УДК 681.513.7

А.М. ЧЕРНОДУБ

ВИБІР НЕЙРОЕМУЛЯТОРА НА ОСНОВІ МЕТОДУ КЕРУЮЧИХ ЛОКАЛЬНИХ ГРАДІЄНТІВ У МЕТОДІ НЕЙРОУПРАВЛІННЯ З ЕТАЛОННОЮ МОДЕЛЛЮ

Анотація. *Розглядаються задачі нейроідентифікації і нейроуправління з еталонною моделлю для нелінійного динамічного об'єкта. Аналізується проблема вибору нейроемулятора для навчання нейроконтролера, пропонується новий критерій на основі аналізу керуючих локальних градієнтів для вхідних нейронів нейроемулятора. Наводяться результати чисельних експериментів щодо навчання нейроконтролерів методами градієнтного спуска і розширеного фільтра Калмана.*
Ключові слова: *нейроуправління, нейроемулятор, рекурентна нейронна мережа, розширений фільтр Калмана.*

Аннотация. *Рассматриваются задачи нейроидентификации и нейроуправления с эталонной моделью для нелинейного динамического объекта. Анализируется проблема выбора нейроэмулятора для обучения нейроконтроллера, предлагается новый критерий на основе анализа управляющих локальных градиентов для входных нейронов нейроэмулятора. Приводятся результаты множественных численных экспериментов по обучению нейроконтроллеров методами градиентного спуска и расширенного фильтра Калмана.*
Ключевые слова: *нейроуправление, нейроэмулятор, рекуррентная нейронная сеть, расширенный фильтр Калмана.*

Abstract. *Neuroidentification and neurocontrol with the reference model problems for nonlinear dynamic object are considered. A problem of proper neuroemulator choosing for neurocontroller training is analyzed. A new criterion on the basis of local control gradients analysis for input neuroemulator's neurons is proposed. Results of numerical simulations of neurocontroller training by a gradient descent method and for an Extended Kalman Filter method are given.*
Keywords: *neurocontrol, neuroemulator, Recurrent Neural Network, Extended Kalman Filter.*

1. Вступ

Нейроуправління є різновидом адаптивного керування, коли як будівельні блоки систем керування використовуються штучні нейронні мережі. Нейронні мережі мають ряд унікальних властивостей, які роблять їх потужним інструментом для створення систем керування: здатність до навчання на прикладах і до узагальнення даних, здатність адаптуватися до зміни властивостей об'єкта керування і зовнішнього середовища, придатність для синтезу нелінійних регуляторів. За останні 20 років було розроблено велику кількість методів нейроуправління, найпопулярнішими на даний момент серед яких є методи нейроуправління з еталонною моделлю [1] (Model Reference Adaptive Neurocontrol) і системи адаптивної критики [2] (Adaptive Critics).

Метод нейроуправління з еталонною моделлю, також відомий як «схема з нейроемулятором і нейроконтролером» або «зворотне поширення в часі», був запропонований на початку 1990-х [1], [3, С. 168], [4–6], [7, С. 978], [8, С. 861]. Цей метод не вимагає знання математичної моделі об'єкта керування. Замість цього прямій динаміці об'єкта керування навчається окрема нейронна мережа, нейроемулятор, далі вона використовується для обчислення похідних при навчанні нейроконтролера.

При цьому з множини навчених нейроемуляторів зазвичай обирають той, що забезпечив найменшу середньоквадратичну помилку моделювання об'єкта керування. Однак, чи є такий критерій кращим, якщо нейромережа використовується з метою подальшого навчання іншої нейромережі, послідовно підключеної до першої, а не власне для моделювання об'єкта керування?

ISSN 1028-9763. Математичні машини і системи, 2012, № 3

У статті пропонується новий критерій відбору нейроемуляторів, який дозволяє в середньому більш ефективно навчати нейроконтролери в методі нейроуправління з еталонною моделлю та приводяться результати чисельних експериментів на типовій завдачі керування динамічним об'єктом.

2. Метод нейроуправління з еталонною моделлю

Нехай задано динамічний об'єкт керування, поведінка якого визначається дискретними формулами (1–2):

$$S(k+1) = \Phi(S(k), u(k)), \tag{1}$$

$$y(k+1) = \Psi(S(k)), \tag{2}$$

де $u(k)$ – вхідний сигнал, $y(k+1)$ – вихідний сигнал на такті k. Формули (1–2) вважаються апріорі невідомими. Незважаючи на те, що стан динамічного об'єкта $S(k)$ є недоступним для зовнішнього спостереження, його можливо оцінити різними способами. Для оцінки стану $S(k)$ ми використовуємо модель NARX:

$$S(k) = [u(k) \quad \ldots \quad u(k-L) \quad y(k) \quad \ldots \quad y(k-N)]^T. \tag{3}$$

Мета керування формулюється таким чином: нехай на такті k об'єкт керування перебуває в положенні $y(k)$ і задана уставка $r(k+1)$, яка є бажаним положенням об'єкта керування на наступному такті. Контролеру необхідно згенерувати такий сигнал керування $u(k)$, щоб зробити відмінність між уставкою $r(k+1)$ та положенням $y(k+1)$ мінімальним. У методі нейроуправління з еталонною моделлю уставка $r(k+1)$ додатково згладжується еталонною моделлю, в ролі якої використовується, як правило, стабільна лінійна динамічна система невеликого порядку. Таким чином, замкнений нейроконтролером об'єкт керування мусить мати таку ж поведінку, як і еталонна модель.

При стандартному використанні нейронних мереж для вирішення задач розпізнавання зазвичай існують 2 етапи функціонування:

1) етап навчання нейронної мережі на прикладах даних;

2) етап використання навченої мережі для розпізнавання.

Метод нейроуправління з еталонною моделлю (рис. 1) передбачає 3 етапи:

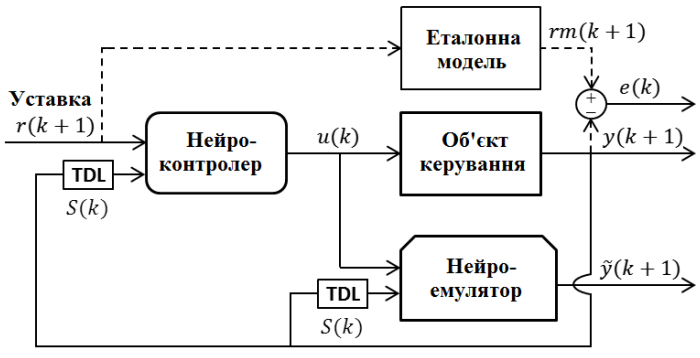

Рис. 1. Структурна схема методу нейроуправління з еталонною моделлю

1) етап навчання в режимі офф-лайн першої нейронної мережі, нейроемулятора, моделюванню об'єкта керування;

2) етап навчання в режимі он-лайн другої нейронної мережі, нейроконтролера, керуванню об'єктом керування;

3) етап використання навченого нейроконтролера для керування об'єктом (нейроемулятор на цьому фінальному етапі не використовується).

3. Навчання нейроемуляторів

Нейроемулятор являє собою нейронну мережу, навчену прямій динаміці об'єкта керування. У статті використано багатошаровий персептрон як нейронну мережу. Навчання ней-

роемулятора виконується згідно зі схемою «навчання з учителем». Нейромережа отримує на вхід $x(k)$ стан $S(k)$ об'єкта керування. Таким чином, нейроконтролер являє собою нерекурентну мережу, його «динамізація» відбувається за рахунок використання динамічних входів (3).

Розрахунок вихідного значення мережі $\tilde{y}(k+1)$ виконується за формулою

$$\tilde{y}(k+1) = g(\sum_j w_j^{(2)} f(\sum_i w_{ji}^{(1)} x_i)), \tag{4}$$

де $w^{(1)}$ – вагові коефіцієнти нейронів прихованого шару, $f(\cdot)$ – активаційні функції нейронів прихованого шару, $w^{(2)}$ – вагові коефіцієнти нейронів вихідного шару, $g(\cdot)$ – активаційні функції нейронів вихідного шару. У наших експериментах були використані персептрони з одним прихованим шаром і тангенціальними активаційними функціями у прихованому і вихідному шарах. На рис. 2 показано нейроемулятор з чотирма нейронами у прихованому шарі, який отримує на вхід стан з параметрами $L=0$ і $N=1$.

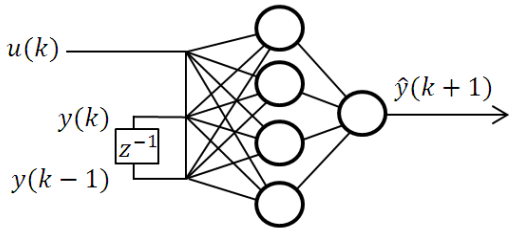

Рис. 2. Багатошаровий персептрон для моделювання динамічного процесу

Подібним чином сконструйовані нейроемулятори можуть навчатися різними градієнтними оптимізаційними методами з обчисленням похідних за методом зворотного поширення помилки (Backpropagation, BP). При цьому розмір обраного часового вікна, заданого параметрами L і N, має бути встановленим приблизно рівним порядку модельованого динамічного процесу, інакше алгоритму навчання буде неможливо знайти кореляцію між входом нейромережі і цільовим виходом.

4. Навчання нейроконтролерів

У нашій роботі нейроконтролери являють собою багатошарові персептрони з одним прихованим шаром, однак для них неможливе навчання з учителем, оскільки цільові значення керуючих сигналів $u(k)$ невідомі. Для корекції ваг нейроконтролера використовується метод зворотного поширення помилки через попередньо навчений нейроемулятор.

Навчання нейроконтролера виконується в режимі он-лайн, паралельно з керуванням об'єктом. На такті k нейроконтролер отримує на вхід значення уставки $r(k+1)$ та вектор стану $S(k)$:

$$x^{CO}(k) = [r(k+1) \quad S(k)]^T \tag{5}$$

і генерує керуючий сигнал $u(k)$ (рис. 1). Сигнал $u(k)$ надходить до об'єкта керування й переводить його в положення $y(k+1)$ і одночасно на нейроемулятор, який генерує реакцію $\tilde{y}(k+1)$. Якщо нейроемулятор є добре навченим, то $\|y(k+1) - \tilde{y}(k+1)\| < \varepsilon$, у наших експериментах $\varepsilon \approx 10^{-3}$. Також значення уставки $r(k+1)$ надходить на еталонну модель, яка генерує цільове значення для навчання нейроконтролера $rm(k+1)$. У ролі еталонної моделі використовується стабільна лінійна динамічна система першого порядку з передатною функцією $T(s) = \dfrac{1}{as+1}$, значення параметра $a > 0$ змінюються, а також просто дублюється значення уставки $r(k+1)$ (цей випадок в описі експериментів зазначено як $a = 0$). На

підставі вихідного значення еталонної моделі $rm(k+1)$ і нового положення об'єкта керування $y(k+1)$ формується поточна помилка керування $e(k)$:

$$e(k) = rm(k+1) - y(k+1). \tag{6}$$

Далі відбувається етап корекції ваг нейроконтролера залежно від обраного алгоритму оптимізації. Нами використано дві версії методу нейроуправління з еталонною моделлю: з методом градієнтного спуска [1, 3] і з методом розширеного фільтра Калмана [4–5].

Метод градієнтного спуска. При використанні цього алгоритму помилка керування $e(k)$ (6) пропускається через нейроемулятор у зворотному напрямку за методом зворотного поширення помилки, при цьому корекція ваг нейроемулятора не виконується. Для вхідних нейронів нейроемулятора розраховується вектор локальних градієнтів δ^{IN}:

$$\delta^{IN} = [\delta_1 \quad ... \quad \delta_{N+L+2}], \tag{7}$$

з компонентів якого вибирається локальний градієнт δ_u, відповідний до нейрона нейроемулятора, на який надходить керування $u(k)$. Назвемо цей локальний градієнт δ_u керуючим локальним градієнтом. При використанні вхідного вектора (5) $\delta_u = \delta_2$. Керуючий локальний градієнт δ_u пропускається далі через нейроконтролер за методом зворотного поширення помилки. Він використовується при розрахуванні локальних градієнтів прихованого і вхідного шарів нейроконтролера, а також похідної помилки по вагових коефіцієнтах прихованого й вихідного шарів нейроконтролера: $\frac{\partial E}{\partial w^{(1)}}$ і $\frac{\partial E}{\partial w^{(2)}}$. Далі проводиться корекція ваг за методом градієнтного спуска: $w(k+1) = w(k) + \alpha\, \Delta w(k)$, де α – швидкість навчання, $\Delta w(k) = \frac{\partial E(k)}{\partial w(k)}$.

Метод розширеного фільтра Калмана. Для корекції ваг нейроконтролера за методом розширеного фільтра Калмана на кожному такті k також використовується механізм зворотного поширення помилки через нейроемулятор, але тепер пропускається не поточне значення помилки $e(k)$ (6), а значення 1, що при тих же обчисленнях забезпечує отримання векторів якобіанів $\frac{\partial \tilde{y}}{\partial w^{(1)}}$ і $\frac{\partial \tilde{y}}{\partial w^{(2)}}$ замість векторів градієнтів $\frac{\partial E}{\partial w^{(1)}}$ і $\frac{\partial E}{\partial w^{(2)}}$, оскільки $\frac{\partial E}{\partial w} = e(k)\frac{\partial \tilde{y}}{\partial w}$. Ці якобіани на кожному такті формують матрицю спостережень фільтра Калмана $H(k)$ розміром $1 \times K$, K – сумарна кількість елементів матриць $w^{(1)}$ і $w^{(2)}$.

На початку роботи алгоритму навчання задається кореляційна матриця P розміром $K \times K$. На першому такті вона встановлюється рівній одиничній матриці: $P(1) = I$. Ми не розбиваємо матрицю P на кілька окремих матриць, як це робиться в методі незв'язного розширеного фільтра Калмана (DEKF) [4], [7, С. 960], [8, С. 855], оскільки подібне розбиття виконується винятково з метою економії обчислювальних ресурсів. Задається матриця швидкості навчання $R = \frac{1}{\eta}$, в нашому прикладі $\eta = 0{,}001$, і матриця шуму процесу Q, у нас $Q = 10^{-4} I$. На такті k, після знаходження помилки керування $e(k)$ і заповнення матриці $H(k)$ описаним вище способом, проводяться розрахунки нових значень вагових коефіцієнтів нейроконтролера $w(k+1)$ і матриці кореляції $P(k+1)$:

$$K(k) = P(k)H(k)^T [H(k)P(k)H(k)^T + R]^{-1}, \tag{8}$$

$$P(k+1) = P(k) - K(k)H(k)P(k) + Q, \tag{9}$$

$$w(k+1) = w(k) + K(k)e(k). \tag{10}$$

Зазначимо, що градієнти $\frac{\partial E}{\partial w}$ і якобіани $\frac{\partial \tilde{y}}{\partial w}$ для навчання нейроконтролера обчислюються звичайним методом зворотного поширення помилки (Backpropagation), а не методом усіченого зворотного поширення помилки в часі (BPTT(h)) [7, С. 942], [8, С. 836], [9] або рекурентного навчання в реальному часі (RTRL) [7, С. 949], [8, С. 840], [9-10]. Не дивлячись на те, що замкнена динамічна система має рекурентні зв'язки, використання статичних похідних є коректним, оскільки, по-перше, використані нами мережі не мають внутрішніх рекурентних зв'язків, а по-друге, тому, що при зворотному поширенні градієнта через послідовно з'єднаний нейроконтролер і нейроемулятор має місце ефект зникнення градієнта, що робить зазначені динамічні похідні чисельно дуже близькими до використовуваних нами статичних похідних вже при глибині усікання $h = 1$.

5. Параметр Θ для нейроемуляторів

У ході експериментів по навчанню нейроконтролерів виявлено, що деякі добре навчені нейроемулятори з помилкою порядка 10^{-6} не завжди забезпечують якісне навчання нейроконтролерів. При навчанні методом фільтра Калмана частина нейроконтролерів взагалі не навчилася.

При аналізі локальних градієнтів вхідних нейронів нейроемуляторів δ^{IN} (7), які розраховуються під час навчання нейроконтролерів, виявлено, що в таких неуспішних нейроемуляторах абсолютні значення керуючих локальних градієнтів δ_u відносно малі в порівнянні з модулями інших локальних градієнтів векторів δ^{IN}. При цьому точність нейроемуляторів у сенсі середньоквадратичної помилки приблизно рівна. Це можна пояснити в такий спосіб. Представимо, що послідовно з'єднані нейроемулятор і нейроконтролер є єдина нейронна мережа, в якій перші два шари ваг відповідають нейроконтролеру, а другі два шари – нейроемулятору. При навчанні нейроконтролера локальні градієнти пропускаються через ваги нейроемулятора і попадають на нейроконтролер через вхідний нейрон нейроемулятора, відповідний до вихідного нейрона нейроконтролера. Це означає, що при цьому використовується тільки керуючий локальний градієнт δ_u із усього вектора δ^{IN}, для інших локальних градієнтів зворотне поширення закінчується на даному етапі і вони не впливають на корекцію ваг нейроконтролера.

Для оцінки відносної величини модулів локальних керуючих градієнтів δ_u ми вводимо параметр θ:

$$\theta = \frac{1}{n}\sum_{k=1}^{n} \frac{\delta_1^2(k) + \delta_2^2(k) + \ldots + \delta_{N+L+2}^2(k)}{\delta_u^2(k)}, \tag{11}$$

де n – кількість елементів навчальної виборки. Локальні градієнти $\delta_1(k),\ldots,\delta_{N+L+2}(k), k = \{1,2,\ldots,n\}$ обчислюються методом зворотного поширення помилки, який застосовується на навчальній виборці після навчання нейроемулятора, для пропуску через нейроемулятор завжди використовується величина 1.

6. Результати навчання нейроемуляторів

У нашій роботі для чисельних експериментів по ідентифікації і керуванню використано нелінійний динамічний об'єкт другого порядку з [11], який, у свою чергу, є незначно модифікованою версією об'єкта керування з [1]. Об'єкт керування задано формулою

$$y(k+1)=\frac{y(k)y(k-1)y(k-2)u(k-1)[y(k-2)-1]+u(k)}{1+y(k-1)^2+y(k-2)^2}, \qquad (12)$$

де $u(k)$ – вхідний сигнал, $y(k+1)$ – вихідний сигнал на такті k.

Навчання нейроемуляторів проводилося в середовищі MATLAB без використання пакета Neural Networks Toolbox. На об'єкт подавався ідентифікаційний сигнал $u(k)=\sin(2\pi k/300)$ і проводилося протоколювання виходів $y(k)$. На основі записаних даних були сформовані навчальна і тестова вибірки прикладів динаміки в кількості 1000 і 200 прикладів відповідно. На них були навчені нейроемулятори методом навчання з учителем. Як алгоритм навчання був використаний метод глобального розширеного фільтра Калмана (Global Extended Kalman Filter, GEKF) [7, С. 955], [8, С. 854], [10]. Для обчислення якобіанів нейромереж $\frac{\partial\tilde{y}}{\partial w}$, які необхідно мати для роботи алгоритму GEKF, використовувався метод зворотного поширення помилки. Для всіх шарів нейромереж були використані тангенціальні активаційні функції, кількість нейронів у прихованому шарі варіювалася від 3 до 7.

Таблиця 1. Характеристики навчених нейроемуляторів

	MSE, мін.	MSE, середнє	θ, мін.	θ, середнє
Нейроемулятори А	$7{,}22\times10^{-6}$	$9{,}17\times10^{-6}$	2,32	18,73
Нейроемулятори Б	$8{,}48\times10^{-6}$	$9{,}32\times10^{-6}$	2,00	11,23

Усього було навчено 2 множини по 250 нейроемуляторів: нейроемулятори А, куди потрапили нейроемулятори, для яких значення MSE менше 1×10^{-5}, та нейроемулятори Б, які містять нейроемулятори, для яких MSE менше 1×10^{-5} і, додаткова умова, $\theta<20$ (табл. 1). Для оцінки якості нейроемуляторів використовувалася стандартна формула середньоквадратичної помилки: $MSE=<(\tilde{y}_k-t_k)^2>$, де $\tilde{y}_k$ – вихід нейромережі, t_k – цільове значення.

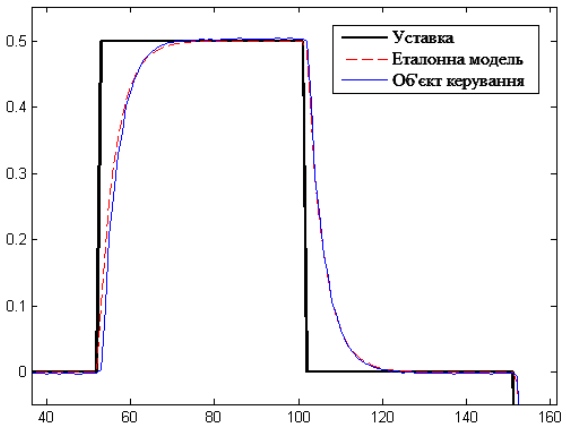

Рис. 3. Приклад траєкторії руху об'єкта керування з навченим нейроконтролером на тестовій ділянці, a=0,001, помилка керування MSE=0,0019

7. Результати навчання нейроконтролерів

Після навчання нейроемуляторів було проведено навчання нейроконтролерів, для чого нейромережі і алгоритми їх навчання були змодельовані в середовищі Simulink, без використання пакета Neural Network Blockset. Кожний експеримент по навчанню нейроконтролерів тривав 100 250 тактів, з яких упродовж 100 000 тактів на нейроконтролер подавався випадковий процес і виконувалося їх навчання методами градієнтного спуска або розширеного фільтра Калмана (навчальна ділянка); далі протягом 250 тактів, що залишилися, ваги нейроконтролерів не коректувалися, на них подавалася тестова уставка, як показано на рис. 3 (тестова ділянка), і фіксувалося середньоквадратичне відхилення MSE траєкторії проходження об'єкта керування від траєкторії еталонної моделі на цій ділянці.

Таблиця 2. Помилки керування навчених нейроконтролерів, навчання методом градієнтного спуска

	a = 0	a = 0,001	a = 0,002	a = 0,005
Нейроемулятори А, MSE, мін.	0,0072	0,0060	0,0066	0,0107
Нейроемулятори Б, $\theta < 20$, MSE, мін.	0,0074	0,0059	0,0064	0,0106
Нейроемулятори А, MSE, середнє	0,0117	0,0101	0,0102	0,0135
Нейроемулятори Б, $\theta < 20$, MSE, середнє	0,0088	0,0072	0,0075	0,0112

Таблиця 3. Помилки керування навчених нейроконтролерів, навчання методом розширеного фільтра Калмана

	a = 0	a = 0,001	a = 0,002	a = 0,005
Нейроемулятори А, MSE, мін.	0,0013	0,0019	0,0018	0,0017
Нейроемулятори Б, $\theta < 20$, MSE, мін.	0,0013	0,0019	0,0018	0,0017
Нейроемулятори А, MSE, середнє	0,0396	0,0420	0,0419	0,0539
Нейроемулятори Б, $\theta < 20$, MSE, середнє	0,0030	0,0045	0,0054	0,0351

Для кожного з нейроемуляторів з множин А і Б було навчено по 4 нейроконтролери з різними параметрами a еталонної моделі, результати наведено в табл. 2–3. При приблизно рівній якості навчання нейроемуляторів (табл. 1) використання запропонованого критерію відбору нейроемуляторів, які склали множину нейроемуляторів Б, забезпечило підвищення якості навчання нейроконтролерів у середньому на 25–30% для навчання методом градієнтного спуска і приблизно на порядок – для навчання методом розширеного фільтра Калмана.

8. Висновки

В роботі запропоновано новий критерій відбору нейроемуляторів для навчання нейроконтролерів у методі нейроуправління з еталонною моделлю. Було проведено експериментальне дослідження пропонованого критерію з навчанням 500 нейроемуляторів і 4000 нейроконтролерів, що показало його ефективність у порівнянні з традиційним способом відбору нейроемуляторів за методом найменшої середньоквадратичної помилки на тестовій вибірці даних.

У рамках подальших досліджень планується випробувати даний критерій разом з іншими методами нейроуправління, в яких передбачено етап попередньої нейроідентифікації об'єкта керування: прогнозуючого модельного нейроуправління [6], [12] і гібридного нейро-ПІД керування [3, С. 159], [6], а також з використанням кубатурного фільтра Калмана [8, С. 787], [13].

СПИСОК ЛІТЕРАТУРИ

1. Narendra K.S. Identification and control of dynamical systems using neural networks / K.S. Narendra, K.K. Parthasarathy // IEEE Transactions on Neural Networks. – 1990. – N 1. – P. 4 – 27.

2. Prokhorov D. Adaptive Critic Designs / D. Prokhorov, D. Wunsch // IEEE Transactions on Neural Networks. – 1997. – Vol. 8, N 5. – P. 997 – 1007.

3. Омату С. Нейроуправление и его приложения / Омату С., Халид М., Юсоф Р.; пер. с англ. – М.: ИПРЖР, 2000. – 272 с.

4. Feldkamp L.A. Training controllers for robustness: multi-stream DEKF / L.A. Feldkamp, G.V. Puskorius // Proc. of International Conference on Neural Networks, (Orlando, FL, USA, 27 Jun – 2 Jul 1994). – 1994. – Vol. 4. – P. 2377 – 2382.
5. Prokhorov D.V. Toyota Prius HEV Neurocontrol and Diagnostics / D.V. Prokhorov // Neural Networks. – 2008. – N 21. – P. 458 – 465.
6. Чернодуб А.Н. Обзор методов нейроуправления / А.Н. Чернодуб, Д.А. Дзюба // Проблемы программирования. – 2011. – № 2. – С. 79 – 94.
7. Хайкин С. Нейронные сети: полный курс / Хайкин С.; пер. с англ. – [2-е изд., испр.]. – М.: Вильямс, 2006. – 1104 с.
8. Haykin S. Neural Networks and Learning Machines, Third Edition / Haykin S. – New York: Prentice Hall, 2009. – 936 p.
9. De Jesus O. Backpropagation: Algorithms for a Broad Class of Dynamic Networks / O. De Jesus, M.T. Hagan // IEEE Transactions on Neural Networks. – 2007. – Vol. 18, N 1. – P. 14 – 27.
10. Cernansky M. Simple recurrent network trained by RTRL and extended Kalman filter algorithms / M. Cernansky, L. Benuskova // Neural Network World. – 2003. – Vol. 3, N 13. – P. 223 – 234.
11. Venelinov Topalov A. Online learning in adaptive neurocontrol schemes with a sliding mode algorithm / A. Venelinov Topalov, O. Kaynak // IEEE Transactions on Systems, Man and Cybernetics. Part B: Cybernetics. – 2001. – Vol. 31, Is. 3. – P. 445 – 450.
12. Hagan M.T. Neural networks for control / M.T. Hagan, H.B. Demuth // Proc. of the American Control Conference. – San Diego, USA, 1999. – Vol. 3. – P. 1642 – 1656.
13. Arasaratnam I. Cubature Kalman Filters / I. Arasaratnam, S. Haykin // IEEE Transactions on Automatic Control. – 2009. – Vol. 54, Is. 6. – P. 1254 – 1269.

Стаття надійшла до редакції 29.03.2012