УДК 004.896:004.032.26:62-503.57

АНАЛИЗ АЛГОРИТМОВ ОБУЧЕНИЯ НЕЙРОННОЙ СЕТИ

[1]Кобзев А.А., [2]Лекарева А.В., [3]Сидорова О.С.

[1]ФГБОУ «Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых», Владимир, e-mail: kobzev42@mail.ru;

[2]ООО «ФС Сервис», Владимир, e-mail: tasya671@rambler.ru;

[3]ИП «Сидорова Оксана Сергеевна», Владимир, e-mail: sidorovao1994@mail.ru

Современный этап развития и проектирования систем управления характеризуется сложностью объектов и технологических процессов управления, непосредственно систем управления и неопределенностью возмущений. При этом установление аналитических зависимостей алгоритмов контуров адаптации не всегда возможно. В этой связи все большее применение получают контуры управления и регуляторы и контуры адаптации, построенные на основе нейронных сетей. Одним из основных вопросов при построении НС является процедура ее обучения. Здесь возможны два подхода: 1) предварительное обучение на симуляторе системы управления и возмущения; 2) в составе непосредственно системы управления, как правило, на основе моделей. В работе анализируются алгоритмы обучения нейронных систем в функции ошибки на основе градиентных методов первого порядка с различными функциями активации. Рассматривались следующие алгоритмы обучения: Backpropagation (MFE); quickProp; Rprop; Nesterov Accelerated Gradient (NAG); AdaDelta; Adam; NAdam при различных значениях коэффициентов вычислительных процедур. Точность работы алгоритмов оценивалась по абсолютной ошибке аппроксимации рассматриваемых функций в режиме онлайн-обучения нейронной сети при её стационарных параметрах, а также в составе адаптивной САУ. Три функции: сигмоидальная, SoftPlusy, ReLU – рассматривались в качестве функций активации в скрытых слоях нейронной сети. Анализ ведется для характерного управляющего и возмущающего воздействия систем автоматического управления – синусоидального сигнала. Даются рекомендации по выбору алгоритмов и функций активации.

Ключевые слова: нейронная сеть, алгоритм, обучение, ошибка, функция активации

ANALYSIS OF NEURAL NETWORK TRAINING ALGORITHMS

[1]Kobzev A.A., [2]Lekareva O.V., [3]Sidorova O.S.

[1]Vladimir State University named after Alexander Grigorevich and Nikolai Grigorevich Stoletovs, Vladimir, e-mail: kobzev42@ mail.ru;

[2] AO «Computer Technologies», Vladimir, e-mail: tasya671@mail.ru;

[3]IP Sidorova O.S., Vladimir, e-mail: sidorovao1994@mail.ru

The current stage of development and design of control systems is characterized by the complexity of objects and technological processes of control, control systems themselves and the uncertainty of disturbances. At the same time, it is not always possible to establish the analytical dependencies of the algorithms of the adaptation contours. In this regard, control loops and regulators and adaptation loops built on the basis of neural networks are increasingly used. One of the main issues in the construction of the NS is the procedure for its training. There are two possible hikes here: 1) preliminary training on the simulator of the control system and perturbation; 2) as part of the control system itself, usually based on models. The paper analyzes algorithms for training neural systems in the error function based on first-order gradient methods with various activation functions. The following learning algorithms were considered: Backpropagation(MFE); quickProp; Rprop; Nesterov Accelerated Gradient (NAG); AdaDelta; Adam; NAdam for different values of coefficients of computational procedures. The accuracy of the algorithms was estimated by the absolute error of the approximation of the functions under consideration in the online training mode of the neural network with its stationary parameters, as well as in the adaptive ACS. Three functions were considered as activation functions in the hidden layers of the neural network: sigmoid, SoftPlusy, and ReLU. The analysis is carried out for the characteristic control and disturbing influence of automatic control systems – a sinusoidal signal. Recommendations on the choice of algorithms and activation functions are given.

Keywords: neural network, algorithm, training, error, activation function

Нейронные сети (НС) получают все большее применение в различных системах управления, сбора и обработки информации, принятия решений и др. Характерные области применения, реализации и функции, выполняемые НС: 1) оптимальный фильтр объекта управления; 2) регулятор; 3) модель объекта управления; 4) комбинированный регулятор – регулятор типа П, ПИ, ПИД в сочетании с регулятором с нечеткой логикой; 5) регуляторы другого типа; 6) распознаватель или классификатор образов; 7) модуль принятия решений. Преимущества контроллеров, построенных с применением НС, в таких системах определяются следующими факторами: 1) быстродействие; 2) универсальность; 3) обучаемость; 4) отказоустойчивость; 5) простота применения.

В контексте нейронной сети обучение рассматривается как процесс настройки весов связей между нейронами из условия минимизации требуемого параметра оптимизируемой системы или процесса управления. По закону изменения параметров сети методы обучения делятся на детерминированные методы и стохастические. Первые основаны на коррекции параметров сети по текущим характеристикам величин входных, фактических и желаемых выходных сигналов. В классе детерминированных методов выделяются следующие основные подклассы в части алгоритмов

обучения [1–3]: 1) по правилу Хэбба и Хопфилда; 2) методом выстраивания показателей; 3) коррекции по ошибке (желаемый вход-выход для всех ситуаций) и др.

Цель исследования: провести анализ алгоритмов настройки нейронной сети на основе градиентных методов первого порядка. При этом оценить влияние различных функций активации на показатели процесса настройки. Процесс настройки анализируется для типового входного воздействия систем автоматического управления – гармоническом входном сигнале.

Материалы и методы исследования

Методы обучения нейронных сетей. Рассмотрим аналитическое представление градиентных алгоритмов первого порядка, подлежащих анализу эффективности процесса обучения [4–6]. Эти алгоритмы основаны на коррекции параметров нейронной сети в функции градиента. В эту группу алгоритмов входят: метод градиентного спуска, метод моментов с регуляризацией, метод quickProp, метод rProp, метод сопряженных градиентов, метод NAG, метод AdaGrad (AdaptiveGradient), метод AdaDelta, метод Adam.

Градиентный метод первого порядка. Общий алгоритм обучения, реализуемый градиентными метода первого порядка, предусматривает следующую последовательность процедур.

1. Инициализация весов нейронной сети W.

2. Вычисление текущей ошибки $E(h(X, W), C)$.

3. Если значение ошибки находится в допустимом диапазоне, то коррекция параметров сети не требуется – конец работы.

4. Вычисление значения градиента функции потери: $\Delta * E(h(X,W),C)$, здесь и далее $\Delta *$ – градиент функции.

5. Вычисление изменения параметров: $\Delta W_t = \eta \Delta * E$.

7. Коррекция параметров сети $W_t = W_{t-1} - \Delta W_t$. Здесь и далее индекс «$t$» обозначает текущую итерацию, индекс «t-1» – предыдущую.

8. Переход на п. 2.

Параметр η (скорость обучения) определяет величину шага процесса оптимизации, значение данного параметра находится в диапазоне $0 < \eta < 1$.

Согласно п. 4 определяем градиент функции потери по выражению

$$\Delta * (W) = \left[\frac{\partial E}{\partial w_1} ..., \frac{\partial E}{\partial w_k} \right],$$

где k – общее количество весов сети.

Составляющие определяются

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_i} \frac{\partial y_j}{\partial s_j} \frac{\partial s_j}{\partial w_{ij}},$$

где E – функция потери; w_{ij} – вес связи нейронов i и j; y_j – выход нейрона j; s_j – состояние нейрона j.

Ошибки определены только для нейронов выходного слоя. Ошибки в скрытых и выходном слое соответственно определяются:

– для выходного слоя $\delta_i = \frac{\partial E}{\partial y_i}$;

– для скрытого слоя $\delta_i = \frac{\partial y_i}{\partial s_i} \sum_i \delta_i w_{ij}$.

Совокупность процедур определения градиента функции потерь методом обратного распространения ошибки:

1) вычисление состояния нейронов s всех слоев сети – прямой проход;

2) определение $\delta_i = \partial E / \partial y_i$ для выходного слоя;

3) вычисление для скрытых слоев δ_i в обратном порядке – обратный проход;

4) определение $\Delta * E = \partial E / \partial W = y \delta^T$ для каждого слоя и вычисление.

Метод обучения на основе обратного распространения ошибки (Backpropagation) предусматривает базовую последовательность процедур с учетом алгоритма обратного распространения ошибки.

На основе базового метода разработаны его модификации, состоящие в коррекции поправок при вычислении поправки ΔW_t. Ниже приведем только результирующие выражения [4].

Метод моментов с регуляризацией. Классический метод градиентного спуска может «застревать» в локальных минимумах функции потери E, для предотвращения данных событий, широкое распространение получила модификация данного метода с использованием стратегии mini-batch и «моментов». Формально это описывается добавлением слагаемого к изменению весов:

$$\Delta W_t = \eta \Delta * E + \Delta W_{t-1},$$

где μ – коэффициент момента.

Одна из модификаций метода состоит в применении регуляризации, которая для борьбы с переобучением налагает штраф на чрезмерный рост значений весов:

$$\Delta W_t = \eta (\Delta * E + \rho W_{t-1}) + \mu W_{t-1},$$

где ρ – коэффициент регуляризации.

Для увеличения скорости сходимости процесса обучения можно ввести адаптивный коэффициент обучения, изменяемый

на каждой итерации t в зависимости от изменения ошибки E.

Метод quick Prop. Отличие данного метода от рассмотренного выше состоит в том, что параметр момента μ и коэффициент скорости обучения η задаются индивидуально для каждого параметра. Изменение параметров описывается соотношением

$$\Delta W_t = \eta(\Delta * E + \rho W_{t-1}) + \mu W_{t-1}.$$

Метод r Prop. Является модификацией рассмотренного выше quickProp, в которой применяется стратегия *full-batch*. При этом параметр скорости обучения η рассчитывается для каждого веса индивидуально. Изменение параметров весов определяется соотношением

$$\Delta W_t = \eta(sign(\Delta * E) + \rho W_{t-1}) + \mu W_{t-1}.$$

Метод сопряженных градиентов. Основан на специальном выборе направления изменения параметров, являющимся ортогональным к предыдущим направлениям. Изменение весов в этом случае имеет вид

$$\Delta W_t = \eta(p + \mu W_{t-1}) + \mu W_{t-1}.$$

Коэффициент скорости обучения η, направление изменения параметров p, коэффициент сопряжения β вычисляются на каждом шаге путем решения задачи оптимизации:

$$\min_\eta E(\Delta W(\eta)).$$

Для компенсации накапливающейся погрешности предусмотрен сброс сопряженного направления, т.е. $\beta = 0$, $p = \Delta * E$ через каждые n циклов, число которых выбирается в зависимости от количества параметров сети.

Метод Nesterov's Accelerated Gradient (NAG). Здесь градиент вычисляется относительно сдвинутых на значение момента весов

$$\Delta W_t = \eta(\Delta * E(W_{t-1} + W_{t-1}) + \rho W_{t-1}) + \mu \Delta W_{t-1}.$$

Метод Adaptive Gradient (AdaGrad). В группе адаптивных оптимизационных алгоритмов (Adagrad, RMSProp, Adadelta, Adam, NAdam) реализована динамическая модификация скорости обучения. Обновления производятся для значений признаков, представленных в меньшинстве, а более слабые обновления – для часто встречаемых значений. Этот принцип реализуется за счет того, что скорость обучения здесь фактически вычисляется отдельно для каждого из параметров на каждом шаге/такте обучения. При этом учитывается история значений градиентов g_t. Выражение для изменения весов имеет вид

$$\Delta W_t = \eta(g_t + \rho W_{t-1}) + \mu \Delta W_{t-1}.$$

Метод AdaDelta. Является модификацией метода Adagrad и также учитывает историю значений градиента и историю изменения весов, однако при этом вместо полной суммы обновлений используется усреднённый по истории квадрат градиента (как экспоненциально затухающее бегущее среднее). Изменение весов аналогично.

Метод Adam (adaptive moment stimation). Сочетает в себе и идею накопления движения, и идею более слабого обновления весов для типичных признаков. Здесь используются «свои» аналитические выражения для коррекции градиента ошибки. Изменение весов аналогично.

Метод N Adam. Данный метод представляет собой модификацию метода Adam. Предусматривает коррекцию параметра учета истории значений градиентов g_t.

Функции активации. Выходной сигнал нейрона определяется непосредственно видом функции активации. Наибольшее распространение получила функция активации в виде логистического сигмоида, обладающая всеми свойствами, необходимыми для нелинейности в нейронной сети: ограниченность (стремление к нулю при $x \to -\infty$ и к единице при $x \to \infty$), дифференцируемость на всём диапазоне определения, малые вычислительные затраты на определение производной. Однако эксперименты Глоро и Бенджи с глубокими сетями с функцией активации в виде сигмоида, показали, что последний уровень сети очень быстро насыщается, и преодолеть эту ситуацию насыщения очень сложно [7].

Еще одной широко распространённой функцией активации является гиперболический тангенс. В отличие от сигмоида, функция гиперболического тангенса имеет более «крутые» характеристики в части нарастания и убывания выходного значения. При этом значение аргумента равное нулю является самой нестабильной промежуточной точкой, т.е. можно легко оттолкнуться от нуля и начать менять аргумент в любую сторону. Данный вид функции активации очень часто используется в области компьютерного зрения. Однако такие функции активации характеризуются недостаточно точным отражением состояния нейрона, т.е., по сути, они дают бинарный выходной сигнал, например активация нейрона «с силой 5» (для сигмоида выходное значение будет 0,9933) слабо отличается от активации «с силой 10» (выходное значение 0,99995). Позднее были разработаны такие функции, как логарифмическая и ReLU. Данные функции имеют сходные выходные характеристики. Однако для вычисления производной функции ReLU требуется

лишь одно сравнение, то есть ReLU-сети при одних и тех же вычислительных затратах на обучение могут быть значительно больше по размеру.

Дальнейшее развитие этого направления – различные модификации и обобщения функции ReLU,– Leaky ReLU, Parameterized ReLU, ELU.

Результаты исследования и их обсуждение

Рассматривались следующие алгоритмы обучения: Backpropagation (MFE); quickProp; Rprop; Nesterov Accelerated Gradient (NAG); AdaDelta; Adam; NAdam. Точность работы алгоритмов оценивалась по абсолютной ошибке аппроксимации рассматриваемых функций в режиме онлайн-обучения нейронной сети при её стационарных параметрах, а также в составе адаптивной САУ [6–8]. В качестве параметров нейронной сети установлены: 1) количество слоев нейронной сети – 4; 2) количество нейронов в 1-м слое – 2; 3) количество нейронов в скрытых слоях – 10, 15 соответственно; 4) количество нейронов в выходном слое – 1; 5) дискретизация сети – 0,01 с; 6) функция активации в выходном слое – линейная. В качестве функций активации в скрытых слоях нейронной сети рассматривались три функции:

$$y = 1/(1 + \exp(-s));$$

$$SoftPlusy = \log(1 + \exp(s));$$

ReLU if(s> = 0) y = s; elsey = 0.

В таблице представлены основные параметры алгоритмов обучения, используемые в процессе исследования.

На рис. 1–3 представлены ошибки обучения при аппроксимации функции вида $y = A\sin(\omega t + \varphi)$ и различных функциях активации.

Параметры алгоритмов обучения

Алгоритм обучения	Сигмоидальная функция активации	SoftPlus	ReLU
Backpropagation	$\eta = 0{,}5$; $\mu = 0{,}1$; $\rho = 0$	$\eta = 0{,}5$; $\mu = 0{,}1$; $\rho = 0$	$\eta = 0{,}5$; $\mu = 0{,}1$; $\rho = 0$
quickProp	$\eta = 0{,}8$; $\mu = 0$; $\rho = 0$	не стабилен	$\eta = 0{,}77$; $\mu = 0$; $\rho = 0$
Rprop	$\eta = 0{,}5$;a = 1,01; b = 0,3 $\mu = 0$; $\rho = 0$	$\eta = 0{,}5$;a = 1,01; b = 0,3 $\mu = 0$; $\rho = 0$	$\eta = 0{,}5$;a = 1,01; b = 0,3 $\mu = 0$; $\rho = 0$
NAG	$\eta = 0{,}5$; m = 0,3; p = 0,5; $\mu = 0$; $\rho = 0$	$\eta = 0{,}5$; m = 0,3; p = 0,5; $\mu = 0$; $\rho = 0$	$\eta = 0{,}5$; m = 0,3; p = 0,5; $\mu = 0$; $\rho = 0$
RMSprop	$\eta = 0{,}03$; $\alpha = 0{,}4$; $\mu = 0$; $\rho = 0$	$\eta = 0{,}03$; $\alpha = 0{,}2$; $\mu = 0$; $\rho = 0$	$\eta = 0{,}3$; $\alpha = 0{,}2$; $\mu = 0$; $\rho = 0$
AdaDelta	$\eta = 0{,}5$; $\mu = 0$; $\rho = 0$	$\eta = 0{,}5$; $\mu = 0$; $\rho = 0$	$\eta = 0{,}5$; $\mu = 0$; $\rho = 0$
Adam	$\eta = 0{,}7$; $\mu = 0$; $\rho = 0$	$\eta = 0{,}5$; $\mu = 0$; $\rho = 0$	$\eta = 0{,}7$; $\mu = 0$; $\rho = 0$
NAdam	$\eta = 0{,}8$; $\mu = 0$; $\rho = 0$	$\eta = 0{,}8$; $\mu = 0$; $\rho = 0$	$\eta = 0{,}8$; $\mu = 0$; $\rho = 0$

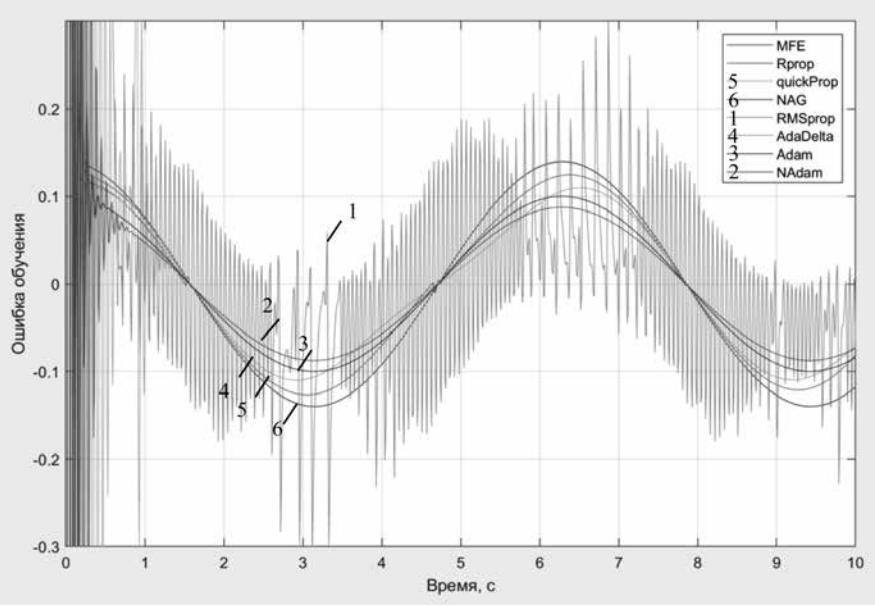

Рис. 1. Ошибки обучения нейронной сети при сигмоидальной функции активации

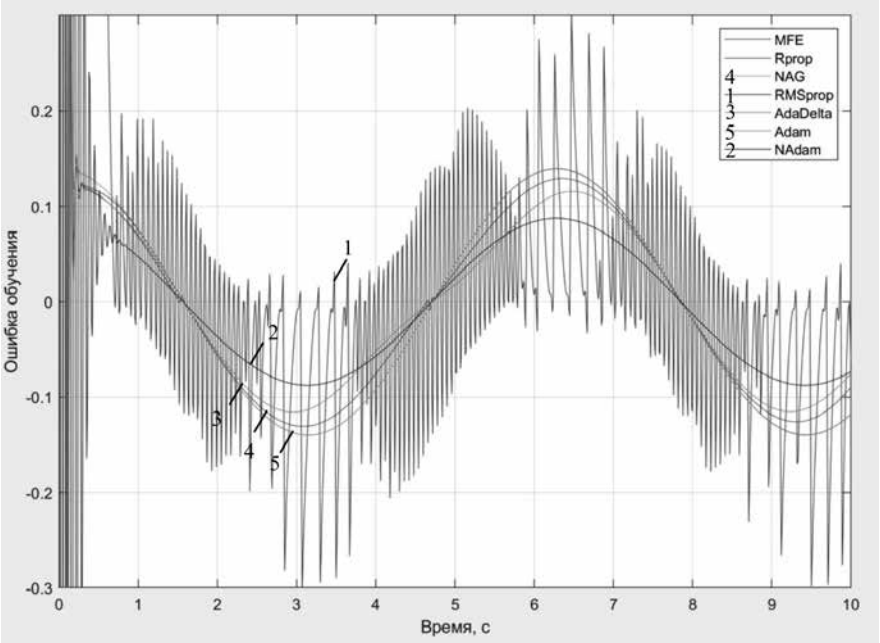

Рис. 2. Ошибки обучения нейронной сети при функции активации SoftPlus

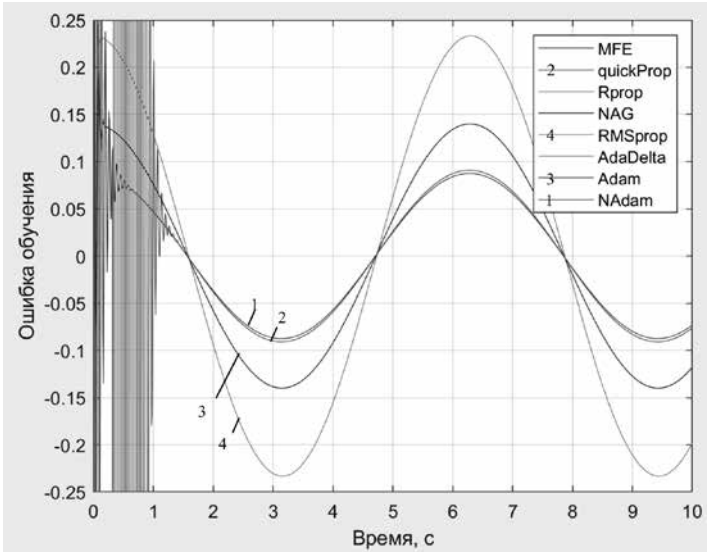

Рис. 3. Ошибки обучения нейронной сети при функции активации ReLU

Заключение

Результаты исследования показали, что рассматриваемые методы обладают примерно одинаковыми точностными характеристиками. Однако метод quickProp при использовании функции активации SoftPlus имел нестабильный характер процесса обучения, при этом варьированием параметров алгоритма не удалось обеспечить сходимость процесса обучения. При использовании метода RMSpropc функциями активации в виде сигмоиды и SoftPlus ошибки обучения имеет колебательный характер. В целом анализ результатов исследования свидетельствует о том, что методы Adam и NAdam с применением ReLU функции активации в скрытых слоях демонстрируют лучшие значения скорости сходимости обучения и меньшую вероятность застревания алгоритма в локальном минимуме, а также меньшие значения ошибки обучения. Наиболее целесообразным является использование метода NAdam.

Список литературы

1. Хайкин С. Нейронные сети. М.: ООО «И.Д. Вильямс», 2017. 1104 с.

2. Еременко Ю.И., Глущенко А.И. О разработке метода выбора структуры нейронной сети для решения задачи адаптации параметров линейных регуляторов // Управление большими системами: сборник трудов. 2016. № 62. С. 75–123.

3. Wu C., Liu J., Jing X., Li H., Wu L. Adaptive Fuzzy Control for Nonlinear Networked Control Systems. IEEE Transactions on Systems, Man, and Cybernetics: Systems. 2017. Vol. 47. № 8. P. 2420–2430.

4. Борисов Е.С. О методах обучения многослойных нейронных сетей. Ч. 2: Градиентные методы первого порядка. 2016. 17 с. [Электронный ресурс]. URL: http://mechanoid.kiev.ua (дата обращения: 22.05.2021).

5. Sheela K.G., Deepa S.N. Review on Methods to Fix Number of Hidden Neurons in Neural Networks. Mathematical Problems in Engineering. 2013. P. 1–11.

6. Каширина И.Л., Демченко М.В. Исследование и сравнительный анализ методов оптимизации, используемых при обучении нейронной сети // Вестник Воронежского государственного университета. 2018. № 4. С. 123–132.

7. Hagan M.T., Demuth H.B. Neural networks for control. Proceedings of the American Control Conference. San Diego, USA, 1999. Vol. 3. P. 1642–1656.

8. Кобзев А.А., Монахов Ю.М., Лекарева А.В. Реализация комплементарной коррекции в системах автоматического управления траекторными перемещениями технологических объектов с использованием нейросетевого регулятора // Динамика сложных систем – XXI век. 2017. Т. 11. № 4. С. 121–130.