

УДК 519.71

А.А. Бессонов

Харьковский национальный университет радиотехники, Харьков

РЕШЕНИЕ ЗАДАЧИ УПРАВЛЕНИЯ С ПРОГНОЗИРУЮЩЕЙ МОДЕЛЬЮ НА ОСНОВЕ ЭВОЛЮЦИОНИРУЮЩЕГО МНОГОСЛОЙНОГО ПЕРСЕПТРОНА

Статья посвящена разработке метода управления нелинейными объектами с помощью эволюционирующего многослойного персептрона. Эволюционирующий многослойный персептрон применяется для построения нелинейной модели объекта, которая впоследствии используется для рекурсивного предсказания поведения объекта в системе управления с прогнозирующей моделью. Для обучения нейронной сети предлагается применение генетического алгоритма, позволяющего значительно ускорить процесс обучения. Приводятся результаты имитационного моделирования, подтверждающие эффективность предложенного метода управления.

Ключевые слова: прогнозирующая модель, генетический алгоритм, многослойный персептрон.

Введение

Управление с прогнозирующей моделью использует последовательности управляющих переменных с целью оптимизации будущего поведения объекта управления. Выходной сигнал объекта при этом предсказывается с помощью заранее построенной модели данного процесса, которая является важнейшим элементом системы, т.к. характеристики управления непосредственно зависят от нее. Как правило, использование более точной модели повышает вероятность получения требуемого качества управления. Однако зачастую более точная модель вызывает повышенную чувствительность системы управления к погрешностям процесса и, тем самым, оказывает негативное влияние на качество управления. Кроме того, использование более сложных моделей приводит к необходимости выполнения большего числа итераций для решения задачи оптимизации, лежащей в основе алгоритма управления. В связи с этим целесообразно представляется разработка систем управления на основе теории искусственных нейронных сетей (ИНС), позволяющих построить довольно простые и в то же время эффективные нейросетевые модели.

Задача управления с прогнозирующей моделью

Задача прогнозирующего управления при использовании нейросетевых моделей заключается в определении значений управляющего сигнала $u(k)$, обеспечивающих минимум функционала

$$J(k) = \sum_{i=1}^{N_p} [\hat{y}(k+i) - r(k+i)]^2, \quad (1)$$

где $\hat{y}(k)$, $r(k)$ – предсказанный и эталонный выходной сигнал; N_p – горизонт предсказания.

Система с прогнозирующей моделью использует предсказание на несколько шагов вперед. Тради-

ционная схема управления с прогнозирующей моделью показана на рис. 1 [1]. В текущий момент времени прогнозирующая модель используется для предсказания влияния будущего управляющего сигнала u (или «манипулирующей переменной») на выходной сигнал объекта y . Предсказание производится на несколько шагов вперед, так, как это показано на рис. 2. Учитывая будущий эталонный сигнал r и будущий выходной сигнал \hat{y} , будущее изменение управляющего сигнала u определяется с помощью оптимизационного процесса с целью уменьшения предсказанной ошибки. Если не существует значительных расхождений между моделью и объектом, т.е. модель идентична объекту и отсутствуют помехи измерений, то выходной сигнал объекта будет точно повторять эталонную траекторию.

Управление с прогнозирующей моделью состоит из следующих шагов: 1) прямое использование модели для предсказания выходного сигнала объекта в пределах заданного горизонта предсказания; 2) вычисление последовательности управляющих сигналов в пределах заданного горизонта управления; 3) смещение горизонтов предсказания и управления в будущее таким образом, чтобы в момент времени k актуальным являлся первый сигнал вычисленной последовательности управляющих сигналов (рис. 3).

Модель нелинейного объекта

При управлении с прогнозирующей моделью непосредственно используется точная и отдельно идентифицированная модель объекта. Рассмотрим, например, многомерный нелинейный динамический объект, описываемый уравнением

$$y(k) = f[y(k-1), \dots, y(k-m), u(k-1), \dots, u(k-n)] + \xi(k), \quad (2)$$

где $y(i) = (y_1(i), \dots, y_{N_y}(i))^T$; $u(i) = (u_1(i), \dots, u_{N_u}(i))^T$ – векторы выходных и управляющих сигналов размерностей $N_y \times 1$ и $N_u \times 1$ соответственно; $f(\bullet) = (f_1(\bullet),$

$f_2(\bullet), \dots, f_{N_y}(\bullet)^T$ – неизвестная нелинейная векторная функция; $\xi(k)=[\xi_1(k), \dots, \xi_{N_y}(k)]^T$ – вектор помех.



Рис. 1. Общая схема прогнозирующего управления с моделью



Рис. 2. Прогнозирование значений выходного сигнала

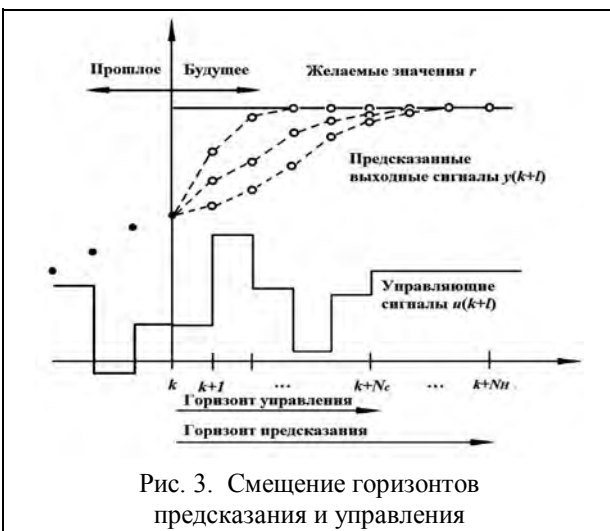


Рис. 3. Смещение горизонтов предсказания и управления

Большое количество задач управления достаточно сложно решить с помощью традиционных методов построения модели нелинейного объекта (2), в связи с чем актуальным является использование для этой цели ИНС. Наибольшее применение для решения задачи идентификации нелинейных динамических объектов (2) в настоящее время получили многослойный персептрон (МП) и радиально-базисные сети (РБС), использующие аппроксимацию нелинейного оператора $f(\bullet)$ нейронной сетью.

Уравнение МП имеет вид

$$\hat{y}(k) = f^q \left[\left(W^q \right)^T f^{q-1} \left[\left(W^{q-1} \right)^T f^{q-2} \left[\dots f^1 \left[\left(W^1 \right)^T \right] x(k) \right] \right] \right] \quad (3)$$

где q – количество слоев в сети; W^i – вектор весовых параметров нейронов i -го слоя сети; $f^i[\bullet]$ – функция активации i -го слоя. В качестве $f^i[\bullet]$ обычно используют сигмоидальные активационные функции, например, вида

$$f(x) = (1 + e^{-\alpha x})^{-1} \quad (4)$$

Задача идентификации состоит в обучении сети, т.е. в такой настройке ее параметров с помощью некоторого алгоритма обучения, которая бы обеспечила минимум функционалу

$$J = e^2(k) = M \left\{ \left[\tilde{y}(k) - \hat{y}(k) \right]^2 \right\} \quad (5)$$

Для нахождения минимума функционала (5) обычно используются градиентные методы минимизации, однако при решении задачи управления с прогнозирующей моделью важным требованием является скорость обучения, в связи с чем возникает необходимость применения неградиентных методов обучения, что позволяет значительно упростить производимые вычисления [2]. Одним из таких методов является обучение с помощью эволюционных алгоритмов.

Нейроэволюционный алгоритм обучения многослойного персептрона

Одним из наиболее распространенных видов эволюционных алгоритмов являются генетические, предложенные Дж. Холландом. Однако при использовании классических генетических алгоритмов для обучения МП удается найти лишь некоторое оптимальное решение, являющееся локальным минимумом функционала (5). Для нахождения же глобального минимума (5) в работах [3, 4] был предложен гибридный метод настройки МП, схема которого представлена на рис. 4.

Основными этапами работы алгоритма являются: инициализация популяции, оценивание популяции (вычисление функции приспособленности), селекция, скрещивание, мутации. Критерием останова алгоритма обычно являются либо прохождение максимально допустимого количества поколений g , либо достижение функцией приспособленности какой-либо особи некоторого заранее заданного порогового значения. В начале работы НА случайным образом инициализируется популяция P_0 , состоящая из N особей (МП): $P_0 = \{H_1, H_2, \dots, H_N\}$. Каждая особь в популяции при этом получает свое уникальное описание, закодированное в хромосоме $H_j = \{h_{1j}, h_{2j}, \dots, h_{Lj}\}$, которая состоит из L генов, где $h_{ij} \in [w_{\min}, w_{\max}]$ – значение i -го гена j -й хромосомы (w_{\min} – минимальное, и w_{\max} – максимальное допустимые значения соответственно). Формат хромосомы и соответствие между генами и параметрами МП представлен на рис. 5.

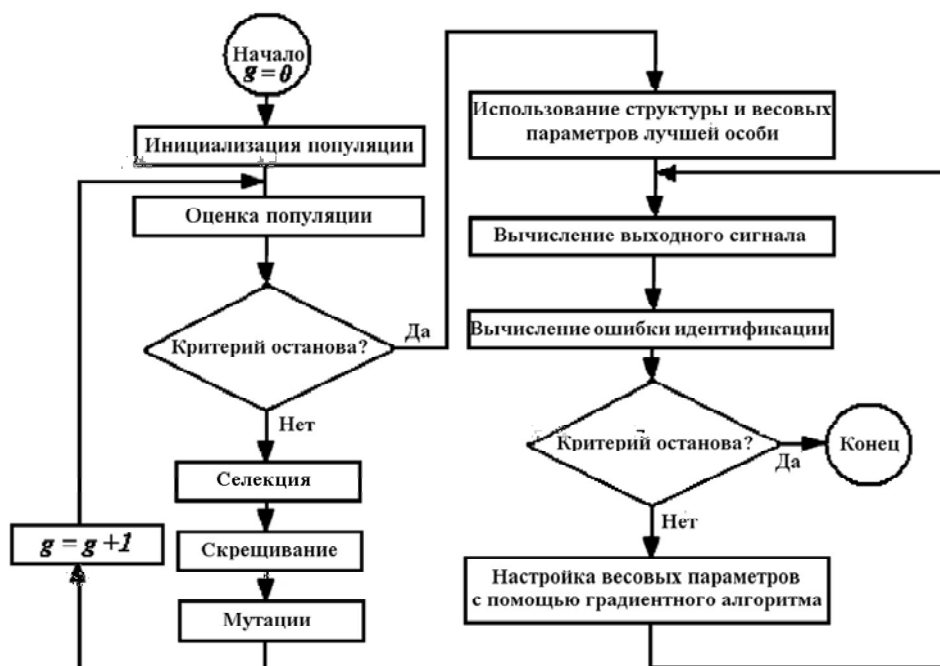


Рис. 4. Схема гибридного метода настройки МП

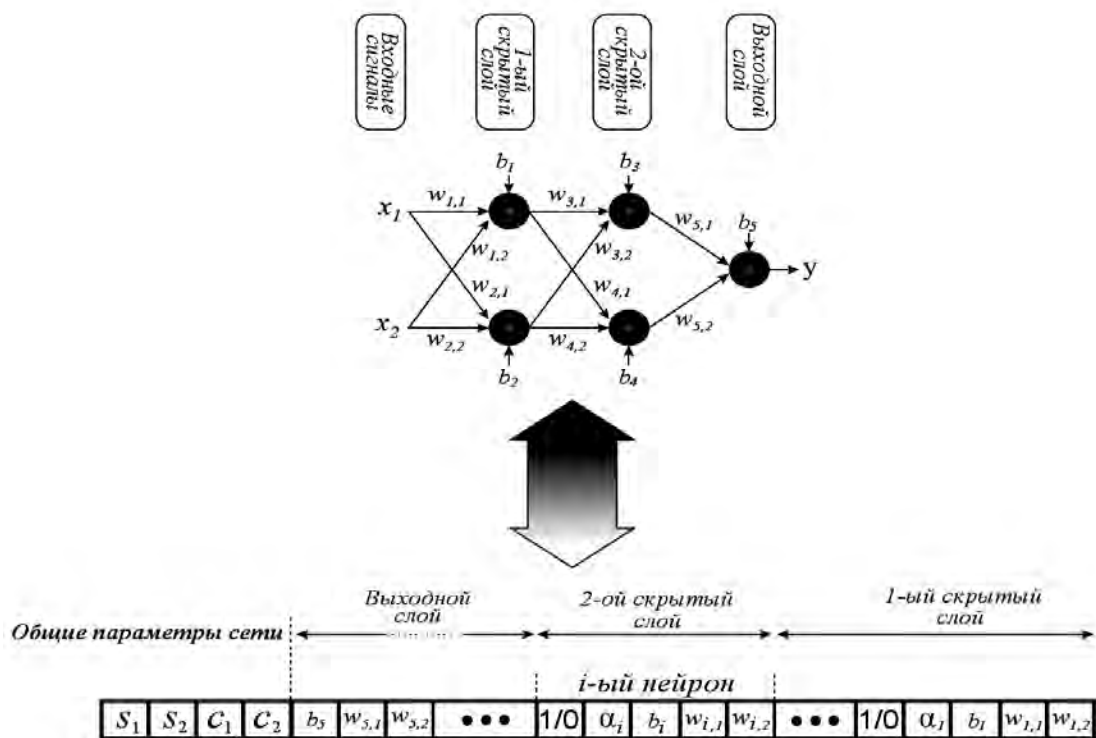


Рис. 5. Формат хромосомы и соответствие между генами и параметрами МП

Следует отметить, что длина хромосомы ограничивается максимально допустимым количеством нейронов. Как видно из рис. 5, каждая хромосома состоит из генов, в которых хранится информация о соответствующих параметрах сети. Первый ген кодирует информацию о смещении нейрона выходного слоя сети. Затем идут блоки генов, кодирующие параметры соответствующих нейронов скрытого слоя. Первый ген каждого такого блока (1/0) определяет, присутствует ли соответствующий нейрон в

структуре сети, т.е. участвует он или нет в вычислении выходной реакции сети на поступивший входной сигнал. Следующий ген (α) определяет форму базисной функции (4) данного нейрона.

Затем в хромосоме идет группа генов, кодирующая непосредственно весовые параметры соответствующего нейрона. Следует отметить, что количество этих параметров, а следовательно, и длина хромосомы, зависит от размерности идентифицируемого объекта. На этапе инициализации всем этим

параметрам с помощью датчика случайных чисел присваивают начальные значения, находящиеся в некотором допустимом диапазоне.

После того, как начальная популяция сформирована, производится оценка приспособленности каждой входящей в нее особи.

Для этого используется функция приспособленности (фитнесс-функция) вида

$$f_i(x_j) = \frac{1}{M} \sum_{j=1}^M |y_j^*(x_j) - \hat{y}_j(x_j)|, \quad (6)$$

где M – размер выборки.

Следующим шагом является отбор особей, хромосомы которых будут принимать участие в формировании нового поколения, и их последующее скрещивание. Задача оператора скрещивания (кроссовера) заключается в передаче генетической информации от родительских особей к их потомкам. После завершения работы оператора кроссовера любой ген любой особи в новой популяции может мутировать, т.е. изменить свое значение.

Моделирование

Решалась задача управления нелинейным объектом, описываемым уравнением

$$y(k+1) = \sin(u(k)) + \frac{y(k)}{1+y^2(k)}, \quad (7)$$

где $u(k)$ – сигнал управления.

Начальная популяция состояла из 128 особей (ИНС). В качестве ИНС использовался МП с двумя скрытыми слоями, в каждом из которых допускалось не более 10 нейронов. На каждой итерации алгоритма каждой особи предъявлялось $M = 5000$ обучающих пар.

На рис. 6, а представлена поверхность, описываемая уравнением (7), а на рис. 6, б – поверхность, восстановленная с помощью эволюционирующего многослойного персептрона. Управляющее воздействие вычислялось по формуле

$$u(k+1) = u(k) + \gamma \nabla_u \hat{y}(k) e_y(k), \quad (8)$$

где γ – параметр, влияющий на скорость настройки алгоритма; $e_y(k) = y(k) - r(k)$; $r(k)$ – требуемый выходной сигнал объекта.

Желаемый закон изменения выходного сигнала объекта задавался следующим образом:

$$r(k) = 0.7 \sin(\pi k / 100).$$

Результаты управления объектом (7) при различных значениях горизонта предсказания приведены на рис. 7. Так для рис. 7, а $N_p = 5$, для рис. 7, б – $N_p = 20$, а для рис. 7, в – $N_p = 30$. На всех рисунках сплошной линией обозначен желаемый выходной сигнал, пунктирной – реальный выходной сигнал, сплошной с кружками – управляющий сигнал.

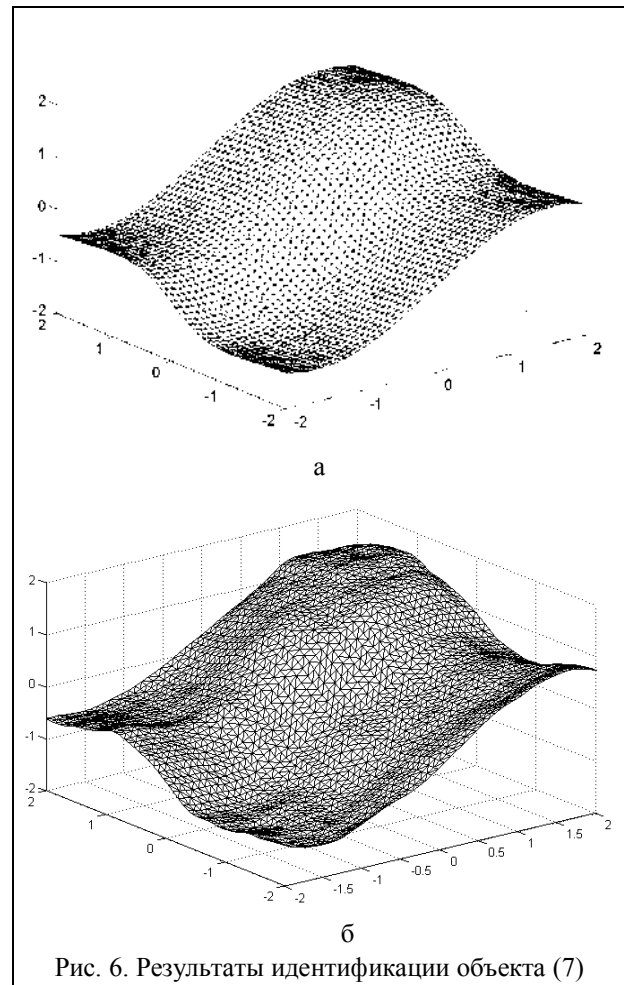


Рис. 6. Результаты идентификации объекта (7)

Выводы

Применение генетических алгоритмов (ГА) для обучения ИНС является достаточно эффективным, так как позволяет сравнительно просто определить (или перестроить в нестационарном случае) как структуру сети, так и ее параметры. Однако при этом возникает ряд вопросов, связанных с размером используемой популяции, выбором способа кодирования параметров сети и требуемой точностью их представления, выбором типа воспроизводства и критерия останова процесса. Обычно ГА кодируют параметры ИНС строкой (битовой) или списком ее параметров.

Хотя для эффективной работы ГА требуется значительное количество строк (хромосом), представляющих большое число наборов параметров данной сети, реализация алгоритма затруднений не вызывает, так как использует параллельные вычисления.

Как видно из результатов моделирования, эволюционирующий МП, использующий ГА как для определения структуры нейросетевой модели, так и оценивания ее параметров, обладает высокой скоростью обучения и способен решать задачу управления нелинейными объектами с прогнозирующей моделью.

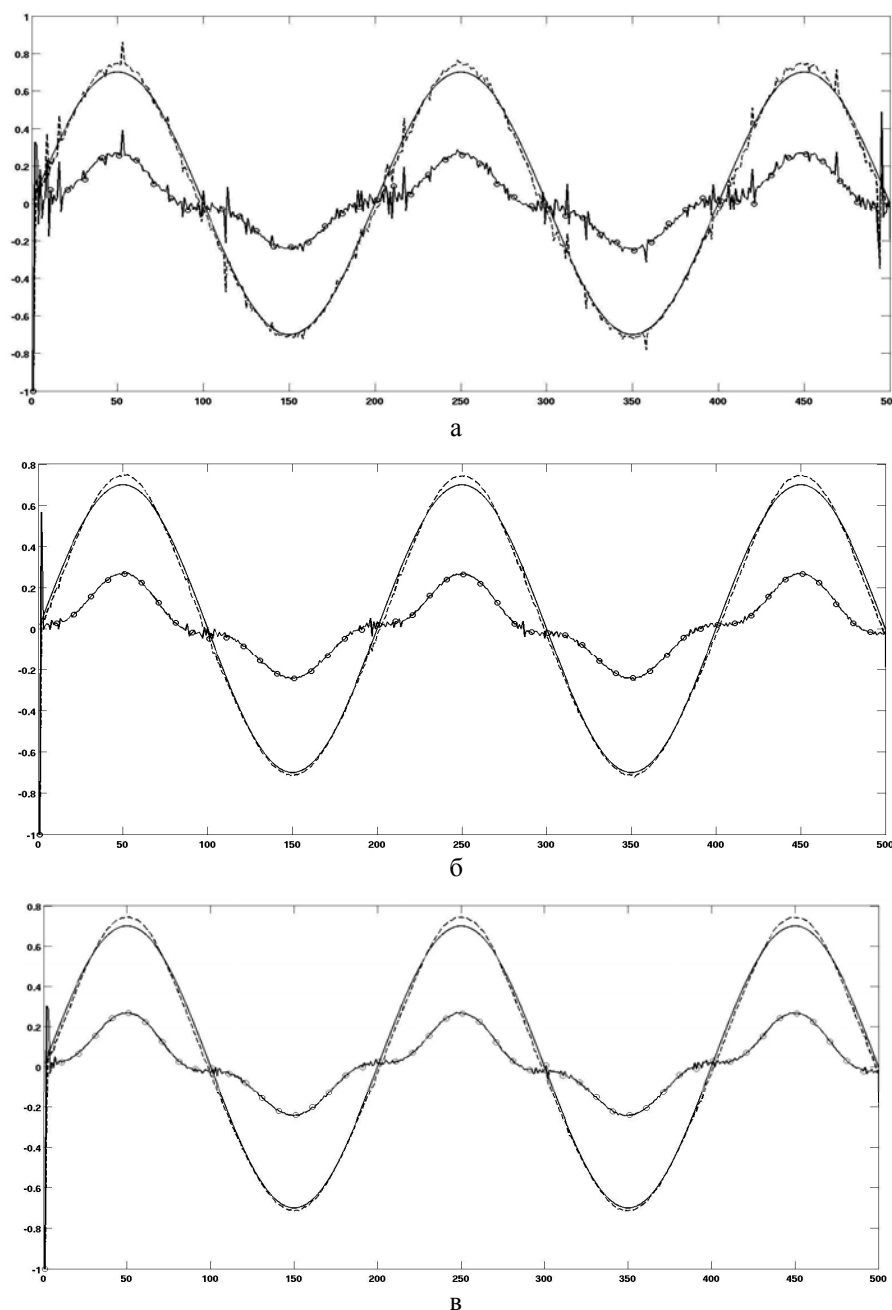


Рис. 7. Результати управління об'єктом (7)

РІШЕННЯ ЗАДАЧІ КЕРУВАННЯ З ПРОГНОЗУЮЧОЮ МОДЕЛЛЮ НА ОСНОВІ БАГАТОШАРОВОГО ПЕРСЕПТРОНУ, ЩО ЕВОЛЮЦІОНУЄ

О.О. Безсонов

Стаття присвячена розробці методу керування нелінійними об'єктами за допомогою багатошарового перцептрону, що еволюціонує. Багатошаровий перцептрон застосовується для побудови нелінійної моделі об'єкта, яка згодом використовується для рекурсивного передбачення поведінки об'єкта в системі керування з прогнозуючою моделлю. Для навчання нейронної мережі пропонується застосування генетичного алгоритму, що дозволяє значно прискорити процес навчання. Наведені результати імітаційного моделювання, що підтверджують ефективність запропонованого методу керування.

Ключові слова: прогнозуюча модель, генетичний алгоритм, багатошаровий перцептрон.

THE MODEL PREDICTIVE CONTROL PROBLEM SOLVING ON THE BASIS OF THE EVOLVING MULTILAYER PERCEPTRON

O.O. Bezsonov

The article is devoted to development of a method of nonlinear objects control with the evolving multilayer perceptron. The evolving multilayer perceptron is used to build a nonlinear model of the object, which is then used to predict the behavior of a recursive object in the model predictive control. For the neural network training a genetic algorithm is used, what significantly speed up the training process. The simulation results confirm the effectiveness of the proposed control method.

Keywords: forecasting model, genetic algorithm, multi-layered perceptron.

Список литератури

1. Potocnik P. Nonlinear model predictive control of a cutting process / P. Potocnik, I. Grabec // *Neurocomputing*. – 2002. – Vol. 43. – P. 107-126.

2. Бессонов А.А. Идентификация нелинейных нестационарных объектов с помощью эволюционного многослойного перцептрона / А.А. Бессонов, С.О. Руденко // *Вестник ХНТУ*. – 2012. – № 1 (44). – С. 130-135.

3. Braun H. ENZO-M – a hybrid approach for optimizing neural networks by evolution and learning / H. Braun, P. Zagorski // *Proc. of the parallel Problem Solving from Nature*. – Berlin: Springer-Verlag, 1994. – P. 440-451.

4. Guo L. Combining genetic optimization with hybrid learning algorithm for radial basis function neural networks / L. Guo, D.-S. Huang, W. Zhao // *Electron. Lett.* – 2003. – Vol. 39, № 22. – P. 1600-1601.

Поступила
в редколлегию 25.12.2014

Рецензент: д-р техн. наук, проф. О.Г. Руденко, Харьковский национальный университет радиоэлектроники, Харьков.