

УДК 004.032.26

UDC 004.032.26

НЕЧЕТКИЙ КОНТРОЛЛЕР С ПРАВИЛАМИ САМОМОДИФИКАЦИИ

FUZZY CONTROLLER WITH SELF-MODIFICATION RULES

Ключко Владимир Игнатьевич
д.т.н., профессор

Kluchko Vladimir Ignatievich
Dr.Sci.Tech., professor

Шумков Евгений Александрович
к.т.н.

Shumkov Eugene Alexandrovich
Cand.Tech.Sci.

Карнизьян Роман Оганесович
аспирант
*Кубанский Государственный Технологический
Университет, Краснодар, Россия*

Karnizian Roman Oganosovich
postgraduate student
*Kuban State Technological University, Krasnodar,
Russia*

В статье предложена нейро-нечеткая реализация нейросетевой топологии «Внутренний учитель»

The article proposes the implementation of neuro-fuzzy neural network topology of "Internal Teacher"

Ключевые слова: ГИБРИДНАЯ НЕЙРОНЕЧЕТКАЯ СЕТЬ, НЕЙРОННАЯ СЕТЬ, ОБУЧЕНИЕ С ПОДКРЕПЛЕНИЕМ, ПОЛИТИКА САМООБУЧЕНИЯ, ANFIS, АДАПТИВНЫЙ КРИТИК

Keywords: HYBRID NEURAL-NETWORK, NEURAL NETWORK, REINFORCEMENT LEARNING, ANFIS, SELF-LEARNING POLICY, ADAPTIVE CRITIC

Введение

В настоящее время стали популярны реализации систем управления объектами на базе топологий с подкреплением [2], в частности все большее развитие получают адаптивные критики [8,9], а также системы с различными реализациями Q – обучения [11]. При этом существует большое количество топологий на все случаи жизни, как адаптивных критиков, так и систем на базе Q – обучения, в частности приведем следующий список: DHP, GDHP, ADHPD, ADGDHP, GIFACL, FACL, Q – Хемминг, Dyna – Q, Q – Kohonen network, CMLP, HQ и т.д.[6, 8,9,10,11,12]. Каждая из топологий имеет свою область применения и нет ни одной универсальной [2,6]. Также у каждой топологии имеются свои достоинства и недостатки. Одной из попыток создать универсальную топологию для создания систем управления на основе обучения с подкреплением, является разработка нейросетевой топологии "Внутренний учитель" [2,3,6]. Рассмотрим дальнейшее развитие данной топологии.

Классическая топология «Внутренний учитель»

Для топологии «Внутренний учитель» используются две основные компоненты – Решатель и Учитель. Подобную функциональную схему можно найти еще в работе 1971 года [1]. Предложенная в [3] топология «Внутренний учитель» состоит из нескольких структурных компонент (см. Рисунок 1). Управляющая компонента системы – «Решатель». Она получает вектор состояния объекта и среды и формирует на выходе управляющий вектор. Вектор состояния среды формируется сенсорной компонентой. Вектор управляющего воздействия обрабатывает моторная компонента. Наиболее важная часть системы – «Учитель», компонента обучения. Компонента обучения оценивает состояние среды на предмет изменения характера воздействия на объект, и если это имеет место быть, то Учитель вмешивается в управление, перенастраивая правила управления или же вообще изменяя их. При этом Учитель в режиме первоначального обучения формирует специального вида управляющие воздействия, позволяющие быстро и полно изучить среду. Таким способом система управления (далее СУ) на базе данной топологии меняет стратегию управления. На вход Учитель получает данные обратной связи от внешней среды и историю управления объекта управления.

В качестве компоненты обратной связи используется коэффициент эффективности λ (далее КЭ¹):

$$I_i = \frac{A_i}{E_i},$$

где i – номер кванта времени (итерации, одного законченного цикла управления), для которого вычисляется коэффициент; A_i – безразмерный показатель полезной деятельности, произведенной объектом; E_i – безразмерный показатель затраченных ресурсов.

¹ По сути – модифицированное подкрепление.

КЭ записан в обобщенном виде и вычисляется по вектору состояния среды и состояния объекта управления. Для использования описываемой топологии в конкретных применениях КЭ должен записываться в частном виде. КЭ обрабатывается и вычисляется в компоненте обратной связи (БКЭ – блок коэффициента эффективности на Рисунке 1).

СУ должна максимизировать коэффициент эффективности, реализуя цели управления. КЭ используется для выявления изменения стратегии поведения улучшившего или ухудшившего общую оценку работы. КЭ является основным источником информации для построения эффективной схемы самообучения. При этом КЭ вычисляется на основе обработки сведений о внешней среде, объекте, и о характере взаимодействия объекта со средой.

Также для СУ на базе «Внутреннего учителя» вводятся адаптационные параметры – АП (один или несколько), регулируя которые можно добиться изменения работы Решателя. АП либо непосредственно входят в вектор подаваемый на вход Решателя, либо влияют на изменение обучающей выборки для Решателя.

В общем случае количество правил самообучения Учителя определяется по формуле:

$$N = 3^{t(KЭ)+t(АП)}$$

где $t(\hat{E}\acute{Y})$ - глубина временного погружения по КЭ и $t(\hat{A}\acute{I})$ - глубина временного погружения по АП, то есть за сколько предыдущих временных итераций учитывать КЭ и АП.

Правила самообучения² учителя изменяют адаптационные параметры и сформулированы следующим образом:

- если снизилось значение КЭ, то изменяем АП в противоположном направлении от предыдущих изменений;

² Обычно трактуются, как «политика самообучения» (англ. self-modifying policies - SMP).

- если значение КЭ выросло, то продолжаем изменять АП в направлении от предыдущих изменений;
- если значение не изменилось, то в зависимости от КЭ, либо оставляем текущие правила (если КЭ устраивает), либо хаотически изменяем применяемое правило.

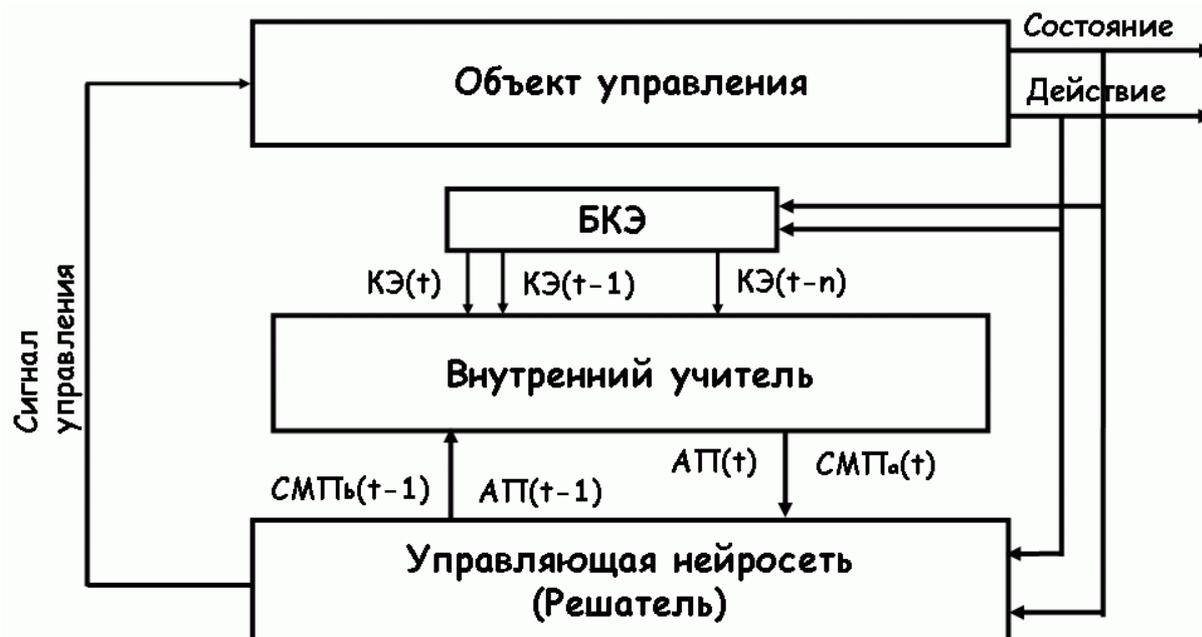


Рисунок 1. Топология "Внутренний учитель"

Подробно со стандартной топологии «Внутренний учитель» ознакомится в работах [2,6], а также в патенте РФ G 06 F 9/00 № 2266558.

Нейро-нечеткая топология «Внутренний учитель»

Предложим нейро–нечеткую реализацию компоненты Учитель. Пусть характеристики работы системы – коэффициент эффективности и адаптационный параметр рассматриваются, как нечеткие переменные с

лингвистическими значениями: *снизился*, *не_изменился* и *увеличился*³. Тогда нечеткие правила работы Учителя будут следующими (для $t(\hat{E}\acute{Y})=2$ и $t(\hat{A}\grave{I})=1$):

ЕСЛИ $\hat{E}\acute{Y}(t) = \text{снизился}$ И $\hat{E}\acute{Y}(t-1) = \text{снизился}$ И $\hat{A}\grave{I}(t-1) = \text{снизился}$, ТО $\hat{A}\grave{I}(t) = \text{увеличить}$

ЕСЛИ $\hat{E}\acute{Y}(t) = \text{снизился}$ И $\hat{E}\acute{Y}(t-1) = \text{не изменился}$ И $\hat{A}\grave{I}(t-1) = \text{снизился}$, ТО $\hat{A}\grave{I}(t) = \text{увеличить}$

ЕСЛИ $\hat{E}\acute{Y}(t) = \text{снизился}$ И $\hat{E}\acute{Y}(t-1) = \text{увеличился}$ И $\hat{A}\grave{I}(t-1) = \text{снизился}$, ТО $\hat{A}\grave{I}(t) = \text{снизить}$

и т.д. (в данном случае 27 правил).

Функции формы для правил можно задавать несколькими способами, но предпочтительней использовать трапециевидную или гауссовую функции принадлежности.

Пусть $t(\hat{E}\acute{Y})=2$ и $t(\hat{A}\grave{I})=1$. Входными переменными будут значения $\hat{E}\acute{Y}(t)$, $\hat{E}\acute{Y}(t-1)$ и $\hat{A}\grave{I}(t-1)$. Выходом компоненты Учитель будет переменная $\hat{A}\grave{I}(t)$.

Обозначим через L_1 - функцию формы для выражения $\hat{E}\acute{Y}(t) = \text{снизился}$, L_2 - для $\hat{E}\acute{Y}(t-1) = \text{снизился}$, L_3 - для $\hat{A}\grave{I}(t-1) = \text{снизился}$. Аналогично введем функцию M : M_1 для $\hat{E}\acute{Y}(t) = \text{не изменился}$, M_2 для $\hat{E}\acute{Y}(t-1) = \text{не изменился}$ и M_3 для $\hat{A}\grave{I}(t-1) = \text{не изменился}$. Также введем функцию H : H_1 для $\hat{E}\acute{Y}(t) = \text{увеличился}$, H_2 для $\hat{E}\acute{Y}(t-1) = \text{увеличился}$ и H_3 для $\hat{A}\grave{I}(t-1) = \text{увеличился}$.

Для выхода введем функции $L_4 = \text{снизился}$, $M_4 = \text{не изменился}$, $H_4 = \text{увеличился}$. Система нечеткого вывода по Цукамото будет следующая. Для четких значений $\hat{E}\acute{Y}(t)$, $\hat{E}\acute{Y}(t-1)$ и $\hat{A}\grave{I}(t-1)$ определим релевантность правил a :

³ Ничто не мешает увеличить количество лингвистических значений, например, использовать такие: *сильно снизился*, *снизился*, *не изменился*, *увеличился*, *сильно увеличился*.

$$a_1 = L_1 + L_2 + L_3$$

$$a_2 = L_1 + M_2 + L_3$$

$$a_3 = L_1 + H_2 + L_3$$

....

$$a_{27} = H_1 + H_2 + H_3$$

Выходы по каждому правил определяются с помощью обратных функций принадлежности правых частей правил:

$$Y_1 = H_4^{-1}(a_1)$$

$$Y_2 = H_4^{-1}(a_2)$$

$$Y_3 = L_4^{-1}(a_3)$$

...

$$Y_{27} = H_4^{-1}(a_{27})$$

Тогда общий выход системы нечетких правил будет

$$y_0 = \frac{\sum_{i=1}^{27} a_i y_i}{\sum_{i=1}^{27} a_i}$$

Реализуем приведенные правила самообучения Учителя с помощью нейро – нечеткой топологии ANFIS [7]. Схема приведена на Рисунке 2.

Слой 1 предназначен для вычисления степени на узлах, с которыми заданные входы удовлетворяют функциям принадлежности, ассоциированных с этими узлами.

Слой 2 предназначен для вычисления силы правил для каждого узла. При этом выход верхнего нейрона равен $a_1 = L_1 \wedge L_2 \wedge L_3$, второго сверху $a_2 = L_1 \wedge M_2 \wedge L_3$ и т.д. Все узлы обозначены как T , так как можно выбрать любую T -норму для моделирования логического И [7].

Слой 3 предназначен для нормализации силы правил:

$$b_i = a_i / (a_1 + a_2 + .. + a_{27})$$

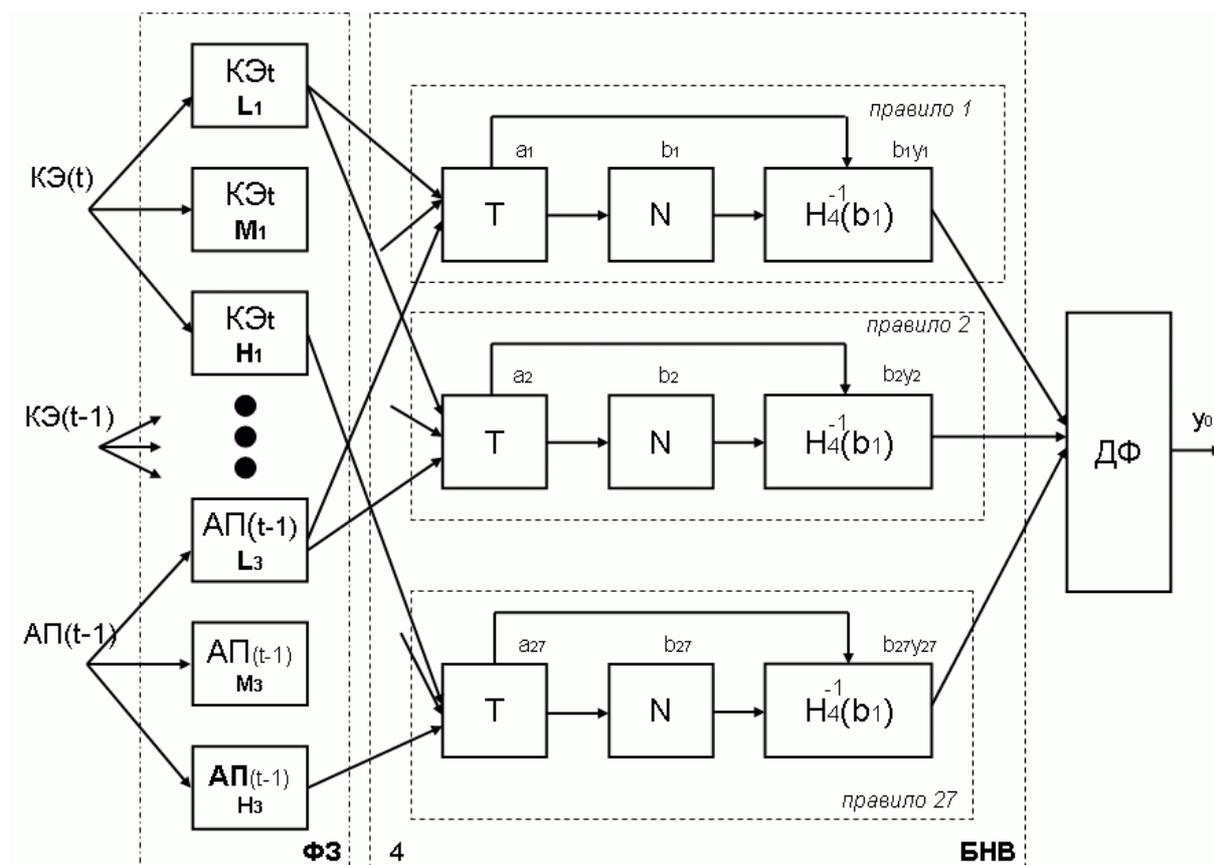


Рисунок 2. Учитель на базе ANFIS

Слой 4 вычисляет произведение нормализованной силы правил и индивидуального выхода соответствующего правила:

$$b_1 Y_1 = b_1 H_4^{-1}(a_1)$$

$$b_2 Y_2 = b_2 H_4^{-1}(a_2)$$

.....

$$b_{27} Y_{27} = b_{27} H_4^{-1}(a_{27})$$

Слой 5 вычисляет выход сети (на единственном нейроне):

$$y_0 = b_1 Y_1 + b_2 Y_2 + \dots + b_{27} Y_{27}$$

Следующим шагом является обучение созданной сети. Для обучения гибридной сети ANFIS используют алгоритм обратного распространения ошибки [7]. Обычно число правил жестко задается сверху. Пусть будет N правил и будем использовать нечеткий вывод по Цукамото (Tsukamoto). При этом обучение разбивается на следующие этапы:

1. назначить начальные значения весов и параметров;
2. на вход ANFIS подать входные сигналы \bar{X}_i из обучающей выборки. При этом i выбирается случайным образом;
3. рассчитать выходные значения во всех слоях;
4. рассчитать ошибку на выходе ANFIS как

$$E^k = \frac{1}{2}(Y_i^k - \bar{Y}_i^k)$$

(после некоторого числа итераций необходимо считать также ошибку по пакетному способу и если ошибка меньше заданной, то остановить обучение);

5. модифицировать параметры функций формы по формулам:
для выходных переменных

$$b_4(t+1) = b_4(t) - V\left(\frac{\partial E}{\partial b_4}\right)$$

$$c_4(t+1) = c_4(t) - V\left(\frac{\partial E}{\partial c_4}\right)$$

и т.д..

Для обучения можно использовать также алгоритм RProp [4].

Заключение

Таким образом, используя нечеткую логику можно реализовать мягкое (нечеткое) управление на базе топологии «Внутренний учитель». Использование нечеткой логики в данном случае позволяет для некритических задач повысить адаптационные свойства СУ. Предложенная топология сочетает в себе преимущества, как нейронных, так и нечетких систем. Возможность использования нечеткой логики в топологии «Внутренний учитель» также говорит об универсальности топологии.

Литература

1. Мкртчян С. О. Нейроны и нейронные сети. – М.: Энергия, 1971. - 232 с.
2. Стасевич В.П. Анализ и адаптивное управление в недетерминированных средах на основе самообучения. 2007. 170 с.
3. Стасевич В.П., Воротников С.А. Использование нейросетевых структур для управления динамическими объектами в недетерминированной среде // Экстремальная робототехника: Материалы XII Научно-технической конференции. - Санкт-Петербург: СПбГТУ, 2002.
4. Хайкин С. Нейронные сети: полный курс. 2-е изд.: Пер. с. англ. – М.: Издательский дом «Вильямс», 2006 – 1104 с.
5. Шумков Е.А. Обучение и самообучение в адаптивных системах управления / Е.А.Шумков, В.П. Стасевич, В.Н. Зуева // Известия вузов Северо-Кавказский регион. Технические науки. Новочеркасск, 2006. - №2. – с. 84 – 86.
6. Шумков Е.А. Система поддержки принятия решений предприятия на основе нейросетевых технологий. 2004. 158 с.
7. Ярушкина Н.Г. Основы теории нечетких и гибридных систем: Учеб. Пособие. – М.: Финансы и статистика. 2004. 320 с.
8. Balakrishnan S.N. and Biega V. "Adaptive Critic Based Neural Networks for Control," in Proc. Am. Contr. Conf., Seattle, WA, June 1995, pp. 335-339.
9. Prokhorov D., Wanch D. Adaptive critic designs. IEEE transactions on Neural Networks, September 1997, pp. 997-1007.
10. Sutton R.S., Reinforcement Learning. Boston: MA: Kluwer Academic, 1996.
11. Watkins C., Dayan P. Q – learning. // Machine Learning, vol. 8, pp. 279 – 292. 1992.
12. Wiering M, Schmidhuber J. HQ – learning. Adaptive behavior, 6(2):219 – 246, 1998.

References

1. Mkrтчjan S. O. Nejronny i nejronnye seti. – M.: Jenergija, 1971. - 232 s.
2. Stasevich V.P. Analiz i adaptivnoe upravlenie v nedeterminirovannyh sredah na osnove samoobuchenija. 2007. 170 s.
3. Stasevich V.P., Vorotnikov S.A. Ispol'zovanie nejrosetevyh struktur dlja upravlenija dinamičeskimi ob#ektami v nedeterminirovannoj srede // Jekstremal'naja robototehnika: Materialy XII Nauchno-tehnicheskoy konferencii. - Sankt-Peterburg: SPbGTU, 2002.
4. Hajkin S. Nejronnye seti: polnyj kurs. 2-e izd.: Per. s. angl. – M.: Izdatel'skij dom «Vil'jams», 2006 – 1104 s.
5. Shumkov E.A. Obuchenie i samoobuchenie v adaptivnyh sistemah upravlenija / E.A.Shumkov, V.P. Stasevich, V.N. Zueva // Izvestija vuzov Severo-Kavkazskij region. Tehnicheskie nauki. Novočerkassk, 2006. - №2. – s. 84 – 86.
6. Shumkov E.A. Sistema podderzhki prinjatija reshenij predprijatija na osnove nejrosetevyh tehnologij. 2004. 158 s.
7. Jarushkina N.G. Osnovy teorii nechetkih i gibridnyh sistem: Učeb. Posobie. – M.: Finansy i statistika. 2004. 320 s.
8. Balakrishnan S.N. and Biega V. "Adaptive Critic Based Neural Networks for Control," in Proc. Am. Contr. Conf., Seattle, WA, June 1995, pp. 335-339.
9. Prokhorov D., Wanch D. Adaptive critic designs. IEEE transactions on Neural Networks, September 1997, pp. 997-1007.
10. Sutton R.S., Reinforcement Learning. Boston: MA: Kluwer Academic, 1996.
11. Watkins C., Dayan P. Q – learning. // Machine Learning, vol. 8, pp. 279 – 292. 1992.

12. Wiering M, Schmidhuber J. HQ – learning. Adaptive behavior, 6(2):219 – 246, 1998.