Нелинейная динамика. 2011. Т. 7. № 4 (Мобильные роботы). С. 859–875.
Полнотекстовая версия в свободном доступе
http://nd.ics.org.ru

УДК: 004.032.26, 004.852
MSC 2010: 68T05, 68Q32

# Обучение с подкреплением спайковой нейронной сети в задаче управления агентом в дискретной виртуальной среде

**О. Ю. Синявский, А. И. Кобрин**

В работе описываются методы обучения с подкреплением спайковой нейронной сети, управляющей роботом или интеллектуальным агентом. Применение спайковых нейронов в качестве базовых элементов сети позволяет использовать как пространственную, так и временную структуру входной сенсорной информации. Обучение сети производится с помощью подкрепляющих сигналов, идущих из внешней среды и отражающих степень успешности недавно выполненных агентом действий. Максимизация получаемого подкрепления ведется путем модулированной минимизации информационной энтропии функционирования нейрона, которая зависит от весов нейронов. Полученные законы изменения весов близки к явлениям синаптической пластичности, наблюдающейся в реальных нейронах. Работа алгоритма обучения с подкреплением проверяется на тестовой задаче поиска ресурсов агентом в дискретной виртуальной среде.

Ключевые слова: спайковый нейрон, адаптивное управление, обучение с подкреплением, информационная энтропия

## 1. Введение

В настоящее время системы управления поведением живых организмов значительно превосходят системы управления искусственных объектов, созданных человеком, по ряду важных аспектов, таких как способность к обучению и адаптации в неизвестных средах и к планированию поведенческих подзадач. Использование моделей нейросетевых структур живых организмов — один из путей достижения этих качеств при построении систем управления, например, робототехническими системами. Биологические нейроны общаются

Получено 6 июня 2011 года
После доработки 30 сентября 2011 года

Синявский Олег Юрьевич
sinyavskiyoleg@gmail.com
Кобрин Александр Исаакович
KobrinAI@mpei.ru
Национальный исследовательский университет «Московский энергетический институт»
111250, Россия, г. Москва, Красноказарменная ул., д. 14

между собой с помощью временных последовательностей пиковых импульсов напряжения одинаковой амплитуды — спайков [1]. Спайковый нейрон — модель биологического нейрона, работающая со спайками как с мгновенными событиями. Процесс генерации спайка зависит от истории импульсных последовательностей, поступивших на входы нейрона [2]. Использование сенсорной истории отдельного спайкового нейрона позволяет обрабатывать и пространственные, и временные соотношения между сенсорными и двигательными сигналами [3]. В работах [4, 5] показано, что плотность кодирования информации с помощью спайковых последовательностей выше, чем при кодировании только с помощью частоты импульсации нейрона, а спайковые нейроны обладают бóльшими вычислительными возможностями, так как активно используют не только пространственную, но и временную составляющую сигналов. Данные преимущества спайковых нейронов перед частотными и бинарными нейронами в решении некоторых типовых информационных задач и задач управления отражены в работах [4, 6, 7].

Наиболее существенной чертой биологических нейронных сетей является способность нейронов к обучению. Большинство работ по применению обучающихся управляющих спайковых нейронных сетей посвящены управлению интеллектуальными агентами в небольших виртуальных средах [8–12], движению (в основном, объезду препятствий) миниатюрных роботов в упрощенных искусственных средах [13–19], а также управлению (или коррекцией управления) двухзвенными роботами-манипуляторами [20–22]. При этом теория обучения спайковых нейронных сетей для построения сложных самообучающихся систем управления роботами, сравнимых с нейросетями живых организмов, в настоящее время еще не достаточно развита.

Спайковая нейронная сеть должна обладать информацией о цели функционирования робота. Без этой информации сеть не имеет возможности выбрать в каком-то смысле лучшее управление из многочисленных вариантов преобразования сенсорной информации в управляющую. Данная информация может быть заложена в сеть до начала ее функционирования в реальной среде на основе опыта наблюдения похожих систем. Например, может применяться копирование структур нейронных сетей живых организмов и механизмов их адаптации [15, 16, 21, 23]. Цель управления также может быть задана явно. Так, если примеры преобразования входной информации в выходную в основном известны, а от сети требуется обобщить закон преобразования на другие входные данные, то используются аналоги супервизорного обучения [20, 21]. В этих случаях достаточно эффективным является применение в управляющих сетях рекуррентных блоков случайно соединенных спайковых нейронов типа машины неустойчивых состояний — Liquid State Machine (LSM) [24]. Богатая внутренняя динамика LSM позволяет представить сложную временную историю входных сенсорных сигналов системы в виде пространственной комбинации спайков большого числа нейронов — «резервуара» LSM. Далее пространственная картина активации LSM считывается управляющими нейронами, которые обучаются с помощью каких-либо простых правил (например, линейной регрессии) [8, 13, 20].

Если возможна оценка результата действий робота, но разработчик системы управления не обладает никакой априорной информацией о возможностях управляемой системы, то для синтеза управляющей спайковой нейронной сети можно использовать эволюционные алгоритмы [12, 14, 19]. При этом возникает необходимость разбивать процесс обучения на итерации, после которых следует производить «селективный отбор». Другим способом сообщить сети об успешном или неуспешном управлении является использование сигналов подкрепления. Их применение делает возможным обучение в реальном времени, одновременно с функционированием сети.

В теории обучения с подкреплением существуют обучающие алгоритмы [25, 26], наиболее эффективно формализованные для марковских процессов принятия решений [27]. Расширение методов обучения с подкреплением на область спайковых нейронных сетей произошло сравнительно недавно [9, 11, 28, 29].

В данной работе представлен способ обучения с подкреплением спайковой нейронной сети, использующий метод минимизации информационной энтропии [30]. Работоспособность метода проверяется в задаче управления тестовой моделью интеллектуального агента в виртуальной дискретной среде.

## 2. Принципиальная схема управления с помощью спайковой нейронной сети

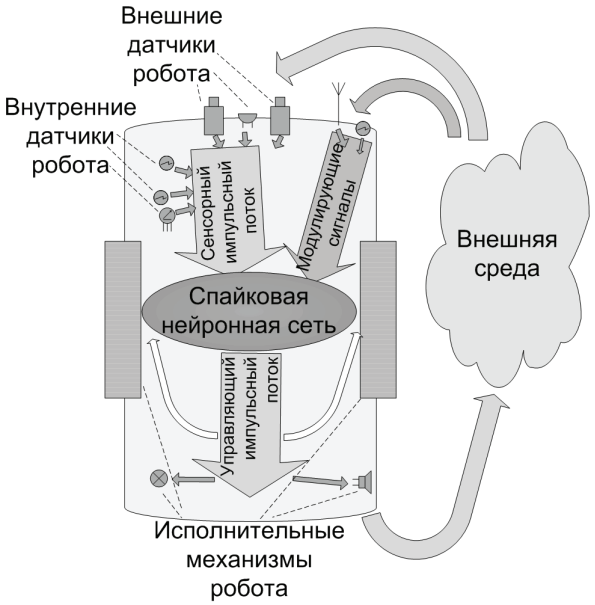


Рис. 1. Принципиальная схема взаимодействия управляющей нейронной сети с управляемым объектом и внешней средой.

Рассмотрим принципиальную схему взаимодействия управляющей спайковой нейронной сети с управляемым объектом (рис. 1). На входные каналы сети поступает поток сигналов в виде последовательности импульсов — спайков. Это могут быть сигналы от датчиков робота, собирающих информацию о внешней для робота среде, таких как визуальные и звуковые датчики, а также информация о внутреннем состоянии робота, например, от датчиков положения активаторов, состояния питания. Любой датчик робота является перекодировщиком информации, полученной им в некоей специфической форме, в форму потока импульсов. Например, аналоговый датчик может перекодировать аналоговый сигнал в импульсный поток с помощью частотно-импульсной модуляции или с помощью рецептивных полей [31]. Также входная информация может иметь вид дискретных событий — отдельных спайков, поступающих в определенные моменты времени, например, от устройства, распознающего голосовые команды человека. Совокупность входных данных от всех датчиков образует многомерный входной спайковый поток информации. Входной поток может иметь сложную пространственно-временную структуру, отражающую структуру событий

во внешней и внутренней среде робота. Если события внешней среды как-то связаны друг с другом и эта связь может быть измерена датчиками робота, то во входном потоке спайков также должны существовать корреляции между сигналами в виде повторяющихся во время функционирования робота временных последовательностей импульсов (спайковых паттернов) или частот импульсов.

Выходами сети служат аксоны некоторого множества нейронов сети. По ним во внешнюю по отношению к сети среду поступают управляющие импульсные потоки. Приемником этих потоков могут быть как исполнительные механизмы робота, непосредственно взаимодействующие с внешней средой (например, активаторы), так и внутренние системы, обеспечивающие корректное функционирование робота (например, системы охлаждения или питания).

В данной работе для сообщения сети цели функционирования робота предлагается использование подкрепляющих сигналов. Информационный поток сигналов подкрепления отделен от потока входных сенсорных сигналов, так как он несет в себе информацию с другой смысловой нагрузкой и по-другому обрабатывается спайковой нейронной сетью. При успешном выполнении роботом требуемой задачи ему приходит сигнал в виде положительного подкрепления. Сигнал отрицательного подкрепления (аналог боли) посылается при выполнении роботом нежелательных действий. Поток сигналов подкрепления не является структурно или пространственно сложным и не содержит паттернов, требующих распознавания. В основе процесса обучения управляющей сети лежит использование сенсорной обратной связи: после случайной или подсказанной извне генерации выходного импульсного паттерна, приведшего к положительному подкреплению (награде), нейронная сеть должна анализировать предшествующие награде входные потоки сигналов, а также выходные потоки сгенерированного до этого управления. После этого нейроны сети изменяют свои параметры таким образом, чтобы при повторении близких сенсорных условий повторить произведенные действия, и, следовательно, снова получить награду. Получение отрицательного подкрепления («боль») должно приводить к изменению параметров сети, которые, напротив, предотвращают выполнение действий, недавно произведенных системой.

При долгом невыполнении роботом поставленной задачи (или при отклонении от благоприятных условий) роботу дополнительно подаются стимулирующие сигналы («голод»). После получения этих сигналов нейронная сеть должна начать активно искать такой управляющий сигнал, который приведет к уменьшению стимулирующего подкрепления — получению награды или избеганию боли. В теории обучения с подкреплением важен компромисс между исследовательским поведением и поведением, использующим накопленные знания (exploration/exploitation dilemma) [25, 32]. Использование полезных навыков гарантирует устойчивое получение награды, однако процесс исследования окружающей среды может открыть новые источники положительного подкрепления. Рост интенсивности стимулирующих сигналов придает поведению робота в основном исследовательский характер, сигнализируя, что текущая стратегия поведения не приносит достаточного подкрепления. Исследовательское поведение в данной работе упрощенно реализуется в виде выполнения агентом достаточно случайных действий, которые лишь отчасти основаны на накопленном опыте.

## 3. Модель нейрона управляющей спайковой сети

В данной работе используется расширенная модель нейрона из класса Spike Response Model (SRM) [2] со стохастическим порогом [30, 33]. В момент прихода каждого спайка на входном канале генерируется фиксированный набор постсинаптических потенциалов,

имеющих вид функций отклика (альфа-функций) с различными временными профилями: $\alpha_i(t) = \frac{t}{\tau_j} e^{1-t/\tau_j} \cdot H(t)$, где $\tau_j$ — время максимума отклика, $H(t)$ — функция Хевисайда. Каждой альфа-функции на $i$-том входном канале приписывается свой вес $w_{ij}$. Взвешенные альфа-функции формируют результирующий постсинаптический потенциал на входном канале. Регулируя веса альфа-функций, можно регулировать форму суммарного постсинаптического потенциала, в частности, величину $\Delta u_i$ и время $\Delta t_i$ его максимума для $i$-го входного канала [34]. В реальном нейроне роль набора альфа-функций могут играть различные комбинации нейромедиаторов и рецепторов в синапсах нейрона, имеющие разные результирующие динамические характеристики влияния на постсинаптический потенциал нейрона. В экспериментах в данной работе использовался нейрон с тремя альфа-функциями на каждом входном канале. Таким образом, если нейрон имеет $n$ входных каналов, то веса нейрона задаются матрицей $\overline{W}$ размерности $n \times 3$. Напряжение на мембране нейрона является суммой постсинаптических потенциалов всех входных каналов. Напряжение для нейрона с $n$ входными каналами и $m$ альфа-функциями в каждом канале вычисляется по формуле

$$u(t) = \sum_{i}^{n} \sum_{t_{i,k}^{in} \in x_T^i} \sum_{j}^{m} w_{ij} \alpha_j (t - t_{i,k}^{in}), \tag{3.1}$$

где $n$ — количество входных каналов нейрона, $m$ — количество альфа-функций на одном входном канале, $\alpha_j(t)$ — альфа-функции, $w_{ij}$ — веса нейрона, $x_T^i$ — входная последовательность спайков, описываемая временами входных спайков $t_{i,k}^{in}$.

Недетерминированная составляющая вводится в модель нейрона с помощью стохастического порога [2]. Мгновенная плотность вероятности $\lambda(u(t))$ генерации спайка в конкретный момент времени $t$ нелинейно зависит от степени приближения напряжения на мембране $u(t)$ к пороговой величине. В экспериментах в данной работе выбрана экспоненциальная форма зависимости $\lambda$ от $u$: $\lambda(t) = e^{\kappa(u(t)-\theta)}$, где $\kappa$ — коэффициент стохастичности нейрона, $\theta$ — пороговая величина напряжения. При малом $\kappa$ вероятность генерации спайков почти не зависит от напряжения: выходные спайки нейрона мало отличаются от пуассоновского потока случайных событий с единичной интенсивностью. При большом $\kappa$ нейрон является детерминированным: выходные спайки генерируются только при превышении напряжением порогового значения $\theta$.

Плотность распределения вероятностей $p_T\{y_T\}$ генерации спайкового паттерна $y_T$ на интервале $T$ имеет следующий вид [33]:

$$p_T(y_T) = \prod_{t_q^{out} \in y_T} \lambda\big(u(t_q^{out})\big) \cdot e^{-\int_T \lambda(u(s))\, ds}, \tag{3.2}$$

где $\lambda(u)$ — мгновенная плотность вероятности спайка, $y_T$ — выходная последовательность спайков на интервале $T$, описываемая временами генерации выходных спайков $t_q^{out}$.

Вероятность генерации выходного паттерна на интервале времени при условии наличия входного спайкового паттерна $\overline{x}_T$ можно увеличить с помощью минимизации информационной энтропии нейрона $h_T(y_T|\overline{x}_T) = -\ln(p\{y_T|\overline{x}_T\})$ по правилу градиентного спуска [30]. В работе [34] было показано, что с помощью алгоритма минимизации частной информационной энтропии нейрон может быть обучен детектировать определенные спайковые паттерны в зашумленном многомерном потоке спайков. Для спайкового нейрона, изменяющего напряжение на мембране согласно (3.1) и генерирующего выходные паттерны с плотностью

вероятности (3.2), изменение весов по направлению, противоположному градиенту информационной энтропии, на интервале $T$ вычисляется по формуле [30]:

$$\Delta w_{ij}^T = \gamma \sum_{t_q^{out} \in y_T} \sum_{t_{i,k}^{in} \in x_T^i} \frac{1}{\lambda(u(t_q^{out}))} \frac{\partial \lambda(u(t_q^{out}))}{\partial u} \alpha_j(t_q^{out} - t_{i,k}^{in}) - \sum_{t_{i,k}^{in} \in x_T^i} \int_T \frac{\partial \lambda(u(s))}{\partial u} \alpha_j(s - t_{i,k}^{in})\, ds, \tag{3.3}$$

где $t_q^{out} \in y_T$ — времена выходных спайков в требуемом учителем выходном паттерне $y_T$, $t_{i,k}^{in} \in x_T^i$ — времена входных спайков в $i$-том входном канале нейрона, индексы $i, j$ перечисляют $n \times 3$ весов нейрона, $\alpha_j(t)$ — альфа-функции, $\lambda(u)$ — мгновенная плотность вероятности спайка.

Выражение (3.3) можно представить в дифференциальной форме:

$$\begin{aligned} \frac{dw_{ij}}{dt} &= \gamma g_{ij}(t) \\ g_{ij}(t) &= \left( \sum_{t_q^{out} \in y_T} \frac{\delta(t - t_q^{out})}{\lambda(u(t))} - 1 \right) \sum_{t_k^{in} \in x_T^i} \frac{\partial \lambda(u(t_q^{out}))}{\partial u} \alpha_j(t - t_k^{in}). \end{aligned} \tag{3.4}$$

Функция $g_{ij}(t)$ является противоположной по знаку производной информационной энтропии по весу $w_{ij}$ в момент времени $t$. Тогда $\Delta w_{ij}^T = \int\limits_T g_{ij}(t)\, dt$. В этой форме особенно заметны аддитивные свойства энтропии на последовательных временных интервалах, позволяющие непрерывно обучать спайковый нейрон с учителем [34].

Сенсорные, а также модулирующие активность нейрона подкрепляющие (положительные и отрицательные) и стимулирующие входные сигналы приходят в управляющую нейронную сеть и должны быть сообщены нейронам сети в виде спайковых последовательностей. Разные входные сигналы несут разную смысловую нагрузку, в связи с этим спайковый нейрон имеет несколько типов входов (рис. 2).

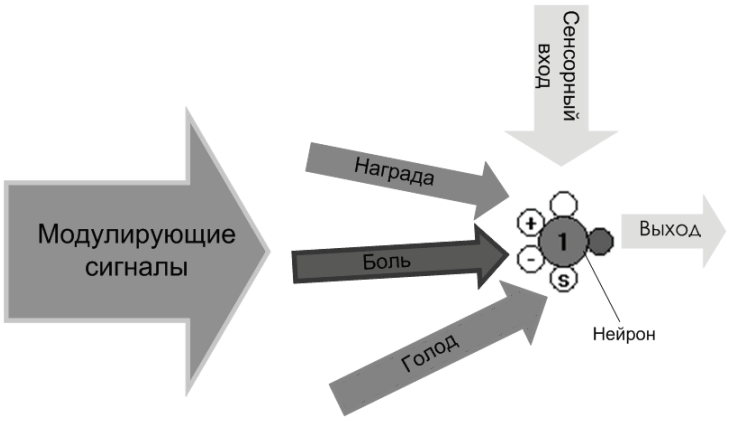


Рис. 2. Структура входов и выходов нейрона.

Сенсорный вход нейрона принимает многомерный поток спайков, который несет информацию о внешней и внутренней среде управляемого объекта, не связанную с модулирующими сигналами. Модулирующие активность нейрона сигналы от внешней среды, несущие

информацию о степени достижения целей функционирования робота, приходят на 3 типа дополнительных входов нейрона. На входы, обозначенные «+» и «−», приходят спайки, сообщающие о получении роботом положительного или отрицательного подкрепления. Приход спайков подкрепления сообщает нейрону, что недавно выполненные управляющей нейронной сетью действия дали определенный результат для управляемого объекта. После получения таких спайков в нейроне происходят процессы обучения, позволяющие нейрону в зависимости от знака награды повторить или избежать генерации выходного спайка при повторении близких входных сенсорных условий.

Третий тип модулирующего входного канала («s» от «stimulating» или «random search») служит для запуска режима исследовательского поведения управляемого объекта, реализуемого в виде случайного поиска управляющих действий. Спайк, приходящий на стимулирующий входной канал, добавляет к напряжению, вычисленному по формуле (3.1), смещение $\Delta u^{st}(t) = w^{st}\alpha^{st}(t-t^{st})$, где $\alpha^{st}(t), w^{st}$ — альфа-функция и вес стимулирующего входа, $t^{st}$ — время прихода стимулирующего спайка. Это приближает величину напряжения к пороговой величине, что может привести к генерации выходного спайка в случайный момент времени согласно распределению (3.2). Параметры альфа-функции и вес стимулирующего входа подобраны таким образом, чтобы в среднем обеспечить большую степень случайности времени генерации управляющего спайка и, следовательно, большую случайность поведения агента.

## 4. Модулированное снижение информационной энтропии

Спайки, приходящие на модулирующие входы, запускают механизмы обучения — изменения весов в сенсорных каналах нейрона. Обучение происходит только тогда, когда нейрон недавно сгенерировал выходной спайк, т. е. произвел действие, которое, возможно, изменило поведение управляемого объекта. Если несколько нейронов последовательно сгенерировали спайки, то наибольшее подкрепление должен получить тот нейрон, который сгенерировал спайк непосредственно перед получением подкрепления, чтобы в будущем ускорить получение награды.

Сигнал положительного подкрепления должен привести к таким изменениям весов, которые увеличат вероятность генерации нейроном близкого выходного паттерна при похожих сенсорных условиях. Сигнал отрицательного подкрепления должен заставить нейрон в дальнейшем избегать генерации недавно сгенерированного выходного паттерна при похожем сенсорном контексте.

Процесс обучения с подкреплением может быть формализован в виде процесса нахождения такой стратегии поведения, которая максимизирует получение суммарной награды в будущем [25]. В работе [26] построен алгоритм OLPOMDP максимизации награды при обучении с подкреплением в случае, если стратегия агента непрерывно зависит от некоторого конечного набора параметров. В работе [35] показано, что применение данного алгоритма для набора процессов принятия решений максимизирует награду, полученную всеми элементами сети. При формализации поведения стохастического спайкового нейрона в виде марковского процесса принятия решений за стратегию поведения принимают генерацию определенного спайкового паттерна в дискретном времени, а параметрами стратегии поведения служат веса нейрона [9, 11, 28, 29]. Корректное применение алгоритма OLPOMDP к спайковому нейрону, работающему в непрерывном времени по формулам (3.1) и (3.2), осуществлено в работе [11]. В обозначениях данной работы правила изменения веса нейрона

имеют вид

$$\begin{aligned} &\frac{dw_j}{dt} = \gamma r(t) z_j(t), \\ &\tau_z \frac{dz_j(t)}{dt} = -z_j(t) + \left( \sum_{t_k^{out} \in y} \frac{\delta(t - t_k^{out})}{\lambda(u(t))} - 1 \right) \frac{\partial \lambda(u(t))}{\partial w_j}, \end{aligned} \tag{4.1}$$

где $r(t)$ — значение сигнала подкрепления, $z_j(t)$ — функция сенсорной истории, $\tau_z$ — коэффициент забывания (насколько далеко в прошлое распространяется подкрепление), $y$ — выходной паттерн нейрона, состоящий из сгенерированных им спайков в моменты времени $t_k^{out}$, $\gamma$ — коэффициент обучения, $\delta(t)$ — дельта-функции.

В данной работе сигналы подкрепления приходят в виде спайков и описываются дельта-функциями: $r(t) = \sum\limits_{reward} \delta(t - t^r) - \sum\limits_{pain} \delta(t - t^p)$, где $t^r$ и $t^p$ — моменты прихода положительного и отрицательного подкрепления. Заметим, что для расширенного нейрона Spike Response Model (SRM) [2] с несколькими альфа-функциями [34] уравнения (4.1) с учетом (3.4) примут вид

$$\begin{aligned} &\frac{dw_{ij}}{dt} = \gamma \left( \sum_{reward} \delta(t - t^r) - \sum_{pain} \delta(t - t^p) \right) z_{ij}(t), \\ &\tau_z \frac{dz_{ij}(t)}{dt} = -z_{ij}(t) + g_{ij}(t). \end{aligned} \tag{4.2}$$

Решение уравнения для функции сенсорной истории $z_{ij}(t)$ является сверткой градиента информационной энтропии с фильтром $e^{-t/\tau_z}$: $z_{ij}(t) = \frac{1}{\tau_z} e^{-t/\tau_z} * g_{ij}(t)$. Функция сенсорной истории накапливает градиент энтропии в направлении максимизации вероятности генерации недавно сгенерированного паттерна $y$ при условии появления входного паттерна $\overline{x}$. При этом происходит забывание старой части паттерна $y$, и наибольшее влияние имеют недавно сгенерированные участки паттерна $y$. При приходе модулирующего сигнала вес изменяется пропорционально значению функции забывания. При положительном подкреплении происходит уменьшение энтропии недавно сгенерированного выходного паттерна (максимизация его вероятности), а при отрицательном подкреплении — увеличение его энтропии.

В работах [30, 33] показано, что при применении правил обучения (3.3) изменения весов приобретают схожесть с явлением Spike Timing-Dependent Plasticity (STDP) [36], наблюдаемом в реальных нейронах. При STDP вес синапса возрастает, если входной спайк пришел до выходного, и убывает, если входной спайк пришел после выходного. Экспоненциальное забывание сенсорной истории при применении правила минимизации энтропии (4.1) приводит к модуляции изменений весов (3.3) значениями сигнала подкрепления [11] — явлению Modulated STDP, также наблюдаемому в биологических нейронах и исследованному в работах [37–39].

На рисунке 3 изображено схематичное влияние модулирующих сигналов на процесс запоминания сенсорных паттернов.

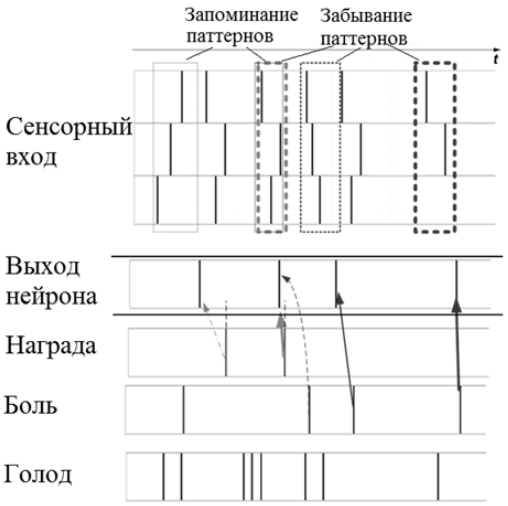


Рис. 3. Модуляция процесса запоминания сенсорных паттернов.

## 5. Описание тестового виртуального агента, управляемого спайковой нейронной сетью

Для тестирования структур управляющих спайковых нейронных сетей использовался модельный агент, существующий в простейшей виртуальной среде, которая представляет собой сеть из квадратных ячеек (размером $3 \times 3$) (рис. 4). Агент может свободно перемещаться между соседними ячейками. Границы всей сетки ячеек огорожены стенкой. Позиция агента в среде изображается черным кругом. Также в ячейках среды могут находиться ресурсы, потребляемые агентом. Позиция ресурса обозначается маленьким серым кругом. Описанная виртуальная среда аналогична использованной в работе [10], где применялся эвристический алгоритм максимизации подкрепления.

При старте моделирования агент имеет некоторое количество энергии для поддержания своей жизни. Со временем эта энергия расходуется и агент начинает «чувствовать голод», что отражается в изменении его цвета. Для пополнения запаса энергии агенту требуется найти ресурсы в виртуальной среде. Для потребления ресурса агенту нужно просто занять ячейку, в которой уже находится ресурс. После потребления ресурса агент приобретает некоторое количество энергии и на время перестает чувствовать голод. Задача агента — обеспечивать себя энергетическими ресурсами и не сталкиваться со стенками виртуальной среды.

Виртуальный агент управляется спайковой нейронной сетью, которая получает сенсорные и подкрепляющие сигналы в виде потоков спайков. На рисунке 4 показана управляющая нейронная сеть, состоящая из четырех спайковых нейронов. Информация о положении объектов в виртуальной среде поступает через две матрицы рецепторов (для положения агента и для положения ресурсов) — аналоги визуальных сенсоров агента. Размерность каждой матрицы рецепторов равна размерности виртуальной среды. Визуальный сенсор начинает генерировать спайки с некоторой частотой, если в соответствующей ячейке среды присутствует соответствующий объект. Матрицы визуальных рецепторов показаны справа от сетки виртуальной среды (рис. 4) в виде матриц серых кружков.

Информация о цели функционирования агента доступна нейронной сети посредством спайков модулирующих выходов агента. Модулирующие выходы показаны снизу от сетки

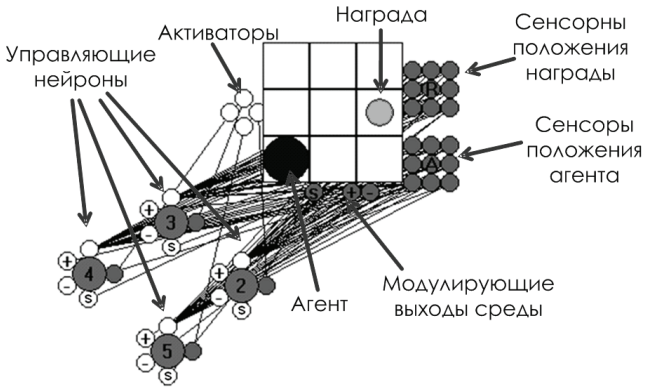


Рис. 4. Виртуальный агент, управляемый с помощью спайковой нейронной сети.

виртуальной среды (рис. 4) в виде серых кружков с метками «s», «+» и «−». При недостатке энергетического ресурса агент чувствует голод, стимулирующий сенсор голода, обозначенный меткой «s», начинает генерировать спайки с частотой, пропорциональной силе голода. Функциональное назначение сенсора голода — сообщить нейронной сети, что объекту требуется предпринять какие-то действия для достижения некоторой неизвестной сети цели. Сенсор голода соединен со стимулирующими входами случайного поиска («s») всех нейронов сети. Модулирующие выходы агента с метками «+» и «−» являются сенсорами положительного и отрицательного подкрепления соответственно. При получении подкрепления один из этих сенсоров (в соответствии со знаком подкрепления) генерирует выходной спайк.

Агент управляется с помощью подачи спайков на его активаторы, изображенные на рисунке 4 слева от сетки виртуальной среды в виде четырех белых кружков. Спайки, поданные на эти четыре активатора, интерпретируются агентом как команды на движение в одном из четырех направлений (вправо, влево, вверх, вниз) по сетке виртуальной среды.

Агент получает положительное подкрепление, если он поглощает ресурс. При этом модулирующий сенсор «+» генерирует спайк. Если агент получил команду двигаться за пределы сетки виртуальной среды, то это интерпретируется как столкновение со стенкой: агент получает отрицательное подкрепление (боль), а модулирующий сенсор «−» генерирует спайки. Агент также получает отрицательное подкрепление, если он получил управляющий сигнал, который он не в состоянии исполнить (например, одновременные спайки на активаторы «влево» и «вправо»).

Сенсор голода агента соединен с входами случайного поиска «s» всех нейронов. Модулирующие сенсоры агента соединены, соответственно, с модулирующими входами подкрепления «+» и «−» всех нейронов. Выходы нейронов соединены с активаторами агента таким образом, что определенная группа нейронов отвечает за определенную двигательную команду. Например, в сети, изображенной на рисунке 4, каждый из четырех нейронов отвечает за свою (одну из четырех) двигательную команду.

## 6. Описание процесса обучения виртуального агента

В начале экспериментов агент обладает некоторым количеством энергии и не испытывает голода. Модулирующие выходы агента не выдают спайков. Визуальные рецепторы агента генерируют спайки с некоторой фиксированной частотой. Однако веса сенсорных входов всех нейронов равны нулю, поэтому в сети не наблюдается никакой активности.

С течением времени внутреннее количество энергии агента уменьшается, и агент начинает испытывать чувство голода. При этом сенсор голода начинает генерировать выходные спайки с постепенно возрастающей частотой. Спайки от сенсора голода поступают на входы «s» случайного поиска всех нейронов. Чем больше частота этих спайков, тем чаще нейроны генерируют выходные спайки в случайные моменты времени. Эти случайные спайки поступают на активаторы агента и заставляют случайно перемещаться по сетке виртуальной среды. В процессе случайного перемещения агент может занять ячейку с находящимся там ресурсом, тем самым получив необходимую энергию. При нахождении ресурса сенсор положительного подкрепления агента генерирует спайки. Эти спайки приходят на модулирующие входы «+» всех нейронов и запускают обучение всех нейронов по правилу (4.1) при $r(t) = \delta(t^r)$. Напомним, что визуальные сенсоры агента все время информируют нейронную сеть о положении ресурса и самого агента. Поэтому все недавно активные нейроны увеличивают вероятность генерации выходного спайка при наличии похожего сенсорного контекста, т. е. при похожем взаимном положении агента и ресурса. Тот нейрон, чей случайно сгенерированный управляющий спайк непосредственно привел к поглощению ресурса, получит наибольшее количество положительного подкрепления. Если действия какого-либо нейрона привели к столкновению агента со стенкой виртуальной среды или если несколько нейронов сгенерировали команды движения одновременно, то сенсор негативного подкрепления агента генерирует спайк, который приходит на модулирующие выходы «−» всех нейронов. В результате прихода этого спайка веса недавно активных нейронов изменяются по правилу (4.1) при $r(t) = -\delta(t^p)$, заставляя понизить вероятность их активности при будущем повторении положений агента и ресурса, предшествующих получению боли. В результате многократного повторения получения сигналов подкрепления различных знаков вследствие случайного блуждания агента по виртуальной среде управляющие нейроны учатся генерировать спайки на основе сенсорного контекста о взаимном положении агента и ресурса. Если обучение было успешным, то управляющие спайки приводят агента к ресурсу и не позволяют ему ударяться о стенки среды.

## 7. Различные конфигурации управляющих сетей

Успешность обучения управлению в описанной виртуальной среде сильно зависит от архитектуры сети, от частоты поступления визуальной сенсорной информации и характерного времени интегрирования сенсорной истории, зависящей, в свою очередь, от форм альфа-функций. Наиболее простой архитектурой управляющей сети является сеть, изображенная на рисунке 4. Данная сеть состоит всего из четырех нейронов, каждый из которых отвечает за определенную двигательную команду («вверх», «вниз», «влево», «вправо»). Она успешно решает задачу поиска ресурса на сетке размера $3 \times 3$, при условии достаточно большой частоты поступления спайков от визуальных рецепторов по сравнению со временем затухания альфа-функций постсинаптического потенциала. Видео с обученным агентом с четырьмя нейронами доступно по адресу http://www.youtube.com/watch?v=yl2rcSKHfLU. В этом случае каждый активный нейрон при получении подкрепления обладает информацией о предыдущих положениях ресурса и агента, так как постсинаптические потенциалы от спайков визуальных сенсоров еще не успели затухнуть. Обучение такой сети можно сравнить с обучением искусственных стационарных нейронных сетей (например, персептронов), так как в этом случае нейронам достаточно использовать только пространственную сенсорную информацию о положении объектов без учета истории их движения, т. е. без учета временной составляющей сенсорных спайковых паттернов. Например, нейрон, генерирующий коман-

ду «вверх», обучается генерировать спайк в случаях, когда агент находится ниже ресурса. Веса визуальных рецепторов, отвечающих за положения агента около верхней стенки, становятся отрицательными, не позволяя нейрону сгенерировать спайк, приводящий к удару о стенку. Аналогично со стационарными нейронными сетями скорость и качество обучения можно увеличить, если увеличить количество нейронов — пространственных опорных функций, аппроксимирующих оптимальную функцию управления агентом.

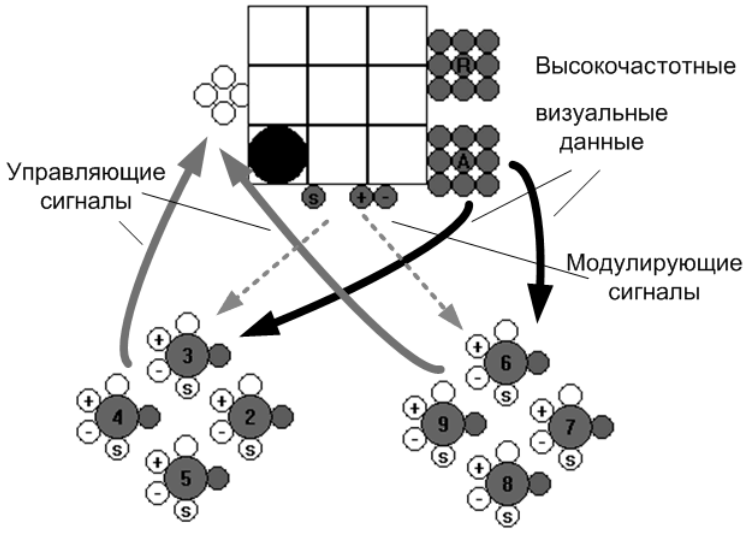


Рис. 5. Управляющая сеть из 8 нейронов, обучающаяся только пространственным ассоциациям.

На рисунке 5 изображена управляющая сеть из 8 нейронов, использующая только пространственные ассоциации между позициями объектов и управляющими сигналами. Данная сеть в среднем обучается немного быстрее, чем сеть из 4 нейронов, однако значительных улучшений поведения агента не наблюдается (http://www.youtube.com/watch?v=5_8oLnpDeW4). Генерирование управляющего сигнала синхронизировано с получением визуальной информации.

Если частота получения визуальных стимулов низка, то сеть нейронов на рисунке 5 практически перестает обучаться правильному управлению агентом, так как часто нейрон получает подкрепляющий сигнал без обладания актуальной пространственной информацией о положении объектов. Корректное обучение после положительного подкрепления происходит очень медленно. Агент вынужден чувствовать голод очень часто, что приводит к увеличению интенсивности случайного поиска поведения и более частому получению негативного подкрепления вследствие удара о стенки или двусмысленных команд, что, в свою очередь, приводит к быстрому забыванию всех положительных пространственных ассоциаций. Для успешного обучения управлению требуется снабдить нейроны дополнительной информацией, что можно сделать, соединив все нейроны друг с другом (рис. 6). В этом случае нейроны сети обладают информацией о недавней активности других нейронов. Кроме того, для успешного обучения агента пришлось добавить еще одну группу нейронов, организованную в виде второго слоя. Эксперименты (http://www.youtube.com/watch?v=SfUHtyLDv80) показывают, что нейроны в такой сети учатся вырабатывать небольшие цепочки управляющих сигналов, которые запускаются визуальными стимулами и сигналом голода. Управляющие команды генерируются чаще, чем приходят визуальные стимулы (агент может сделать несколько движений между приходом последовательных визуальных стимулов).

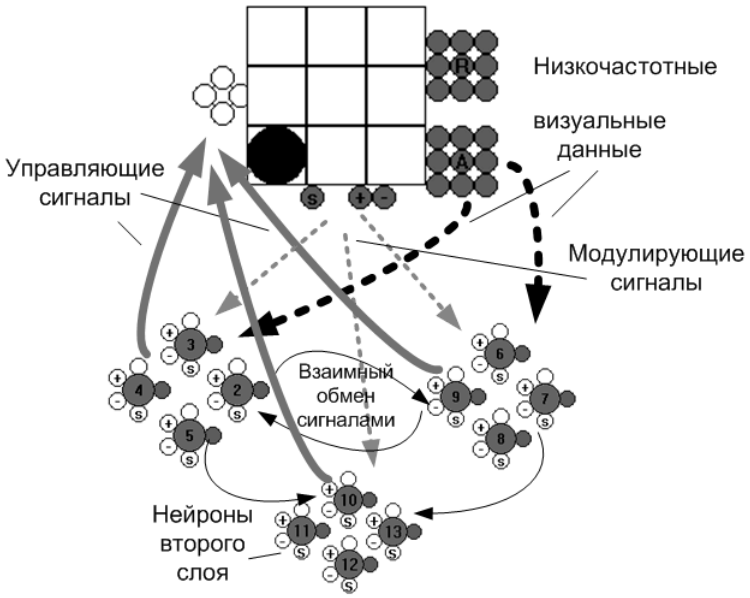


Рис. 6. Двухслойная сеть из 12 нейронов с взаимными соединениями, обучающаяся пространственно-временным ассоциациям.

Данная сеть запоминает не только пространственные, но и временные ассоциации между сенсорными сигналами, которые хранятся во внутренней активности сети. Второй слой нейронов анализирует паттерны активности нейронов первого слоя и обучается без использования визуальных сигналов. Следует отметить, что обратная сенсорная связь регулирует среднюю активность сети. При малой нейронной активности, не позволяющей агенту добывать награду, активизируются стимулирующие сигналы голода, поднимающие активность сети. При слишком большой активности нейроны генерируют большое количество управляющих сигналов, приводящее к получению сигнала боли, что, в свою очередь, понижает веса нейронов и снижает общую активность.

## 8. Управление агентом в виртуальной среде большой размерности

В теории обучения с подкреплением существует дилемма между исследованием окружающей среды и использованием уже накопленных знаний (в иностранной литературе «exploration-exploitation dilemma» [32]). В момент принятия решения объект может использовать уже накопленные в процессе обучения знания об окружающей среде и выбрать оптимальное, по его мнению, действие. Однако накопленные знания агента часто не совсем точно отражают реальную динамику внешней среды. Возможно, существует действие, которое приведет к еще большей награде в будущем, т. е. агенту иногда следует выбрать действие наугад («исследовать среду»), чтобы обнаружить действительно оптимальную политику. Однако слишком частые исследования будут приносить гораздо меньше награды, чем простое использование накопленных знаний. Описанные проблемы ярко проявляются при обучении управлению агентом в виртуальных средах больших размерностей. Параметром, определяющим степень исследовательского поведения агента, является параметр стохастичности

нейронов. При большой стохастичности нейронов выбранное действие агента достаточно детерминировано только при длительном обучении, когда агент набрал достаточно опыта функционирования в среде. При размере среды $10 \times 10$ количество возможных положений награды и агента возрастает до 9 900. Исследовать все пространство положений возможно только при достаточном уровне шума в нейронах. Если шум в нейронах мал, то агент выбирает субоптимальную политику — всегда оставаться на месте. Такая политика не приносит агенту боли, так как он никогда не ударяется о стены и не испытывает противоречивого управления. Но, естественно, такая политика не позволяет нейрону находить награду.

При исследовании с помощью случайных управляющих сигналов агент гораздо чаще ударяется о стены, чем находит награду. Во всех экспериментах в среде размера $10 \times 10$ агент выбирал субоптимальную политику — всегда оставаться на месте. Для того чтобы все же обучить нейронную сеть управлять агентом в этой среде, влияние негативного подкрепления на обучение нейронов было снижено на несколько порядков. Это было реализовано путем применения двух разных коэффициентов обучения $\gamma$ в выражении (4.1). При приходе спайка положительного подкрепления применялся коэффициент $\gamma^+ = 0.005$, а при приходе спайка отрицательного подкрепления применялся коэффициент $\gamma^- = 0.01\gamma^+$.

На рисунке 7 слева показана виртуальная среда размером $10 \times 10$. Агент управляется 32 нейронами. Каждый нейрон получает спайки от $10 \times 10 \times 2 = 200$ визуальных сенсоров. Нейроны разбиты на 4 группы, каждая группа ответственна за движение агента в одном из четырех направлений. Справа на рисунке 7 показаны графики получаемых агентом награды и боли в процессе обучения. Для исследования большой внешней среды сеть сначала поднимает уровень средней активности нейронов, что приводит агента к получению большего количества болевых сигналов (начальная часть верхнего графика серого цвета). Это связано с тем, что при большой активности нейронов агент часто получает сразу несколько управляющих спайков, а также часто сталкивается со стенами. После нахождения близкой к оптимальной политики управления количество наград возрастает (нижний график черного цвета). Далее происходит подстройка управления, направленная на снижение количества получаемой боли (конечная часть верхнего графика серого цвета). При этом количество получаемой награды остается практически постоянным. Видео с поведением обученного агента представлено на http://www.youtube.com/watch?v=aEoVvnr7OYk.

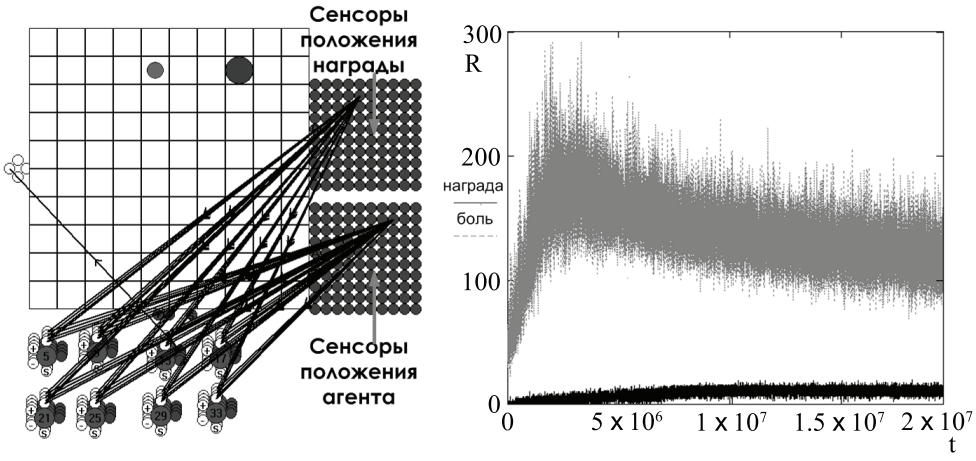


Рис. 7. Вид виртуальной среды размером $10 \times 10$ (слева). Графики получаемой агентом награды и боли в процессе обучения (справа).

## 9. Заключение

В работе показано, как с помощью спайковой нейронной сети можно построить самообучающуюся систему управления виртуальным агентом. Нейронная сеть не обладала априорной информацией о внешней среде, однако с помощью сигналов подкрепления из внешней среды научилась управлять объектом с учетом его потребностей с целью максимизации получаемой им награды. При получении подкрепления управляющие нейроны запоминают (или забывают) сенсорные импульсные паттерны, ориентируясь на свою недавнюю активность. При этом сеть обучается переводить входную многомерную пространственно-временную информацию от датчиков объекта в выходную управляющую информацию, направляемую на активаторы объекта, таким образом, чтобы в будущем увеличить количество получаемого агентом подкрепления. Основой для законов обучения с подкреплением спайковой нейронной сети является применение информационной энтропии как оценочной функции преобразования стохастическим нейроном спайковых паттернов: метод минимизации энтропии с забыванием позволил получить правила изменения весов нейрона. Способность спайковых нейронов использовать не только пространственную, но и временную составляющую входных сигналов позволила управлять агентом при отсутствии полной пространственной картины окружающей среды.

## Список литературы

[1] Nicholls J. G., Martin A. R., Wallace B. G., Fuchs P. A. From neuron to brain: A cellular and molecular approach to the function of the nervous system. 4th ed. Sunderland, MA: Sinauer Associates, 2001. 679 pp. [Николлс Дж. Г., Мартин А. Р., Валлас Б. Дж., Фукс П. А. От нейрона к мозгу. М.: УРСС, 2003. 688 с.]

[2] Gerstner W., Kistler W. M. Spiking neuron models: Single neurons, populations, plasticity. Cambridge: Cambridge Univ. Press, 2002. 494 pp.

[3] Melamed O., Gerstner W., Maass W., Tsodyks M., Markram H. Coding and learning of behavioral sequences // Trends in Neurosciences, 2004, vol. 27, no. 1, pp. 11–14.

[4] Maas W. Networks of spiking neurons: The third generation of neural network models // Trans. Soc. Comput. Simul. Int., 1997, vol. 14, no. 4, pp. 1659–1671.

[5] Rieke F., Warland D., de Ruyter van Steveninck R., Bialek W. Spikes: Exploring the neural code. (Computational Neurosciences series.) Cambridge, MA: MIT Press, 1997. 395 pp.

[6] Di Paolo E. A. Spike-timing dependent plasticity for evolved robots // Adaptive Behavior, 2002, vol. 10, nos. 3–4, pp. 243–263.

[7] Saggie K., Keinan A., Ruppin E. Solving a delayed response task with spiking and McCulloch–Pitts agents // Advances in Artificial Life: Proc. of the 7th European Conf. on Artificial Life (ECAL) (Dortmund, Germany, 2003) / W. Banzhaf, T. Christaller, P. Dittrich, J. T. Kim, J. Ziegler (Eds.). Berlin–Heidelberg: Springer, 2003. Vol. 2801, pp. 199–208.

[8] Antonelo E. A., Schrauwen B., Stroobandt D. Mobile robot control in the road sign problem using Reservoir Computing networks // IEEE Internat. Conf. on Robotics and Automation (ICRA) (Pasadena, CA, 2008) / S. Hutchinson et al. (Ed.). Pasadena, CA, 2008. P. 911–916.

[9] Queiroz M. S., Braga A., Berredo R. C. Reinforcement learning of a simple control task using the spike response model // Neurocomputing, 2006, vol.,70, nos. 1–3, pp. 14–20.

[10] Lee K., Kwon D.-S. Synaptic plasticity model of a spiking neural network for reinforcement learning // Neurocomputing, 2008, vol. 71, no. 13, pp. 3037–3043.

[11] Florian R. V. A reinforcement learning algorithm for spiking neural networks // SYNASC'05: Proc. of the 7th Internat. Symp. on Symbolic and Numeric Algorithms for Scientific Computing (Timisoara, Romania, 2005). Timisoara, 2005. P. 299–306.

[12] Florian R. V. Spiking neural controllers for pushing objects around // From Animals to Animats 9 (SAB'06): Proc. of the 9th Internat. Conf. on the Simulation of Adaptive Behavior (Rome, Italy, 2006) / S. Nolfi, G. Baldassare, R. Calabretta, J. Hallam, D. Marocco, O. Miglino, J.-A. Meyer, D. Parisi. (Lecture Notes in Artificial Intelligence, vol. 4095.) Berlin–Heidelberg: Springer, 2006. P. 570–581.

[13] Burgsteiner H. Training networks of biological realistic spiking neurons for real-time robot control // Proc. of the 9th Internat. Conf. on Engineering Applications of Neural Networks (Lille, France, 2005). Lille, 2005. P. 129–136.

[14] Floreano D., Zufferey J.-C., Mattiussi C. Evolving spiking neurons from wheels to wings // Dynamic Systems Approach for Embodiment and Sociality, 2003, vol. 6, pp. 65–70.

[15] Wiles J., Ball D., Heath S., Nolan C., Stratton P. Spike-time robotics: A rapid response circuit for a robot that seeks temporally varying stimuli // Australian J. of Intelligent Information Processing Systems, 2010, vol. 11, no. 1, 10 pp.

[16] Damper R. I., French R. L. B., Scutt T. W. ARBIB: An autonomous robot based on inspirations from biology // Robotics and Autonomous Systems, 1998, vol. 31, no. 4, pp. 247–274.

[17] Alnajjar F., Murase K. A simple Aplysia-like spiking neural network to generate adaptive behavior in autonomous robots // Adaptive Behavior, 2008, vol 16, no. 5, pp. 306–324.

[18] Soula H., Alwan A., Beslon G. Learning at the edge of chaos: Temporal coupling of spiking neurons controller for autonomous robotic // Proc. of American Association for Artificial Intelligence (AAAI) Spring Symposia on Developmental Robotic (Stanford, CA, 2005) / D. Bank, L. Meeden (Eds.). Menlo Park, CA: AAAI Press, 2005. 6 pp.

[19] Nolfi S., Floreano D. Synthesis of autonomous robots through evolution // Trends in Cognitive Sciences, 2002, vol. 6, no. 1, pp. 31–37.

[20] Joshi P., Maass W. Movement generation with circuits of spiking neurons // Neural Computation, 2005, vol. 17, no. 8, pp. 1715–1738.

[21] Carrillo R., Ros E., Boucheny C., Coenen O. J.-M. D. A real-time spiking cerebellum model for learning robot control // Biosystems, 2008, vol. 94, nos. 1–2, pp. 18–27.

[22] Boucheny Ch., Carrillo R., Ros E., Coenen O. J.-M. D. Real-time spiking neural network: An adaptive cerebellar model // Proc. of the 8th Internat. Work-Conf. on Artificial Neural Networks, Computational Intelligence and Bioinspired Systems / J. Cabestany, A. Prieto, F. Sandoval Hernández (Eds.). (Lecture Notes in Computer Science, vol. 3512.) Berlin–Heidelberg: Springer, 2005. P. 136–144.

[23] Manoonpong P., Woegoetter F., Pasemann F. Biological inspiration for mechanical design and control of autonomous walking robots: Towards life-like robots // The International Journal of Applied Biomedical Engineering (IJABME), 2010, vol. 3, no. 1, pp. 1–12.

[24] Maass W., Natschlager T., Markram H. Real-time computing without stable states: A new framework for neural computation based on perturbations // Neural Computations, 2002, vol. 14, no. 11, pp. 2531–2560.

[25] Sutton R.S., Barto A.G. Reinforcement learning: An introduction. Cambridge, MA: MIT Press, 1998. 323 pp.

[26] Baxter J., Weaver L., Bartlett P. L. Direct gradient-based reinforcement learning: II. Gradient ascent algorithms and experiments: Technical report. Australian National University, Research School of Information Sciences and Engineering. 1999.

[27] Bellman R. A Markovian decision process // J. Math. Mech., 1957, vol. 6, pp. 679–684.

[28] Farries M. A., Fairhall A. L. Reinforcement learning with modulated spike timing-dependent synaptic plasticity // Neurophysiol., 2007, vol. 98, pp. 3648–3665.

[29] Baras D., Meir R. Reinforcement learning, spike-time-dependent plasticity and the BCM rule // Neural Computation, 2007, vol. 19, no. 8, pp. 2245–2279.

[30] Синявский О. Ю., Кобрин А. И. Использование информационных характеристик потока импульсных сигналов для обучения спайковых нейронных сетей // Интегрированные модели

и мягкие вычисления в искусственном интеллекте (Коломна, 2009): Сб. научн. тр.: Т. 2. М.: 2009, с. 678–687.

[31] Levine M.W. and Shefner, J.M. Fundamentals of sensation and perception. 2nd ed. Pacific Grove, CA: Brooks/Cole, 1991. 675 pp.

[32] Rejeb L., Guessoum Z. and M'Hallah R. An adaptive approach for the exploration-exploitation dilemma for learning agents // Multi-Agent Systems and Applications IV: 4th Internat. Central and Eastern European Conf. on Multi-Agent Systems (Budapest, Hungary, 2005): Proc. CEEMAS 2005 / M. Pechoucek, P. Petta, L. Zsolt Varga (Eds.). (Lecture Notes in Comput. Sci., vol. 3690.) Berlin: Springer, 2005. P. 316–325.

[33] Pfister J. P., Toyoizumi T., Barber D., Gerstner W. Optimal spike-timing-dependent plasticity for precise action potential firing in supervised learning // Neural Comput., 2006, vol. 18, no. 6, pp. 1318–1348.

[34] Синявский О. Ю., Кобрин А. И. Обучение спайкового нейрона с учителем в задаче детектирования пространственно-временного импульсного паттерна // Нейрокомпьютеры: разработка и применение, 2010, т. 8, с. 69–76.

[35] Bartlett P. L., Baxter J. A biologically plausible and locally optimal learning algorithm for spiking neurons. http://arp.anu.edu.au/ftp/papers/jon/brains.pdf.gz (2000).

[36] Bi G. Q., Poo M. M. Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type // The Journal of Neuroscience, 1998, vol. 18, no. 24, pp. 10464–10472.

[37] Legenstein R. ,Pecevski D., Maass W. A learning theory for reward-modulated Spike-Timing-Dependent Plasticity with application to biofeedback // PLoS Comput. Biol., 2008, vol. 4, no. 10, e1000180.

[38] Izhikevich E. M. Solving the distal reward problem through linkage of STDP and dopamine signaling // Cerebral Cortex, 2007, vol. 17, pp. 2443–2452.

[39] Frémaux N., Sprekeler H., Gerstner W. Functional requirements for reward-modulated spike-timing-dependent plasticity // The Journal of Neuroscience, 2010, vol. 30, no. 40, pp. 13326–13337.

## Reinforcement learning of a spiking neural network in the task of control of an agent in a virtual discrete environment

Oleg Y. Sinyavskiy[1], A. I. Kobrin[2]

National Research University "Moscow Power Engineering Institute"
Krasnokazarmennaya st. 14, Moscow, 111250, Russia
[1]sinyavskiyoleg@gmail.com, [2]KobrinAI@mpei.ru

Method of reinforcement learning of spiking neural network that controls robot or virtual agent is described. Using spiking neurons as key elements of a network allows us to exploit spatial and temporal structure of input sensory information. Teaching of the network is implemented with a use of reinforcement signals that come from the external environment and reflect the success of agent's recent actions. A maximization of the received reinforcement is done via modulated minimization of neurons' informational entropy that depends on neurons' weights. The laws of weights changes were close to modulated synaptic plasticity that was observed in real neurons. Reinforcement learning algorithm was tested on a task of a resource search in a virtual discrete environment.

MSC 2010: 68T05, 68Q32
Keywords: spiking neuron, adaptive control, reinforcement learning, informational entropy

Received November 29, 2011, accepted December 13, 2011
Citation: *Rus. J. Nonlin. Dyn.*, 2011, vol. 7, no. 4 (Mobile Robots), pp. 859–875 (Russian)