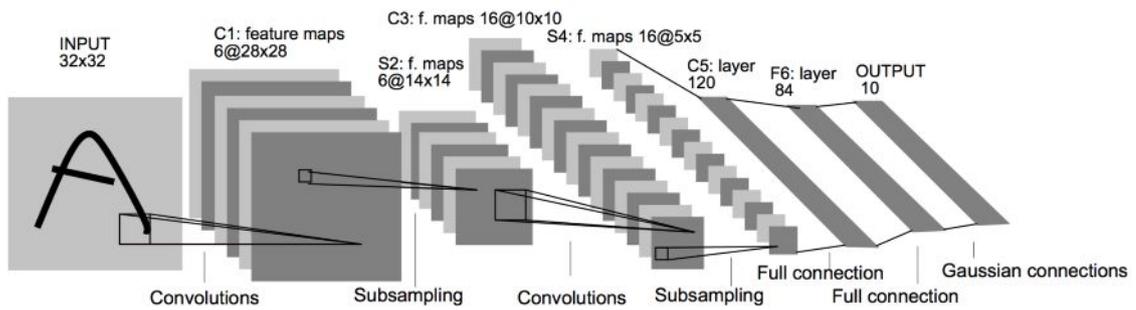


Рисунок 1.3 - Схема сверточной нейронной сети LeNet-5 [29]

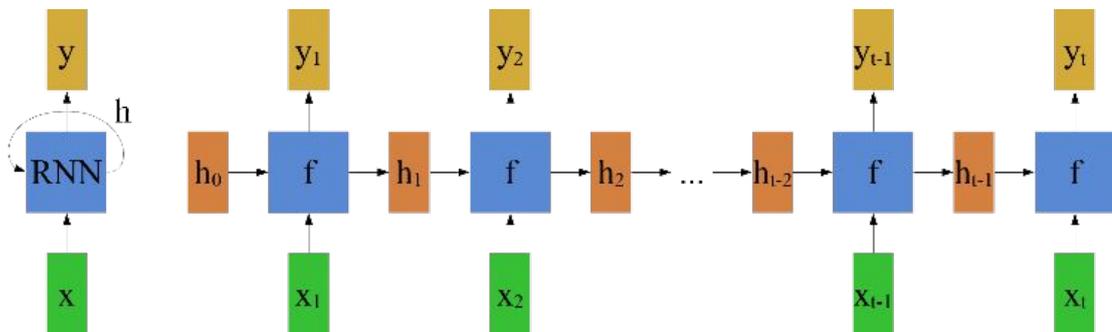


Рисунок 1.4 - Схема рекуррентной нейронной сети

В 1949 году был разработан перцептрон Розенблатта [72] с одним скрытым слоем. Фактически это первая нейронная сеть прямого распространения. В ней использовалась пороговая функция активации, а работала она только с бинарными данными (0 или 1). В 1986 г. Румельхард Д. Е. [73] предлагает использовать нелинейные функции активации: сигмоидальная (sigmoid, логистическая кривая) (1.1) (рисунок 1.5), и гиперболический тангенс (tanh) (рисунок 1.6):

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \quad (1.1)$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \quad (1.2)$$

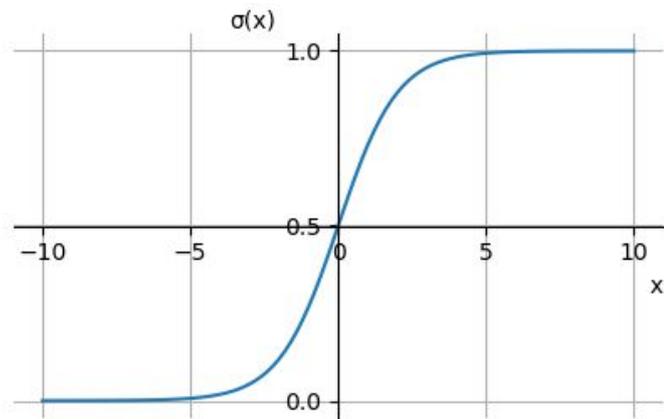


Рисунок 1.5 - График логистической функции (сигмоида)

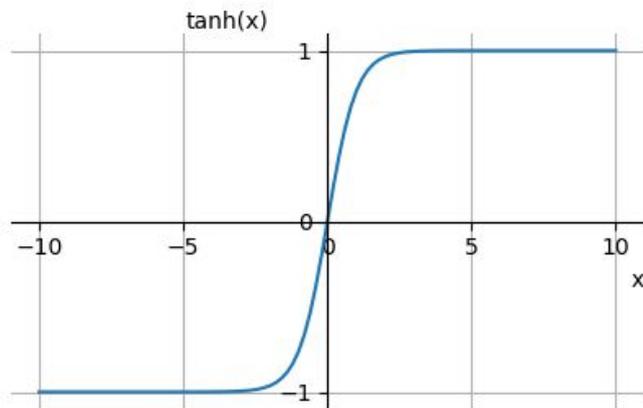


Рисунок 1.6 - График гиперболического тангенса

Структуры нейронных сетей прямого распространения и рекуррентных нейронных сетей отличаются наличием в рекуррентных нейронных сетях внутренних обратных связей. Для них в настоящее время разработаны несколько вариантов рекуррентных слоев (LSTM [45] и GRU [64]). Они были разработаны для улучшения качества обучения в задачах, связанных с необходимостью обрабатывать длительные последовательности данных, особенно где новые данные имеют неявные зависимости от предыдущих данных, а также для борьбы с эффектами затухания и взрыва градиента [45]. Рекуррентные нейронные сети в настоящее время часто используются для анализа и классификации по различным критериям текстов и звуков, переводов текстов, распознавая речи, аппроксимации нелинейной динамики [76] и, похожим образом, они способны обрабатывать и любые другие временные последовательности.

Для более эффективного решения задач компьютерного зрения были разработаны сверточные нейронные сети. В них используются ядра свертки, используемые в классических методах обработки изображений, которые позволяют выделить для следующего слоя некоторые свойства и признаки изображения, например контуры. Первую сверточную сеть leNet-5 разработал Ян Лекун с соавторами в 1989 году [29, 44].

В начале 2000-х годов основной идеей для улучшения качества распознавания на изображениях с помощью нейронных сетей было увеличение количества слоев в их структурах. Однако активно используемые на тот момент функции активации, такие как sigmoid и tanh, не позволяли обучать множество слоев сразу (обычно не более 3-5), так как в процессе обучения градиент от слоя к слою слишком сильно затухал. В 2010 году Дж. Хинтон предложил функцию активации ReLU [40, 62], которая позволила обучать многослойную нейронную сеть как единое целое (обучение всех слоев сразу). Подход к обучению большого количества слоев получил название глубокое обучение. В 2012 году А. Крижевский (аспирант Дж. Хинтона) в соревнованиях ImageNet применил подход на основе глубоких нейронных сетей. Его сверточная нейронная сеть AlexNet [51] победила с существенным отрывом. В 2014 году Симонян К. и Зиссерман А. предлагают VGG-сеть [74] с 19 скрытыми слоями. Осенью 2014 года Кристиан Жегеди с коллегами предлагают GoogLeNet [42], включающую в свой состав модули Inception, в которых ключевую роль играют ядра размером 1x1, и имеющую еще больше параметров. В конце 2015 г. He K., Zhang X. и др., которые являлись сотрудниками исследовательского центра Microsoft, публикуют работу [33], предлагавшую Residual Networks (ResNet, свёрточная нейронная сеть с остаточными блоками). Основной идеей ResNet является суммирование не изменённых входных данных с нелинейно преобразованными через несколько слоев. Это сразу же привело к стабильной обучаемости сетей глубиной в 100, а в последствии и в 1000 слоёв на достаточно сложных данных.

В настоящее время сверточные нейронные сети активно применяются в системах компьютерного зрения. Они позволяют улучшить качество распознавания и обнаружения необходимых объектов. Например, для детектирования человека по лицу [28, 37], для определения местоположения и навигации автономных роботов на заранее неизвестной местности [30, 85].

В 2016 году предложен способ применения сверточных нейронных сетей для работы с последовательными данными, и позволяющий заменить в некоторых задачах рекуррентные нейронные сети [69, 80]. Такие сети называются временные сверточные сети (Temporal Convolutional Networks, TCN).

В 2017 году исследователями из Google Brains была представлена структура Transformer [27]. Она, также, как и рекуррентные нейронные сети, предназначена для обработки последовательностей. Эта структура хорошо показала себя при работе с текстами, и сформированная на ней языковая модель GPT3 [54] показывает впечатляющие результаты. Например, эта модель используется в чат-боте ChatGPT, который активно обсуждается в настоящее время, благодаря генерируемым текстам, по качеству приближенным к человеческим. Такого эффекта удалось достичь благодаря дополнительному обучению с подкреплением, позволяющему использовать опыт человека при обучении [50].

Современные структуры нейронных сетей для работы с последовательностями, в том числе TCN и Transformer, рассчитаны на параллельную обработку фиксированной, статической информации. Это позволяет анализировать данные, просто учитывая их расположение, при этом строится связь данных, не обращая внимания на их близость. Такой подход требует наличия уже всей последовательности для анализа (например, абзац в тексте, книга). По этой причине данный тип нейронных сетей применяется для анализа и синтеза текста, переводов, то есть в сфере обработки естественного языка (NLP).

Регулятор в системах управления может опираться только на уже существующую историю сигналов или на вектор, представляющий в каком-то виде текущее состояние объекта управления. Таким образом, нужно рассматривать такие структуры нейронных сетей, которые позволяют работать в данных условиях: многослойные нейронные сети прямого распространения (за счет введения задержек в обратные связи и сигнал управления системой) и рекуррентные нейронные сети (являются динамическим объектом и при обучении позволяют формировать внутреннее состояние для расчета сигнала управления на текущем шаге).

### **1.3. Методы обучения искусственной нейронных сетей**

Обучение искусственной нейронной сети – это процесс корректировки весов, при которых сеть пытается достигнуть лучших результатов по заданному критерию. Обучение происходит на тренировочных данных или по критериям, которых она должна добиться.

После обучения нейронная сеть становится способна выполнять свою задачу на новых данных, которых не было в тренировке.

Есть следующие методы обучения [43]: обучение с учителем, обучение без учителя, обучение с подкреплением. Данные методы обучения могут быть скомбинированы в зависимости от требуемой задачи.

В **обучении с учителем** используется обучающая выборка, которая содержит в себе входные данные и эталонные данные. Такую выборку формирует проектировщик нейронной сети, «учитель». Задачей нейронной сети является поиск зависимостей между входными и эталонными данными по заданному критерию, который, обычно, выражается в виде функции потерь (loss function). Функция потерь — это функция, которая в результате сравнения полученных данных от нейронной сети на основе тренировочной выборки и эталонных значений выдает число, означающее ошибку между данными. В методах корректировки весов, основанных на градиентном спуске, данная функция должна

быть дифференцируема. После обучения искусственная нейронная сеть представляет из себя уравнение регрессии, что позволяет использовать ее на новых (которых не было в обучении) данных для прогнозирования выхода.

Обучение **без учителя** также называется самообучение. В данном методе сеть не имеет информации о конечном результате. Чаще всего используется для разделения по группам входных объектов на основе информации о них (задача кластеризации и задача обобщения).

Обучение **с подкреплением** [43, 79] применяется, когда имеется возможность оценить работу нейронной сети в виде критериев поощрения и штрафов. Обычно данный метод рассматривается в виде агента, взаимодействующего с некоторой средой. В процессе обучения среда дает отклик (сигналы подкрепления и обратную связь со своим текущим состоянием) о верности действий агента (рисунок 1.7).

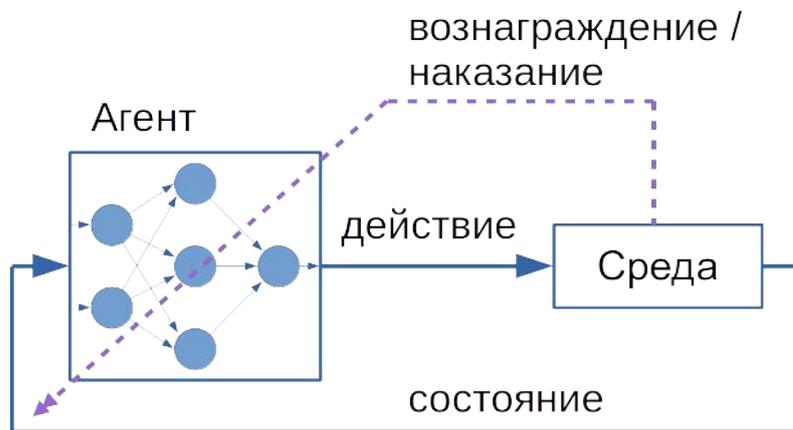


Рисунок 1.7 - Схема обучения с подкреплением

Этот метод является частным случаем обучения с учителем, только учителем является среда или ее модель, и ошибка обучения формируется в процессе обучения по заданному критерию, а не вычисляется на основе заранее записанных требуемых результатов. Если рассматривать это обучение с точки зрения САУ, то тогда замкнутую систему управления с нейрорегулятором можно рассматривать как агента, а откликом среды является оценка анализа динамических процессов замкнутой системы управления.

В работе используются обучение с учителем и обучение с подкреплением, так как имеется возможность записывать данные динамики объекта управления, которые можно использовать для формирования обучающей выборки, а также имеются критерии, которые нужно оптимизировать на основе анализа динамики замкнутой системы управления (например, время переходного процесса).

#### **1.4. Методы оптимизации параметров искусственных нейронных сетей в процессе обучения.**

Самые распространённые методы, используемые для поиска значений весов связей нейронных сетей:

- метод обратного распространения ошибки (Backpropagation),
- метод упругого распространения ошибки (Resilient propagation или Rprop),
- генетический алгоритм (Genetic Algorithm).

Метод обратного распространения ошибки [2, 87] позволяет полученную ошибку между текущим выходом нейросети и требуемым значением учесть в каждом весе нейронной сети. Данный метод основан на методе градиентного спуска.

Ошибку «распространяют» (с помощью вычисления градиента функции потерь (Loss Function)  $L$ ) от выхода к входу ИНС. Функция  $L$  учитывает ошибку каждого выхода сети по сравнению эталонным.

Для того, чтобы найти новое значение веса потребуется найти градиент по этому весу от суммарной ошибки. Затем найденный градиент умножается на параметр, называемый скоростью обучения  $\eta$  (является гиперпараметром, т.е. значение выбирается разработчиком нейронной сети на основе его опыта перед началом обучения) и найденное значение вычитается (так как требуется найти минимум функции) из текущего значения веса:

$$w_i = w_i - \eta \frac{\partial L}{\partial w_i}. \quad (1.3)$$

В результате получим смещение весов нейронной сети, которое приводит к, обычно, небольшому уменьшению значения ошибки. Для достижения хорошего результата данную процедуру следует повторить множество раз.

Для рекуррентных нейронных сетей ошибка также распространяется во времени (между шагами ее вызова). Алгоритм обучения рекуррентных нейронных сетей называется «алгоритм обратного распространения ошибки сквозь время» (Backpropagation Through Time) [86].

**Метод упругого распространения ошибки** [71] похож на метод распространения ошибки. Но в нем учитывается только знак градиента по весу и варьируется шаг изменения веса.

На каждой итерации метода проверяется значение текущего знака градиента, предыдущего знака градиента и условия, остановки обучения:

- если знак градиента не изменился, то значение шага увеличивается для более быстрого достижения результата;
- если знак градиента изменился, то значение веса следует вернуть к предыдущему и уменьшить шаг изменения веса.

Чтобы значение шага не было слишком большим или малым устанавливают ограничения.

**Генетический алгоритм (genetic algorithm)** [6] представляет собой упрощённую интерпретацию естественного отбора. Основой данного метода является поиск популяции, которая лучше всех выполнит задачу.

С помощью этого алгоритма можно жестко задать структуру и искать веса сети. Но есть алгоритмы, основанные на генетическом алгоритме, которые подбирают не только веса, но и свою топологию. Например, NEAT [78], HyperNEAT [77], SOINN.

В работе используется метод обратного распространения ошибки и его модификации, например Adam [49].

### Пример расчета метода обратного распространения ошибки для нейронной сети прямого распространения.

Для данного примера будем использовать нейронную сеть с двумя входными, двумя скрытыми и одним выходным нейронами. Так же скрытые и выходные нейроны будут иметь веса смещения (рисунок 1.8). В данной сети скрытые нейроны  $h1$  и  $h2$  имеют функцию активации – сигмоиду (1.1). Входные нейроны являются условными и используются для схематичного отображения источника данных. Нейрон последнего слоя является сумматором.

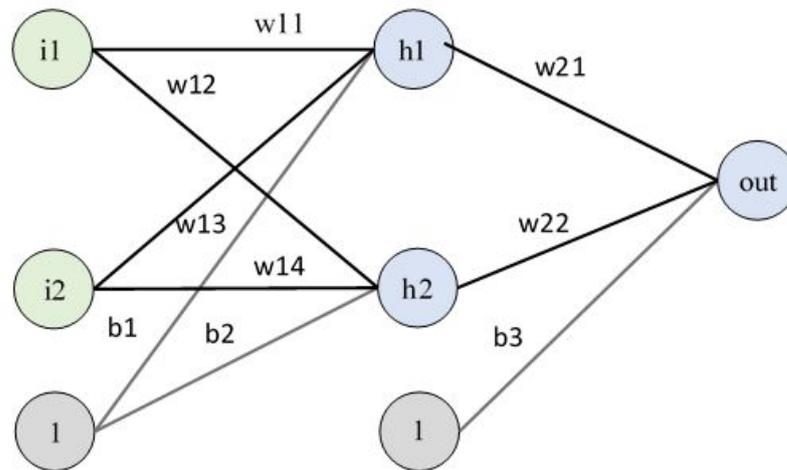


Рисунок 1.8 - Схема нейронной сети для примера расчета методом обратного распространения ошибки

Положим значения на входе  $i1 = 0.8, i2 = 0.541$ , и требуемое значение выхода сети  $target = 0.717$ .

Изначально нейронная сеть задается случайными разными значениями весов, обычно распределенными по нормальному закону распределения. Выберем начальные значения весов рассматриваемой нейронной сети (таблица 1.1):

Таблица 1.1. Начальные значения весов нейронной сети

$w_{11} = 0.15$	$w_{12} = 0.66$	$w_{21} = 0.74$
$w_{13} = 0.54$	$w_{14} = 0.34$	$w_{22} = 0.55$
$b_1 = 0.12$	$b_2 = -0.62$	$b_3 = 0.1$

Выполним расчет инициализированной, не обученной нейронной сети. Посчитаем значение выхода нейронов  $h_1$  и  $h_2$  (после функции активации):

$$h_{1_{out}} = \text{sigmoid}(i_1 \cdot w_{11} + i_2 \cdot w_{13} + b_1) = 0.63, \quad (1.4)$$

$$h_{2_{out}} = \text{sigmoid}(i_1 \cdot w_{12} + i_2 \cdot w_{14} + b_2) = 0.523. \quad (1.5)$$

Таким образом выход системы будет равен:

$$out = h_{1_{out}} + h_{2_{out}} + b_3 = 1.342. \quad (1.6)$$

Рассчитаем ошибку между полученным и требуемым результатом. Для этого воспользуемся следующей формулой:

$$E_{out} = \frac{1}{2} (target - out)^2. \quad (1.7)$$

Формула (1.7) это квадрат ошибки, деленный на 2, для небольшого упрощения дальнейших расчетов. Эта константа тут не важна, так как в итоге значение умножается на значение скорости обучения, которое является гиперпараметром. Таким образом, значение ошибки равно:

$$E_{out} = 0.5 \cdot (0.717 - 1.272)^2 = 0.143 \quad (1.8)$$

Рассмотрим  $w_{21}$ . Для того, чтобы узнать требуемое смещение веса  $w_{21}$  следует найти градиент общей ошибки по отношению к  $w_{21}$ :  $\frac{\partial E_{out}}{\partial w_{21}}$ .

Применяя правило дифференцирования сложной функции, получим:

$$\frac{\partial E_{out}}{\partial w_{21}} = \frac{\frac{\partial E_{out}}{\partial out} \cdot \partial out}{\partial w_{21}}; \quad (1.9)$$

$$\frac{\partial E_{out}}{\partial out} = -\frac{2 \cdot 1}{2} (target - out) = 0.536; \quad \frac{\partial out}{\partial w_{21}} = 1; \quad (1.10)$$

так как выходной нейрон является просто сумматором.

В итоге градиент по отношению к  $w_{21}$  равен:

$$\frac{\partial E_{out}}{\partial w_{21}} = \frac{\frac{\partial E_{out}}{\partial out} \cdot \partial out}{\partial w_{21}} = 0.536 \cdot 1 = 0.536. \quad (1.11)$$

Для уменьшения ошибки значение найденного градиента вычитается из текущего веса, умноженного на некоторую скорость обучения

$$w_{21}^+ = w_{21} - \frac{\eta \cdot \partial E_{out}}{\partial w_{21}} = 0.74 - 0.5 \cdot 0.536 = 0.472. \quad (1.12)$$

По такому же принципу рассчитаем веса  $w_{22}$  и  $b_3$ , которые станут равны:  $w_{22}^+ = 0.282$ ,  $b_3^+ = -0.168$ .

Для продолжения обучения остаются старые веса, пока не будут посчитаны все новые.

Рассмотрим вес  $w_{11}$ . Для нахождения градиента ошибки по отношению к  $w_{11}$ :

$$\frac{\partial E_{out}}{\partial w_{11}} = \frac{\frac{\partial E_{out}}{\partial h_{1_{out}}} \cdot \partial h_{1_{out}}}{\partial w_{11}}, \quad (1.13)$$

где:

$$\frac{\partial E_{out}}{\partial h_{1_{out}}} = \frac{\frac{\partial E_{out}}{\partial out} \cdot \partial out}{\partial h_{1_{out}}} \cdot \partial w_{22} = \frac{\partial E_{out}}{\partial w_{21}} \cdot w_{21} = 0.472 \cdot 0.74 = 0.396, \quad (1.14)$$

$$\frac{\partial h_{1_{out}}}{\partial w_{11}} = h_{1_{out}}(1 - h_{1_{out}}) = 0.233, \quad (1.15)$$

так как производная от логистической функции:

$$\sigma'(x) = \sigma(x) \cdot (1 - \sigma(x)). \quad (1.16)$$

В итоге для  $w_{11}$ :

$$w_{11}^+ = w_{11} - \frac{\eta \cdot \partial E_{out}}{\partial w_{11}} = 0.15 - 0.5 \cdot (0.396 \cdot 0.233) = 0.104. \quad (1.17)$$

Таким же образом найдем новые значения для  $w_{12}$ ,  $w_{13}$ ,  $w_{14}$ ,  $b_1$ ,  $b_2$ :

$$w_{12^+}=0.624; \quad w_{13^+}=0.494; \quad w_{14^+}=0.303; \\ b_{1^+}=0.074; \quad b_{2^+}=-0.657.$$

После применения новых значений весов и смещений мы получили более близкий результат к требуемому и меньшее значение ошибки

$$out = 1.032; \quad E_{out} = 0.05. \quad (1.18)$$

Таким образом, с каждой итерацией метода обратного распространения ошибки, последняя будет стремиться к нулю.

### **Пример расчета метода обратного распространения ошибки во времени для рекуррентной нейронной сети**

В качестве примера обучения рассмотрим искусственную нейронную сеть, состоящую из рекуррентного слоя и линейного слоя (1.19)

$$h_t = \tanh(W_{xh}x_t + W_{hh}h_{t-1} + b_h); \\ y_t = W_{hy}h_t + b_y; \quad (1.19)$$

где  $W$  – это матрица весов между указанными слоями;  $t$  – номер элемента в последовательности;  $h_t$  – вектор скрытого состояния рекуррентного слоя (он же является выходом этого слоя);  $b_t$  – вектор весов для нейрона смещения указанного слоя;  $x_t$  – вектор значений входного слоя;  $y_t$  – вектор значений выходного слоя;  $h_t$  – вектор выходных значений рекуррентного слоя;  $n$  - длина последовательности.

В качестве функции потерь (loss function) рассмотрим среднеквадратичную функцию потерь (MSE loss) (1.20):

$$L = \frac{1}{n} \sum_{t=1}^n (y_n - target_n)^2, \quad (1.20)$$

где  $y_t$  – результат на выходе нейронной сети на  $i$ -м индексе последовательности,  $target_t$  – требуемый выход нейронной сети.

Схема развернутого графа вычислений во времени для 2-х временных шагов приведена на рисунке 1.9.

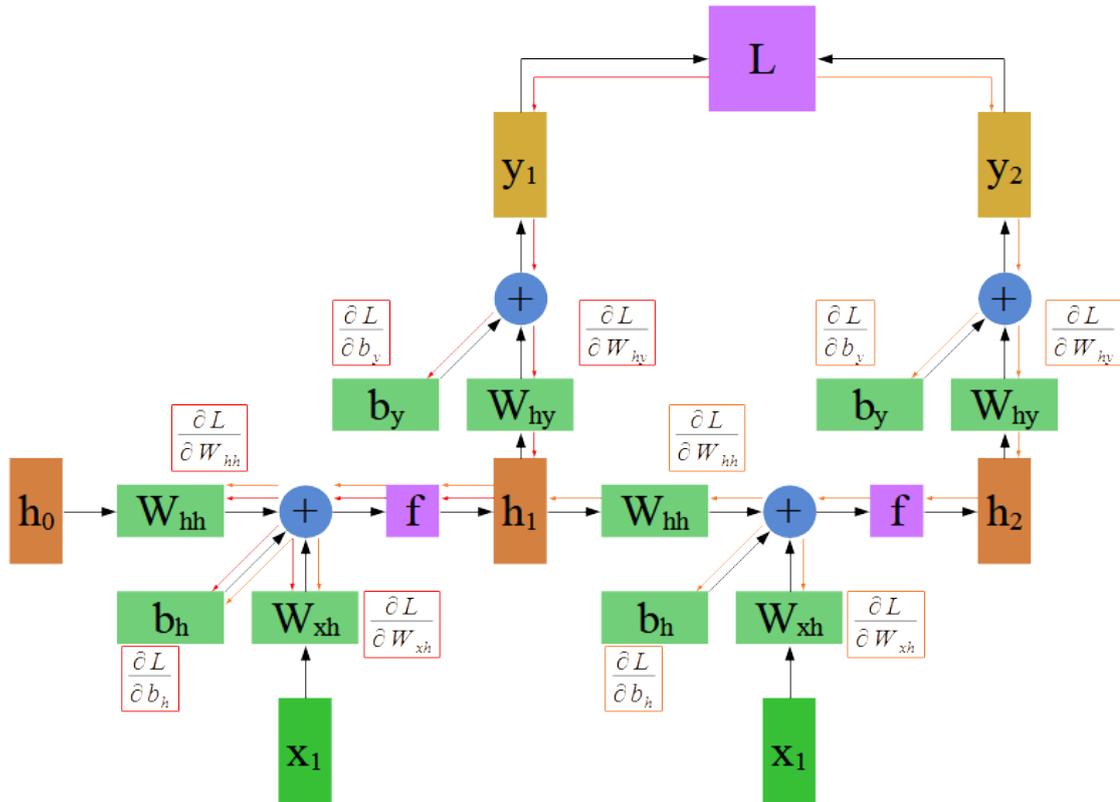


Рисунок 1.9 - Распространение градиента при наличии рекуррентного слоя

Сначала найдем  $\frac{\partial L}{\partial y_t}$  (1.21):

$$\frac{\partial L}{\partial y_t} = \frac{1}{n} (2y_t - 2 \cdot \text{target } t_t). \quad (1.21)$$

Найдем градиент для  $W_{hy}$  (1.22):

$$\begin{aligned} \frac{\partial L}{\partial W_{hy}} &= \sum_{t=0}^n \frac{\partial L}{\partial y_t} \cdot \frac{\partial y_t}{\partial W_{hy}}; \\ \frac{\partial y_t}{\partial W_{hy}} &= h_t; \\ \frac{\partial L}{\partial W_{hy}} &= \sum_{t=0}^n \frac{\partial L}{\partial y_t} \cdot h_t. \end{aligned} \quad (1.22)$$

Аналогично вычисляется для  $b_y$  (1.23):

$$\begin{aligned}
\frac{\partial L}{\partial b_y} &= \sum_{t=0}^n \frac{\partial L}{\partial y_t} \cdot \frac{\partial y_t}{\partial b_y} \\
\frac{\partial y_t}{\partial b_y} &= 1; \\
\frac{\partial L}{\partial b_y} &= \sum_{t=0}^n \frac{\partial L}{\partial y_t}.
\end{aligned} \tag{1.23}$$

Теперь вычислим градиент для  $W_{hh}$  (1.24),  $W_{xh}$  (1.26) и  $b_h$  (1.27), которые используются в каждом шаге прямого прохода нейронной сети.

$$\frac{\partial y_t}{\partial W_{xh}} = \sum_{i=0}^t \frac{\partial y_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial h_i} \frac{\partial h_i}{\partial W_{xh}} = \sum_{i=0}^n \frac{\partial y_t}{\partial h_t} \cdot \left( \prod_{j=i+1}^t \frac{\partial h_j}{\partial h_{j-1}} \right) \frac{\partial h_i}{\partial W_{xh}}, \tag{1.24}$$

так как (1.25):

$$\frac{\partial h_3}{\partial h_1} = \frac{\partial h_3}{\partial h_2} \frac{\partial h_2}{\partial h_1}. \tag{1.25}$$

Аналогично:

$$\frac{\partial y_t}{\partial W_{hh}} = \sum_{i=0}^t \frac{\partial y_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial h_i} \frac{\partial h_i}{\partial W_{hh}} = \sum_{i=0}^n \frac{\partial y_t}{\partial h_t} \cdot \left( \prod_{j=i+1}^t \frac{\partial h_j}{\partial h_{j-1}} \right) \frac{\partial h_i}{\partial W_{hh}}; \tag{1.26}$$

$$\frac{\partial y_t}{\partial b_h} = \sum_{i=0}^t \frac{\partial y_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial h_i} \frac{\partial h_i}{\partial b_h} = \sum_{i=0}^n \frac{\partial y_t}{\partial h_t} \cdot \left( \prod_{j=i+1}^t \frac{\partial h_j}{\partial h_{j-1}} \right) \frac{\partial h_i}{\partial b_h}. \tag{1.27}$$

Частные производные для  $W_{hh}$  (1.28),  $W_{xh}$  (1.29) и  $b_h$  (1.30):

$$\frac{\partial L}{\partial W_{xh}} = \sum_{i=0}^t \frac{\partial L}{\partial y_t} \cdot \frac{\partial y_t}{\partial W_{xh}}; \tag{1.28}$$

$$\frac{\partial L}{\partial W_{hh}} = \sum_{i=0}^t \frac{\partial L}{\partial y_t} \cdot \frac{\partial y_t}{\partial W_{hh}}; \tag{1.29}$$

$$\frac{\partial L}{\partial b_h} = \sum_{i=0}^t \frac{\partial L}{\partial y_t} \cdot \frac{\partial y_t}{\partial b_h}. \tag{1.30}$$

После подсчета градиентов обновляются веса (1.31):

$$\begin{aligned}
W_{hh} &= W_{hh} - lr \cdot \frac{\partial L}{\partial W_{hh}}; \\
W_{xh} &= W_{xh} - lr \cdot \frac{\partial L}{\partial W_{xh}}; \\
W_{hy} &= W_{hy} - lr \cdot \frac{\partial L}{\partial W_{hy}}; \\
b_h &= b_h - lr \cdot \frac{\partial L}{\partial b_h}; \\
b_y &= b_y - lr \cdot \frac{\partial L}{\partial b_y};
\end{aligned} \tag{1.31}$$

где  $lr$  – шаг обучения.

Применяя данный метод итерационно, ошибка с каждым шагом будет стремиться к нулю.

### 1.5. Процесс обучения искусственной нейронной сети.

Для метода обучения с учителем сначала необходимо подготовить обучающую выборку (dataset), содержащую входные значения и эталонные значения.

Желательно, чтобы данные были приведены к общему диапазону, иначе при обучении влияние одного входа будет сильнее, чем другого, что, обычно, приводит к плохому качеству обучения. Например, это актуально, когда данные, подаваемые на вход одного нейрона, меняются в диапазоне от -100 до 100, а на другой нейрон от -1 до 1. Таким образом, при обучении градиент в методе обратного распространения ошибки для первого нейрона будет больше.

В общем случае данные для обучения нормируются. Особенно это актуально для функций активации sigmoid и tanh.

Метод пропуска ошибки для получения качественного результата требует множества итераций применения. Одну итерацию называют эпохой. **Эпоха** (epoch) – это когда весь тренировочный набор данных был пропущен через нейронную сеть и полученные ошибки применены в весах нейронной сети методом обратного распространения ошибки один раз.

Для качественного обучения нейросети необходимо повторить множество эпох.

Есть 3 способа обучения на тренировочных данных:

- стохастический метод,
- пакетный (batch) метод,
- мини-пакетный метод (mini-batch).

При **стохастическом методе** обновление весов нейросети производится для каждой пары векторов входа и эталонного выхода из тренировочного набора. Используется так называемый стохастический градиентный спуск [61]. В результате за одну эпоху в среднем нейронная сеть сдвинется в сторону минимизации функции потерь. Однако, в таком методе достижение минимума может занять больше времени.

При **пакетном методе** обновление весов нейросети производится только после того, когда будут найдены все градиенты весов, которые будут суммированы в общее значение. В этом методе используется классический градиентный спуск. В отличие от предыдущего метода, в этом требуется хранить много параметров для вычисления частных производных всех данных в выборке, и не всегда возможно реализовать данный метод из-за ограничений памяти.

**Мини-пакетный метод** является чем-то средним между стохастическим методом, и пакетным методом. Вся обучающая выборка делится на несколько частей, пакетов. Обновление весов происходит на основании вычисления градиента для каждого пакета. В этом методе также используется стохастический градиентный спуск, но выбирается небольшая группа данных, что позволяет использовать плюсы стохастического и пакетного методов, т. е. реализуется баланс между необходимым количеством памяти и скоростью обучения.

При большом размере сети и маленькой обучающей выборке нейронная сеть может **переобучиться**, то есть не приобретает способности обобщать данные. Это такое состояние, когда ИНС «заучивает» тренировочные данные и

показывает отличный результат на них, но не способна хорошо обработать новые данные, которых не было в тренировочных.

При неподходящей структуре или недостаточном наборе данных нейронная сеть может **недообучиться**. Т. е. сеть плохо отрабатывает, как и тренировочные, так и новые данные.

Для выявления таких случаев данные для обучения делят на 3 части: тренировочную, валидационную и на проверочную (тестовую).

В течение каждой эпохи нейронную сеть обучают на тренировочной выборке, и в конце эпохи проверяют на валидационной выборке. Если ошибка на тренировочных данных уменьшается, а на валидационной увеличивается, то нейронная сеть переобучилась и необходимо увеличить количество примеров в обучающей выборке, изменить гиперпараметры, изменить функцию потерь или добавить к ней регуляризацию (дополнительные условия, ограничивающие обучение). В качестве регуляризации часто используют сокращение весов (weight decay), при котором штраф накладывается на слишком большие веса [41, 90]. Если обе ошибки уменьшаются, и ошибка на тренировочных данных достигла малого значения, то готовая нейронная сеть проверяется на тестовых данных. По этому результату можно сказать, насколько хорошо работает сеть.

## **1.6. Применение нейронных сетей в системах управления**

Основным направлением исследований в области искусственных нейронных сетей было и остается их развитие в сферах, которые лучше выполняются человеком, чем детерминированными алгоритмами. Это прямая работа с данными, которыми оперирует человек: изображения, видео, текст, звук. Также способности ИНС к обучению, адаптивности, параллельной обработке данных, которые являются ее преимуществом, в последнее время стали применяться для решения обширного круга задач робототехники и мехатроники [81]. Можно выделить некоторые основные направления современных

исследований в области применения нейронных сетей и глубокого обучения в задачах управления сложными динамическими системами.

1. Идентификация неопределённостей в процессе функционирования. Часто при обучении нейросетевых регуляторов допускается, что они будут применяться в условиях закрытого множества, т. е. классы сигналов, встречающихся в процессе эксплуатации, известны и такие же, как и во время обучения. Однако, объекту управления часто приходится работать в постоянно меняющихся и неконтролируемых условиях реального мира, что может привести к режимам работы, которые не были охвачены обучающими данными. Исследованию проблем идентификации неопределённостей посвящены работы [25, 47].

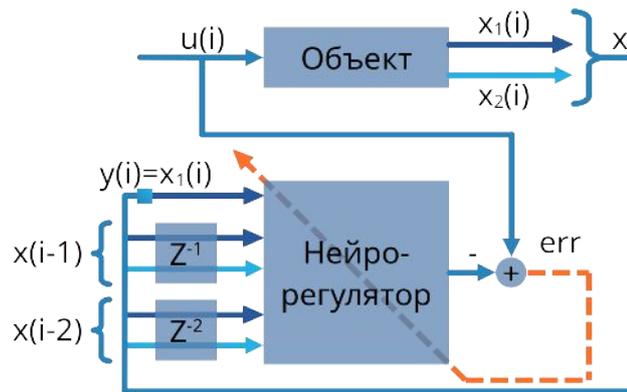
2. Выбор архитектуры нейронной сети для нейрорегулятора.

Основными архитектурами ИНС для применения в нейрорегуляторах считаются многослойные, рекуррентные и радиально-базисные нейронные сети. К настоящему времени разработаны типовые схемы систем автоматического управления на основе нейрорегуляторов [23]. Эти схемы используют архитектуру многослойной нейронной сети прямого распространения с временными задержками на входе [53, 66]. Рассмотрим эти схемы.

### **Обобщенное инверсное нейруправление.**

В обобщенном инверсном нейруправлении в качестве регулятора используется нейронная сеть, обученная имитировать инверсную динамику объекта управления (рисунок 1.10).

## Обучение



## Применение

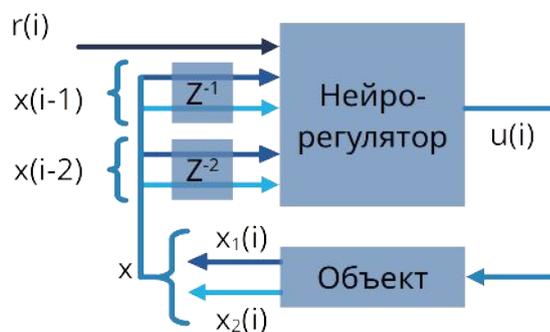


Рисунок 1.10 - Схема обучения нейрорегулятора в методе инверсного нейруправления

Нейрорегулятор обучается выдавать управляющий сигнал  $u$  (1.32), используя информацию текущего и предыдущих выходов системы  $x$ , иногда, предыдущие управляющие значения в качестве входного вектора:

$$u(i) = f(x(i), x(i-1), \dots, x(i-n)), \quad (1.32)$$

где  $i$  — номер текущего такта,  $n$  — количество входных нейронов, на которые подается задержанный сигнал,  $x$  — обратные связи от объекта управления. В случае дискретных моделей задержки берутся равными такту квантования.

Обученная ИНС при управлении объектом подключается как регулятор, получая на вход задающее значение  $r$  и текущее значение объекта управления  $x(i)$ , на основе которых формируется управляющее значение  $u(i)$ , которое приведет объект к реакции  $y(i+1)$  равной задающему значению  $r$ .

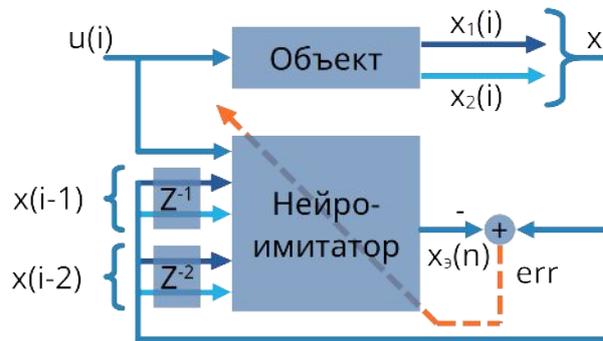
Недостатком этого метода является трудность в составлении обучающей выборки, выраженная в подборе идентифицирующего процесса. Также трудности могут быть, если инверсия объекта управления является неоднозначной функцией. Это может привести процесс обучения в тупик.

Однако, этот метод обладает важным достоинством: ему не требуется математическая модель объекта, а только записанные данные.

**Метод обратного пропуска ошибки через прямой нейроэмулятор (нейроимитатор).**

В методе обратного пропуска ошибки через прямой нейроэмулятор (также называемый нейроимитатором) используются две нейронные сети, одна из которых выполняет функцию регулятора, а вторая – модели объекта управления, которая называется прямой нейроимитатор. Схема обучения и управления представлена на рисунке 1.11.

### Обучение нейроимитатора



### Обучение нейрорегулятора

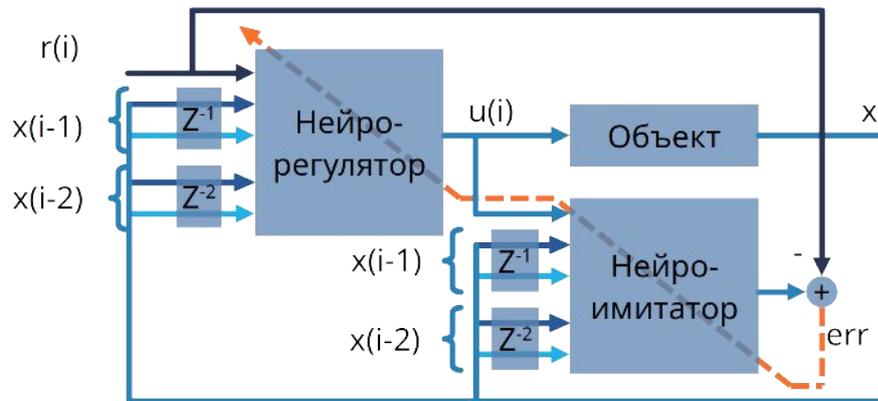


Рисунок 1.11 - Схема обучения в методе обратного пропускания ошибки через нейроимитатор

Нейроимитатор обучается повторять поведение объекта управления. При обучении входом являются сигналы, которые позволяют полностью идентифицировать объект. Желаемым выходным значением ИНС является значение реакции объекта.

После обучения нейроимитатора происходит обучение нейрорегулятора. На вход нейрорегулятора на  $k$ -ом такте поступает желаемое положение объекта управления на следующем такте  $r(i+1)$ , при этом на выходе регулятора формируется управляющее воздействие  $u(i)$ . После отработки воздействия  $u(i)$ , объект управления переходит в состояние  $y(i+1)$ , а нейроимитатор выдает значение  $y'(i+1)$ . На основании этих сигналов вычисляется ошибка регулирования  $e(i+1) = r(i+1) - y'(i+1)$ , которая по методу обратного распространения проходит через нейроимитатор и нейрорегулятор. При этом корректируются веса связей

нейрорегулятора, а веса нейроимитатора не изменяются. Т.е. нейроимитатор учитывается в качестве дополнительных слоев нейрорегулятора с замороженными весами.

Так же такая структура предусматривает обучение в реальном времени, что позволяет адаптировать регулятор к изменяющимся условиям работы.

Недостатком данного метода является обучение сразу двух нейронных сетей. Причем нейрорегулятор обучается в режиме онлайн, что не со всеми объектами возможно выполнять, так как в начале обучения могут появляться неверные управляющие воздействия, которые могут повредить объект управления. Далее в работе показано, что этот недостаток может быть устранен, например, путем начального обучения нейрорегулятора на упрощенной приближенной математической модели, а затем обучение может быть продолжено в режиме онлайн. Другой подход заключается в предварительном обучении нейрорегулятора с помощью инверсного нейроуправления, а, далее, использовать метод обратного пропуска ошибки через прямой нейроэмулятор.

#### **Прогнозирующее модельное нейроуправление.**

Методы обучения нейросетевых регуляторов, при которых минимизируется отклонение текущего выхода объекта управления от входного воздействия, не всегда хорошо справляются с задачей регулирования. Во-первых, если переходный процесс длится дольше, чем один такт, то нейронная сеть, стремясь минимизировать рассогласование, может выдать большое управляющее воздействие, что приведет к перерегулированию. Во-вторых, в таких системах всегда существует запаздывание из-за обратной связи, что ухудшает качество переходных процессов.

В связи с этим, был разработан метод прогнозирующего модельного нейроуправления, который минимизирует функционал стоимости интегральной ошибки, прогнозируемой на несколько тактов вперед. Схема метода показана на рисунке 1.12.

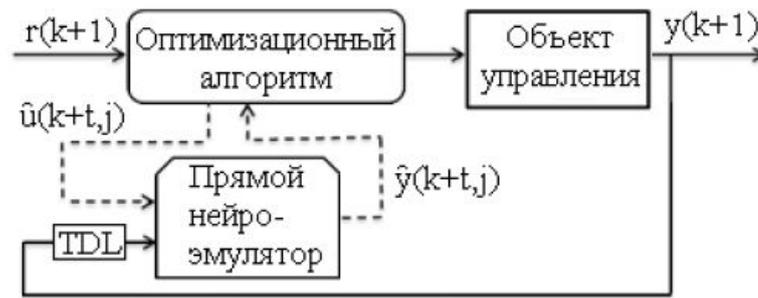


Рисунок 1.12 - Схема прогнозирующего модельного нейруправления

[23]

Для прогнозирования поведения объекта управления и вычисления ошибок управления используется нейроимитатор, обученный также, как в методе обратного распространения ошибки через нейроимитатор. Отличие состоит в отсутствии обучаемого нейрорегулятора. Вместо него используется оптимизационный модуль, работающий в режиме реального времени, который на такте  $k$  формирует желаемую траекторию на несколько тактов вперед. После этого он подает на нейроимитатор различные управляющие воздействия, прогнозирует поведение объекта управления и вычисляет наилучшую стратегию управления. В результате выбирается оптимальное управляющее воздействие, приводящее объект управления к нужному положению на целевой траектории.

Недостатком данного метода является невозможность его применения в системах с высокой частотой дискретизации, т. к. оптимизационный модуль за один такт может не успевать находить оптимальное управляющее воздействие.

### **Гибридное нейруправление.**

Гибридным нейруправлением называется совместное функционирование нейросетевых и стандартных регуляторов классической теории управления.

Одним из видов гибридного нейруправления является параллельное гибридное нейруправление, схема которого представлена на рисунке 1.13.

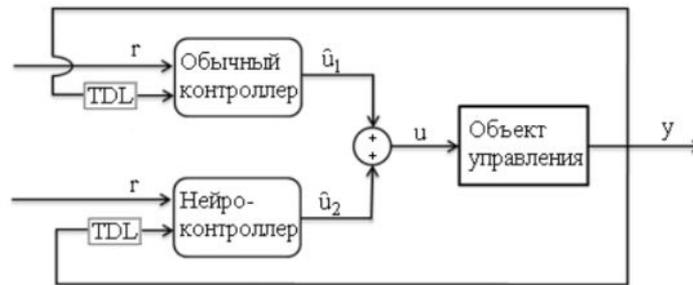


Рисунок 1.13 - Схема параллельного гибридного нейруправления [23]

Существует несколько способов обучения и подключения нейросетевого регулятора к системе управления. Например, сначала контур управления объектом замыкается классическим регулятором. Далее, нейрорегулятор учится управлять этой замкнутой системой, после чего он подключается к системе и управляющие воздействия нейросетевого и обычного регуляторов складываются.

Либо в другом порядке: сначала контур управления замыкается нейросетевым регулятором, который проходит обучение. После этого настраивается классический регулятор для управления этой системой. Он подключается к системе и управляющие воздействия обоих регуляторов складываются.

Третьим вариантом параллельного гибридного нейруправления является разграничение областей действия классического регулятора и нейросетевого. В результате на объект управления поступает управляющее воздействие либо от одного регулятора, либо от другого в зависимости от состояния объекта управления.

Параллельное гибридное нейруправление является самым простым решением для перехода от обычных регуляторов к нейросетевым.

Другим методом гибридного нейруправления является гибридное нейро-ПИД управление (или нейроморфное управление), сутью которого является самонастройка ПИД-регулятора с помощью искусственных нейронных сетей в режиме онлайн. Схема данного метода представлена на рисунке 1.14. На входе нейронная сеть получает значение требуемого положения объекта и предыдущего

его состояния. На выходе получают коэффициенты управления ПИД-регулятора  $K_p, K_i, K_d$ .

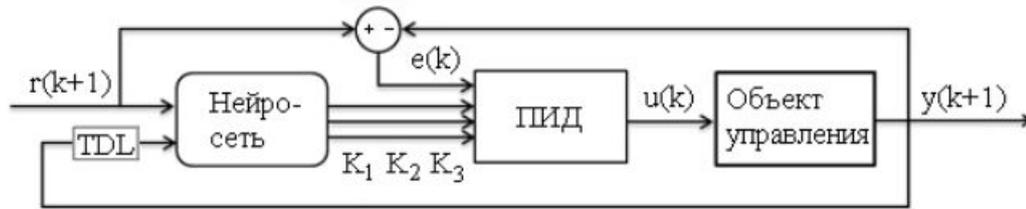


Рисунок 1.14 - Схема гибридного нейро-ПИД управления [23]

Сгенерированные значения параметров вместе с текущей ошибкой управления поступают на ПИД-регулятор, который в свою очередь генерирует управляющее воздействие на объект управления.

Достоинством этого метода является то, что ПИД-регулятор становится нелинейным, а это обеспечивает более высокое качество управления нелинейными объектами.

Как видно, представленные типовые схемы предлагают общие способы использования нейронных сетей в контуре систем автоматического управления, и каждая имеет свои недостатки и не является универсальной. Так же в них не предлагается конкретных алгоритмов и методов формирования структуры искусственных нейронных сетей и формирования обучающей выборки.

В результате революции глубокого обучения появились новые функции активации, изменены и доработаны методы обучения нейронных сетей, появились новые архитектуры нейронных сетей. Также активно ведутся исследования в области обучения с подкреплением с использованием глубоких ИНС [63, 67]. При этом описанные новые разработки слабо представлены при использовании нейронных сетей в системах автоматического управления.

В работе используются два метода синтеза нейросетевого регулятора: обобщенное инверсное нейроуправление и метод обратного пропуска ошибки через нейроимитатор, на основе которых были разработаны новые методы и

алгоритмы синтеза, использующие некоторые последние наработки из области глубокого обучения.

### 3. Устойчивость нейросетевого управления.

Несмотря на многочисленные примеры успешного применения ИНС в задачах управления динамическими системами, нейросетевые системы управления все еще не имеют теоретических обоснований (гарантий) их эффективности и устойчивости, что может препятствовать их применению во многих критически важных областях, связанных с безопасностью. Основное объяснение этому кроется в их репутации "черных ящиков", поведение которых невозможно предсказать. Существующие подходы к оценке обученных моделей в основном основаны на тестировании с использованием наборов данных [34, 83]. Однако, тестирование показывает наличие, а не отсутствие ошибок. В связи с этим нейросетевые регуляторы не используются для критически важных и опасных объектов, например, для управления автономными летательными аппаратами.

Для устранения этого недостатка некоторые исследователи пытаются использовать формальные методы. Например, в работе [89] впервые предложен метод проверки простых нейронных сетей с одним скрытым слоем. Однако, все подходы в рассмотренных работах [70, 83] имеют один и тот же недостаток, они не масштабируются для больших сетей, встречающихся во многих реальных приложениях.

Таким образом, важнейшим критерием, который в настоящее время не может быть проверен для нейросетевых регуляторов, является устойчивость замкнутой системы. Устойчивость по Ляпунову [5] дает гарантию предсказуемого поведения замкнутой системы для любого состояния внутри области притяжения [7].

Существуют различные подходы к численному построению функций Ляпунова [75] для нелинейных систем с нейросетевым управлением [65, 70]. Многие подходы основаны на полиномиальной аппроксимации динамики и

поиске суммы квадратов полиномов в качестве функций Ляпунова с помощью полуопределенного программирования (SDP) [70]. Такой подход исследовался в большом количестве теоретических работ, однако на практике полиномиальные приближения накладывают значительные ограничения на системы и структуру функций Ляпунова. Более того, хорошо известны проблемы численной чувствительности в SDP [65], что делает поиск решений, полностью удовлетворяющих условиям Ляпунова, очень трудным. Также имеются подходы, в которых предложено использовать нейронную сеть в качестве функции Ляпунова [31, 56, 58]. Однако, в этих работах показывается функционирование данного метода для гладких нелинейностей и не указаны способы формирования структуры самих нейронных сетей.

Таким образом, одной из актуальных проблем является формирование метода синтеза нейрорегулятора, позволяющего гарантировать асимптотическую устойчивость замкнутой системы в заданной локальной области фазового пространства, в том числе для объектов с ограничителями.

## 1.7. Выводы

- Определены актуальные прикладные области и способы применения нейронных сетей в системах автоматического управления.
- Проанализированы известные структуры искусственных нейронных сетей и способы их обучения в контексте использования в системах автоматического управления. Перспективными были выбраны нейронные сети прямого распространения и рекуррентные нейронные сети. В качестве способов обучения выбраны обучение с учителем и обучение с подкреплением.
- Проанализированы существующие методы формирования систем управления с использованием искусственных нейронных сетей и выявлены преимущества и недостатки каждого из них. Выделены перспективные

методы синтеза, позволяющие получить нейронную сеть, выполняющую роль регулятора.

## 2. Методы синтеза нейросетевых регуляторов для объектов с ограничителями

### 2.1. Выбор рациональной структуры нейросетевого имитатора и нейросетевого регулятора на основе нейронных сетей прямого распространения

Как правило ИНС реализуются на цифровых устройствах, и по этой причине нейросетевой регулятор можно считать дискретным. Если рассматривать линейный объект управления, то его разностное уравнение имеет вид (2.1):

$$y(i) + a_1 y(i-1) + \dots + a_K y(i-K) = b_0 u(i) + b_1 u(i-1) + \dots + b_M u(i-M), \quad (2.1)$$

где  $a_k$  и  $b_m$  – коэффициенты числителя и знаменателя  $Z$ -передаточной функции соответственно,  $k=0, 1 \dots K$ ,  $m=0, 1 \dots M$ ,  $a_0=1$ ,  $u$  – сигнал на входе объекта управления,  $y$  – выход объекта управления (один из элементов вектора состояния  $x$ ),  $i$  – текущий такт квантования.

Выразим  $x(i)$ :

$$y(i) = b_0 u(i) + b_1 u(i-1) + \dots + b_M u(i-M) - a_1 y(i-1) - \dots - a_K y(i-K), \quad (2.2)$$

Таким образом структура нейронной сети может быть основана на уравнении (2.2), т. е. для имитации линейных дискретных объектов достаточно одного нейрона без функции активации (рисунок 2.1).

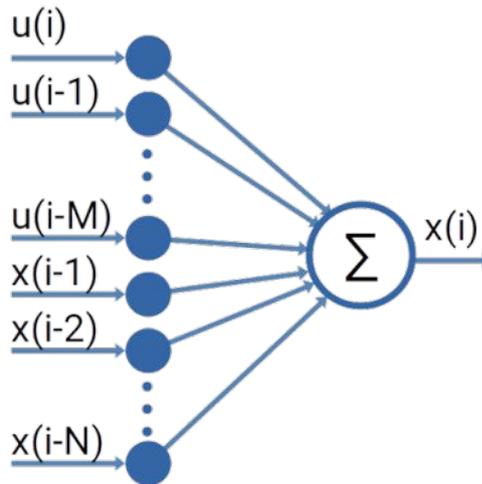


Рисунок 2.1 - Схема искусственной нейронной сети с одним нейроном

При включении искусственной нейронной сети в контур в качестве регулятора на ее вход подается полезный управляющий сигнал вместо главной обратной связи (рисунок 1.10). Так как после обучения такой регулятор должен выдавать сигнал, который на следующий такт приведет систему в заданное значение, то  $y(i+1) = r(i)$ ,  $r$  – полезный входной сигнал. Следовательно

$$y(i+1) = b_0 u(i+1) + b_1 u(i) + \dots + b_M u(i-M+1) - a_1 y(i) - \dots - a_K y(i-K+1). \quad (2.3)$$

Так как  $b_0$  в реальных системах обычно равен нулю, то

$$y(i+1) = b_1 u(i) + \dots + b_M u(i-M+1) - a_1 y(i) - \dots - a_K y(i-K+1). \quad (2.4)$$

Выразим значения управления  $u(i)$  из (2.4) и заменим  $y(i+1) = r(i)$

$$u(i) = \frac{r(i) + a_1 y(i) + \dots + a_K y(i-K+1) - b_2 u(i-1) - \dots - b_M u(i-M+1)}{b_1}. \quad (2.5)$$

Таким образом, согласно уравнению (2.5), возможно сформировать дискретный нейрорегулятор, состоящий из одного нейрона без функции активации для управления линейным объектом, который приведет систему в заданное состояние за один такт. Также из разностного дискретного уравнения объекта можно определить необходимое количество входов нейронной сети.

### **Пример обучения ИНС в качестве нейроимитатора линейного объекта.**

В качестве простого примера рассмотрим объект управления в виде колебательного звена (2.6).

$$W(s) = \frac{k}{T^2 s^2 + 2T \xi s + 1}. \quad (2.6)$$

Представим объект в виде дискретной передаточной функции:

$$W(z) = \frac{b_0 z^2 + b_1 z + b_2}{z^2 + a_1 z + a_2}. \quad (2.7)$$

Таким образом, выход системы вычисляется следующим образом в виде разностного уравнения:

$$y(n) = b_0 u(n) + b_1 u(n-1) + b_2 u(n-2) - a_1 y(n-1) - a_2 y(n-2). \quad (2.8)$$

Из уравнения (2.8) видно, что на вход нейронной сети, структура которой состоит из двух слоев, необходимо подать текущее и два задержанных значения выхода объекта, текущее и задержанные управляющие значения.

В первом варианте структура ИНС соответствовала уравнению (2.8). После обучения веса нейронитатора стали близки к ожидаемым, т. е. близки к значениям коэффициентов в уравнении (2.8). Результат моделирования нейронитатора и объекта управления представлен на рисунке 2.2.

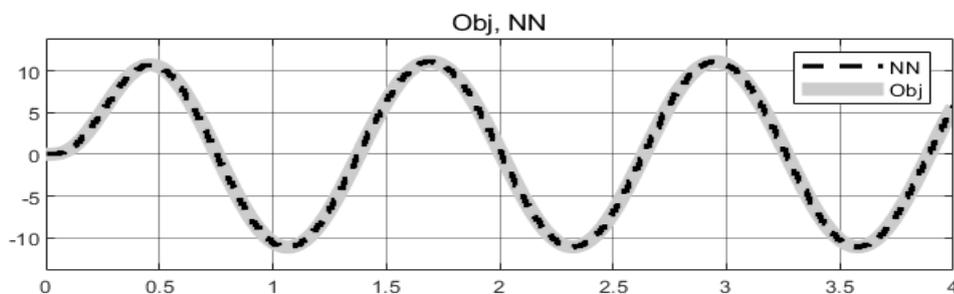


Рисунок 2.2 - Выход нейронитатора и выход объекта

Так же были проведены эксперименты с другими размерами входного слоя и другим количеством слоев. Например, в структуру нейронной сети добавлялся скрытый слой с 30 нейронами и функцией активации ReLU, а на вход подавалось 4 задержанных состояния объекта, сигнал управления с 3-мя задержанными значениями. Обучение происходило гораздо дольше, но результат остался таким же, как со структурой ИНС по уравнению (2.8). Результат моделирования данного нейронитатора представлен на рисунке 2.3.

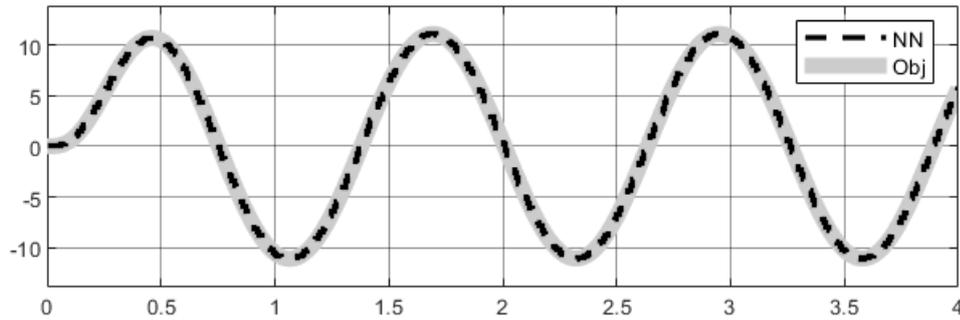


Рисунок 2.3 - Выход нейронимитатора с дополнительным скрытым слоем и выход объекта

Таким образом, нейронная сеть с количеством входов не меньше, чем  $2n$ , где  $n$  – порядок ОУ, способна качественно повторять динамику линейного объекта.

**Пример синтеза нейрорегулятора для управления линейным объектом методом инверсного нейроуправления.**

В качестве объекта рассмотрим, как и в предыдущем примере, колебательное звено. Согласно уравнению (2.5) для объекта управления получим:

$$u(n) = \frac{y(n+1) + a_1 y(n) + a_2 y(n-1) - b_2 u(n-1)}{b_1}. \quad (2.9)$$

Из рисунка 2.4 видно, что при моделировании системы с регулятором, описываемым уравнением (2.9) система имеет высокую колебательность. Однако, если рассматривать эту систему только в моменты дискретизации, то видно, что объект установился в требуемое значение за один такт квантования.

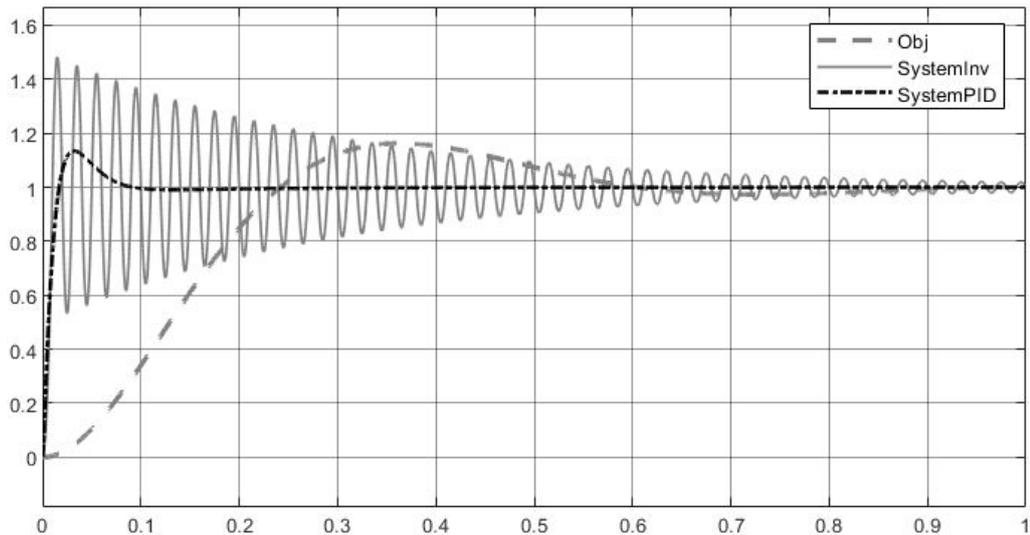


Рисунок 2.4 - Реакция на ступенчатый сигнал: объекта управления, системы с регулятором по уравнению (2.9) и системы с ПИД регулятором

Для обучения нейронных сетей использовался сбор данных с объекта (моделирование в Simulink) на гармоническом, ступенчатом и случайном сигналах. При экспериментах было замечено, что при небольшом количестве, или отсутствии ступенчатых сигналов в наборе данных для обучения система имеет заметную статическую ошибку. При большом количестве случайных сигналов нейронная сеть плохо обучается и имеет большую ошибку слежения.

Результат работы системы автоматического управления с обученным нейрорегулятором по методу «инверсного управления» представлен на рисунке 2.5.

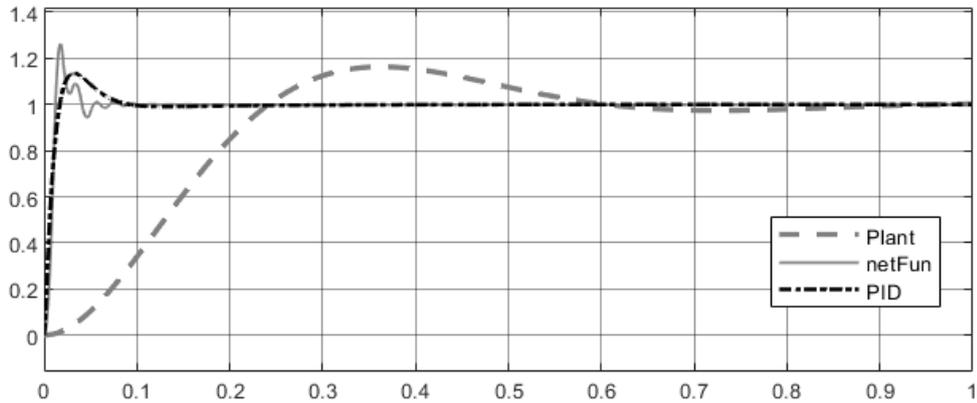


Рисунок 2.5 - Реакция на ступенчатый сигнал: объекта управления, системы с нейрорегулятором, обученным по уравнению (2.9) и системы с ПИД регулятором

Как видно из рисунка 2.5, нейрорегулятор может управлять системой. Однако, при таком методе нельзя контролировать качество переходных процессов в процессе обучения.

## 2.2. Класс нелинейностей

В работе рассматриваются наиболее часто встречающиеся нелинейности — ограничители, такие как: ограничители типа насыщение (рисунок 2.6) и типа жесткий механический упор (рисунок 2.7).

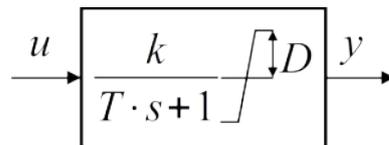


Рисунок 2.6 - Звено с ограничителем типа насыщение

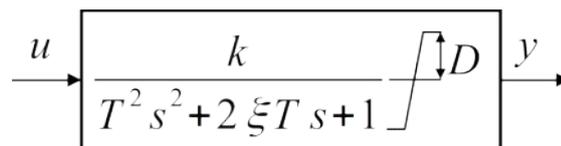


Рисунок 2.7 - Звено с ограничителем типа жесткий механический упор

Звено первого типа описывается следующим образом (2.10):

$$\dot{x} = \begin{cases} \frac{(ku-x)}{T}, & \text{если } |x| < D \text{ или } |x| = D \text{ и } \frac{(ku-x)}{T} \text{sign}(x) \leq 0; \\ 0, & \text{если } |x| = D \text{ и } \frac{(ku-x)}{T} > 0. \end{cases} \quad (2.10)$$

Звено с ограничителем типа жесткий механический упор имеет следующее описание (2.11):

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \begin{cases} \frac{ku-x_1}{T^2} - \frac{2\xi x_2}{T}, & \text{если } |x_1| < D \text{ или } |x_1| = D \text{ и } \text{sign}(x_1) \leq 0; \\ 0, & |x_1| = D \text{ и } (ku-x_1) \text{sign}(x_1) > 0. \end{cases} \end{aligned} \quad (2.11)$$

При этом предполагается, что удар об упор является абсолютно неупругим и сход с ограничителя является непрерывным:

$$\begin{aligned} x_1(t^*+0) &= x_1(t^*-0), \\ x_2(t^*+0) &= 0. \end{aligned} \quad (2.12)$$

Как видно, объект, содержащий ограничители, представляет собой нелинейную систему специального вида, динамика которой описывается дифференциальными уравнениями с разрывными правыми частями, причем разрывными могут быть и фазовые траектории. Такие объекты обладают рядом специфических особенностей, вследствие чего синтез регуляторов для них является сложной задачей.

Для имитации таких объектов и управления ими в работе используются нейронные сети с нелинейной функцией ReLU [62], которая описывается следующим уравнением (2.13) и представлена на рисунке 2.8:

$$f(x) = \begin{cases} x, & x \geq 0; \\ 0, & x < 0. \end{cases} \quad (2.13)$$

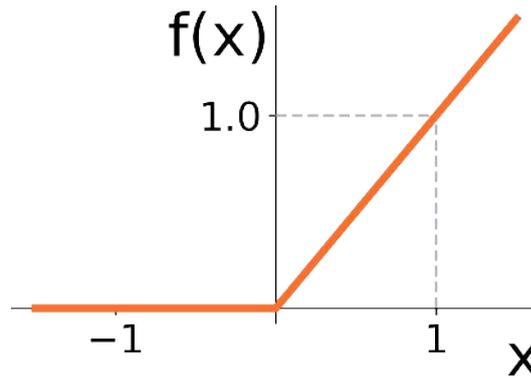


Рисунок 2.8 - График функции ReLU

Для имитации рассмотренных звеньев с ограничителями предлагаются нейросетевые структуры, позволяющие точно повторять динамику указанных объектов. Интегратор с ограничителем, звено с ограничителем типа насыщение (рисунок 2.6) и звено с ограничителем типа жесткий механический упор (рисунок 2.7) можно представить в виде нейронных сетей прямого распространения с функциями активациями ReLU.

Ограничение фазовой переменной

$$y(x) = \begin{cases} x, & \text{если } -D \leq x \leq D; \\ D, & \text{если } x \geq D; \\ -D, & \text{если } x \leq -D; \end{cases} \quad (2.14)$$

можно выразить, используя функцию активации  $\text{relu}$ :

$$y(x) = -\text{relu}(x - D) + \text{relu}(x) - \text{relu}(-x) + \text{relu}(-x - D),$$

где положительная линейная часть  $\text{relu}(x)$ , отрицательная линейная часть  $-\text{relu}(-x)$ , положительное (верхнее) ограничение  $-\text{relu}(x - D) + \text{relu}(x)$ , отрицательное (нижнее) ограничение  $-\text{relu}(-x) + \text{relu}(-x - D)$ .

Интегратор с ограничителем может быть представлен с помощью 4 нейронов в скрытом слое с функциями активации ReLU (рисунок 2.9).

Для звена с ограничителем типа насыщение также необходимо 4 нейрона в скрытом слое с функцией активации ReLU (рисунок 2.10). Значения весовых коэффициентов взяты из дискретного разностного уравнения линейной части звена.

Для звена с ограничением типа жесткий механический упор необходимо 4 нейрона на входе, 4 нейрона в скрытом слое с функцией активации ReLU и один выходной нейрон (рисунок 2.11). Чтобы получить фазовую переменную  $x_2$  в скрытый слой следует добавить еще 2 нейрона с функцией активации ReLU, которые будут компенсировать нелинейности друг друга, а в выходном слое добавить нейрон с единичными коэффициентами.

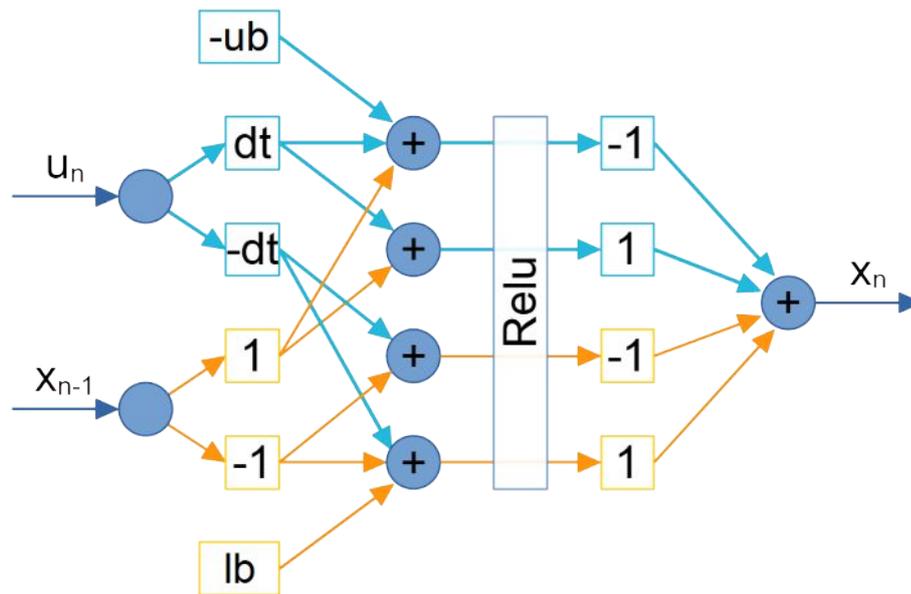


Рисунок 2.9 - Схема нейронной сети, точно повторяющей динамику дискретного интегратора с ограничителем ( $dt$  – время дискретизации,  $ub$  и  $lb$  – значение верхнего и нижнего ограничения соответственно)

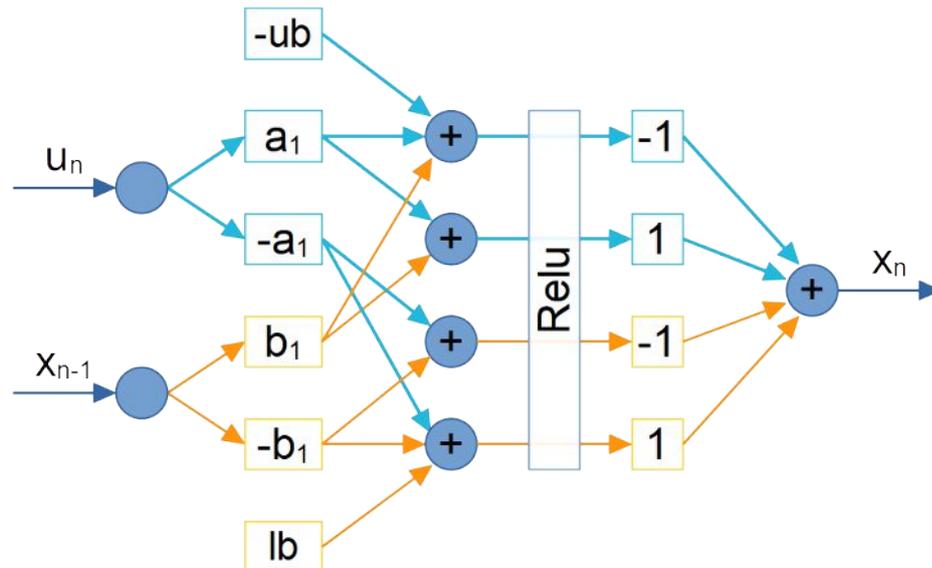


Рисунок 2.10 - Схема нейронной сети, точно повторяющей динамику дискретного звена с ограничителем типа насыщение. (коэффициенты  $a_1$  и  $b_1$  – значения коэффициентов из разностного уравнения,  $ub$  и  $lb$  – значение верхнего и нижнего ограничения соответственно)

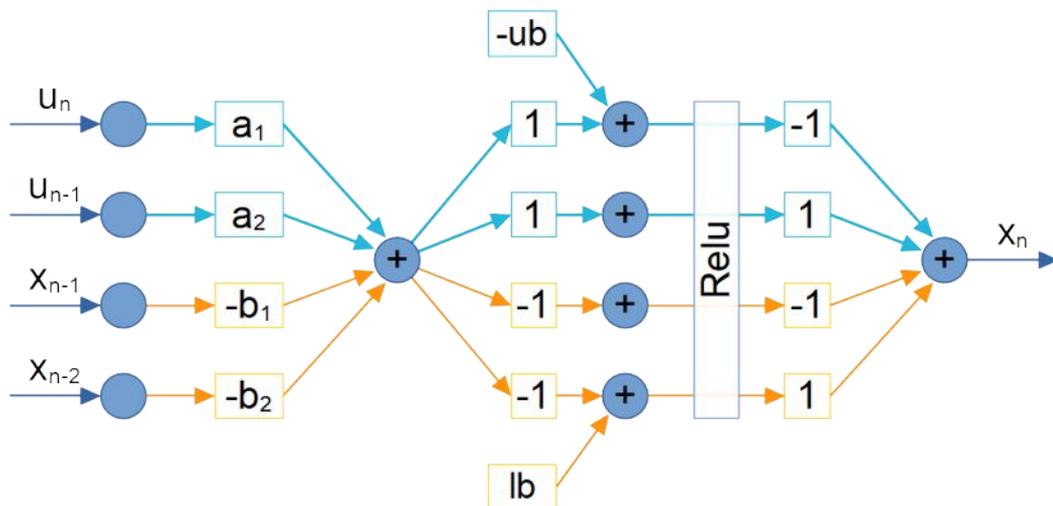


Рисунок 2.11 - Схема нейронной сети, точно повторяющей динамику дискретного звена с ограничителем типа жесткий механический упор. (коэффициенты  $a_1$ ,  $a_2$ ,  $b_1$ ,  $b_2$  – значения коэффициентов из разностного уравнения,  $ub$  и  $lb$  – значение верхнего и нижнего ограничения соответственно)

Из схем можно определить вид структуры и размеры нейронных сетей для объектов с ограничителями. Например, для объекта, состоящего из

последовательно соединенных интегратора с ограничителем и звена с ограничителем типа насыщение (2.10) для точного описания нейронной сетью требуется минимум 8 нейронов в скрытом слое с функцией активации ReLU. При этом на вход объекта подается полный вектор состояния и значение управления. А на выходе такой нейронной сети формируется полный вектор состояния на следующий дискретный такт.

Для неограниченных фазовых переменных достаточно одного нейрона в скрытом слое (вместо 4-х для ограниченных). Однако, так как в этом слое должна присутствовать функция активации ReLU (обычно в нейронных сетях одна функция активации на весь слой), то вместо одного требуется 2 нейрона, принимающие на вход одинаковые значения по модулю с разными знаками. В таком случае две нелинейные функции активации ReLU компенсируют друг друга, оставляя значение неизменным (т. е. фактически это равно одному нейрону без функции активации).

Таким образом, можно вывести следующую зависимость минимального размера скрытого слоя нейронной сети, выступающей в качестве нейроимитатора:

$$n_{hidden} = 4 n_{sat} + 2 n_{lin}, \quad (2.15)$$

где  $n$  – порядок системы,  $n = n_{sat} + n_{lin}$ ,  $n_{sat}$  – количество ограниченных фазовых переменных вектора состояния,  $n_{lin}$  – количество неограниченных фазовых переменных вектора состояния.

Структуру нейронной сети можно формировать в зависимости от имеющихся обратных связей: на основе введенных задержек в обратную связь по регулируемой величине или на основе обратных связей, составляющих полный вектор состояния объекта управления.

Минимальная структура нейроимитатора, основанная на полном векторе состояния, для объекта с ограничителями:

- 1-й слой (входной) –  $n+1$  (вектор состояния объекта и сигнал управления);
- 2-й слой (скрытый) –  $4 n_{sat} + 2 n_{lin}$ ;
- 3-й слой (выходной) –  $n$  (вектор состояния на новом шаге).

Минимальная структура нейроимитатора, основанная на введении задержек в обратную связь, для объекта с ограничителями:

- 1-й слой (входной) –  $2n$ , состоящий из сигнала управления объектом (1) и его задержанных значений ( $n-1$ ), обратной связи от объекта управления (1) и ее задержанных значений ( $n-1$ );
- 2-й слой (скрытый) –  $4n_{sat} + 2n_{lin}$ ;
- 3-й слой (выходной) – 1 (выход системы).

Предложенные структуры являются минимальными, так как увеличение структуры для обучения может показать улучшение результатов. Например, в случае, когда параметры объекта сильно различаются (на несколько порядков). Обычно в таком случае входные и выходные значения, на которых обучается нейронная сеть, нормализуются, чтобы при вычислении градиента влияние одного выхода сети не было более значимым, однако такой подход не всегда гарантирует успех. Также в структуре имеются коэффициенты точно равные 0, 1 и -1, которые при обучении градиентными методами не всегда возможно получить. Увеличение структуры сети позволит распределить коэффициенты по всей структуре, что позволит получить лучшие результаты.

В случае, когда получить полный вектор состояния невозможно, но имеется несколько обратных связей, то можно объединить оба предложенных варианта структур. Для этого следует добавить задержанные сигналы обратных связей и сигнал управления к входному вектору в нейроимитатор, при этом может потребоваться увеличение структуры скрытых слоев.

В общем случае для нейросетевого регулятора со структурой ИНС прямого распространения, как и для линейных звеньев, требуется  $2n$  входов с задержанными сигналами, где  $n$  - порядок объекта управления.

Это правило можно распространить и на нейросетевой регулятор для ограничителей. Т. е. для внутреннего слоя обязательно требуется использовать функции активации класса ReLU, которые из-за своего кусочно-линейного

характера могут действовать как «переключатели» и активируют различные части регулятора, сформированные в процессе обучения.

Универсальное описание структуры нейрорегулятора для произвольного объекта найти невозможно. Здесь предлагаются следующие соображения: порядок объекта определяет минимальный размер входа и количество задержек для обратных связей, а для скрытых слоев следует пользоваться правилом поэтапного увеличения нейронов и количества слоев до достижения результата, соответствующего заданным требованиям.

**Пример синтеза нейроимитатора на основе нейронной сети прямого распространения для звена с ограничением типа насыщение.**

Для примера была выбрана ИНС со структурой из одного скрытого слоя размером 10 нейронов и с функцией активации ReLU.

На вход нейронной сети подавался вектор из 6 значений:

3 – текущее и 2 задержанных значения управления,

3 – задержанные значения выхода объекта.

Данная структура сложнее, чем в предложенном выше алгоритме. Это сделано для демонстрации различных возможных решений. В процессе обучения выход ИНС сравнивался с выходом объекта, который должен быть получен на текущем шаге.

Результаты работы обученного нейросетевого имитатора при гармоническом и ступенчатом входных сигналах, с достижением и без достижения ограничений представлены на рисунках 2.12 - 2.15.

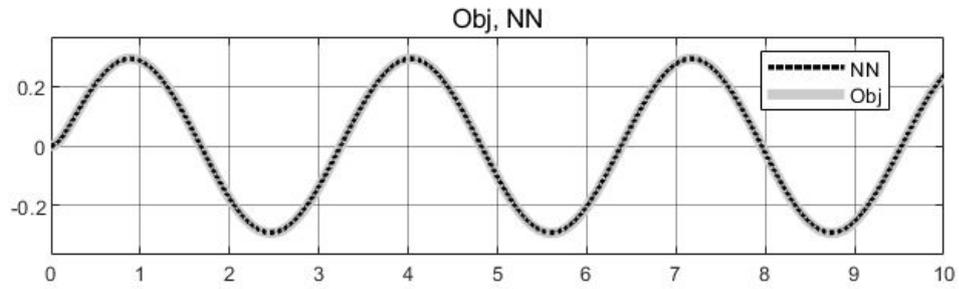


Рисунок 2.12 - Реакция объекта и нейроимитатора на гармонический сигнал, при котором не достигаются ограничения

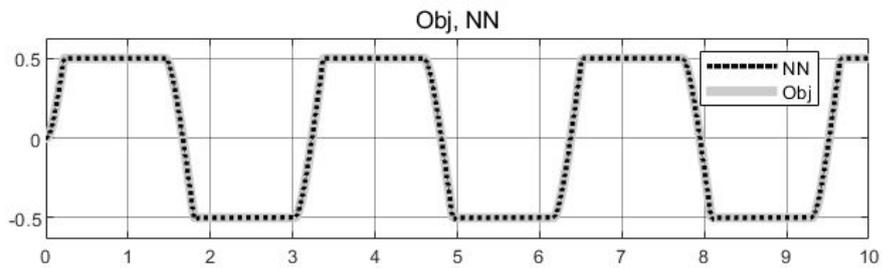


Рисунок 2.13 - Реакция объекта и нейроимитатора на гармонический сигнал, при котором достигаются ограничения

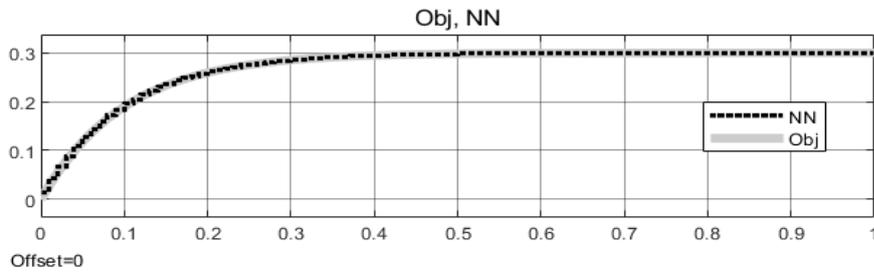


Рисунок 2.14 - Реакция объекта и нейроимитатора на ступенчатый сигнал амплитудой 0.3 (ограничения не достигаются)

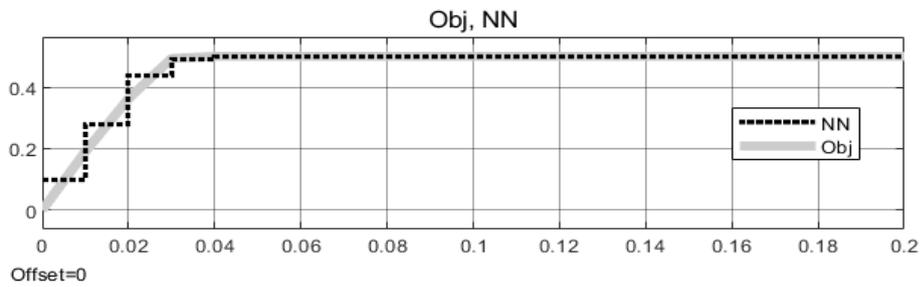


Рисунок 2.15 - Реакция объекта и нейроимитатора на ступенчатый сигнал амплитудой 2 (ограничения достигаются)

Как видно из рисунков нейронная сеть с предложенной структурой достаточно хорошо имитирует простой объект с ограничителем.

**Пример синтеза нейроимитатора на основе нейронной сети прямого распространения для звена с ограничителем типа жесткий механический упор.**

Используется аналогичная предыдущему примеру структура ИНС. Результаты работы обученного нейроимитатора при гармоническом и ступенчатом входных сигналах с достижением и без достижения ограничений представлены на рисунках 2.16 - 2.19.

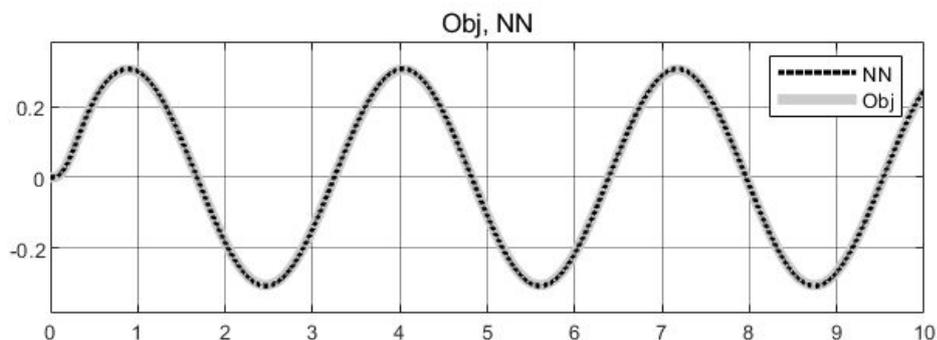


Рисунок 2.16 - Реакция объекта и нейроимитатора на гармонический сигнал, при котором ограничения не достигаются

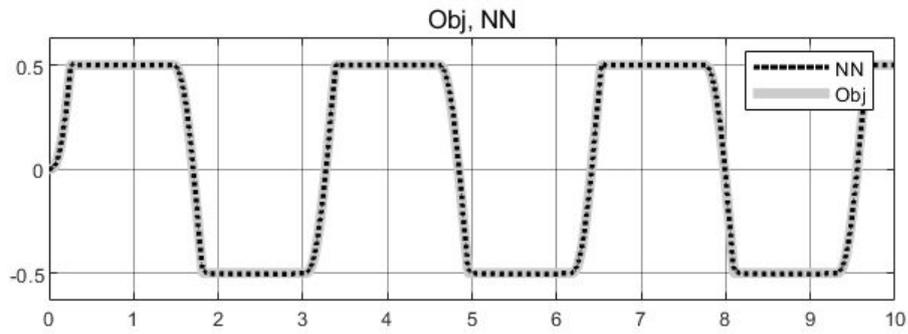


Рисунок 2.17 - Реакция объекта и нейроимитатора на гармонический сигнал, при котором ограничения достигаются

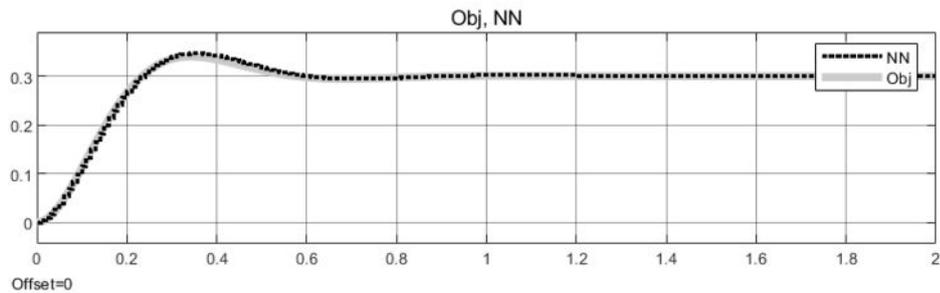


Рисунок 2.18 - Реакция объекта и нейроимитатора на ступенчатый сигнал амплитудой 0.3 (ограничения не достигаются)

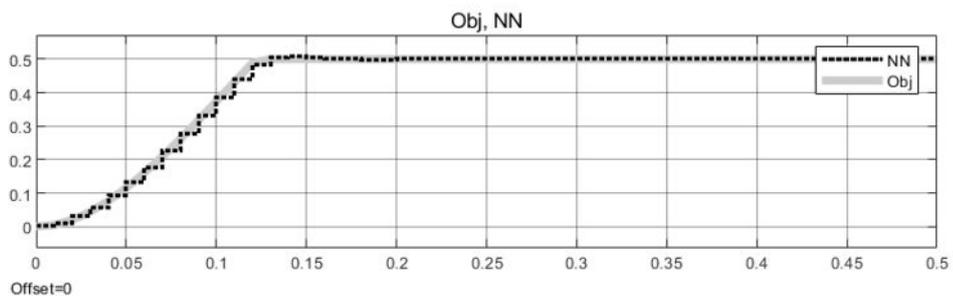


Рисунок 2.19 - Реакция объекта и нейроимитатора на ступенчатый сигнал амплитудой 2 (ограничения достигаются)

Как видно из рисунков 2.16 - 2.19, нейронная сеть способна имитировать данную нелинейность, однако точность несколько хуже, чем для звена с ограничением типа насыщение. Это связано с возросшей сложностью объекта.

### **2.3. Структура нейроимитатора и нейрорегулятора на основе рекуррентных нейронных сетей для объектов с ограничителями.**

Основным недостатком сетей прямого распространения является необходимость введения, не всегда определенного количества, задержек в обратные связи, или использование полного вектора состояния. Для гладких нелинейностей проблем при синтезе нейрорегулятора и нейроимитатора не возникает. А в случае объектов с несколькими последовательно включенными звеньями с ограничителями может возникнуть ситуация, когда как минимум два ограничения достигнуты, и с помощью задержек невозможно восстановить весь вектор состояния. При этом необходимо выполнять интегрирование фазовых переменных, но структура нейронной сети прямого распространения не позволяет это реализовать.

В этом случае более универсальным решением является использование рекуррентных нейронных сетей. В этой структуре имеется внутреннее состояние, представленное обратными связями внутри слоя, коэффициенты которых настраиваются совместно с весами прямых связей в процессе обучения. В результате это позволяет рекуррентной сети использовать необходимые значения из предыдущих расчетов для формирования выхода на текущем шаге.

Таким образом разработчику не требуется как-то специально формировать данные для передачи в нейронную сеть. Достаточно просто использовать в качестве входа все текущие обратные связи, без создания каких-либо искусственных задержек. Однако, для такого типа нейронных сетей формирование общих требований к структуре является затруднительным, так как обратные связи формируются автоматически.

## 2.4. Алгоритм формирования обучающей выборки для нейронных сетей разных структур с учетом наличия ограничителей в объекте управления.

Исходя из накопленного опыта в виде алгоритма сформулированы общие рекомендации по формированию данных для обучения нейроимитатора и нейрорегулятора для объектов с наличием ограничителей.

1. Входные сигналы должны отражать особенности работы объекта как в динамике, так и в статике. В общем случае предлагается использовать типовые непрерывные и разрывные сигналы (синус и меандр разных амплитуд и частот). Сигнал меандра используется для получения информации о работе объекта на скачкообразное изменение сигнала и на постоянный входной сигнал. Синус используется для получения информации о плавной динамике. Сигналы следует разделить на 2 категории:

- при которых ограничения не достигаются;
- при которых достигаются ограничения, причем если ограничений в объекте несколько, то каждое ограничение по отдельности, так и вместе.

Желательно, чтобы в обучающей выборке количество данных в категориях было примерно одинаковым.

2. На все выбранные сигналы необходимо записать реакцию объекта управления.

3. Собранные данные следует подготовить для обучения нейронных сетей.

Для сетей прямого распространения записанные сигналы разбиваются на пары так, чтобы они отображали зависимость входных данных и результата на одном такте квантования. Т. е. для нейроимитатора ( $nn_{\text{э}}$ ):  $x_i = nn_{\text{э}}(u_i, x_{i-1}, x_{i-2}, \dots)$ , а для нейрорегулятора ( $nn_{\text{р}}$ ):  $u_i = nn_{\text{р}}(r_i, x_{i-1}, x_{i-2}, \dots)$ , где  $i$  - номер дискретного шага,  $u$  - сигнал управления объектом,  $x$  - состояние объекта,  $r$  - сигнал управления системой (рисунок 1.10). Таким образом, собранные пары никак не зависят друг от друга и от того, из какого сигнала они были получены, что позволяет ускорить

обучение за счет параллельных вычислений. Схемы для пояснения представлены на рисунках 1.10 и 1.11.

Для обучения рекуррентных нейронных сетей должна сохраняться временная последовательность записанных сигналов. Входной и целевой сигналы нельзя разделять, так как при обучении используется метод обратного распространения ошибки сквозь время, благодаря которому настраиваются веса обратных связей в рекуррентных слоях. Также, обычно, в процессе обучения начальное состояние (внутренний вектор обратных связей) берется равным нулю.

## **2.5. Метод синтеза нейрорегулятора на основе схемы обучения инверсного нейруправления для объектов с ограничителями.**

Метод синтеза нейрорегулятора на основе схемы обучения инверсного нейруправления для объектов с ограничителями заключается в следующем.

1. Выбирается структура нейрорегулятора. Т. е. определяются типы слоев, количество нейронов в каждом слое, функции активации (обязательно ReLU и, возможно, для других слоев гладкие нелинейности (sigmoid, tanh), если качество обучения с ними ожидается лучше). Для нейронной сети прямого распространения ограничения на выбор минимальной структуры представлены в п. 2.1. Для рекуррентной нейронной сети общих рекомендаций по структуре нет, однако было замечено, что слои LSTM и GRU показывают лучший результат на длительных сигналах.

2. Формируется набор тренировочных данных, отображающий инверсную динамику, согласно приведенному алгоритму формирования обучающей выборки для объектов с ограничителями в п. 2.4. Т.е. выборка с тренировочными данными должна состоять из двух массивов векторов. В первом находятся данные для входа в нейронную сеть (обратные связи от объекта управления), во втором содержится вектор управления  $u$ , который привел объект в данное состояние из первого массива. Для рекуррентных нейронных сетей в таких массивах должна сохраняться последовательность данных.

Полученную тренировочную выборку следует разбить на 2 части: тренировочную и проверочную. Обычно выборку делят 90% к 10% соответственно [43]. Можно добавить еще тестовую выборку (5-15%). Однако, если имеется возможность, то лучше осуществить проверку на объекте или математической модели. Для рекуррентных нейронных сетей в тренировочную и проверочную выборки входит полная и упорядоченная информация о каждом записанном сигнале. В каждую выборку должны попасть сигналы из линейной зоны, и с достижением ограничений и других нелинейностей. Соотношение частей остается приблизительно таким же.

3. Производится обучение нейрорегулятора методом обратного распространения ошибки или методом обратного распространения ошибки во времени в зависимости от типа нейронной сети. Более универсальным способом обучения является разделение данных на мини-пакеты (mini-batch) с использованием вариаций адаптивного стохастического градиентного спуска, например Adam [49]. Для рекуррентных сетей в мини-пакеты входит полная информация о записанных сигналах. В качестве функции потерь обычно используется среднеквадратичная функция потерь (MSE-loss) (2.16).

$$\text{loss}(\mathbf{a}, \tilde{\mathbf{a}}) = \frac{1}{k} \sum_{i=1}^k (a_i - \tilde{a}_i)^2, \quad (2.16)$$

где  $\mathbf{a}$  и  $\tilde{\mathbf{a}}$  - векторы с входными (полученными в результате расчета ИНС на данных из обучающей выборки) и целевыми значениями соответственно,  $k$  – количество значений в обучающей выборке.

В процессе обучения в каждой эпохе следует контролировать переобучение (процесс описан в главе 1). В случае если было достигнуто переобучение, то обучение останавливается и необходимо менять структуру сети, или увеличивать количество данных для обучения.

Для рекуррентной нейронной сети при подаче каждого нового сигнала необходимо сбрасывать внутренне состояние слоев в начальное состояние (обычно нулевой вектор).

Необходимо повторять эпохи обучения до тех пор, пока ошибка обучения не окажется достаточно маленькой (определяется разработчиком нейрорегулятора).

4. Завершается метод проверкой качества полученного нейрорегулятора при работе в замкнутом контуре системы управления.

Блок схема метода представлена на рисунке 2.20.

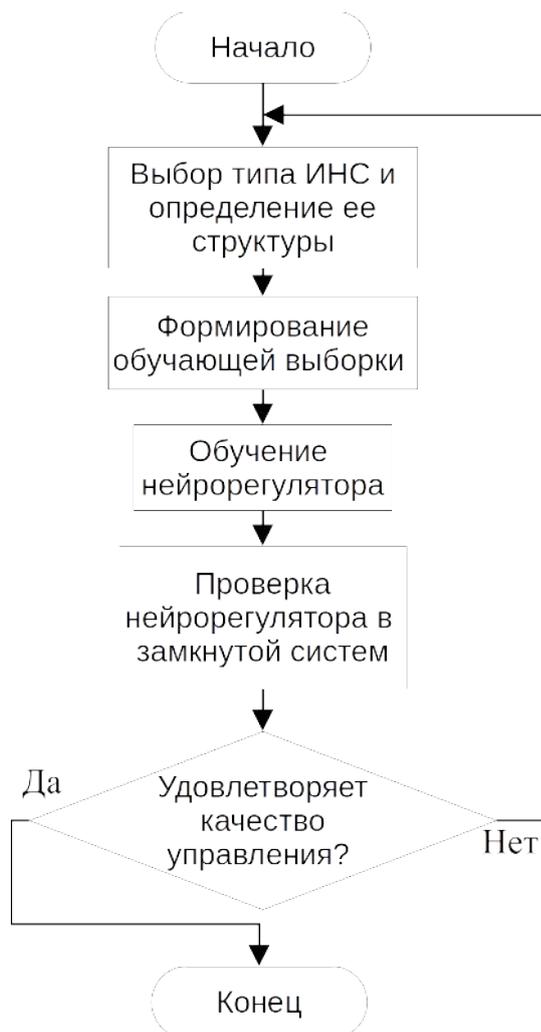


Рисунок 2.20 - Метод синтеза нейрорегулятора на основе схемы обучения инверсного нейрорууправления для объектов с ограничителями

### Пример синтеза нейрорегулятора методом инверсного нейруправления для звена с ограничителем типа насыщение.

Структура нейронной сети нейрорегулятора для управления звеном с ограничителем типа насыщение представлена в таблице 2.1.

Таблица 2.1. Структура нейронной сети для нейрорегулятора со звеном с ограничителем типа насыщение

№	Слой	Количество нейронов
1	Входной	6
2	Линейный с функцией активации ReLU	100
3	Линейный с функцией активации ReLU	40
4	Выходной	1

Для обучения подается текущее и 3 задержанных значения выхода объекта, 2 задержанных значения входа объекта.

Результат работы полученного нейросетевого регулятора в сравнении с ПИД-регулятором, синтезированным при помощи средств Matlab для системы с объектом, являющимся звеном с ограничением типа насыщение показан на рисунках 2.21, 2.22.

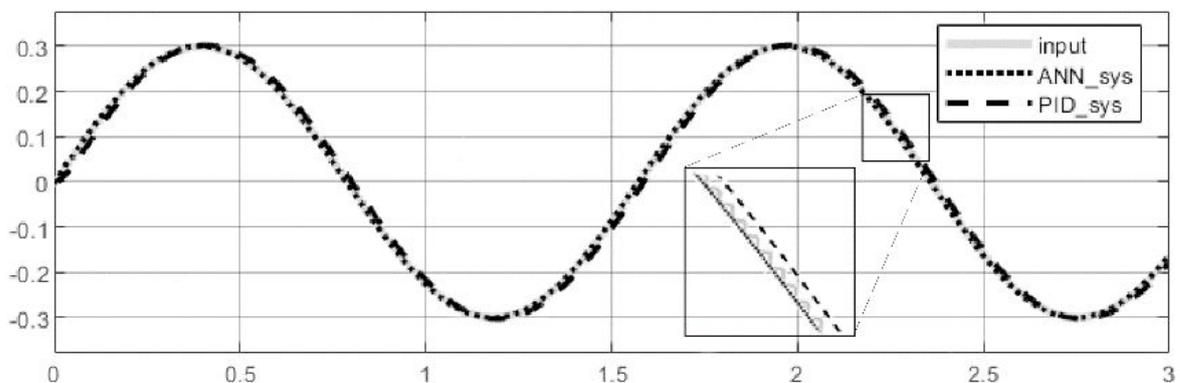


Рисунок 2.21 - Реакция систем с нейросетевым регулятором и ПИД регулятором для звена с ограничителем типа насыщение на

гармонический сигнал амплитудой 0.3, при котором ограничения не достигаются

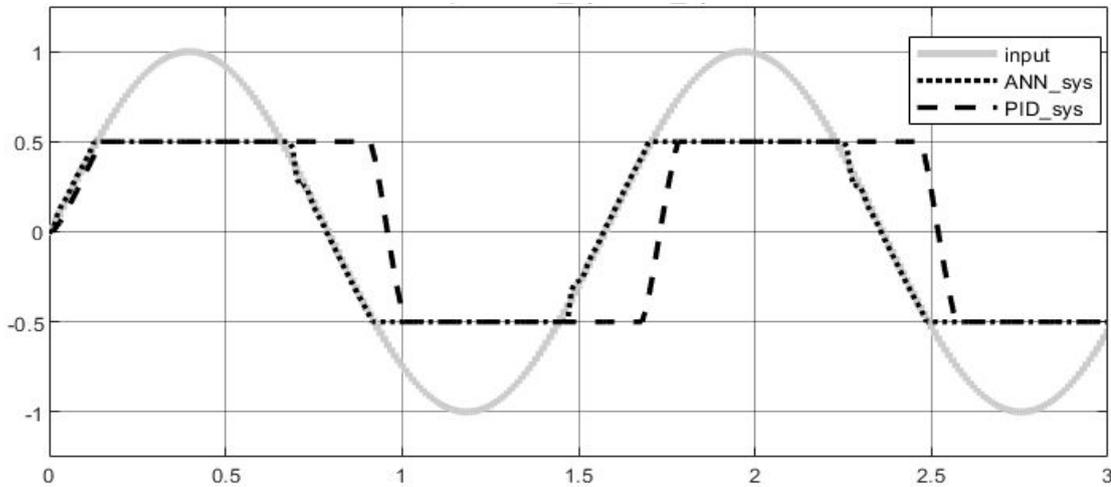


Рисунок 2.22 - Реакция системы с нейросетевым регулятором и ПИД регулятором для звена с ограничителем типа насыщение на гармонический сигнал амплитудой 1, при которых ограничения достигаются

Как можно увидеть, полученный нейронный регулятор может качественно управлять звеном с ограничением типа насыщение.

**Пример синтеза нейрорегулятора методом инверсного нейроуправления для звена с ограничителем типа жесткий механический упор.**

Описание структуры нейронной сети, используемой в качестве нейрорегулятора для управления звеном с ограничителем типа механический упор представлено в таблице 2.2.

Таблица 2.2. Описание структуры нейрорегулятора для звена с ограничителем типа жесткий механический упор.

№	Слой	Количество нейронов
1	Входной	5
2	Линейный с функцией активации ReLU	500
3	Dropout	-
4	Линейный с функцией активации ReLU	50
5	Выходной	1

Слой Dropout [35] используется для получения стабильного обучения и улучшения результатов работы сети.

Результат работы полученного нейросетевого регулятора в сравнении с ПИД-регулятором, синтезированным при помощи средств Matlab для системы с объектом, являющимся звеном с ограничением типа насыщение показан на рисунках 2.23, 2.24.

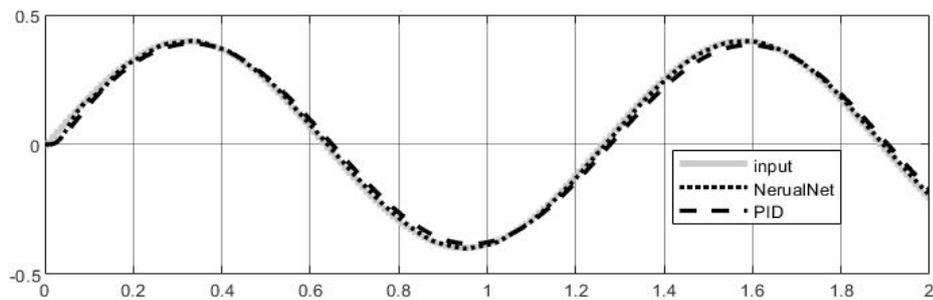


Рисунок 2.23 - Реакция системы с нейросетевым регулятором и ПИД регулятором для звена с ограничителем типа жесткий механический упор на гармонический сигнал амплитудой 0.4, при котором ограничения не достигаются

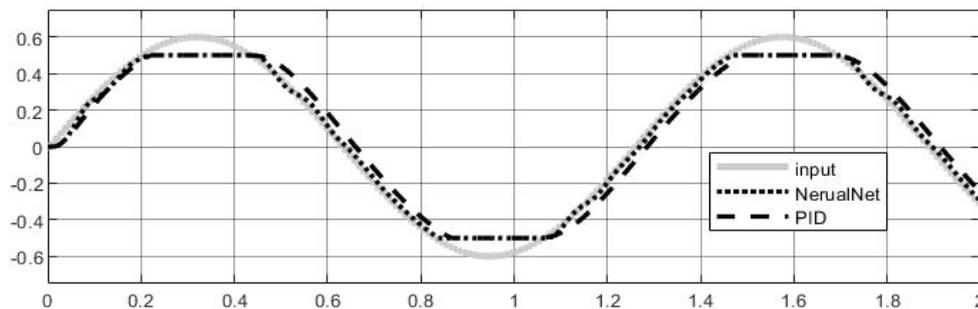


Рисунок 2.24 - Реакция системы с нейросетевым регулятором и ПИД регулятором для звена с ограничителем типа жесткий механический упор на гармонический сигнал амплитудой 0.6, при которых ограничения достигаются

Как можно увидеть, полученный нейронный регулятор может адекватно управлять звеном с ограничением типа механический упор.

**Пример синтеза нейрорегулятора на основе рекуррентной нейронной сети методом инверсного нейроуправления для звена с ограничителем типа жесткий механический упор.**

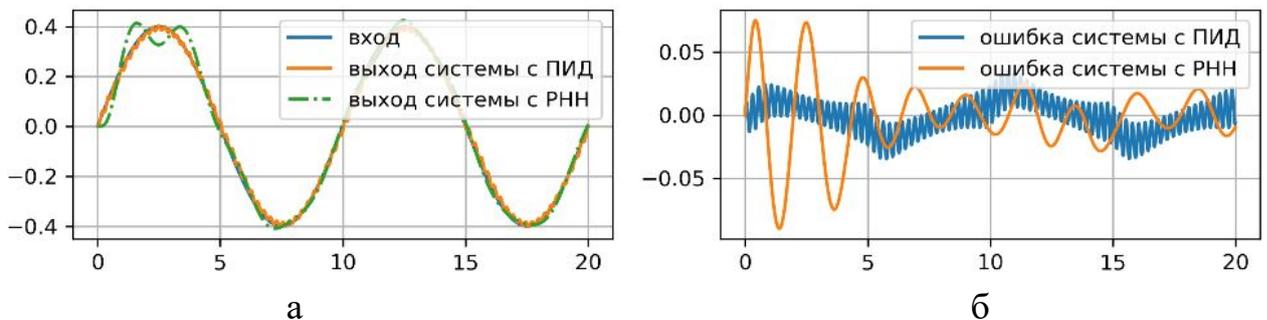
Использовалась нейронная сеть со структурой, представленной в таблице 2.3. Выбор структуры проводился экспериментально с последовательным наращиванием количества слоев и нейронов.

Таблица 2.3. Структура нейронной сети для нейрорегулятора

№ п/п	Слой	Кол-во нейронов
1	Входной	2
2	GRU	200
3	Dropout 25%	-
4	Линейный	200
5	ReLU	-
6	Dropout 25%	-
7	Линейный	200
8	ReLU	-
9	Выходной	1

Обучение производилось на гармонических ( $\sin$ ) сигналах.

На рисунках 2.25 и 2.26 представлены результаты моделирования в Simulink. Сравнивается работа нейрорегулятора с ПИД регулятором, синтезированным средствами Matlab.

Рисунок 2.25 - Реакция объекта и РНН на гармонический сигнал ( $\sin$ )

амплитудой 0,4 и частотой 0,1 Гц: а – реакция систем, б – ошибка

слежения

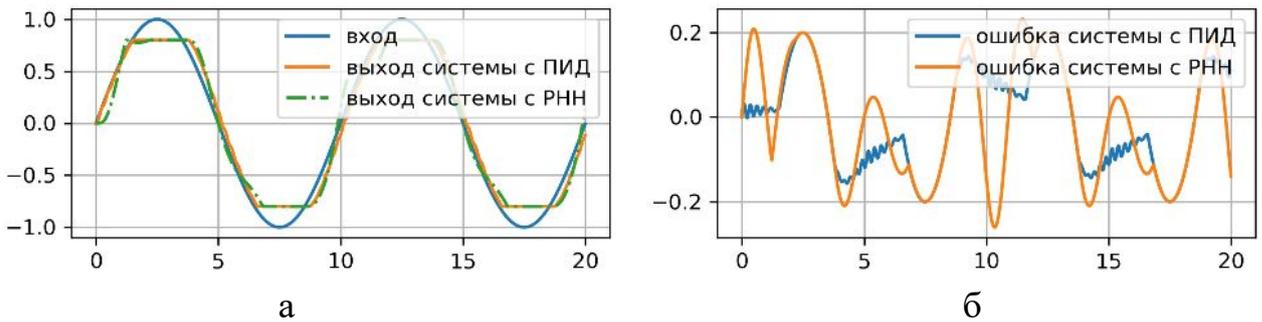


Рисунок 2.26 - Реакция объекта и РНН на гармонический сигнал ( $\sin$ ) амплитудой 1 и частотой 0,1 Гц: а – реакция систем, б – ошибка слежения

Как видно, нейронная сеть смогла обучиться управлять объектом почти также хорошо, как и ПИД-регулятор.

## 2.6. Метод синтеза нейросетевого регулятора на основе схемы обучения обратного пропуска ошибки через нейроимитатор.

Метод синтеза нейрорегулятора на основе схемы обучения обратного пропуска ошибки через нейроимитатор для объектов с ограничителями состоит из следующих этапов:

1. Выбирается структура нейросетевого имитатора согласно разработанным рекомендациям (п. 2.3).

2. Формируется выборка, отражающая динамику работы объекта, согласно алгоритму описанному в п. 2.4, которая будет использоваться для обучения нейроимитатора.

3. Выполняется обучение нейроимитатора.

Обучение выполняется так же, как и для нейрорегулятора из п. 2.5. Т. е. тренировочная выборка делится на mini-batch, а в качестве оптимизатора предлагается использовать алгоритм Adam [49].

4. Определяется структура нейрорегулятора. Она никак не зависит от нейроимитатора. Разработчик может выбрать сеть прямого распространения или

рекуррентную нейронную сеть, а ее размеры определяются рекомендациями из п. 2.1 и п. 2.3.

5. Выполняется обучение нейрорегулятора методом обратного пропуска ошибки через нейроимитатор. Так как обучение нейрорегулятора происходит в реальном времени, то собирать тренировочные данные не требуются. Вместо этого ошибка работы регулятора вычисляется во время моделирования или работы системы. Вся система рассматривается как дискретная. На каждом такте квантования выполняется вычисление значения управления на выходе нейрорегулятора на основе задающего сигнала для системы и обратных связей от объекта. Результатом вычислений является значение управляющего сигнала, на основе которого нейроимитатором определяется новое состояние объекта управления. Затем полученное состояние сравнивается с задающим сигналом через среднеквадратичную функцию потерь (2.16). Полученное рассогласование методом обратного распространения ошибки передают через нейроимитатор и нейрорегулятор. Однако, обучается только нейрорегулятор. Такой подход в идеальном случае должен обеспечить возможность достижения системой заданного состояния за один такт, но это не всегда возможно. Кроме того, часто требуется оценивать другие параметры динамики системы, например, минимизацию энергопотребления в процессе регулирования. Для добавления возможности контроля таких параметров, данный подход можно дополнить. Ошибка регулирования вычисляется не за один конкретный такт квантования, а за небольшой промежуток времени, по которому можно оценить время регулирования, затраты энергии и другие параметры. Все эти параметры оцениваются в виде численного показателя с помощью более сложной дифференцируемой функции, вид которой зависит от измеряемых параметров динамики системы. Данная оценка распространяется методом обратного распространения ошибки в нейрорегуляторе.

6. Производится оценка качества полученной системы в различных режимах работы. Так как обучение происходит в реальном времени, то в случае

ухудшения качества управления (например, при изменении параметров объекта управления) возможно выполнить дополнительное обучение как нейроимитатора, так и нейрорегулятора. Таким образом система получает свойство адаптивности, которое, однако, может быть использовано только для медленно изменяющихся параметров. Это обусловлено тем, что процесс обучения требует больших вычислительных ресурсов и времени по сравнению с расчетом нейронной сети и временем такта квантования.

Блок схема метода представлена на рисунке 2.27.

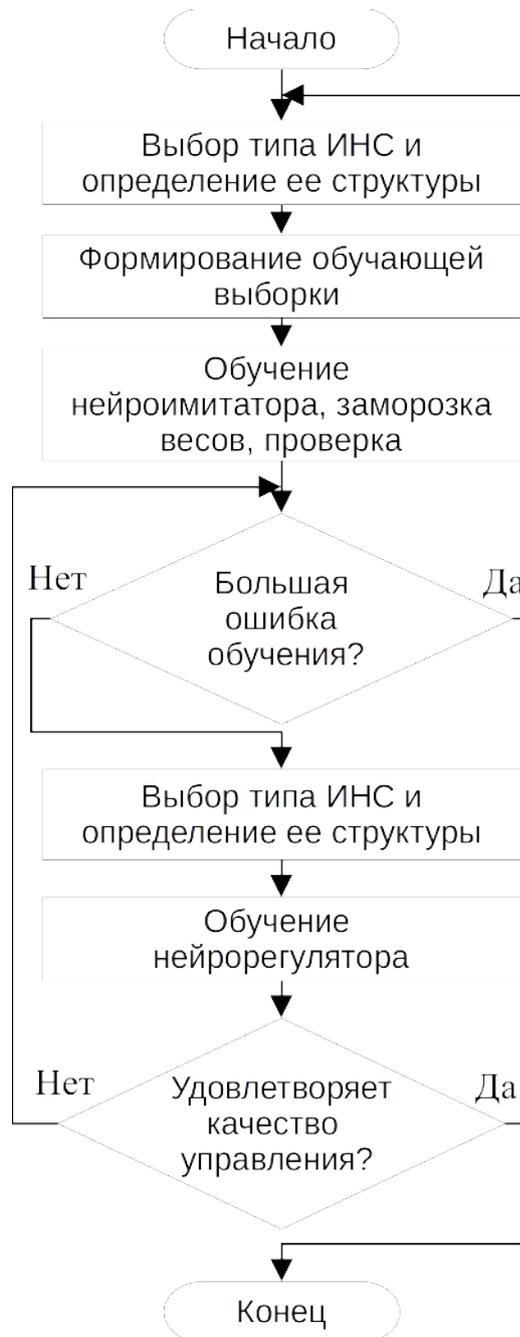


Рисунок 2.27 - Блок схема метода синтеза нейросетевого регулятора на основе схемы обучения обратного пропуск ошибки через нейроимитатор

**Пример синтеза нейрорегулятора методом обратного пропуск ошибки через нейроимитатор для объекта с ограничителем типа насыщение.**

В качестве примера рассмотрим звено с ограничением типа насыщение (рисунок 2.6). Его параметры:  $T = 0.3$ ,  $k = 1$ ,  $D = \pm 0.5$ .

В качестве нейроимитатора выбрана сеть прямого распространения, а структура определена согласно разработанному алгоритму из п. 2.1 для данного звена, т. е. 2 нейрона во входном слое, 4 нейрона в скрытом слое с наличием весов смещения и с функцией активации ReLU, и один нейрон в выходном слое.

Для формирования обучающего набора данных были сгенерированы 40 гармонических сигналов со случайно выбранными параметрами в заданных ограничениях так, чтобы 20 сигналов не затрагивали ограничений, а другие 20 затрагивали.

В результате обучения была получена ошибка слежения  $6 \cdot 10^{-6}$  (рисунок 2.28).

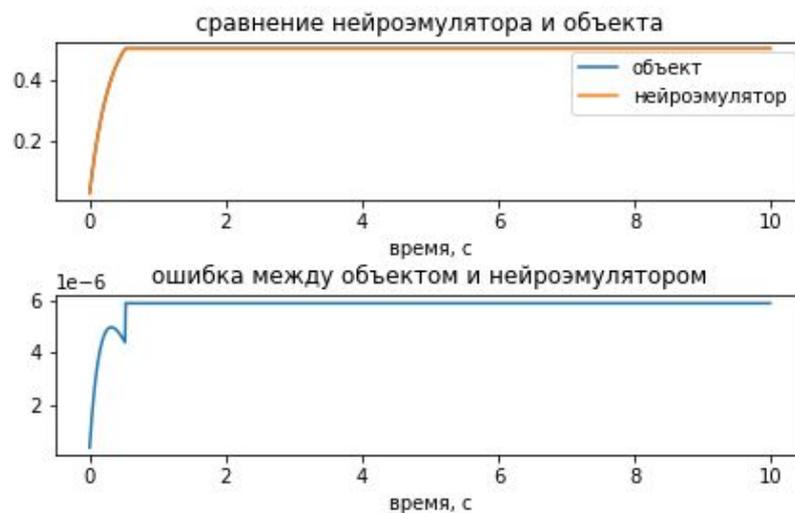


Рисунок 2.28 - Сравнение выхода объекта и обученного нейроимитатора

Для нейрорегулятора была выбрана архитектура из 4 слоев, в первом 2 нейрона, в скрытых слоях (втором и третьем) по 4 нейрона в каждом с функцией активации ReLU и выходным слоем с одним нейроном.

Сигналы для обучения нейрорегулятора генерировались таким же образом, как и для нейроэмулятора. Обучение заняло 500 эпох и было остановлено со значением ошибки  $\sim 1.4 \cdot 10^{-6}$ . Результаты моделирования представлены на рисунках 2.29 и 2.30.

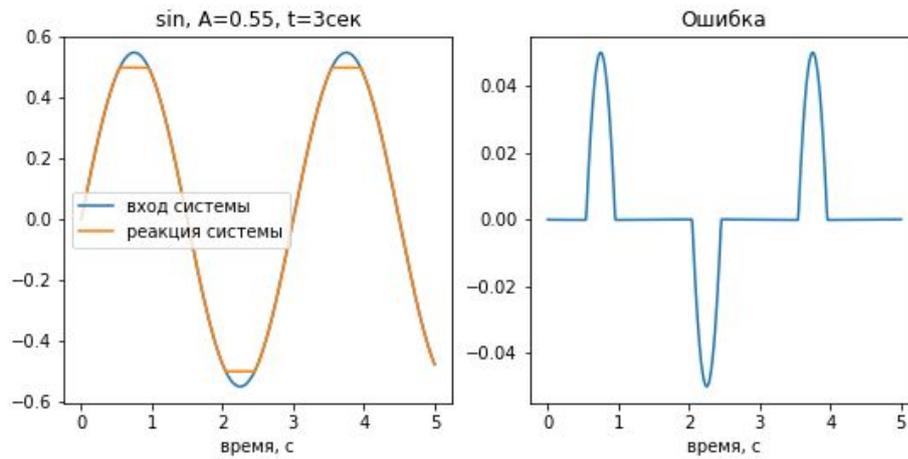


Рисунок 2.29 - Результат моделирования системы с нейрорегулятором  
( $\sin$ ,  $A=0.55$ ,  $T=3c$ )

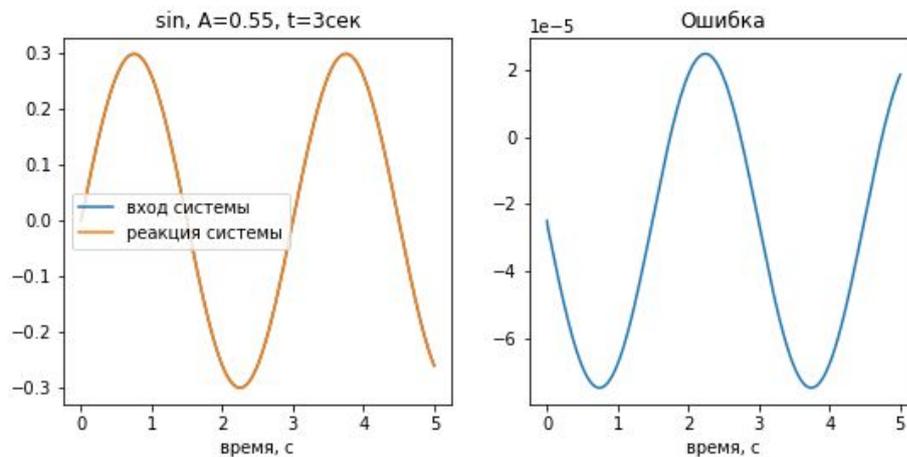


Рисунок 2.30 - Результат моделирование системы с нейрорегулятором  
( $\sin$ ,  $A=0.3$ ,  $T=3c$ )

Ошибка слежения в линейной зоне составляет до  $7 \cdot 10^{-5}$ . Таким образом, синтезирован качественный регулятор при сравнительно небольших размерах нейронных сетей в сравнении с методом инверсного нейроуправления.

## 2.7. Выводы

Рассмотрены способы выбора структуры ИНС как для имитации линейных объектов управления, так и для управления ими. Так как нейронные сети реализуются на цифровом оборудовании, то их структура может быть основана на

разностном уравнении линейного объекта управления. Т.е. в качестве структуры рассматривается нейронная сеть прямого распространения, состоящая из 2-х слоев: выходным с одним нейроном и входным с количеством нейронов, зависящим от предполагаемого порядка объекта и необходимым количеством задержек обратных связей.

Рассмотрены объекты с ограничителями и показано, что нейронные сети с нелинейной функцией активации ReLU могут быть обучены для имитации и управления такими объектами. Разработаны алгоритмы формирования минимальных структур ИНС для таких объектов как на основе нейронной сети прямого распространения, так и рекуррентной нейронной сети.

Предложен алгоритм формирования тренировочных данных на основе объектов с ограничителями для обучения нейронных сетей.

Разработаны методы синтеза нейросетевого регулятора для управления объектом с ограничителями, позволяющие в качестве основы выбрать как нейронную сеть прямого распространения, так и рекуррентную нейронную сеть. Первый метод основан на схеме обучения «инверсное нейроуправление», второй на схеме обучения «обратный пропуск ошибки через нейроимитатор».

### 3. Синтез устойчивой замкнутой следящей системы с нейросетевым регулятором и объектом управления с ограничителем

Как было показано выше, одной из главных фундаментальных проблем нейросетевых регуляторов является отсутствие теоретически обоснованной гарантии устойчивости синтезированной замкнутой нейросетевой системы управления.

В работе предложен метод синтеза нейросетевого регулятора, позволяющий гарантировать асимптотическую устойчивость замкнутой системы в заданной локальной области фазового пространства, в том числе для объектов с ограничителями. Для этого вместе с нейросетевым регулятором синтезируется функция Ляпунова, которая формируется в виде нейронной сети прямого распространения с функцией активации leakyReLU [59]. Для подтверждения выполнения условий устойчивости используется решение задачи целочисленного программирования [24, 31, 56, 82], которое также позволяет найти максимальное нарушение заданных условий (контр-примеры) и обеспечить их исключение.

#### 3.1. Преобразование ИНС к задаче целочисленного программирования.

В ИНС, которые используются в предлагаемом методе используются нейронные сети прямого распространения с функцией активации leakyReLU (или ReLU - ее частный случай), которая описывается уравнением (3.1), а ее график представлен на рисунке 3.1.

$$\text{leakyReLU}(y) = \begin{cases} y, & y \geq 0; \\ cy, & y < 0; \end{cases} \quad (3.1)$$

где  $c$  – значение коэффициента отрицательной части leakyReLU.

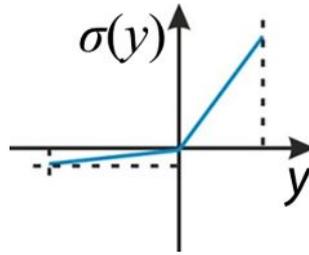


Рисунок 3.1 - График функции leakyReLU

Функция активации leakyReLU выбрана по причине того, что ее можно представить в виде системы ограничений:

$$\begin{aligned}
 \sigma &\geq y; \\
 \sigma &\geq cy; \\
 \sigma &\leq cy + (1-c)y_{up}\beta; \\
 \sigma &\leq y + (1-c)y_{lo}(\beta-1); \\
 \beta &= 0 \text{ или } 1; \quad \sigma \in \mathbb{R};
 \end{aligned} \tag{3.2}$$

где  $\sigma$  – выход leakyReLU,  $\beta$  – бинарная переменная ( $\beta = 0$  – для отрицательных входов,  $\beta = 1$  – для положительных входов),  $c$  – значение коэффициента наклона для отрицательной части leakyReLU (для ReLU значение равно 0),  $y$  – значение входа leakyReLU,  $y_{up}$  и  $y_{lo}$  – ограничения на входную переменную. Для расчета  $y_{up}$  и  $y_{lo}$  для каждого нейрона используется интервальная арифметика [88].

В случае, если значение на входе элемента leakyReLU ожидается одного знака, т. е. верхнее и нижнее ограничения имеют одинаковый знак, то систему неравенств (3.2) можно заменить на заранее известную линейную зависимость, описываемую в виде равенства (3.1). Таким приемом можно сократить количество неравенств в системе для ускорения расчета оптимизатором.

Используя систему неравенств (3.2) для каждого нейрона можно сформировать общую систему неравенств, которая используется для задачи целочисленного программирования (mixed integer linear programming или MIP).

**Пример преобразования ИНС с функцией активации leakyReLU к задаче целочисленного программирования.**

Данный пример демонстрирует способ приведения нейронной сети с функциями активации leakyReLU к задаче целочисленного программирования.

В качестве нейронной сети рассматривается сеть прямого распространения с одним скрытым слоем, состоящим из 2-х нейронов с функцией активации ReLU (рисунок 3.2). Задачей является поиск минимального возможного значения на выходе данной нейронной сети и соответствующего ему входного вектора при наличии ограничения на входные значения  $x \in [-2, 2]$ .

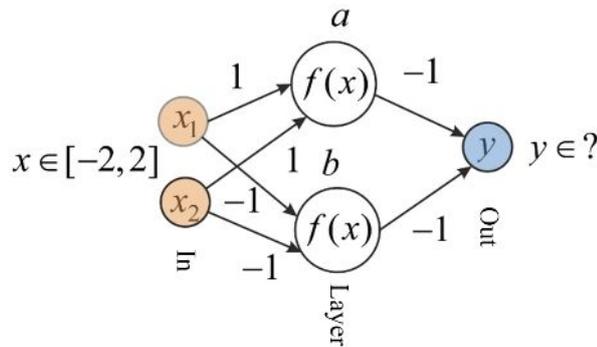


Рисунок 3.2 - Схема ИНС для преобразования в МIP задачу

Данную нейронную сеть можно представить в виде следующей системы неравенств (3.3):

$$\begin{aligned}
 & -2 \leq x_1 \leq 2; \\
 & -2 \leq x_2 \leq 2; \\
 & a_i = x_1 + x_2; \\
 & b_i = -x_1 - x_2; \\
 & a \geq a_i; & b \geq b_i; \\
 & a \geq 0; & b \geq 0; \\
 & a \leq a_i^{up} \beta_a; & b \leq b_i^{up} \beta_b; \\
 & a \leq a_i + a_i^{lo} (\beta_a - 1); & b \leq b_i + b_i^{lo} (\beta_b - 1); \\
 & y = -a - b; \\
 & \beta_a, \beta_b \in [0 \text{ или } 1];
 \end{aligned} \tag{3.3}$$

где  $x$  - значение на входе нейронов 1 и 2 соответственно,  $a_i$  и  $b_i$  - значения до функции активации нейронов  $a$  и  $b$  в скрытом слое,  $a$  и  $b$  - значения

соответствующих нейронов в скрытом слое,  $y$  - выход системы,  $\beta$  — бинарная переменная, т. е. принимающая значения либо 0, либо 1,  $a_i^{up}$ ,  $a_i^{lo}$ ,  $b_i^{up}$ ,  $b_i^{lo}$  найдены с помощью интервальной арифметики:

$$\begin{aligned} a_i^{up} &= 4, \\ a_i^{lo} &= -4, \\ b_i^{up} &= 4, \\ b_i^{lo} &= -4. \end{aligned} \tag{3.4}$$

Имея систему неравенств (3.3) можно решать ее как задачу целочисленного программирования. Так как нужно найти минимальное значение выхода сети, то целевой функцией является  $\min(y)$ . С помощью различных специализированных библиотек для решения задач целочисленного программирования CBC, SCIP и GUROBI [84] был выполнен расчет этой задачи с одинаковым результатом  $y = -4$  при  $x_1 = -2$ ,  $x_2 = -2$ . Если рассмотреть все решения, которые возможны (рисунок 3.3), то окажется, что минимальное значение  $y$  будет при  $x_1 = 2$ ,  $x_2 = 2$  или при  $x_1 = -2$ ,  $x_2 = -2$ . Все решатели находят одну оптимальную точку. Если имеется несколько точек, в которых целевая функция оптимальна, то в качестве результата решатель определит первую найденную.

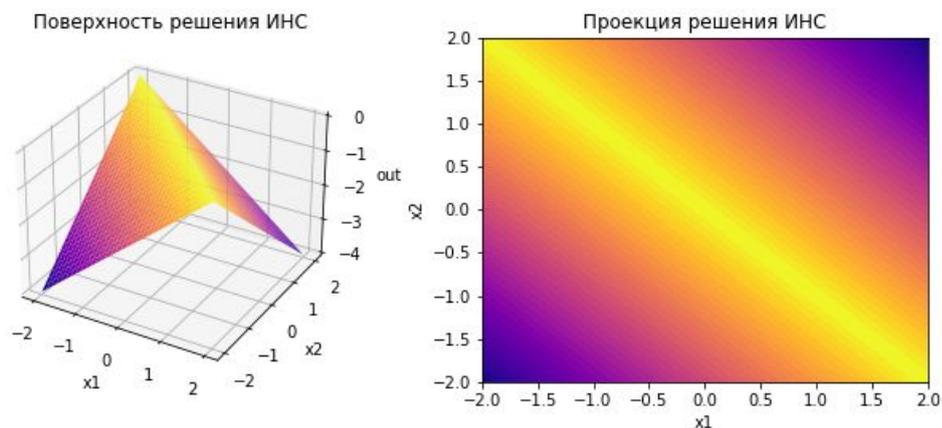


Рисунок 3.3 - Поверхность решения НС

Таким образом, приводя нейронную сеть к системе неравенств, можно решать задачи целочисленного линейного программирования для поиска входных значений, удовлетворяющим целевой функции.

В основе решателей используется метод ветвлений и границ [8, 52], который использует симплекс-метод [8]. Для первого решения в системе снимаются все требования целочисленности и симплекс-методом находится оптимальное решение. Затем оно проверяется на выполнение требований целочисленности. Если найденное решение удовлетворяет им, то найдено оптимальное решение для исходной системы. Если нет, то создаются новые подзадачи (ветвления), в которых добавляется условие (границы), по которому найденное решение уже не будет входить в область возможных решений. Каждая новая задача решается симплекс-методом и дальше происходит ветвление, пока не будет найдено оптимальное решение.

### 3.2. Структурный синтез устойчивой нейросетевой системы управления.

В данном методе рассматривается замкнутая дискретная система управления. Структура представлена на рисунке 3.4.

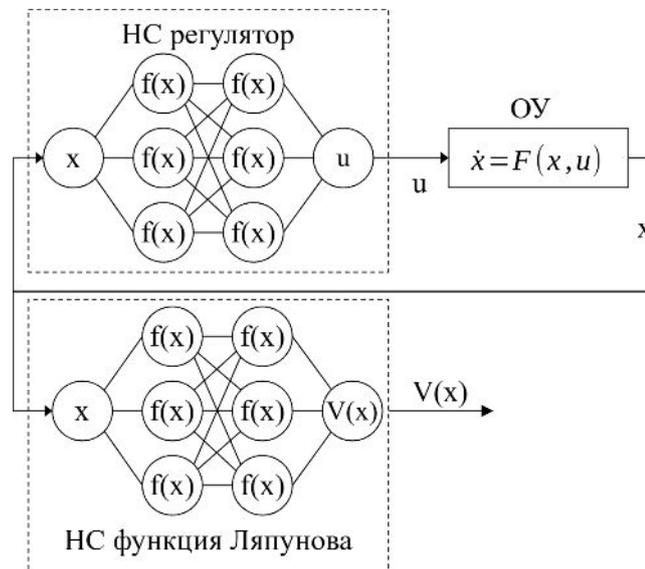


Рисунок 3.4 - Структура замкнутой нейросетевой системы управления для метода синтеза устойчивой замкнутой системы управления с нейрорегулятором

Нейрорегулятор основан на нейронной сети прямого распространения:

$$u_i(x_i) = \varphi_r(x_i) - \varphi_r(x^*) + u^*, \quad (3.5)$$

где  $\phi_r$  – нейронная сеть прямого распространения с функцией активации leakyReLU (или ReLU, т.к. ее можно рассматривать как частный случай leakyReLU),  $x_i$  – вектор состояния объекта управления в дискретный такт  $i$ ,  $x^*$  – вектор состояния в положении равновесия,  $u^*$  – сигнал управления в точке устойчивости,  $u_{min}$  и  $u_{max}$  – нижний и верхний пределы сигнала управления объектом соответственно. Уравнение (3.5) гарантирует, что в точке равновесия сигнал управления будет равен  $u^*$ .

При этом значение сигнала управления  $u_t$  ограничено:

$$u_i = \begin{cases} u_i, & \text{если } u_{min} \leq u_i \leq u_{max}; \\ u_{max}, & \text{если } u_i > u_{max}; \\ u_{min}, & \text{если } u_i < u_{min}; \end{cases} \quad (3.6)$$

Где  $u_{min}$ ,  $u_{max}$  – минимальное и максимальное ограничения на сигнал управления соответственно. Уравнениям (3.5) и (3.6) соответствует рисунок 3.5.

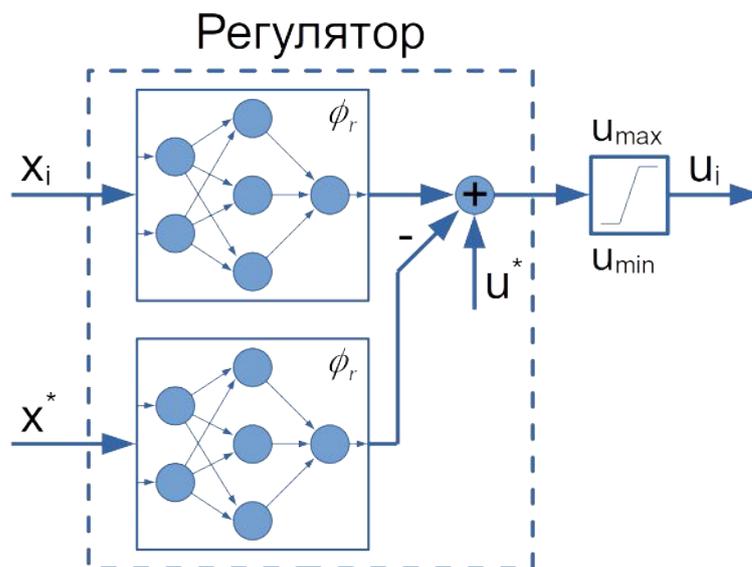


Рисунок 3.5 - Схема нейрорегулятора по методу синтеза нейросетевой системы управления с теоретической гарантией устойчивости. На схеме одна и та же нейронная сеть используется дважды для расчета сигнала управления объектом

Для гарантии устойчивости системы управления находится функция Ляпунова. Как известно, по второму методу Ляпунова система является

асимптотически устойчивой в локальной области, если имеется такая функция, называемая функцией Ляпунова  $V(x)$ , которая в этой области является положительной, а ее производная строго отрицательной, кроме положения равновесия, в которой она равна нулю:

$$V(x) > 0 \quad \forall x \in S, \quad x \neq x^*, \quad (3.7)$$

$$\dot{V}(x) < 0 \quad \forall x \in S, \quad x \neq x^*, \quad (3.8)$$

$$V(x^*) = 0, \quad (3.9)$$

где  $x$  — вектор состояния объекта управления,  $x^*$  — вектор состояния в точке устойчивости,  $S$  — локальная область фазового пространства, в которой система должна быть устойчива.

В случае экспоненциальной сходимости, уравнение (3.8) выглядит следующим образом [31, 58]:

$$\dot{V}(x) \leq -\varepsilon_2 V(x) \quad \forall x \in S, \quad x \neq x^*, \quad (3.10)$$

где  $\varepsilon_2$  — положительный скаляр  $\varepsilon_2 > 0$ .

Так как нейросетевые системы управления в данной работе рассматриваются как дискретные, то условия 3.7 и 3.8 преобразуются в 3.11 и 3.12 соответственно:

$$V(x_i) > 0 \quad \forall x_i \in S, \quad x_i \neq x^*, \quad (3.11)$$

$$V(x_{i+1}) - V(x_i) \leq -\varepsilon_2 V(x) \quad \forall x_i \in S, \quad x_i \neq x^*, \quad (3.12)$$

где  $x_i$  — вектор состояния объекта управления в дискретный шаг  $i$ .

В качестве кандидата в функцию Ляпунова рассматривается функция следующего вида:

$$V(x_i) = \phi_V(x_i) - \phi_V(x^*) + \|R(x_i - x^*)\|_1, \quad (3.13)$$

где  $\phi_V$  — нейронная сеть, являющаяся частью функции Ляпунова,  $R$  — матрица с полным рангом столбца,  $\|R(x_i - x^*)\|_1$  — это 1-я норма вектора. Уравнение (3.13) гарантирует, что  $V(x^*) = 0$ . Добавление  $\|R(x_i - x^*)\|_1$  позволяет гарантировать выполнение условия (3.7) в малой области вокруг  $x^*$  (рисунок 3.6).

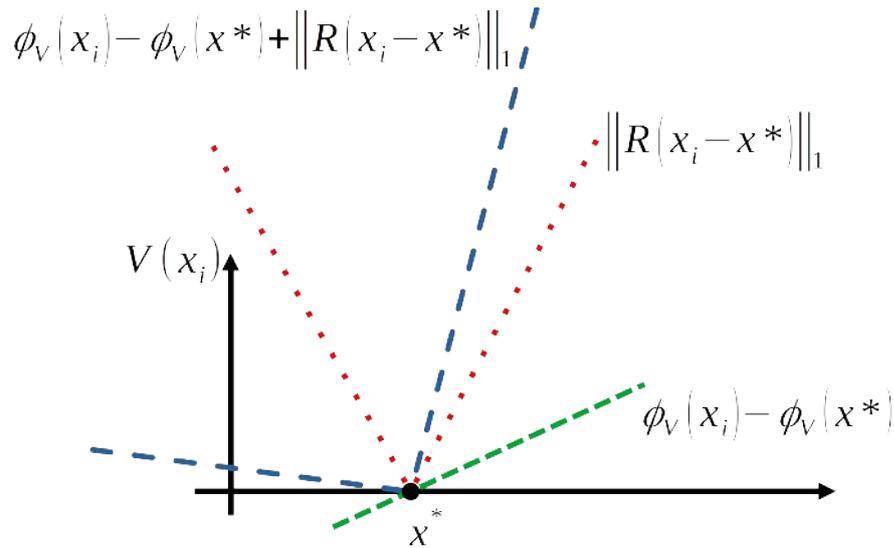


Рисунок 3.6 - Пример влияния  $\|R(x_i - x^*)\|_1$  на функцию Ляпунова в окрестности точки равновесия  $x^*$

Как показано на рисунке 3.6  $\phi_V(x_i) - \phi_V(x^*)$  представляет собой кусочно-линейную функцию, и, с большой вероятностью, точка  $(x^*, 0)$  находится на одной из линейных частей, а не на границе отрезка. Следовательно,  $\phi_V(x_i) - \phi_V(x^*)$  является линейной функцией, которая станет отрицательной вдали от  $x^*$ , что нарушает условие положительности (3.11). Добавление  $\|R(x_i - x^*)\|_1$  к функции Ляпунова при достаточно большом  $R$  позволяет гарантировать, что в локальной окрестности  $x^*$  функция Ляпунова будет положительна.

Уравнению (3.13) соответствует рисунок 3.7.

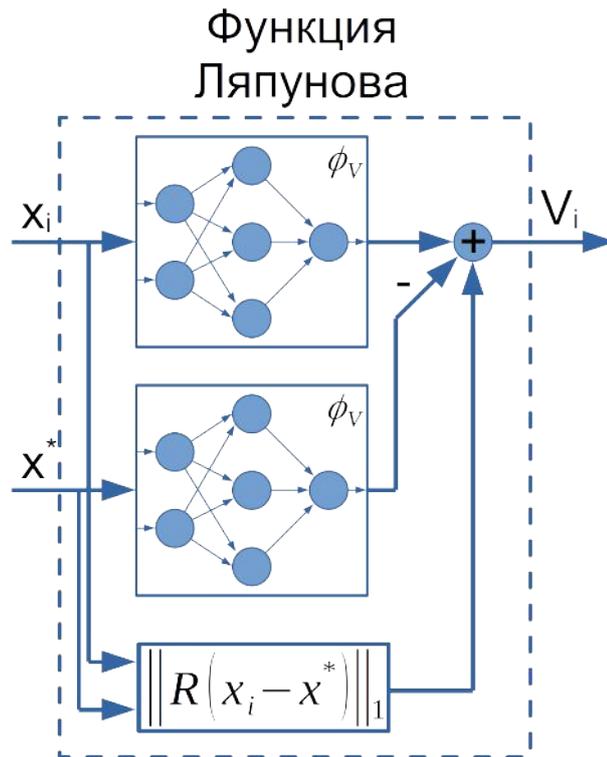


Рисунок 3.7 - Схема нейросетевой функции Ляпунова

Так как для обучения нейрорегулятора используется метод обратного распространения ошибки, то необходимо создать нейроимитатор для аппроксимации динамики объекта управления, как в методе «обратного пропуска ошибки через нейроимитатор». Для этого нейроимитатор в общем виде представлен в следующем виде:

$$x_{i+1} = \varphi_{dyn}(x_i, u_i) - \varphi_{dyn}(x^*, u^*) + x^*, \quad (3.14)$$

$$u_{min} \leq u_i \leq u_{max}, \quad (3.15)$$

где  $\varphi_{dyn}$  - нейронная сеть прямого распространения с функцией активации leakyReLU или ReLU. Уравнение (3.14) гарантирует, если объект находится в точке равновесия  $x_i = x^*$ ,  $u_i = u^*$ , то следующее состояние  $x_{i+1}$  также будет точкой равновесия  $x_{i+1} = x^*$ . Данному уравнению соответствует рисунок 3.8.

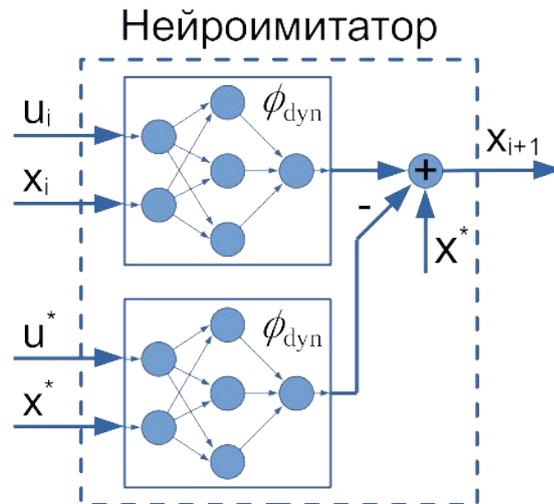


Рисунок 3.8 - Схема нейроимитатора

Нейроимитатор обучается заранее на записанных данных с объекта управления (рисунок 3.9).

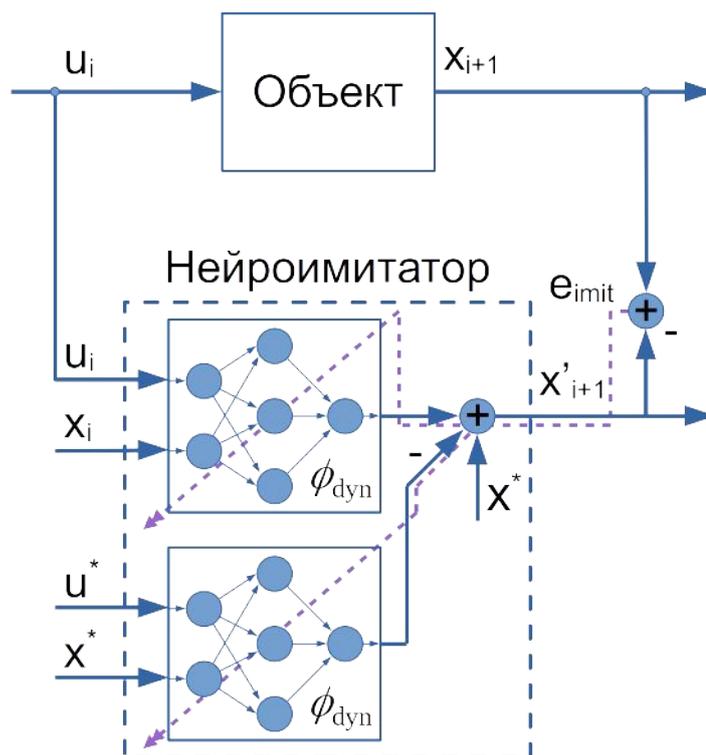


Рисунок 3.9 - Схема обучения нейроимитатора

В случае, если объект управления имеет ограничители, то следует использовать разработанные алгоритмы формирования структуры (п. 2.3) и обучающей выборки (п. 2.4) для нейронной сети прямого распространения.

### 3.3. Поиск контр-примеров, нарушающих условия устойчивости

Для обучения нейрорегулятора и функции Ляпунова необходимо найти точки фазового пространства в рассматриваемой области  $S$ , в которых максимально нарушаются условия (3.11) и (3.12), т. е. контр-примеры. Если найденные значения удовлетворяют условиям, то это означает, что найдена функция Ляпунова, гарантирующая асимптотическую устойчивость замкнутой системы управления в заданной локальной области фазового пространства.

С помощью целочисленного программирования можно гарантированно найти оптимальное значение, соответствующее целевой функции. Однако, для поиска контр-примеров таким способом необходимо преобразовать условия (3.11) и (3.12), так как допустимы только нестрогие неравенства.

Условие (3.11) преобразуется к следующему виду:

$$V(x_i) \geq \varepsilon_1 \|R(x_i - x^*)\|_1 \quad \forall x_i \in S, \quad (3.16)$$

где  $0 < \varepsilon_1 < 1$ ,  $R$  — матрица с полным рангом. В неравенстве (3.16) правая часть может быть равна 0 только при  $x_i = x^*$ , а в остальных точках она больше нуля. Следовательно, преобразование от строгого неравенства (3.11) к нестрогому (3.16) является верным, так функция Ляпунова в точке равновесия должна быть равна 0.

Для возможности обучения нейрорегулятора и нейросетевой функции Ляпунова следует иметь алгоритм нахождения координат точек фазового пространства, которые нарушают условия устойчивости (3.7), (3.8), т. е. контр-примеры. Если таких примеров не будет найдено, то условия Ляпунова не нарушены, а значит замкнутая система управления является устойчивой в заданной локальной области фазового пространства.

Для этого следует решить задачу оптимизации целевых функций:

$$\max_{x_i \in S} (\varepsilon_1 \|R(x_i - x^*)\|_1 - V(x_i)), \quad (3.17)$$

$$\max_{x_i \in S} (V(x_{i+1}) + (\varepsilon_2 - 1)V(x_i)). \quad (3.18)$$

Таким образом, необходимо обучить нейромимитатор с функциями активации leakyReLU, который будет повторять заданный объект управления. Создать не обученный нейрорегулятор (или взять заранее обученный каким-либо другим способом) и не обученную нейронную сеть в качестве кандидата в функцию Ляпунова. Затем представить все эти нейронные сети в виде системы неравенств и найти значения, соответствующие (3.17) и (3.18). Затем найденные значения используются как контр-примеры для обучения нейросетевой функции Ляпунова и нейрорегулятора.

### 3.4. Обучение нейронных сетей на контр-примерах

На основе найденных точек в фазовом пространстве, нарушающих условия устойчивости по Ляпунову, производится обучение нейросетевой функции Ляпунова и нейрорегулятора. Таким образом, получаются два критерия, которые необходимо минимизировать на найденных точках:

$$\eta_1 = \max(\varepsilon_1 \|R(x_i - x^*)\|_1 - V(x_i), 0), \quad (3.19)$$

$$\eta_2 = \max(V(x_{i+1}) + (\varepsilon_2 - 1)V(x_i), 0). \quad (3.20)$$

Общая функция потерь выглядит следующим образом:

$$loss = \|\eta_1\|_p + \|\eta_2\|_p, \quad (3.21)$$

где  $p$  – норма, может быть 1-й, 4-й нормой и т.п.

Таким образом, минимизируя (3.21) обучаются нейросетевая функция Ляпунова и нейрорегулятор. В результате при достижении нулевого значения функцией (3.21) обеспечивается устойчивость замкнутой системы в заданной области. Фактическая область устойчивости в фазовом пространстве может быть равна или больше, чем заданная локальная область  $S$ .

Общая схема обучения представлена на рисунке 3.10.

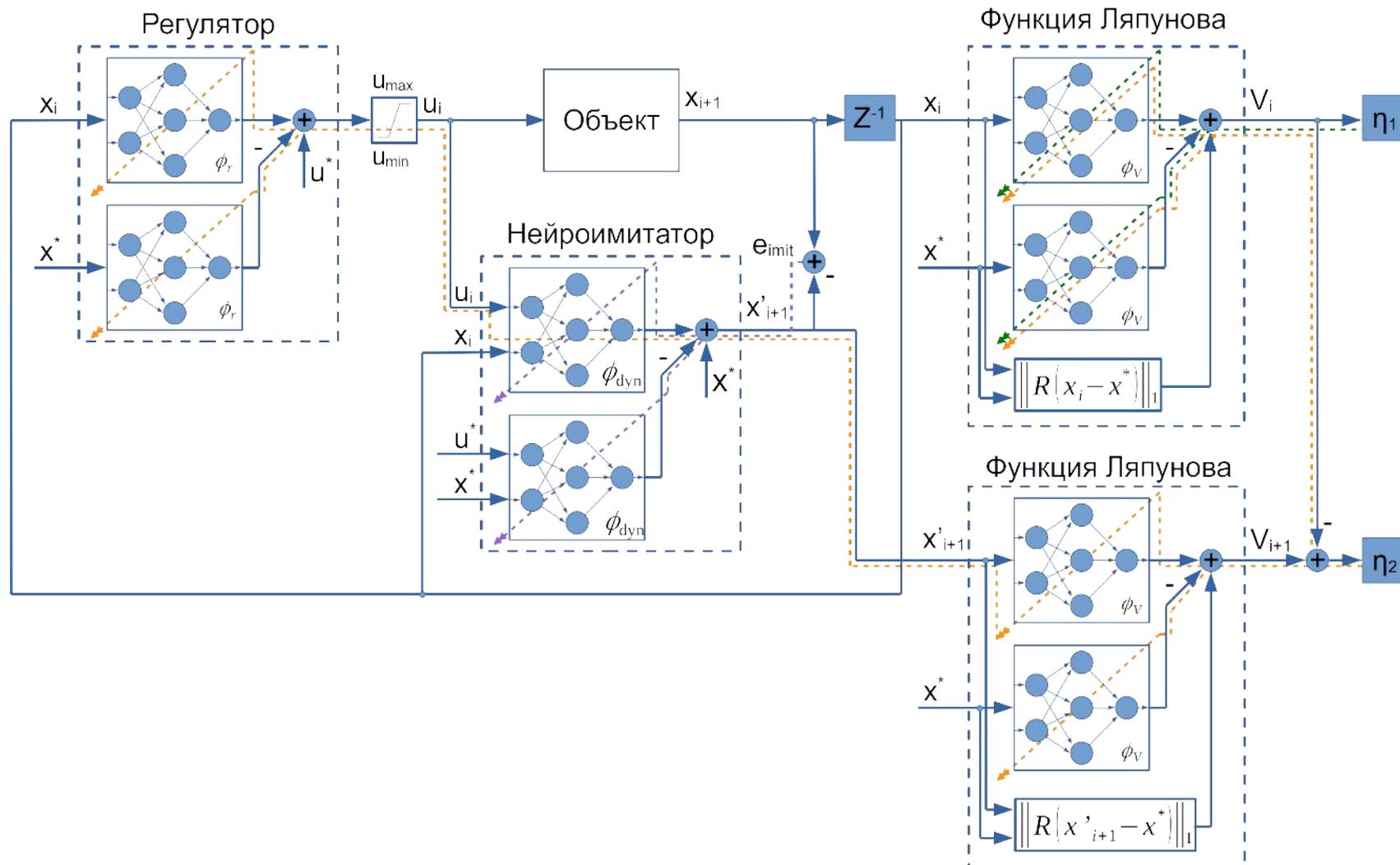


Рисунок 3.10 - Общая схема обучения нейронных сетей по методу синтеза устойчивой замкнутой системы управления в заданной локальной области фазового пространства

На рисунке 3.11 показана сжатая схема обучения по данному методу.

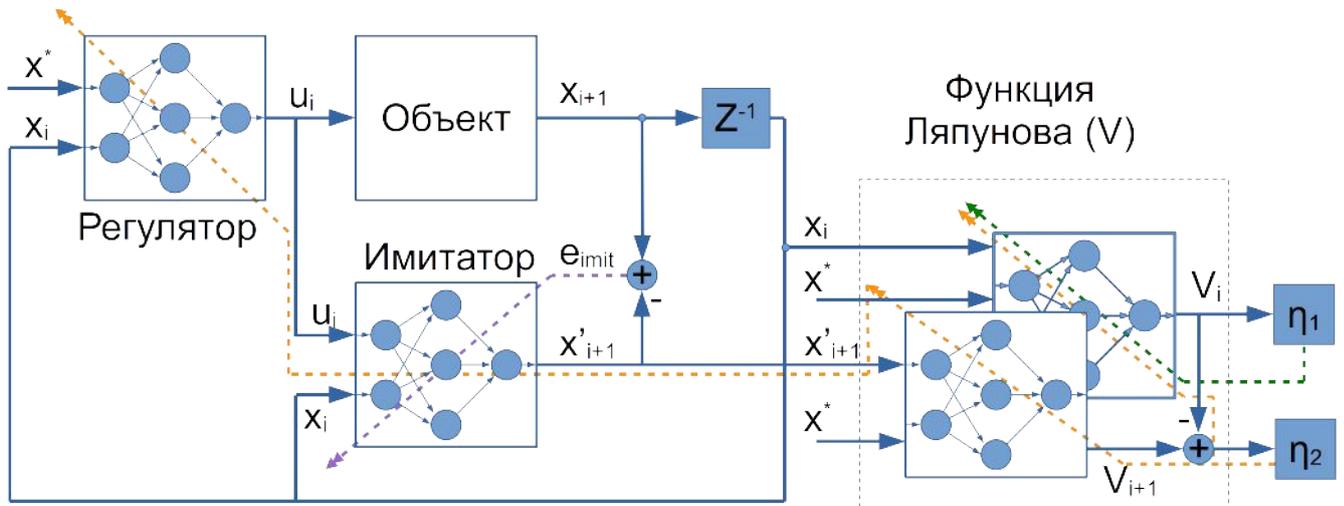


Рисунок 3.11 - Сокращенная схема обучения нейронных сетей по методу синтеза устойчивой замкнутой системы управления в заданной области фазового пространства

### Пример синтеза устойчивой системы управления обратным маятником.

Рассмотрим пример синтеза нейросетевого стабилизирующего регулятора для упрощенной модели перевернутого маятника (рисунок 3.12). Динамика системы представлена в следующем виде (3.22):

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{u - m \cdot g \cdot l \cdot \sin(x_1) - d \cdot x_2}{m \cdot l^2} \end{aligned} \quad (3.22)$$

$m = 1$  кг — масса маятника,  $l = 1$  м — длина маятника,  $g = 9.81$  м/с<sup>2</sup> — ускорение свободного падения,  $d = 0.1$  — коэффициент демпфирования,  $x_1$  — угловое положение маятника,  $x_2$  — угловая скорость маятника,  $u$  — управляющее воздействие.

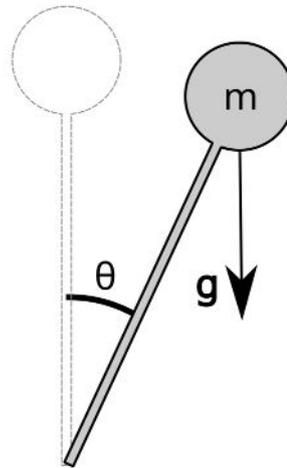


Рисунок 3.12 - Перевернутый маятник

Обратный маятник имеет точку равновесия в углах  $0^\circ$  (внизу) и  $180^\circ$  (вверху). Однако под силой тяжести маятник из любых углов, кроме ровно  $180^\circ$ , стремится в угол  $0^\circ$ .

Задачей является синтез регулятора, который удерживал бы маятник в вертикальном положении. Таким образом, должна получиться автономная система, где точкой устойчивости является положение  $180^\circ$  с нулевыми угловой скоростью и нулевым управлением.

На первом этапе требуется создать нейроимитатор объекта. В качестве нейронной сети выбрана сеть прямого распространения со структурой, представленной в таблице 3.1:

Таблица 3.1. Структура нейроимитатора обратного маятника

№ слоя	Количество нейронов	Функция активации	Смещение
1	3	-	-
2	5	LeakyReLU (neg_slope = 0.01)	+
3	5	LeakyReLU (neg_slope = 0.01)	+
4	1	-	+

В данном примере для обучения нейросетевого имитатора тренировочные данные сформированы по результатам моделирования объекта управления на один дискретный шаг из множества различных начальных условий. В качестве начальных данных брались точки фазового пространства, построенные по равномерной сетке внутри заданных ограничений (3.23). Для фазовых переменных была взята 101 точка, для сигнала управления 401 точка, т. е. 4090601 уникальных точек.

$$\begin{aligned} -0.5\pi &\leq x_1 \leq 0.5\pi; \\ -5 &\leq x_2 \leq 5; \\ -15 &\leq u \leq 15; \end{aligned} \quad (3.23)$$

Нейроимитатор описан согласно уравнению (3.14).

Нейрорегулятор представлен функцией (3.5). Для нейрорегулятора выбрана нейронная сеть прямого распространения, структура которой представлена в таблице 3.2.

Таблица 3.2. Структура нейрорегулятора для управления обратным маятником

№ слоя	Количество нейронов	Функция активации	Смещение
1	2	-	-
2	3	LeakyReLU (neg_slope = 0.1)	+
3	2	LeakyReLU (neg_slope = 0.1)	+
4	1	-	+

Нейросетевая функция Ляпунова представлена уравнением (3.13). Для этой функции также выбрана нейронная сеть прямого распространения, представленная в таблице 3.3.

Таблица 3.3. Структура нейронной сети, являющейся основой функции-кандидата в функцию Ляпунова.

№ слоя	Количество нейронов	Функция активации	Смещение
1	2	-	-
2	8	LeakyReLU (neg_slope = 0.1)	+
3	8	LeakyReLU (neg_slope = 0.1)	+
4	6	LeakyReLU (neg_slope = 0.1)	+
5	1	-	+

Для поиска контр-примеров все нейронные сети были представлены в виде систем неравенств и составлены две целочисленные задачи, каждая со своей целевой функцией (3.17) и (3.18) соответственно. В качестве решателя использовался алгоритм из библиотеки GUROBI.

Далее, на найденных контр-примерах производится обучение нейросетевого регулятора и нейросетевого имитатора, путем минимизации функции потерь (3.21) методом обратного распространения ошибки. При этом на контр-примерах, найденных для целевой функции (3.17), обучается только нейросетевая функция Ляпунова, а для целевой функции (3.18), обучается и нейросетевая функция Ляпунова и нейрорегулятор.

Процесс обучения повторяется, каждый раз получая новые задачи целочисленного программирования, и производится обучение на накапливаемых контр-примерах, пока ошибка функции потерь (3.21) не станет нулевой или достаточно малой.

На промежуточном этапе обучения функция Ляпунова имеет следующий вид (рисунок 3.13):

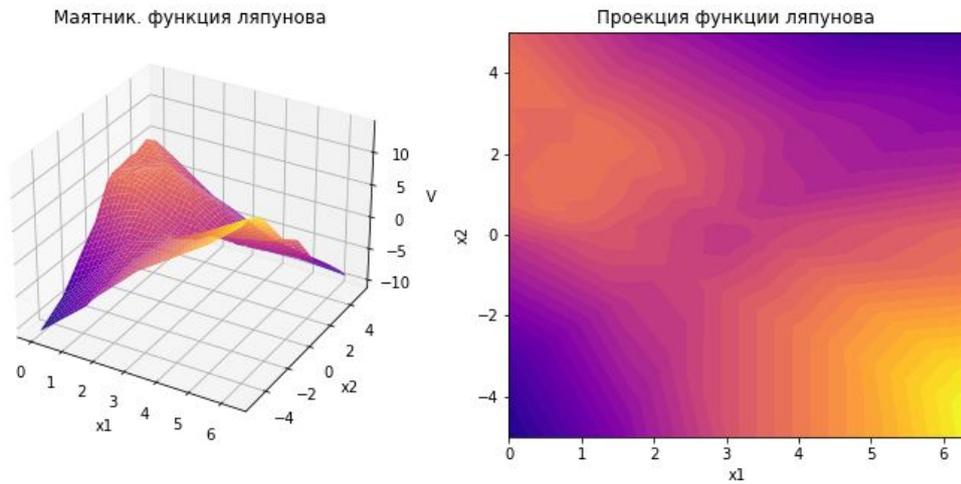


Рисунок 3.13 - Поверхность нейросетевой функции Ляпунова на промежуточном этапе обучения

После окончания обучения, когда ошибка стала нулевой, функция Ляпунова имеет следующий вид (рисунок 3.14):

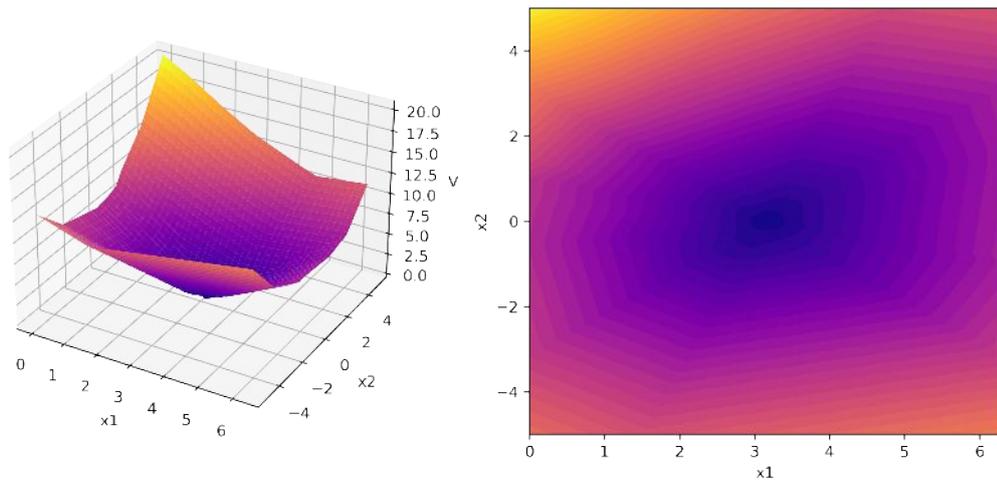


Рисунок 3.14 - Поверхность нейросетей функции Ляпунова

Таким образом, во всей прямоугольной области фазового пространства  $S$ , полученной из ограничений на фазовые переменные (3.23), функция Ляпунова гарантирует устойчивость замкнутой системы.

На рисунке 3.15 показан переходный процесс полученной системы.

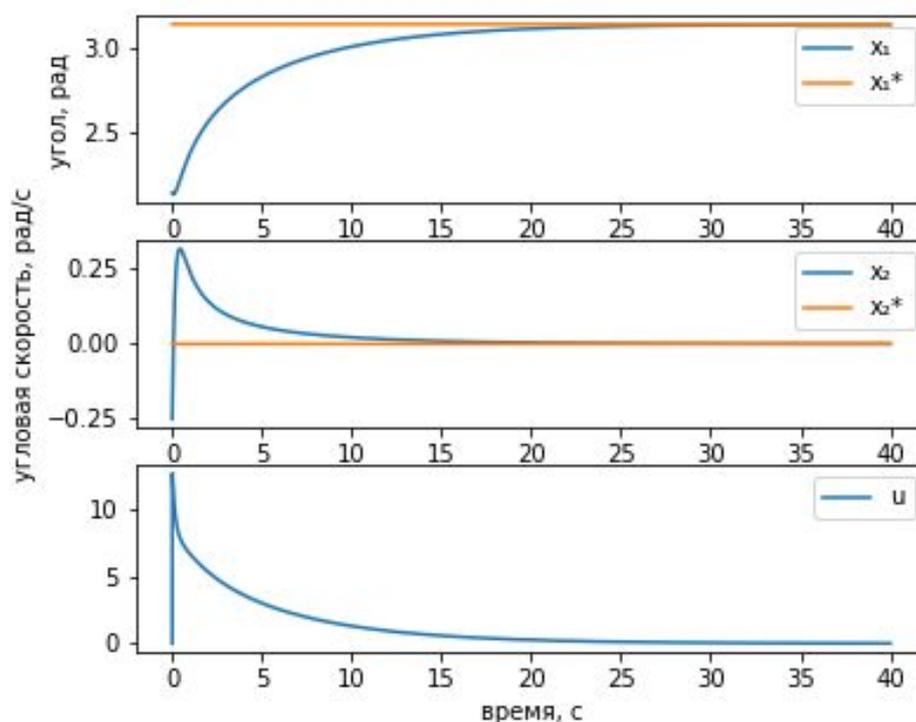


Рисунок 3.15 - График переходного процесса системы с перевернутым маятником, переходящий из начальной точки фазового пространства в точку равновесия без требований к времени регулирования

Таким образом, по данному методу синтезируется нейросетевой регулятор, гарантирующий устойчивость полученной замкнутой системы автоматического управления в заданной локальной области фазового пространства. Недостатком данного метода является то, что никак не контролируется качество переходного процесса. Из данного примера видно, что время стабилизации маятника занимает около 20 секунд, хотя при заданных ограничениях на сигнал управления это время может быть существенно уменьшено. В следующей главе предлагается метод синтеза, учитывающий требования по быстродействию.

### 3.5. Выводы

Рассмотрена проблема отсутствия теоретических гарантий устойчивости нейросетевых систем и предложен метод, позволяющий гарантировать асимптотическую устойчивость замкнутой нейросетевой системы управления в заданной локальной области фазового пространства. Для этого синтезируется

функция Ляпунова совместно с нейросетевым регулятором. Функция Ляпунова основана на нейронной сети, а доказательством выполнения условий устойчивости служит решение задачи целочисленного программирования для поиска значений, нарушающих эти условия. Если такого значения не найдено, значит определена функция Ляпунова, гарантирующая асимптотическую устойчивость в заданной локальной области фазового пространства.

#### 4. Методика синтеза устойчивых нейросетевых регуляторов для объектов с ограничителями.

Разработанная методика синтеза нейрорегулятора позволяет оптимизировать качество переходного процесса и гарантировать устойчивость замкнутой системы с объектом управления, имеющим ограничители, в заданной локальной области фазового пространства. В ней объединены разработанные в предыдущих главах методы и алгоритмы. В качестве нейронных сетей используются только ИНС прямого распространения. Методика состоит из следующих этапов.

1. Синтез нейроимитатора объекта. Для этого формируется обучающая выборка, согласно алгоритму из п. 2.4, и выбирается структура нейронной сети, согласно алгоритму из п. 2.1. На полученных данных выполняется обучение методом обратного распространения ошибки. Схема обучения представлена на рисунке 4.1.

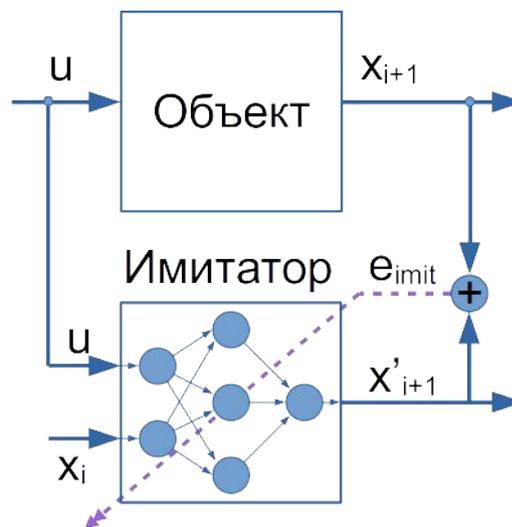


Рисунок 4.1 - Схема обучения нейроимитатора

2. Выбор структуры нейрорегулятора по алгоритму, предложенному в п. 2.1.

3. Предварительная настройка нейрорегулятора на аппроксимацию любого выбранного классического регулятора, например, LQR [4, 26], который, может быть легко синтезирован без жестких требований к качеству управления. Данное действие позволяет добиться хороших начальных условий для метода градиентного спуска, а значит стабильности получения результатов при дальнейшем обучении. Схема предварительного обучения представлена на рисунке 4.2. Также можно предварительно обучить нейрорегулятор методом инверсного нейруправления, как в п. 2.5, однако на практике такой способ показал менее стабильное получение результатов.

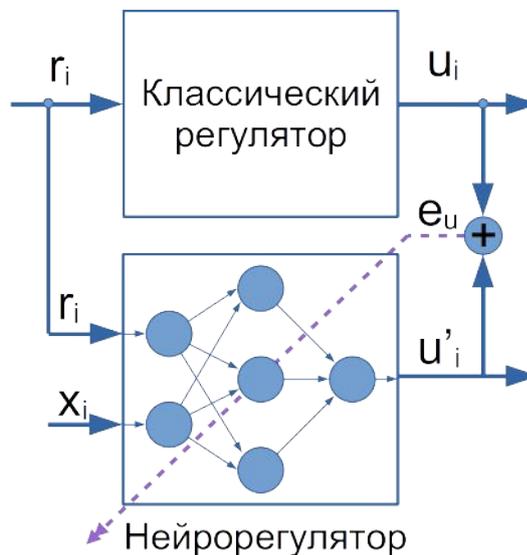


Рисунок 4.2 - Схема предварительного обучения нейрорегулятора на аппроксимацию классического регулятора

4. Обучение нейросетевого регулятора на основе нейросетевого имитатора объединенным методом, который сочетает в себе модернизированный метод пропуска ошибки через нейроимитатора из п. 2.6 и метод синтеза устойчивой нейросетевой системы управления из главы 3. Нейросетевая функция Ляпунова обучается сразу на два критерия:  $\eta_1$  (рисунок 4.3) и  $\eta_2$  (рисунок 4.4), тогда как нейросетевой регулятор только на критерий  $\eta_2$  (рисунок 4.4).

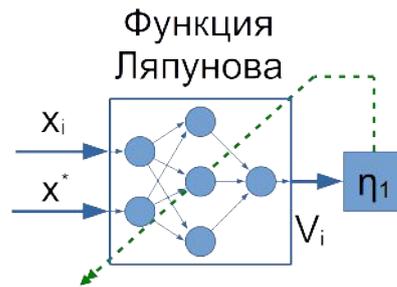


Рисунок 4.3 - Схема обучения функции Ляпунова по критерию  $\eta_1$

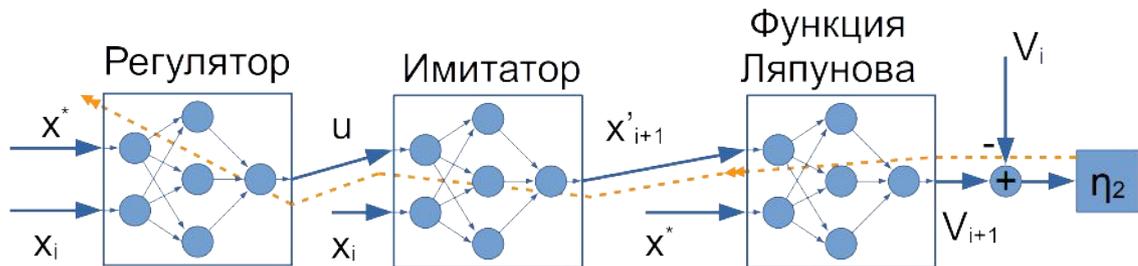


Рисунок 4.4 - Схема обучения функции Ляпунова на критерий  $\eta_2$

Объединение двух подходов позволяет получить нейросетевой регулятор, обеспечивающий устойчивость замкнутой системы в локальной области с подтверждающей функцией Ляпунова и заданным временем регулирования. Это достигается за счет решения задачи целочисленного программирования, в результате которого определяются точки в пространстве состояний, в которых максимально нарушаются условия устойчивости по Ляпунову.

Для контроля получаемого времени регулирования в процессе обучения проводится численное моделирование системы, состоящей из нейрорегулятора и нейроимитатора, с начальными условиями в найденных точках. Время моделирования устанавливается больше, чем требуемое время регулирования. Если модель не смогла стабилизироваться за отведенное время, то ошибка регулирования используется для обучения нейрорегулятора (рисунок 4.5). Причем эта ошибка берется с коэффициентом, который увеличивается с временем моделирования, тем самым сильнее «наказывая» нейронную сеть за неправильные действия. При таком подходе регулятор минимизирует время регулирования, пытаясь достигнуть требуемого. Градиент при обучении распространяется через

весь процесс моделирования, как в обучении сквозь время для рекуррентных нейронных сетей.

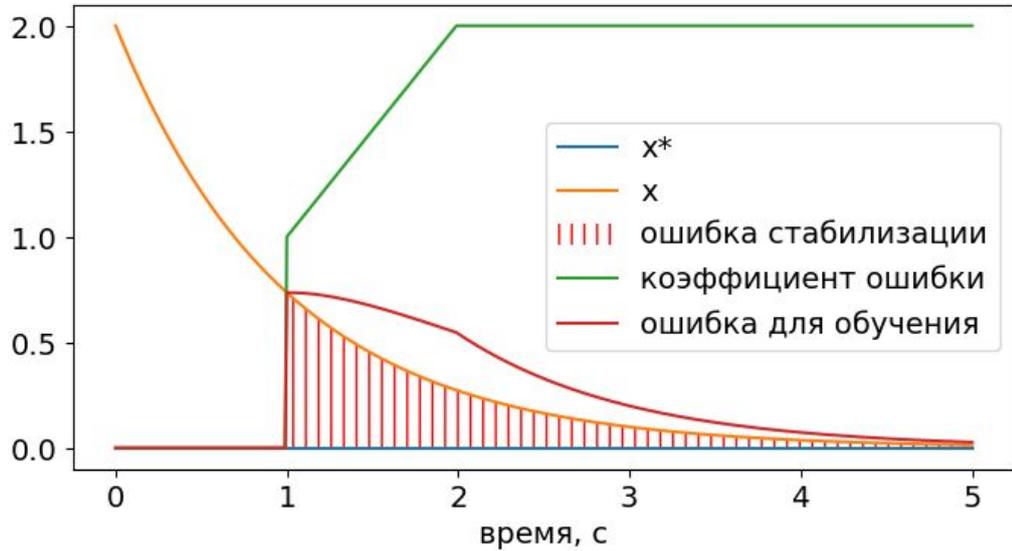


Рисунок 4.5 - График ошибки для обучения нейрорегулятора на выполнение требований по времени регулирования (на графике 1 сек.)

Таким образом, модифицированная функция потерь выглядит следующим образом:

$$\eta_3 = \sum \left( (x^* - x_i) k(t) \right)^2, \quad (4.1)$$

$$loss = \|\eta_1\| + \|\eta_2\| + \eta_3, \quad (4.2)$$

где  $\eta_3$  - квадратичная функция потерь с  $k(t)$  - коэффициент по ошибке,  $\eta_1$  и  $\eta_2$  задаются выражениями (3.19) и (3.20) соответственно.

Общая схема обучения представлена на рисунок 4.6.

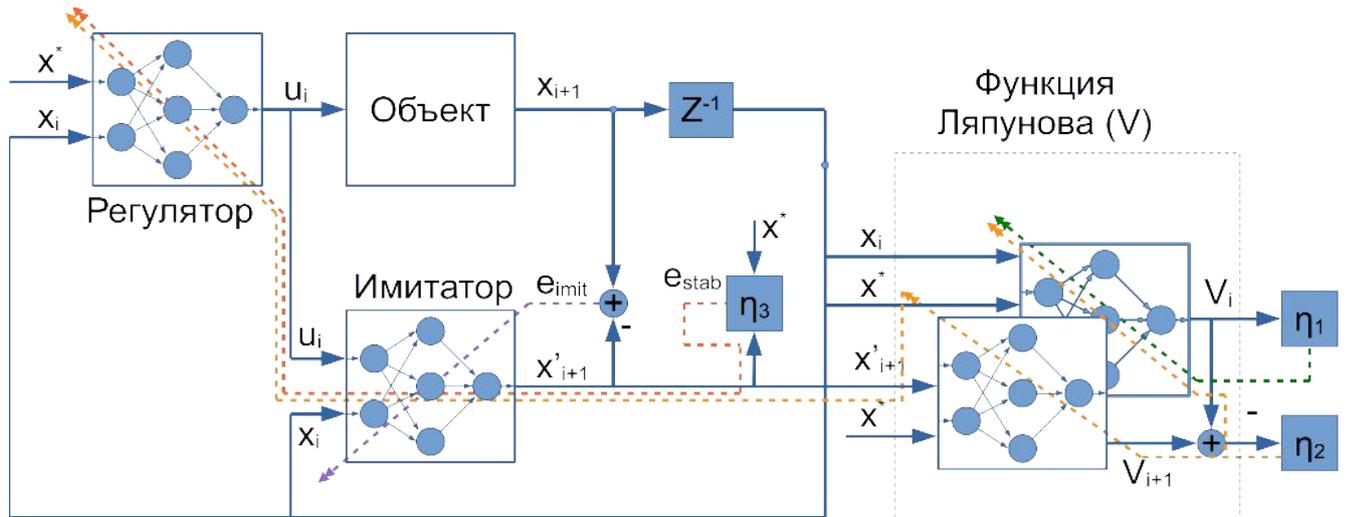


Рисунок 4.6 - Схема синтеза нейросетевого регулятора по разработанной методике

5. Во время функционирования системы можно дополнительно производить обучение в реальном времени для получения адаптивных свойств при нестационарности параметров объекта. Однако, такие свойства могут быть получены только для достаточно медленных (по сравнению со временем настройки) процессов, так как обучение требует больших вычислительных ресурсов.

### **Пример синтеза устойчивой нейросетевой системы управления перевернутым маятником с учетом времени регулирования.**

В качестве объекта рассматривается перевернутый маятник из примера синтеза устойчивой нейросетевой системы (глава 3). В отличие от указанного примера, здесь используется условие, по которому система должна стабилизироваться не более, чем за 1 сек.

Нейроимитатор не меняется, структуры нейросетевой функции Ляпунова и нейрорегулятора остаются прежними и обучаются по п. 4 данной методики.

В результате обучения функция Ляпунова получила следующий вид (рисунок 4.7). На рисунке 4.8 показан график перехода системы с перевернутым маятником из начальной точки в точку равновесия (вертикальное положение).

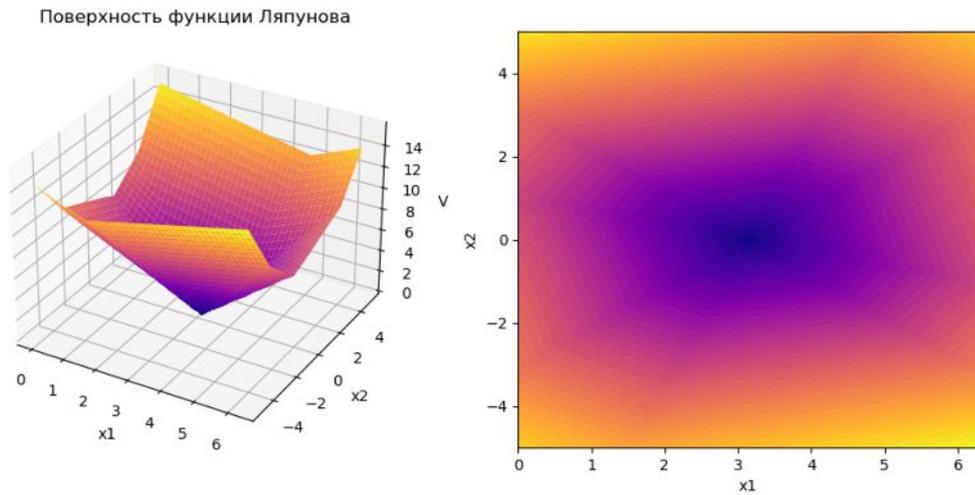


Рисунок 4.7 - Поверхность функции Ляпунова для замкнутой системы стабилизации перевернутым маятником

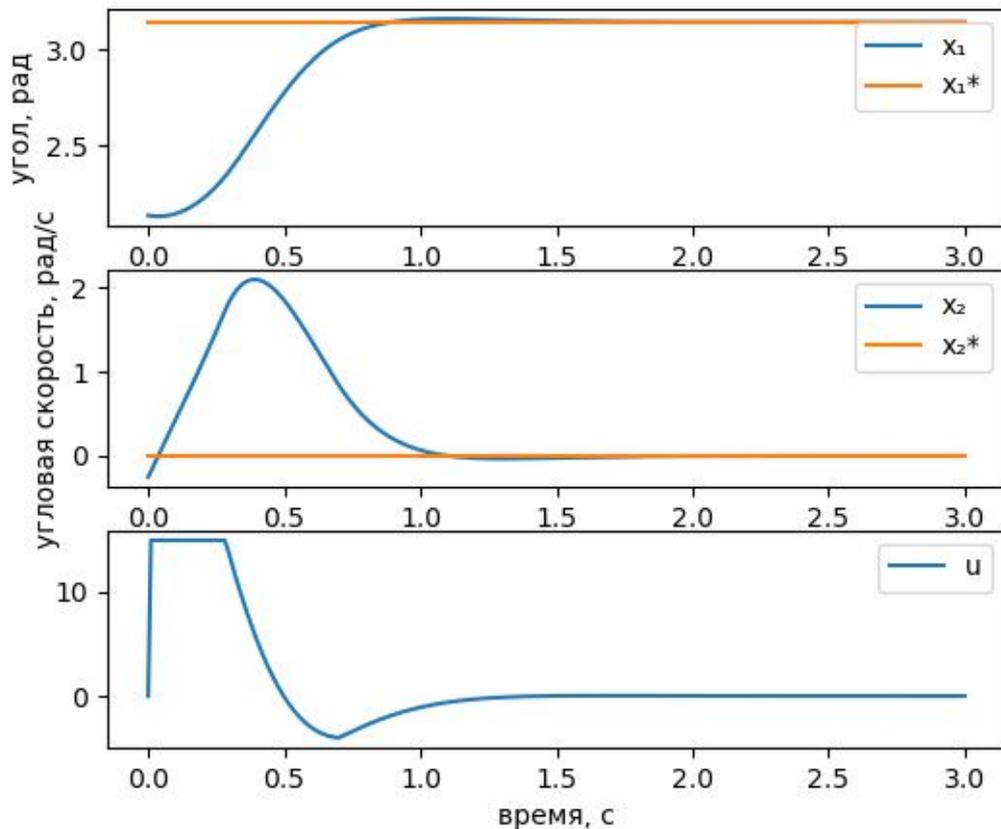


Рисунок 4.8 - График переходного процесса системы стабилизации перевернутого маятника с оптимизацией времени регулирования

Из рисунков 4.7 и 4.8 можно сделать вывод, что система устойчива в локальной области, образованной ограничением фазовых переменных (3.23).

Также видно, что система стабилизируется приблизительно за 1 сек., т. е. за требуемое время регулирования.

### Пример синтеза устойчивой замкнутой системы управления гидроприводом.

Рассмотрим пример синтеза нейросетевого регулятора по данной методике. В качестве объекта рассматривается гидропривод, упрощенная функциональная модель которого представлена на рисунке 4.9.

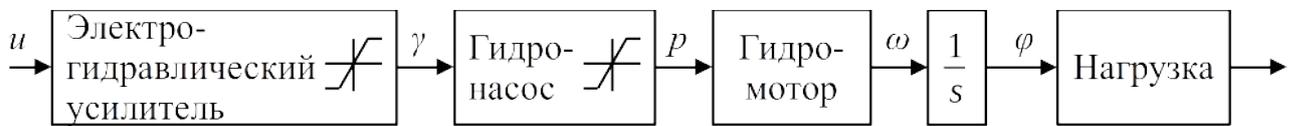


Рисунок 4.9 - Структурно-функциональная схема гидропривода

На рисунке 4.9:  $\gamma$  - угол поворота люльки гидронасоса,  $p$  - перепад давления в гидромоторе,  $\omega$  - скорость вращения выходного вала,  $\varphi$  - угол поворота вала. В структурной схеме учитываются присущие гидроприводу ограничитель на угол поворота люльки насоса:  $|\gamma| \leq D_1$  и безынерционный ограничитель перепада давления масла в силовых гидромагистралях:  $|p| \leq D_2$ .

Задачей синтезируемого регулятора является слежение за заданным гармоническим входным воздействием.

Для формирования структур нейронных сетей и синтеза нейрорегулятора достаточно структурно-функциональной схемы гидропривода (рисунок 4.9) и знания предельных значений ограничителей. Для проверки будет использоваться упрощенная модель четвертого порядка, которая описана далее.

Математическая модель гидропривода, квазиоптимальные регуляторы по быстродействию и по точности, использованные для сравнения, были разработаны в диссертации на соискание ученой степени кандидата технических наук Хоанга Ч. К. [22].

Математическую модель гидропривода можно рассматривать как систему дифференциальных уравнений десятого порядка. Структурная схема

гидропривода как объекта управления представлена на рисунке 4.10. Данная модель учитывает четыре звена с ограничителями, сухое трение и люфт редуктора.

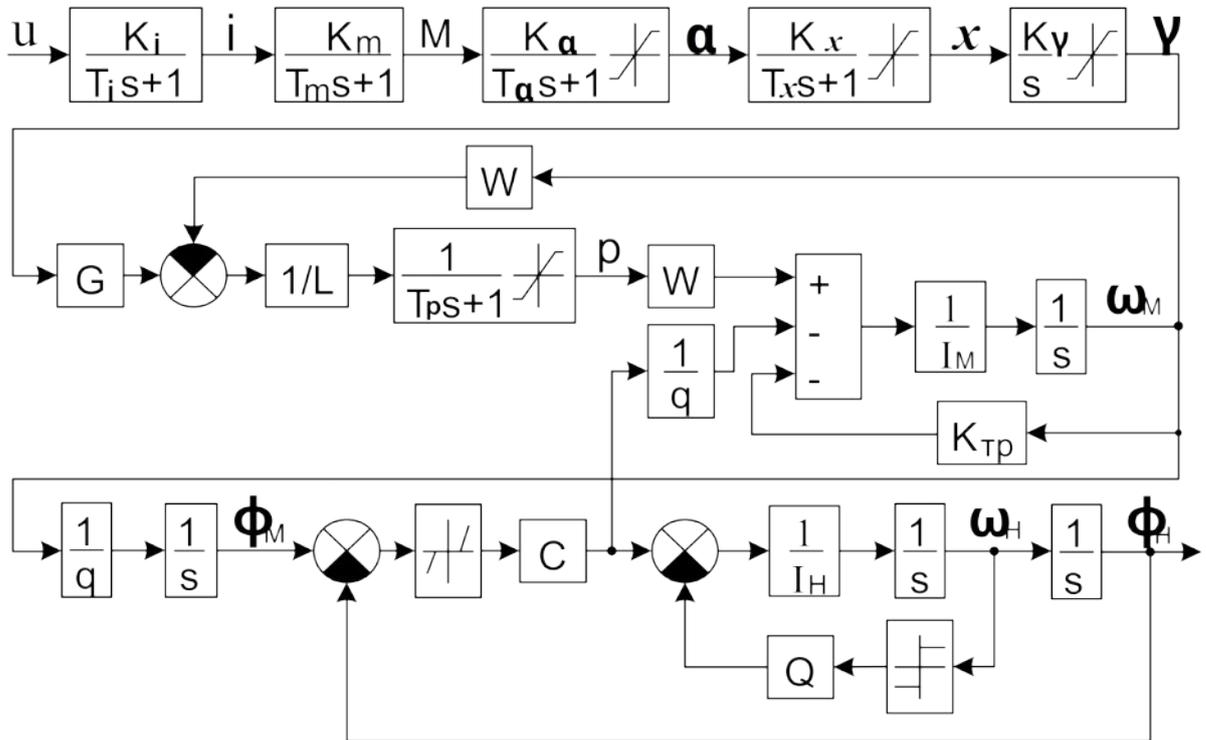


Рисунок 4.10 - Структурная схема гидропривода [76]

На рисунке 4.10:

$u$  — напряжение источника ЭДС,

$\alpha$  — угол поворота якоря,

$x$  — перемещение золотника,

$y$  — угол поворота люльки гидронасоса,

$G$  - коэффициент усиления насоса регулируемой производительности по расходу,

$L$  — суммарный коэффициент утечек жидкости,

$W$  — удельный объем гидромотора и гидронасоса,

$E$  — приведенное значение модуля объемной упругости жидкости,

$T_p = L * E / V$ ,

$p$  — перепад давления в гидромоторе,

$q$  — передаточное отношение редуктора,

$I_m$  — момент инерции ротора гидромотора,

$K_{mp}$  — коэффициент вязкого трения гидромотора,

$\omega_m$  — угловая скорость на выходном валу редуктора,

$\phi_m$  — угол поворота редуктора,

$C$  — коэффициент жесткости силовой проводки,

$I_n$  — момент инерции нагрузки,

$\omega_n$  — угловая скорость нагрузки,

$\phi_n$  — координата движения массы нагрузки, приведенная к оси гидромотора.

В полной модели содержатся звенья, имеющие малые и большие постоянные времени, которые отличаются на несколько порядков. Упрощенная модель гидропривода, полученная приравниванием нулю малых постоянных времени, представлена на рисунке 4.11.

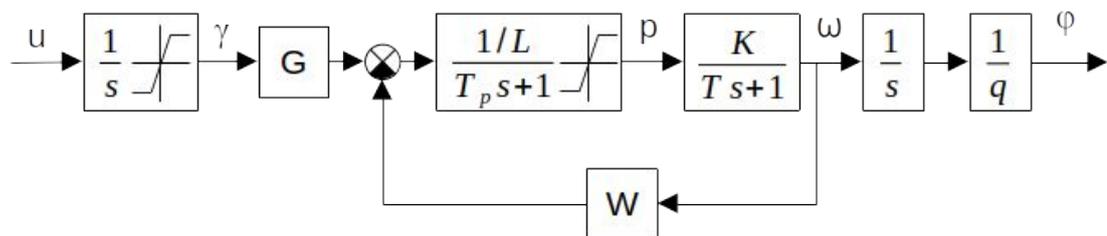


Рисунок 4.11 - Структурная схема базовой системы гидропривода

На рисунке 4.11:  $\gamma$  - угол поворота люльки гидронасоса,  $p$  - перепад давления в гидромоторе,  $\omega$  - скорость вращения выходного вала гидромотора,  $\phi$  - угол поворота вала после редуктора,  $G$  - коэффициент усиления насоса регулируемой производительности по расходу,  $L$  - суммарный коэффициент утечек жидкости;  $T_p = L * E / V$ , где  $E$  - приведенное значение модуля объемной упругости жидкости,  $V$  - объём жидкости в магистрали высокого давления.

В структурной схеме учитываются присущие гидроприводу ограничитель на угол поворота люльки насоса:  $|\gamma| \leq D_1$  и ограничитель (в данной структурной

схеме – безынерционный) перепада давления масла в силовых гидромагистралях:  
 $|p| \leq D_2$ .

Параметры схеме базовой системы гидропривода:

$$\begin{aligned}
 D_1 &= 0.523333333 \text{ рад}; \\
 y_{up} &= D_1; \\
 y_{lo} &= -D_1; \\
 G &= 0.0066879 \text{ Па}; \\
 E &= 1.4e9 \text{ Н/м}^2; \\
 V &= 1e-3 \text{ м}^3; \\
 L &= 1e-11 \text{ м}^5/(\text{Н с}); \\
 W &= 2.2293e-5 \text{ м}^3/\text{рад}; \\
 D_2 &= 32e6 \text{ Па}; \\
 p_{up} &= D_2; \\
 p_{lo} &= -D_2; \\
 K &= 9.2887e-5 \text{ м}^2/(\text{Н м с}); \\
 T &= 6.9479 \text{ с}; \\
 q &= 100; \\
 Ts &= 0.001 \text{ с};
 \end{aligned} \tag{4.3}$$

Далее, для проверки полученных регуляторов будет использована базовая математическая модель гидропривода четвертого порядка.

### **Выбор структур нейронных сетей.**

В качестве нейросетевого имитатора выбирается структура по алгоритму из п. 2.2. То есть, согласно схеме, объект управления рассматривается как модель четвертого порядка, что определяет 5 нейронов на входе, в первом скрытом слое 4 нейрона без функций активации, во втором слое 12 нейронов с функцией активации ReLU и с весами смещения, в выходном слое 4 нейрона. Так как значения фазовых переменными сильно различаются по величине, то нейронная сеть должна обучаться и работать с нормализованными данными, что позволит получить более качественное обучение. В этом случае значения верхнего и нижнего ограничений используются для нормализации данных, а в структуре эти значения заменяются на -1 и 1. Таким образом можно легко сформировать третий

и четвертый (выходной) слой, согласно алгоритму, так как в них весовые коэффициенты могут быть равными только 1 или -1.

Так как весовые коэффициенты последнего слоя уже известны, то оставшиеся слои обучаются достаточно быстро и с ошибкой близкой к нулю. По этой причине нейроимитатор представлен в следующем виде:

$$x_{i+1} = \phi_{\text{dyn}}(x_i, u_i). \quad (4.4)$$

Для обучения нейроимитатора формируется выборка согласно алгоритму из п. 2.4, которая нормализуется. В результате обучения квадратичная ошибка имеет порядок  $10^{-30}$ .

В качестве структуры для нейрорегулятора выбрана сеть прямого распространения с 4-мя входами (вектор состояния), двумя скрытыми слоями, в первом 24 нейрона, во втором 8 нейронов с функциями активации leakyReLU и на выходе один нейрон, формирующий сигнал на объект управления. Нейрорегулятор представлен в виде (3.5).

Функция Ляпунова представлена 4-мя нейронами на входе, двумя скрытыми слоями с функцией активации leakyReLU и одним нейроном на выходе. Первый скрытый слой имеет 24 нейрона, второй 8 нейронов. Функция Ляпунова представлена в виде (3.13).

Структура нейросетевой системы, состоящей из нейроимитатора, нейрорегулятора и нейросетевой функции Ляпунова представлена на рисунке 4.12.

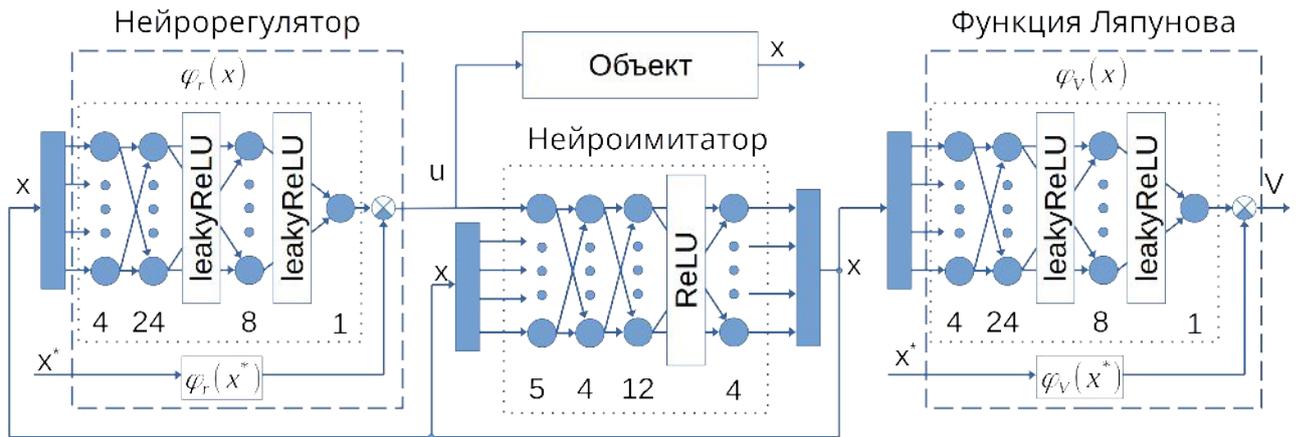


Рисунок 4.12 - Структура нейронных сетей в нейросетевой системе, используемой при обучении нейрорегулятора

**Обучение нейрорегулятора.**

Согласно разработанной методике нейрорегулятор можно предварительно обучить либо модифицированным методом инверсного нейруправления, либо аппроксимировать имеющийся классический регулятор. В данном примере выбран второй способ. Средствами Matlab был синтезирован линейно-квадратичный регулятор (LQR) по модели объекта, в которой не учитывались ограничители. Схема модели, для которой был синтезирован LQR, представлена на рисунке 4.13.

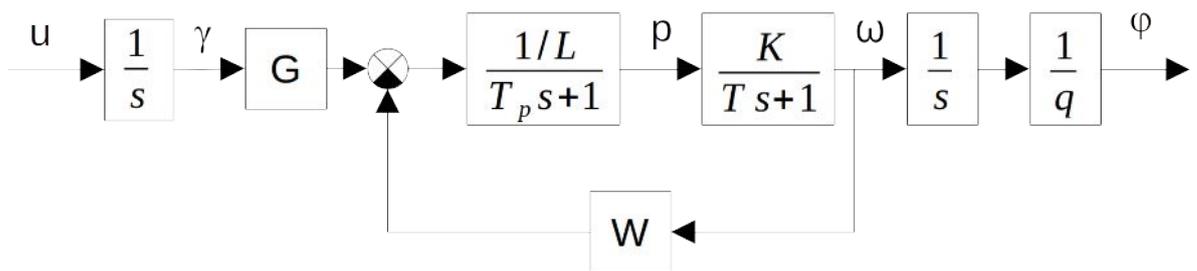


Рисунок 4.13 - Схема линейной базовой модели гидропривода

При синтезе LQR минимизируется следующий критерий оптимальности:

$$J = \int_0^{\infty} (x^T Q(t)x + u^T R(t)u) dt, \tag{4.5}$$

где  $R$  и  $Q$  — симметрично положительные определенные матрицы.

При этом непрерывный линейный объект управления описывается в пространстве состояний:

$$\dot{x} = A(t)x + B(t)u, \quad (4.6)$$

где  $A$  — матрица системы,  $B$  — матрица управления,  $u$  — вектор управления объектом.

Закон управления имеет вид:

$$u = -R^{-1} B^T P x, \quad (4.7)$$

где  $P$  находится из решения уравнения Риккати:

$$A^T P + P A - P B R^{-1} B^T P + Q = -\dot{P}. \quad (4.8)$$

Таким образом, при синтезе линейно-квадратичного регулятора разработчику необходимо стандартным методом выбрать матрицы  $R$  и  $Q$ .

Для предварительного обучения нейрорегулятора использовался линейно-квадратичный регулятор, синтезированный с матрицами

$$R = [1], \quad (4.9)$$

и

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}. \quad (4.10)$$

При выборе этих параметров главным фактором служило то, чтобы система отслеживала входной сигнал, но без требований к качеству управления. В результате решения средствами Matlab получается вектор коэффициентов для обратных связей по каждой фазовой переменной.

Значения этих коэффициентов:

$$k_1 = 4.380217, k_2 = 1.024574e-09, k_3 = 0.0010653, k_4 = 3.162278. \quad (4.11)$$

Так как в параметре по регулируемой фазовой переменной коэффициент  $k_4$  не равен 1, то необходимо сигнал управления умножить на данный коэффициент. Обучение заняло 5 минут, ошибка в конце обучения составила  $1.3 \cdot 10^{-5}$ . Графики

сравнения синтезированного LQR и аппроксимирующего нейросетевого регулятора представлены на рисунке 4.14.

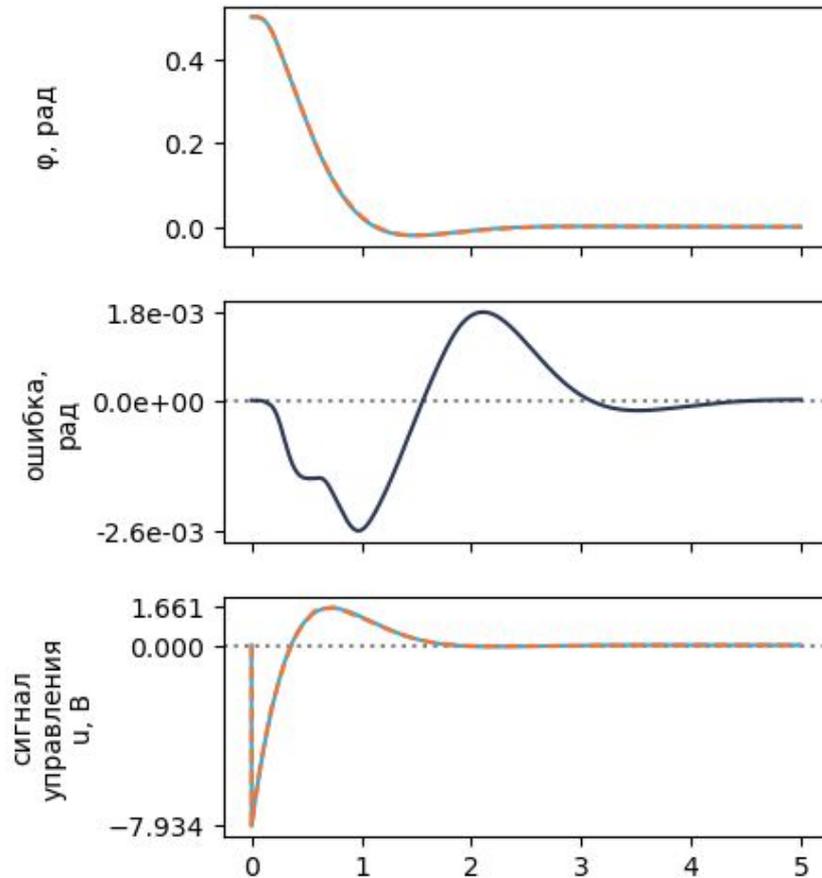


Рисунок 4.14 - Переходный процесс из начальных условий в нулевое положение системы управления с линейно квадратичным и нейросетевым регулятором

Далее, было выполнено окончательное обучение регулятора с требованием к времени регулирования, линейно зависящим от разницы между текущим угловым положением и заданным (функция сформирована на основе реакции объекта на сигнал, при котором достигаются оба ограничителя).

#### **Преобразование системы для слежения за входными сигналами.**

По данной методике система обучается как автономная, переходящая из начального состояния в состояние равновесия  $x^*$ . В данном методе система будет следить за входным сигналом, если его подать в качестве требуемой точки равновесия  $x^*$ . Если сигнал статический, то все компоненты вектора, кроме угла

поворота гидромотора, будут равны 0. Однако, если так поступить с динамическим сигналом, то ошибка слежения резко возрастает. Для исключения этого необходимо из динамики входного сигнала вычислить требуемый вектор состояния объекта управления, которое соответствует входному сигналу в каждый дискретный такт.

Так как на вход нейрорегулятора подается заданный вектор состояния  $x^*$ , то необходимо преобразовать входное воздействие в указанный вектор. Для этого можно воспользоваться либо математической моделью, если она имеется, либо выполнить обучение нейронной сети. Здесь рассматривается второй способ, так как предполагается, что математического описания объекта управления не существует.

Обучение нейронной сети, выполняющей роль преобразователя задающего сигнала управления системой в заданный вектор состояния  $x^*$ , основывается на аппроксимации сформированной тренировочной выборки. Для этого нейроимитатор моделируется из случайного начального состояния с нулевым сигналом управления на количество тактов, равное порядку объекта (предполагаемый порядок объекта, на который обучался нейроимитатор) минус 1 ( $n-1$ ). На последнем такте результатом будет вектор состояния, который рассматривается как целевой результат обучения. В качестве входа в нейронную сеть используются значения управляемой фазовой переменной с каждого такта моделирования. Схема обучения для объекта 3-го порядка представлена на рисунке 4.15.

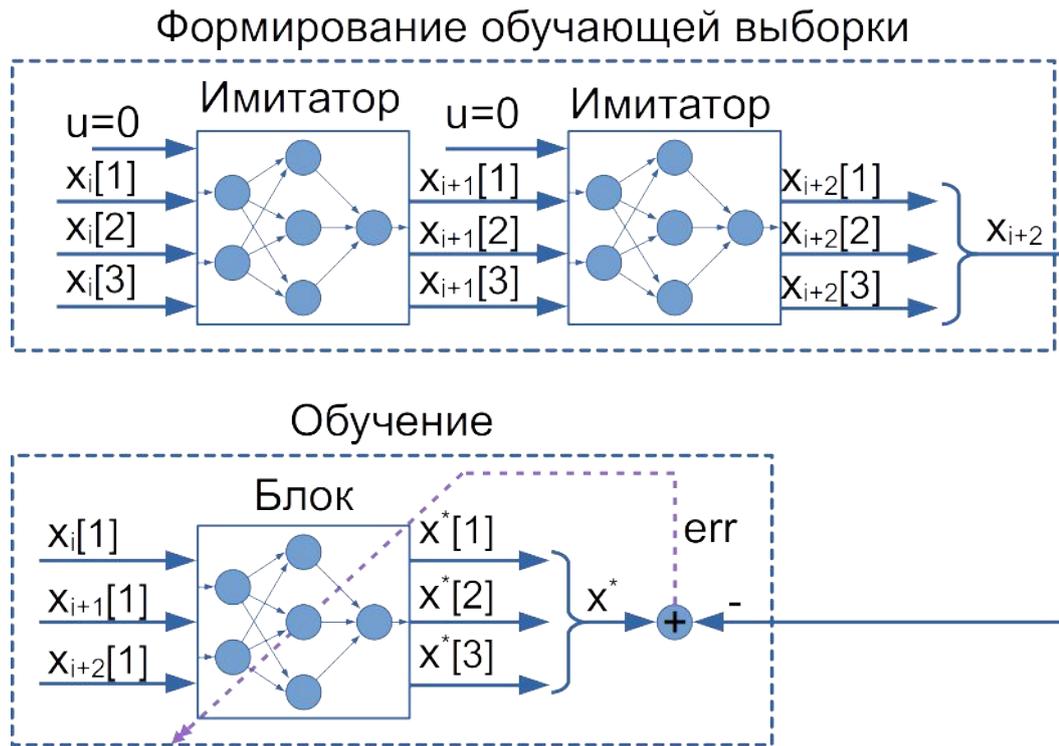


Рисунок 4.15 - Схема обучения нейронной сети преобразованию сигнала управления системой в заданный вектор состояния для объекта 3 порядка

После обучения такая нейронная сеть позволит получить качественное слежение за различными динамическими задающими сигналами. Более подробное изучение способов формирования данного блока в виде нейронной сети требует дальнейших исследований.

Структура нейронной сети, преобразующей задающий сигнал управления системой в заданный вектор состояния  $x^*$ , для гидропривода была выбрана в виде одного слоя без функций активации размером 4 нейрона (равным размеру вектора состояния). Обучение такой нейронной сети выполнялось на основе обученной линейной части нейроимитатора (матрица весовых коэффициентов, соединяющая входной и первый скрытый слой ИНС, которая по сути является объединением матрицы  $A$  и вектора  $B$  из модели дискретного пространства состояний). На рисунке 4.16 приведена схема обучения и структуры нейронных сетей для гидропривода.

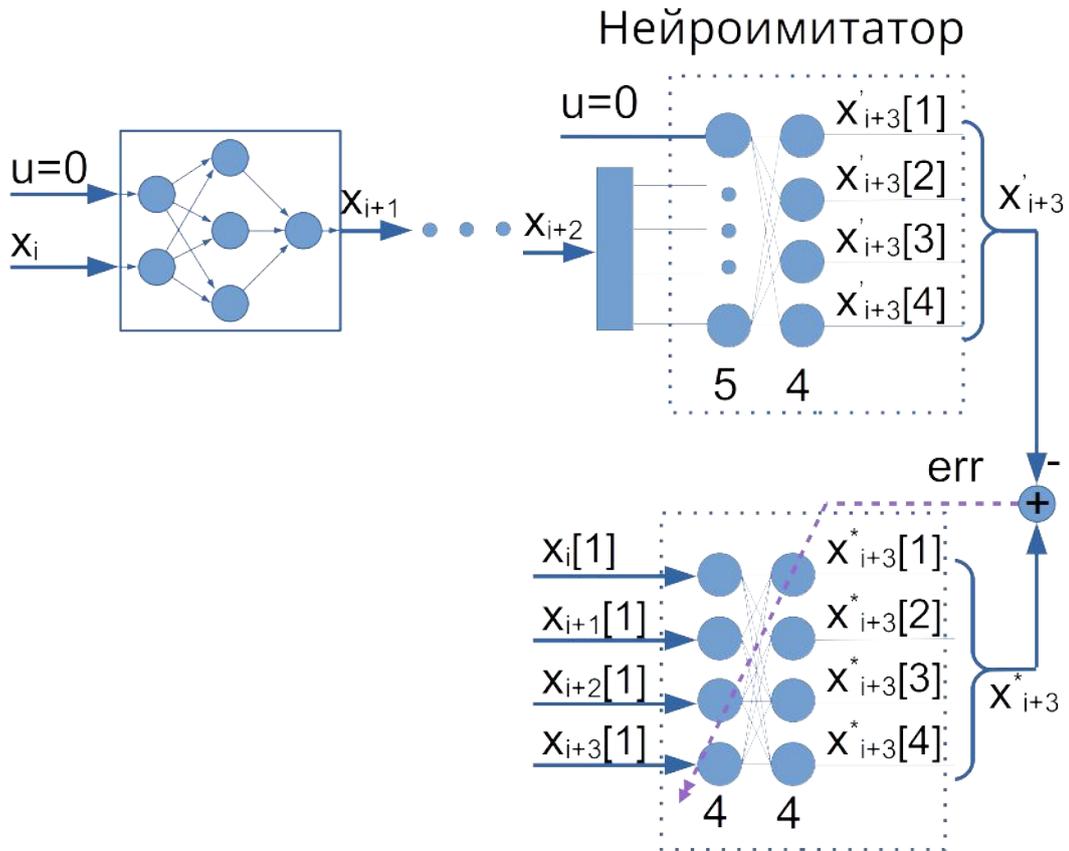


Рисунок 4.16 - Схема обучения нейронной сети, преобразующей входное воздействие в заданный вектор состояния, для гидропривода, обучаемого на линейной части нейроимитатора

Так как данная нейронная сеть состоит из одного слоя, то для рассматриваемого примера преобразование задающего сигнала управления системой в заданный вектор состояния  $x^*$  можно описать матрицей 4 на 4.

Результаты моделирования полученной системы управления с нейрорегулятором сравниваются с тремя системами, синтезированными классическими методами. В первой системе в качестве регулятора используется линейно-квадратичный, во второй — квазиоптимальный по точности, в третьей — квазиоптимальный по быстродействию. Рассмотрим синтез таких регуляторов.

### Синтез LQR для управления гидроприводом.

Для сравнения по линейной модели (рисунок 4.13) был синтезирован линейно-квадратичный регулятор с увеличенным коэффициентом по быстродействию в матрице  $Q$  с параметрами:

$$R=[1], \quad (4.12)$$

и

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1000 \end{bmatrix}. \quad (4.13)$$

Коэффициенты обратных связей равны:

$$k_1=14.490713, k_2=1.121325 \cdot 10^{-8}, k_3=0.0120428, k_4=31.622777. \quad (4.14)$$

Так как такой регулятор синтезировался по модели без учета нелинейностей, то его работа на модели с ограничителями отличается (рисунок 4.17).

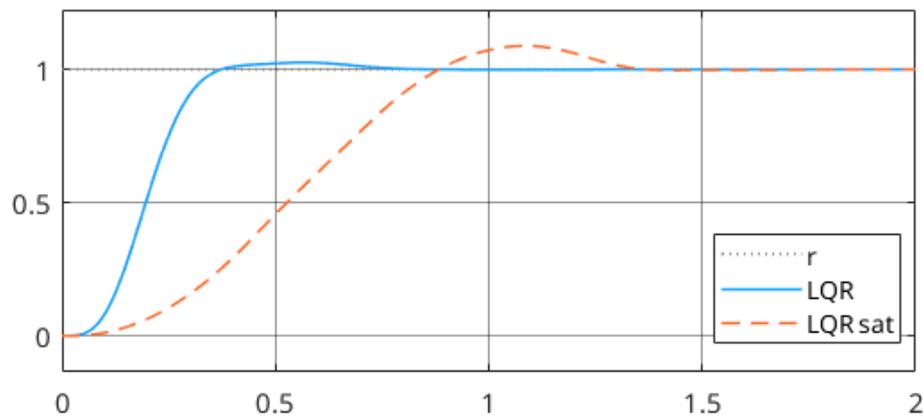


Рисунок 4.17 - Переходный процесс замкнутой системы с гидроприводом с LQR без учета ограничителей и с их учетом

Результат работы системы управления гидроприводом с линейно-квадратичным и с нейросетевым регулятором представлены на рисунках 4.18 - 4.21.

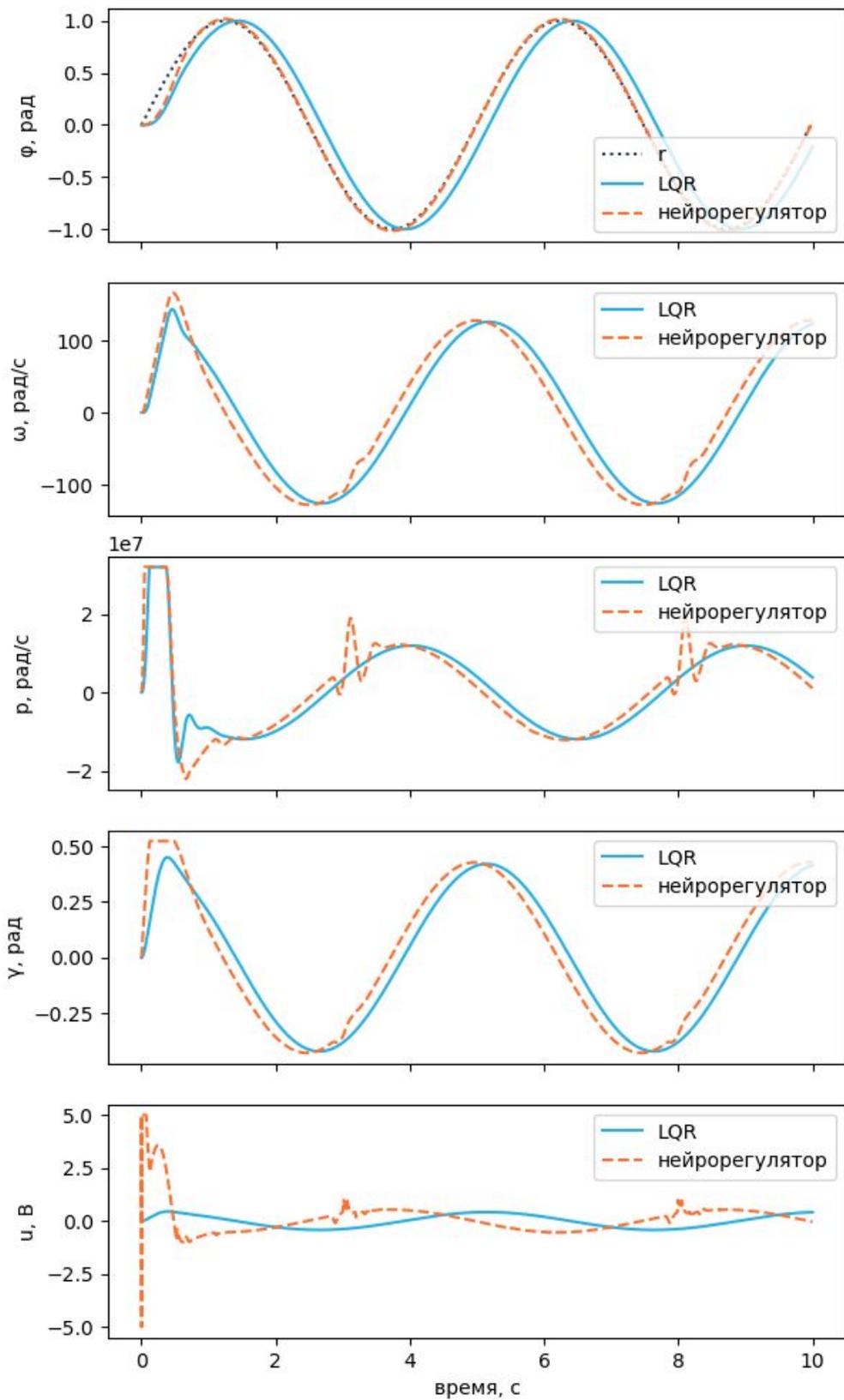


Рисунок 4.18 - Реакция систем управления гидроприводом с линейно-квадратичным и нейросетевым регуляторами на гармонический сигнал (частота 1.27 рад/с, амплитуда 1 рад)

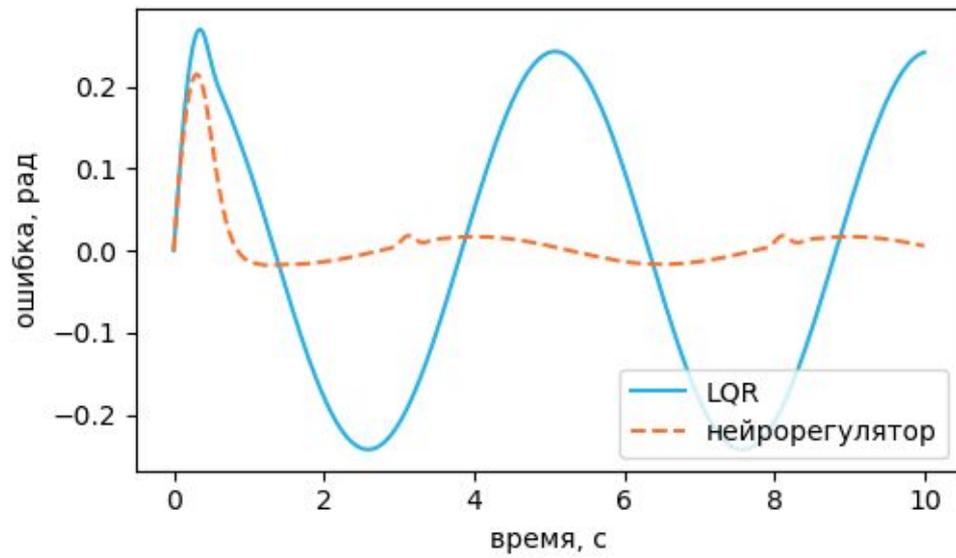


Рисунок 4.19 - Ошибки слежения систем управления гидроприводом с линейно-квадратичным регулятором и нейрорегулятором за гармоническим сигналом (частота 1.27 рад/с, амплитуда 1 рад)

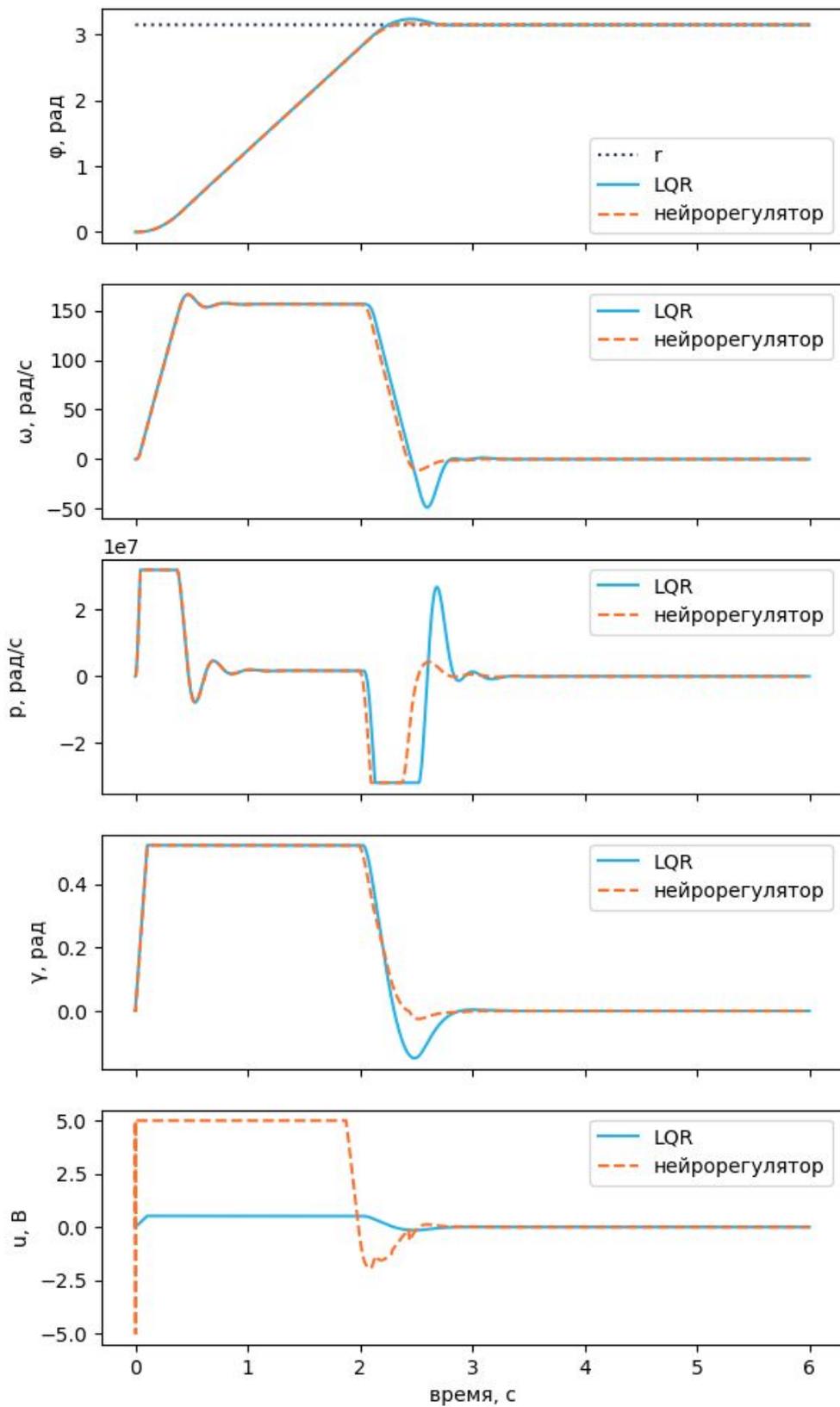


Рисунок 4.20 - Реакция систем управления гидроприводом с линейно-квадратичным и нейросетевым регулятором на ступенчатый сигнал амплитудой  $\pi$  рад

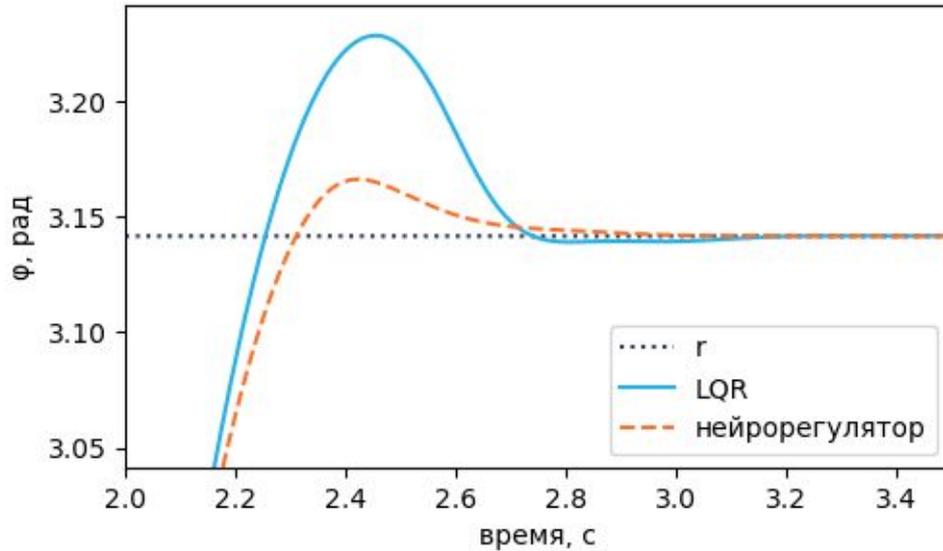


Рисунок 4.21 - Реакции систем управления гидроприводом с линейно-квадратичным и нейросетевым регулятором на ступенчатый сигнал амплитудой  $\pi$  рад (масштабированный угол)

Как видно из рисунка 4.19, система с нейрорегулятором точнее следует за входным сигналом (ошибка 0.24 рад. с LQR и 0.018 рад. с нейрорегулятором). Из рисунка 4.21 видно, что система с нейрорегулятором отрабатывает переброс с меньшим перерегулированием и приблизительно с той же скоростью.

### Квазипотимальные регуляторы для управления гидроприводом

Как было сказано ранее, квазиоптимальные регуляторы по быстродействию и по точности были разработаны в диссертации Хоанга Ч. К. [22].

Оба квазипотимальных регулятора имеют следующий вид:

$$u = A \operatorname{sign}[r + R \dot{r} - \phi - d_1 \dot{\gamma} - d_2 p - d_3 \omega] \quad (4.15)$$

где:  $A$  — значение усиления реле,  $r$  — сигнал управления системой,  $R$  — коэффициент производной управляющего сигнала,  $d_1$ ,  $d_2$ ,  $d_3$  — коэффициенты обратных связей.

$$R = (d_1(\frac{L}{GK} + \frac{W}{G}) + \frac{d_2}{K} + d_3) \cdot 100 \quad (4.16)$$

Параметры квазиоптимального регулятора по точности для базовой системы:

$$d_1 = 0.069284; d_2 = 9.966 \cdot 10^{-11}; d_3 = 4.310533 \cdot 10^{-4} \quad (4.17)$$

Параметры квазиоптимального регулятора по быстродействию для базовой системы:

$$d_1 = 0.127446; d_2 = 1.184876 \cdot 10^{-9}; d_3 = 1.615897 \cdot 10^{-3} \quad (4.18)$$

В работе [22] сформирован комплексный закон управления, переключающий синтезированные квазиоптимальные регуляторы. В данном примере отдельно сравнивается работа нейросетевого регулятора с каждым из них. Результаты работы систем управления с нейрорегулятором и квазиоптимальным по точности регулятором приведены на рисунках 4.22 - 4.25. Результаты работы систем управления с нейрорегулятором и квазиоптимальным по быстродействию регулятором приведены на рисунках 4.26 - 4.30.

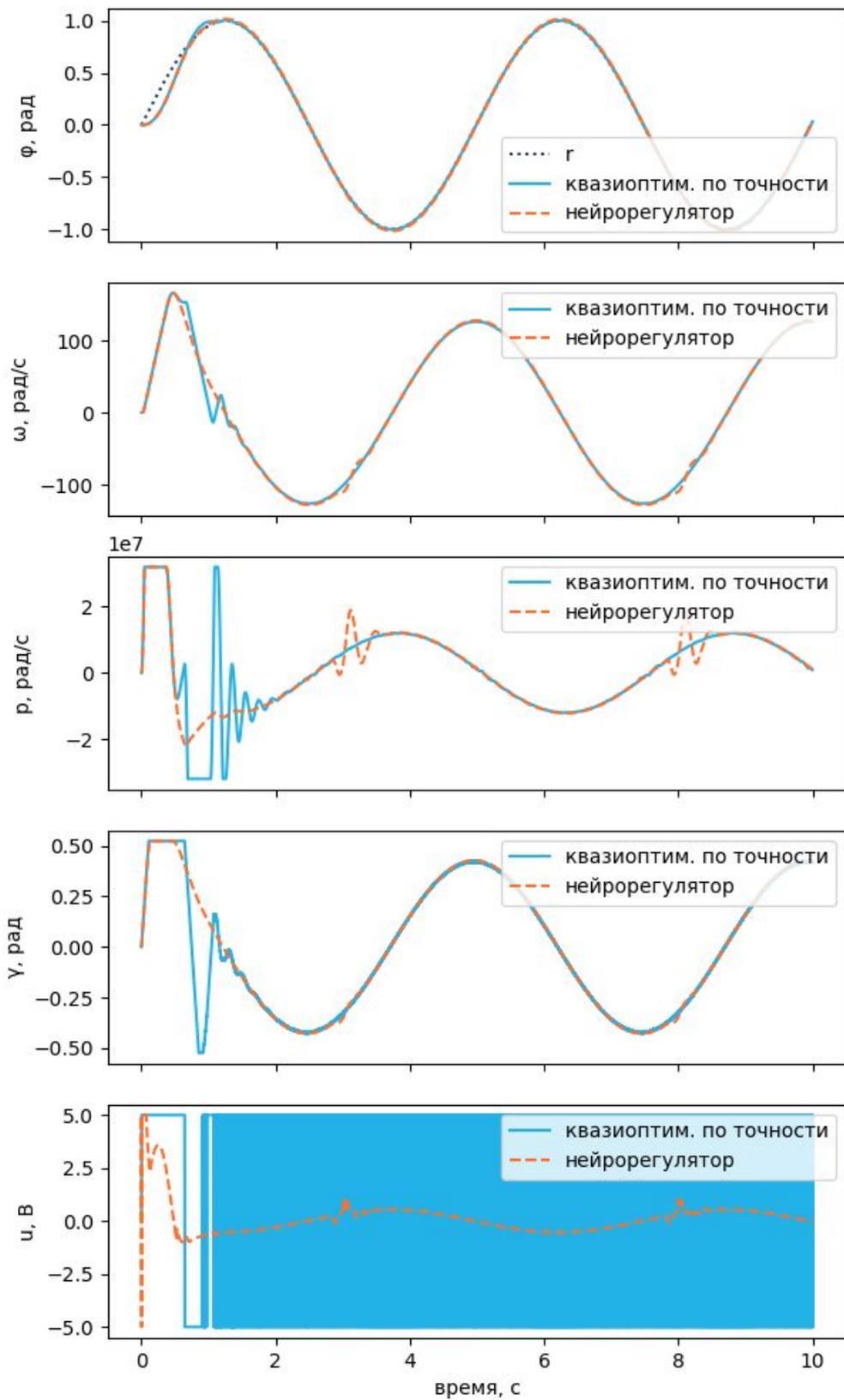


Рисунок 4.22 - Реакция систем управления гидроприводом с квазиоптимальным по точности и нейросетевым регуляторами на гармонический сигнал (частота 1.27 рад/с, амплитуда 1 рад)

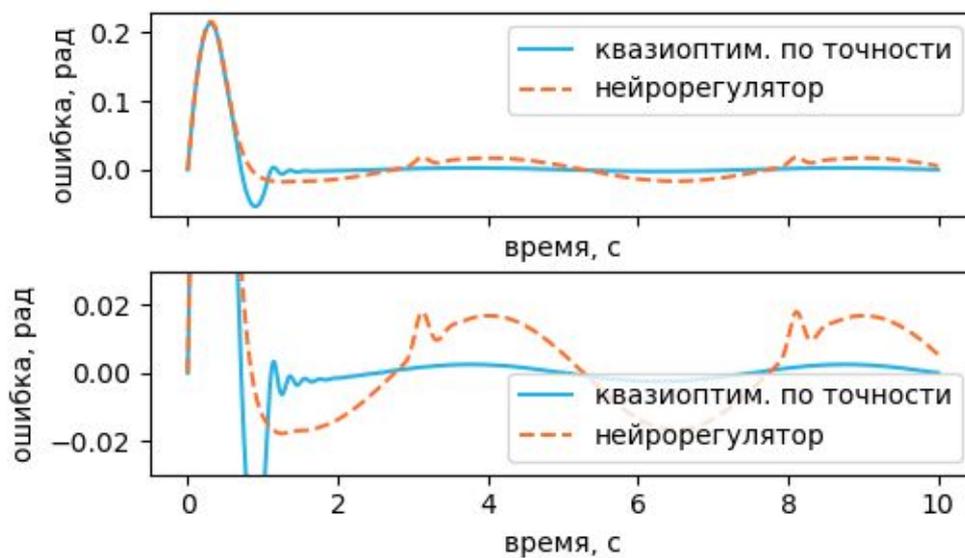


Рисунок 4.23 - Ошибка слежения систем управления гидроприводом с квазиоптимальным по точности и нейросетевым регуляторами за гармоническим сигналом (частота 1.27 рад/с, амплитуда 1 рад)

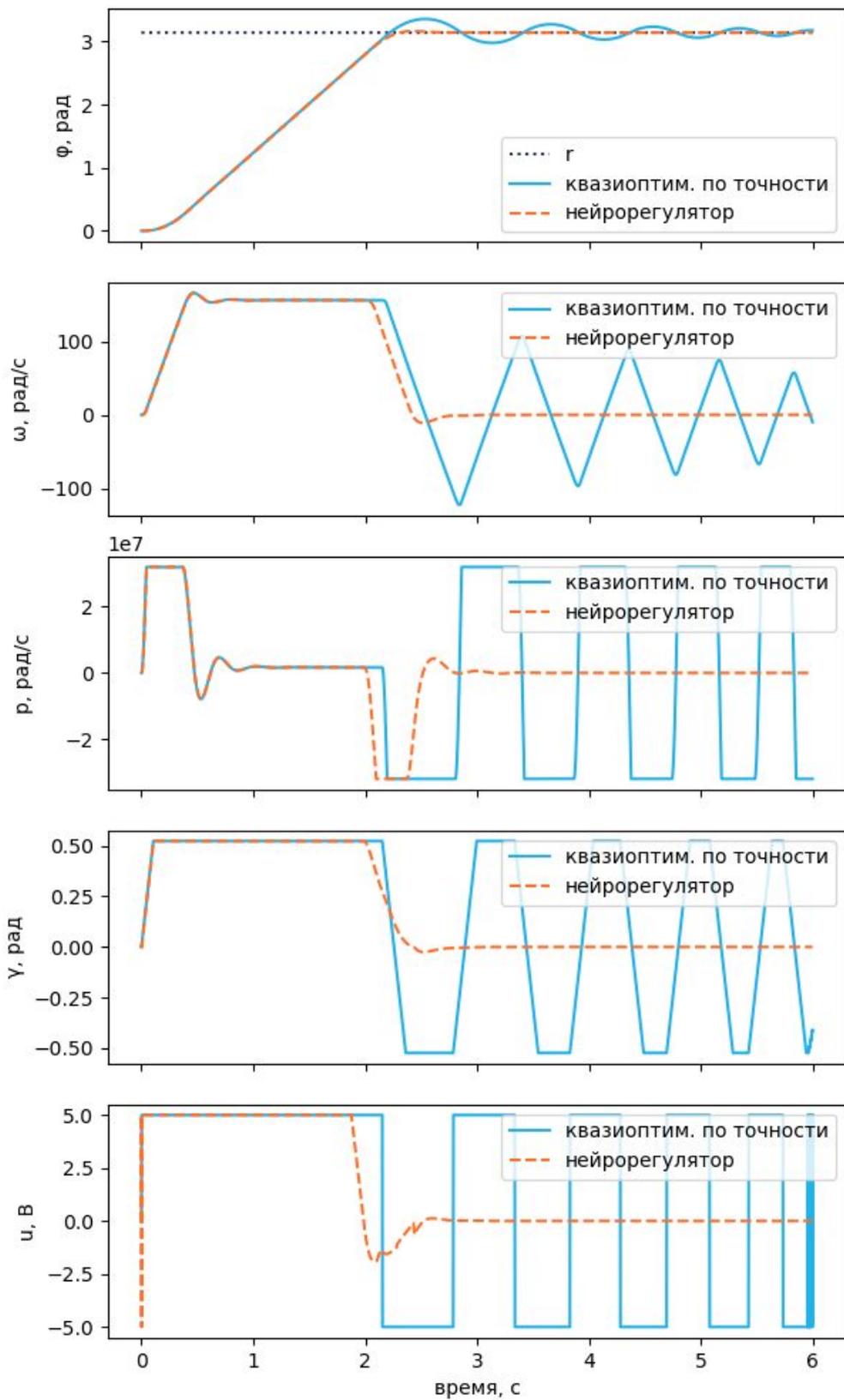


Рисунок 4.24 - Реакция систем управления гидроприводом с квазиоптимальным по точности и нейросетевым регуляторами на ступенчатый сигнал амплитудой  $\pi$  рад

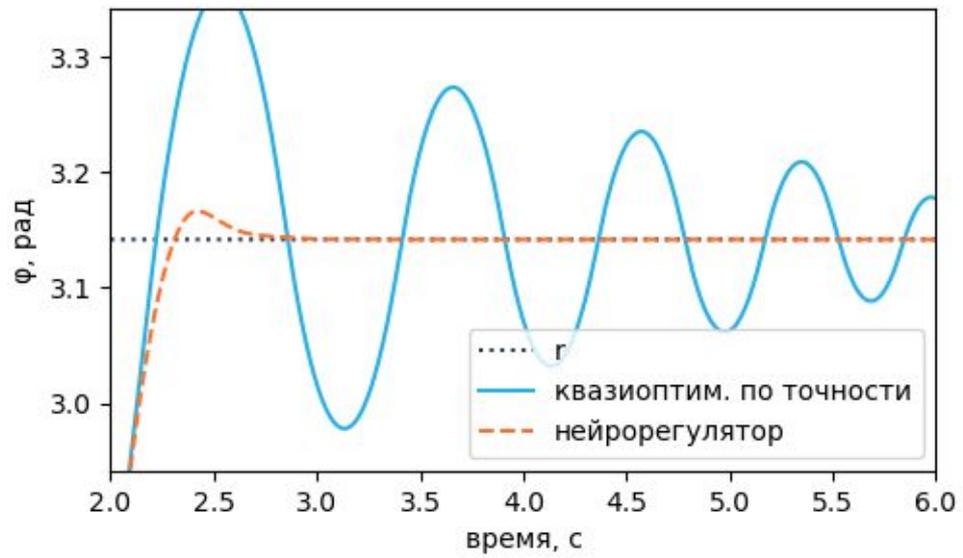


Рисунок 4.25 - Реакция систем управления гидроприводом с квазиоптимальным по точности и нейросетевым регуляторами на ступенчатый сигнал амплитудой  $\pi$  рад (масштабированный угол)

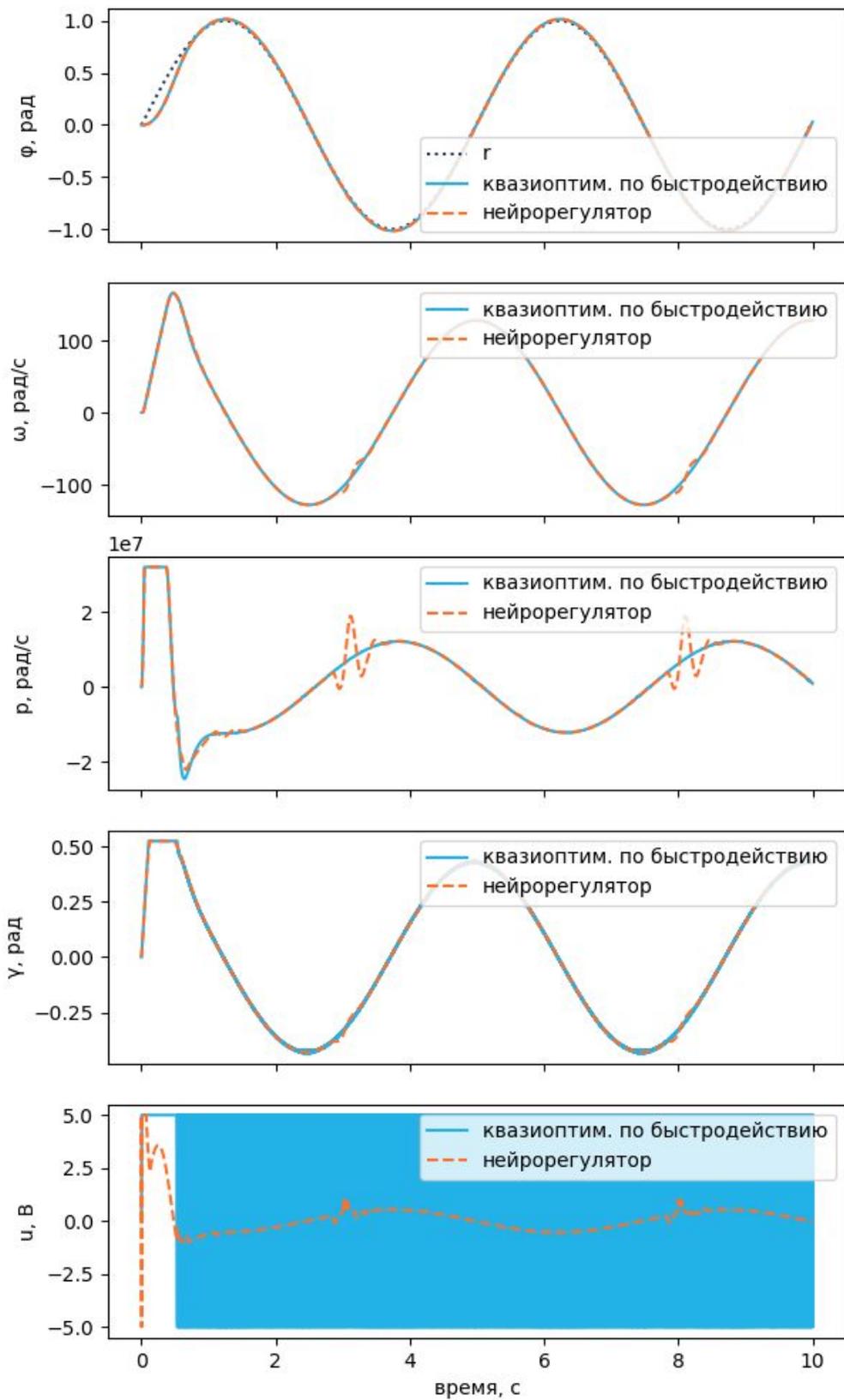


Рисунок 4.26 - Реакция систем управления гидроприводом с квазиоптимальным по быстродействию и нейросетевым регуляторами на гармонический сигнал (частота 1.27 рад/с, амплитуда 1 рад)

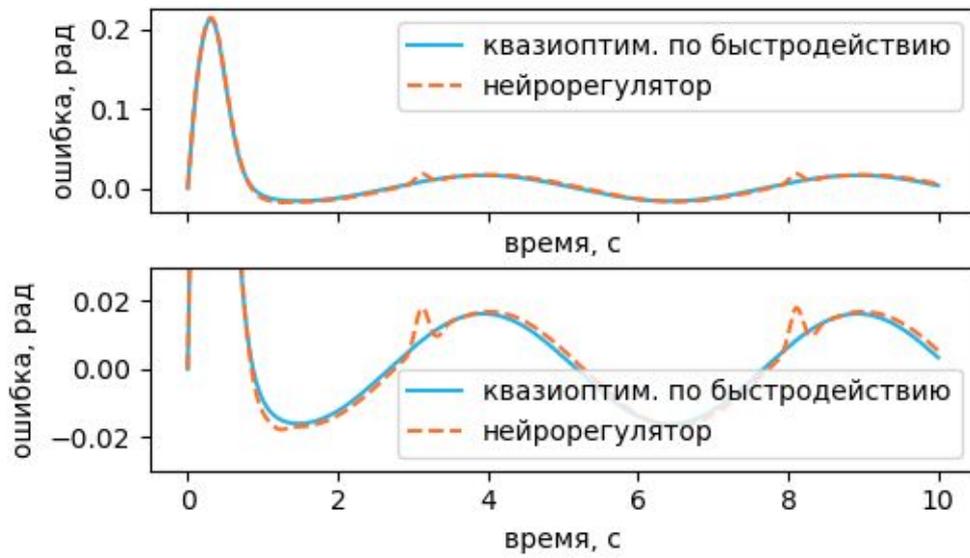


Рисунок 4.27 - Ошибка слежения систем управления гидроприводом с квазиоптимальным по быстродействию и нейросетевым регуляторами за гармоническим сигналом (частота 1.27 рад/с, амплитуда 1 рад)

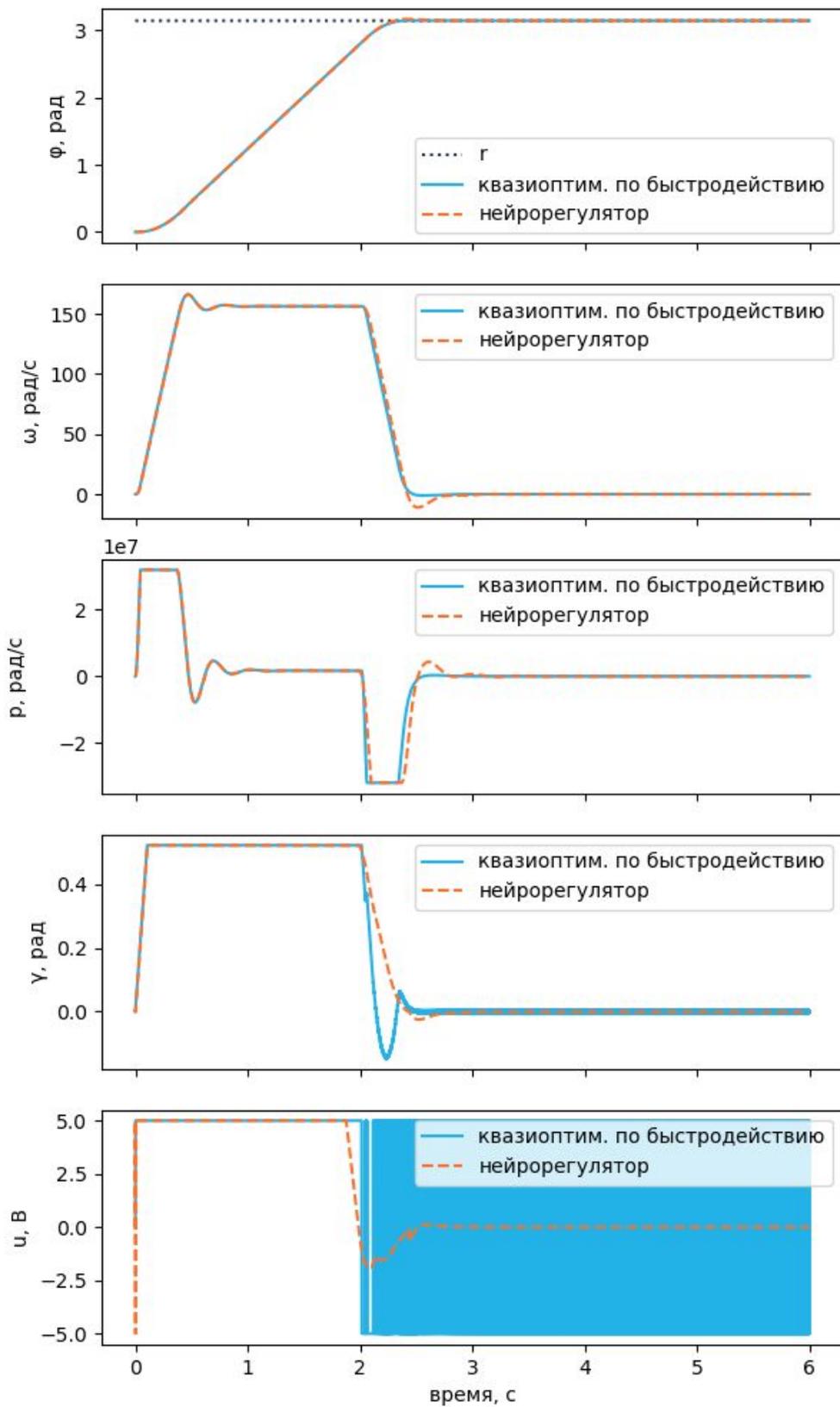


Рисунок 4.28 - Реакция систем управления гидроприводом с квазиоптимальным по быстродействию и нейросетевым регуляторами на ступенчатый сигнал амплитудой  $\pi$  рад

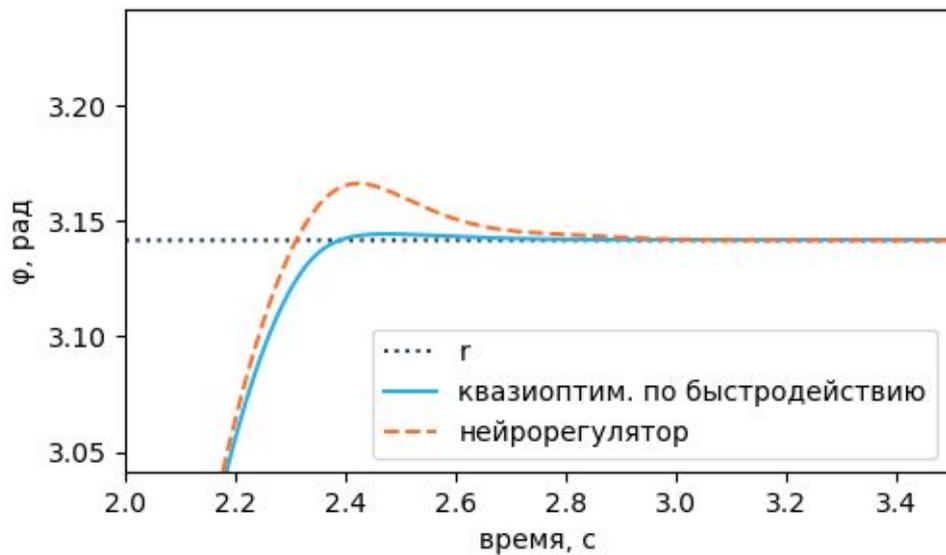


Рисунок 4.29 - Реакция систем управления гидроприводом с квазиоптимальным по быстродействию и нейросетевым регуляторами на ступенчатый сигнал амплитудой  $\pi$  рад (масштабированный угол)

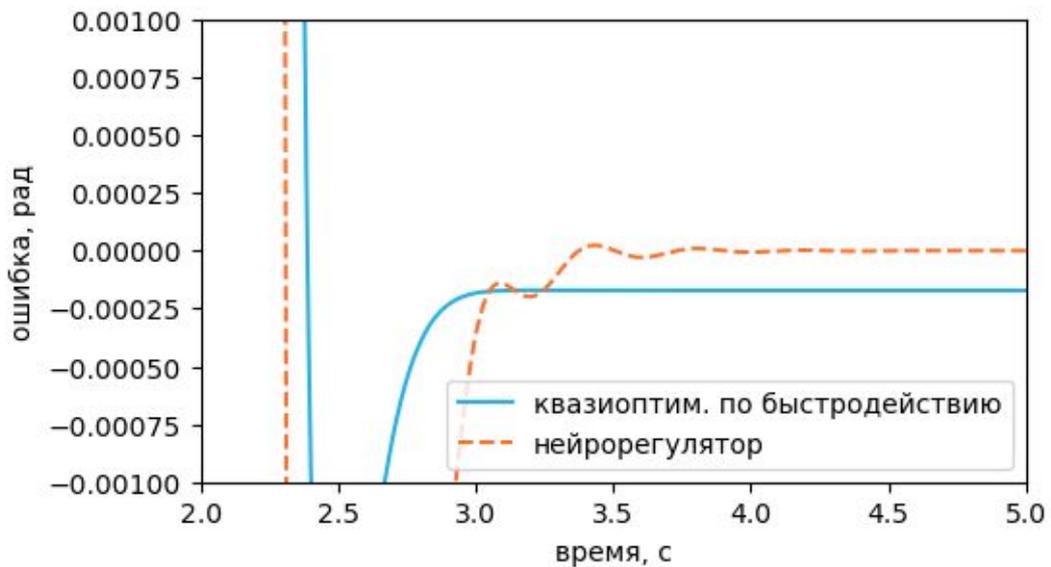


Рисунок 4.30 - Установившаяся ошибка слежения систем управления гидроприводом с квазиоптимальным по быстродействию и нейросетевым регуляторами на ступенчатый сигнал амплитудой  $\pi$  рад. (масштабированный угол)

Из рисунка 4.23 видно, что система с нейросетевым регулятором имеет точность слежения за входным сигналом хуже, чем система с квазиоптимальным регулятором (ошибка системы с нейросетевым регулятором 0.018 рад., а у системы с квазиоптимальным регулятором 0.0025 рад.). При этом из рисунка 4.25 видно, что при обработке статического сигнала у квазиоптимального по точности регулятора имеется высокая колебательность и затянутый переходный процесс.

Из рисунка 4.27 видно, что системы с квазиоптимальным регулятором по быстродействию и нейрорегулятором имеют близкую ошибку слежения за входным сигналом управления (0.016 рад. у системы с квазиоптимальным регулятором по быстродействию и 0.018 рад. у системы с нейрорегулятором). Скорость переходного процесса у системы с квазиоптимальным по быстродействию регулятором немного выше, чем у системы с нейрорегулятором (2.6 с. против 2.8 с), что видно из рисунка 4.29.

#### **Адаптация нейросетевой системы управления гидроприводом.**

В данном методе предусмотрена возможность выполнять дополнительное обучение нейрорегулятора, например, при ухудшении параметров регулирования, которые часто возникают из-за изменения характеристик объекта управления. В данном примере для наглядности был увеличен на 10% момент инерции нагрузки. Для сравнения рассматривается система с квазиоптимальным регулятором для измененного объекта управления. Пример адаптации нейросетевой системы управления гидроприводом представлен на рисунке 4.31.

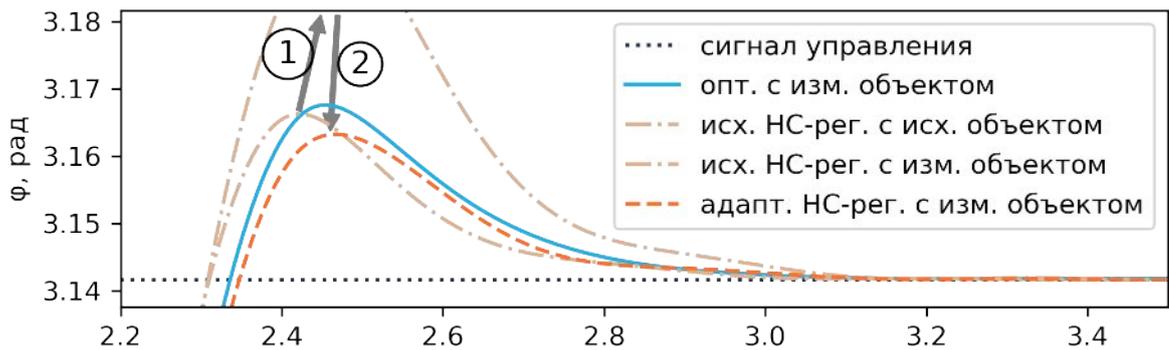


Рисунок 4.31 - Реакция системы с измененным ОУ на ступенчатый сигнал: 1 - ухудшение качества переходного процесса при увеличении момента нагрузки ОУ, 2 - нейрорегулятор адаптировался под измененный ОУ

Из результатов моделирования видно, что система с нейросетевым регулятором имеет свойства адаптивности и подстраивается под изменение параметров нагрузки, что существенно улучшает качество регулирования. Система с квазиоптимальным по быстродействию регулятором таких свойств не имеет, а значит чувствительна к изменению параметров объекта.

#### 4.1. Выводы

Разработана методика синтеза устойчивой системы в заданной локальной области с нейросетевым регулятором для объектов с ограничителями. Она объединяет разработанные алгоритмы и методы и позволяет оптимизировать качество переходного процесса по заданным критериям, например, по быстродействию. На примере гидропривода, как объекта управления, выполнено сравнение синтезированных систем управления с нейросетевым регулятором, LQR, квазиоптимальными по точности и быстродействию. В результате получается, что нейросетевая система управления близка к системе с квазиоптимальным по быстродействию регулятором и позволяет адаптироваться к однократно или медленно изменяющимся параметрам объекта управления.

## Заключение

1. Выполнен анализ существующих типов искусственных нейронных сетей и известных методов синтеза нейросетевых регуляторов, выбраны перспективные направления для работы с ограничителями. Для синтеза нейросетевых регуляторов, управляющих объектами с ограничителями, были выбраны нейронные сети прямого распространения и рекуррентные нейронные сети. В качестве методов синтеза были выбраны метод инверсного нейроуправления и метод пропуска ошибки через нейроимитатор. В качестве функции активации были выбраны ReLU и leakyReLU.
2. Разработан алгоритм формирования структуры нейрорегулятора и нейроимитатора для объекта управления с ограничителями, позволяющий определить тип искусственной нейронной сети, минимальное количество слоев и нейронов в них. Для нейрорегулятора по данному методу можно определить тип нейронной сети, используемые функции активации в скрытых слоях, размер входного и выходного слоев. Для скрытых слоев следует пользоваться правилом поэтапного увеличения нейронов и количества слоев до достижения результата, соответствующего заданным требованиям.
3. Разработан алгоритм формирования обучающей выборки с учетом наличия в объекте управления ограничителей.
4. Разработаны два метода синтеза нейрорегулятора для управления объектом с ограничителями, позволяющие определить основные параметры используемых нейронных сетей и алгоритмы обучения. Первый основан на схеме обучения «инверсное нейроуправление», второй на схеме «обратный пропуск ошибки через нейроимитатор».

5. Разработан метод синтеза нейрорегулятора и нейросетевой функции Ляпунова, позволяющей гарантировать устойчивость замкнутой системы в заданной локальной области фазового пространства.
6. Разработана методика синтеза устойчивой системы с нейросетевым регулятором для объектов с ограничителями. Она объединяет разработанные алгоритмы и методы и позволяет оптимизировать процесс регулирования.

*Автор выражает благодарность коллективу кафедры «Системы автоматического управления», за поддержку и внимание, оказанное моей работе.*

*Особую благодарность автор выражает своему научному руководителю доктору технических наук, доценту Сергею Владимировичу Феофилу за постановку задачи и внимательное руководство.*

**Список сокращений и условных обозначений**

ИНС — искусственная нейронная сеть;

НС — нейронная сеть;

ОУ — объект управления;

ПИД-регулятор — пропорционально-интегрально-дифференцирующий регулятор;

САУ — системы автоматического управления;

LQR — linear quadratic regulator;

ReLU - rectified linear unit.

## Список литературы

1. Бураков, М.В. Нейронные сети и нейроконтроллеры: учебное пособие. Нейронные сети и нейроконтроллеры / М.В. Бураков. – СПб.: ГУАП, 2013. – 282 с.
2. Галушкин, А.И. Синтез многослойных систем распознавания образов. - М.: Энергия, 1974. - 367 с. / А.И. Галушкин. – М.: Энергия, 1974. – 367 с.
3. Интеллектуальные системы автоматического управления : Профессиональное управление / ред. И.М. Макаров, В.М. Лохин. – М.: ФИЗМАТЛИТ, 2001. – 575 с.
4. Линейные оптимальные системы управления [Электронный ресурс] / Х. Квакернаак [и др.]. – М.: Мир, 1977. – 650 с. – Текст.
5. Ляпунов, А.М. Общая задача об устойчивости движения [Электронный ресурс] : Классики естествознания. Математика. Механика. Физика. Астрономия / А.М. Ляпунов. – М.: Гос. изд-во техн.-теорет. лит, 1950. – 472 с. – Текст.
6. Рутковская, Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский. – М.: Горячая линия - Телеком, 2008. – 383 с.
7. Руш, Н. Прямой метод Ляпунова в теории устойчивости / Н. Руш. – М: Мир, 1980. – 300 с.
8. Таха, Х.А. Введение в исследование операций / Х.А. Таха. – 7-е изд. – М. [и др.]: Вильямс, 2005. – 901 с.
9. Теория управления (дополнительные главы): Учебное пособие / Б.Р. Андриевский [и др.]; ред. Д.А. Новикова. – М.: ЛЕНАНД, 2019. – 552 с.
10. Терехов, В.А. Нейросетевые системы управления: Учеб. пособие для студентов вузов, обучающихся по специальности «Упр. и информатика в техн.

системах» и направлению «Автоматизация и упр.» / В.А. Терехов, Д.В. Ефимов, И.Ю. Тюкин. – М: Высш. шк, 2002. – 182 с.

11. Феофилов, С.В. Анализ методов обучения искусственной нейронной сети / С.В. Феофилов, Д.Л. Хапкин // Сборник конференции XVIII Всероссийской научно-технической конференции студентов, магистрантов, аспирантов и молодых ученых «Техника XXI века глазами молодых ученых и специалистов». – Тула: изд. ТулГУ, 2020. – С. 88-96.

12. Феофилов, С.В. Базовый синтез нейросетевых регуляторов для нелинейных следящих систем управления / С.В. Феофилов, Д.Л. Хапкин // ИНФОРМАТИКА: ПРОБЛЕМЫ, МЕТОДЫ, ТЕХНОЛОГИИ, Материалы XXI Международной научно-методической конференции. – ООО «Вэлборн», 2021. – С. 1314-1322.

13. Феофилов, С.В. Базовый синтез нейросетевых регуляторов для следящих систем управления / С.В. Феофилов, Д.Л. Хапкин // Известия ТулГУ. Технические Науки. – 2020. – № 11. – С. 256-261.

14. Феофилов, С.В. Некоторые подходы к синтезу нейросетевых регуляторов для следящих систем управления / С.В. Феофилов, Д.Л. Хапкин // Проблемы совершенствования робототехнических и интеллектуальных систем летательных аппаратов: Сборник докладов XI Международной юбилейной научно-технической конференции. – М.: ООО «Эдитус», 2021. – С. 342-346.

15. Феофилов, С.В. Применение рекуррентных нейронных сетей в замкнутых следящих системах с нелинейными объектами / С.В. Феофилов, Д.Л. Хапкин // Известия ТулГУ. Технические Науки. – 2021. – № 11. – С. 72-80.

16. Хайкин, С. Нейронные сети: полный курс, 2-е издание / С. Хайкин : Пер. с англ. – М.: Издательский дом «Вильямс», 2006. – 1104 с.

17. Свидетельство о государственной регистрации программы для ЭВМ 2023611080 Российская Федерация. Программа синтеза устойчивого

нейрорегулятора для управления перевернутым маятником. / Д.Л. Хапкин; Правообладатель ФГБОУ ВО ТулГУ (RU). – 2022686750; заявл. 26.12.2022; опубл. 16.01.2023, Реестр программ для ЭВМ. – С. 1.

18. Свидетельство о государственной регистрации программы для ЭВМ 2023611079 Российская Федерация. Программа формирования модели MIP из нейронной сети с функциями активации leakyReLU. / Д.Л. Хапкин; Правообладатель ФГБОУ ВО ТулГУ (RU). – 2022686748; заявл. 26.12.2022; опубл. 16.01.2023, Реестр программ для ЭВМ. – С. 1.

19. Хапкин, Д.Л. Исследование устойчивости нейросетевого управления на основе смешанного линейного целочисленного программирования / Д.Л. Хапкин, С.В. Феофилов, А.В. Козырь // Управление большими системами: труды XVIII Всероссийской школы-конференции молодых ученых. – Челябинск: Издательский центр ЮУрГУ, 2022. – С. 384-390.

20. Хапкин, Д.Л. Синтез гарантированно устойчивых нейросетевых регуляторов с оптимизацией качества переходного процесса / Д.Л. Хапкин, С.В. Феофилов, А.В. Козырь // Известия ТулГУ. Технические науки. – 2022. – № 12. – С. 128-133.

21. Хапкин, Д.Л. Синтез нейросетевых регуляторов с гарантией устойчивости и качества переходного процесса / Д.Л. Хапкин, С.В. Феофилов, А.В. Козырь // Приборостроение-2022: материалы 15-й Международной научно-технической конференции. – Минск, Республика Беларусь, 2022. – С. 122-124.

22. Хоанг, Ч.К. Оптимизация объемных силовых следящих гидроприводов по быстродействию и по точности режима слежения: диссертация ... кандидата технических наук: 05.13.01 / Ч.К. Хоанг. – Тула, 2006. – 131 с.

23. Чернодуб, А.Н. Обзор методов нейроруления / А.Н. Чернодуб, Д.А. Дзюба // Проблемы программирования. – 2011. – № 2. – С. 79-94.

24. A Unified View of Piecewise Linear Neural Network Verification / R.R. Bunel [и др.] // Proceedings of Advances in Neural Information Processing Systems. – Curran Associates, Inc., 2018. – Т. 31.
25. Aggressive Online Control of a Quadrotor via Deep Network Representations of Optimality Principles / S. Li [и др.] // Proceedings of IEEE International Conference on Robotics and Automation (ICRA). – Paris, France: IEEE, 2020. – С. 6282-6287.
26. Anderson, B.D.O. Linear Optimal Control / B.D.O. Anderson, J.B. Moore, B.P. Molinari // Proceedings of IEEE Transactions on Systems, Man, and Cybernetics. – 1972. – Т. SMC-2. – № 4. – С. 559-559.
27. Attention is All you Need / A. Vaswani [и др.] // ArXiv. – 2017.
28. Aung, H. Face Detection in Real Time Live Video Using Yolo Algorithm Based on Vgg16 Convolutional Neural Network / H. Aung, A.V. Bobkov, N.L. Tun // Proceedings of International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM). – 2021. – С. 697-702.
29. Backpropagation Applied to Handwritten Zip Code Recognition / Y. LeCun [и др.] // Neural Computation. – 1989. – Т. 1. – № 4. – С. 541-551.
30. CNN-SLAM: Real-Time Dense Monocular SLAM with Learned Depth Prediction / K. Tateno [и др.] // Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2017. – CNN-SLAM. – С. 6565-6574.
31. Counter-example guided synthesis of neural network Lyapunov functions for piecewise linear systems / H. Dai [и др.] // Proceedings of 59th IEEE Conference on Decision and Control (CDC). – 2020.
32. Cybenko, G. Approximation by superpositions of a sigmoidal function / G. Cybenko // Mathematics of Control, Signals, and Systems. – 1989. – Т. 2. – № 4. – С. 303-314.

33. Deep Residual Learning for Image Recognition / К. Хе [и др.] // Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – Las Vegas, NV, USA: IEEE, 2016. – С. 770-778.
34. Differentiable Convex Optimization Layers / A. Agrawal [и др.] // Proceedings of 33rd Conference on Neural Information Processing Systems (NeurIPS 2019). – Vancouver, Canada, 2019.
35. Dropout: A Simple Way to Prevent Neural Networks from Overfitting / N. Srivastava [и др.] // Journal of Machine Learning Research. – 2014. – Т. 15. – Dropout. – С. 1929-1958.
36. Elman, J.L. Finding Structure in Time / J.L. Elman // Cognitive Science. – 1990. – Т. 14. – № 2. – С. 179-211.
37. Farfadi, S.S. Multi-view Face Detection Using Deep Convolutional Neural Networks / S.S. Farfadi, M.J. Saberian, L.-J. Li // Proceedings of the 5th ACM on International Conference on Multimedia Retrieval : ICMR '15. – New York, NY, USA: Association for Computing Machinery, 2015. – С. 643-650.
38. Feofilov, S.V. Application of Recurrent Neural Networks in Closed Loop Tracking Systems for Controlling Essentially Nonlinear Objects / S.V. Feofilov, D. Khapkin // Proceedings of 3rd International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA). – Lipetsk, Russian Federation: IEEE, 2021. – С. 467-472.
39. Feofilov, S.V. Synthesis of neural network controllers for objects with non-linearity of the constraint type / S.V. Feofilov, D.L. Khapkin // Journal of Physics: Conference Series. – 2021. – Т. 1958. – № 1. – С. 012014.
40. Glorot, X. Deep Sparse Rectifier Neural Networks / X. Glorot, A. Bordes, Y. Bengio // Proceedings of Journal of Machine Learning Research. – 2010. – Т. 15.

41. Gnecco, G. The weight-decay technique in learning from data: an optimization point of view / G. Gnecco, M. Sanguineti // Computational Management Science. – 2009. – Т. 6. – The weight-decay technique in learning from data. – № 1. – С. 53-79.
42. Going deeper with convolutions / C. Szegedy [и др.] // Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – Boston, MA, USA: IEEE, 2015. – С. 1-9.
43. Goodfellow, I. Deep learning : Adaptive computation and machine learning / I. Goodfellow, Y. Bengio, A. Courville. – Cambridge, Massachusetts: The MIT Press, 2016. – 775 с.
44. Gradient-based learning applied to document recognition / Y. Lecun [и др.] // Proceedings of the IEEE. – 1998. – Т. 86. – № 11. – С. 2278-2324.
45. Hochreiter, S. Long Short-Term Memory / S. Hochreiter, J. Schmidhuber // Neural Computation. – 1997. – Т. 9. – № 8. – С. 1735-1780.
46. Hodgkin, A.L. A quantitative description of membrane current and its application to conduction and excitation in nerve / A.L. Hodgkin, A.F. Huxley // The Journal of Physiology. – 1952. – Т. 117. – № 4. – С. 500-544.
47. Jabali, M.B.A. Uncertain polytopic LPV modelling of robot manipulators and trajectory tracking / M.B.A. Jabali, M.H. Kazemi // International Journal of Control, Automation and Systems. – 2017. – Т. 15. – № 2. – С. 883-891.
48. Khapkin, D.L. Study of neural network control stability based on mixed linear integer programming / D.L. Khapkin, S.V. Feofilov, A.V. Kozyr // Proceedings of 4th International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA). – Lipetsk: IEEE, 2022. – С. 326-329.
49. Kingma, D.P. Adam: A Method for Stochastic Optimization. Adam / D.P. Kingma, J. Ba arXiv:1412.6980. – arXiv, 2017.

50. Knox, W. Augmenting Reinforcement Learning with Human Feedback / W. Knox. – 2011.
51. Krizhevsky, A. ImageNet classification with deep convolutional neural networks / A. Krizhevsky, I. Sutskever, G.E. Hinton // Communications of the ACM. – 2017. – T. 60. – № 6. – С. 84-90.
52. Land, A.H. An Automatic Method of Solving Discrete Programming Problems / A.H. Land, A.G. Doig // Econometrica. – 1960. – Т. 28. – № 3. – С. 497-520.
53. Lang, K.J. A time-delay neural network architecture for isolated word recognition / K.J. Lang, A.H. Waibel, G.E. Hinton // Neural Networks. – 1990. – Т. 3. – № 1. – С. 23-43.
54. Language Models are Few-Shot Learners / Т. Brown [и др.] // Proceedings of Advances in Neural Information Processing Systems. – Curran Associates, Inc., 2020. – Т. 33. – С. 1877-1901.
55. Large scale model predictive control with neural networks and primal active sets / S.W. Chen [и др.] // Automatica. – 2022. – Т. 135. – С. 109947.
56. Learning Lyapunov Functions for Hybrid Systems / S. Chen [и др.] // Proceedings of 55th Annual Conference on Information Sciences and Systems (CISS). – Baltimore, MD, USA: IEEE, 2021. – С. 1-1.
57. Learning-based Balancing of Model-based and Feedback Control for Second-order Mechanical Systems / Т. Beckers [и др.] // Proceedings of IEEE 61st Conference on Decision and Control (CDC). – IEEE, 2022. – С. 4667-4673.
58. Lyapunov-stable neural-network control / Н. Dai [и др.] arXiv:2109.14152. – arXiv, 2021.
59. Maas, A.L. Rectifier nonlinearities improve neural network acoustic models / A.L. Maas, A.Y. Hannun, A.Y. Ng // Proc. icml. – Atlanta, Georgia, USA, 2013. – Т. 30. – С. 3.

60. McCulloch, W.S. A logical calculus of the ideas immanent in nervous activity / W.S. McCulloch, W. Pitts // *Bulletin of Mathematical Biology*. – 1990. – Т. 52. – № 1-2. – С. 99-115.
61. Mei, S. A mean field view of the landscape of two-layer neural networks / S. Mei, A. Montanari, P.-M. Nguyen // *Proceedings of the National Academy of Sciences*. – 2018. – Т. 115. – № 33. – С. E7665-E7671.
62. Nair, V. Rectified Linear Units Improve Restricted Boltzmann Machines / V. Nair, G.E. Hinton // *Proceedings of the 27th International Conference on International Conference on Machine Learning : ICML'10 / event-place: Haifa, Israel*. – Madison, WI, USA: Omnipress, 2010. – С. 807-814.
63. Nakamura-Zimmerer, T. QRnet: Optimal Regulator Design With LQR-Augmented Neural Networks / T. Nakamura-Zimmerer, Q. Gong, W. Kang // *IEEE Control Systems Letters*. – 2021. – Т. 5. – QRnet. – № 4. – С. 1303-1308.
64. On the Properties of Neural Machine Translation: Encoder--Decoder Approaches / К. Cho [и др.] // *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. – 2014. – On the Properties of Neural Machine Translation.
65. Parrilo, P.A. Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization / P.A. Parrilo Google-Books-ID: hZxAQwAACAAJ. – California Institute of Technology, 2000. – 234 с.
66. Phoneme recognition using time-delay neural networks / A. Waibel [и др.] // *IEEE Transactions on Acoustics, Speech, and Signal Processing*. – 1989. – Т. 37. – № 3. – С. 328-339.
67. Patent US9750399B2 United States. Real-time depth estimation from monocular endoscope images / A. Popovic. – US13266465; заявл. 25.03.2010; опубл. 05.09.2017.

68. Practical Mathematical Optimization : Applied Optimization. T. 97. – New York: Springer-Verlag, 2005.
69. Probabilistic Forecasting with Temporal Convolutional Neural Network / Y. Chen [и др.] arXiv:1906.04397. – arXiv, 2020.
70. Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks / G. Katz [и др.] // Proceedings of Computer Aided Verification : Lecture Notes in Computer Science / ред. R. Majumdar, V. Kunčak. – Cham: Springer International Publishing, 2017. – Reluplex. – С. 97-117.
71. Riedmiller, M. A direct adaptive method for faster backpropagation learning: the RPROP algorithm / M. Riedmiller, H. Braun // Proceedings of IEEE International Conference on Neural Networks. – 1993. – С. 586-591 т.1.
72. Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. / F. Rosenblatt // Psychological Review. – 1958. – Т. 65. – № 6. – С. 386-408.
73. Rumelhart, D.E. Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations. Parallel Distributed Processing / D.E. Rumelhart, J.L. McClelland, AU. – The MIT Press, 1986.
74. Simonyan, K. Very Deep Convolutional Networks for Large-Scale Image Recognition / K. Simonyan, A. Zisserman arXiv:1409.1556. – arXiv, 2015.
75. Slotine, J.-J.E. Applied nonlinear control / J.-J.E. Slotine, W. Li. – Englewood Cliffs, N.J: Prentice Hall, 1991. – 459 с.
76. Solovyeva, E. Recurrent neural networks as approximators of non-linear filters operators / E. Solovyeva // Journal of Physics: Conference Series. – 2018. – Т. 1141. – № 1. – С. 012115.
77. Stanley, K.O. A Hypercube-Based Encoding for Evolving Large-Scale Neural Networks / K.O. Stanley, D.B. D'Ambrosio, J. Gauci // Artificial Life. – 2009. – Т. 15. –

№ 2. – С. 185-212.

78. Stanley, K.O. Evolving Neural Networks through Augmenting Topologies / K.O. Stanley, R. Miikkulainen // *Evolutionary Computation*. – 2002. – Т. 10. – № 2. – С. 99-127.
79. Sutton, R.S. Reinforcement Learning, second edition: An Introduction. Reinforcement Learning, second edition / R.S. Sutton, A.G. Barto. – MIT Press, 2018. – 549 с.
80. Temporal Convolutional Networks: A Unified Approach to Action Segmentation / C. Lea [и др.] // *Computer Vision – ECCV 2016 Workshops : Lecture Notes in Computer Science* / ред. G. Hua, H. Jégou. – Cham: Springer International Publishing, 2016. – Т. 9915. – Temporal Convolutional Networks. – С. 47-54.
81. The Limits and Potentials of Deep Learning for Robotics / N. Sünderhauf [и др.] arXiv:1804.06557. – arXiv, 2018.
82. Tjeng, V. Evaluating Robustness of Neural Networks with Mixed Integer Programming / V. Tjeng, K. Xiao, R. Tedrake arXiv:1711.07356. – arXiv, 2019.
83. Verification of Binarized Neural Networks via Inter-neuron Factoring / C.-H. Cheng [и др.] // *Proceedings of Verified Software. Theories, Tools, and Experiments : Lecture Notes in Computer Science* / ред. R. Piskac, P. Rümmer. – Cham: Springer International Publishing, 2018. – С. 279-290.
84. Vigerske, S. SCIP: global optimization of mixed-integer nonlinear programs in a branch-and-cut framework / S. Vigerske, A.M. Gleixner // *Optimization Methods and Software*. – 2018. – Т. 33. – С. 563-593.
85. Wang, H. CraterIDNet: An End-to-End Fully Convolutional Neural Network for Crater Detection and Identification in Remotely Sensed Planetary Images / H. Wang, J. Jiang, G. Zhang // *Remote Sensing*. – 2018. – Т. 10. – CraterIDNet. – № 7. – С. 1067.

86. Werbos, P.J. Backpropagation through time: what it does and how to do it / P.J. Werbos // Proceedings of the IEEE. – 1990. – T. 78. – Backpropagation through time. – № 10. – C. 1550-1560.
87. Werbos, P.J. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Beyond Regression / P.J. Werbos. – Harvard University, 1975. – 906 c.
88. Wong, E. Provable defenses against adversarial examples via the convex outer adversarial polytope / E. Wong, Z. Kolter // Proceedings of International Conference on Machine Learning. – PMLR, 2018. – C. 5286-5295.
89. Xiang, W. Output Reachable Set Estimation and Verification for Multi-Layer Neural Networks / W. Xiang, H.-D. Tran, T.T. Johnson arXiv:1708.03322. – arXiv, 2018.
90. Xie, Z. Understanding and Scheduling Weight Decay / Z. Xie, I. Sato, M. Sugiyama arXiv:2011.11152. – arXiv, 2021.

## Приложение А

РОССИЙСКАЯ ФЕДЕРАЦИЯ



## СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2023611079

**Программа формирования модели МР из нейронной  
сети с функциями активации leakyReLU**

Правообладатель: *Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Тулский государственный университет» (ТулГУ) (RU)*

Автор(ы): *Хапкин Дмитрий Леонидович (RU)*



Заявка № **2022686748**

Дата поступления **26 декабря 2022 г.**

Дата государственной регистрации

в Реестре программ для ЭВМ **16 января 2023 г.**

*Руководитель Федеральной службы  
по интеллектуальной собственности*

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ  
Сертификат 68b80077e14e40f0a94e1ba24145d5c7  
Владелец **Зубов Юрий Сергеевич**  
Действителен с 20.03.2022 по 26.05.2023

*Ю.С. Зубов*

## Приложение Б

РОССИЙСКАЯ ФЕДЕРАЦИЯ



## СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2023611080

**Программа синтеза устойчивого нейрорегулятора для  
управления перевернутым маятником**

Правообладатель: *Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Тульский государственный университет» (ТулГУ) (RU)*

Автор(ы): *Хапкин Дмитрий Леонидович (RU)*



Заявка № 2022686750

Дата поступления 26 декабря 2022 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 16 января 2023 г.

*Руководитель Федеральной службы  
по интеллектуальной собственности*

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ  
Сертификат 68b80077e14e40f0a94e6bd24145d5c7  
Владелец **Зубов Юрий Сергеевич**  
Действителен с 20.11.2022 по 26.05.2023

*Ю.С. Зубов*