УДК 658.62.018.012 DOI 10.21685/2307-4205-2019-4-4

А. Н. Дарьина, И. В. Прокопьев

МЕТОД НЕЙРОСЕТЕВОГО УПРАВЛЕНИЯ В РЕАЛЬНОМ ВРЕМЕНИ НА ОСНОВЕ СИНТЕЗА ФУНКЦИИ ВЫБОРА

A. N. Daryina, I. V. Prokopiev

REAL-TIME NEURAL NETWORK CONTROL METHOD BASED ON THE SYNTHESIS OF THE COST FUNCTION

Аннотация. При работе с беспилотным транспортным средством обычно ставится задача нахождения такого управления, которое осуществляет его переход из начального состояния в заданное конечное. Разрабатываемые законы управления должны обеспечивать требуемые показатели качества (точность, быстродействие и т.п.) по всем управляемым координатам с учетом заданных ограничений на управление и состояние беспилотного транспортного средства. С практической точки зрения очень важно, чтобы эти законы управления были оптимальными по отношению к заданному функционалу качества. Однако в ряде случаев этого недостаточно и требуется синтезировать законы управления, обеспечивающие достижение цели управления в широком классе неопределенности модели динамики беспилотного транспортного средства. При этом система управления должна обеспечить управление с учетом динамических ограничений в реальном времени, параметры которых заранее не известны. Решение таких задач классическими методами осложняется тем, что требуется большое количество вычислений, и они не могут быть реализованы на борту робота в процессе его эксплуатации. В данной статье рассматривается решение задачи квазиоптимального управления в реальном времени, для этого объединяются традиционный подход к синтезу законов управления и приближение нейронных сетей к этим законам с учетом информации, которую традиционный подход не учитывает. В результате метод состоит в последовательном выполнении трех этапов: генерации траекторий, синтезе функции выбора и обучении нейронной сети. Функция выбора определяется эволюционными методами символьной регрессии и учитывает динамические ограничения в реальном времени. Для этого используется нейросетевое представление алгоритмов программного, адаптивного и интеллектуального управления, т.е. объединенная сеть, в которой нейронная сеть прямого распространения подражает, например, пропорциональному интегральному дифференциальному регулятору, а сверточная применяется для определения характеристик дороги и принятия правильных решений по управлению. Приводится пример применения метода на реальном роботе.

Abstract. When working with an unmanned vehicle, the task is usually to find such a control that carries out its transition from the initial state to the given final state. The developed control laws should provide the required quality indicators (accuracy, speed, etc.) for all controlled coordinates, taking into account the set control restrictions and the condition of the unmanned vehicle. From a practical point of view, it is very important that these control laws are optimal with respect to a given quality functional. However, in some cases this is not enough and it is necessary to synthesize control laws that ensure the achievement of the control goal in a wide class of uncertainty of the dynamics model of an unmanned vehicle. In this case, the control system should provide control taking into account dynamic constraints in real time, the parameters of which are not known in advance. The solution of such problems by classical methods is complicated by the fact that a large number of calculations are required, and they cannot be implemented on board the robot during its operation. This article discusses the solution of the quasi-optimal control problem in real time; for this, the traditional approach to the synthesis of control laws and the approximation of neural networks to these laws are combined taking into account information that the traditional approach does not take into account. As a result, the method consists in sequentially performing three stages: generating trajectories, synthesizing the selection function and training the neural network. The selection function is determined by the evolutionary methods of symbolic regression, and takes into account dynamic constraints in real time. For this, a neural network representation of the algorithms of software, adaptive and intelligent control is used, i.e. an integrated network in which a direct distribution neural network imitates, for example, a proportional integral differential controller, and a convolutional network is used to determine the characteristics of the road and make the right management decisions. An example of applying the method to a real robot is given.

© Дарьина А. Н., Прокопьев И. В., 2019

Ключевые слова: беспилотное транспортное средство, нейросетевое управление, генератор траекторий, синтез функции выбора, оптимальное управление.

Keywords: unmanned vehicle, neural network control, trajectory generator, synthesis of the cost function, optimal control.

Введение

Целью управления беспилотного транспортного средства (БТС) обычно является перевод БТС из начального состояния в заданное конечное состояние или осуществление (отслеживание) заданного программного движения БТС. Синтезируемые законы управления должны обеспечивать требуемые показатели качества (точность, быстродействие и т.п.) по всем управляемым координатам с учетом заданных ограничений на управление и состояние БТС. Кроме того, практически важно, чтобы эти законы управления были оптимальными по отношению к заданному функционалу качества. Однако в ряде случаев этого недостаточно и требуется синтезировать законы управления, обеспечивающие достижение цели управления в широком классе неопределенности модели динамики БТС. С другой стороны, система управления должна обеспечить управление с учетом динамических ограничений в реальном времени, параметры которых заранее не известны.

Наиболее общий подход к решению задач управления в реальном времени основан на использовании метода предсказания траектории, обеспечивающей возможность добиться прогресса на пути достижения цели [1]. Основная проблема реализации этого подхода состоит в том, что при моделировании необходимо знать не только информацию о модели, но и информацию об окружающей среде. В работе [1] предлагается генерировать множество траекторий с различным управлением и выбирать их по дополнительному функционалу, который учитывает внешние факторы, в том числе и динамические фазовые ограничения. Данный функционал определяется разработчиком системы управления и ставится на борт объекта до реализации алгоритма управления. Построение такой функции требует существенного предварительного анализа внешней среды и сопряжено с возможностью внесения корректировки при выборе траектории.

Известные подходы к предотвращению столкновений для мобильных роботов можно условно разделить на две категории: глобальные и локальные. Преимущество глобальных методов, таких как дорожная карта, декомпозиция клеток и методов потенциального поля [2], заключается в том, что полная траектория от начальной точки до целевой точки может быть вычислена в реальном времени, но они не подходят для быстрого предотвращения препятствий.

Локальные или реактивные подходы [3–6] используют лишь небольшую часть пространства состояния. Это приводит к очевидному недостатку, что они не могут вычислять оптимальные решения для всей траектории. Большинство этих локальных подходов генерируют команды движения для робота на двух отдельных этапах [1, 6]. На первом этапе определяется желаемое направление движения. На втором этапе генерируются команды управления, приводящие к движению в нужном направлении. Вычисление направления движения должно соответствовать глобальному направлению с учетом ограничений. При синтезе управления БТС при таком подходе необходимо решить, насколько достаточны ресурсы по управлению для достижения тех или иных областей пространства из текущего или заданного состояния и какой вид имеет траектория движения, если такие области достижимы.

В настоящей работе рассматривается подход, при котором функция для отбора траекторий определяется эволюционными методами символьной регрессии [7–9]. Для управления используется глубокое сверточное и трансферное обучение для определения характеристик дороги и принятия правильных решений по управлению. Сверточная сеть, объединенная с сетью прямого распространения, отображает характеристики дороги и характеристики маневров, в результате выход нейросетевого алгоритма получает решение по управлению по всем управляемым координатам с учетом заданных ограничений на управление и состояние в реальном времени с учетом оптимальности по отношению к заданному функционалу качества.

Постановка задачи

Рассмотрим объект, поведение которого описывается системой n обыкновенных дифференциальных уравнений

$$\dot{x} = f(x,u), \tag{1}$$

где $x \in \mathbb{R}^n$ – вектор состояния объекта контроля; $u \in U \subseteq \mathbb{R}^m$ – вектор управления, $x = [x_1, x_2, ..., x_n]^T$; $u = [u_1, u_2, ..., u_m]^T$; $f(x,u) = [f_1(x,u), f_2(x,u), ..., f_n(x,u)]^T$; $m \le n$.

В общем случае мы предполагаем, что значения переменной времени t принадлежат промежутку $[0, t^+]$, заданы начальные условия $x(0) \in \mathbb{R}^n$ и время $t_f \in [0, t^+]$, попадания в терминальные условия, заданные соотношением

$$x(t_f) = x^{t_f} \in \mathbb{R}^n. \tag{2}$$

Заданы статические $\varphi_i(x^j) \le 0$, $i = 1, ..., r$ и динамические $\gamma(x^p, x^s) \le 0$, $p = 1, ..., K-1$ $s = p+1, ..., K$ ограничения.

В рассматриваемых условиях ставится задача определения такого вектора управлений $u(t)$ и соответствующего ему вектора фазовых координат $x(t)$, которые бы за время всей миссии достигали терминального условия (2) и доставляли экстремальное значение некоторому фиксированному функционалу, конкретный вид и физический смысл которого определяются характером исследуемого объекта:

$$J = \int_0^{t_f} f_0(x,u)\, dt \to \min. \tag{3}$$

Метод нейросетевого управления

Рассмотрим особенности постановки задачи нейросетевого управления БТС. Сложность решения рассматриваемой задачи классическими методами (метод Понтрягина [10], метод редукции к задаче нелинейного программирования и другие эвристические методы) заключается в том, что требуется большое количество вычислений, и он не может быть реализован на борту робота в процессе его эксплуатации.

В работе в отличие от вышеизложенного рассматривается метод нейросетевого управления БТС с функцией выбора, учитывающей динамические ограничения в реальном времени. Для этого необходимо нейросетевое представление алгоритмов программного, адаптивного и интеллектуального управления. В данном случае предлагается использовать объединенную сеть, в которой нейронная сеть прямого распространения подражает, например, пропорциональному интегральному дифференциальному (ПИД) регулятору, и сверточную сеть, сочетающую геометрическую абстракцию с помощью модуля сверточных слоев (рис. 1, 2).

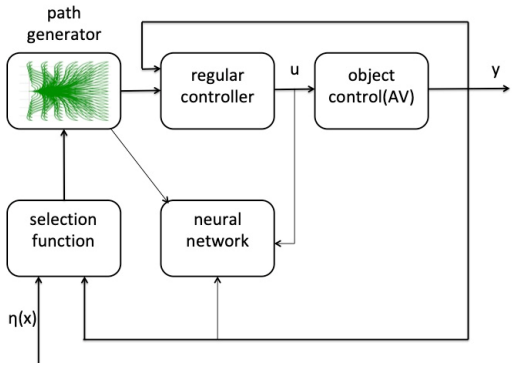

Рис. 1. Схема подражающего нейроуправления с функцией выбора. Режим обучения

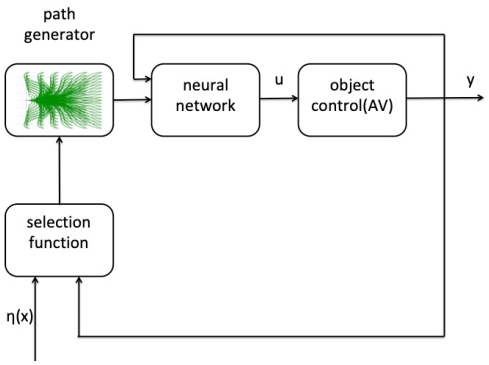

Рис. 2. Схема подражающего нейроуправления с функцией выбора

Математическая модель многослойного перцептрона (на примере трехслойного) с одним выходом задается уравнением

$$\overline{y} = Y(x) = f_3\left\{\omega^3_{b\,l} + \sum_{q=1}^{k_3} \omega^3_q \cdot f_2\left[\omega^2_{b\,q} + \sum_{j=1}^{k_2} \omega^2_{j\,q} \cdot f_1\left\{\omega^1_{b\,j} + \sum_{i=1}^{k_1} \omega^1_{i\,j} \cdot x_i\right\}\right]\right\},$$

где $f_i(*)$ – функция активации i-го слоя; $\omega_{b\,j}^{i}$ – вес смещения j-го нейрона i-го слоя; $\omega_{i\,j}^{k}$ – вес i-го входа j-го нейрона k-го слоя; x_i – входной сигнал на i-м входе нейронной сети. Такая нейронная сеть обладает аппроксимирующими свойствами. Однако не существует однозначной методики выбора числа слоев и нейронов в каждом слое, а также функции активации.

Рассмотрим математическую модель сверточного слоя (остальные сверточные слои формируются подобным образом): каждый элемент карты признаков соединен с областью размером 3×3 на входном изображении. Следственно, каждый элемент карты имеет девять обучаемых коэффициентов и обучаемый сдвиг. Значение элемента карты вычисляется по формуле

$$X_i^{h,l} = f\left(\sum_{k=1}^{n_l-1}\sum_{j=-\infty}^{\infty} X_{i-j}^{k,l} W_{i-j,i}^{h,k,l} + B_i^{h,l}\right),$$

где $X_i^{h,l}$ – значение элемента i в карте признаков h слоя l; n_l – количество карт признаков в слое l; $B_i^{h,l}$ – значение сдвига для элемента i в карте признаков слоя l; W – синаптический вес связи между элементом i в карте признаков h слоя l и элементом $i-j$ карты k слоя $l-1$.

В этом случае объединяют традиционный подход к синтезу законов управления и приближение нейронных сетей к этим законам с учетом информации, которую традиционный подход не учитывает. Для этого нужно получить функцию выбора эволюционным методом.

В результате метод состоит в последовательном выполнении трех этапов: генерации траекторий, синтеза функции выбора и обучения нейронной сети.

Генерация траекторий

Для решения задачи синтеза системы управления движением объекта по пространственной траектории используем генератор траекторий. В данном подходе на основе модели предсказания траектории на первом этапе выполняется численная оптимизация, которая минимизирует функционал качества (3). Желательно установить начальные предположения параметров управления близко к фактическому решению, тем меньше итераций алгоритма необходимо для страхования от попадания в неверный минимум. Исторически аппроксимации решений были найдены ручной настройкой полиномиальных функций данных из нескольких измерений.

На втором этапе выполняется предсказание движения (численное интегрирование, чтобы предсказать движение).

Для генератора траекторий зададим вектор, состоящий из начального и конечного состояния, начальной и конечной кривизны, а также скорости: $x_l = [x, y, \psi, k, \nu]^T$, $x_F = [x, y, \psi, k, \nu]^T$, что приводит к пятимерной таблице поиска векторов параметров. Разрешение и размерность сохраненной таблицы является функцией хранения и вычислительной мощности робота и, следовательно, зависит от платформы. Пример генерации соседних траекторий приведен на рис. 3.

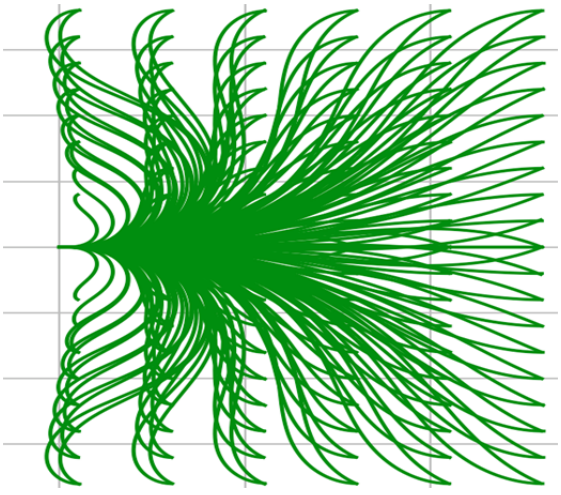

Рис. 3. Пример генерации соседних траекторий

Глобальный планировщик может создать предпочтительный путь для робота или поле затрат, или функция навигации $\varphi(x, y)$, кодирование оптимального расстояния до цели от каждого состояния.

Как правило, управление с обратной связью применяется к компенсированию расхождений робота с моделью движения, но в целом оно не может компенсировать неосуществимые траектории. Применяя метод, который интегрирует более сложные модели транспортного средства на стадии траекторного планирования, мы можем уменьшить количество ошибок, которые контроллер обратной связи должен компенсировать, имея более точный критерий стоимости кандидата для траектории на время его действия. Поэтому важно создать функцию выбора, которая правильно считает некоторую смесь риска столкновения, потребление энергии, задержку наклона (время, проведенное на склонах) глобального планировщика, гладкости траектории и другие «издержки», которые могут быть связаны с траекторией.

Синтез функции выбора

С помощью вычислительного метода находим функцию выбора, обеспечивающую минимум отклонения от глобальной траектории, по которой должен двигаться мобильный робот, избегая столкновения и другие ограничения вектора состояния и управления [11].

Поскольку функция навигации обычно используется в свертке с траекторией, то лучшая траектория вычисляется функцией выбора C_N для каждого действительного управления, который соединяет пару состояний, избегая препятствий:

$$C_N = \left[c_0, c_1, ..., c_n\right]^T.$$

Большинство современных методов предполагает ручную настройку операторов выбора, а перспективным подходом является применение методов машинного обучения, чтобы обучить правильному отображению действий и окружающей среды на функцию выбора.

Обучение нейронной сети

После того, как получим функцию выбора, мы должны обучать нейронную сеть следовать глобальной траектории, а также маневрам. После обучения нейронная сеть в точности должна воспроизводить функции исходного контроллера, а также характеристики дороги на основе технического зрения. Переход к другому поведению нейроконтроллера осуществляется с помощью функции выбора.

Последние достижения в области глубокого обучения позволяют конструировать самые современные модели нейронных сетей для решения сложных задач. Задача нейросетевого управления БТС требует мультимодальных входов. На входы могут поступать данные с разных источников, обрабатывая каждый вид данных с использованием разных типов нейронных слоев.

Контролируемое обучение может проводиться на основе данных, полученных с помощью обычного контроллера, отслеживающего движения примитивов (фиксированных маневров), называемых порождающей решеткой, которые приводят к регулярным графам. Такие примитивы соответствуют решению двуточечной краевой задачи. Функция выбора должна выбрать те маневры, которые соответствуют целевой глобальной оптимальной траектории по отношению к заданному функционалу качества.

Пример использования предложенного метода в реальном времени

Рассмотрим движение робота с шасси геометрии Акерманна [6]:

$$\begin{cases} \dot{x} = u_1 \cos\theta, \\ \dot{y} = u_1 \sin\theta, \\ \dot{\theta} = \dfrac{u_1}{L} \operatorname{tg} u_2, \end{cases}$$

где x, y – координаты центра задней оси мобильного робота; u_1 – его линейная скорость; θ – угол его поворота относительно оси x; u_2 – угол поворота (положительный против часовой стрелки); L – расстояние между передней и задней осями колес мобильного робота.

Заданы статические фазовые ограничения

$$\varphi(x,y) = r^2 - \left(x_i^* - x\right)^2 - \left(y_i^* - y\right)^2 \le 0,$$

где r – габариты препятствия; x_i^*, y_i^* – координаты центра препятствия.

Заданы динамические фазовые ограничения

$$\eta(x^p,x,y) = d^2 - \left(x_p^* - x\right)^2 - \left(y_p^* - y\right)^2 \le 0,$$

где d – габариты робота, а x_p^*, y_p^* – координаты другого робота.

Заданы терминальные условия x^f, y^f и θ^f, задано предельное время процесса управления t^+, задан функционал качества

$$J = t^f \to \min.$$

Для учета ограничений добавим к функционалу штраф за нарушение ограничений:

$$\tilde{J} = t^f + \sum_{i=1}^{r} \alpha_i \vartheta\left(\varphi(x,y)\right) + \sum_{i=1}^{K} \beta_i \vartheta\left(\eta(x^p,x,y)\right),$$

где ϑ – функция Хевисайда

$$\vartheta(a) = \begin{cases} 0, & a < 0, \\ 1, & a \ge 0. \end{cases}$$

В процессе отбора используются лазерные датчики расстояния и камеры с программой распознавания препятствий. На основании полученной с датчиков информации формируется вектор y, учитывающий реальные условия прохождения траектории.

При отборе траекторий используем следующие функции штрафа (γ – весовые коэффициенты, $i = 1,2,3$):

– $y_1 = \gamma_1 \operatorname{arctg}\left(\dfrac{z_{i+1} - z_i}{\Delta z}\right)$ – положение высоты робота над плоскостью, где Δ – шаг измерения;

– $y_2 = \gamma_2 \min\limits_{\tilde{x},\tilde{y},\tilde{\theta}} \left|\Omega(x,y,\theta) - \Omega(\tilde{x},\tilde{y},\tilde{\theta})\right|$ – отклонение от первого приближения оптимальной траектории;

– $y_3 = \gamma_3 \vartheta\left(|u_1| - \nu^+\right)$ – штраф за превышение допустимой скорости на участках, где скорость ограничена, где ν^+ – допустимая скорость на данном участке траектории.

Базовой функцией при синтезе функционала отбора использовалась сумма штрафов

$$F(Z_k) = y_1 + y_2 + y_3.$$

В качестве встраиваемой вычислительной платформы был использован NVIDIA Jetson TX2 с 256 ядрами CUDA и четырехядерным процессором ARM Cortex-A57, это позволило распараллелить компьютерную сеть [13, 14].

Функцию выбора находим методом сетевого оператора [4–6], который позволяет с помощью структуры данных находить структуру и параметры многомерной синтезирующей функции, описывающей зависимость функции выбора от координат пространства состояний и ограничений.

Метод использует кодирование математического выражения в виде ориентированного графа, который представляется в компьютере целочисленной матрицей сетевого оператора. Для поиска решения минимизируем функционал, описывающий точность движения по траектории, используем вариационный генетический алгоритм со следующими параметрами: размер начальной популяции –

50; число поколений – 10; число скрещиваемых пар в одном поколении – 30; число вариаций в одном решении – 10; число поколений между сменой базисного решения – 2; число элитарных решений – 16; вероятность мутации – 7.0 ; параметр для скрещивания – 4.0.

Для конструирования нейросети используется фреймворк глубокого обучения Keras, написанный на Python, и библиотека Tensor Flow в качестве внутреннего механизма.

В качестве модели была выбрана сверточная сеть с пятью слоями (рис. 4–8), за которыми следуют два плотных слоя перед выводом, за векторными данными следуют три плотных (полносвязанных) слоя, затем они объединяются с выходом *X* до двух плотных контрольных слоев. На рис. 9 приведены графики ошибки обучения при обучении с помощью контроллера, копирующего ПИД-регулятор.

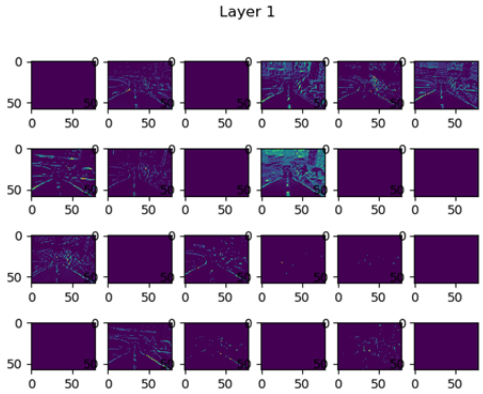

Рис. 4. Сверточная сеть. Первый слой

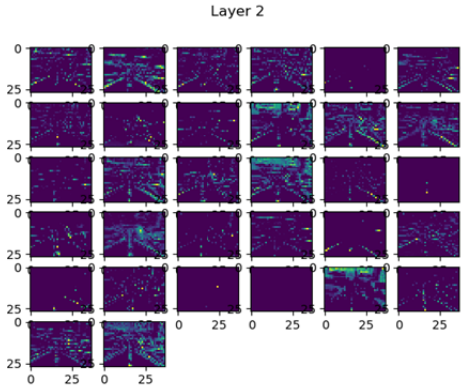

Рис. 5. Сверточная сеть. Второй слой

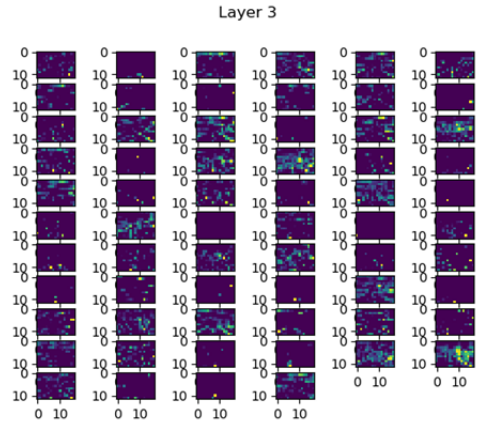

Рис. 6. Сверточная сеть. Третий слой

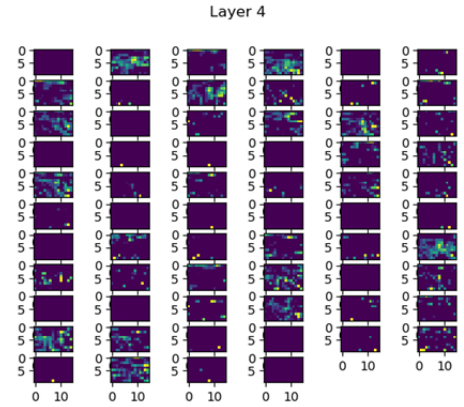

Рис. 7. Сверточная сеть. Четвертый слой

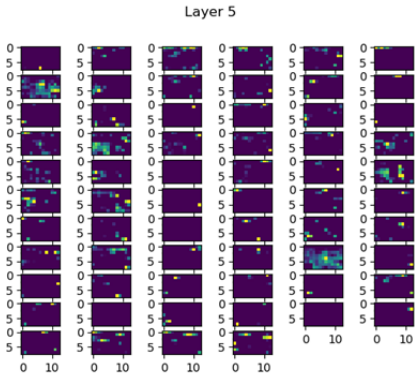

Рис. 8. Сверточная сеть. Пятый слой

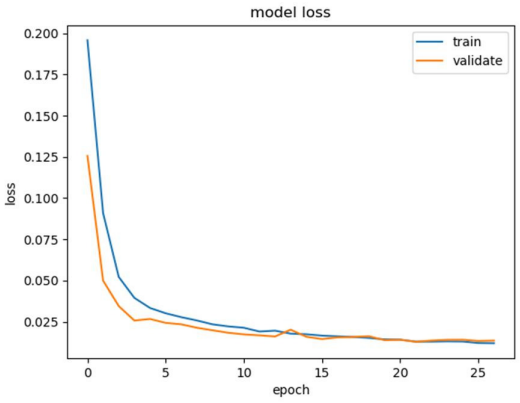

Рис. 9. Ошибка обучения при обучении с помощью контроллера, копирующего ПИД-регулятор

Блок-схема метода нейросетевого управления представлена на рис. 10. Контролируемое обучение по трассе в ручном режиме показано на рис. 11.

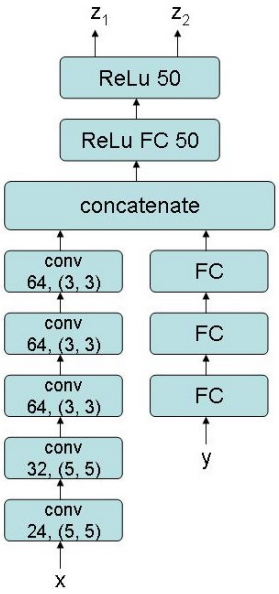

Рис. 10. Блок-схема метода нейросетевого управления

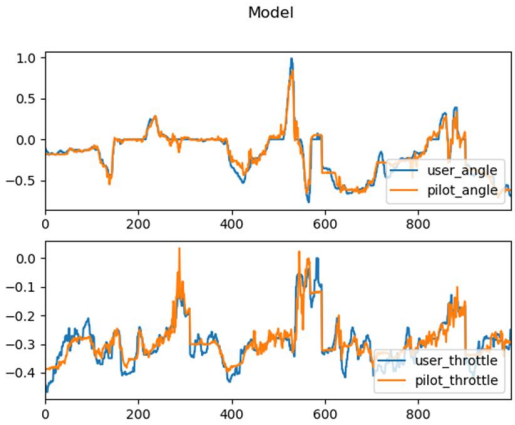

Рис. 11. Контролируемое обучение по трассе в ручном режиме

Вычислительные эксперименты проводились на симуляторе, написанном на языке Python [12]. Фото трассы с камеры беспилотного транспортного средства показано на рис. 12.

Рис. 12. Трасса. Вид с камеры робота

Библиографический список

1. *Howard, T. M.* State space sampling of feasible motions for high performance mobile robot navigation in complex environments / T. M. Howard, C. Green, D. Ferguson, A. Kelly // Journal of Field Robotics. – 2008. – Vol. 25, № 6-7. – P. 325–345.
2. *Howard, T. M.* Trajectory and spline generation for all-wheel steering mobile robots / T. M. Howard, A. Kelly // Proceedings of the IEEE/RSJ International Conference on Intelligent Robotics and Systems. – Beijing, China, 2006. – October.
3. *Howard, T. M.* Optimal rough terrain trajectory generation for wheeled mobile robots / T. M. Howard, A. Kelly // International Journal of Robotics Research. – 2007. – Vol. 26, № 2. – P. 141–166.

4. *Howard, T. M.* Receding horizon model-predictive control for mobile robot navigation of intricate paths / T. M. Howard, C. Green, A. Kelly // Proceedings of the 7th International Conference on Field and Service Robotics. – Chennai, India, 2009. – July.
5. *Kelly, A.* Reactive nonholonomic trajectory generation via parametric optimal control / A. Kelly, B. Nagy // International Journal of Robotics Research. – 2003. – Vol. 22 (7-8). – P. 583–601.
6. Motion planning in complex environments using closed-loop prediction / Y. Kuwata, J. Teo, S. Karaman, G. Fiore, E. Frazzoli, J. P. How // AIAA Guidance, Navigation and Control Conference and Exhibit. – Honolulu, HI, AIAA–2008–7166, 2008. – August.
7. *Diveev, A. I.* Application of network operator method for synthesis of optimal structure and parameters of automatic control system / A. I. Diveev, E. A. Sofronova // Proceedings of 17-th IFAC World Congress (05.07.2008 – 12.07.2008). – Seoul, 2008. – P. 6106–6113.
8. *Diveev, A. I.* Numerical method of network operator for multi-objective synthesis of optimal control system / A. I. Diveev, E. A. Sofronova // Proceedings of Seventh International Conference on Control and Automation (ICCA'09). – Christchurch, New Zealand, 2009. – P. 701–708.
9. *Diveev, A. I.* A Numerical Method for Network Operator for Synthesis of a Control System with Uncertain Initial Values / A. I. Diveev // Journal of Computer and Systems Sciences International. – 2012. – Vol. 51, № 2. – P. 228–243.
10. *Понтрягин, Л. С.* Принцип максимума в оптимальном управлении / Л. С. Понтрягин. – Москва : Наука, 1989.
11. *Beckov, A. V.* Problem of Cost Function Synthesis for Mobile Robot's Trajectory and the Network Operator Method for its Solution / A. V. Beckov, I. V. Prokopiev // 13th International Symposium Intelligent Systems. – St. Petersburg, Russia, 2018. – P. 695–701.
12. *Дивеев, А. И.* Библиотека Python для синтеза интеллектуальных систем управления / А. И. Дивеев, А. В. Доценко // Вестник РУДН. Сер.: Инженерные исследования. – 2018. – Vol. 19, № 2. – С. 177–189.

References

1. Howard T. M., Green C., Ferguson D., Kelly A. *Journal of Field Robotics*. 2008, vol. 25, no. 6-7, pp. 325–345.
2. Howard T. M., Kelly A. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robotics and Systems.* Beijing, China, 2006, October.
3. Howard T. M., Kelly A. *International Journal of Robotics Research*. 2007, vol. 26, no. 2, pp. 141–166.
4. Howard T. M., Green C., Kelly A. *Proceedings of the 7th International Conference on Field and Service Robotics*. Chennai, India, 2009, July.
5. Kelly A., Nagy B. *International Journal of Robotics Research*. 2003, vol. 22 (7-8), pp. 583–601.
6. Kuwata Y., Teo J., Karaman S., Fiore G., Frazzoli E., How J. P. *AIAA Guidance, Navigation and Control Conference and Exhibit*. Honolulu, HI, AIAA–2008–7166, 2008, August.
7. Diveev A. I., Sofronova E. A. *Proceedings of 17-th IFAC World Congress (05.07.2008 – 12.07.2008)*. Seoul, 2008, pp. 6106–6113.
8. Diveev A. I., Sofronova E. A. *Proceedings of Seventh International Conference on Control and Automation (ICCA'09)*. Christchurch, New Zealand, 2009, pp. 701–708.
9. Diveev A. I. A *Journal of Computer and Systems Sciences International*. 2012, vol. 51, no. 2, pp. 228–243.
10. Pontryagin L. S. *Printsip maksimuma v optimal'nom upravlenii* [Maximum principle in optimal control]. Moscow: Nauka, 1989. [In Russian]
11. Beckov A. V., Prokopiev I. V. *13th International Symposium Intelligent Systems*. St. Petersburg, Russia, 2018, pp. 695–701.
12. Diveev A. I., Dotsenko A. V. *Vestnik RUDN. Ser.: Inzhenernye issledovaniya* [Vestnik PFUR. Ser.: Engineering study]. 2018, vol. 19, no. 2, pp. 177–189. [In Russian]

Дарьина Анна Николаевна
кандидат физико-математических наук, доцент,
ведущий научный сотрудник,
Федеральный исследовательский центр
«Информатика и управление»
Российской академии наук
(Вычислительный центр
им. А. А. Дородницына РАН)
(119333, Россия, г. Москва, ул. Вавилова, 40)
E-mail: daryina@ccas.ru

Darina Anna Nikolaevna
candidate of physical and mathematical sciences,
associate professor, leading researcher,
Federal research center
«Computer science and control» of RAS
(Dorodnitsyn computer center
of the Russian Academy of Sciences)
(119333, 40 Vavilova street, Moscow, Russia)

Прокопьев Игорь Витальевич
доктор технических наук,
старший научный сотрудник,
отдел управления
робототехническими устройствами,
Федеральный исследовательский центр
«Информатика и управление»
Российской академии наук,
(Вычислительный центр
им. А. А. Дородницына РАН)
(119333, Россия, г. Москва, ул. Вавилова, 40)
E-mail: fvi-2003@mail.ru

Prokopyev Igor Vitalevich
doctor of technical sciences, senior stuff scientist,
department of control of robotic devices,
Federal research center
«Computer science and control» of RAS
(Dorodnitsyn computer center
of the Russian Academy of Sciences)
(119333, 40 Vavilova street, Moscow, Russia)

Образец цитирования:
Дарьина, А. Н. Метод нейросетевого управления в реальном времени на основе синтеза функции выбора / А. Н. Дарьина, И. В. Прокопьев // Надежность и качество сложных систем. – 2019. – № 4 (28). – С. 41–50. – DOI 10.21685/2307-4205-2019-4-4.