http://dx.doi.org/10.35596/1729-7648-2019-126-8-125-132

Оригинальная статья
Original paper

УДК 681.3;004.896

МЕТОД ПОСТРОЕНИЯ МОДЕЛИ НЕЙРОРЕГУЛЯТОРА ПРИ ОПТИМИЗАЦИИ СТРУКТУРЫ УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМ ЦИКЛОМ

СМОРОДИН В.С., ПРОХОРЕНКО В.А.

Гомельский государственный университет имени Франциска Скорины,
г. Гомель, Республика Беларусь

Поступила в редакцию 16 октября 2019

Аннотация. Цель работы, результаты которой представлены в рамках данной статьи, состояла в разработке метода построения модели нейрорегулятора для случая оптимизации структуры управления технологическим циклом, реализация которого осуществляется на базе средств автоматизации производственного процесса при наличии физического контроллера, который осуществляет управление технологическим процессом в соответствии с заданной программой. Для достижения поставленной цели были решены задачи, связанные с применением нейросетевых технологий при построении математической модели нейрорегулятора. При этом математическая модель нейрорегулятора разработана на основе физического прототипа, а процедура синтеза управления в режиме реального времени (адаптивного управления) основана на процедуре обучения рекуррентной нейронной сети, построенной с использованием блоков LSTM, которые имеют возможность хранить информацию в течение длительного времени. Предложен метод построения модели нейрорегулятора для реализации управления технологическим циклом производства при решении задачи поиска оптимальной траектории на фазовой плоскости параметров состояний технологического цикла. В рассматриваемой задаче поиска оптимальной траектории математическая модель нейрорегулятора в каждый момент времени получает информацию о текущем состоянии системы, данные о смежных состояниях объекта управления и направление движения по фазовой плоскости состояний, которое определяется действующими критериями оптимизации управления. С учетом полученных результатов установлено, что рекуррентные сети с LSTM-модулями могут успешно применяться в качестве аппроксиматора Q-функции агента для решения поставленной задачи в условиях, когда частично наблюдаемая область состояний системы имеет сложную структуру. Выбор предложенного в работе метода адаптации к управляющим воздействиям и внешним возмущениям окружающей среды удовлетворяет требованиям к быстродействию процесса адаптации, равно как и требованиям к качеству процессов управления для случаев, когда актуальная информация о природе случайных возмущений управления отсутствует. Среда для проведения экспериментов, а также модели нейронных сетей реализованы на языке программирования Python с использованием библиотеки TensorFlow.

Ключевые слова: модель нейрорегулятора, адаптивное управление, оптимизация параметров функционирования, фазовая плоскость состояний, оптимальная траектория.

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Для цитирования. Смородин В.С., Прохоренко В.А. Метод построения модели нейрорегулятора при оптимизации структуры управления технологическим циклом. Доклады БГУИР. 2019; 7–8(126): 125-132.

METHOD OF CONSTRUCTION OF A NEUROREGULATOR MODEL WHEN OPTIMIZING THE CONTROL STRUCTURE OF A TECHNOLOGICAL CYCLE

VIKTOR S. SMORODIN, VLADISLAV A. PROKHORENKO

Gomel State University named after Francisk Skorina, Gomel, Republic of Belarus

Submitted 16 October 2019

Abstract. In this paper authors present the results of a research that had a purpose to develop a method of constructing a neuroregulator model for the case of optimization of the control structure of a technological cycle. The method's implementation is based upon the automation of a production process when a physical controller, that operates the technological process according to a given program, is present. In order to achieve this goal, the artificial neural network approaches were implemented to create a mathematical model of the neuroregulator. The mathematical model of the neuroregulator is based on a physical prototype, and the procedure of a real-time control synthesis (adaptive control) is based on recurrent neural network training. The neural network architecture includes LSTM blocks, which are capable of storing information for long periods of time. A method is proposed for constructing a neuroregulator model for control of a production cycle when solving the task of the optimal trajectory finding on the phase plane of the technological cycle states. In the considered task of the optimal trajectory finding the mathematical model of the neuroregulator receives at each moment of time information about the current system state, the adjacent system states and the movement direction on the phase plane of states. Movement direction is determined by the given control optimization criteria. Based on the research results it was found that recurrent networks with LSTM modules can be used successfully as an approximator for the agent's *Q*-function to solve the given problem when the partially observed region of system states has a complex structure. The choice of the method of adaptation to the control actions and the external environmental disturbances proposed in the paper satisfies the requirements for the adatation process performance, as well as the requierments for the control processes quality, when there is lack of information about the nature of random control disturbances. The experimental environment, as well as the neural network models was implemented using the Python programming language with TensorFlow library.

Keywords: neuroregulator model, adaptive control, optimization of functioning parameters, phase plane of states, optimal trajectory.

Conflict of interests. The authors declare no conflict of interests.

For citation. Smorodin V.S., Prokhorenko V.A. Method of construction of a neuroregulator model when optimizing the control structure of a technological cycle. Doklady BGUIR. 2019; 7–8(126): 125-132.

Введение

Современный обзор состояния разработок в области анализа функционирования систем управления показывает, что проблема динамического определения параметров системы в рамках решения многокритериальной задачи оптимизации управления часто возникает ввиду наличия случайных внешних управляющих воздействий, включая и наличие человеческого фактора. Это имеет место особенно в тех случаях, когда ввиду наличия внешних управляющих воздействий меняется структура управления в процессе реализации рабочего цикла сложной технологической системы. Рассмотрим один из вариантов подхода к формализации технологического цикла производства, работающего под управлением автоматизированной системы управления технологическим процессом (АСУТП) при наличии контроллера, который администрирует работу системы управления в соответствии с заданной программой.

В основу формализации структуры управления технологическим циклом производства положены результаты исследований авторов в области анализа функционирования вероятностных технологических систем [1]. Под адаптивным управлением в настоящей статье будем понимать способность системы управления изменять свои параметры в зависимости

от штатных управляющих воздействий контроллера системы и внешних возмущений. При этом будем понимать, что вероятностно-технологические системы (ВТС) – это технологические объекты, параметры работы которых имеют вероятностный характер. Технологический цикл производства состоит из набора технологических операций и ресурсов, которые потребляются технологическими операциями в процессе их реализации, а технологические операции могут конкурировать за требуемые ресурсы. Подобное представление позволяет рассматривать технологический цикл производства как замкнутую систему, для изучения которой могут быть применены нейросетевые технологии, основанные на построении математических моделей искусственных нейронных сетей (ИНС). Математическая модель ВТС строится в пространстве состояний и включает в себя набор входных и выходных данных, а также переменных состояния конечного набора взаимосвязанных математических моделей компонентов управления. Моделирование подобных типов технологических процессов осуществляется на основе критических или средних значений показателей расхода ресурсов.

В настоящей статье рассматриваются технологические процессы, которые имеют непрерывный характер и работают в режиме реального времени. Контроль функционирования технологического цикла производства осуществляется с использованием соответствующих средств автоматизации. Математическая модель нейрорегулятора строится на основе существующего прототипа, а процедура синтеза адаптивного управления основана на обучении рекуррентной нейронной сети с помощью блоков LSTM [2], которые имеют возможность хранить информацию в течение длительного времени.

Подобный подход позволяет создать аналог нейронной сети базы знаний об окружающей среде системы управления (случайных помехах и предыдущих состояниях контроллера). Выбор данного метода адаптации к внешним управляющим воздействиям удовлетворяет как требованиям к быстродействию процесса адаптации, так и требованиям к качеству процессов в управления, когда актуальная информация о природе возможных случайных помех отсутствует.

Общая постановка задачи

В рассматриваемой задаче поиска оптимальной траектории на фазовой плоскости состояний системы управления в каждый момент времени модель нейрорегулятора получает вектор, состоящий из нескольких элементов: данные о текущем состоянии системы, данные о смежных состояниях объекта управления и направление движения по фазовой плоскости сотояний в соответствии с заданными критериями оптимизации управления. Результатом работы модели нейрорегулятора в каждый момент времени является вектор, определяющий выбор следующего состояния системы управления на фазовой плоскости. Перемещения на плоскости состояний осуществляются до тех пор, пока не будет получена оптимальная траектория перемещений по фазовой плоскости возможных состояний в рамках заданного критерия качества управления.

Одним из подходов к решению поставленной задачи в предложенной формализации может быть использование методов обучения модели с учителем: построение математической модели нейрорегулятора таким образом, чтобы она воспроизводила правильные последовательности действий в заданной последовательности состояний системы. Однако при решении реальных задач подобный подход, как известно, может привести к трудностям при формировании достаточно полного обучающего множества исходной модели. При этом наличие большой выборки статистики о работе системы не является гарантией, что таковая будет отражать важные области фазового пространства состояний системы, если на практике они встречаются относительно редко.

Поэтому в работе рассмотрена группа алгоритмов обучения с подкреплением (в частности, *Q*-обучение), примененных к имитационной модели рассматриваемой технологической системы. Данные алгоритмы предполагают исследовательскую деятельность агента [3], управляемого контроллером, и потому имеют потенциал обучить контроллер в более полной мере.

Используемая идея *Q*-обучения состоит в том, что агент в процессе обучения строит верную функцию *Q* оценки следующего состояния, к которому может привести выбор некоторого управления. В качестве аппроксиматора данной функции может быть использована нейронная сеть. Подобный подход дал хорошие результаты при решении сложных задач с частично наблюдаемым окружением, в которых был достигнут уровень человека-эксперта [3, 4].

Постановка задачи *Q*-обучения агента

На каждом шаге обучения агент получает данные о текущем состоянии окружения s_t, выбирает действие a_t в соответствии с политикой выбора действий *p*, и получает от окружения сигнал о вознаграждении r_t. Задачей агента в процессе обучения является поиск такой политики действий, которая будет максимизировать ожидаемое суммарное вознаграждение R_t.

Q-функция (функция качества) данной политики π определяется как ожидаемое вознаграждение от использования действия *a* в состоянии окружения *s*:

$$Q^{\pi}(s,a) = E\left[R_t \mid s_t = s, a_t = a\right].$$

Аппроксиматором функции *Q* выступает нейронная сеть. Задачей обучения является поиск таких значений настраиваемых параметров нейросети, при которых приближенная функция будет достаточно близкой к оптимальной функции *Q**, определяемой следующим образом[1]:

$$Q^*(s,a) = \max_{\pi} E\left[R_t \mid s_t = s, a_t = a\right] = \max_{\pi} Q^{\pi}(s,a).$$

Для оптимальной функции *Q** выполняется уравнение Беллмана[4][1]:

$$Q^*(s,a) = E\left[r + \gamma \max_{a'} Q^*(s',a') \mid s,a\right].$$

Отсюда функция потерь (ошибки) для текущей $Q(\theta)$[1]:

$$L_t(\theta_t) = E_{s,a,r,s'}\left[(y_t - Q_{\theta_t}(s,a))^2\right],$$

где $y_t = r + \gamma \max_{a'} Q_{\theta_t}(s',a)$.

Отсюда – градиент функции потерь:

$$\nabla_{\theta_t} L_t(\theta_t) = E_{s,a,r,s'}\left[(y_t - Q_{\theta}(s,a))\nabla_{\theta_t} Q_{\theta_t}(s,a)\right],$$

На практике обычно вычисляется его приближение [4][1]:

$$\nabla_{\theta_t} L_t(\theta_t) \approx \left(y_t - Q_{\theta}(s,a)\right)\nabla_{\theta_t} Q_{\theta_t}(s,a).$$

Во время обучения агента на практике обычно используется не онлайн-обучение (одно обновление настраиваемых параметров после каждого действия), а какой-либо механизм демонстрации алгоритму обучения опытов, или последовательностей опытов, полученных в предыдущие моменты времени [4][1].

При решении практических задач агенту часто бывает недоступна полная информация о состоянии среды. В этой ситуации агент, который пользуется аппроксиматором *Q*, зависящим только от текущего наблюдаемого состояния среды может быть неэффективным при достаточно сложной структуре среды. Существует несколько подходов к решению данной проблемы, которые предполагают наличие у агента внутреннего состояния, которое сохраняется при переходе к следующему моменту времени. Один из подходов предполагает подачу на вход аппроксиматора не только текущего, но также и некоторого числа предыдущих наблюдаемых состояний среды $s_{t-1}, s_{t-2}, ..., s_{t-n}$ [4]. Другой подход основан на применении рекуррентных нейронных сетей, обладающих внутренним состоянием h_t[2]. В качестве структурного элемента, обеспечивающего сохранение внутреннего состояния h_t, может использоваться LSTM (Long short-term memory) [2].

[1] Lample G., Chaplot D.S. Playing FPS Games with Deep Reinforcement Learning. *AAAI'17 Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. 2140-2146, AAAI Press; 2017.

[2] Hausknecht M., Stone P. Deep recurrent *q*-learning for partially observable mdps. *2015 AAAI Fall Symposium Series*. AAAI Press; 2015.

Формирование функции вознаграждения

Определение функции вознаграждения играет важную роль, определяя поведение агента, формируемое при обучении. В экспериментах использовалась функции вознаграждения следующего вида:

$$r_t(s,a) = lr + mtgr \cdot d\ ,$$

где lr – константа, прибавляемая к награде на каждом шаге независимо от действий агента, $mtgr$ – коэффициент вознаграждения за движение по направлению к цели, d – изменение расстояния до цели.

Значения, использованные в экспериментах: Lr = –2,0; $mtgr$ = 3,0.

В случае совершения недопустимого действия r_t = –100.

В случае достижения цели r_t = 500.

Стратегии исследования в процессе обучения

В отличие от обучения с учителем, агенту доступно окружение только через его собственные действия. Следовательно, выбор стратегии исследования окружения в процессе обучения представляет собой важную задачу.

В процессе данной работы использовались стратегии выбора очередного действия агентом при обучении:

– ε-greedy – выбор случайного действия с вероятностью, которая уменьшается в процессе обучения [3, 4];

– softmax/Boltzmann – случайный выбор одного из доступных действий с распределением

$$P_t(a) = \frac{\exp(q_t(a)/\tau)}{\sum_{i=1}^{n}\exp(q_t(i)/\tau)}\ ,$$

где $q_t(a)$ – оценки Q, построенные текущей моделью, τ – параметр, убывающий с течением времени [3].

Выбор модели для решения задачи в описанной формализации

В процессе работы рассмотрено несколько нейросетевых агентов различных типов при решении задачи поиска траектории в двумерной области. Интересно сравнить их способности обучиться и затем осуществлять навигацию в области сложной структуры.

sfDQN – агент, использующий нейронную сеть типа МСП (многослойный персептрон). На вход сети на каждом шаге подается только текущее наблюдаемое состояние среды.

Структура сети:

1) Dense x8 ReLU
2) Dense x16 ReLU
3) Dense x32 ReLU
4) Dense x4 no activation.

mfDQN – агент, использующий МСП. На вход сети на каждом шаге подаётся текущее наблюдаемое состояние среды, а также несколько предыдущих (в дальнейших экспериментах это 3 предыдущих состояния).

Структура сети:

1) Dense x8 ReLU
2) Dense x16 ReLU
3) Dense x32 ReLU
4) Dense x4 no activation.

DQRN1 – агент, использующий рекуррентную нейросеть на базе МСП с LSTM блоком. На вход сети подается текущее состояние среды. Обучается на длине последовательности 4.

Структура сети:

1) Dense x16 ReLU
2) Dense x16 ReLU
3) Dense x32 ReLU
4) LSTM x32 ReLU
5) Dense x4 no activation

Генерация данных и обучение моделей

Для обучения данных моделей сгенерированы с помощью клеточных автоматов [5] двумерные области 30×30 клеток с различным отношением проходимых и непроходимых клеток, различным расположением начальной и целевой позиции агента в области.

Сгенерировано 5 датасетов, на которых будут тестироваться способности моделей решать задачу в описанной формализации. Датасеты различаются отношением непроходимых клеток.

– Empty – 0 % – области не имеют непроходимых клеток;
– CA5 – 5 % непроходимых клеток;
– CA15 – 15 % непроходимых клеток;
– CA30 – 30 %;
– CA45 – 45 %.

В условиях, когда в среднем больше 45 % области не проходимо, становится невозможно получить достаточно удаленные друг от друга начальную и конечную точки в большей части сгенерированных областей.

Процедура обучения следующая.

1. Агент участвует в одном эпизоде решения задачи (новая область с новыми положениями стартовой и целевой клеток). Агент последовательно получает текущее наблюдаемое состояние среды и выбирает действие в соответствии с избранной стратегией исследования. Эпизод длится до тех пор, пока агент не достигнет цели, либо не совершит недопустимое действие (перемещение в непроходимую клетку), либо не достигнут лимит на число шагов (50).

2. Опыт, полученный агентом во время эпизода, сохраняется в формате (наблюдаемое состояние; выбранное действие; вознаграждение; следующее наблюдаемое состояние).

3. Осуществляется отбор сохраненного опыта для очередного обновления весов нейросети-аппроксиматора. В соответствии с выбранными параметрами обучения часть этого опыта отбирается случайно по всей памяти, а другая часть представляет собой последние записанные в память элементы.

4. Вычисляются обновленные значения Q, рассчитанные в соответствии со следующими наблюдаемыми агентом состояниями из опыта:

$$q_t(a) = r_t(s,a) + \gamma \max_i(q_{t+1}(i))$$

5. Обучение нейросети на обучающем множестве размера 32 в течение одной эпохи с помощью алгоритма RMSprop. Коррекции весов в процессе обучения – после предъявления 4 элементов множества. (Для МСП элементы множества это единичные элементы (состояние, q), в случае рекуррентных сетей это последовательности <(состояние, q)>).

6. Вернуться к п. 1 и проиграть новый эпизод.

Обучение агента длится до истечения лимита по эпизодам (5000) либо до тех пор, пока в среднем 50 из 50 предыдущих эпизодов не будут успешными.

Практика показала, что стратегия исследования e-greedy плохо подходит для нейросетевых агентов, имеющих внутреннее состояние. Наилучшие результаты обучения всех агентов получены с использованием стратегии softmax.

Результаты тестирования

Процедура тестирования состоит в проигрывании агентом 5000 эпизодов на областях, не входивших в обучающее множество. Результаты тестирования обученных моделей представлены в табл. 1.

wins – процент областей, в которых агент дошел до целевой клетки.

fails – процент областей, в которых агент совершил недопустимое действие.

Таблица 1. Результаты тестирования обученных моделей
Table 1. Testing results for the trained models

Dataset	sfDQN		mfDQN		DRQN1	
	wins, %	fails, %	wins, %	fails, %	wins, %	fails, %
Empty	95	5,1	**98**	4,2	92	6,3
CA5	**93,2**	4,6	93,1	6,4	82,4	5,1
CA15	68	45,4	63,1	39,4	**89,2**	5,3
CA30	32,5	68,1	31,6	68,25	**63,74**	36,19
CA45	13	86,7	16,2	83,45	**42,2**	54,3

Заключение

В настоящей работе предложен метод построения модели нейрорегулятора для реализации процедуры управления при решении задачи динамического определения параметров функционирования системы в соответствии с заданными критериями качества, основанный на реализации задачи поиска оптимальной траектории на фазовой плоскости состояний объекта исследований. Математическая модель нейрорегулятора реализована в виде программного кода на языке программирования Python, модели архитектур нейронных сетей с блоками LSTM построены на основе технологии TensorFlow.

Совместная работа математической модели и системы управления технологическим циклом осуществляется на базе программно-аппаратного интерфейса между вычислительной системой и блоками управления АСУТП. Установлено, что рекуррентные сети с LSTM-модулями могут успешно применяться в качестве аппроксиматора *Q*-функции агента для решения задачи в описанной формализации в условиях, когда частично наблюдаемая область состояний системы имеет сложную структуру.

Новизна данного подхода состоит в обеспечении возможности разработки алгоритмов адаптивного управления технологическим циклом производства на основе нейросетевых технологий, учитывающих допустимые диапазоны изменений параметров функционирования системы и обратные связи по управлению. Реализация подобных алгоритмов позволяет разработать дополнительные схемы резервирования контура управления объекта исследования при наличии условий неопределенности и риска возникновения аварийных ситуаций в процессе реализации технологического цикла производства.

Список литературы

1. Максимей И.В., Смородин В.С., Демиденко О.М. *Разработка имитационных моделей сложных технических систем.* Гомель: ГГУ им. Ф. Скорины; 2014.
2. Hochreiter S., Schmidhuber J. Long short-term memory. *Neural Computation*. 1997;9(8):1735-1780. DOI:10.1162/neco.1997.9.8.1735.
3. Sutton R.S., Barto A.G. *Reinforcement Learning: An Introduction*. Cambridge: The MIT Press; 1998.
4. Mnih V., Kavukcuoglu K., Silver D., Rusu A., Veness J., Bellemare M., Graves A., Riedmiller M., Fidjeland A., Ostrovski G., Petersen S., Beattie C., Sadik A., Antonoglou I., King H., Kumaran D., Wiestra D., Legg S., Hassabis D. Human-level control through deep reinforcement learning. *Nature.* 2015;518(7540):29-533. DOI:10.1038/nature14236.
5. Toffoli T., Margolus N. *Cellular Automata Machines: A New Environment for Modeling*. Cambridge: The MIT Press; 1987.

References

1. Maksimej I.V., Smorodin V.S., Demidenko O.M. [*Development of simulation models of complex technical systems*]. Gomel: GGU im. F. Skoriny; 2014. (in Russ.)
2. Hochreiter S., Schmidhuber J. Long short-term memory. *Neural Computation*. 1997;9(8):1735-1780. doi:10.1162/neco.1997.9.8.1735.
3. Sutton R.S., Barto A.G. *Reinforcement Learning: An Introduction.* Cambridge: The MIT Press; 1998.
4. Mnih V., Kavukcuoglu K., Silver D., Rusu A., Veness J., Bellemare M., Graves A., Riedmiller M., Fidjeland A., Ostrovski G., Petersen S., Beattie C., Sadik A., Antonoglou I., King H., Kumaran D., Wiestra D., Legg S., Hassabis D. Human-level control through deep reinforcement learning. *Nature* 2015;518(7540):529-533. DOI:10.1038/nature14236.
5. Toffoli T., Margolus N. *Cellular Automata Machines: A New Environment for Modeling*. Cambridge: The MIT Press; 1987.

Вклад авторов

Прохоренко В.А. разработал метод построения модели нейрорегулятора и модели архитектур нейронных сетей с блоками LSTM.

Смородин В.С. определил задачи, которые необходимо было решить в ходе проведения исследований, а также принимал участие в интерпретации их результатов.

Authors contribution

Prokhorenko V.A. has developed a method of construction of the neuroregulator model and implemented the neural network architectures that include LSTM blocks.

Smorodin V.S. has defined the set of research problems and took part in interpreting the research results.

Сведения об авторах

Смородин В.С., д.т.н., профессор, заведующий кафедрой математических проблем управления и информатики Гомельского государственного университета имени Франциска Скорины.

Прохоренко В.А., ассистент кафедры математических проблем управления и информатики Гомельского государственного университета имени Франциска Скорины.

Information about the authors

Smorodin V.S., D.Sci., Professor, Head of the Department of Mathematical Problems of Control and Informatics of Gomel State University named after Francisk Skorina.

Prokhorenko V.A., M.Sci., Assistant of the Department of Mathematical Problems of Control and Informatics of Gomel State University named after Francisk Skorina.

Адрес для корреспонденции

246019, Республика Беларусь,
г. Гомель, ул. Советская, д. 104,
Гомельский государственный университет
имени Франциска Скорины
тел. +375-29-329-27-99;
тел. 8-023-251-03-04;
e-mail: smorodin@gsu.by
Смородин Виктор Сергеевич

Address for correspondence

246019, Republic of Belarus,
Gomel, Sovetskaya st., 104,
Gomel State University
named after Francisk Skorina
тел. +375-29-329-27-99;
тел. 8-023-251-03-04;
e-mail: smorodin@gsu.by
Smorodin Victor Sergeevich