

## АВТОМАТИЧЕСКИЙ СИНТЕЗ НЕПРЕРЫВНОЙ ДИНАМИЧЕСКОЙ СИСТЕМЫ СТАБИЛИЗАЦИИ НА ОСНОВЕ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

А.В. Доценко

anton.dozenko@gmail.com

РУДН, Москва, Российская Федерация

---

### Аннотация

Рассмотрен автоматический процесс синтеза непрерывной динамической системы стабилизации на основе использования многослойных нейронных сетей. Предложено двухэтапное решение задачи синтеза. На первом этапе осуществлена генерация обучающей выборки. Для этого многократно решена задача оптимального управления рассматриваемой динамической системой при ее разных начальных условиях. Для каждого начального условия найдено оптимальное управление как функция состояний объекта. На втором этапе выполнена аппроксимация сгенерированной обучающей выборки с помощью нейронной сети типа многослойный персептрон. Полученный персептрон реализует блок управления, покрывающий всю область начальных условий объекта управления. Новизна работы состоит в том, что задача синтеза сводится к задаче аппроксимации: поиск замкнутой стабилизирующей системы осуществлен с помощью аппроксимации оптимальных управлений, найденных в ходе минимизации функционала интегрального типа. Представлен вычислительный эксперимент по синтезу системы стабилизации гусеничного робота. Для тестовой проверки синтезированной стабилизирующей системы рассмотрены начальные условия вне обучающей выборки, а также продемонстрирована работа системы в условиях внешних возмущений

### Ключевые слова

*Персептрон, синтез управления, генетический алгоритм, обучение с учителем, динамическая система*

Поступила 21.02.2020

Принята 09.05.2020

© Автор(ы), 2020

---

**Введение.** Задача синтеза управления является чрезвычайно важной, однако для ее решения существует не так много методов. Метод аналитического конструирования оптимальных регуляторов (АКОР) [1] применим только в случае линейных систем с квадратичным функционалом. Метод

аналитического конструирования агрегированных регуляторов (АКАР) [2] и метод бэкстеппинга [3] составляют важный раздел теории и практики синтеза систем оптимального управления. Однако перечисленные методы требуют «ручного» вмешательства. Возможность использования для решения задачи синтеза управления алгоритмов стабилизации, построенных на основе ПИД-регуляторов, исследована в [4, 5]. Однако и ПИД-регуляторы требуют квалифицированного вмешательства эксперта, так как перед их внедрением необходимо исследовать все каналы управления.

Приведенные выше аналитические подходы применимы в основном для линейных объектов управления с квадратичным функционалом. Численный критерий, наделяющий терминальное многообразие произвольной размерности притягивающими свойствами, сформулирован в [6], там же показано, как с помощью метода символьной регрессии [7] автоматизировать процесс синтеза синергетического управления [2].

Задача парковки в условиях фазовых ограничений рассмотрена в [8]. Здесь использован двухэтапный алгоритм, в котором сначала проводится поиск оптимальной структуры искусственной нейронной сети (ИНС), а затем — оптимизация весов найденной структуры. Для поиска оптимальной структуры применен метод грамматической эволюции. Последующая оптимизация весов осуществляется генетическим алгоритмом [9]. Такой подход, по сути, универсален, как и любой метод символьной регрессии, так как предусматривает наличие недифференцируемых функций активации в структуре ИНС. Однако в [8] осуществлен поиск управления лишь для нескольких начальных условий.

**Постановка задачи.** Задана математическая модель объекта управления

$$\dot{\mathbf{s}} = \mathbf{f}(\mathbf{s}, \mathbf{u}), \quad (1)$$

где  $\mathbf{s} = [s_1 \dots s_n]^T$  — вектор состояния объекта;  $\mathbf{s} \in \mathbb{R}^n$ ,  $\mathbf{u} = [u_1 \dots u_m]^T$  — вектор управления;  $\mathbf{u} \in U \subseteq \mathbb{R}^m$ ,  $U$  — ограниченное замкнутое множество.

Заданы множество начальных состояний

$$S_0 \subseteq \mathbb{R}^n, \quad (2)$$

терминальное состояние

$$\mathbf{s}(t_f) = \mathbf{s}^f \quad (3)$$

и функционал качества

$$J = \int_0^{t_f} f_0(\mathbf{s}(t), \mathbf{u}(t)) dt \rightarrow \min_{\mathbf{u} \in U} \quad (4)$$

Здесь  $t_f$  — ограниченное время процесса управления, которое может быть задано или определено по достижении терминального условия.

Требуется найти управление в форме функции компонентов вектора состояний объекта

$$\mathbf{u} = \mathbf{h}(\mathbf{s}), \quad (5)$$

где  $\mathbf{h}(\mathbf{s}): \mathbb{R}^n \rightarrow \mathbb{R}^m$ ,  $\mathbf{h}(\mathbf{s}) \subseteq U, \forall \mathbf{s} \in \mathbb{R}^n$ .

Подставив (5) в (1), получим модель объекта с функцией управления, зависящей от состояния:

$$\dot{\mathbf{s}} = \mathbf{f}(\mathbf{s}, \mathbf{h}(\mathbf{s})). \quad (6)$$

Требуется найти функцию управления (5), переводящую объект управления в терминальное состояние (3) и удовлетворяющую функционалу качества (4) для всех возможных начальных условий из множества (2).

*Этап 1. Генерация обучающей выборки.* Для генерации обучающей выборки используется многократное решение задачи оптимального управления объектом (1) при разных начальных условиях. Оптимальное управление находится в виде функции отклонения текущего состояния объекта от терминального.

Формируется конечное множество начальных условий

$$D = \{\mathbf{s}_1^0, \dots, \mathbf{s}_k^0\}, \mathbf{s}_i^0 \in S_0, i = \overline{1, k},$$

и для каждого из этих условий  $\mathbf{s}_i^0$  инициализируется нейросетевой аппроксиматор с начальным набором весовых коэффициентов  $\Psi^0 \in \mathbb{R}^{n \times m \times l}$ , где  $n$  — число слоев аппроксиматора;  $m, l$  — число входов и выходов слоя  $v = \overline{1, n}$ . Этот аппроксиматор для данного  $\mathbf{s}_i^0$  должен аппроксимировать функцию отклонения текущего состояния объекта от его терминального состояния

$$\Delta \mathbf{s}_i(t) = \mathbf{s}^f - \mathbf{s}_i(t), i \in \overline{1, k}, \quad (7)$$

где  $\mathbf{s}_i(t)$  — решение системы (6) для начальных условий  $\mathbf{s}_i^0$ . Для поиска оптимальных весовых коэффициентов  $\Psi^0$  аппроксиматора используется генетический алгоритм [9]. Генерируется популяция весовых коэффициентов

$$P = [\Psi_1, \dots, \Psi_p], \Psi_j = [\mathbf{W}_1 \dots \mathbf{W}_n]^T,$$

$$\mathbf{W}_v \in \mathbb{R}^{m \times l},$$

где  $m, l$  — число входов и выходов слоя  $v = \overline{1, n}$  нейронной сети;  $\Psi_j$ ,  $j = \overline{1, p}$  —  $j$ -й индивид популяции,  $p$  — размерность популяции. В ходе

итеративного процесса над индивидами  $\Psi_j, j = \overline{1, p}$ , выполняются операции генетического алгоритма — отбор, скрещивание, мутации с оптимизацией функционала (4) — и находится оптимальный для текущего начального условия  $\mathbf{s}_i^0$  вектор весов  $\Psi^o$ .

Для каждого начального условия  $\mathbf{s}_i^0, i = \overline{1, k}$ , для объекта (1) получают решения задачи оптимального управления  $\mathbf{u}_i = \mathbf{h}(\Delta \mathbf{s}_i | \Psi^o)$ . Далее выполняется моделирование процесса эволюции объекта (1) из начальных условий  $\mathbf{s}_i^0, i = \overline{1, k}$ , и запоминаются полученные отображения «отклонение–управление»:

$$\Delta \mathbf{s}_i(t) \rightarrow \mathbf{u}_i(t), i = \overline{1, k}, t \in [0, t_f].$$

При завершении этапа 1 для элементов множества начальных состояний  $D$  формируется обучающее множество пар «отклонение–управление»

$$T = \{(\Delta \mathbf{s}_j, \mathbf{u}_j)\}, j = \overline{1, q}, \quad (8)$$

где  $q$  — размер обучающей выборки.

*Этап 2. Аппроксимация функции управления (5).* Учитывая, что объект управления (1) может иметь сложную нелинейную динамику, необходимо выбрать аппроксиматор, который мог бы описать такое сложное поведение объекта. Согласно теореме Хорника [10], любую непрерывную функцию можно сколь угодно точно приблизить нейронной сетью с одним скрытым слоем. С использованием числа слоев и нейронов в слое можно управлять степенью нелинейности и обобщающей способностью указанной функции.

Нейронная сеть с весовыми коэффициентами  $\Theta$  инициализируется, и аппроксимируются функциональные связи  $\mathbf{u}_j(\Delta \mathbf{s}_j), j = \overline{1, q}$ , определяемые множеством (8) методом обратного распространения ошибки [11]. Минимизируется квадратичная функция ошибки

$$L(\Delta \mathbf{s}_j | \Theta) = \frac{1}{2} \|\mathbf{u}_j - \hat{\mathbf{u}}_j\|_2^2 \rightarrow \min_{\Theta}, j = \overline{1, q}, \quad (9)$$

где  $\hat{\mathbf{u}}_j$  — вектор выходов нейронной сети.

Получают нейронную сеть, реализующую замкнутую по состоянию функцию управления

$$\mathbf{u} = \mathbf{h}(\Delta \mathbf{s} | \Theta^o),$$

где  $\Theta^o$  — обученные весовые коэффициенты нейронной сети.

**Вычислительный эксперимент.** В качестве объекта управления рассматривается гусеничный мобильный робот [12], динамика которого описывается системой

$$\begin{aligned}\dot{x} &= \frac{(u_1 + u_2) \cos \theta}{2}; \\ \dot{y} &= \frac{(u_1 + u_2) \sin \theta}{2}; \\ \dot{\theta} &= \frac{u_1 - u_2}{2},\end{aligned}\quad (10)$$

где  $x, y$  — координаты центра масс робота;  $\theta$  — угол между осью симметрии робота и осью абсцисс инерциальной системы координат  $Oxy$ ;  $u_1, u_2 \in [-1, 1]$  — управления гусеницами робота.

Терминальное состояние системы фиксировано

$$\mathbf{s}^f = [0 \ 0 \ 0]^T. \quad (11)$$

Если выполняется условие

$$\|\mathbf{s}^f - \mathbf{s}(t)\|_2 < \varepsilon, \quad t \in [0, t^f],$$

где  $\varepsilon = 0,01$  — допустимая ошибка, то полагается, что объект (10) достиг терминальное состояние.

Подлежащий минимизации функционал имеет вид

$$J = \|\Delta \mathbf{s}(t^f)\|_2 + t^f = \|\mathbf{s}^f - \mathbf{s}(t^f)\|_2 + t^f, \quad (12)$$

$$t^f = \begin{cases} t, & \text{если } t < t^+; \\ t^+, & \text{иначе,} \end{cases}$$

где  $t \in [0, t^f]$ ,  $t^+ = 5$  с — максимальное допустимое время движения робота; вектор

$$\Delta \mathbf{s}(t) = \mathbf{s}^f - \mathbf{s}(t) = [x_f - x(t) \quad y_f - y(t) \quad \theta_f - \theta(t)]^T \quad (13)$$

определяет отклонение текущего состояния системы (10) от терминального состояния (11) в момент времени  $t \in [0, t^f]$ .

Дискретное множество начальных условий системы (10) полагается заданным:

$$D = \{\mathbf{s}_1^0, \dots, \mathbf{s}_k^0\}, \quad \mathbf{s}_i^0 = [x_i^0 \quad y_i^0 \quad \theta_i^0]^T, \quad k = 500, \quad i = \overline{1, k}, \quad (14)$$

где  $x_i^0, y_i^0 \in [-10, 10]$ ,  $\theta_i^0 \in [0, 2\pi]$  — величины, равномерно случайно распределенные в указанных интервалах.

*Этап 1.* Решается задача оптимального управления объектом (10) для каждого множества начальных условий (14). Для начального условия  $\mathbf{s}_i^0 \in D, i = \overline{1, k}$ , инициализируется персептронная нейронная сеть с двумя скрытыми слоями по 15 нейронов в каждом и выходным слоем — из двух нейронов, которая реализует систему управления объектом (10). Такая архитектура нейронной сети выбрана эвристическим путем. Ее главным достоинством является небольшое число свободных параметров и способность обеспечивать достаточно высокую скорость сходимости обучающего генетического алгоритма.

Алгоритм функционирования первого слоя нейронной сети можно записать в виде

$$\mathbf{y}_1 = \begin{bmatrix} y_1^1 \\ \vdots \\ y_1^{15} \end{bmatrix} = f_1(\mathbf{W}_1 \Delta \mathbf{s}) = \begin{bmatrix} f_1((\mathbf{w}_1^1)^T \Delta \mathbf{s}) \\ \vdots \\ f_1((\mathbf{w}_1^{15})^T \Delta \mathbf{s}) \end{bmatrix}, \quad (15)$$

$$\mathbf{W}_1 = \begin{bmatrix} \mathbf{w}_1^1 \\ \vdots \\ \mathbf{w}_1^{15} \end{bmatrix} = \begin{bmatrix} w_1^{1,1} & w_1^{1,2} & w_1^{1,3} \\ \vdots & & \vdots \\ w_1^{15,1} & w_1^{15,2} & w_1^{15,3} \end{bmatrix},$$

где  $\mathbf{y}_1$  — выходной вектор первого слоя;  $\mathbf{W}_1$  — матрица весовых коэффициентов первого слоя. Выходной вектор первого слоя является входным вектором для второго слоя, так что аналогично имеем

$$\mathbf{y}_2 = \begin{bmatrix} y_2^1 \\ \vdots \\ y_2^{15} \end{bmatrix} = f_2(\mathbf{W}_2 \mathbf{y}_1) = \begin{bmatrix} f_2((\mathbf{w}_2^1)^T \mathbf{y}_1) \\ \vdots \\ f_2((\mathbf{w}_2^{15})^T \mathbf{y}_1) \end{bmatrix}, \quad (16)$$

$$\mathbf{W}_2 = \begin{bmatrix} \mathbf{w}_2^1 \\ \vdots \\ \mathbf{w}_2^{15} \end{bmatrix} = \begin{bmatrix} w_2^{1,1} & \dots & w_2^{1,15} \\ \vdots & & \vdots \\ w_2^{15,1} & \dots & w_2^{15,15} \end{bmatrix}.$$

В качестве выходного сигнала сети получается вектор управления

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = f_3(\mathbf{W}_3 \mathbf{y}_2) = \begin{bmatrix} f_3((\mathbf{w}_3^1)^T \mathbf{y}_2) \\ f_3((\mathbf{w}_3^2)^T \mathbf{y}_2) \end{bmatrix}, \quad (17)$$

$$\mathbf{W}_3 = \begin{bmatrix} \mathbf{w}_3^1 \\ \mathbf{w}_3^2 \end{bmatrix} = \begin{bmatrix} w_3^{1,1} & \dots & w_3^{1,15} \\ w_3^{2,1} & \dots & w_3^{2,15} \end{bmatrix}.$$

В (15)–(17)  $f_i(\cdot)$  — функция активации  $i$ -го слоя,  $i = \overline{1, 3}$ . Каждая компонента функций  $f_1(\cdot)$ ,  $f_2(\cdot)$  имеет вид  $\phi(x) = \max(0, x)$ , а функции  $f_3(\cdot)$  —  $\phi(x) = \text{th } x$ .

Система (10) интегрируется на временном интервале  $[0, t^f]$  с постоянным шагом интегрирования 0,01. На каждом шаге интегрирования персептрон, реализующий систему управления, принимает в качестве входного вектора отклонение (13) и выдает управление (17).

Для обучения каждой  $k$  нейронной сети используется генетический алгоритм [9] с одноточечным скрещиванием и случайным отбором родителей. Параметры алгоритма:  $g = 100$ ,  $p = 500$  — число эпох обучения и особей в популяции соответственно;  $c = 100$  — число скрещиваний за одну эпоху по схеме «два родителя–два потомка»; вероятность мутации индивида 0,1. В качестве ошибки обучения используется функционал (12). Обучение останавливается, когда лучшее достигнутое популяцией значение этого функционала  $J_o$  становится меньше заданной ошибки обучения  $\rho = 0,1$  или когда достигнуто заданное число эпох обучения  $g$ , т. е. когда выполнено одно из условий

$$J_o < \rho, \quad e = g,$$

где  $e$  — текущая эпоха обучения.

Для рассматриваемого начального условия в результате обучения персептрона получается набор его оптимальных весовых коэффициентов вида

$$\Psi_i^o = [\mathbf{W}_1 \ \mathbf{W}_2 \ \mathbf{W}_3]^T.$$

Интегрируя систему (10) с использованием в качестве весовых коэффициентов персептрона коэффициентов  $\Psi_i^o$ , получаем для начального условия  $\mathbf{s}_i^0$  множество пар

$$T_i = \left\{ \left( \Delta \mathbf{s}(t^0), \mathbf{u}(t^0 | \Psi_i^o) \right), \dots, \left( \Delta \mathbf{s}(t_i^f), \mathbf{u}(t_i^f | \Psi_i^o) \right) \right\}, \quad i = \overline{1, k}. \quad (18)$$

Объединяя множества (18), получаем множество

$$T_D = T_1 \cup T_2 \cup \dots \cup T_k.$$

Формируем на основе множества  $T_D$  обучающую выборку в виде неупорядоченного набора пар вида  $(\Delta \mathbf{s}, \mathbf{u})$ :

$$T = \{(\Delta \mathbf{s}_j, \mathbf{u}_j)\}, \quad j = \overline{1, q},$$

где  $q = \sum_{i=1}^k |T_i|$ ,  $|\cdot|$  — мощность рассматриваемого множества. Из случайно выбранных пар  $(\Delta \mathbf{s}_j, \mathbf{u}_j)$  множества  $T$  формируется обучающая ( $T^{train}$ ) и тестовая ( $T^{test}$ ) выборки, мощности которых находятся в пропорции 4 : 1.

*Этап 2.* Суть этого этапа решения задачи состоит в параметрическом синтезе нейронной сети, которая аппроксимирует функциональные связи  $\mathbf{u}_j(\Delta \mathbf{s}_j)$ ,  $j \in \overline{1, q}$ , определяемые набором  $T$ . В результате указанная нейронная сеть оказывается способной отобразить состояние  $\Delta \mathbf{s} \in \mathbb{R}^3$  в оптимальное управление  $\mathbf{u} \in \mathbb{R}^2$ .

Используются методы обратного распространения ошибки и пакетного обучения [13]. Алгоритм обучения нейронной сети состоит из шести шагов.

1. Инициализация числа эпох обучения  $e^+ = 4000$  и начальных весов  $\Theta_0$  нейронной сети.
2. Равномерное случайное выделение из выборки  $T^{train}$  обучающей подвыборки  $T_b \subset T^{train}$  на каждой эпохе обучения  $e = 0, e^+ - 1$ .
3. Вычисление для каждой пары  $(\Delta \mathbf{s}_j, \mathbf{u}_j) \in T_b$  значения локальной ошибки обучения

$$L_j^{train}(\Delta \mathbf{s}_j | \Theta_e) = \frac{1}{2} \|\mathbf{u}_j - \hat{\mathbf{u}}_j\|_2^2 \rightarrow \min_{\Theta}, j = \overline{1, |T_b|},$$

и градиентов весовых коэффициентов  $\Delta \Theta_{e,j} = \frac{dL_j^{train}(\Delta \mathbf{s}_j | \Theta)}{d\Theta} |_{\Theta = \Theta_{e,j}}$ .

Здесь  $\hat{\mathbf{u}}_j$  — вектор выходов нейронной сети.

4. Обновление веса нейронной сети по правилу  $\Theta_e \leftarrow \Theta_{e-1} - \alpha \sum_{j=1}^{|T_b|} \Delta \Theta_{e,j}$ ,

где  $\alpha$  — коэффициент скорости обучения.

5. Вычисление значения ошибки обучения на тестовом множестве

$$L^{test}(\Delta \mathbf{s} | \Theta_e) = \frac{1}{|T^{test}|} \sum_{n=1}^{|T^{test}|} \|\mathbf{u}_n - \hat{\mathbf{u}}_n\|_2^2, \Delta \mathbf{s} \in T^{test}.$$

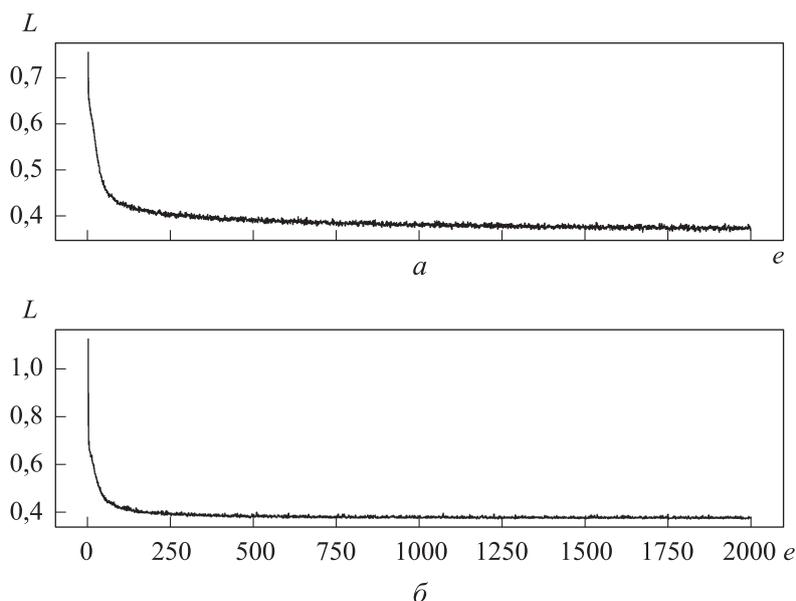
6. Остановка обучения, если  $e = e^+$  или если значение ошибки обучения  $L^{test}(\Theta_e)$  не снижается в течение 100 эпох. Иначе переход к шагу 2.

В результате получается замкнутая по состоянию функция управления

$$\mathbf{u} = \mathbf{h}(\Delta \mathbf{s} | \Theta^o),$$

где  $\Theta^o$  — обученные весовые коэффициенты нейронной сети.

Процесс стагнации ошибки для персептрона с двумя скрытыми слоями по 100 нейронов в каждом показан на рис. 1.



**Рис. 1.** Зависимость ошибки (9) от числа эпох обучения для обучающего  $T^{train}$  (a) и тестового  $T^{test}$  (б) множеств

Вычислительные эксперименты показали, что для повышения качества аппроксимации функциональной связи  $\mathbf{u}_j(\Delta \mathbf{s}_j)$ ,  $j \in \overline{1, q}$ , целесообразно использовать не одну нейронную сеть, как было рассмотрено выше, а комитет из трех персептронных нейронных сетей, имеющих по два скрытых слоя и состоящих из 80, 90 и 100 нейронов. Пусть  $\mathbf{u}^r = (u_1^r, u_2^r)$ ,  $r = \overline{1, 3}$  — выход  $r$ -й нейронной сети. Тогда управление, формируемое комитетом нейронных сетей, имеет вид

$$u_1 = \frac{1}{3}(u_1^1 + u_1^2 + u_1^3), \quad u_2 = \frac{1}{3}(u_2^1 + u_2^2 + u_2^3).$$

Траектории движения робота на плоскости в терминальное состояние  $\mathbf{s}^f = [0 \ 0 \ 0]^T$  из начальных условий

$$\begin{aligned}
 \mathbf{s}_1^0 &= \begin{bmatrix} -10 & 0 & \frac{\pi}{4} \end{bmatrix}^T, \quad \mathbf{s}_2^0 = \begin{bmatrix} -10 & 10 & \frac{\pi}{2} \end{bmatrix}^T, \\
 \mathbf{s}_3^0 &= \begin{bmatrix} 0 & 10 & \frac{3\pi}{4} \end{bmatrix}^T, \quad \mathbf{s}_4^0 = \begin{bmatrix} 10 & 10 & \pi \end{bmatrix}^T, \\
 \mathbf{s}_5^0 &= \begin{bmatrix} 10 & 0 & \frac{5\pi}{4} \end{bmatrix}^T, \quad \mathbf{s}_6^0 = \begin{bmatrix} 10 & -10 & \frac{3\pi}{2} \end{bmatrix}^T, \\
 \mathbf{s}_7^0 &= \begin{bmatrix} 0 & -10 & \frac{7\pi}{4} \end{bmatrix}^T, \quad \mathbf{s}_8^0 = \begin{bmatrix} -10 & -10 & 2\pi \end{bmatrix}^T,
 \end{aligned} \tag{19}$$

не принадлежащих обучающей ( $\Gamma^{train}$ ) и тестовой ( $\Gamma^{test}$ ) выборкам с использованием в качестве системы управления комитета нейронных сетей, приведены на рис. 2.

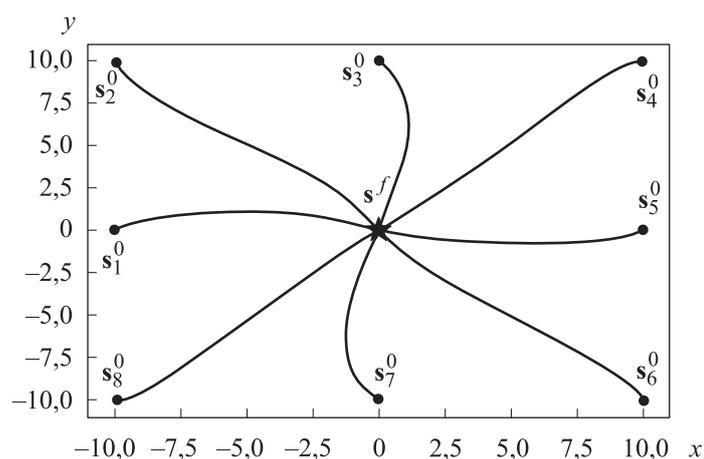


Рис. 2. Траектории движения робота (10) из начальных состояний (19) в терминальное  $s^f$

Зависимость оптимального управления от времени для четырех из восьми начальных состояний (19) приведена на рис. 3. Зависимость показывает, что найденные оптимальные управления не имеют разрывов и существенных скачков и удовлетворяют заданным ограничениям.

Для того чтобы понять, как ведет себя синтезированная система стабилизации в условиях внешних возмущений, к каждой переменной состояния системы (10) на каждом шаге интегрирования добавлялся нормальный шум  $N(0, 1)$  с нулевым математическим ожиданием и единичным среднеквадратичным отклонением. Результаты этого исследования представлены на рис. 4 и 5. Способность синтезированной системы стабили-

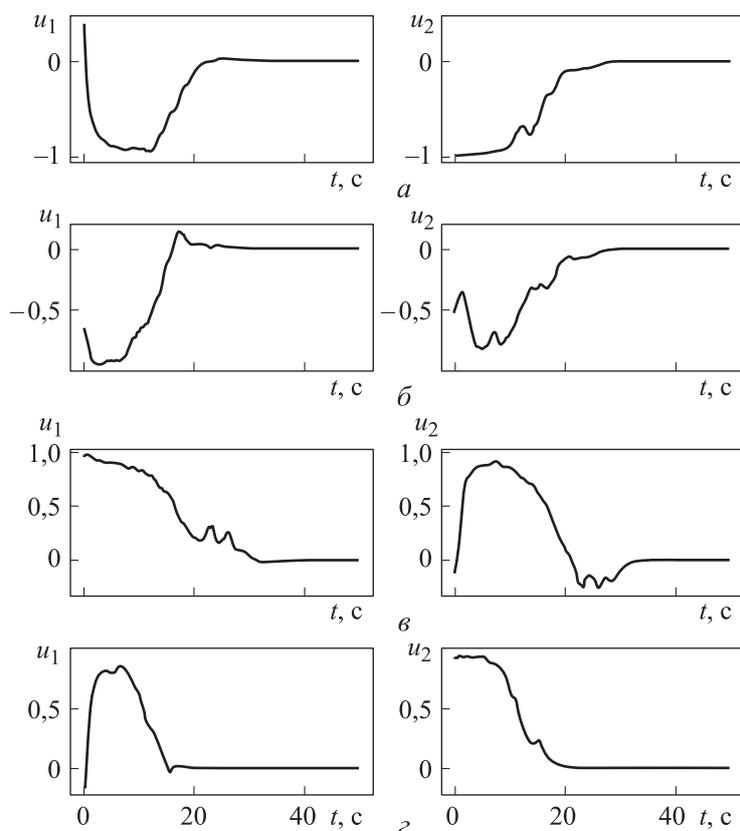


Рис. 3. Зависимость оптимального управления от времени для  $s_2^0$  (а),  $s_3^0$  (б),  $s_4^0$  (в) и  $s_5^0$  (г)

зации достигать терминального состояния в условиях внешних возмущений продемонстрирована на рис. 4. Зависимость оптимального управления от времени при воздействии возмущений приведена на рис. 5. Следует отметить, какое сильное изменение претерпели зависимости оптимального управления в результате воздействия на систему указанного возмущения.

Траектории, приведенные на рис. 6, демонстрируют способность системы (10) достигать терминального состояния, находясь за пределами области обучения вследствие наложения возмущения на начальные условия (19):

$$(\mathbf{s}_i^0)^* = \mathbf{s}_i^0 + [N(0,1) \ N(0,1) \ N(0,1)]^T; \quad i \in \overline{1,8}. \quad (20)$$

Траектории на рис. 6 свидетельствуют о высокой обобщающей способности рассматриваемого комитета нейронных сетей: несмотря на тот факт, что некоторые начальные состояния вышли за границы области обучения, комитет нейронных сетей перевел объект в терминальное состояние. Зависимость оптимального управления от времени приведена на рис. 7.

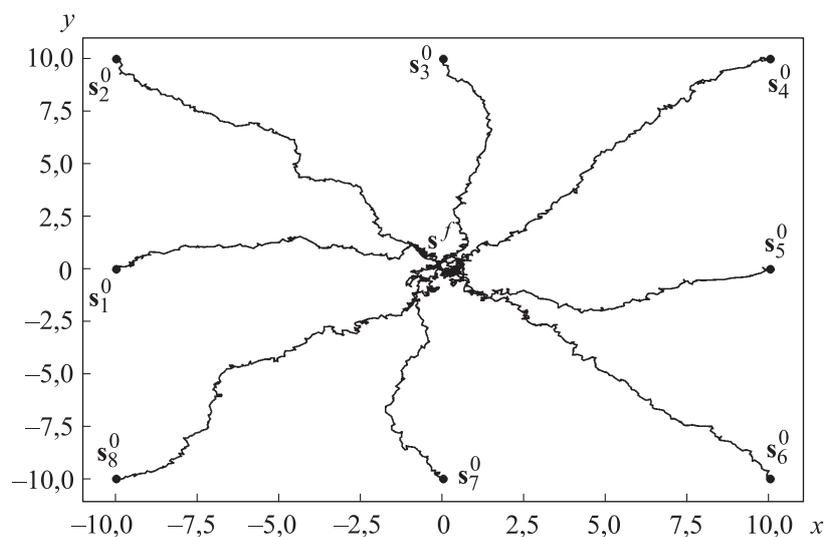


Рис. 4. Траектории движения робота (10) из начальных состояний (19) в терминальное  $s^f$  при наличии возмущений

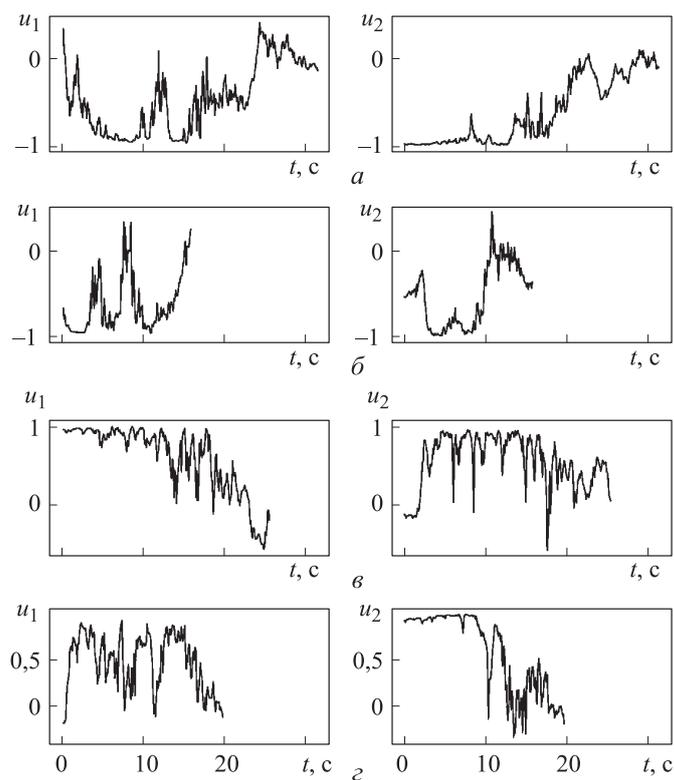


Рис. 5. Зависимость оптимального управления от времени при воздействии возмущений для  $s_2^0$  (а),  $s_3^0$  (б),  $s_4^0$  (в) и  $s_5^0$  (г)

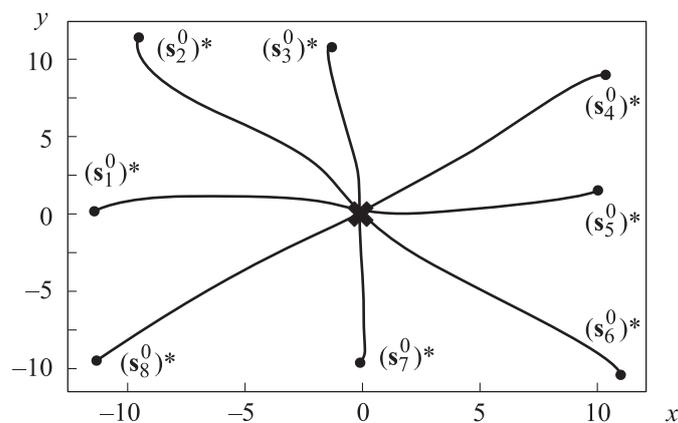


Рис. 6. Траектории движения робота (10) из возмущенных начальных состояний (19) в терминальное  $\mathbf{s}^f$

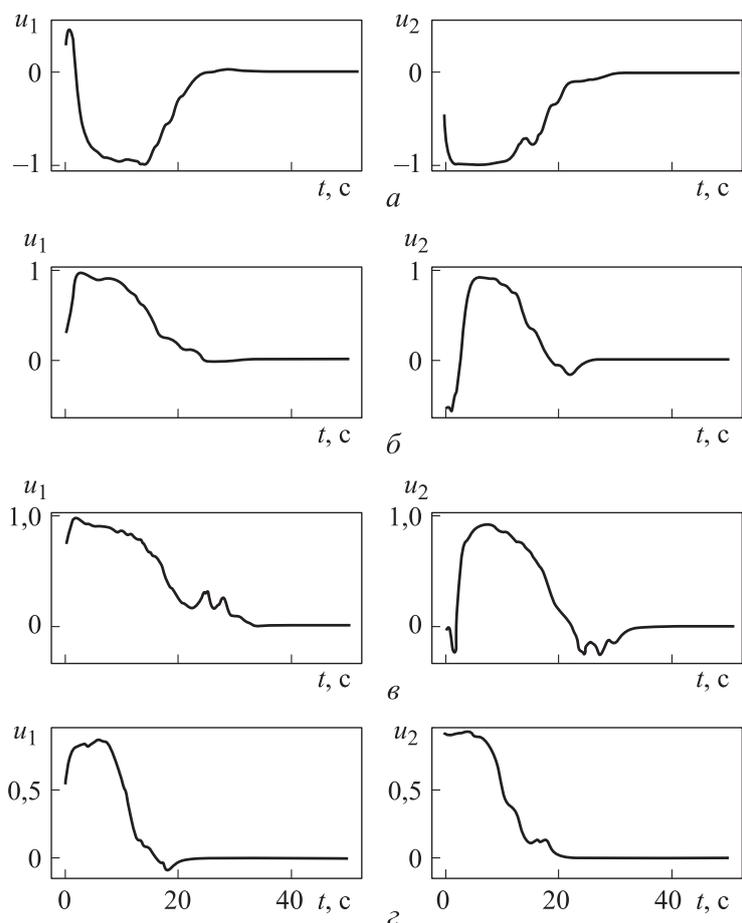


Рис. 7. Зависимость оптимального управления от времени при возмущенных начальных условиях (19), (20) для  $\mathbf{s}_2^0$  (а),  $\mathbf{s}_3^0$  (б),  $\mathbf{s}_4^0$  (в) и  $\mathbf{s}_5^0$  (г)

В качестве инструмента для синтеза рассмотренных нейронных сетей используется библиотека *Keras* [14], которая ориентирована на работу с нейронными сетями и спроектирована как компактная, модульная и расширяемая.

Обучение осуществлялось с использованием модификации алгоритма обратного распространения ошибки, называемой *Adam* [15]. Следует отметить, что при использовании в качестве функции активации функции *ReLU* [16, 17] в скрытых слоях и инициализации весовых коэффициентов по схеме Ксавье [18] многослойный персептрон обучается быстрее, чем при использовании сигмоидальных функций со случайной инициализацией. Существенно ускоряется процесс обучения нормализация входных сигналов слоев по мини-батчам [19].

**Заключение.** Представлена методика автоматического синтеза системы стабилизации непрерывной динамической системы путем аппроксимации оптимальных управлений, определенных в процессе минимизации функционала интегрального типа. Все, что необходимо выполнить оператору в рамках такой методики — это задать характеристики объекта управления, параметры генетического поиска и параметры нейронной сети, реализующей систему стабилизации. В качестве примера рассмотрена задача синтеза системы стабилизации гусеничного робота. Синтезированная в эксперименте система стабилизации оказалась способной доставлять объект управления в терминальное состояние из любого начального условия, лежащего внутри границ области обучения. Показано, что эта система является малочувствительной к внешним возмущениям.

Для продолжения указанных исследований планируется разработать систему управления группой роботов, осуществляющую перестроения в условиях фазовых и динамических ограничений.

## ЛИТЕРАТУРА

- [1] Колесников А.А., Колесников А.А., Кузьменко А.А. Методы АКАР и АКОР в задачах синтеза нелинейных систем управления. *Мехатроника, автоматизация, управление*, 2016, т. 17, № 10, с. 657–669.  
DOI: <https://doi.org/10.17587/mau.17.657-669>
- [2] Колесников А.А. Синергетическая теория управления: концепции, методы, тенденции развития. *Известия ТРТУ*, 2001, т. 23, № 5, с. 7–27.
- [3] Халил Х.К. Нелинейные системы. М., Ижевск, ИКИ РАН, 2009.
- [4] Гэн К., Чулин Н.А. Алгоритмы стабилизации и управления полетом квадрокоптера. *Молодежный научно-технический вестник*, 2014, № 11.  
URL: [http://masters.donntu.org/2017/etf/shichanin/library/pdf\\_5.pdf](http://masters.donntu.org/2017/etf/shichanin/library/pdf_5.pdf)

- [5] Гэн К., Чулин Н.А. Алгоритмы стабилизации для автоматического управления траекторным движением квадрокоптера. *Наука и образование: научное издание МГТУ им. Н.Э. Баумана*, 2015, № 5. DOI: <http://dx.doi.org/10.7463/0515.0771076>
- [6] Diveev A., Shmalko E., Sofronova E. Multipoint numerical criterion for manifolds to guarantee attractor properties in the problem of synergetic control design. *ITM Web Conf.*, 2018, vol. 18, art. 01001. DOI: <https://doi.org/10.1051/itmconf/20181801001>
- [7] Diveev A., Kazaryan D., Sofronova E. Symbolic regression methods for control system synthesis. *22nd Mediterranean Conf. Control and Automation*, 2014, pp. 587–592. DOI: <https://doi.org/10.1109/MED.2014.6961436>
- [8] Kazaryan D.E., Savinkov A.V. Grammatical evolution for neural network optimization in the control system synthesis problem. *Procedia Comput. Sci.*, 2017, vol. 103, no. C, pp. 14–19. DOI: <https://doi.org/10.1016/j.procs.2017.01.002>
- [9] Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы. М., ФИЗМАТЛИТ, 2010.
- [10] Hornik K., Stinchcombe M., White H. Multilayer feedforward networks are universal approximators. *Neural Netw.*, 1989, vol. 2, iss. 5, pp. 359–366. DOI: [https://doi.org/10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8)
- [11] Goodfellow I., Bengio Y., Courville A. Deep learning. MIT press, 2016.
- [12] Dubins L.E. On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents. *Amer. J. Math.*, 1957, vol. 79, pp. 497–516. DOI: <https://doi.org/10.2307/2372560>
- [13] Li M., Zhang T., Chen Y., et al. Efficient mini-batch training for stochastic optimization. *Proc. 20th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2014, pp. 661–670. DOI: <https://doi.org/10.1145/2623330.2623612>
- [14] Keras: веб-сайт. URL: <https://keras.io> (дата обращения: 15.03.2020).
- [15] Kingma D.P., Ba J. Adam: a method for stochastic optimization. *arxiv.org: веб-сайт*. URL: <https://arxiv.org/abs/1412.6980> (дата обращения: 15.03.2020)
- [16] Glorot X., Bordes A., Bengio Y. Deep sparse rectifier neural networks. *Proc. 14th Int. Conf. Artificial Intelligence and Statistics*, 2011, pp. 315–323. URL: <http://proceedings.mlr.press/v15/glorot11a/glorot11a.pdf> (дата обращения: 15.03.2020).
- [17] Nair V., Hinton G.E. Rectified linear units improve restricted Boltzmann machines. *Proc. 27th ICML-10*, 2010, pp. 807–814. URL: <http://www.cs.toronto.edu/~hinton/absps/reluICML.pdf> (дата обращения: 15.03.2020).
- [18] Glorot X., Bengio Y. Understanding the difficulty of training deep feedforward neural networks. *Proc. 13th Int. Conf. Artificial Intelligence and Statistics*, 2010, pp. 249–256. URL: <http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf> (дата обращения: 15.03.2020).
- [19] Ioffe S., Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. *arxiv.org: веб-сайт*. URL: <https://arxiv.org/abs/1502.03167v3> (дата обращения: 15.03.2020).

**Доценко Антон Викторович** — аспирант департамента механики и мехатроники Инженерной академии РУДН (Российская Федерация, Москва, 117198, ул. Миклухо-Маклая, д. 6).

**Просьба ссылаться на эту статью следующим образом:**

Доценко А.В. Автоматический синтез непрерывной динамической системы стабилизации на основе искусственных нейронных сетей. *Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение*, 2020, № 3 (132), с. 66–83.

DOI: <https://doi.org/10.18698/0236-3933-2020-3-66-83>

**AUTOMATIC SYNTHESIS OF A CONTINUOUS DYNAMIC STABILIZATION SYSTEM BASED ON ARTIFICIAL NEURAL NETWORKS**

**A.V. Dotsenko**

[anton.dozenko@gmail.com](mailto:anton.dozenko@gmail.com)

**RUDN University, Moscow, Russian Federation**

---

**Abstract**

The article considers an automated process of synthesising a continuous dynamic stabilization system based on multilayer neural networks. We propose a two-stage solution to the synthesis problem. The first stage involved generating the training dataset. It meant solving the optimum control problem for the dynamic system under consideration multiple times with various initial conditions. We derived optimum control for every initial condition as a function of object states. The second stage involved approximation of the training dataset we generated, which used a multilayer perceptron neural network. This perceptron represents a control unit that covers the entire initial condition space of the control object. The innovative aspect of our work is as follows: we reduced the synthesis problem to the approximation problem, that is, we sought a closed stabilising system by means of approximating the optimum controls obtained while minimising a functional of the integral type. The paper presents a numerical experiment in synthesising a stabilisation system for a track-laying robot. In order to test the stabilisation system synthesised, we consider initial conditions outside the training dataset and show how the system operates when externally perturbed

**Keywords**

*Perceptron, control synthesis, genetic algorithm, supervised learning, dynamic system*

Received 21.02.2020

Accepted 09.05.2020

© Author(s), 2020

## REFERENCES

- [1] Kolesnikov A.A., Kolesnikov A.A., Kuz'menko A. The ADAR method and theory of optimal control in the problems of synthesis of nonlinear control systems. *Mekhatronika, avtomatizatsiya, upravlenie*, 2016, vol. 17, no. 10, pp. 657–669 (in Russ.). DOI: <https://doi.org/10.17587/mau.17.657-669>
- [2] Kolesnikov A.A. Synergetic control theory: conceptions, methods, development tendencies. *Izvestiya TRTU*, 2001, vol. 23, no. 5, pp. 7–27 (in Russ.).
- [3] Khalil H.K. Nonlinear systems. Prentice Hall, 2002.
- [4] Gen K., Chulin N.A. Stabilization and control algorithms for quadcopter flight. *Molodezhnyy nauchno-tekhnicheskiy vestnik*, 2014, no. 11 (in Russ.). Available at: [http://masters.donntu.org/2017/etf/shichanin/library/pdf\\_5.pdf](http://masters.donntu.org/2017/etf/shichanin/library/pdf_5.pdf)
- [5] Gen K., Chulin N.A. Stabilization algorithms for automatic control of the trajectory movement of quadcopter. *Nauka i obrazovanie: nauchnoe izdanie MGTU im. N.E. Bauman* [Science and Education: Scientific Publication], 2015, no. 5 (in Russ.). DOI: <http://dx.doi.org/10.7463/0515.0771076>
- [6] Diveev A., Shmalko E., Sofronova E. Multipoint numerical criterion for manifolds to guarantee attractor properties in the problem of synergetic control design. *ITM Web Conf.*, 2018, vol. 18, art. 01001. DOI: <https://doi.org/10.1051/itmconf/20181801001>
- [7] Diveev A., Kazaryan D., Sofronova E. Symbolic regression methods for control system synthesis. *22nd Mediterranean Conf. Control and Automation*, 2014, pp. 587–592. DOI: <https://doi.org/10.1109/MED.2014.6961436>
- [8] Kazaryan D.E., Savinkov A.V. Grammatical evolution for neural network optimization in the control system synthesis problem. *Procedia Comput. Sci.*, 2017, vol. 103, no. C, pp. 14–19. DOI: <https://doi.org/10.1016/j.procs.2017.01.002>
- [9] Gladkov L.A., Kureychik V.V., Kureychik V.M. Geneticheskie algoritmy [Genetic algorithms]. Moscow, FIZMATLIT Publ., 2010.
- [10] Hornik K., Stinchcombe M., White H. Multilayer feedforward networks are universal approximators. *Neural Netw.*, 1989, vol. 2, iss. 5, pp. 359–366. DOI: [https://doi.org/10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8)
- [11] Goodfellow I., Bengio Y., Courville A. Deep learning. MIT press, 2016.
- [12] Dubins L.E. On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents. *Amer. J. Math.*, 1957, vol. 79, pp. 497–516. DOI: <https://doi.org/10.2307/2372560>
- [13] Li M., Zhang T., Chen Y., et al. Efficient mini-batch training for stochastic optimization. *Proc. 20th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2014, pp. 661–670. DOI: <https://doi.org/10.1145/2623330.2623612>
- [14] Keras: *website*. Available at: <https://keras.io> (accessed: 15.03.2020).
- [15] Kingma D.P., Ba J. Adam: a method for stochastic optimization. *arxiv.org: website*. Available at: <https://arxiv.org/abs/1412.6980> (accessed: 15.03.2020).

[16] Glorot X., Bordes A., Bengio Y. Deep sparse rectifier neural networks. *Proc. 14th Int. Conf. Artificial Intelligence and Statistics*, 2011, pp. 315–323.

Available at: <http://proceedings.mlr.press/v15/glorot11a/glorot11a.pdf>  
(accessed: 15.03.2020).

[17] Nair V., Hinton G.E. Rectified linear units improve restricted Boltzmann machines. *Proc. 27th ICML-10*, 2010, pp. 807–814.

Available at: <http://www.cs.toronto.edu/~hinton/absps/reluICML.pdf>  
(accessed: 15.03.2020).

[18] Glorot X., Bengio Y. Understanding the difficulty of training deep feedforward neural networks. *Proc. 13th Int. Conf. Artificial Intelligence and Statistics*, 2010, pp. 249–256.

Available at: <http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf>  
(accessed: 15.03.2020).

[19] Ioffe S., Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. *arxiv.org: website*.

Available at: <https://arxiv.org/abs/1502.03167v3> (accessed: 15.03.2020).

**Dotsenko A.V.** — Post-Graduate Student, Department of Mechanics and Mechatronics, Academy of Engineering, RUDN University (Miklukho-Maklaya ul. 6, Moscow, 117198 Russian Federation).

**Please cite this article in English as:**

Dotsenko A.V. Automatic synthesis of a continuous dynamic stabilization system based on artificial neural networks. *Herald of the Bauman Moscow State Technical University, Series Instrument Engineering*, 2020, no. 3 (132), pp. 66–83 (in Russ.). DOI: <https://doi.org/10.18698/0236-3933-2020-3-66-83>