



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ
ТЕХНОЛОГИИ

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
НА ТЕМУ:
Метод выделения продукционных правил изменения
состояния системы на основе статистических
данных о ее поведении в прошлом

Студент ИУ7-41М
(Группа)

(Подпись, дата)

О.В. Яковлева
(И.О.Фамилия)

Руководитель ВКР

(Подпись, дата)

И.В. Рудаков
(И.О.Фамилия)

Консультант

(Подпись, дата)

Ю.В. Строганов
(И.О.Фамилия)

Нормоконтролер

(Подпись, дата)

Ю.В. Строганов
(И.О.Фамилия)

Реферат

Расчетно-пояснительная записка содержит 75 с., 34 рис., 21 табл., 59 источников, 1 прил.

Метод выделения продукционных правил изменения состояния системы на основе статистических данных о ее поведении в прошлом.

Ключевые слова: прогнозирование, система, состояние системы, аккумулятор, нечеткая модель, субтрактивная кластеризация, слияние правил.

Объектом исследования работы являются методы автоматической генерации нечетких систем типа Такаги-Сугено по набору данных наблюдений.

Цель работы – разработка метода выделения нечетких продукционных правил изменения состояния аккумулятора.

Областью применения являются системы контроля и управления аккумуляторными батареями (battery management system, BMS).

В аналитической части проведен анализ существующих методов прогнозирования состояния системы на примере аккумуляторов и методов построения нечетких систем на основе данных наблюдений. В конструкторской части разработан метод прогнозирования состояний системы на основе нечетких моделей, предложен подход для устранения недостатка алгоритма субтрактивной кластеризации применительно к задаче построения нечёткой модели. В технологической части приведены аспекты реализации предлагаемого метода и аналогов. В исследовательской части проведено сравнение нечётких систем, построенных предлагаемым методом и аналогами, представлены результаты работы предлагаемого метода определения состояния и полученных нечётких моделей при различных параметрах субтрактивной кластеризации.

Поставленная цель была достигнута: разработан метод построения нечеткой модели и метод прогнозирования состояния аккумулятора на основе нечетких моделей.

Содержание

Реферат	2
Введение.....	5
1 Аналитический раздел	7
1.1 Описание предметной области.....	7
1.2 Понятие системы	9
1.3 Классификация систем.....	10
1.4 Аккумулятор как система	13
1.5 Методы и модели прогнозирования динамических систем	13
1.6 Постановка задачи	16
1.7 Основные понятия нечеткой логики.....	17
1.8 Методы автоматического построения нечеткой системы типа Такаги-Сугено на основе данных наблюдений.....	19
1.8.1 Алгоритмы структурной идентификации	19
1.8.1.1 Полный перебор.....	20
1.8.1.2 Кластеризация	21
1.8.1.3 Метаэвристические алгоритмы	25
1.8.2 Методы параметрической идентификации	26
1.9 Выводы.....	27
2 Конструкторский раздел.....	28
2.1 Постановка задачи	28
2.2 Методы прогнозирования состояния аккумулятора	29
2.2.1 Метод пространства состояний.....	29
2.2.2 СЯк-метод.....	31
2.3 Предлагаемый метод построения нечеткой системы типа Такаги-Сугено	33
2.3.1 Метод сокращения количества правил.....	36
2.3.2 Алгоритм сокращения количества правил в нечетких системах прогнозирования напряжения и состояния заряда.....	37
2.4 Структуры данных.....	41
2.5 Выводы.....	42
3 Технологический раздел.....	43
3.1 Выбор языка программирования	43
3.2 Описание набора данных	43
3.3 Структура разработанного программного обеспечения.....	43
3.3.1 Модуль загрузки данных	44

3.3.2 Модуль построения нечетких систем.....	45
3.3.3 Модуль тестирования и визуализации нечетких систем.....	53
3.3.4 Модуль прогнозирования состояния аккумулятора	54
3.4 Выводы	54
4 Исследовательский раздел	55
4.1 Сравнение результатов работы предлагаемого алгоритма построения нечеткой системы при разных параметрах субтрактивной кластеризации.....	55
4.2 Сравнение результатов работы предлагаемого метода построения нечеткой системы и аналогов	58
4.3 Сравнение предлагаемого метода построения нечеткой модели и его модификации, использующей метод роя частиц.....	64
4.4 Результаты прогнозирования состояния	65
4.5 Выводы.....	66
Заключение	67
Список использованных источников	68
Приложение А	72

Введение

В настоящее время существует проблема углеродного следа автомобилей, работающих на двигателях внутреннего сгорания. В качестве альтернативных, более экологичных транспортных средств рассматриваются автомобили с электрической силовой частью – электромобили, гибридные или электромобили на водородных топливных элементах. Во всех этих случаях в качестве одного из основных источников питания используются аккумуляторные батареи.

Аккумуляторные батареи необходимо использовать вместе с системой управления (battery management system, BMS), так как в противном случае может произойти необратимая деградация емкости, выход из строя и даже возгорание батареи из-за саморазогрева [1].

Система управления должна решать задачи прогнозирования и диагностики состояния аккумуляторных батарей. Под состоянием могут пониматься такие параметры, как напряжение на клеммах, температура, разрядная и зарядная емкости, степень заряда, степень деградации емкости.

Для решения этих задач требуется построить математические модели батареи и отдельного аккумулятора. В настоящее время чаще всего используются модели на основе схем замещения [1] и физические модели – уравнения, описывающие электрохимические процессы в батарее [2][3][4][5][6][7][8].

С увеличением точности модели предметной области растет и сложность её построения. Так, физические модели аккумулятора точнее, чем модели на основе схем замещения, однако требуют проведения большого количества сложных экспериментов для подбора коэффициентов уравнений.

Альтернативным подходом является нечеткое, нейросетевое и нейронечеткое моделирование. Например, в [9] предлагается нечеткая модель оценки разрядной емкости никель-метал-гидридного аккумулятора, в [10] – нейросетевая модель оценки состояния заряда литий-ионной батареи, в [11] –

нейронечёткая модель прогнозирования напряжения литий-ионного аккумулятора.

Целью данной работы является разработка метода выделения нечетких продукционных правил изменения состояния системы на примере аккумулятора. Для достижения поставленной цели необходимо выполнить следующие задачи:

1. Провести анализ существующих методов прогнозирования состояния системы на примере аккумуляторов, анализ существующих методов автоматического построения нечетких моделей на основе данных наблюдений.
2. Разработать метод выделения нечетких продукционных правил изменения состояния аккумулятора. Спроектировать систему, позволяющую реализовать данный метод.
3. Реализовать разработанный метод в программном обеспечении.
4. Провести сравнение точности нечетких систем, построенных предлагаемым методом и аналогами, определить зависимость точности от количества правил.

1 Аналитический раздел

1.1 Описание предметной области

Аккумулятор (от лат. *accumulator* – собиратель) – устройство для накопления энергии с целью ее последующего использования. Аккумулятор преобразует электрическую энергию в химическую и, по мере надобности, наоборот, т. е. может работать в одном из двух режимов: заряд или разряд. Заряд аккумулятора производится пропуском через него электрического тока, при этом вызванные химические реакции приводят к тому, что один из электродов получает положительный заряд, а другой отрицательный. При разряде электрический ток пропускается в направлении, обратном направлению при заряде, а накопленная химическая энергия используется для автономного электропитания различных электротехнических устройств и оборудования [12].

Существуют следующие типы промышленно выпускаемых аккумуляторов [12]:

- свинцово-кислотные (Pb);
- никель-кадмиевые (Ni-Cd);
- никель-железные (Ni-Fe);
- никель-металлгидридные (Ni-MH);
- никель-цинковые (Ni-Zn);
- никель-водородные (Ni-H₂);
- литий-ионные (Li-Ion);
- литий-полимерные (Li-Pol).

Li-Ion аккумуляторы являются наиболее перспективными на рынке перезаряжаемых химических источников тока, предназначенных для питания портативных электронных устройств. Главное их преимущество – высокая удельная энергия (до 150 Вт·ч/кг). Саморазряд Li-Ion ниже, чем у Ni-MH аккумуляторов. Несмотря на высокую цену литиевого источника тока, он постоянно совершенствуется и практически полностью вытеснил Ni-Cd и Ni-MH аккумуляторы [13][14][15].

Термины “аккумуляторная ячейка” и “аккумулятор” синонимичны и обозначают один гальванический элемент. Аккумуляторной батареей называется последовательное или параллельное соединение нескольких аккумуляторов. Силовые высоковольтные аккумуляторные батареи современных автомобилей строятся из последовательно подключаемых блоков по несколько ячеек, соединяемых в блоке параллельно.

Литий-ионные аккумуляторные батареи используются вместе с системой управления (battery management system, BMS), которая обеспечивает балансировку ячеек батареи, диагностику их состояния, контроль за изоляцией, следит за тепловым режимом, управляя системой охлаждения или, наоборот, обеспечивая подогрев батареи. Другие системы автомобиля также нуждаются в точной информации о состоянии батареи, например, чтобы использовать рекуперацию энергии при торможении или распределять нагрузку между разными источниками энергии в гибридных автомобилях. Необходимость использования системы управления батареями обусловлена ещё и тем, что при неправильной эксплуатации литий-ионного аккумулятора может произойти необратимая деградация емкости, выход из строя и даже возгорание из-за саморазогрева [1].

Система управления батареями должна решать задачи прогнозирования и диагностики состояния аккумуляторных ячеек. Под состоянием могут пониматься такие параметры, как напряжение на клеммах, температура, разрядная и зарядная емкости, степень заряженности, степень годности.

Степень заряженности формально выражается как $SOC = Q/Q_{max}$, где $Q = \int_0^t I(t)$ – текущий заряд аккумулятора (кулоновский способ оценки), Q_{max} – емкость аккумулятора, определяемая как максимальное количество электрической энергии (в Ач), которое аккумулятор отдает в нагрузку с момента полного заряда до состояния разряда, не приводящего к его преждевременной деградации [1].

По мере эксплуатации аккумулятора его емкость уменьшается. Степень годности (state of health, SOH) – качественный показатель, характеризующий

степень деградации емкости аккумулятора. Потеря емкости является следствием эффекта памяти. Эффект памяти заключается в приспособлении аккумулятора к рабочему циклу, по которому он работал определенный период времени. Если часто заряжать аккумулятор, но при этом он не полностью разряжен, аккумулятор «запоминает» неразряженное состояние и в последующие разы не может разрядиться полностью и поэтому емкость уменьшается. Чем чаще проводить цикл заряда и разряда, тем эффект памяти выражается четче.

Результатом оценки SOH является не численное значение, а ответ на вопрос: «необходимо ли заменить батарею в данный момент?». В настоящее время нет стандарта, регламентирующего на основе каких параметров аккумулятора должен вычисляться SOH. Для этого используются различные показатели, например, разность между исходной и действительной емкостями батареи Q_{max} или внутреннее сопротивление [1].

Для решения задач прогнозирования и диагностики состояния требуется построить модель аккумулятора как системы.

1.2 Понятие системы

В настоящее время существует несколько десятков определений понятия «система». Например, «... совокупность элементов, находящихся в определенных отношениях друг с другом и со средой» [16], «... упорядоченное определенным образом множество взаимосвязанных элементов, образующих некоторое целостное единство» [17], «... полный целостный набор элементов, взаимосвязанных между собой так, чтобы могла реализоваться функция системы» [18].

В общей теории систем и в математике типичным является определение системы в теоретико-множественных терминах. Элементы системы задаются с помощью некоторых множеств, а саму систему определяют как отношение на этих множествах [19]:

$$S \subset \times \{V_i, i \in I\},$$

где I – множество индексов. Множество V_i называется объектом системы.

Для определения системы “вход-выход” (в кибернетической литературе её называют иногда “черным ящиком”) используют два множества: U – множество входных воздействий на систему со стороны внешней для него среды (входной объект) и Y – множество выходных величин системы (выходной объект). Отношение S может быть задано отображением $U \rightarrow Y$ (в таком случае система называется функциональной [19]).

Если каждый элемент входного и выходного объектов системы является значением функции времени, система называется временной. При этом вводится вспомогательное множество моментов времени T – упорядоченное подмножество множества вещественных чисел. Значения функций из U и Y в момент времени $t \in T$ будут обозначаться как $u(t)$ и $y(t)$ соответственно. Более формально понятия времени и временных систем вводятся в [19].

1.3 Классификация систем

Принципы и подходы системных исследований должны учитывать тип, класс, особенности систем, поэтому важным является вопрос классификации. Далее приводится классификация систем по таким признакам, которые непосредственно отражаются на типах используемых для них математических моделей.

Если для каждого $t \in T$ значение выходной величины $y(t) \in Y$ можно определить однозначно по значению входного воздействия $u(t) \in U$ в тот же момент времени, система называется *статической*. Особенностью *динамических* систем является то, что для определения $y(t) \in Y$ недостаточно информации о $u(t) \in U$ в тот же момент времени.

При моделировании динамических систем одним из способов связать единственным образом значение $y(t) \in Y$ с каждым значением $u(t) \in U$ состоит в приписывании каждой паре вход-выход $(u(t), y(t))$ некоторого значения $x(t)$, называемого состоянием системы, такого, что $y(t)$ в момент времени t единственным образом определяется заданием $u(t)$ и $x(t)$ в тот же момент времени. Более формально понятие состояния вводится в [10]. Этот способ называют методом пространства состояний. Состояние системы нужно задавать

так, чтобы в данный момент времени оно содержало всю память о прошлом системы, существенную для её развития в будущем. В случае систем, описываемых дифференциальными уравнениями, компонентами вектора состояния могут являться производные различных порядков входного воздействия, взятые в момент времени t [20].

Согласно определению по Калману [21], динамическая система задается с использованием шести множеств и двух отображений:

- множества моментов времени T ;
- множества состояний X ;
- множества мгновенных значений входных воздействий U ;
- множества допустимых входных воздействий $\Omega = \{\omega: T \rightarrow U\}$,

причем класс функций на множестве Ω должен быть таким, чтобы две различные функции ω_1 и ω_2 на соседних полуинтервалах времени $(t_1, t_2]$ и $(t_2, t_3]$ могли быть заменены одной функцией ω_3 на суммарном полуинтервале $(t_1, t_3]$;

- множества мгновенных значений выходных величин Y ;
- множества выходных величин $\Gamma = \{\gamma: T \rightarrow Y\}$;
- переходной функции состояний $\varphi: T \times T \times X \times \Omega \rightarrow X$,

определяющей состояние $x(t) = \varphi(t, \tau, x(\tau), \omega) \in X$ (для стационарных систем $x(t) = \varphi(x(\tau), \omega) \in X$), в котором оказывается система в момент времени $t \in T$, если в начальный момент времени $\tau \in T$ она была в начальном состоянии $x(\tau) \in X$ и если на полуинтервале времени $(\tau, t]$ на неё действовало входное воздействие $\omega = \omega \in \Omega$;

- выходного отображения $\eta: T \times X \rightarrow Y$, определяющего выходную величину $y(t) = \eta(t, x(t)) \in Y$ (для стационарных систем $y(t) = \eta(x(t)) \in Y$) для момента времени $t \in T$ и состояния $x(t) \in X$.

Выходное отображение также может быть представлено в виде $\psi: T \times T \times X \times U \rightarrow Y$, т.е. выходная величина в момент времени $t \in T$ определяется как $y(t) = \psi(t, \varphi(t, \tau, x(\tau), \omega)) \in Y$ (для стационарных систем $y(t) = \zeta(x(\tau), \omega) \in$

$Y, \zeta = \psi \circ \varphi$), где $x(\tau) \in X$ – состояние системы в начальный момент времени $\tau \in T$, $\omega = \omega \in \Omega$ – входное воздействие, действующее на систему на полуинтервале времени $(\tau, t]$. В этом случае выходную величину $y(t) \in Y$ также называют *реакцией* системы на входное воздействие [19][21].

Также формальное определение динамической системы в теоретико-множественных терминах дано в [19].

По степени предсказуемости состояний и/или выходов различают *детерминированные* и *вероятностные* динамические системы [22]. Если информации о текущем состоянии системы и входном воздействии достаточно для установления с необходимой степенью точности её состояния и выходной величины в последующий момент времени, то такая система называется детерминированной. Её поведение полностью предсказуемо на основе вышеуказанной информации.

Для вероятностной (случайной, стохастической) системы вышеуказанных знаний достаточно лишь для предсказания вероятности нахождения её в том или ином состоянии, либо для вероятностной оценки выходной величины системы в последующие моменты времени.

По изменению структуры системы во времени различают *стационарные* и *нестационарные* динамические системы. Система называется постоянной или стационарной, если её реакция на входное воздействие при условии, что система находилась в заданном состоянии, не зависит от того, в каком промежутке времени осуществлялся этот опыт. Другими словами, для таких систем основные соотношения (структура) не меняются во времени [21].

В противном случае систему называют нестационарной.

Динамическая система S называется системой с *непрерывным временем*, тогда и только тогда, когда T совпадает с множеством вещественных чисел, и называется системой с *дискретным временем*, тогда и только тогда, когда T есть множество целых чисел [21].

1.4 Аккумулятор как система

На системном уровне аккумулятор может быть описан в виде черного ящика с одним входным воздействием – током в цепи $I(t)$ и одним выходом – напряжением на клеммах $U(t)$ [1].

В соответствии с приведенными в предыдущем разделе классификациями аккумулятор может быть определен как динамическая детерминированная нестационарная система с непрерывным временем.

Аккумулятор можно отнести к динамическим системам, поскольку по значению входного воздействия нельзя однозначно определить значение выхода. Например, при заряде/разряде постоянным током напряжение на клеммах будет меняться не только по мере накопления/отдачи аккумулятором энергии или в зависимости от величины тока или температуры в данный момент времени, но ещё и от предыстории.

По мере эксплуатации аккумулятора его емкость уменьшается, поэтому аккумулятор можно классифицировать как нестационарную систему.

1.5 Методы и модели прогнозирования динамических систем

Задача прогнозирования для динамических систем состоит в определении выходной величины $y(t) \in Y$ и/или состояния системы $x(t) \in X$ в момент времени $t \in T$ в зависимости от состояния системы $x(\tau) \in X$ в момент времени $\tau \in T$ и входного воздействия $\omega \in \Omega$, действующего на систему на полуинтервале времени $(\tau, t]$.

На первом этапе классификации методы прогнозирования обычно делят на две группы: *интуитивные* и *формализованные* [23].

Интуитивные методы имеют дело с суждениями и оценками экспертов. На сегодняшний день они часто применяются в маркетинге, экономике, политике, так как система, поведение которой необходимо спрогнозировать, или очень сложна и не поддается математическому описанию, или очень проста и в таком описании не нуждается. Методы такого рода подробно описаны в [24].

Формализованные методы предполагают построение математической модели системы для прогнозирования. По возможности описания изменения состояния системы модели, как и сами системы, можно разделить на *статические* и *динамические*.

Статические модели не учитывают внутренние изменения в системе под влиянием входных воздействий. Эти модели могут быть использованы только для прогнозирования выходной величины по состоянию системы и входному воздействию.

Динамические модели позволяют описать эволюцию системы и соответственно могут быть использованы для полного прогнозирования поведения системы (как следующего состояния, так и выходной величины).

По виду зависимостей величин модели могут быть классифицированы на *функциональные* и *нефункциональные*. В функциональных моделях все зависимости задаются в виде отображений. К нефункциональным относятся нейросетевые модели, модели на базе классификационно-регрессионных деревьев, нечетко-логические модели и т.д.

К статическим функциональным моделям относятся регрессионные и авторегрессионные модели, модель экспоненциального сглаживания, модель по выборке максимального правдоподобия [25] и т.д. Например, существуют модели аккумулятора в виде функции регрессии, построенные на базе имеющихся экспериментальных результатов [26][27][28]. Недостатком данных моделей является их применимость только для тех режимов работы, из которых она получена. Тем не менее, данные модели являются основой для построения более сложных и более глобальных моделей.

Примером статической нефункциональной модели может быть нейронечеткая модель, прогнозирующая напряжение ячейки литий-ионной батареи [11]. Входным параметром этой модели является, в том числе, степень заряженности ячейки.

Примерами динамических функциональных моделей могут быть модели аккумуляторов на базе известных законов транспорта компонент (ионов,

электронов и нейтральных частиц) в различных средах [2][3][4][5][6][7][8]. Они описывают физические процессы в жидкой фазе межэлектродного пространства, в жидкой фазе внутри пористого электрода, в металлической матрице пористого электрода и на границах фаз. Эти модели аккумуляторов наиболее глобальные. Однако, практики, использующие их в конкретных электротехнических системах, указывают, как правило, на два недостатка [29]. Во-первых, они содержат очень много параметров, описывающих множество локальных процессов и явлений. На практике данные параметры для конкретного аккумулятора или невозможно, или очень трудно измерить. Во-вторых, данные модели очень громоздки и сложны для практического использования.

При построении динамических моделей может быть использован описанный в разделе 1.3 метод пространства состояний. Примером динамической функциональной модели в пространстве состояний может быть эмпирическая модель аккумулятора на основе схемы замещения (эквивалентной схемы). Практически хорошо зарекомендовавший вариант эквивалентной схемы (рисунок 1.1) для литий-ионного аккумулятора основан на последовательном соединении активного внутреннего сопротивления и двух RC-цепочек [1].

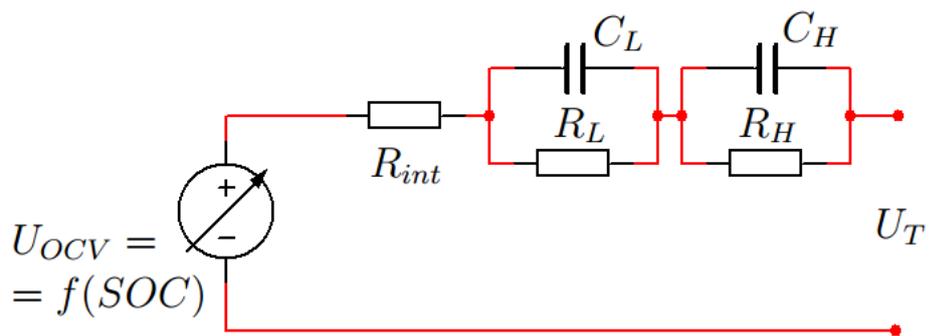


Рисунок 1.1. Схема замещения для динамической модели литий-ионной батареи

Представленная на рисунке 1.1 динамическая модель второго порядка в пространстве состояний записывается в виде [1]:

$$\dot{u}_L = \frac{I}{C_L} - \frac{u_L}{R_L C_L}$$

$$\dot{u}_H = \frac{I}{C_H} - \frac{u_H}{R_H C_H}$$

$$U_T = U_{OCV} - I * R_{int} - u_L - u_h$$

где $U_{OCV} = f(SOC)$ – напряжение холостого хода, параметры C_L, R_L, C_H, R_H подбираются на основе экспериментальных данных, снятых с конкретного типа аккумулятора, U_T – напряжение на клеммах, $R_{int} = f(SOC, T)$ – внутреннее активное сопротивление, T – температура аккумулятора.

Степень заряда аккумулятора можно рассматривать как еще одно состояние системы. В этом случае данная модель дополняется еще одним уравнением для переменной SOC [1].

В [25] предлагается классификация математических моделей прогнозирования по способу их получения, согласно которой выделяется две их группы: *модели предметной области* и *модели временных рядов*.

Модели предметной области – такие математические модели прогнозирования, для построения которых используют законы предметной области. К ним относятся рассмотренные выше модели аккумуляторов, описывающие законы транспорта компонент в различных средах [2][3][4][5][6][7][8], и модели аккумуляторов на основе схем замещения [1].

Для построения моделей временных рядов используются данные наблюдений (выходные величины и, возможно, входные воздействия, если проводится эксперимент) за динамической системой. Это модели “черного ящика”: моделируются только входные и выходные связи системы со средой. Примерами могут быть уже упомянутые модели аккумуляторов в виде функции регрессии [26][27][28] и нейро-нечёткая модель, прогнозирующая напряжение ячейки литий-ионной батареи [11].

1.6 Постановка задачи

В данной работе ставится задача прогнозирования напряжения на клеммах U_1, U_2, \dots, U_N и состояния заряда q_1, q_2, \dots, q_N аккумулятора для моментов времени t_1, t_2, \dots, t_N при начальном состоянии заряда q_0 и воздействии токами $I_0, I_1, I_2, \dots, I_N$ в моменты времени $t_0, t_1, t_2, \dots, t_N$.

Для решения этой задачи предлагается использовать аппарат нечеткой логики. Нечеткие системы, как и нейронные сети, являются универсальными нелинейными аппроксиматорами данных, однако, в отличие от нейронных сетей, функционирование которых осуществляется по принципу «черного ящика», обеспечивают возможность интеграции априорных знаний в модель исследуемого процесса.

1.7 Основные понятия нечеткой логики

Нечетким множеством A на универсальном множестве X называется совокупность пар $(\mu_A(x), x)$, где $\mu_A(x)$ — степень принадлежности элемента $x \in X$ нечеткому множеству.

Степень принадлежности — это число из диапазона $[0, 1]$. Чем выше степень принадлежности, тем в большей мере элемент универсального множества соответствует свойствам данного нечеткого множества.

Функцией принадлежности называется функция, которая позволяет вычислить степень принадлежности произвольного элемента универсального множества нечеткому множеству. Следовательно, область значений функций принадлежности должна принадлежать диапазону $[0, 1]$. В большинстве случаев функция принадлежности — это монотонная непрерывная функция.

Лингвистической (нечеткой) переменной называется переменная, значениями которой могут быть слова или словосочетания некоторого естественного или искусственного языка.

Терм-множеством называется множество всех возможных значений, которые способна принимать лингвистическая переменная.

Термом называется любой элемент терм-множества. В теории нечетких множеств терм формализуется нечетким множеством с помощью функции принадлежности.

База правил — это совокупность нечетких правил вида: "если, то", определяющих взаимосвязь между входами и выходами исследуемой системы.

Нечетким логическим выводом называется получение четкого значения выходной переменной, соответствующей четким значениям входных переменных, с использованием нечеткой базы знаний и нечетких операций.

В настоящее время наиболее широко используются три типа систем нечеткого логического вывода: Такаги-Сугено [30], Мамдани [31] и синглтон. Они различаются структурой правой части нечетких правил. В системах типа Мамдани в правой части содержатся нечеткие множества:

$$R^\theta: \text{если } x_1 \text{ есть } X_1^\theta, x_2 \text{ есть } X_2^\theta, \dots, x_N \text{ есть } X_N^\theta, \text{ то } y \text{ есть } Y^\theta$$

В системах типа Такаги-Сугено – полиномы θ -й степени:

$$R^\theta: \text{если } x_1 \text{ есть } X_1^\theta, x_2 \text{ есть } X_2^\theta, \dots, x_N \text{ есть } X_N^\theta, \text{ то } y = f^\theta(x_1, \dots, x_N)$$

И в системах типа синглтон – константы:

$$R^\theta: \text{если } x_1 \text{ есть } X_1^\theta, x_2 \text{ есть } X_2^\theta, \dots, x_N \text{ есть } X_N^\theta, \text{ то } y \text{ есть } a^\theta$$

где $X_1^\theta, X_2^\theta, \dots, X_N^\theta$ – лингвистические значения входных переменных x_1, x_2, \dots, x_N в θ -ом правиле, $\theta = \overline{1, \Theta}$; y – выходная переменная; f^θ – полином N -й степени, порядок которого также определяет и порядок модели Такаги-Сугено. Так, в модели Такаги-Сугено первого порядка $f^\theta(x_1, \dots, x_N) = a_0^\theta + a_1^\theta x_1^\theta + \dots + a_N^\theta x_N^\theta$.

Выбор типа системы определяется решаемой задачей. Авторы [32] отмечают, что если решается задача интерполяции или аппроксимации, причем точность является основным критерием качества модели, то выбор должен быть сделан в пользу нечеткой системы типа Такаги-Сугено. Если решается задача извлечения знаний из данных (в виде лингвистических правил) или поиск ассоциативных связей во множестве данных, то необходимо использовать нечеткую систему типа Мамдани. Достоинство таких моделей заключается в их понятности и интерпретируемости. Система типа синглтон обычно используется для решения задач нечеткой классификации и кластеризации [33].

1.8 Методы автоматического построения нечеткой системы типа Такаги-Сугено на основе данных наблюдений

После того, как экспертами определены входные и выходные переменные моделируемой системы и типы функций принадлежности для каждой входной переменной, автоматическое построение нечеткой системы типа Такаги-Сугено можно разделить на следующие этапы:

1. Идентификация структуры – определение количества лингвистических термов для каждой входной переменной, количества и структуры правил.
2. Идентификация параметров функций принадлежности для термов входных переменных.
3. Идентификация параметров линейных уравнений в правых частях правил (параметров консеквентов).

Эти этапы не обязательно должны быть выполнены последовательно. В зависимости от применяемого метода построения нечеткой системы этапы 1 и 2, 2 и 3 или все три этапа могут быть объединены. Поэтому существующие методы предлагается классифицировать в зависимости от явно выделяемых этапов работы, при этом используемый на очередном этапе алгоритм может быть заменен на аналогичный. На рисунке 1.2 приведены выделенные классы методов и используемые на этапах методов алгоритмы.

Далее алгоритмы структурной и параметрической идентификации будут рассмотрены без привязки к методам, в которых они используются.

1.8.1 Алгоритмы структурной идентификации

К алгоритмам структурной идентификации нечетких систем относятся:

- полный перебор;
- алгоритмы кластеризации;
- метаэвристические алгоритмы (рой частиц, генетический алгоритм и т.д.).

1.8.1.1 Полный перебор

В алгоритме полного перебора antecedentes правил формируются как все возможные сочетания термов входных переменных, consequents генерируются уникальными для каждого правила. Данный алгоритм используется обязательно в сочетании с алгоритмами параметрической идентификации, которые будут рассмотрены в разделе 1.8.2. К недостаткам алгоритма относится экспоненциальный рост правил с ростом количества термов входных переменных и возможность появления “лишних” правил – таких, которые не будут срабатывать ни при одной возможной комбинации значений входных переменных.

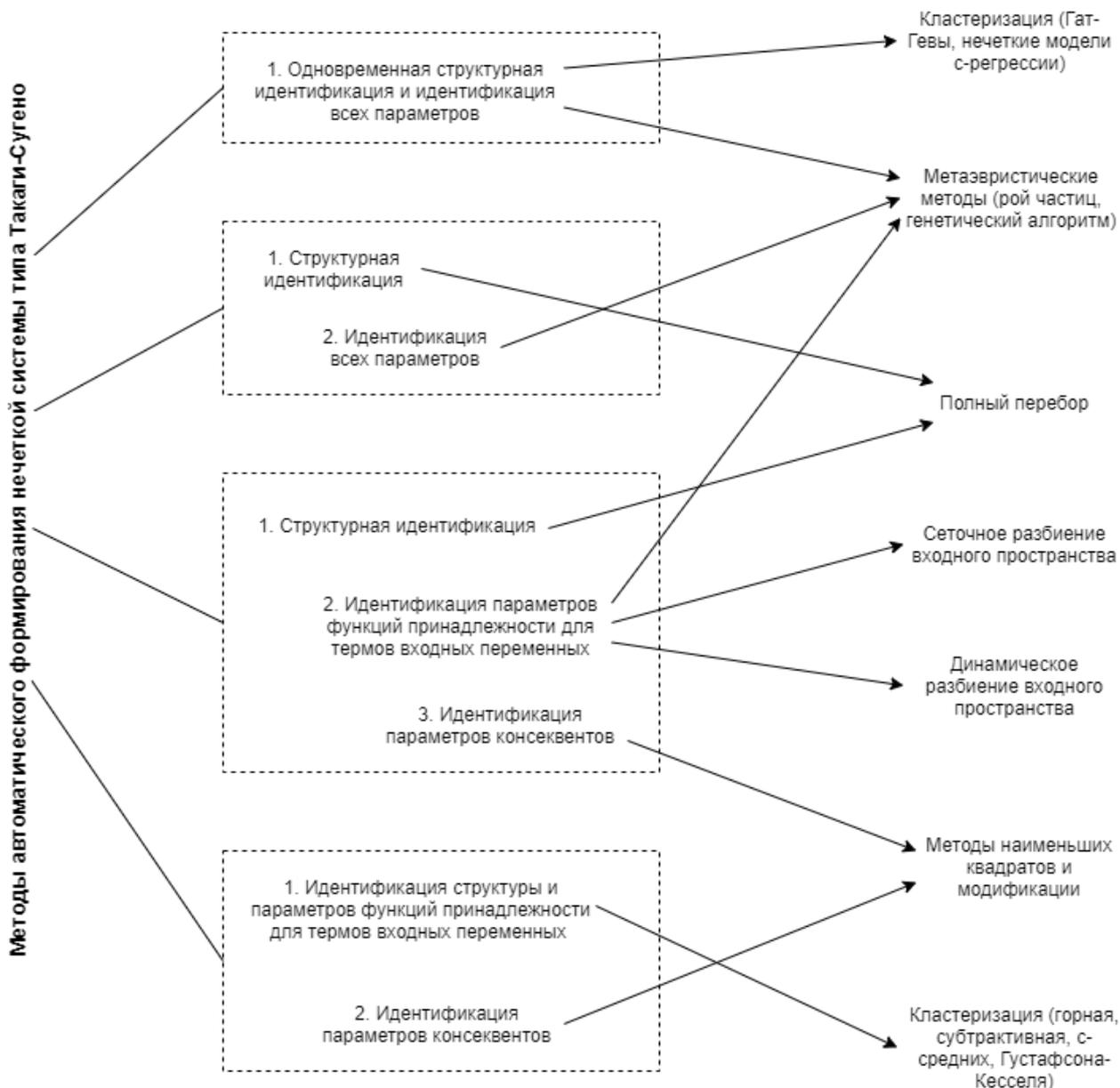


Рисунок 1.2. Методы автоматического формирования нечеткой системы типа Такаги-Сугено и используемые на различных этапах методов алгоритмы

1.8.1.2 Кластеризация

Использование алгоритмов кластеризации предполагает разбиение входного пространства (или объединения входного и выходного пространств) на кластеры и сопоставление каждому кластеру одного нечеткого правила.

Исходной информацией для кластеризации является матрица наблюдений:

$$X = \begin{matrix} x_{11} & x_{12} & \dots & x_{1N} \\ x_{21} & x_{22} & \dots & x_{2N} \\ \dots & \dots & \dots & \dots \\ x_{M1} & x_{M2} & \dots & x_{MN} \end{matrix}$$

Объекты из X , описываемые строками матрицы X , разбиваются на несколько подмножеств (кластеров), в которых объекты более схожи между собой, чем с объектами из других кластеров. В метрическом пространстве “схожесть” обычно определяют через расстояние. Расстояние может рассчитываться как между исходными объектами, так и от этих объектов к прототипу кластеров. Обычно координаты прототипов заранее неизвестны – они находятся одновременно с разбиением данных на кластеры.

Алгоритмы кластеризации можно классифицировать на четкие и нечеткие. Четкие алгоритмы разбивают исходное множество объектов X на несколько непересекающихся подмножеств. При этом любой объект из X принадлежит только одному кластеру. Нечеткие методы кластеризации позволяют одному и тому же объекту принадлежать одновременно нескольким (или даже всем) кластерам, но с различной степенью.

Нечеткие кластеры при заданном количестве кластеров c описываются матрицей нечеткого разбиения $U = [u_{ij}]$, $u_{ij} \in [0,1]$, $i = 1, \dots, M$, $j = 1, \dots, c$, где i -я строка содержит степени принадлежности объекта $(x_{i1}, x_{i2}, \dots, x_{iN})$ кластерам A_1, A_2, \dots, A_c . Условия для матрицы выглядят следующим образом:

$$\sum_{i=1}^M u_{ij} > 0 \text{ для } j = 1, \dots, c$$

$$\sum_{j=1}^c u_{ij} = 1 \text{ для } i = 1, \dots, M$$

Первое условие обеспечивает непустоту кластеров, второе – распределенность всех элементов из набора данных по всем кластерам, то есть сумма степеней принадлежности по всем кластерам должна быть 1 для каждого элемента данных.

Единственным отличием матрицы степеней принадлежности четкого разбиения от нечеткого является то, что элементы матрицы u_{ij} принимают значения из двухэлементного множества $\{0,1\}$, а не из интервала $[0,1]$.

Для идентификации нечетких систем наиболее часто используются алгоритмы нечеткой c -регрессии (fuzzy c -regression model, FCRM) [34], Густафсона-Кесселя [35], Гат-Гевы [36] и субтрактивной кластеризации [37][38]. Первые три алгоритма относятся к алгоритмам нечеткой кластеризации, четвертый – к четкой.

Алгоритмы *Густафсона-Кесселя* и *нечеткой c -регрессии* являются модификациями широко известного алгоритма нечетких c -средних [39]. Алгоритм нечетких c -средних формирует кластеры в форме гиперсфер, алгоритм Густафсона-Кесселя – в форме гиперэллипсов, а алгоритм нечеткой c -регрессии – в форме гиперплоскостей. Для работы алгоритмам требуется количество кластеров c и начальная матрица нечеткого разбиения. Их недостатком является зависимость результатов работы от выбора начальной матрицы разбиения – если она выбрана неудачно, алгоритмы “застревают” в локальных минимумах. Для решения этой проблемы в [40] предлагается задавать начальную матрицу с помощью алгоритма роя частиц, а в [41] – использовать преобразование Хафа [42][43].

Кроме того, может быть неизвестно количество кластеров. Для преодоления этой проблемы используется два подхода. Первый подход состоит в запуске алгоритма кластеризации с разными количествами кластеров и выборе наилучшего разбиения с помощью индексов достоверности [44]. Второй подход заключается в формировании большого количества кластеров и последующем слиянии совместимых кластеров [45].

Далее полученные нечеткие кластеры аппроксимируются многомерными или одномерными функциями принадлежности. Параметры одномерных функций могут быть получены проецированием кластеров на оси входных переменных [46]. При этом, очевидно, возникает два типа ошибок: ошибка аппроксимации и ошибка проецирования кластеров [47]. Многомерные функции принадлежности в данной работе рассматриваться не будут. Из параметров нечеткой c -регрессии также можно вычислить и параметры консеквентов правил [48].

Алгоритм *Гат-Гевы* позволяет получить параметры функций принадлежности и консеквентов правил непосредственно из параметров кластеризации, что позволяет избавиться от упомянутой выше ошибки проецирования кластеров. Однако для этого требуется линейное преобразование входных переменных, что усложняет интерпретацию полученных правил. Чтобы избавиться от этого недостатка, авторы [47] предлагают модифицированный алгоритм Гат-Гевы на основе модели гауссовой смеси, идентифицируемой алгоритмом максимизации ожидания.

Основная идея алгоритма *субтрактивной кластеризации* состоит в поиске в пространстве признаков областей с наибольшей плотностью точек данных. Параметрами алгоритма являются радиусы кластера в каждом измерении. На первом шаге для каждой точки из таблицы наблюдений рассчитываются значения потенциала, показывающие возможность формирования кластера в её окрестности. Чем плотнее расположены объекты в окрестности потенциального центра кластера, тем выше значение его потенциала. Далее центром первого кластера назначают точку с наибольшим потенциалом. Обычно наивысшая вершина окружена несколькими достаточно высокими пиками, поэтому назначение центром следующего кластера точки с максимальным потенциалом среди оставшихся вершин привело бы к выделению большого числа близко расположенных центров кластеров. Чтобы выбрать следующий центр кластера необходимо вначале исключить влияние только что найденного кластера. Для этого значения потенциала для оставшихся возможных центров кластеров пересчитывается: от текущих значений потенциала вычитают вклад центра только что найденного кластера (поэтому кластеризацию называют субтрактивной или вычитающей). Центр второго кластера определяется по максимальному значению обновленного потенциала. Итерационная процедура пересчета потенциалов и выделения центров кластеров продолжается до тех пор, пока максимальное значение потенциала превышает некоторый порог [49].

Достоинствами алгоритма субтрактивной кластеризации являются более высокая скорость работы относительно алгоритма с-средних и его модификаций, так как по таблице наблюдений совершается только один проход, и устойчивость к шумам и выбросам. Недостаток алгоритма состоит в том, что найденные центры кластеров (точки с наибольшим потенциалом) могут не совпадать с реальными центрами кластеров [50]. Кроме того, алгоритм выделяет кластеры одинакового размера, что не всегда корректно для реальных наборов данных.

В [37] предложен метод идентификации нечеткой системы на основе субтрактивной кластеризации.

1.8.1.3 Метаэвристические алгоритмы

При использовании метаэвристических алгоритмов (рой частиц, генетический алгоритм) построение нечеткой системы формулируется как задача оптимизации в многомерном пространстве. Каждая точка этого пространства соответствует одной нечеткой системе, т.е. описывается вектором, содержащим закодированную базу правил и параметры функций принадлежности и консеквентов. Целевой функцией может являться, например, среднеквадратичная ошибка аппроксимации системы [51].

К недостаткам метаэвристических алгоритмов исследователи относят высокие вычислительные затраты в сравнение с алгоритмами кластеризации [47]. Кроме этого, применение метаэвристик не гарантирует нахождения глобального минимума и, как правило, связано с эмпирической настройкой параметров используемых алгоритмов [52].

В таблице 1.1 приведено сравнение различных алгоритмов структурной идентификации.

Таблица 1.1. Алгоритмы структурной идентификации нечетких систем

Алгоритм	Одновременная идентификация параметров функций принадлежности входных переменных	Одновременная идентификация параметров консеквентов правил	Вычислительная сложность	Рост числа правил с ростом количества лингвистических термов
Полный перебор	-	-	Низкая	Экспоненциальный
Кластеризация (Густафсона-Кесселя, субтрактивная)	+	-	Средняя	Линейный
Кластеризация (Гата-Гевы, нечеткой с-регрессии)	+	+	Средняя	Линейный
Метаэвристические (рой частиц, генетический алгоритм)	+	+	Высокая	Зависит от данных

1.8.2 Методы параметрической идентификации

Для параметрической идентификации функций принадлежности используются следующие алгоритмы:

- сеточное разбиение входного пространства (СРВП) [53];
- динамическое разбиение входного пространства (ДРВП) [53].

В алгоритме СРВП для каждой переменной входного пространства генерируется заданное количество одинаковых термов по всему диапазону своего изменения.

Алгоритм ДРВП следующий: пока ошибка аппроксимации больше заданной, на каждой итерации определяется регион входного пространства с наибольшей ошибкой, в этом регионе выбирается входная переменная с наибольшей ошибкой. Для этой переменной генерируется новый терм и новая треугольная функция принадлежности, центр которой вычисляется по формуле,

а границами являются центры термов, между которыми будет помещён новый терм. В [54] описано использование метода ДРВП в сочетании с алгоритмом полного перебора для формирования структуры правил и с рекуррентным методом наименьших квадратов для идентификации параметров консеквентов.

Для параметрической идентификации консеквентов правил используются:

- метод наименьших квадратов [55];
- рекуррентный метод наименьших квадратов [56][54].

Также для инициализации параметров как функций принадлежности, так и консеквентов, широко применяются метаэвристические методы и ANFIS [57][58] – комбинированный метод, использующий метод градиентного спуска с обратным распространением ошибки для идентификации параметров antecedентов и метод наименьших квадратов для идентификации параметров консеквентов. При этом в качестве начальных значений параметров могут быть выбраны те, которые были получены на основе кластеризации [58][59] или алгоритмами СРВП и ДРВП [59].

1.9 Выводы

Приведено определение понятия системы, классификация систем. В соответствии с этой классификацией показано, что аккумулятор является динамической детерминированной нестационарной системой с непрерывным временем. Проведен анализ существующих методов прогнозирования состояния системы на примере аккумуляторов и методов построения нечетких систем на основе данных наблюдений.

2 Конструкторский раздел

2.1 Постановка задачи

Пусть I – множество значений тока нагрузки, U – множество значений напряжения на клеммах аккумулятора, Q – множество значений состояния заряда, T – множество значений моментов времени. При этом, если момент времени относится к прошлому, его значение отрицательно, если к будущему – положительно, если к настоящему – равно 0.

Пусть состоянием аккумулятора называется тройка (I_j, U_j, Q_j) , где $I_j \in I, U_j \in U, Q_j \in Q$ – значения тока нагрузки, напряжения на клеммах и состояния заряда в момент времени $t_j \in T$ соответственно, $j = \overline{N, M}, N < 0, M > 0$. Это понятие состояния несколько отличается от классического, введенного в разделе 1.3. Согласно классическому определению, выходная переменная (в данном случае напряжение) не должна являться компонентой состояния. Чтобы избежать путаницы, далее *внутренним состоянием* будет называться состояние в классическом смысле, а *полным состоянием* (или просто *состоянием*) описанная выше тройка (I_j, U_j, Q_j) .

Пусть дана таблица наблюдений Z , в строках которой содержатся значения компонент состояния в прошлые моменты времени $t_i < 0 \in T, i = \overline{N, -1}, N < 0$:

$$Z = \begin{matrix} I_N & U_N & Q_N \\ I_{N+1} & U_{N+1} & Q_{N+1} \\ \dots & \dots & \dots \\ I_{-1} & U_{-1} & Q_{-1} \end{matrix}$$

На основе таблицы наблюдений для заданного набора P , содержащего значения токов нагрузки в будущие моменты времени $t_k > 0 \in T, k = \overline{1, M}$,

$$P = \begin{matrix} I_1 \\ I_2 \\ \dots \\ I_M \end{matrix}$$

и состояния (I_0, U_0, Q_0) в настоящий момент времени $t_0 = 0 \in T$ требуется

определить состояния (I_k, U_k, Q_k) в будущие моменты времени $t_k > 0 \in T, k = \overline{1, M}$.

На рисунке 2.1. приведена постановка задачи в нотации IDEF0.



Рисунок 2.1. Постановка задачи

2.2 Методы прогнозирования состояния аккумулятора

2.2.1 Метод пространства состояний

Принципы метода пространства состояний были описаны в разделе 1.3: требуется определить внутреннее состояние, переходное отображение (зависимость внутреннего состояния в следующий момент времени от входного воздействия и внутреннего состояния в текущий момент времени) и выходное отображение (зависимость выходной переменной от внутреннего состояния).

Напряжение на клеммах аккумулятора имеет нелинейную зависимость от тока нагрузки, температуры и состояния заряда. В данной работе считается, что температура постоянна для всех моментов времени $t_j \in T, j = \overline{N, M}, N < 0, M > 0$. Таким образом, внутреннее состояние определяется двумя компонентами: током нагрузки и состоянием заряда. Пусть x_j – значение внутреннего состояния в момент времени $t_j \in T: x_j = (I_j, Q_j) \in X$.

Выходное отображение $f: X \rightarrow U$ (или $f: I \times Q \rightarrow U$) осуществляется нечеткой системой (рисунок 2.2).

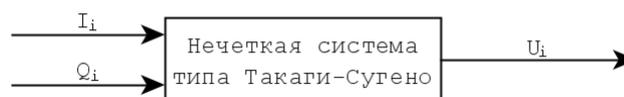


Рисунок 2.2. Входы и выходы нечеткой системы прогнозирования напряжения

Переходная функция определяется согласно кулоновскому способу оценки заряда, где значение интеграла рассчитывается методом правых прямоугольников: $Q_{j+1} = Q_j + I_{j+1} * dt$, где $dt = t_{j+1} - t_j = const$.

На рисунке 2.3 приведена декомпозиция постановки задачи для описанного выше метода, на рисунке 2.4 – алгоритм определения будущих состояний согласно методу пространства состояний.



Рисунок 2.3. Декомпозиция постановки задачи прогнозирования состояния при её решении методом пространства состояний

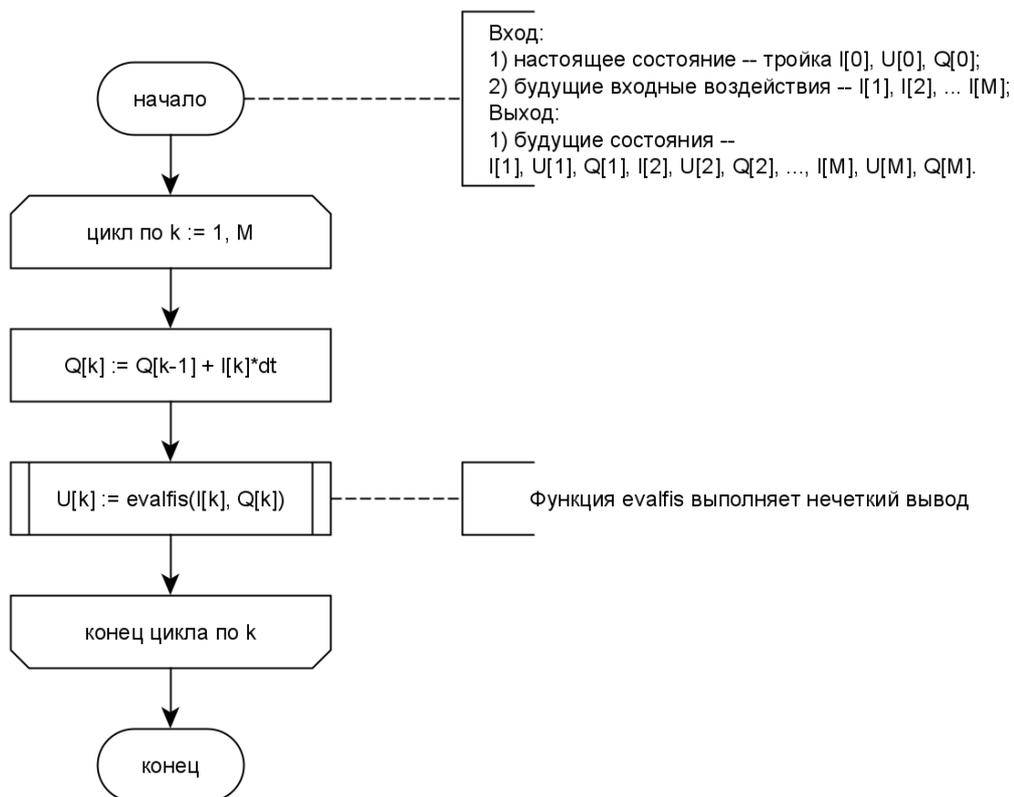


Рисунок 2.4. Алгоритм определения будущих состояний

2.2.2 СЯк-метод

Метод кулоновского счета для оценки состояния заряда состоит в вычислении интеграла тока нагрузки по времени. При использовании численных методов ошибка вычисления интеграла увеличивается при уменьшении количества узлов расчетной сетки (кроме частного случая, когда интегрируемая функция равна константе), поэтому метод кулоновского счета, гипотетически, требует довольно частого снятия значений тока с датчиков. Чтобы уменьшить частоту снятия данных, предлагается альтернативный метод прогнозирования состояния аккумулятора, в котором не используется кулоновский счет.

Согласно данному методу предлагается определять текущее напряжение на клеммах не по текущему, а по предыдущему состоянию заряда, а текущее состояние заряда диагностировать по текущим напряжению на клеммах и току нагрузки. Для этого нужно построить две нечеткие системы (рисунки 2.5 и 2.6).

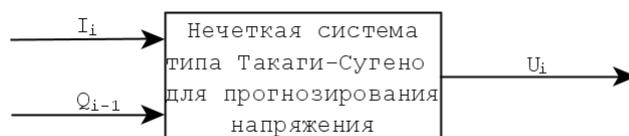


Рисунок 2.5. Входы и выходы нечеткой системы для прогнозирования напряжения

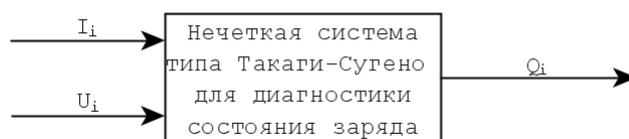


Рисунок 2.6. Входы и выходы нечеткой системы для диагностики состояния заряда

На рисунках 2.7 и 2.8 приведена декомпозиция постановки задачи и алгоритм определения будущих состояний согласно описанному выше методу.

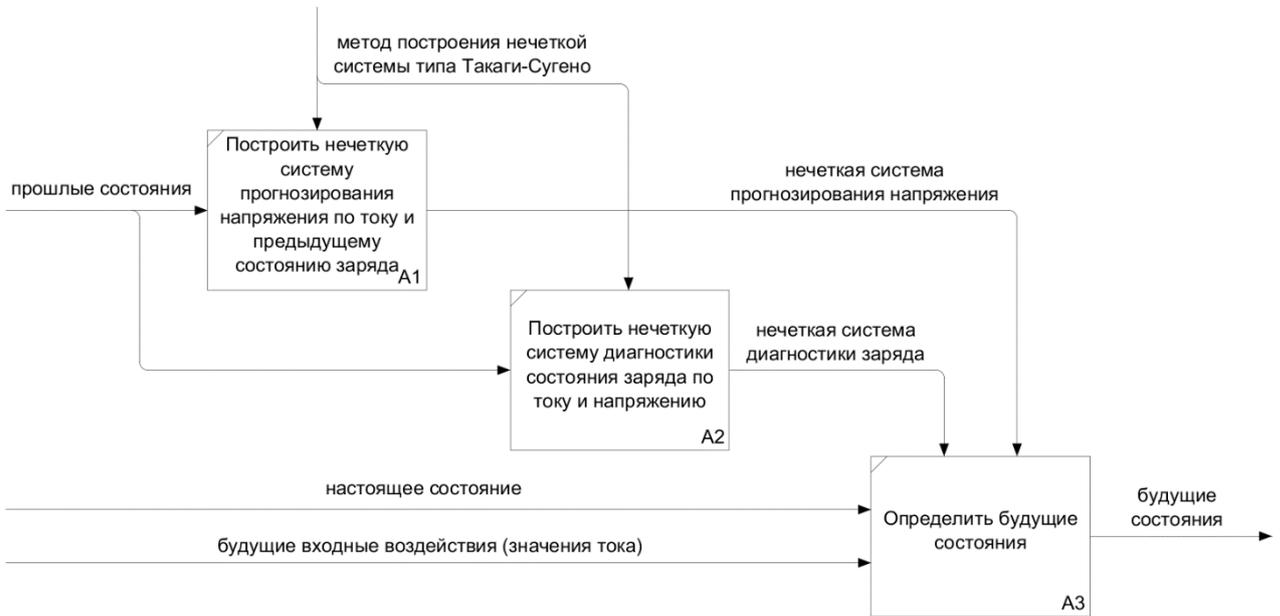


Рисунок 2.7. Декомпозиция постановки задачи прогнозирования состояния при её решении СЯк-методом

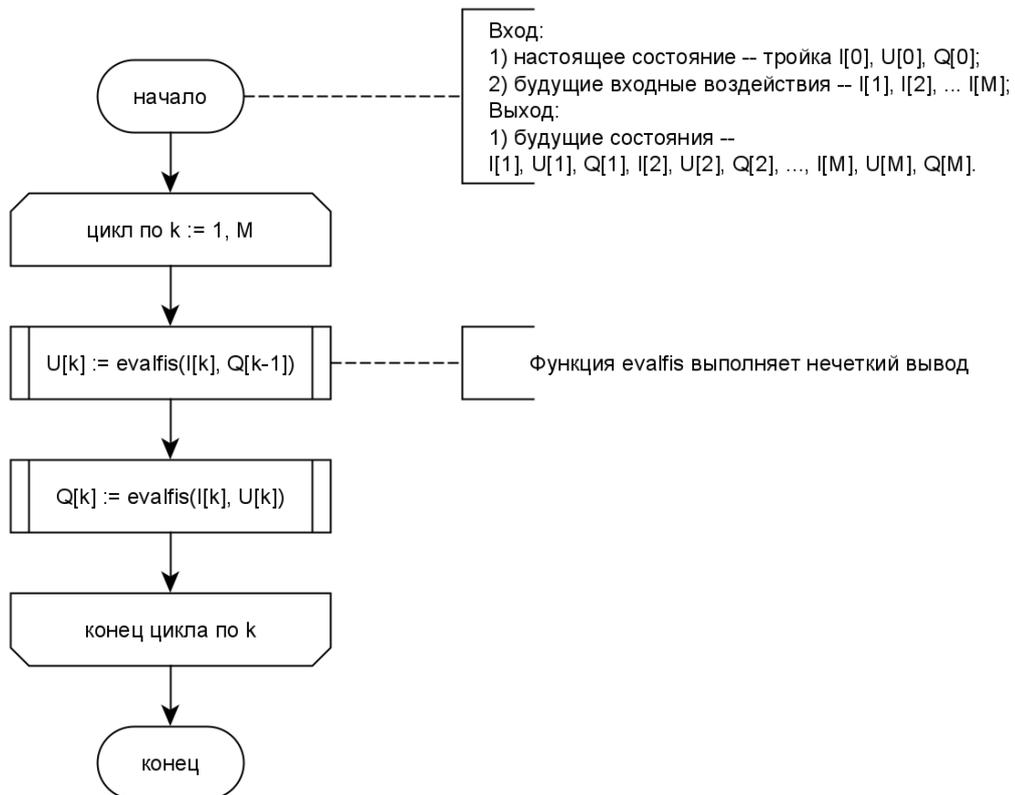


Рисунок 2.8. Алгоритм определения будущих состояний

В таблице 2.1 приведена сравнительная характеристика двух предлагаемых методов. Здесь *evalfis* – функция, выполняющая нечеткий вывод.

Таблица 2.1. Предлагаемые методы прогнозирования состояния

	Метод пространства состояний	СЯк-метод
Как определяется напряжение	$U_j = evalfis(Q_j, I_j)$	$U_j = evalfis(Q_{j-1}, I_j)$
Как определяется заряд	$Q_j = Q_{j-1} + I_j * dt$	$Q_j = evalfis(U_j, I_j)$
Частота снятия значений тока	часто	редко
Независимость нечеткой модели прогнозирования напряжения от dt	+	–

2.3 Предлагаемый метод построения нечеткой системы типа Такаги-Сугено

В качестве метода структурной инициализации выбрана субтрактивная кластеризация. Как было сказано в разделе 1.8.1, алгоритмы кластеризации лучше всего подходят для этой задачи, поскольку, в отличие от алгоритма полного перебора, дают линейный, а не экспоненциальный рост правил с ростом количества термов входных переменных, и имеют меньшую вычислительную сложность, чем метаэвристические алгоритмы.

Для настройки нечеткой системы нужны значения состояния аккумулятора при нескольких циклах заряда/разряда разными токами. При этом при заряде-разряде постоянным или ступенчатым током значения состояния заряда могут регистрироваться в диапазоне 0% до 100% с маленьким и почти равномерным шагом, а вот выборка различных значений тока будет разрежена. Алгоритм субтрактивной кластеризации позволяет задать априорную информацию о том, что диаметр кластеров по оси тока большой, поэтому из всех алгоритмов кластеризации выбран именно он. Остальные алгоритмы могут выделить для каждого значения тока в рамках каждого цикла отдельные кластеры, что приведет к тому, что нечеткая система будет хорошо

прогнозировать состояния аккумулятора для тех значений тока, на которых она обучалась, и плохо для промежуточных.

Недостаток алгоритма субтрактивной кластеризации – одинаковые радиусы для каждого кластера – предлагается устранять следующим образом: сначала получить большое количество маленьких кластеров (для этого нужно задать маленькие радиусы кластеров по осям состояния заряда и напряжения) и, следовательно, сформировать нечеткую систему с большим количеством правил, а затем объединить правила так, чтобы среднеквадратичная ошибка нечеткой системы сильно не увеличилась.

На рисунке 2.9 приведен предлагаемый метод в нотации IDEF0.

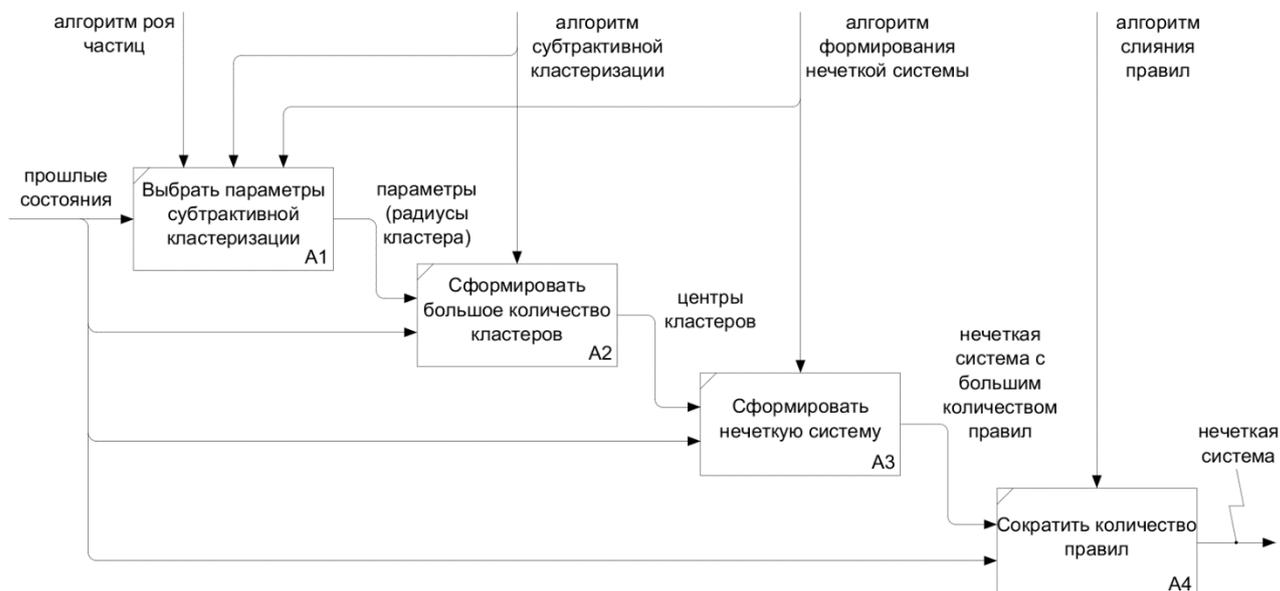


Рисунок 2.9. Предлагаемый метод построения нечеткой системы

Для выбора параметров алгоритма (радиусов кластера) предлагается использовать рой частиц, при этом аргументами целевой функции являются радиусы кластера, а значение рассчитывается следующим образом:

1. Алгоритмом субтрактивной кластеризации определяются количество C кластеров заданных радиусов и центры кластеров. На вход алгоритма подается таблица наблюдений Z :

$$Z = \begin{matrix} x_{11} & x_{12} & \dots & x_{1n} & y_1 \\ x_{21} & x_{22} & \dots & x_{2n} & y_2 \\ \dots & \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mn} & y_m \end{matrix}$$

где x_{ij} – значение i -й входной переменной, y_i – значение выходной переменной, $i = \overline{1, m}, j = \overline{1, n}, n$ – количество входных переменных.

2. Формируется нечеткая система согласно алгоритму из [35]:

2.1. Для каждой входной переменной формируется C функций принадлежности гауссова типа. Под формированием i -й функции принадлежности j -й входной переменной понимается идентификация её параметров: среднеквадратичного отклонения σ_j и математического ожидания $\mu_j^k, j = \overline{1, n}, k = \overline{1, C}$. В качестве параметра μ_j^k берется координата центра k -го кластера на оси, соответствующей j -й входной переменной, параметр σ_j рассчитывается по формуле (2.1) и принимает одинаковые значения для всех C функций принадлежности.

$$\sigma_j = \frac{r_j * (\max_j - \min_j)}{\sqrt{8}} \quad (2.1)$$

где r_j – радиус кластера в j -м измерении, \max_j и \min_j – максимальное и минимальное значение j -й входной переменной соответственно, $j = \overline{1, n}, n$ – количество входных переменных.

2.2. Генерируется C правил:

R^k : если x_1 есть X_1^k, x_2 есть X_2^k, \dots, x_n есть X_n^k , то $y = a_0^k + a_1^k x_1 + \dots + a_n^k x_n$

где X_j^k – терм j -й входной переменной в k -ом правиле, формализуемый гауссовой функцией принадлежности с параметрами μ_j^k и σ_j , идентифицированными на предыдущем шаге, $a_0^k = a_1^k = \dots = a_n^k = 0, k = \overline{1, C}$.

2.3. Методом наименьших квадратов рассчитываются параметры $a_0^k, a_1^k, \dots, a_n^k$ консеквентов правил, $k = \overline{1, C}$.

3. С помощью полученной нечеткой системы рассчитываются значения \bar{y}_i выходной переменной, $i = \overline{1, m}$, и по формуле (2.2) вычисляется среднеквадратичная ошибка прогнозирования модели (mean-squared error, MSE)

$$MSE = \frac{1}{m} * \sum_{i=1}^m (y_i - \bar{y}_i)^2 \quad (2.2)$$

где y_i и \bar{y}_i – истинное и спрогнозированное значение выходной переменной модели соответственно, $i = \overline{1, m}$.

После того, как получены оптимальные радиусы кластеров, снова выполняются п. 1 и п. 2, описанные выше, так как алгоритм роя частиц возвращает только значения параметров, при которых достигается минимум целевой функции, а полученная при вычислении значения целевой функции нечеткая система не сохраняется.

2.3.1 Метод сокращения количества правил

Предлагается итеративный метод сокращения количества правил. На каждой итерации из нечеткой системы S объединением каждого правила с каждым из его k ближайших соседей генерируются все возможные новые нечеткие системы, из которых выбирается наилучшая, согласно некоторой метрике качества, и принимается за S на следующей итерации. Таким образом, на каждой итерации количество правил сокращается на единицу. Под объединением правил

$$R^p: \text{если } x_1 \text{ есть } X_1^p, x_2 \text{ есть } X_2^p, \dots, x_n \text{ есть } X_n^p, \text{ то } y = a_0^p + a_1^p x_1 + \dots + a_n^p x_n$$

и

$$R^q: \text{если } x_1 \text{ есть } X_1^q, x_2 \text{ есть } X_2^q, \dots, x_n \text{ есть } X_n^q, \text{ то } y = a_0^q + a_1^q x_1 + \dots + a_n^q x_n$$

нечеткой системы S понимается такое правило

$$R^{pq}: \text{если } x_1 \text{ есть } X_1^{pq}, x_2 \text{ есть } X_2^{pq}, \dots, x_n \text{ есть } X_n^{pq}, \text{ то } y = a_0^{pq} + a_1^{pq} x_1 + \dots + a_n^{pq} x_n,$$

где X_j^{pq} есть новый терм j -й входной переменной, формализуемый гауссовой функцией принадлежности с параметрами μ_j^{pq} и σ_j^{pq} , полученный объединением термов X_j^p и X_j^q , формализуемых гауссовыми функциями принадлежности с параметрами μ_j^p, σ_j^p и μ_j^q, σ_j^q соответственно, причем $\mu_j^{pq} = f(\mu_j^p, \mu_j^q, \sigma_j^p, \sigma_j^q)$ и $\sigma_j^{pq} = \psi(\mu_j^p, \mu_j^q, \sigma_j^p, \sigma_j^q)$, а параметры $a_0^{pq}, a_1^{pq}, \dots, a_n^{pq}$ идентифицируются методом наименьших квадратов вместе с параметрами консеквентов остальных правил в нечеткой системе S .

Для работы метода требуется задать функции f и ψ , функцию расстояния между правилами $\zeta(R^p, R^q)$, количество k рассматриваемых ближайших правил и метрику качества нечеткой системы. На рисунке 2.10 представлен описанный метод в нотации IDEF0 (блок A4 рисунка 2.9).

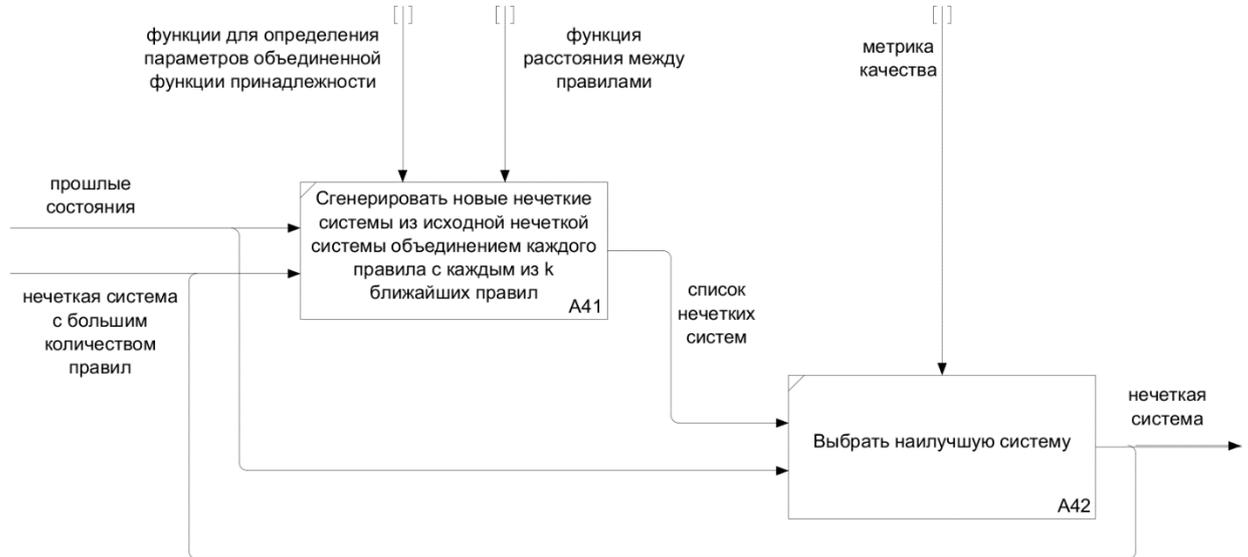


Рисунок 2.10. Метод сокращения количества правил

Подобный подход реализован в других методах построения нечеткой системы на основе кластеризации [11], однако метод слияния правил из [12] жестко привязан к методу разбиения Густафсона-Кесселя, а методы из [13] и [14] не используют функцию ρ

2.3.2 Алгоритм сокращения количества правил в нечетких системах прогнозирования напряжения и состояния заряда

Для нечетких систем прогнозирования напряжения (рис. 2.2 и 2.5) и состояния заряда (2.6) выбраны функции (2.3) и (2.4) расстояния между правилами соответственно:

$$\zeta(R^p, R^q) = |\mu_{\text{напряжение}}^p - \mu_{\text{напряжение}}^q| \quad (2.3)$$

$$\zeta(R^p, R^q) = |\mu_{\text{состояние заряда}}^p - \mu_{\text{состояние заряда}}^q| \quad (2.4)$$

Количество k рассматриваемых ближайших правил задано равным 2.

Функции f и ψ определяются как (2.5) и (2.6) соответственно, при условии, что $\mu_j^p < \mu_j^q$.

$$f(\mu_j^p, \mu_j^q, \sigma_j^p, \sigma_j^q) = \begin{cases} \frac{(\mu_j^p - 3\sigma_j^p) + (\mu_j^q + 3\sigma_j^q)}{2}, & \text{если } \mu_j^p - 3\sigma_j^p \leq \mu_j^q - 3\sigma_j^q \\ & \text{и } \mu_j^p + 3\sigma_j^p \leq \mu_j^q + 3\sigma_j^q \\ \mu_j^p, & \text{если } \mu_j^p - 3\sigma_j^p < \mu_j^q - 3\sigma_j^q \text{ и } \mu_j^p + 3\sigma_j^p > \mu_j^q + 3\sigma_j^q \\ \mu_j^q, & \text{если } \mu_j^p - 3\sigma_j^p > \mu_j^q - 3\sigma_j^q \text{ и } \mu_j^p + 3\sigma_j^p < \mu_j^q + 3\sigma_j^q \end{cases} \quad (2.5)$$

$$\psi(\mu_j^p, \mu_j^q, \sigma_j^p, \sigma_j^q) = \begin{cases} \frac{(\mu_j^p - 3\sigma_j^p) + (\mu_j^q + 3\sigma_j^q)}{6} - \sigma_j^p, & \text{если } \mu_j^p - 3\sigma_j^p \leq \mu_j^q - 3\sigma_j^q \\ & \text{и } \mu_j^p + 3\sigma_j^p \leq \mu_j^q + 3\sigma_j^q \\ \sigma_j^p, & \text{если } \mu_j^p - 3\sigma_j^p < \mu_j^q - 3\sigma_j^q \text{ и } \mu_j^p + 3\sigma_j^p > \mu_j^q + 3\sigma_j^q \\ \sigma_j^q, & \text{если } \mu_j^p - 3\sigma_j^p > \mu_j^q - 3\sigma_j^q \text{ и } \mu_j^p + 3\sigma_j^p < \mu_j^q + 3\sigma_j^q \end{cases} \quad (2.6)$$

Графическая интерпретация (2.5) и (2.6) представлена на рисунках 2.11-2.13.

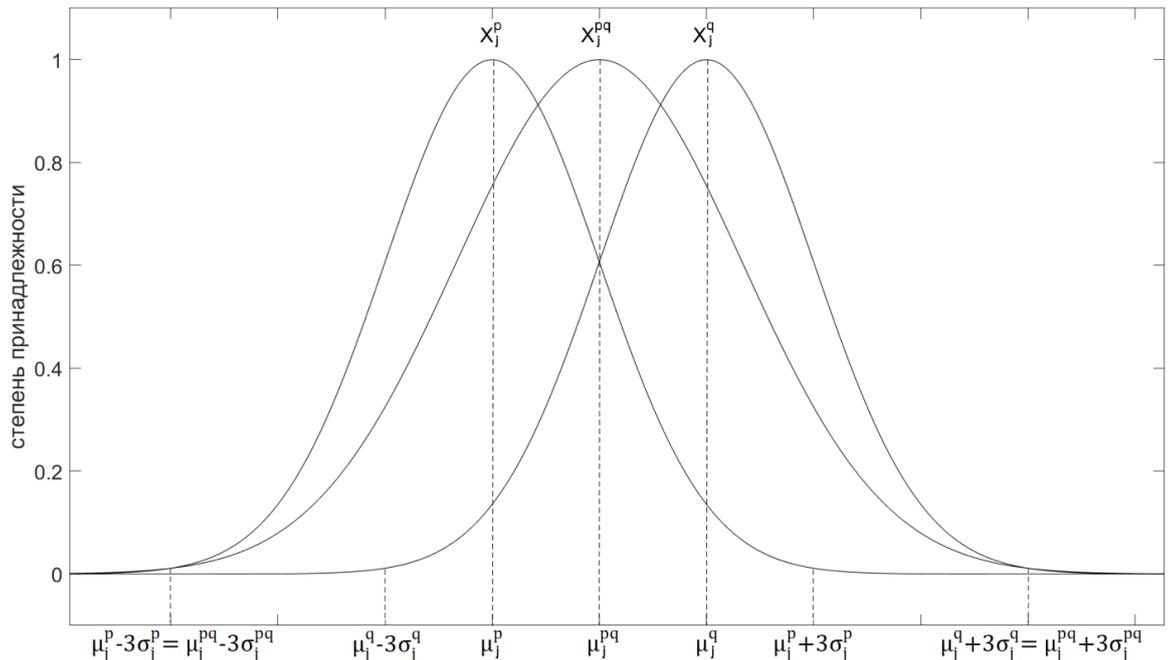


Рисунок 2.11. Случай 1: $\mu_j^p - 3\sigma_j^p \leq \mu_j^q - 3\sigma_j^q$ и $\mu_j^p + 3\sigma_j^p \leq \mu_j^q + 3\sigma_j^q$

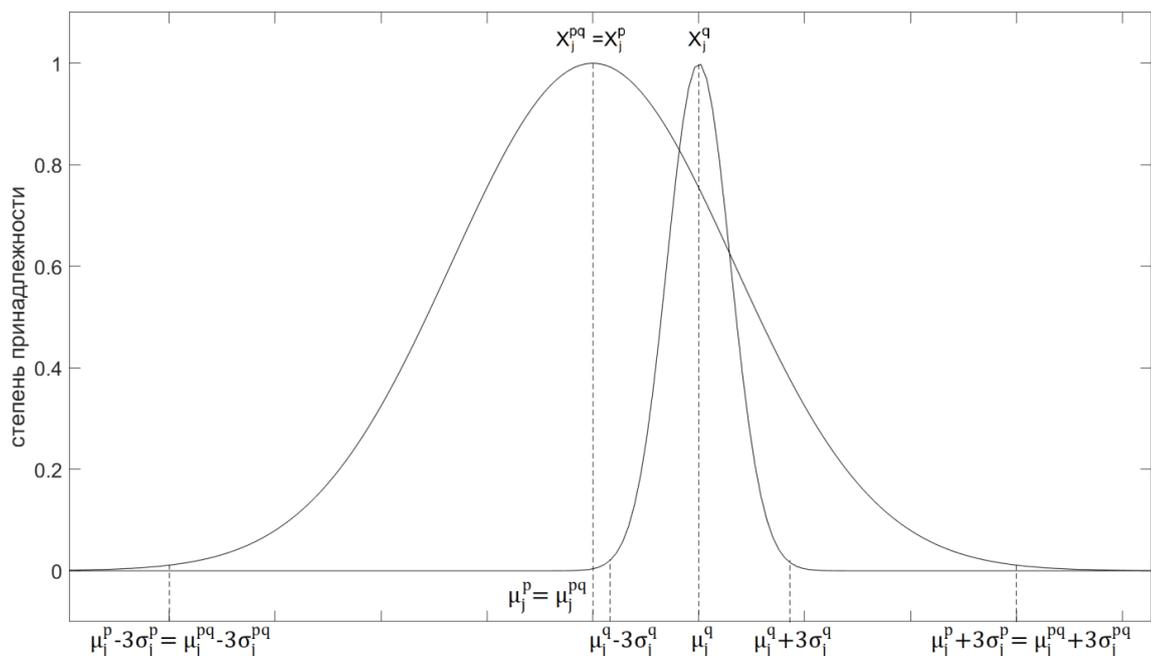


Рисунок 2.12. Случай 2: $\mu_j^p - 3\sigma_j^p < \mu_j^q - 3\sigma_j^q$ и $\mu_j^p + 3\sigma_j^p > \mu_j^q + 3\sigma_j^q$

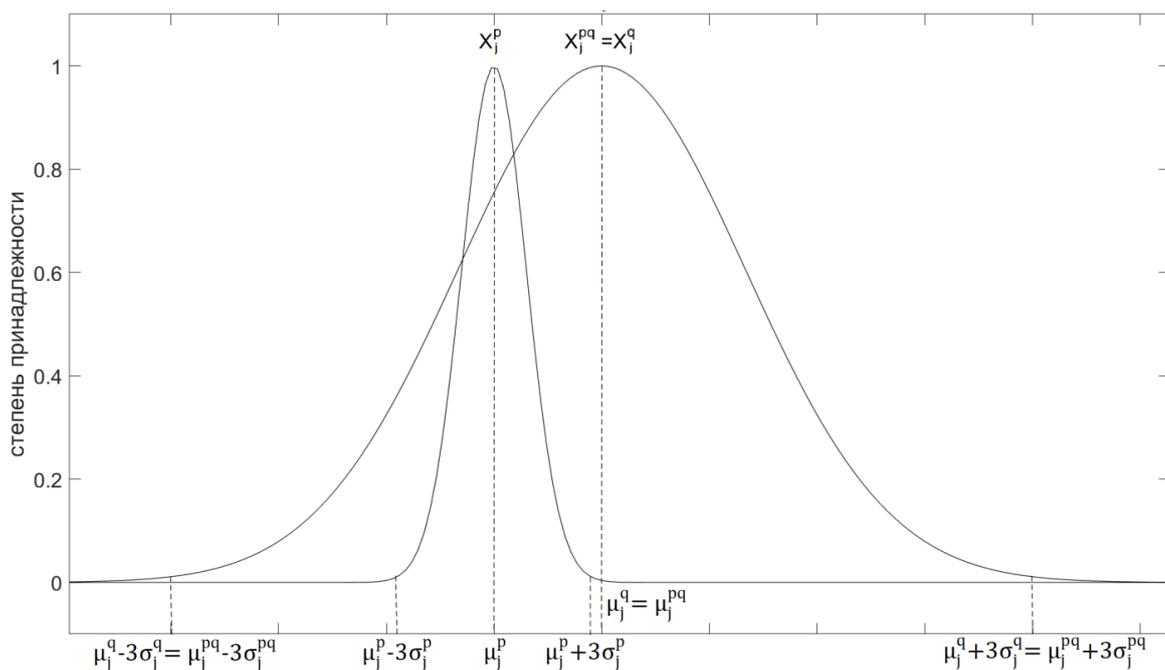


Рисунок 2.13. Случай 3: $\mu_j^p - 3\sigma_j^p > \mu_j^q - 3\sigma_j^q$ и $\mu_j^p + 3\sigma_j^p < \mu_j^q + 3\sigma_j^q$

Метрикой качества нечеткой системы является среднеквадратичная ошибка (2.2).

Далее приведен алгоритм сокращения количества правил в нечетких системах прогнозирования напряжения (состояния заряда).

Вход: максимальное количество итераций M , нечеткая система S .

Выход: новая нечеткая система S' .

Алгоритм:

1. Отсортировать правила из нечеткой системы S по значениям $\mu_{\text{напряжения}} (\mu_{\text{состояния заряда}})$ функций принадлежности, соответствующих терму напряжения (состояния заряда) в левой части правила.

2. Пока количество итераций меньше M :

2.1. Для каждого правила R^p нечеткой системы S , кроме последнего:

2.1.1. Выбрать правило R^p и правило $R^q = R^{p+1}$ в качестве объединяемых правил.

2.1.2. Сформировать нечеткую систему S'' на основе S :

2.1.2.1. Для каждой входной переменной $x_j, j = 1, 2$:

2.1.2.1.1. Сформировать новый терм X_j^{pq} , установить параметры его функции принадлежности по (2.5) и (2.6).

2.1.2.1.2. Удалить термы X_j^p и X_j^q .

2.1.2.2. Сформировать новое правило

$$R^{pq}: \text{если } x_1 \text{ есть } X_1^{pq}, x_2 \text{ есть } X_2^{pq}, \text{ то } y = a_0^{pq} + a_1^{pq} x_1 + a_2^{pq} x_2,$$

где $a_0^{pq} = a_1^{pq} = a_2^{pq} = 0$.

2.1.2.3. Заменить правило R^p правилом R^{pq} (так упорядоченность правил сохраняется).

2.1.2.4. Удалить правило R^q .

2.1.2.5. Инициализировать параметры консеквентов всех правил методом наименьших квадратов.

2.1.3. Рассчитать среднеквадратичную ошибку нечеткой системы S'' по (2.2).

2.1.4. Если среднеквадратичная ошибка меньше наименьшей:

2.1.4.1. Выбрать нечеткую систему S'' как лучшую S' .

2.1.4.2. Обновить наименьшую среднеквадратичную ошибку.

2.2. Задать лучшую нечеткую систему S' в качестве S .

2.3. Увеличить количество итераций на 1.

3. Вернуть S' .

В качестве модификации данного алгоритма предлагается добавление оптимизации параметров новой функции принадлежности на каждой итерации методом роя частиц, причем в качестве начальных задаются параметры, вычисленные по формулам (2.5)-(2.6), а значением целевой функции является среднеквадратичная ошибка (2.2) на обучающей выборке.

2.4 Структуры данных

На рисунке 2.14 приведены структуры, требуемые для описания нечеткой системы с множественными входами и одним выходом. В таблицах 2.2 – 2.5 приведено более подробное описание полей этих структур.

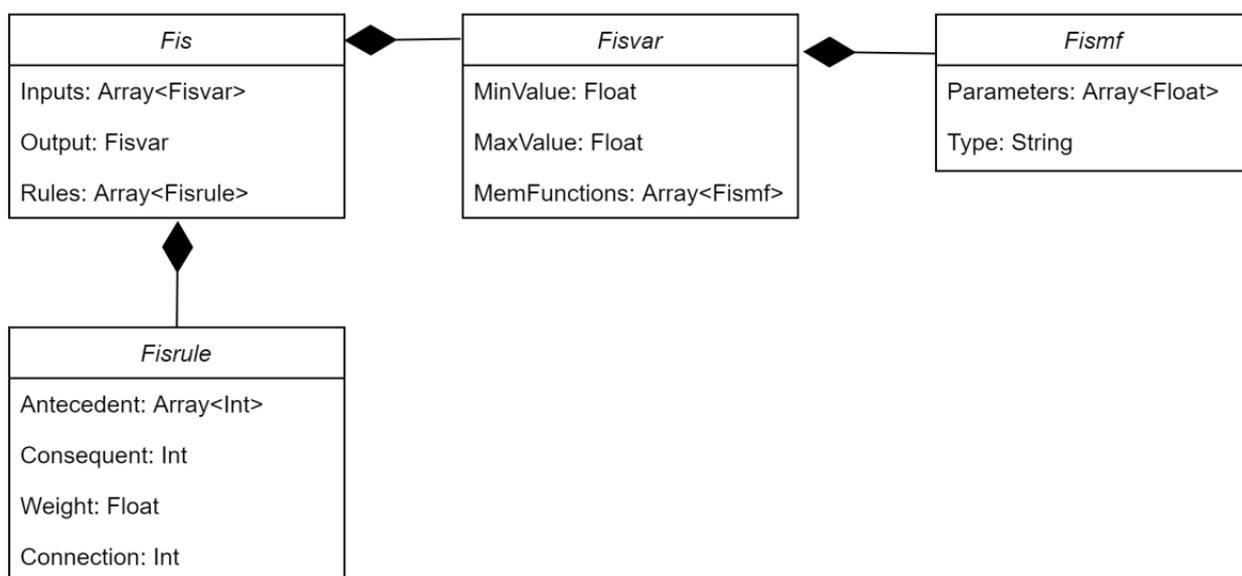


Рисунок 2.14. Структуры для описания нечеткой системы

Таблица 2.2. Структура нечеткой системы *Fis*

Название поля	Тип поля	Описание
Inputs	Array<Fisvar>	Список входных переменных
Output	Fisvar	Выходная переменная
Rules	Array<Fisrule>	Список правил

Таблица 2.3. Структура переменной (входной либо выходной) *Fisvar*

Название поля	Тип поля	Описание
MinValue	Float	Минимальное допустимое значение
MaxValue	Float	Максимальное допустимое значение
MemFunctions	Array<Fismf>	Список термов (функций принадлежности)

Таблица 2.4. Структура функции принадлежности *Fismf*

Название поля	Тип поля	Описание
Parameters	Array<Float>	Список параметров
Type	String	Тип функции принадлежности: “gaussmf” – гауссова, “trimf” – треугольная, “linear” – линейная. Для выходной переменной допустима только линейная функция

Таблица 2.5. Структура правила *Fisrule*

Название поля	Тип поля	Описание
Antecedent	Array<Int>	Список индексов, где <i>i</i> -й элемент есть индекс в массиве MemFunctions <i>i</i> -й входной переменной
Consequent	Int	Индекс в массиве MemFunctions выходной переменной
Weight	Float	Вес правила (от 0 до 1)
Connection	Int	Тип связи между входными переменными: 0 – “или”, 1 – “и”

2.5 Выводы

Разработан метод прогнозирования состояния аккумулятора на основе нечетких моделей. Разработан метод построения нечеткой системы на основе субтрактивной кластеризации, устраняющий её недостаток, заключающийся в формировании кластеров одинакового диаметра. Метод заключается в выделении множества маленьких кластеров и соответственно множества правил и дальнейшей редукции правил. Предложен алгоритм редукции правил. Разработаны структуры данных, необходимые для реализации предложенных методов в программном обеспечении.

3 Технологический раздел

3.1 Выбор языка программирования

В качестве языка программирования в данной работе выбран язык MATLAB, так он имеет пакеты расширения с доступным для редактирования исходным кодом, в которых реализованы алгоритмы построения нечетких систем типа Такаги-Сугено и Мамдани (Fuzzy Logic Toolbox) и алгоритмы оптимизации: генетический алгоритм, рой частиц, алгоритм имитации отжига, поиск по шаблону (Global Optimization Toolbox).

3.2 Описание набора данных

Исходный набор данных представляет собой 5 матриц, каждая из которых содержит 4 столбца: значения отметки времени, тока нагрузки, напряжения и состояния заряда аккумуляторной ячейки AMP20M1HD-A в течение цикла заряда-разряда постоянным током при температуре 25 градусов. Каждой матрице соответствует свой постоянный ток цикла заряда-разряда – 12, 25, 37, 50 и 62 А.

Для построения нечеткой системы использовались данные, собранные при разряде ячейки токами в 12, 25, 50 и 62 А (далее обучающая выборка). Тестирование полученной системы проводилось на данных, полученных при разряде током в 37 А (далее тестовая выборка).

3.3 Структура разработанного программного обеспечения

Разработанное программное обеспечение содержит 4 модуля: модуль загрузки данных, модуль построения нечетких систем, модуль тестирования и визуализации нечетких систем и модуль прогнозирования состояния аккумулятора. Модуль загрузки данных считывает данные о состоянии аккумулятора из файлов в матрицу, модуль построения нечетких систем реализует предложенный метод построения нечеткой системы и аналогичные методы, модуль тестирования и визуализации вычисляет среднеквадратичную ошибку прогнозирования нечеткой системы и отображает различные графики, модуль прогнозирования состояния аккумулятора реализует предложенные методы прогнозирования. На рисунке 3.1 изображена структура

разработанного программного обеспечения.

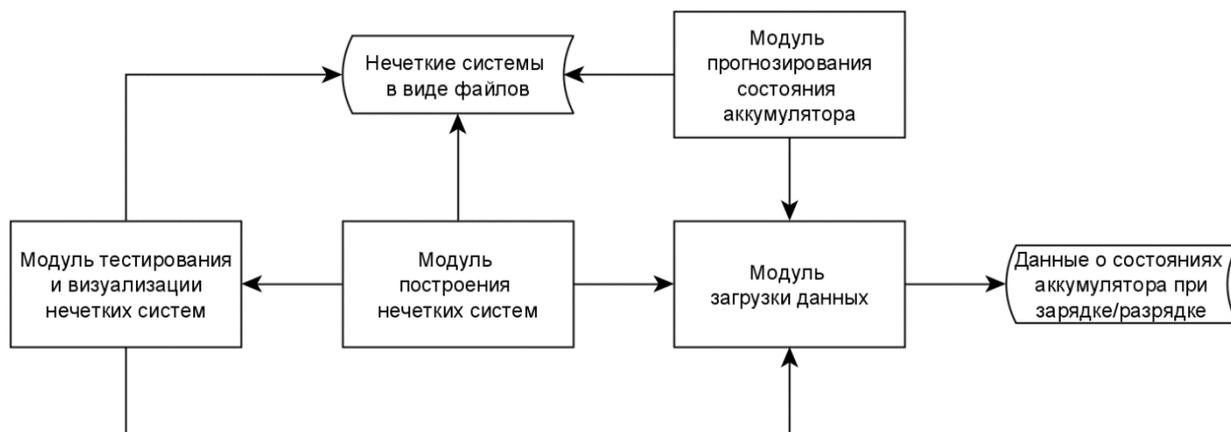


Рисунок 3.1. Структура разработанного программного обеспечения

3.3.1 Модуль загрузки данных

Модуль загрузки данных содержит 12 функций для загрузки данных о состоянии ячейки при разряде: 6 из них загружают данные с постоянным шагом по оси состояния заряда и 6 – с постоянным шагом по оси напряжения. Эти функции перечислены в таблице 3.1.

Таблица 3.1. Функции модуля загрузки данных

Название функции	Описание функции
load_12_Q_step	Загрузка данных о состоянии ячейки при разряде током в 12А с постоянным шагом по оси состояния заряда
load_25_Q_step	То же, что и load_12_Q_step, но для тока в 25 А
load_37_Q_step	То же, что и load_12_Q_step, но для тока в 37 А
load_50_Q_step	То же, что и load_12_Q_step, но для тока в 50 А
load_62_Q_step	То же, что и load_12_Q_step, но для тока в 62 А
load_train_Q_step	То же, что и load_12_Q_step, но для токов в 12, 25, 50 и 62 А
load_12_U_step	Загрузка данных о состоянии ячейки при разряде током в 12 А с постоянным шагом по оси напряжения
load_25_U_step	То же, что и load_12_U_step, но для тока в 25 А
load_37_U_step	То же, что и load_12_U_step, но для тока в 37 А
load_50_U_step	То же, что и load_12_U_step, но для тока в 50 А
load_62_U_step	То же, что и load_12_U_step, но для тока в 62 А
load_train_U_step	То же, что и load_12_U_step, но для токов в 12, 25, 50 и 62 А

В таблице 3.2. приведен интерфейс функций модуля загрузки данных.

Таблица 3.2. Интерфейс функций модуля загрузки данных

Параметр	Тип	Описание	In/Out
step	float	Шаг по оси состояния заряда (для функций load_*_Q_step) или напряжения (для функций load_*_U_step)	In
part	int	Какой участок данных загружать: 0: 20-90% 1: 0-100% 2: 90-100% 3: 97-100% 4: 0-97% 5: 0-20% 6: 20-100%	In
I	array<float>	Значения тока	Out
U	array<float>	Значения напряжения	Out
Q	array<float>	Значения состояния заряда	Out
t	array<float>	Значения отметки времени	Out

3.3.2 Модуль построения нечетких систем

Модуль построения нечетких систем содержит 6 функций, реализующих различные алгоритмы построения нечеткой системы, 2 вспомогательные функции для подбора параметров субтрактивной кластеризации, точку входа – функцию **fis_main**. Эти функции перечислены в таблице 3.3.

Таблица 3.3. Функции модуля построения нечетких систем

Название функции	Описание функции
fis_brute_force	Построение нечеткой системы методом полного перебора для структурной инициализации и методом роя частиц для параметрической инициализации
fis_particle_swarm	Построение нечеткой системы методом роя частиц
fis_grid_anfis	Построение нечеткой системы методом полного перебора для структурной инициализации, методом сеточного разбиения для предварительной инициализации параметров функций принадлежности, методом ANFIS для дополнительной оптимизации всех параметров

Название функции	Описание функции
fis_subclust	Построение нечеткой системы методом субтрактивной кластеризации для структурной инициализации и инициализации параметров функций принадлежности, методом наименьших квадратов для инициализации параметров консеквентов
fis_subclust_anfis	Функция аналогична функции fis_subclust, но параметры построенной нечеткой системы дополнительно оптимизируются методом ANFIS
fis_subclust_merge	Построение нечеткой системы предложенным в данной работе методом
subclust_parms_brute_force	Выбор параметров субтрактивной кластеризации полным перебором
subclust_parms_particle_swarm	Выбор параметров субтрактивной кластеризации методом роя частиц
fis_main	Загрузка данных, построение нечеткой системы одним из шести методов, сохранение построенной системы в файл

В таблице 3.4 представлен интерфейс функции **fis_brute_force**.

Таблица 3.4. Интерфейс функции **fis_brute_force**

Параметр	Тип	Описание	In/Out
x_counts	array<int>	Значения количеств термов входных переменных	In
x	array<array<float>>	Матрица значений входных переменных	In
y	array<float>	Массив значений выходной переменной	In
fis	sugfis	Нечеткая система	Out

В листинге 1 представлен код функции **fis_brute_force** для нечеткой системы с двумя входными переменными. Сначала создается нечеткая система типа Такаги-Сугено, добавляются входные и выходные переменные, для каждой из них указывается диапазон значений, количество термов и тип функций принадлежности. Затем добавляются правила с помощью функции MATLAB **addRule**. Этой функции на вход передается массив, первыми тремя

элементами которого являются индексы функций принадлежности входных и выходных переменных, четвертым – вес правила, пятый отвечает за связь входных переменных в правиле – 0, если “ИЛИ”, и 1, если “И”. Далее заполняются настройки метода обучения системы. Так как структура правил уже определена и требуется инициализировать только параметры, в поле **OptimizationType** указывается значение **tuning** (если бы требовалось инициализировать и структуру правил, и параметры, нужно было бы указать значение **learning**) и вызывается функция MATLAB **tunefis**, которая настраивает систему.

Листинг 1. Функция **fis_brute_force**

```
function fis = fis_brute_force(x_counts, x, y)
    % создание нечеткой системы типа Такаги-Сугено
    fisin = sugfis;
    for i=1:numel(x_counts)
        fisin = addInput(fisin,[min(x(i)),max(x(i))], 'NumMFs',x_counts(i), "MFType", "gaussmf");
    end
    fisin = addOutput(fisin,[min(y), max(y)], 'NumMFs', prod(x_counts), "MFType", "linear");

    % инициализация структуры правил полным перебором
    t = 1;
    for j=1:x_counts(1)
        for k=1:x_counts(2)
            fisin = addRule(fisin, [j k t 1 1]);
            t = t + 1;
        end
    end

    % инициализация параметров рою частиц
    options = tunefisOptions("Method", "particleswarm");
    options.OptimizationType = "tuning";
    [in,out] = getTunableSettings(fisin);
    [fis,~] = tunefis(fisin,[in;out], x, y, options);
end
```

В таблице 3.5 представлен интерфейс функции **fis_particle_swarm**.

Таблица 3.5. Интерфейс функции **fis_particle_swarm**

Параметр	Тип	Описание	In/Out
x_y_count_max	array<int>	Максимальные значения количеств термов входных и выходной переменных	In
x	array<array<float>>	Матрица значений входных переменных	In
y	array<float>	Массив значений выходной переменной	In
fis	sugfis	Нечеткая система	Out

В функции **fis_particle_swarm** так же, как и в функции **fis_brute_force**, создается объект нечеткой системы типа Такаги-Сугено, добавляются входные и выходные переменные с количествами термов, указанных в параметре **x_y_count_max**, и вызывается функция **tinefis** с настроечным параметром **learning**. Эта функция в MATLAB реализована так, что в сгенерированной системе могут получаться правила, у которых в левой части отсутствует одна из входных переменных, что противоречит знаниям предметной области (напряжение зависит и от состояния заряда, и от тока, а состояние заряда – и от тока, и от напряжения). Настроечного параметра, чтобы принудительно включить обе переменные в левые части правил, нет, поэтому пришлось вносить изменения в исходный код MATLAB.

Функция **tunefis** вызывает функцию **createFISProblemData** для создания структуры **problem**, описывающей задачу оптимизации, которая далее передается в функцию **particleswarm** для поиска решения методом роя частиц. Функция **createFISProblemData**, в свою очередь, вызывает функцию **preprocessFIS**, которая генерирует систему с начальной структурой правил. До внесения изменений начальная структура представляла собой $(N_1 + 1) \times (N_2 + 1) \times \dots \times (N_i + 1) \times \dots \times (N_m + 1) - 1$ правил, где N_i – количество термов i -й входной или выходной переменной, $i = \overline{1, m}$, m – суммарное количество входных и выходных переменных, каждое правило описывалось массивом из всех единиц, то есть генерировались правила вида (3.1).

$$\text{если } x_1 \text{ есть } X_1 \text{ И } x_2 \text{ есть } X_1 \dots \text{ то } y_1 \text{ есть } Y_1 \text{ (вес 1)} \quad (3.1)$$

После внесения изменений генерируется $N_1 \times N_2 \times \dots \times N_i \times \dots \times N_m$ правил, так как больше не нужно учитывать правила с неполной левой частью. В листинге 2 представлены изменения, внесенные в функцию **preprocessFIS**.

Листинг 2. Изменения в функции **preprocessFIS**

```
numMFs = fuzzy.internal.utility.getVarMFs(fisin.Inputs);
% New rules will consider zero MF indices.
% numAllCombRules = prod(numMFs+1) - 1;
% начало изменений, строчкой выше – как было раньше
numAllCombRules = prod(numMFs);
% конец изменений
```

После вызова функции **preprocessFIS** заполняются диапазоны изменения значений параметров и остальные поля структуры **problem**. При заполнении диапазонов проверяется, может ли входной или выходной параметр отсутствовать в правиле. Соответственно, перед выполнением данной проверки нужно запретить нулевое значение индексов функций принадлежности для каждого правил, означающее, что параметр отсутствует. В листинге 3 приведен добавленный в функцию **createFISProbleData** код.

Листинг 3. Код, добавленный в функцию **createFISProbleData**

```
% Update rule spec for learning
if options.OptimizationType == "learning"
    [~,~,rule] = getTunableSettings(fisin);
    % начало добавленного кода
    nr = numel(fisin.Rules);
    in = numel(fisin.Inputs);
    for i = 1:nr
        for j = 1:in
            rule(i).Antecedent.AllowEmpty(j) = false;
        end
        rule(i).Consequent.AllowEmpty(1) = false;
    end
end
% конец добавленного кода
```

В таблице 3.6 представлен интерфейс функции **fis_grid_anfis**.

Таблица 3.6. Интерфейс функции **fis_grid_anfis**

Параметр	Тип	Описание	In/Out
x_counts	array<int>	Значения количеств термов входных переменных	In
x	array<array<float>>	Матрица значений входных переменных (обучающая выборка)	In
y	array<float>	Массив значений выходной переменной (обучающая выборка)	In
x_test	array<array<float>>	Матрица значений входных переменных (тестовая выборка)	In
y_test	array<float>	Массив значений выходной переменной (тестовая выборка)	In
epochs	int	Количество эпох обучения ANFIS	In
fis	sugfis	Нечеткая система	Out

В листинге 4 представлен код функции **fis_grid_anfis**. Эта функция использует функцию MATLAB **anfis**, которая настраивает систему, используя указанную обучающую выборку и параметры. Параметрами являются начальная нечеткая система, тестовая выборка для предотвращения

переобучения (после каждой эпохи будут печататься значения среднеквадратичной ошибки для обучающей и тестовой выборки, и можно будет отследить, когда ошибка для тестовой выборки перестает уменьшаться), алгоритм обучения: градиентный спуск (по умолчанию) или метод наименьших квадратов, и количество эпох. Функция **fis_grid_anfis** в качестве начальной нечеткой системы задает массив количеств термов входных переменных, что означает автоматическую генерацию начальной системы методом сеточного разбиения.

Листинг 4. Функция **fis_grid_anfis**

```
function fis = fis_grid_anfis(x_counts, x, y, x_test, y_test, epochs)
    opt = anfisOptions('InitialFIS', x_counts, 'EpochNumber', epochs);
    opt.ValidationData = [x_test y_test];
    fis = anfis([x y], opt);
end
```

В таблице 3.7 представлен интерфейс функции **fis_sublust**

Таблица 3.7. Интерфейс функции **fis_sublust**

Параметр	Тип	Описание	In/Out
clusters_radii	array<float>	Диаметры кластеров	In
x	array<array<float>>	Матрица значений входных переменных	In
y	array<float>	Массив значений выходной переменной	In
fis	sugfis	Нечеткая система	Out

В листинге 5 представлен код функции **fis_sublust**. Эта функция использует функцию MATLAB **genfis**.

Листинг 5. Функция **fis_sublust**

```
function fis = fis_subclust(clusters_radii, x, y)
    opt = genfisOptions('SubtractiveClustering', 'ClusterInfluenceRange', clusters_radii);
    fis = genfis(x, y, options);
end
```

В таблице 3.8 представлен интерфейс функции **fis_subclust_anfis**.

Таблица 3.8. Интерфейс функции **fis_subclust_anfis**

Параметр	Тип	Описание	In/Out
cluster_radii	array<float>	Диаметры кластеров	In
x	array<array<float>>	Матрица значений входных переменных (обучающая выборка)	In
y	array<float>	Массив значений выходной переменной (обучающая выборка)	In
x_test	array<array<float>>	Матрица значений входных переменных (тестовая выборка)	In
y_test	array<float>	Массив значений выходной переменной (тестовая выборка)	In
epochs	int	Количество эпох обучения ANFIS	In
fis	sugfis	Нечеткая система	Out

Код этой функции аналогичен коду функции **fis_grid_anfis** (листинг 4) за тем исключением, что в качестве начальной передается нечеткая система, полученная в результате вызова функции **fis_sublust**.

В таблице 3.9 представлен интерфейс функции **fis_sublust_merge**

Таблица 3.9. Интерфейс функции **fis_sublust_merge**

Параметр	Тип	Описание	In/Out
clusters_radii	array<float>	Диаметры кластеров	In
iters	int	Количество итераций	In
x	array<array<float>>	Матрица значений входных переменных	In
y	array<float>	Массив значений выходной переменной	In
fis	sugfis	Нечеткая система	Out

Код этой функции представлен в листинге 6. Функция **initial_split** выполняет формирование нечеткой системы методом субтрактивной кластеризации и методом наименьших квадратов, функция **get_centers_and_sigmas** возвращает параметры функций принадлежности для заданной нечеткой системы, функция **sort_by_in2** выполняет сортировку по центрам кластеров на оси второй входной переменной (заряда или напряжения), функция **create_new_fis** формирует нечеткую систему на основе параметров функций принадлежности и обучающей выборки, функция

merge_two_rules рассчитывает параметры объединенной функции принадлежности.

Листинг 6. Функция **fis_sublust_merge**

```

fis = initial_split([clusters_radii,x, y);
[centers,sigmas] = get_centers_and_sigmas(fis);
[centers,sigmas] = sort_by_in2(centers,sigmas);
fis = create_new_fis(centers, sigmas, x, y);

for k=1:iters
    min_error = 10000;
    best_fis = fis;
    num_rule = numel(fis.Rules);
    for i=1:num_rule-1
        j = i + 1;
        [new_centers, new_sigmas] = merge_two_rules(centers, sigmas, i, j);
        new_fis = create_new_fis(new_centers, new_sigmas, x, y);
        error = test_fis2(new_fis, x, y);
        if (error < min_error)
            min_error = error;
            best_fis = new_fis;
        end
    end
    fis = best_fis;
end

```

Для подбора параметров субтрактивной кластеризации реализованы две функции: **subclust_parms_brute_force**. В таблице 3.10 представлен интерфейс функции **subclust_parms_brute_force**.

Таблица 3.10. Интерфейс функции **subclust_parms_brute_force**

Параметр	Тип	Описание	In/Out
min_radii	array<float>	Минимальные значения диаметров кластеров	In
max_radii	array<float>	Максимальные значения диаметров кластеров	In
max_rules_count	int	Максимальное количество правил	In
step	float	Шаг перебора	In
radii_count	int	Количество строк в выходной матрице параметров	In
x	array<array<float>>	Матрица значений входных переменных	In
y	array<float>	Массив значений выходной переменной	In
radii	array<array<float>>	Матрица значений параметров	Out

Функция **subclust_parms_brute_force** выполняет полный перебор значений параметров с шагом **step** в диапазоне от **min_radii** до **max_radii** и возвращает **radii_count** массивов значений параметров, при которых

полученная нечеткая система имеет наименьшую среднеквадратичную ошибку на обучающей выборке (2.2) и количество правил не больше **max_rules_count**.

Интерфейс функции **subclust_parms_particle_swarm** приведен в таблице 3.11.

Таблица 3.11. Интерфейс функции **subclust_parms_particle_swarm**

Параметр	Тип	Описание	In/Out
min_radii	array<float>	Минимальные значения диаметров кластеров	In
max_radii	array<float>	Максимальные значения диаметров кластеров	In
max_rules_count	int	Максимальное количество правил	In
x	array<array<float>>	Матрица значений входных переменных	In
y	array<float>	Массив значений выходной переменной	In
radii	array<float>	Матрица значений параметров	Out

Функция **subclust_parms_particle_swarm** использует функцию MATLAB **particleswarm** из пакета Global Optimization Toolbox, которой на вход передается целевая функция, количество её аргументов и диапазоны значений аргументов. Чтобы учесть ограничение на количество правил, значение целевой функции устанавливается равным 10000000 в случае, если количество правил превышает **max_rules_count**, иначе значением целевой функции является среднеквадратичная ошибка нечеткой системы на обучающей выборке (2.2).

3.3.3 Модуль тестирования и визуализации нечетких систем

Данный модуль содержит 3 функций: функцию для построения графиков, позволяющих оценить точность прогнозирования, функцию для вычисления среднеквадратичной ошибки, функцию для визуализации нечеткой системы. Эти функции перечислены в таблице 3.12.

Таблица 3.12. Функции модуля тестирования и визуализации

Название функции	Описание функции
plot_errors_fis	Построение графиков значения ошибки прогнозирования выходной переменной и реального и спрогнозированного значения выходной переменной
test_fis2	Вычисление среднеквадратичной ошибки прогнозирования
plot_fis	Построение графиков функций принадлежности и выходной поверхности нечеткой системы

3.3.4 Модуль прогнозирования состояния аккумулятора

Данный модуль содержит одну функцию **predict_by_2_fis**, которая итеративно применяет нечеткий вывод для двух нечетких систем, чтобы спрогнозировать будущие значения состояния заряда и напряжения. Интерфейс этой функции приведен в таблице 3.13.

Таблица 3.13. Интерфейс функции **predict_by_2_fis**

Параметр	Тип	Описание	In/Out
fisU	sugfis	Нечеткая система прогнозирования напряжения	In
fisQ	sugfis	Нечеткая система диагностики заряда	In
Q0	float	Начальное значение состояния заряда	In
I	array<float>	Массив значений токов	In
U	array<float>	Массив спрогнозированных значений напряжения	Out
Q	array<float>	Массив спрогнозированных значений состояния заряда	Out

3.4 Выводы

На языке MATLAB разработано программное обеспечение, состоящее из 4-х модулей: модуля загрузки данных, модуля построения нечетких систем, модуля тестирования и визуализации нечетких систем и модуля прогнозирования состояния аккумулятора.

4 Исследовательский раздел

4.1 Сравнение результатов работы предлагаемого алгоритма построения нечеткой системы при разных параметрах субтрактивной кластеризации

На рисунке 4.1 представлена зависимость среднеквадратичной ошибки нечеткой системы для прогнозирования напряжения по току и состоянию заряда от количества правил для обучающей выборки, на рисунке 4.2 представлена та же зависимость для тестовой выборки.

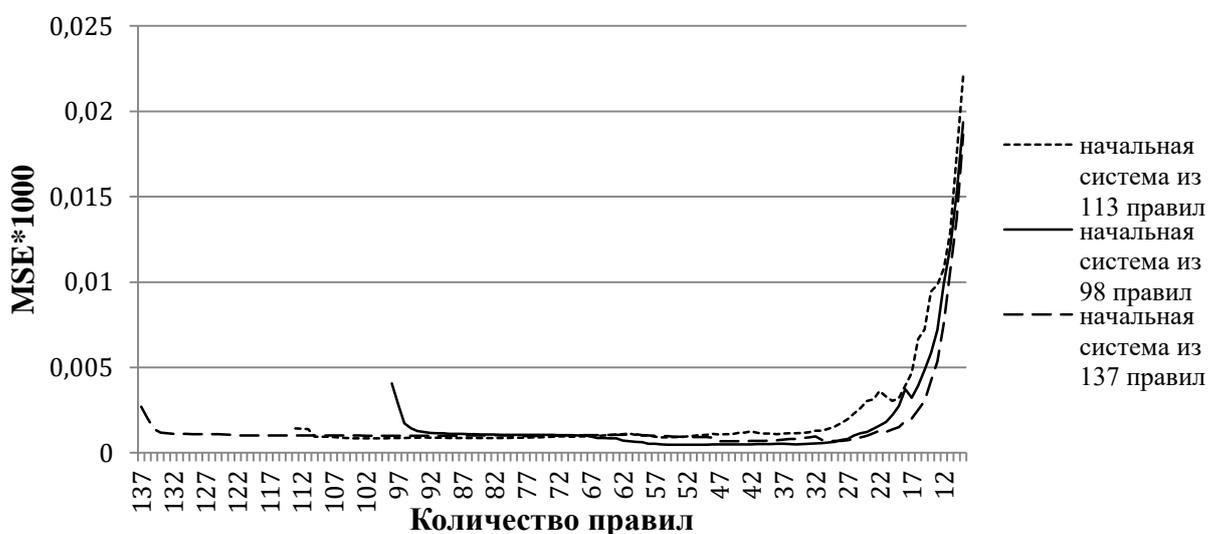


Рисунок 4.1. Зависимость среднеквадратичной ошибки нечеткой системы для прогнозирования напряжения от количества правил для обучающей выборки

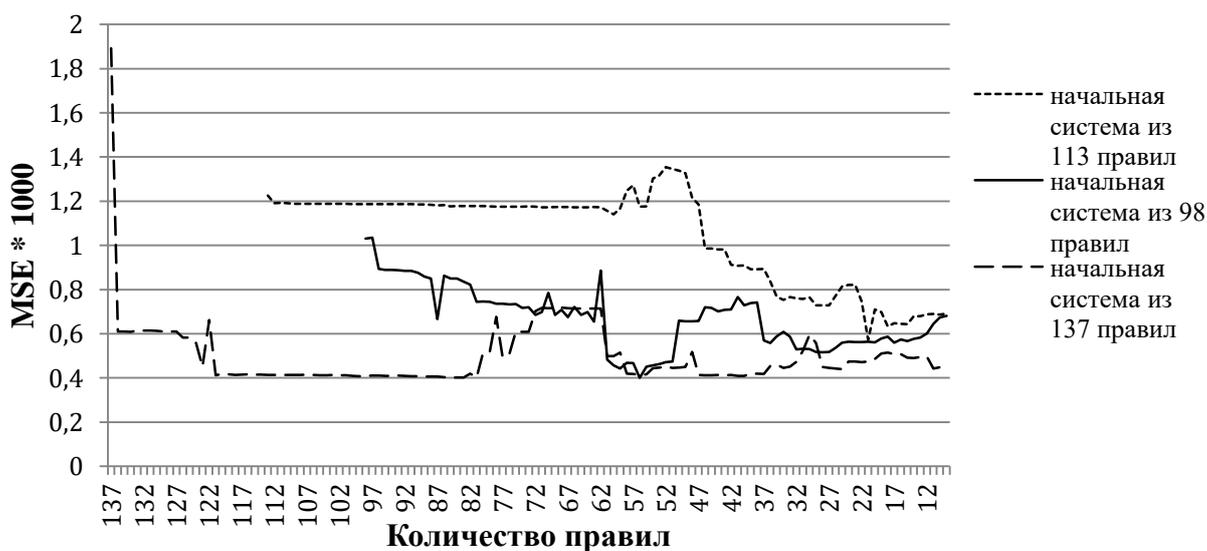


Рисунок 4.2. Зависимость среднеквадратичной ошибки нечеткой системы для прогнозирования напряжения от количества правил для тестовой выборки

На рисунке 4.3 представлены результаты прогнозирования напряжения системой из 25 правил (начальная система содержала 137 правил).

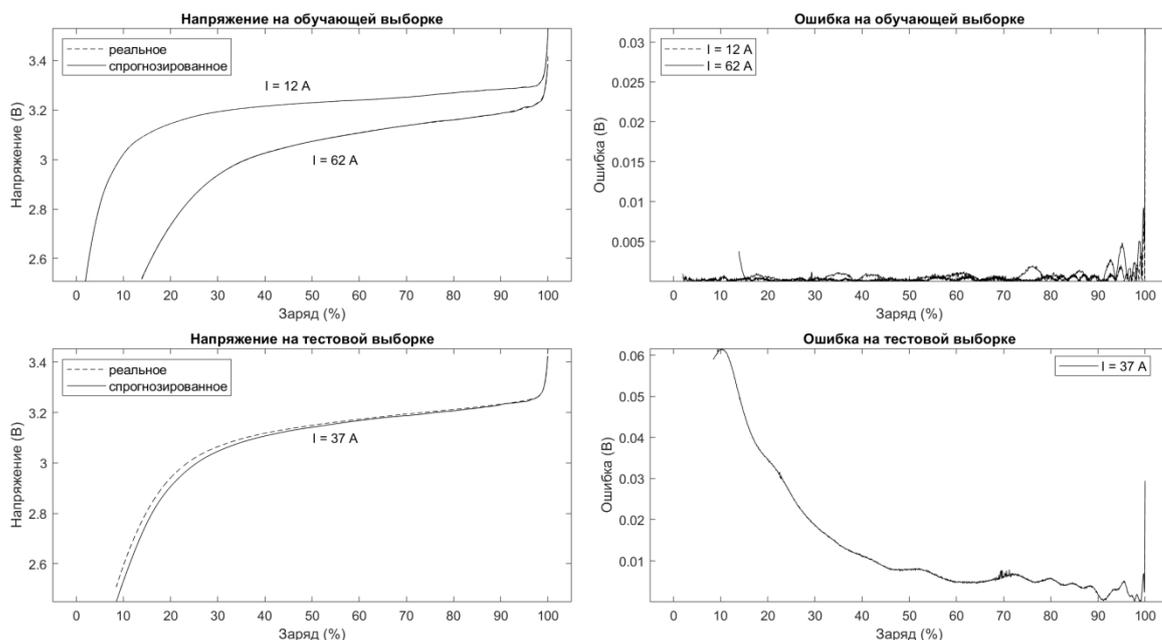


Рисунок 4.3. Результаты прогнозирования напряжения

На рисунке 4.4 представлена зависимость среднеквадратичной ошибки нечеткой системы для прогнозирования состояния заряда по току и напряжению от количества правил для обучающей выборки, на рисунке 4.5 представлена та же зависимость для тестовой выборки.



Рисунок 4.4. Зависимость среднеквадратичной ошибки нечеткой системы для прогнозирования состояния заряда от количества правил для обучающей выборки



Рисунок 4.5. Зависимость среднеквадратичной ошибки нечеткой системы для прогнозирования состояния заряда от количества правил для тестовой выборки

Видно, что для обучающей выборки среднеквадратичная ошибка растет экспоненциально при уменьшении количества правил, для тестовой – явно выраженной зависимости нет. Это может говорить о недостаточном количестве данных для построения и тестирования нечеткой системы.

На рисунке 4.6 представлены результаты прогнозирования состояния заряда системой из 58 правил (начальная система содержала 155 правил).

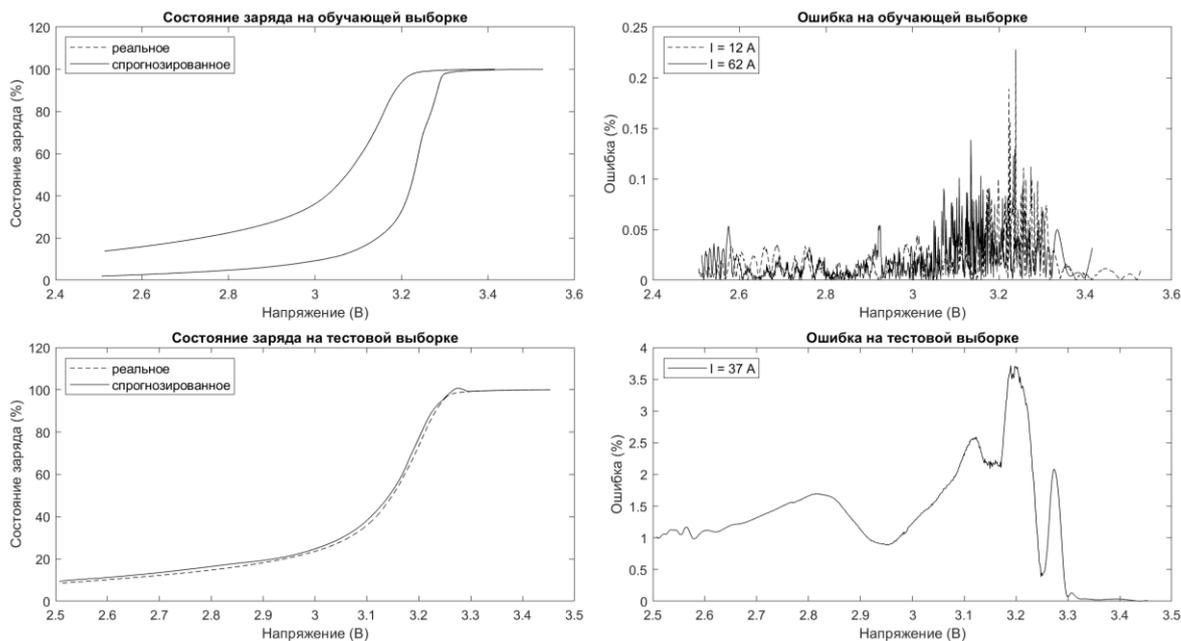


Рисунок 4.6. Результаты прогнозирования состояния заряда

4.2 Сравнение результатов работы предлагаемого метода построения нечеткой системы и аналогов

Проведено сравнение шести методов, реализованных в модуле построения нечетких систем (см. таблицу 3.3). Выбор параметров субтрактивной кластеризации (диаметров кластеров) для методов 4 (субтрактивная кластеризация + МНК) и 5 (субтрактивная кластеризация + МНК + ANFIS) осуществлялся полным перебором всех возможных комбинаций параметров с шагом 0.1 и выбором 10 наилучших комбинаций. На рисунке 4.7 представлен график значений среднеквадратичных ошибок моделей, построенных методами 4 и 5 при разных параметрах субтрактивной кластеризации. Для метода 6 (предлагаемый) задавались следующие параметры субтрактивной кластеризации: 0.98, 0.02, 0.06

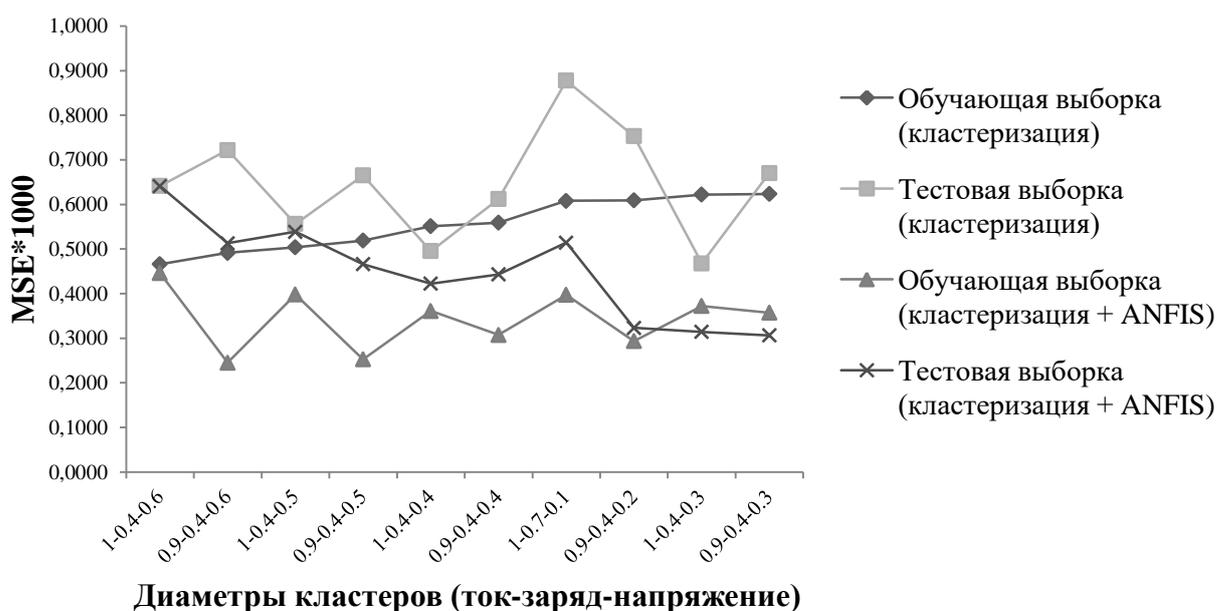


Рисунок 4.7. Зависимость среднеквадратичной ошибки систем 4 и 5 от параметров кластеризации

Видно, что уменьшение среднеквадратичной ошибки после применения ANFIS чувствительно к выбору параметров кластеризации.

В таблице 4.1 представлены значения среднеквадратичных ошибок и количества термов и правил систем, полученных методами 1-6. Для метода 2 (рой частиц) в скобках указано заданное максимальное количество термов. Жирным шрифтом в таблице 4.1 выделены строки, соответствующие системам

с наименьшей среднеквадратичной ошибкой на тестовой выборке, курсивом – строки, соответствующие системам с некорректной выходной поверхностью (рисунок 4.8).

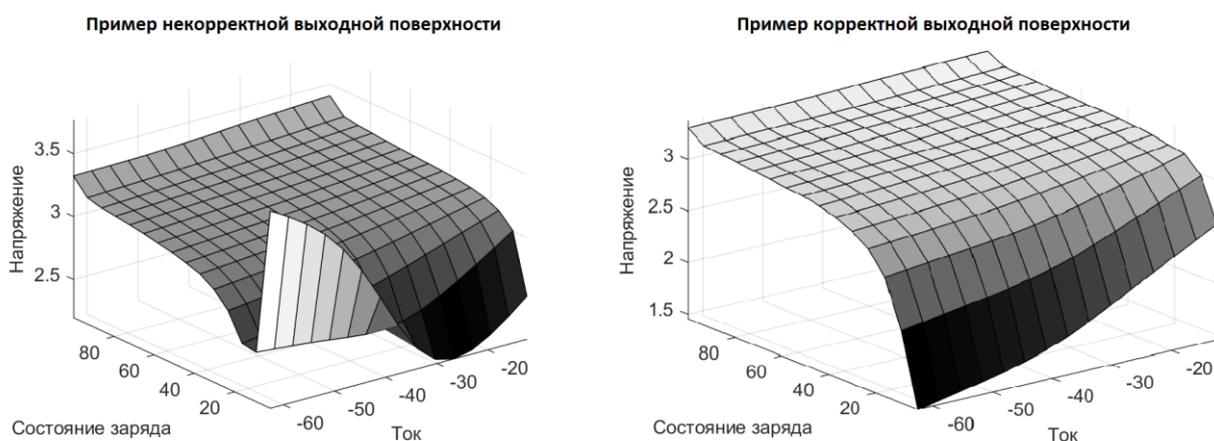


Рисунок 4.8. Примеры некорректной и корректной выходных поверхностей

Система 1 (полный перебор + рой частиц) дает наименьшую среднеквадратичную ошибку при двух термах для тока и семи для состояния заряда. Примерно такую же среднеквадратичную ошибку, как на обучающей, так и на тестовой выборке, имеет система 2 (рой частиц), но содержит при этом в 2,3 раза меньше правил, чем система 1.

Система 3 (полный перебор + сеточное разбиение + ANFIS) демонстрирует наилучшие характеристики при двух термах для тока и от двух до четырех термов для состояния заряда. Дальнейшее увеличение количества термов как для тока, так и для состояния заряда приводит к некорректной выходной поверхности модели и ухудшению её аппроксимирующей способности.

Среди всех систем наименьшую среднеквадратичную ошибку прогнозирования на тестовой выборке демонстрирует система 5 (субтрактивная кластеризация + МНК + ANFIS), на обучающей выборке – система 6 (предлагаемый метод).

Таблица 4.1. Сравнение нечетких систем для прогнозирования напряжения

Метод	Кол-во термов для тока	Кол-во термов для заряда	Кол-во правил	MSE * 1000 на обучающей выборке	MSE * 1000 на тестовой выборке	Максимальная ошибка на обучающей выборке	Максимальная ошибка на тестовой выборке
Полный перебор + рой частиц	2	4	8	0.5765	1.6964	0.2335	0.2013
	2	5	10	2.0155	1.0399	0.2124	0.1635
	2	6	12	0.4533	1.4662	0.1937	0.1866
	2	7	14	0.1869	0.4360	0.1992	0.1519
	2	8	16	0.1262	0.8564	0.1711	0.1329
	2	9	18	1.9238	0.7466	0.1850	0.1424
	3	4	12	0.9354	1.7047	0.2130	0.1562
	3	5	15	2.1739	3.6666	0.2170	0.1445
	3	6	18	2.3718	1.2228	0.2079	0.1875
	4	4	16	0.9695	3.3300	0.2399	0.1934
	4	5	20	0.2585	2.2947	0.2322	0.1799
Рой частиц	2	3(4)	4(4)	2.4103	3.8287	0.2471	0.2558
	2	3(4)	5(8)	0.2863	0.8115	0.2190	0.1642
	2	2(5)	3(5)	1.3206	2.9935	0.2748	0.2544
	2	4(5)	5(10)	0.2347	0.8344	0.2300	0.1938
	2	3(6)	4(6)	0.1730	0.7872	0.1925	0.1684
	2	5(6)	6(12)	0.1538	0.4263	0.1958	0.1709
	3	5(5)	7(15)	0.1884	0.8086	0.2098	0.1832
Полный перебор + сеточное разбиение + ANFIS	2	2	4	0.4510	1.1533	0.1086	0.0704
	2	3	6	0.3918	1.2288	0.1623	0.1400
	2	4	8	0.0956	1.3377	0.1561	0.1275
	2	5	10	0.1734	1.0119	0.1825	0.1554
	2	8	16	0.0295	0.9325	0.1199	0.1028
	2	10	20	0.0106	0.9522	0.0693	0.0935
	3	3	9	0.1787	27.388	0.1828	0.1172
	3	4	12	0.2609	58.878	0.1833	0.1874
	3	5	15	0.0185	2.7839	0.0766	0.1208
	4	4	16	0.0855	546.55	0.1352	1.1812
Субтр. класт. + МНК	5	5	5	0.4661	0.6413	0.1983	0.1753
Субтр. класт. + ANFIS	6	6	6	0.2933	0.3231	0.1982	0.1732
Предлагаемый метод	25	25	25	0.0009	0.4396	0.03169	0.06165

На рисунках 4.9–4.10 для нечетких систем 1-3 с теми характеристиками, которые выделены в таблице 4.2 жирным шрифтом, представлены результаты прогнозирования напряжения. Видно, что ошибка принимает наибольшие значения на левом крае функции, где состояние заряда уменьшается от 30% до 0%, и на правом крае, где состояние заряда уменьшается от 100% до 97%.

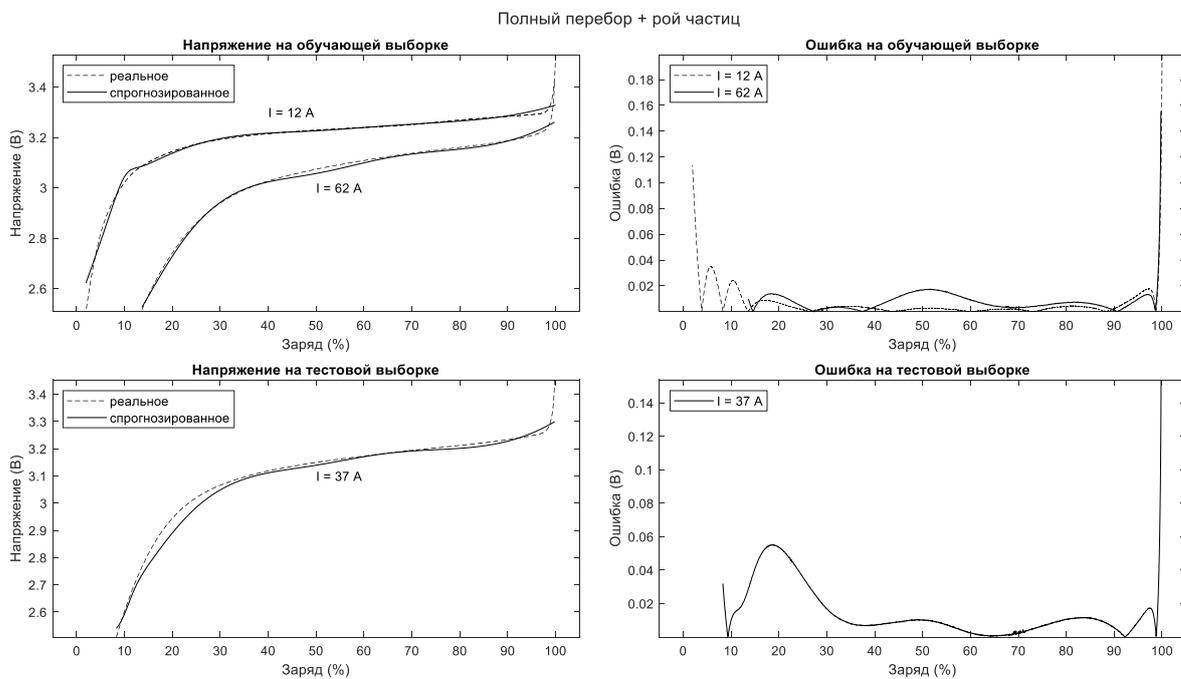


Рисунок 4.9 Результаты прогнозирования системой, полученной полным перебором и роем частиц

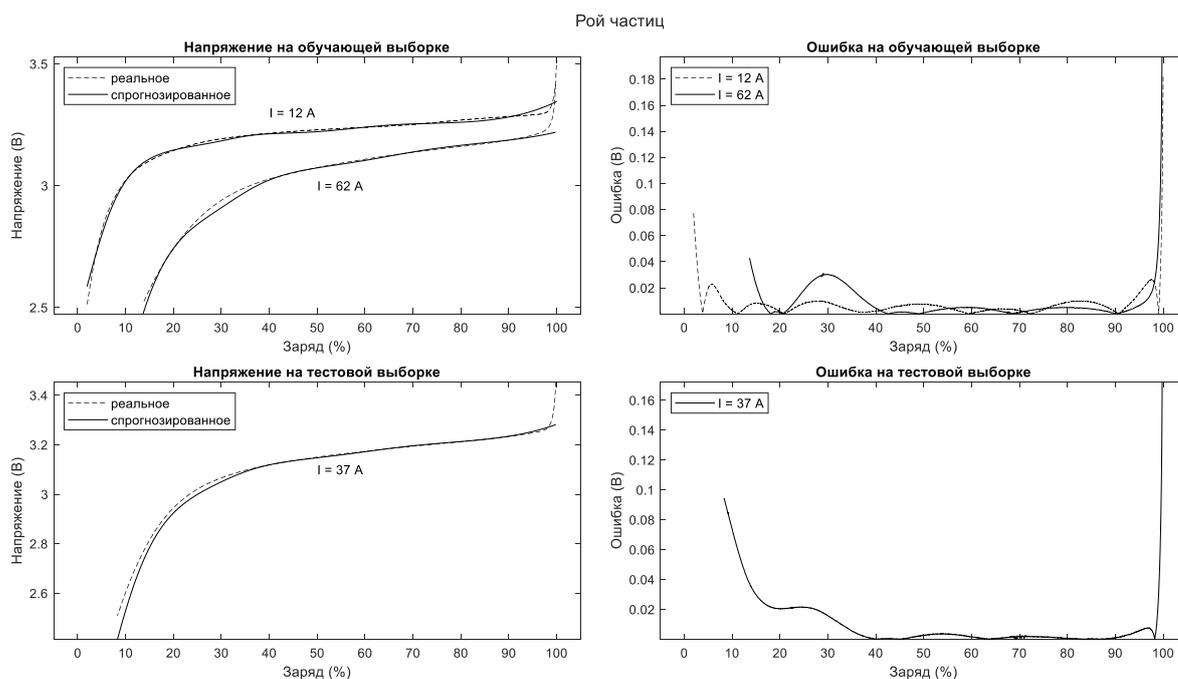


Рисунок 4.10. Результаты прогнозирования системой, полученной роем частиц

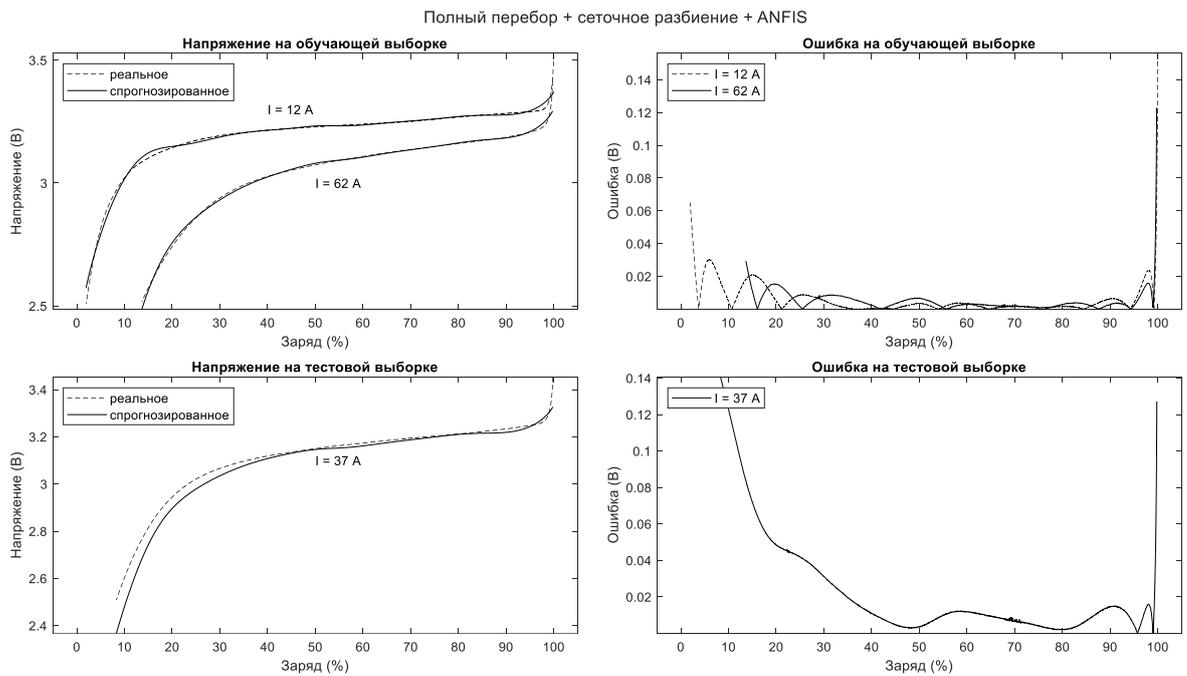


Рисунок 4.11. Результаты прогнозирования системой, полученной полным перебором, сеточным разбиением и ANFIS

На рисунках 4.12-4.13 представлены результаты прогнозирования напряжения для систем 4 и 5. У них, как и у систем 1-3, ошибка принимает наибольшие значения на правом крае, а на интервале, где состояние заряда изменяется от 97% до 0%, изменяется скачками, причем у системы 5 эти скачки меньше.

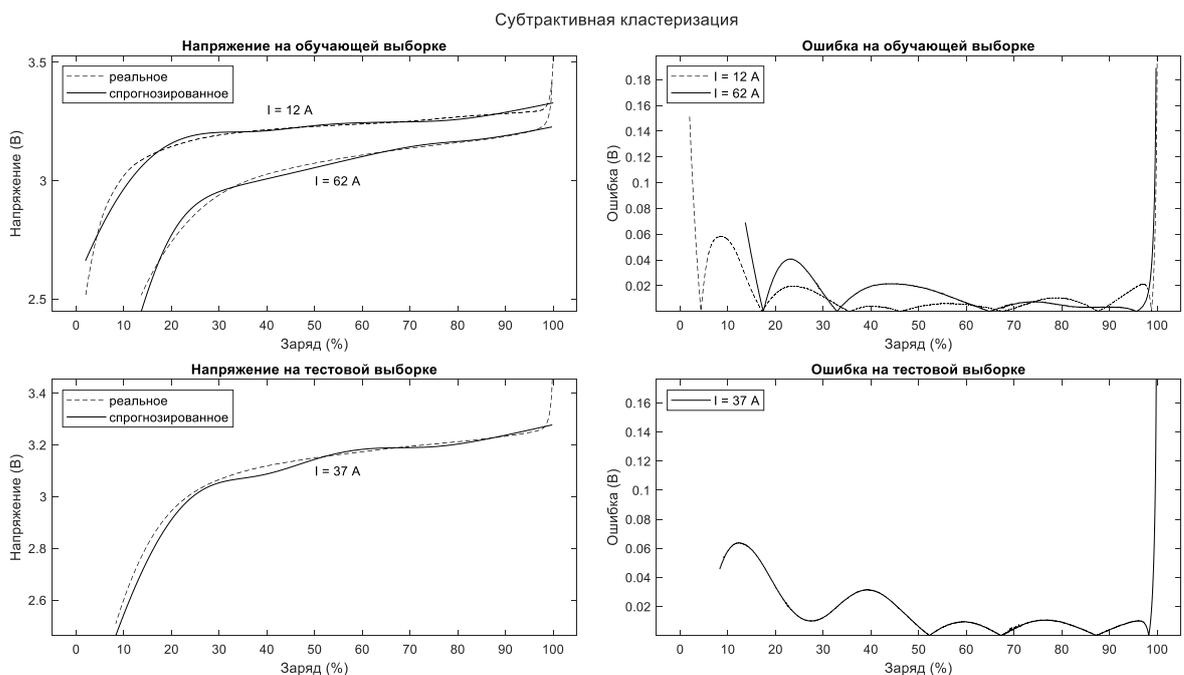


Рисунок 4.12. Результаты прогнозирования системой, полученной субтрактивной кластеризацией и МНК

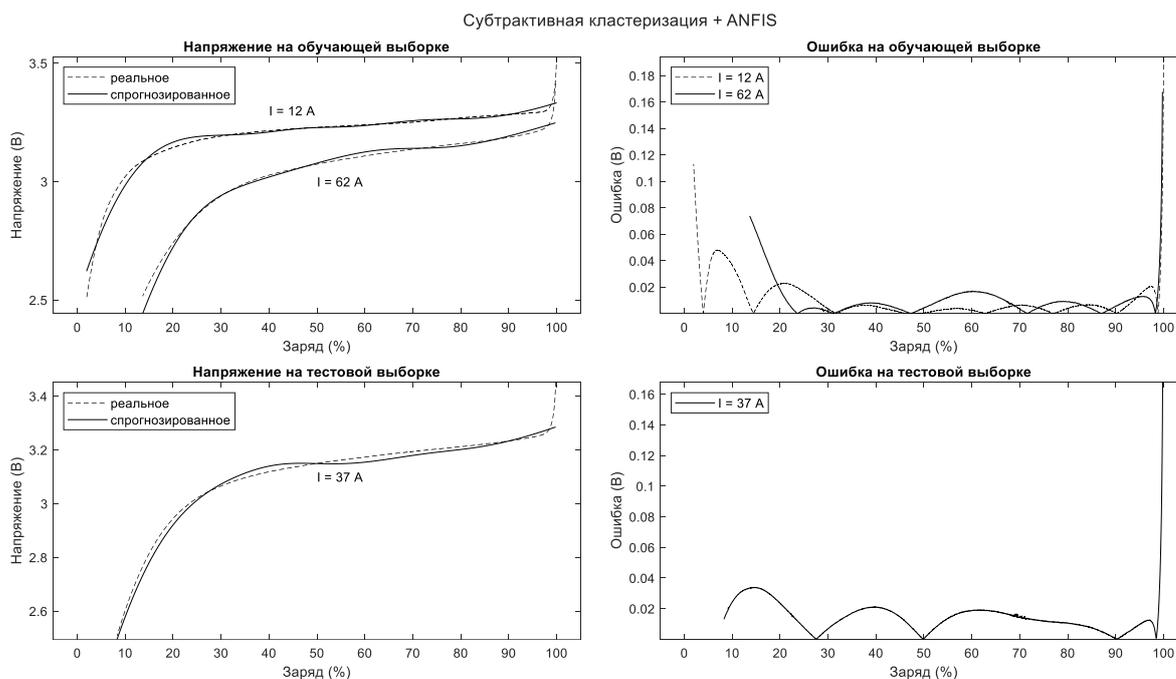


Рисунок 4.13. Результаты прогнозирования системой, полученной субтрактивной кластеризацией, МНК и ANFIS

Все рассмотренные методы могут генерировать нечеткую модель, которая довольно точно прогнозирует напряжение по току нагрузки и состоянию заряда. При этом у всех моделей ошибка распределена неравномерно вдоль оси состояния заряда.

Неравномерность распределения ошибки при использовании методов оптимизации (рой частиц, ANFIS) связана с тем, что эти методы работают только со среднеквадратичной ошибкой прогнозирования и никак не выделяют области с наибольшей ошибкой, чтобы её уменьшить.

Причина скачков и увеличенной ошибки на правом крае при использовании алгоритма субтрактивной кластеризации заключается в его недостатке – формировании кластеров одинакового размера (рисунок 4.14). В измерении состояния заряда корректней было бы задать маленькие диаметры кластеров в интервале от 100% до 97%, большие в интервале от 97% до 30% и средние в интервале от 30% до 0%, так как для этих интервалов значение напряжения меняется сильно, слабо и средне соответственно. Именно так и получается предлагаемым методом (рисунок 4.15), поэтому у модели,

построенной им, ошибка на правом крае меньше, чем у остальных моделей, и нет скачков ошибки.

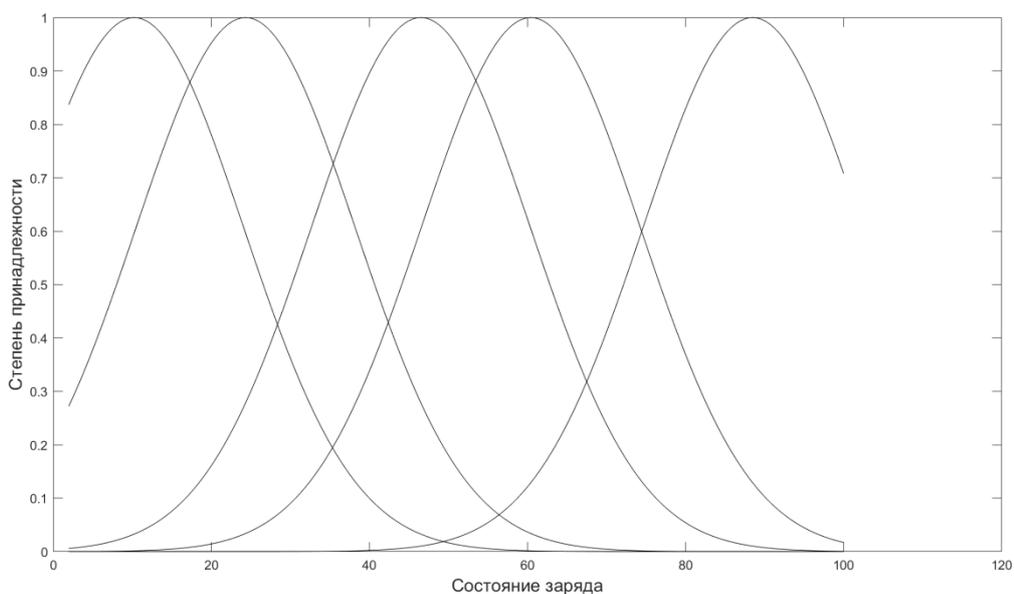


Рисунок 4.14. Функции принадлежности для состояния заряда, получаемые методом на основе субтрактивной кластеризации

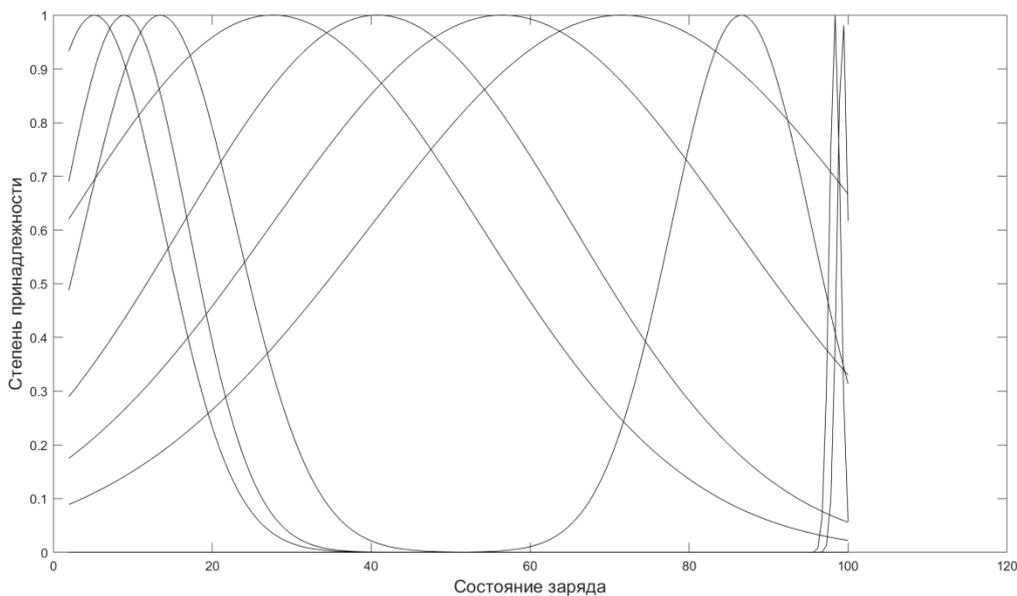


Рисунок 4.15. Функции принадлежности для состояния заряда, получаемые предлагаемым методом

4.3 Сравнение предлагаемого метода построения нечеткой модели и его модификации, использующей метод роя частиц

На рисунке 4.16 представлены зависимости среднеквадратичной ошибки нечеткой системы для прогнозирования состояния заряда по току и

напряжению, построенной предлагаемым методом и его модификацией (см. раздел 2.3.2), от количества правил для обучающей выборки, на рисунке 4.17 представлена та же зависимость для тестовой выборки.

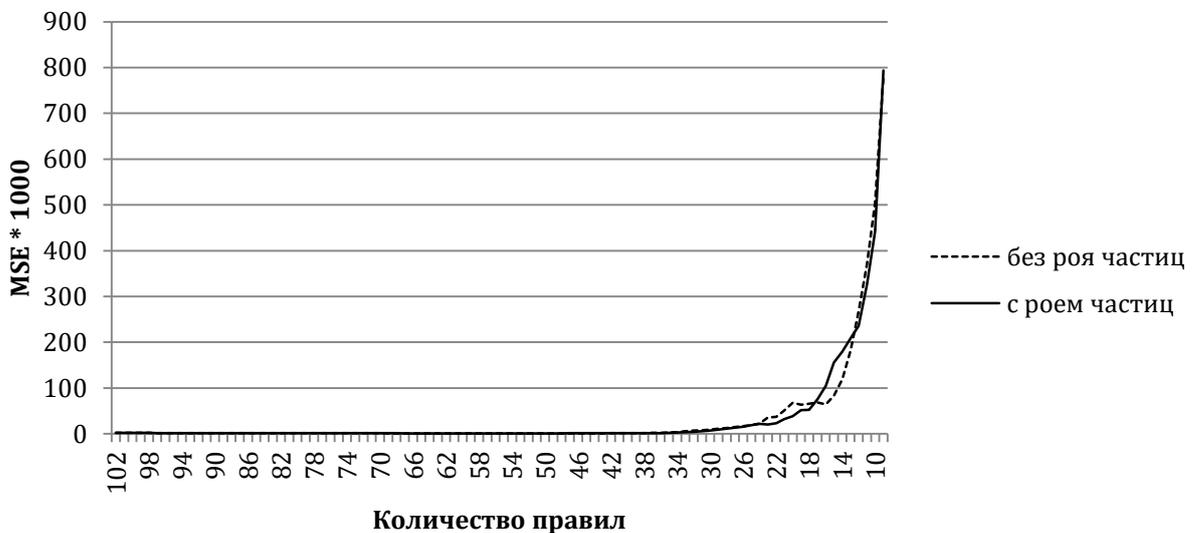


Рисунок 4.16. Среднеквадратичная ошибка нечетких систем, построенных предлагаемым методом и его модификацией, для обучающей выборки



Рисунок 4.17. Среднеквадратичная ошибка нечетких систем, построенных предлагаемым методом и его модификацией, для тестовой выборки
 Таким образом, применение роя частиц не дало улучшений.

4.4 Результаты прогнозирования состояния

В таблице 4.2 приведены результаты прогнозирования напряжения и состояния заряда аккумулятора, полученные предлагаемым методом.

Таблица 4.2. Результаты прогнозирования напряжения и состояния заряда аккумулятора

<i>dt</i>	MSE для обучающей выборки (12 А)		MSE для обучающей выборки (25 А)		MSE для обучающей выборки (50 А)		MSE для обучающей выборки (62 А)		MSE для тестовой выборки (37 А)	
	Напряжение	Состояние заряда	Напряжение	Состояние заряда						
1	0.0258	2793	0.0248	508.5	0.0047	15.48	0.0047	19.21	0.0788	2754
3	0.0161	647.6	0.0078	23.2	0.0022	1.536	0.0021	0.792	0.0384	1878
5	0.0127	208.3	0.0055	9.057	0.0024	0.119	0.0028	0.719	0.0122	109.3
7	0.0113	113.2	0.0034	2.748	0.0068	2.411	0.0072	1.605	0.0201	151.2
10	0.0099	48.76	0.0066	3.528	0.0048	0.089	0.0043	0.064	0.0049	19.48
12	0.0052	26.63	0.0048	1.532	0.0067	0.1341	0.0124	0.159	0.0077	26.64
14	0.0090	22.39	0.0061	0.892	0.0169	0.2116	0.0196	0.002	0.0093	23.70

4.5 Выводы

Нечеткая система, построенная предлагаемым методом, имеет меньшую максимальную среднеквадратичную ошибку, чем нечеткие системы, построенные аналогами. При уменьшении количества правил среднеквадратичная ошибка нечеткой системы, построенной предлагаемым методом, растет экспоненциально для обучающей выборки и изменяется случайным образом для тестовой, что может быть связано с недостаточными размерами обучающей и тестовой выборок. Среднеквадратичная ошибка нечетких моделей прогнозирования состояния аккумулятора, построенных предлагаемым методом, уменьшается с увеличением шага взятия значений переменных по времени.

Заключение

В ходе выполнения выпускной квалификационной работы разработан метод выделения нечетких продукционных правил изменения состояния системы на основе субтрактивной кластеризации.

Проведен анализ существующих методов прогнозирования состояния системы на примере аккумуляторов, анализ существующих методов автоматического построения нечетких моделей на основе данных наблюдений. Выявлено, что лучшими являются методы построения нечетких моделей на основе кластеризации.

Разработан метод, устраняющий недостаток субтрактивной кластеризации, заключающийся в формировании кластеров одинакового размера. Метод заключается в выделении множества маленьких кластеров и соответственно множества правил и дальнейшей редукции правил. Предложен алгоритм редукции правил.

Разработанный метод реализован в программном обеспечении.

Для задачи определения изменения состояния аккумулятора проведено сравнение точности нечетких систем, построенных предлагаемым методом и аналогами: нечеткая система, построенная предлагаемым методом, имеет меньшую максимальную среднеквадратичную ошибку, чем нечеткие системы, построенные аналогами.

Список использованных источников

1. Борисевич А.В. Моделирование литий-ионных аккумуляторов для систем управления батареями: обзор текущего состояния // Современная техника и технологии. 2014. № 5
2. Chizmadzhev Ju. A., Markin V. S., Chirkov Ju. G. Makrokinetika processov v poristyh sredah [The macrokinetics of processes in porous media], Moscow, Nauka Publ., 1971 (in Russian).
3. Siniard K., Xiao M., Choe S-Y. J. Power Sources, 2010, vol. 195, pp. 7102.
4. Venkatraman M., Van Zee J. W. J. Power Sources, 2007, vol. 166, pp.537.
5. Zavalis T.G., Behm M., Lindbergh G. J. Electrochem. Soc., 2012, vol. 159, pp. A848.
6. Boovaragavan V., Methakar R. N., Ramadesiga V., Subramanian V. R. J. Electrochem. Soc., 2009, vol. 156, pp. A854.
7. Albertus P., Coutts J., Srinivasan V., Newman J. J. Power Sources, 2008, vol. 183, pp. 771.
8. Johansen J. F., Farrell T. W., Please C. P. J. Power Sources, 2006, vol. 156, pp. 645
9. Soltani M., Telmoudi A.J., Belgacem Y.B. et al. Parameters identification and discharge capacity prediction of Nickel–Metal Hydride battery based on modified fuzzy c-regression models // Neural Comput & Applic. 2020. V. 32. P. 11361–11371. DOI:10.1007/s00521-019-04631-w
10. Yuanjun Guo, Zhile Yang, Kailong Liu, Yanhui Zhang, Wei Feng. A compact and optimized neural network approach for battery state-of-charge estimation of energy storage system // Energy. 2021. V. 219. 119529. ISSN 0360-5442. DOI: 10.1016/j.energy.2020.119529
11. Fleischer C, Waag W, Bai Z, Uwe Sauer D. On-line self-learning time forward voltage prognosis for lithium-ion batteries using adaptive neuro-fuzzy inference system // Journal of Power Sources. 2013. V. 243. P. 728-749.
12. Химические источники тока: Справочник / Под ред. Н. В. Коровина и А. М. Скундина. – М.: Издательство МЭИ, 2003. 740 с., ил
13. World secondary lithium-ion battery market //Adv. Battery Technol.2008.V.44. N4.p. 13.
14. Смирнов С.Е., Пуцылов И.А., Смирнов С.С. Твердофазные литиевые источники тока. М. 2010. Изд-во «Компания Спутник+».77 с.
15. Колосницын В.С., Карасева Е.В., Кузьмина Е.В. Новые направления развития электрохимических накопителей энергии// XIX Менделеевский съезд по общей и прикладной химии. Волгоград. 2011. Изд. ВГТУ. Тезисы докладов. Т. 4.С. 28.
16. Бертуланфи, Л. История и статус общей теории систем / Л. Бертуланфи // Системные исследования. Ежегодник, 1973. - М. : Наука, 1974.
17. Садовский, Н.В. Основания общей теории систем. Логикометодологический анализ / Н.В. Садовский. – М. : Наука, 1974.

18. Абовский, Н.П. Творчество: системный подход, законы развития, принятие решений / Н.П. Абовский. – М. : СИНТЕГ, 1998. – (Информатизация России на пороге XXI века).
19. Месарович М., Такахара Я. Общая теория систем: математические основы. – М.: Мир, 1978
20. Заде Л., Дезоер Ч. Теория линейных систем. Метод пространства состояний. М.: Наука, 1970
21. Калман Р., Фалб П., Арбиб М. Очерки по математической теории систем. Пер. с англ. / Под ред. Я. З. Цыпкина. Предисл. Э. Л. Наппельбаума. Изд. 2-е, стереотипное. — М.: Едиториал УРСС, 2004. - 400 с.
22. Малышенко А.М. Математические основы теории систем: учебник для вузов/ А.М. Малышенко.- Томск: Изд-во Томского политехнического университета, 2008.- 364 с.
23. Тихонов Э.Е. Прогнозирование в условиях рынка. Невинномысск, 2006. 221 с.
24. Armstrong J.S. Forecasting for Marketing // Quantitative Methods in Marketing. London: International Thompson Business Press, 1999. P. 92 – 119.
25. Чучуева И.А. Модель прогнозирования временных рядов по выборке максимального подобия, диссертация... канд. тех. наук / Московский государственный технический университет им. Н.Э. Баумана. Москва, 2012.
26. Achaibou N., Haddadi M., Malek A. // J. Power Sources. 2008. Vol. 185. P. 1484–1491.
27. Michelis De, Ferella F., Karakaya E., Beolchini F., Vegli'o F. // J. Power Sources. 2007. Vol. 172. P. 975–983.
28. Schiffer J., Sauer D. U., Bindner H., Cronin T. Lundsager P., Kaiser R. // J. Power Sources. 2007. Vol. 168. P. 66–78.
29. Hausmann A., Depcik C. // J. Power Sources. 2013. Vol. 235. P. 148–158.
30. Takagi T, Sugeno M. Fuzzy identification of systems and its application to modeling and control // IEEE Trans. Systems, Man and Cybernetics. — 1985. — Vol. 15. — P. 116—132.
31. Mamdani, E.H., and S. Assilian. 'An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller'. International Journal of Man-Machine Studies 7, no. 1 (January 1975): 1–13, doi: 10.1016/S0020-7373(75)80002-2.
32. Ходашинский Илья Александрович Идентификация нечетких систем: методы и алгоритмы // Проблемы управления. 2009. №4. URL: <https://cyberleninka.ru/article/n/identifikatsiya-nechetkih-sistem-metody-i-algoritmy> (дата обращения: 29.05.2021).
33. C. Juang, S. Chiu and S. Shiu, "Fuzzy System Learned Through Fuzzy Clustering and Support Vector Machine for Human Skin Color Segmentation," in IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, vol. 37, no. 6, pp. 1077-1087, Nov. 2007, doi: 10.1109/TSMCA.2007.904579.

34. R.J. Hathaway, J.C. Bezdek "Switching regression models and fuzzy clustering", IEEE Trans. Fuzzy Syst., 1 (3) (1993), pp. 195-204
35. D. E. Gustafson and W. C. Kessel, "Fuzzy clustering with fuzzy covariance matrix", Proc. IEEE CDC, pp. 761-766, 1979.
36. I. Gath and A. B. Geva, "Unsupervised optimal fuzzy clustering", IEEE Trans. Pattern Anal. Machine Intell., vol. 7, pp. 773-781, 1989.
37. Chiu S. L. Extracting fuzzy rules for pattern classification by cluster estimation //Proc. IFSA. – 1995. – Т. 95. – С. 273-276.
38. R. R. Yager and D. P. Filev, "Approximate clustering via the mountain method," in IEEE Transactions on Systems, Man, and Cybernetics, vol. 24, no. 8, pp. 1279-1284, Aug. 1994, doi: 10.1109/21.299710.
39. Dunn J.C. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters // Journal of Cybernetics. — 1973. — 17 сентября (т. 3, № 3). — С. 32–57. — ISSN 0022-0280. — doi:10.1080/01969727308546046
40. Soltani M. et al. Takagi-Sugeno fuzzy model parameters identification based on fuzzy c-regression model clustering algorithm and particle swarm optimization // 16th IEEE Mediterranean Electrotechnical Conference. 2012. P. 1059-1062.
41. K. M. Alexiev and O. I. Georgieva, "Improved fuzzy clustering for identification of Takagi-Sugeno model," 2004 2nd International IEEE Conference on 'Intelligent Systems'. Proceedings (IEEE Cat. No.04EX791), 2004, pp. 213-218 Vol.1, doi: 10.1109/IS.2004.1344669.
42. J. Illingworth, J. Kittler, A Survey of the Hough Transform, Computer vision, graphics and image processing, vol.44, pp. 87-116, 1988.
43. P.V.C. Hough, "Method and means for recognizing complex patterns", U.S. Patent 3,069,654. Dec. 1962.
44. Babuska R. Fuzzy Modeling for Control, Kluwer Academic Publishers. Boston, USA. 1998.
45. Kaymak U. and Babuska R. Compatible cluster merging for fuzzy modelling // Proceedings of 1995 IEEE International Conference on Fuzzy Systems. 1995. V. 2. P. 897-904. DOI: 10.1109/FUZZY.1995.409789
46. Алтухов А.В. Формирование базы знаний нечеткой системы на основе нечеткой кластеризации // Системный анализ и информационные технологии, № 1, 2008.
47. J. Abonyi, R. Babuska and F. Szeifert, "Modified Gath-Geva fuzzy clustering for identification of Takagi-Sugeno fuzzy models," in IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 32, no. 5, pp. 612-621, Oct. 2002, doi: 10.1109/TSMCB.2002.1033180.
48. Euntai Kim, Minkee Park, Seunghwan Ji and Mignon Park, "A new approach to fuzzy modeling," in IEEE Transactions on Fuzzy Systems, vol. 5, no. 3, pp. 328-337, Aug. 1997, doi: 10.1109/91.618271.
49. Егоров Александр Вадимович, Куприянова Наталия Игоревна Особенности методов кластеризации данных // Известия ЮФУ. Технические науки. 2011. №11. URL:

<https://cyberleninka.ru/article/n/osobennosti-metodov-klasterizatsii-dannyh>
(дата обращения: 29.05.2021).

50. Horng-Lin Shieh, Robust validity index for a modified subtractive clustering algorithm, *Applied Soft Computing*, Volume 22, 2014, Pages 47-59, ISSN 1568-4946, doi: 10.1016/j.asoc.2014.05.001
51. A. Khosla, S. Kumar and K. R. Ghosh, "A Comparison of Computational Efforts between Particle Swarm Optimization and Genetic Algorithm for Identification of Fuzzy Models," *NAFIPS 2007 - 2007 Annual Meeting of the North American Fuzzy Information Processing Society*, 2007, pp. 245-250, doi: 10.1109/NAFIPS.2007.383845.
52. Ходашинский И.А. Основанные на производных и метаэвристические методы идентификации параметров нечетких моделей / И.А. Ходашинский, В.Ю. Гнездилова, П.А. Дудин, А.В. Лавыгина // Труды VIII международной конференции «Идентификация систем и задачи управления» SICPRO '08. М: Институт проблем управления им. В.А. Трапезникова РАН. 2009. С. 501–529.
53. Guillaume S. Designing fuzzy inference systems from data: An Interpret ability-oriented review // *Fuzzy Systems*, *IEEE Transactions*. 2001. V. 9. P. 426 - 443. DOI:10.1109/91.928739
54. Ходашинский И.А., Сарин К.С., Черепанов С. А. Об одном методе инициализации нечетких систем типа Такаги-Сугено // *Автометрия*. – 2016. – Том 52, № 2. – С. 61-70
55. Shun-Hung T., Yu-Wen Ch. A novel identification method for Takagi–Sugeno fuzzy model // *Fuzzy Sets and Systems*. 2018. V. 338. P. 117-135. ISSN 0165-0114, DOI:10.1016/j.fss.2017.10.012
56. Kalman R.E. A New Approach to Linear Filtering and Prediction Problems // *Journal of Basic Engineering*. 1960. V. 82, N. 1. P. 35-45.
57. Jang J.-S.R. ANFIS: adaptive network based fuzzy inference system // *IEEE Transaction Systems, Man and Cybernetics*. 1993. V. 23, N. 3. P. 665- 684.
58. Junhong Y., Jizhen L., Xiangjie L. and Wen T., Identification of Nonlinear System Based on ANFIS with Subtractive Clustering // *2006 6th World Congress on Intelligent Control and Automation*. 2006. P. 1852-1856. DOI: 10.1109/WCICA.2006.1712675.
59. Benmouiza, K., Cheknane, A. Clustered ANFIS network using fuzzy c-means, subtractive clustering, and grid partitioning for hourly solar radiation forecasting // *Theor Appl Climatol*. 2019. V. 137. P. 31–43 DOI:10.1007/s00704-018-2576-4

Приложение А

Граф вызовов функций:

