

ПРИНЦИПЫ ПОСТРОЕНИЯ НЕЙРОРЕГУЛЯТОРА ДЛЯ УПРАВЛЕНИЯ ПРОМЫШЛЕННЫМ ОБЪЕКТОМ

В.П.Войтенко, М.А.Хоменко

Черниговский государственный технологический университет

Украина, 14027, г. Чернигов, ул. Шевченко, 95, ЧГТУ, кафедра промышленной электроники

Тел. (04622) 316-96-152, E-mail: vvp@inel.stu.cn.ua, shadow@inel.stu.cn.ua

Annotation – The relevant problem of the synthesis of industrial regulator ensuring the management quality in the wide spectrum of tasks and object parameters is formulated. The solution option with the application of artificial neural networks based on the selected parameters and the network learning method is considered. The neural regulator operation is simulated on the DC motor example.

Key words – optimal control, neural systems.

ВВЕДЕНИЕ

При автоматизации технологических процессов используется большое количество замкнутых систем автоматического регулирования, часто образующих иерархическую управляющую структуру. Наиболее популярными остаются ПИ- и ПИД-регуляторы, хотя теоретически проработаны и экспериментально испытаны и более продвинутые алгоритмы [1]. По-прежнему сохраняется актуальность вопроса настройки регулятора, работающего в реальных условиях как широкого варьирования задания на регулирование, так и разброса параметров объектов, их нелинейностей, а также ряда случайных и трудно предсказуемых на стадии проектирования факторов. Несмотря на существенно различающийся смысл коэффициентов настройки для различных классов регуляторов, их объединяет общность требуемого результата. Практически всегда, независимо от свойств объекта регулирования, необходимо обеспечить заданную точность установления регулируемого параметра и переходный процесс без перерегулирования за минимальное время (например, близкий к апериодическому). Разумеется, в каждом конкретном случае могут выдвигаться и другие специфические, но важные с точки зрения технологии требования к качеству регулирования. При решении подобных задач могут оказаться востребованными так называемые интеллектуальные алгоритмы, опирающиеся не только на строгие математические описания, но и на представления о процессе регулирования, успешно выполняемого человеком-оператором.

Применение нейронных сетей в системах управления позволяет повысить качество функционирования систем с нелинейными объектами, а также с объектами, параметры которых изменяются в процессе функционирования. Развитие элементной базы систем управления – микроконтроллеров, программируемых логических интегральных схем, сигнальных процессоров открывает новые возможности в использовании нейросетевых технологий во встраиваемых системах, в первую очередь – в промышленных регуляторах.

Данная статья посвящена оценке использования нейрорегулятора для управления промышленным объектом на примере двигателя постоянного тока (ДПТ).

ВЫБОР ПАРАМЕТРОВ И МЕТОДА ОБУЧЕНИЯ СЕТИ

Для решения задач управления наиболее прием-

лемой является сеть, построенная по многослойной архитектуре [2], [3]. В качестве простейшего с точки зрения требований к вычислительным ресурсам возьмем сеть прямого распространения (статическую), состояние выходов которой зависит только от текущего состояния входов и не зависит от их предыдущего состояния.

Работу j -го нейрона k -го слоя можно описать следующими соотношениями [3]:

$$s_{jk} = \sum_{i=1}^N w_{ijk} x_{ijk} + b_{jk}, \quad (1)$$

$$y_{jk} = f_{jk}(s_{jk}), \quad (2)$$

где i – номер входа;

N – количество входов;

x_{ijk} – сигналы на входах нейрона;

w_{ijk} – весовые коэффициенты;

b_{jk} – коэффициент смещения;

f_{jk} – активационная функция;

s_{jk} – взвешенная сумма;

y_{jk} – сигналы на выходе нейрона.

(1) можно записать в матричной форме:

$$\mathbf{s}_{jk} = \mathbf{W}_{jk} \mathbf{X}_{jk} + \mathbf{b}_{jk}, \quad (3)$$

Активационная функция f_j может иметь различный вид в зависимости от функции, выполняемой искусственным нейроном. Одной из наиболее подходящих для решения задач управления является функция гиперболического тангенса:

$$f(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}}. \quad (4)$$

Эта функция обеспечивает сжатие динамического диапазона входного сигнала и знакопеременна, что способствует более быстрому (с точки зрения количества итераций) обучению сети, использующей нейроны с данной активационной функцией.

Адаптацию нейронной сети к условиям поставленной задачи или обучение целесообразно реализовать на основе метода обучения с учителем [4], что позволяет задать, каким именно образом сеть должна реагировать на входные сигналы. Этот метод обучения предполагает, что для каждого входного вектора существует целевой вектор, представляющий собой требуемый выход. Вместе они образуют обучающие пары, ограниченное множество которых позволяет

обучать сеть. Предъявляется входной вектор, вычисляется выход сети и сравнивается с соответствующим целевым вектором. Разность (ошибка) по цепи обратной связи подается в сеть, а веса изменяются в соответствии с алгоритмом, стремящимся минимизировать ошибку. Векторы обучающего множества предъявляются последовательно, вычисляются ошибки, и веса подстраиваются для каждого вектора до тех пор, пока ошибка по всему обучающему массиву не достигнет приемлемо низкого уровня.

Конкретная реализация метода обучения сети может быть выполнена с помощью алгоритма обратного распространения ошибки, который дает возможность эффективно настраивать многослойные сети прямого распространения. Этот итерационный градиентный алгоритм обучения позволяет минимизировать среднеквадратичное отклонение текущего выхода от желаемого в многослойных нейронных сетях [5].

Целевая функция, подлежащая минимизации, имеет вид [3, 6]:

$$E = \frac{1}{2} \sum_j \sum_m (d_j^m - y_j^m)^2, \quad (5)$$

где m – номер обучающего вектора;

j – номер нейрона в выходном слое;

y_j^m – реальный выход;

d_j^m – желаемый выход.

При обучении на каждой итерации будем корректировать параметры (веса и смещения) в направлении антиградиента E . Такой метод в теории оптимизации получил название метода наискорейшего спуска. В соответствии с данным методом запишем формулу для приращения весовых коэффициентов:

$$\Delta w_{ijk} = -\varepsilon \left(\frac{\partial E(\mathbf{W}, \mathbf{B})}{\partial w_{ijk}} \right), \quad (6)$$

$$w_{ijk}(l+1) = w_{ijk}(l) + \Delta w_{ijk}, \quad (7)$$

где \mathbf{W}, \mathbf{B} – матрицы текущих значений весов и смещений соответственно;

l – номер итерации;

ε – коэффициент скорости обучения.

Аналогичным образом рассчитываются коррекции коэффициентов смещения.

Для выходного слоя запишем компоненты градиента как производную сложной функции.

$$\frac{\partial E}{\partial w_{ijk}} = \frac{\partial E}{\partial y_{jk}} \frac{\partial y_{jk}}{\partial s_{jk}} \frac{\partial s_{jk}}{\partial w_{ijk}}, \quad (8)$$

в данном случае k – номер выходного слоя.

Компоненты градиента по смещению имеют вид:

$$\frac{\partial E}{\partial b_{jk}} = \frac{\partial E}{\partial y_{jk}} \frac{\partial y_{jk}}{\partial s_{jk}} \frac{\partial s_{jk}}{\partial b_{jk}}. \quad (9)$$

Производную функции ошибки для выходного слоя можно записать как:

$$\frac{\partial E}{\partial y_{jk}} = \sum_m (y_{jk}^m - d_{jk}^m). \quad (10)$$

Поскольку в качестве активационной функции

используется гиперболический тангенс, выражение для производной функции активации примет вид:

$$\frac{\partial f(s_{jk})}{\partial s_{jk}} = 1 - f^2(s_{jk}). \quad (11)$$

Из (11) видно, что производная рассчитывается через текущие значения выходов нейронов ($f(s_{jk}) = y_{jk}$).

Так как производная по входному значению x_{ijk} выходного слоя соответствует по смыслу производной по выходу предыдущего слоя, можно записать:

$$\frac{\partial E}{\partial x_{ijk}} = \frac{\partial E}{\partial y_{j(k-1)}}. \quad (12)$$

(12) обеспечивает рекурсивный переход от последующего слоя к предыдущему и является аналогом (10) для предыдущих слоёв.

Таким образом, метод обратного распространения позволяет сократить вычислительные затраты на расчёт градиента. Зная градиент, можно применить различные методы теории оптимизации для обучения сети.

Современная теория искусственных нейронных сетей не дает универсальных рекомендаций для определения количества слоёв и нейронов в каждом слое.

При выборе количества нейронов и слоёв следует помнить, что малое количество нейронов либо не позволит получить желаемую точность, либо сеть вообще не сможет обучиться решать поставленную задачу. Слишком большое количество нейронов приводит к снижению быстродействия, увеличению необходимого объёма памяти и вычислительной мощности. Кроме того, возможна потеря способности к обобщению. В результате сеть будет неадекватно реагировать на входные векторы, отличающиеся от векторов обучающего множества. Опираясь на результаты моделирования, была выбрана структура сети 2-9-6-6-1 (2 входных нейрона, 9-6-6 три скрытых слоя, 1 нейрон в выходном слое). Число входных нейронов соответствует количеству входных переменных, а число выходных – количеству выходных переменных. В качестве входных переменных выбраны ошибка по углу поворота ротора и скорость вращения ротора ДПТ. Выходной переменной является угол поворота ротора.

Активационные функции всех нейронов, кроме выходного, представляют собой гиперболический тангенс, что позволяет эффективно использовать алгоритм обратного распространения ошибки для обучения сети. Активационная функция выходного нейрона линейная. Это дает возможность получать выходную переменную необходимой амплитуды без последующего масштабирования.

На Рис. 1 представлена схема обучения нейроконтроллера, в которой использован эталонный регулятор и модель ДПТ. На вход схемы подаются различные сигналы задания $\theta_s(k)$. Эталонный регулятор в соответствии с сигналом ошибки $e(k)$ вырабатывает управляющее воздействие $u_s(k)$, которое подаётся на вход модели ДПТ, стремясь привести выход модели $\theta_m(k)$ (сигнал об угле поворота ротора) к заданному значению. Входными переменными для нейроконтроллера являются сигнал ошибки и вспомогательная переменная, характеризующая динамику объекта. В

данном случае это – скорость вращения ротора $\omega_m(k)$. Нейроконтроллер в соответствии с входными переменными вырабатывает управляющее воздействие $u_{yHK}(k)$, сравниваемое с выходным сигналом эталонного регулятора. Результатом сравнения является рассогласование $e_{HK}(k)$, которое используется алгоритмом обратного распространения ошибки для настройки весов нейроконтроллера.

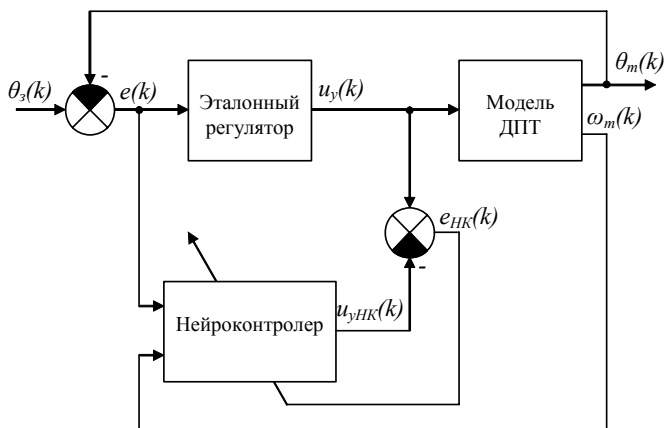


Рис. 1

Следует подчеркнуть, что в данной схеме для обучения нейроконтроллера использована модель ДПТ, а не сам двигатель, что позволяет избежать нежелательных эффектов во время обучения. Параметры модели можно определить, например, с помощью методов идентификации. Это дает возможность использовать полученные коэффициенты передаточной функции двигателя для расчёта эталонного регулятора. Другой – нетривиальный способ – заключается в использовании в качестве модели нейронной сети, обученной моделированию динамики двигателя.

Эталонный регулятор выбирается исходя из требований, предъявляемых к системе. Чаще всего самый лучший регулятор теряет оптимальность при регулировании в широком диапазоне заданий вследствие нелинейностей реальных объектов, что приводит к необходимости перестройки коэффициентов регулятора. Кроме того, продвинутые алгоритмы управления чрезвычайно ресурсозатратны, что затрудняет их практическую реализацию в системах реального времени. Эти два фактора и обуславливают необходимость использования нейроконтроллера как звена системы автоматического управления.

В качестве эталонного регулятора выбран оптимальный регулятор [7, 8], рассчитанный с помощью метода переменного коэффициента усиления [9]. Этот регулятор позволяет получить процессы конечной длительности и без перерегулирования в нагрузке с известными параметрами.

МОДЕЛИРОВАНИЕ НЕЙРОКОНТРОЛЛЕРА

Для обучения нейроконтроллера разработана модель в пакете *Matlab*. Управление работой модели осуществляется программой, которая, в частности, используя полученные во время симуляции данные,

обучает нейроконтроллер по алгоритму обратного распространения ошибки. Кроме того, с ее помощью генерируется *Simulink*-блок нейроконтроллера.

Обучение нейроконтроллера проводилось на модели ДПТ со следующими параметрами: сопротивление обмоток $R=0,7$ Ом; индуктивность обмоток $L=90$ мкГн; коэффициент противоЭДС $C_e=0,0448$ В·с/рад.; момент инерции $J=130 \cdot 10^{-7}$ кг·м². Эталонный регулятор рассчитан таким образом, что управляющее напряжение во всём диапазоне регулирования не превышает 20 В. При этом шаг квантования регулятора составил 14 мс, а время позиционирования – 42 мс (3 шага квантования).

Для оценки качества работы системы управления с нейроконтроллером разработана *Simulink*-модель, представленная на Рис. 2. Блок DCM моделирует работу двигателя постоянного тока. Ключ Sw1 позволяет установить сигнал задания либо постоянным (источник сигнала задания – блок Const1), либо изменяющимся во времени (источник сигнала задания – блок Rand1). Сигнал задания и значение угла поворота поступают на входы сумматора Sum1, на выходе которого образуется сигнал ошибки. Блоки 1/Anglemax, Sw2 обеспечивают возможность масштабирования сигнала ошибки. В цепи обратной связи по скорости находятся блоки Sum4, Const2, Rand2, Sw4, с помощью которых можно имитировать присутствие шумов в сигнале датчика скорости (переключатель Sw4 подключает блок Rand2 на вход сумматора Sum4). Цепь 1/Smax, Sw3 служит для получения масштабированного сигнала скорости. Сигналы ошибки и скорости поступают на входы мультиплексора Mux1, с выхода которого они подаются на вход нейроконтроллера (блок Neural Reg). Нейроконтроллер в соответствии с входными сигналами вырабатывает управляющее воздействие и подает его на вход ДПТ.

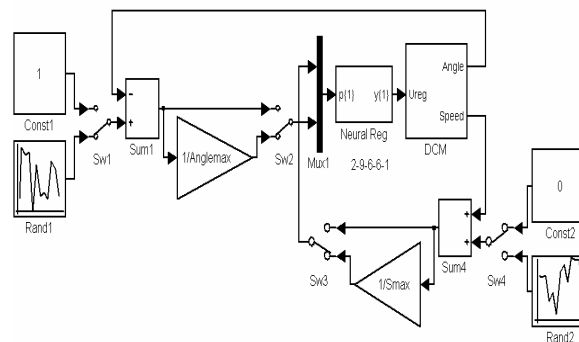


Рис. 2

Результаты моделирования работы системы управления ДПТ на базе обученного нейроконтроллера представлены на Рис. 3. На диаграмме под номером 1 показан сигнал задания, а под номером 2 – реакция системы (угол поворота ротора). Очевидно, нейроконтроллер обеспечивает позиционирование без перерегулирования. Время установления совпадает с обеспечиваемым эталонным регулятором.

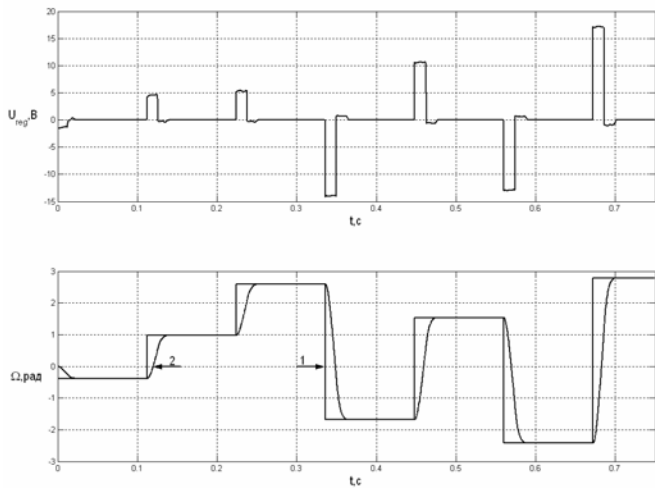


Рис. 3

Рассмотрим случай, когда параметры двигателя отличаются от тех, которые использовались при обучении нейроконтролера.

Выберем максимальное отклонение параметров от нормы равным $+100\%$ и -50% , что соответствует их увеличению или уменьшению в 2 раза. На Рис. 4 (а) приведены диаграммы позиционирования для случая, когда $\beta = \beta + 100\%$, а на Рис. 4 (б) – для случая, когда $\beta = \beta - 50\%$ [8].

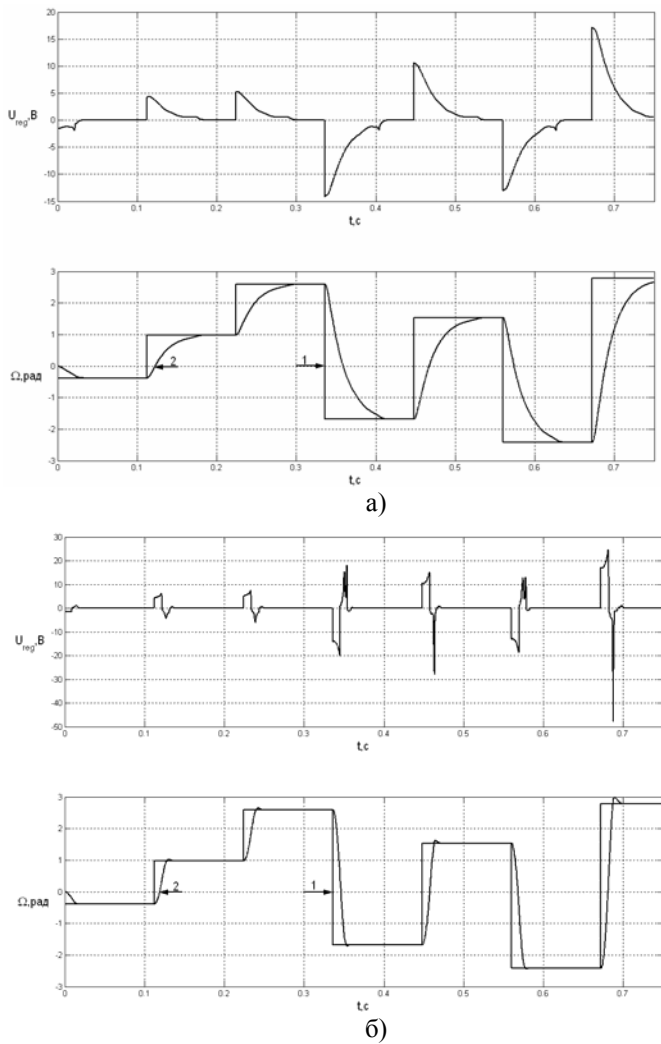


Рис. 4

Увеличение β , эквивалентное уменьшению электромеханической постоянной времени, не влияет на точность позиционирования системы, хотя и увеличивает время позиционирования. Уменьшение параметра β не приводит к затягиванию переходного процесса, однако вызывает ухудшение его качества (появляется перерегулирование). Изменение параметра α [8] приводит к аналогичным результатам, однако, в меньшей степени. Изменение параметра k [8] не ухудшает качество процесса, хотя при уменьшении k в 2 раза увеличивается время позиционирования.

Результаты моделирования и сравнение с поведением системы, содержащей только лишь эталонный регулятор, указывают на то, что введение нейроконтроллера позволяет получить систему, обладающую низкой чувствительностью к изменению параметров двигателя и обладающую всеми достоинствами эталонного регулятора.

ВЫВОДЫ

1. Сформулирована актуальная задача синтеза промышленного регулятора, обеспечивающего качество регулирования в широком диапазоне заданий и параметров объекта.
2. Рассмотрен вариант решения с использованием искусственных нейронных сетей на основе выбранных параметров и метода обучения сети. На примере ДПТ промоделирована работа нейрорегулятора.
3. Полученные путем моделирования результаты позволяют судить о перспективности синтеза высокоэффективных промышленных нейрорегуляторов.

[1] Войтенко В. П. Экспериментальное исследование алгоритмов функционирования квазиоптимального промышленного регулятора// Технічна електродинаміка, тем. випуск “Силова електроніка та енергоефективність”. – Київ–2005, ч. 3. – С. 71 – 74.

[2] Anil K. Jain, Jianchang Mao, K.M. Mohiuddin. Artificial Neural Networks: A Tutorial, Computer, Vol.29, No.3, March/1996, pp. 31-44.

[3] Roland S. Burns Advanced Control Engineering. Linacre House, Jordan Hill, Oxford, – 2001.

[4] Уоссерман Ф. Нейрокомпьютерная техника: Теория и практика. – М.: Мир, 1992. – 237 с.

[5] Дьяконов В., Круглов В. Математические пакеты расширения Matlab. Специальный справочник. – СПб.: Питер, 2001. – 480 с.

[6] Заенцев И. В. Нейронные сети: основные модели. Воронеж – 1999.

[7] Гостев В.И. Системы управления с цифровыми регуляторами: Справочник.– К.: Техніка, 1990.– 280 с.

[8] Денисов А. И., Войтенко В. П. Микропроцессорное управление импульсным преобразователем в системе точного позиционирования// Технічна електродинаміка, тем. випуск “Проблеми сучасної електротехніки” – Київ – 2000 Ч. 4 – С. 34-37.

[9] Ту Ю. Современная теория управления. Пер. с англ. Я.Н.Гибатулина, под ред. В.В.Солодовникова. – М.: Машиностроение, 1971. – 472 с.