



АППРОКСИМАЦИЯ ГАУССОВСКИХ БАЗИСНЫХ ФУНКЦИЙ В ЗАДАЧЕ АДАПТИВНОГО УПРАВЛЕНИЯ НЕЛИНЕЙНЫМИ ОБЪЕКТАМИ

Ключевые слова: *нейронная сеть, обучение, нелинейный объект, идентификация, адаптивное управление, базисная функция.*

Введение. Основой классической теории управления является теория линейных систем. Зачастую элементы этой теории используются и при исследовании нелинейных объектов, однако хорошие результаты могут быть получены лишь в тех случаях, когда нелинейность объекта несущественна либо когда исследуемый объект характеризуется большими постоянными времени и устойчивостью в разомкнутом состоянии. Отсутствие достаточно полной информации о свойствах исследуемых объектов, условиях их функционирования обусловили развитие альтернативного нейросетевого подхода к синтезу систем управления нелинейными объектами, основанного на использовании искусственных нейронных сетей (ИНС) [1, 2]. Данный подход, как и традиционный, требует построения модели (нейросетевой) исследуемого объекта, служащей основой для синтеза нейросетевого регулятора.

Среди большого количества существующих ИНС для решения задач управления нелинейными динамическими объектами достаточно широкое распространение получили радиально-базисные сети (РБС) [3–11]. Это обусловлено прежде всего простотой их архитектуры, хорошими аппроксимирующими свойствами и наличием эффективных алгоритмов оптимизации, в частности рекуррентных, обеспечивающих быстрое обучение данного типа. РБС осуществляют аппроксимацию нелинейного оператора исследуемого объекта реализуемой нейронами некоторой системой базисных функций (БФ) — нелинейных функций $\Phi(\mathbf{x}, \boldsymbol{\mu})$, зависящих только от расстояния (радиального) $r = \|\mathbf{x} - \boldsymbol{\mu}\|$, где \mathbf{x} — вектор независимых переменных, $\boldsymbol{\mu}$ — вектор постоянных параметров (центров БФ). Веса и параметры данных функций определяются в результате обучения сети, заключающемся в минимизации некоторого критерия обучения. Хотя в настоящее время существует много различных БФ, наибольшее распространение получила гауссовская БФ (ГБФ), которая, как показано в [11], оптимальна в случае нормально распределенных входных сигналов и применения квадратичного критерия обучения. Однако вычисление данных БФ связано со значительными временными затратами, что приводит, с одной стороны, к затягиванию процесса обучения (идентификации), а с другой — к запаздыванию управления.

Цель данной работы — исследование возможностей ускорения процесса обучения РБС и упрощения вычисления значений управляющих сигналов путем использования кусочно-линейной аппроксимации гауссовских БФ.

Задача управления. Рассмотрим нелинейный динамический объект, описываемый уравнением

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

где $y(i)$ и $u(i)$ — выходные и управляющие сигналы соответственно; $f(\cdot)$ — неизвестная нелинейная функция; $\xi[k]$ — помеха.

Задача управления состоит в поиске управляющего воздействия $u(k)$, минимизирующего ошибку $\varepsilon[k] = y^*[k] - y[k]$, где y^* — требуемое значение выходного сигнала. Это эквивалентно задаче оптимизации

$$\min_{u(k)} |y^*[k] - y[k]| \quad (2)$$

или

$$\min_{u(k)} |y^*[k+1] - f(y[k-1], \dots, y[k-m], u[k-1], \dots, u[k-n])|. \quad (3)$$

Нейросетевая модель. Обозначив $M = (m+n-1)$ -мерный вектор обобщенного сигнала, поступающего на вход модели, как

$$\mathbf{x}[k] = (y[k-1], \dots, y[k-m], u[k-1], \dots, u[k-n])^T, \quad (4)$$

уравнение (1) перепишем в виде

$$y[k] = f(\mathbf{x}[k]) + \xi[k]. \quad (5)$$

Задача идентификации заключается в оценивании функции $f(\cdot)$ по измерениям входных $u[k]$ и выходных $y[k]$ переменных.

Аппроксимация нелинейности $f(\cdot)$ радиальными базисными функциями $\Phi_i(x, \mu)$ приводит к нейросетевым моделям нулевого

$$\hat{y}[k] = \sum_{i=1}^N c_i \Phi_i(\mathbf{x}, \mu) = \mathbf{c}^T \Phi(\mathbf{x}, \mu) \quad (6)$$

и первого

$$\hat{y}[k] = (\mathbf{c} + V\mathbf{x})^T \Phi(\mathbf{x}, \mu) \quad (7)$$

порядков.

Здесь \mathbf{c} и V — подлежащие определению вектор и матрица весовых коэффициентов соответственно; N — количество нейронов.

Как отмечалось выше, чаще всего используются гауссовские БФ (ГБФ) вида

$$\varphi(x, \mu) = \exp\left\{-\frac{(x-\mu)^2}{\sigma^2}\right\}, \quad (8)$$

где μ и σ — центры и радиусы БФ соответственно.

Если же при исследовании объектов осуществляется нормализация сигналов, целесообразно применение модифицированной гауссовской базовой функции (МГБФ) вида

$$\hat{\varphi}(x, \lambda_1, \lambda_2) = \begin{cases} \exp\left\{-\frac{(\lambda_2 - \lambda_1)^2 / 4}{(x - \lambda_1)(\lambda_2 - x)}\right\} & \text{при } x \in (\lambda_1, \lambda_2), \\ 0 & \text{в противном случае,} \end{cases} \quad (9)$$

которая, в отличие от (8), равна нулю при $x \notin (\lambda_1, \lambda_2)$.

В ряде случаев, в частности, когда центры БФ распределены равномерно, а радиусы отличаются незначительно, весьма эффективно использование нормализованных БФ (НБФ)

$$\bar{\varphi}_i(x, \mu_i) = \frac{\varphi_i(x, \mu_i)}{\sum_{m=1}^N \varphi_m(x, \mu_m)}. \quad (10)$$

Благодаря свойству

$$\sum_{m=1}^N \varphi_m(x, \mu_m) = 1$$

данная сеть используется во многих приложениях.

Отметим, что в случае M -мерного сигнала $x[k]$ БФ i -го нейрона имеет вид

$$\Phi_i(\mathbf{x}, \boldsymbol{\mu}) = \prod_{m=1}^M \varphi_{im}(x_m, \mu_{im}), \quad (11)$$

т.е.

$$\Phi_i(\mathbf{x}[k], \boldsymbol{\mu}) = \prod_{j=1}^M \exp \left\{ -\frac{(x_j[k] - \mu_{ij})^2}{\sigma_{ij}^2} \right\} \quad (12)$$

при выборе ГБФ и

$$\hat{\Phi}_i(\mathbf{x}[k], \lambda_1, \lambda_2) = \begin{cases} \prod_{m=1}^M \exp \left\{ -\frac{(\lambda_{im,2} - \lambda_{im,1})^2 / 4}{(x_m[k] - \lambda_{im,1})(\lambda_{im,2} - x_m[k])} \right\} & \text{при } \mathbf{x}[k] \in (\lambda_{i1}, \lambda_{i2}), \\ 0 & \text{в противном случае} \end{cases} \quad (13)$$

при выборе МГБФ.

НБФ будет иметь вид

$$\bar{\Phi}_i(\mathbf{x}[k], \boldsymbol{\mu}_i) = \frac{\Phi_i(\mathbf{x}[k], \boldsymbol{\mu}_i)}{\sum_{m=1}^N \Phi_m(\mathbf{x}[k], \boldsymbol{\mu}_m)}. \quad (14)$$

Применение нормализованных РБС (НРБС), в отличие от РБС, обеспечивает в равной степени покрытие всех точек входного пространства, делая сеть менее чувствительной к неудачному выбору центров. Однако нормализация существенно изменяет форму БФ, причем на эту форму влияет как ширина исходных БФ (при уменьшении σ_i^2 форма нормализованных функций становится более прямоугольной), так и их взаимное расположение, характеризуемое параметрами μ_i . При неравномерном распределении центров исходных БФ или при их различной ширине максимумы НБФ могут быть опущены по отношению к центрам, а сами НБФ могут стать немонотонно убывающими. Последнее приводит к неправильной реакции нейрона на входные сигналы.

Изменение структуры сети осуществляется ее постепенным усложнением путем добавления новых нейронов, проводимого каждый раз, когда при появлении очередного входного сигнала возникает ошибка идентификации $e = y - \hat{y}$, превышающая допустимую. В этом случае, если в l -й момент времени сеть содержала N нейронов, а появление сигнала $\mathbf{x}[l]$ привело к появлению ошибки на выходе $e[l] > e_{\text{доп}} = \alpha$, в сеть вводится новый, $(N+1)$ -й, нейрон, центр базисной функции $\boldsymbol{\mu}_{N+1}$ которого принимается равным $\boldsymbol{\mu}_{N+1} = \mathbf{x}[l]$, вес — $c_{N+1} = e[l]$ и $\sigma_{N+1} = \|\mathbf{x}[l] - \boldsymbol{\mu}_m[l]\|$, где $\boldsymbol{\mu}_m[l]$ — центр базисной функции для l -го входа сигнала. Таким образом, условием введения нового нейрона является выполнение неравенств

$$e(l) > \alpha, \quad (15)$$

$$\|\mathbf{x}[l] - \boldsymbol{\mu}_m[l]\| > \beta, \quad (16)$$

где α и β — априорно устанавливаемые предельно допустимые значения ошибки реакции сети и отклонения обобщенного сигнала $\mathbf{x}(l)$ от ближайшего к данному входу центра.

Обучение сети. При использовании нейросетевого подхода задача идентификации, в свою очередь, сводится к обучению сети, состоящему в настройке ее параметров на основе сравнения выходных сигналов объекта $y[k]$ и модели $\hat{y}[k]$ и заключающемся в минимизации обычно квадратичного функционала ошибки $l[k] = e[k]$, где $e[k] = y[k] - \hat{y}[k]$.

Обучение состоит в определении параметров сети μ_i, σ_i, c_i и может сводиться к одному из следующих вариантов [12]:

- центры и отклонения задаются, а вычисляются только веса выходного слоя;
- центры и отклонения определяются путем самообучения, а коррекция весов выходного слоя осуществляется путем обучения с учителем;
- все параметры сети определяются с помощью обучения с учителем.

Первые два варианта применяются в сетях, использующих БФ с жестко заданными отклонениями, третий предполагает использование любых БФ.

Хотя третий вариант обучения является наиболее сложным и трудоемким, результаты, получаемые при этом, оказываются в большинстве своем значительно лучшими. Наиболее простой алгоритм обучения — градиентный, в соответствии с которым коррекция всех параметров сети осуществляется по правилу (ниже для упрощения обозначено $\Phi[k] = \Phi(\mathbf{x}[k], \mu, \sigma)$):

$$\begin{aligned} \Delta c_i[k] &= -\gamma_c e[k] \frac{\partial \hat{y}(k)}{\partial c_i}, \quad \Delta v_{ij}[k] = -\gamma_v e[k] x_i[k] \tilde{\Phi}_j[k], \\ \Delta \mu_{ij}[k] &= -\gamma_\mu e[k] \frac{\partial \hat{y}[k]}{\partial \mu_{ij}}, \quad \Delta \sigma_{ij}[k] = -\gamma_\sigma e[k] \frac{\partial \hat{y}[k]}{\partial \sigma_{ij}}. \end{aligned} \quad (17)$$

Здесь

$$\begin{aligned} \frac{\partial \hat{y}[k]}{\partial c_i} &= \tilde{\Phi}_i[k]; \quad \frac{\partial \hat{y}[k]}{\partial \mu_{ij}} = \tilde{c}_i \frac{\partial \tilde{\varphi}_{ij}[k]}{\partial \mu_{ij}} \prod_{\substack{m=1 \\ m \neq j}}^M \tilde{\varphi}_{jm}[k], \\ \frac{\partial \hat{y}[k]}{\partial \sigma_{ij}} &= \tilde{c}_i \frac{\partial \tilde{\varphi}_{ij}[k]}{\partial \sigma_{ij}} \prod_{\substack{m=1 \\ m \neq j}}^M \tilde{\varphi}_{jm}[k], \\ \tilde{\Phi}_i[k] &= \begin{cases} \Phi_i[k] & \text{для РБС,} \\ \bar{\Phi}_i[k] & \text{для НРБС,} \end{cases} \\ \tilde{c}_i &= \begin{cases} c_i & \text{для РБС нулевого порядка,} \\ c_i + \sum_{j=1}^N v_{ij} x_j[k] & \text{для РБС первого порядка,} \\ \frac{c_i \sum_{j=1}^N c_j[k] \Phi_j[k]}{\sum_{m=1}^M \Phi_m[k] \left(\sum_{m=1}^M \Phi_m[k] \right)^2} & \text{для НРБС нулевого порядка,} \\ \frac{c_i + \sum_{j=1}^N v_{ij} x_j[k]}{\sum_{i=1}^N \left(c_i + \sum_{j=1}^N v_{ij} x_j[k] \right)} \Phi_j[k] & \text{для НРБС первого порядка,} \\ \frac{\sum_{m=1}^M \Phi_m[k]}{\left(\sum_{m=1}^M \Phi_m[k] \right)^2} & \end{cases} \end{aligned}$$

$\gamma_c, \gamma_v, \gamma_\mu, \gamma_\sigma$ — итерационные параметры, влияющие на скорость обучения.

Например, при использовании ГБФ (6) в сети нулевого порядка имеем

$$\begin{aligned} \frac{\partial \hat{y}[k]}{\partial \mu_{ij}} &= c_i \frac{2(x_j[k] - \mu_{ij})}{\sigma_{ij}^2} \prod_{m=1}^M \varphi_{im}(x_m[k]), \\ \frac{\partial \hat{y}[k]}{\partial \sigma_{ij}} &= c_i \frac{2(x_j[k] - \mu_{ij})}{\sigma_{ij}^3} \prod_{m=1}^M \varphi_{im}(x_m[k]). \end{aligned}$$

После окончания процесса обучения параметры сети используются для реализации алгоритма управления.

Алгоритм управления. Так как задача управления заключается в определении значений $u(k)$, обеспечивающих минимум функционалу

$$J[k] = (\varepsilon[k])^2 = \frac{1}{2} (y^*[k] - \hat{y}[k])^2,$$

как и в случае идентификации, для нахождения его минимума могут использоваться различные рекуррентные алгоритмы, в частности градиентные:

$$\Delta_u = -\gamma_u \varepsilon[k] \frac{\partial \hat{y}[k]}{\partial u}, \quad (18)$$

где γ_u — итерационный параметр, влияющий на скорость сходимости.

Выражение для алгоритма управления можно легко получить с учетом того, что для РБС нулевого порядка

$$\left. \frac{\partial \hat{y}[k]}{\partial x_i} \right|_{x_i=x_i[k]} = \sum_{j=1}^N c_j \prod_{\substack{m=1 \\ m \neq i}}^M \tilde{\varphi}_{jm}(x_m[k]) \frac{\partial \tilde{\varphi}_{ij}(x_j[k])}{\partial x_i[k]}, \quad (19)$$

а для РБС первого порядка

$$\left. \frac{\partial \hat{y}[k]}{\partial x_i} \right|_{x_i=x_i[k]} = \sum_{j=1}^N \left\{ v_{ij} \tilde{\varphi}_j(x_i[k]) + \left(c_j + \sum_{l=1}^N v_{li} x_l[k] \right) \prod_{\substack{m=1 \\ m \neq i}}^M \tilde{\varphi}_{jm}(x_m[k]) \frac{\partial \tilde{\varphi}_{ij}(x_j[k])}{\partial x_i[k]} \right\}. \quad (20)$$

Как следует из приведенных формул, и при обучении сети, и при реализации алгоритма управления необходимо вычисление экспоненциальных БФ и их производных, что требует значительных временных затрат, существенно возрастающих с ростом размерности объекта (1) и увеличением количества БФ. Один из возможных способов упрощения вычислений — использование кусочно-линейной аппроксимации БФ.

Аппроксимация БФ. Достаточно простой и в то же время эффективной является следующая аппроксимация:

ГБФ (8) [11]

$$\varphi(x) = \begin{cases} \frac{0.77}{1.2\sigma} (x - \mu + 1.7\sigma) & \text{при } c - 1.7\sigma \leq x < c - 0.5\sigma, \\ 0.77 + \frac{0.23}{0.5\sigma} (x - \mu + 0.5\sigma) & \text{при } c - 0.5\sigma \leq x < c, \\ 0.77 + \frac{0.23}{0.5\sigma} (x - \mu - 0.5\sigma) & \text{при } c \leq x < c + 0.5\sigma, \\ \frac{0.77}{1.2\sigma} (x - \mu - 1.7\sigma) & \text{при } c + 0.5\sigma \leq x < c + 1.7\sigma; \end{cases} \quad (21)$$

МГБФ (9)

$$\varphi(x) = \begin{cases} 0 & \text{при } c - \sigma \leq x < c - 0.9\sigma, \\ 0.73 + \frac{0.73}{0.4\sigma} (x - \mu + 0.5\sigma) & \text{при } c - 0.9\sigma \leq x < c - 0.5\sigma, \\ 0.91 + \frac{0.18}{0.2\sigma} (x - \mu + 0.3\sigma) & \text{при } c - 0.5\sigma \leq x < c - 0.3\sigma, \\ 0.99 + \frac{0.08}{0.2\sigma} (x - \mu + 0.1\sigma) & \text{при } c - 0.3\sigma \leq x < c - 0.1\sigma, \\ 0.99 & \text{при } c - 0.1\sigma \leq x < c + 0.1\sigma, \\ 0.99 - \frac{0.08}{0.2\sigma} (x - \mu - 0.1\sigma) & \text{при } c + 0.1\sigma \leq x < c + 0.3\sigma, \\ 0.91 - \frac{0.18}{0.2\sigma} (x - \mu - 0.3\sigma) & \text{при } c + 0.3\sigma \leq x < c + 0.5\sigma, \\ 0.73 - \frac{0.73}{0.4\sigma} (x - \mu - 0.5\sigma) & \text{при } c + 0.5\sigma \leq x < c + 0.9\sigma, \\ 0 & \text{при } c + 0.9\sigma \leq x < c + \sigma. \end{cases} \quad (22)$$

В (22) $\sigma = \lambda_2 - \lambda_1$.

Аппроксимация НБФ осуществляется с помощью соотношений (21), (22), а также выражения (14).

На рис. 1 сплошными линиями показаны ГБФ (8) (рис. 1, *a*), ее производные по центрам (рис. 1, *б*) и радиусам (рис. 1, *в*), пунктирными — соответствующие аппроксимации. Аналогично на рис. 2 сплошная линия отражает зависимости МГБФ (9) (рис. 2, *a*) и ее производных (рис. 2, *б* и 2, *в*), пунктирная линия — соответствующие аппроксимации. На рис. 3 сплошная линия показывает виды двух (рис. 3, *a*) и трех (рис. 3, *б*) НБФ, пунктирная линия — их аппроксимации.

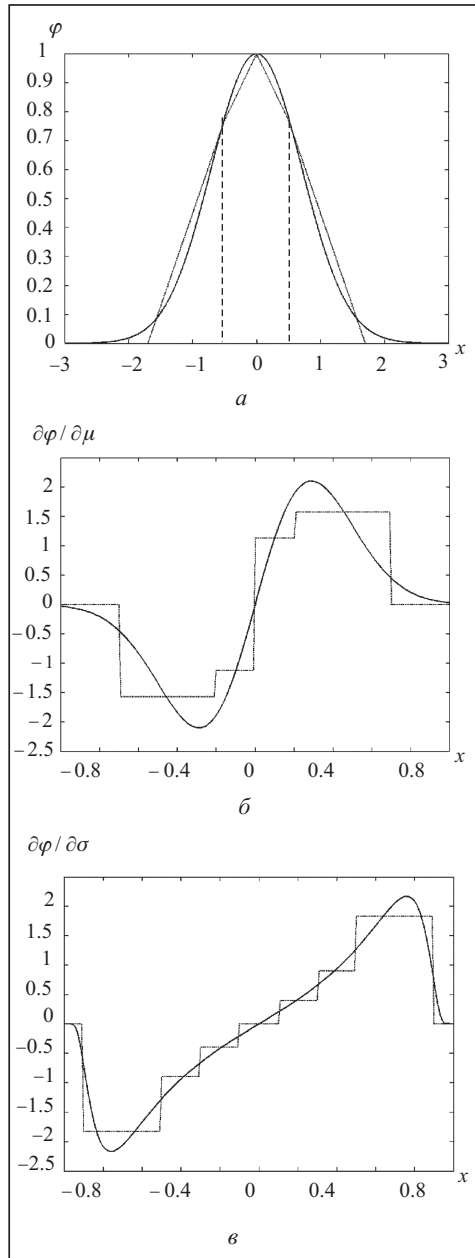


Рис. 1

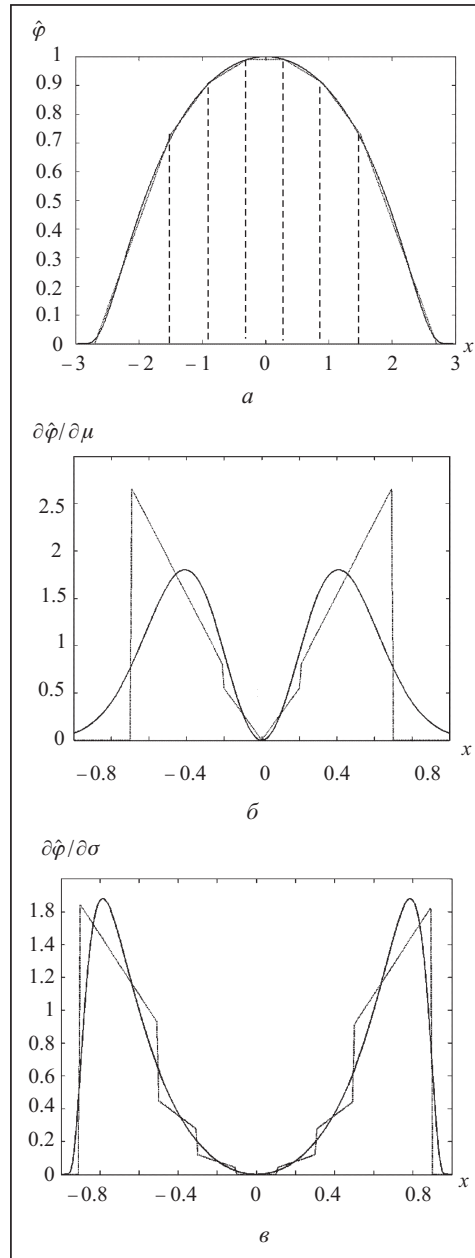


Рис. 2

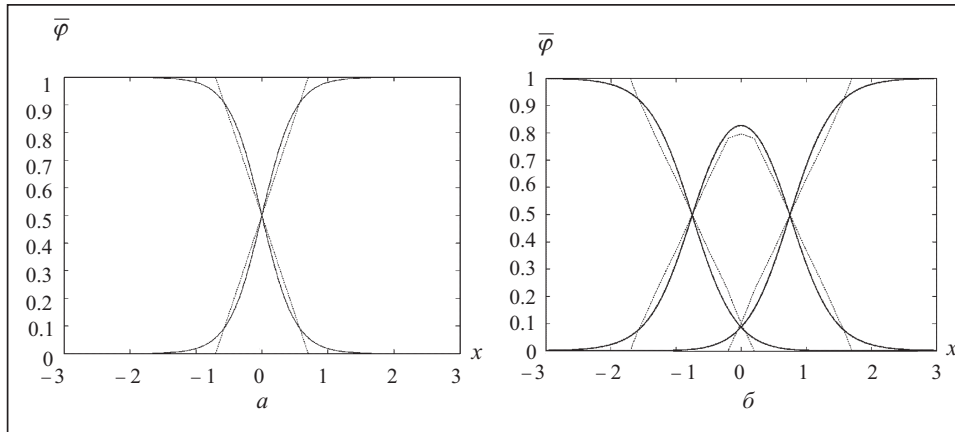


Рис. 3

Моделирование. Эксперимент 1. Рассматривалась задача идентификации и управления нелинейным динамическим объектом, описываемым уравнением [2, 10]

$$y[k] = 0.725\beta[k] \sin \left(\frac{16u[k-1] + 8y[k-1]}{\beta[k](3 + 4u^2[k-1] + 4y^2[k-1])} \right) + 0.2u[k-1] + 0.2y[k-1] + \xi[k], \quad (23)$$

где $\beta[k] = 0.8$ — параметр, задающий степень нестационарности объекта; $\xi[k]$ — равномерно распределенная помеха в интервале $[-0.3; 0.3]$, $u[k]$ — входной сигнал, который представлял собой стационарную случайную последовательность с равномерным законом распределения в интервале $[-1, 1]$, генерируемую датчиком случайных чисел.

При исследовании данного объекта использовалось 10 тыс. обучающих пар. Результаты идентификации объекта (23) с помощью РБС нулевого порядка и с использованием ГБФ приведены на рис. 4. Поверхность, описываемая (23), при наличии и отсутствии помехи $\xi[k]$ показана на рис. 4, а и рис. 4, б соответственно. Результаты идентификации зашумленного объекта (23) с использованием алгоритма (17) представлены на рис. 4, в. График изменения ошибки идентификации, которая вычислялась на каждом шаге по формуле

$$e[k] = \lg \left(\frac{\sqrt{\sum_{i=1}^k e^2[i]}}{k} \right), \quad (24)$$

показан на рис. 4, г.

График управления объектом (23) с помощью алгоритма (18) при задании в качестве эталонного управляющего сигнала

$$y^*[k] = 0.48 + 0.07 \sin[\pi k / 200] - 0.05 \cos[\pi k / 100] \quad (25)$$

приведен на рис. 5. Сплошными линиями показаны реальные сигналы, пунктирными — желаемый сигнал (25), а линией с кружками — рассчитанный с помощью (18) управляющий сигнал (эти же обозначения приняты на рис. 6 и 7). На рис. 5, а показаны результаты управления объектом после 10 тыс. шагов предварительной идентификации, а результаты на рис. 5, б отражают поведение системы управления в случае, когда на первых 500 шагах осуществляется идентификация объекта, а затем одновременно проводится идентификация и управление.

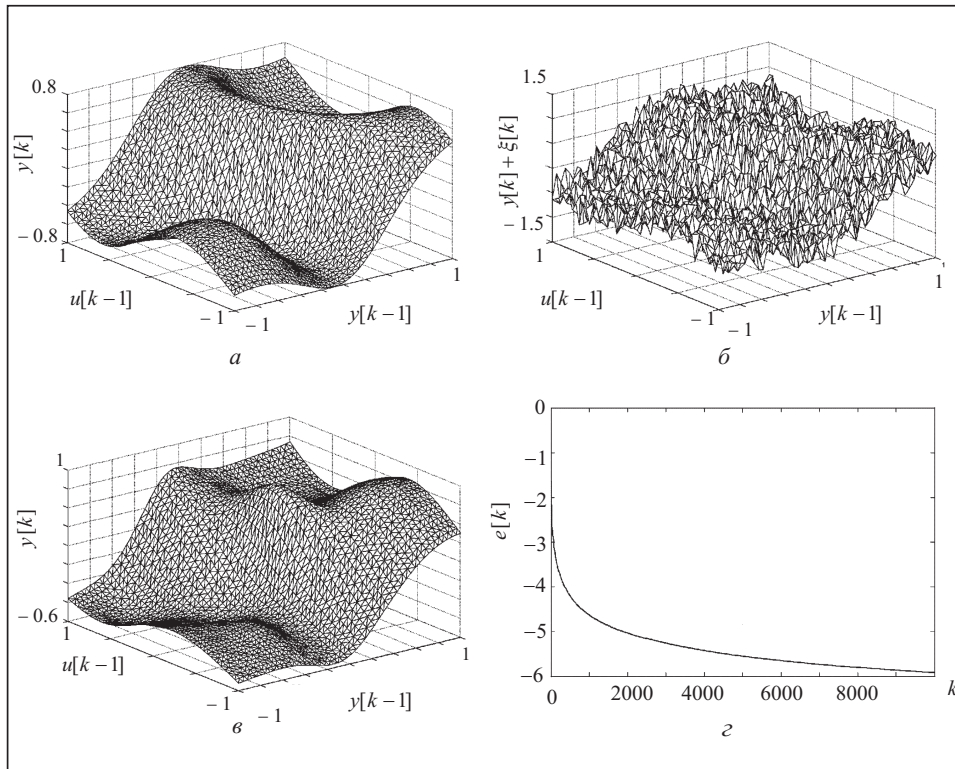


Рис. 4

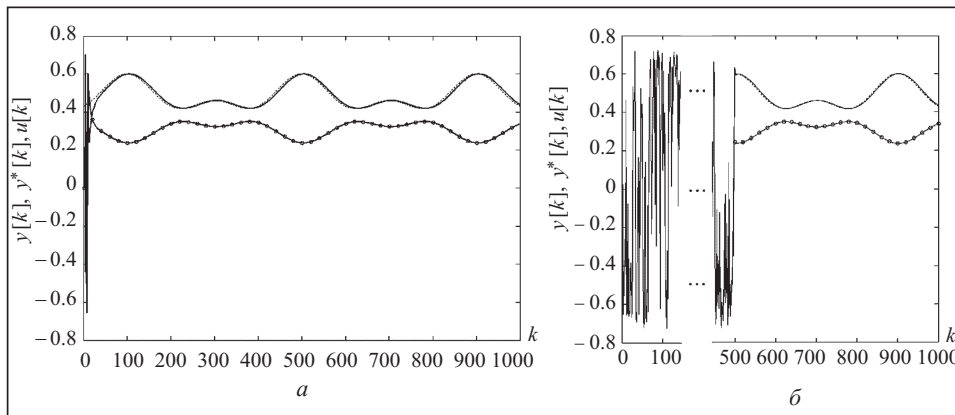


Рис. 5

Результаты идентификации и управления объектом (23) с применением РБС различных порядков, различных БФ, с использованием и без аппроксимации и нормализации сведены в табл. 1. Данные для идентификации получены с помощью формулы (24) на последнем 10-тысячном шаге обучения сети, а для управления — на последнем 1000-м шаге управления.

Эксперимент 2. Идентифицируемый объект описывался уравнением

$$y[k] = \frac{u[k-1] y[k-1] (u[k] + 2.5)}{1 + u^2[k-1] + y^2[k-1]}, \quad (26)$$

где $u(k)$ — сигнал управления.

Таблица 1. Результаты идентификации и управления объектом (23)

Порядок РБС	БФ	Наличие нормализации	Наличие аппроксимации	Время, с		Значение критерия (24)	
				Идентификация	Управление	Идентификация	Управление
0	ГБФ	–	–	54.703	1.625	–7.2702	–14.374
0	ГБФ	–	+	52.812	1.500	–7.1879	–10.649
0	ГБФ	+	–	58.109	4.000	–7.2863	–14.555
0	ГБФ	+	+	55.656	3.562	–7.1931	–13.495
1	ГБФ	–	–	60.329	2.164	–7.3318	–14.694
1	ГБФ	–	+	56.891	1.812	–7.1762	–14.517
0	МГБФ	–	–	59.688	1.863	–7.2305	–12.945
0	МГБФ	–	+	56.594	1.737	–7.1553	–12.456
0	МГБФ	+	–	62.734	4.406	–7.1149	–13.777
0	МГБФ	+	+	60.140	4.225	–7.0889	–13.052
1	МГБФ	–	–	64.578	2.386	–7.3261	–14.062
1	МГБФ	–	+	59.812	1.910	–7.2171	–13.860

В качестве входного сигнала $u(k)$ при идентификации выбиралась стационарная случайная последовательность с равномерным законом распределения в интервале $[-5, 5]$. Идентификация данного объекта осуществлялась на основе 10 тыс. обучающих пар. При этом использовалась аппроксимация ГБФ (21). На рис. 6, *a* приведена поверхность, описываемая уравнением (26), а на рис. 6, *б* — поверхность, восстановленная нейронной сетью. Рис. 6, *в* отражает результаты управления объектом (26) при задании в качестве желаемого управляющего сигнала

$$y^*[k] = 2 + 0.5 \sin[\pi k / 200] - 0.25 \cos[\pi k / 100]. \quad (27)$$

Эксперимент 3. На рис. 7 приведены результаты идентификации и управления нестационарным объектом, на рис. 7, *a*, *б* — результаты построения одномерного нелинейного нестационарного объекта, описываемого уравнениями

$$y(k+1) = \begin{cases} \frac{u(k)[\sin(y(k)) + 1.5]}{1 + y^2(k)}, & k = 1 \div 1250, \\ \frac{u(k)[\cos(y(k)) + 0.5]}{2 + y^2(k)}, & k = 1251 \div 2500, \end{cases} \quad (28)$$

и управления им. В качестве требуемого выходного сигнала задавался

$$y^*[k] = 0.5 \sin[\pi k / 200] - 0.25 \cos[\pi k / 100].$$

Первые 500 шагов используются для идентификации объекта с помощью алгоритма (17). После достижения заданной точности модели сетью с числом нейронов, равным 17, управление осуществляется по алгоритму (18). При этом на каждом шаге проводилась идентификация и управление.

Из рис. 7, *в* видно, что после изменения параметров объекта на 1251-м шаге, что приводит к появлению большой ошибки идентификации $e[k]$, корректируются параметры модели в течение примерно 40 шагов, а затем реализуется управление.

Моделирование для БФ других видов привело к результатам, аналогичным приведенным в табл. 1.

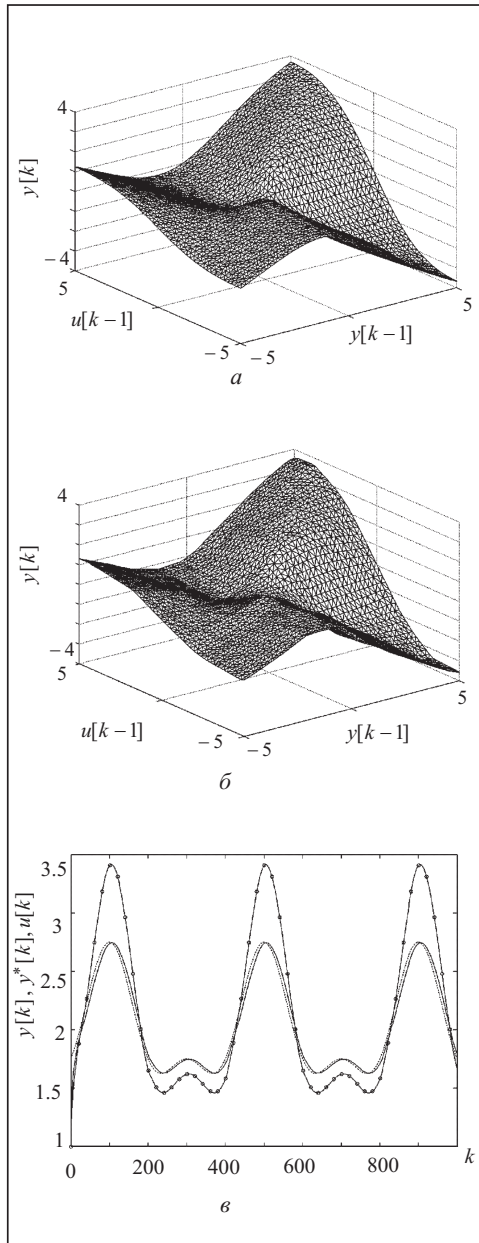


Рис. 6

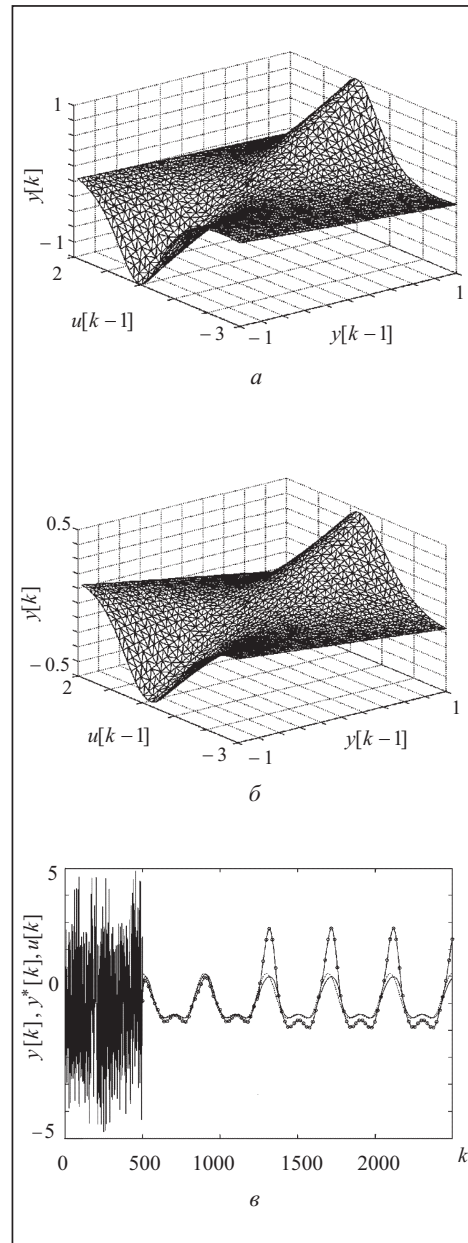


Рис. 7

Заключение. Как следует из результатов моделирования, рассмотренный подход к синтезу нейросетевого управления нелинейными нестационарными динамическими объектами, использующий кусочно-линейную аппроксимацию ГБФ в РБС, позволяет существенно упростить вычисления, сопутствующие процессам построения модели исследуемого объекта и управления им.

Переход от РБС нулевого порядка к РБС первого порядка обеспечивает повышение точности получаемой модели, однако при этом возрастает длительность процесса обучения вследствие необходимости настройки дополнительной матрицы весовых коэффициентов. Кроме того, усложняется и сам алгоритм управления.

Нормализация БФ в ряде случаев также позволяет повысить точность модели, однако неудобна при решении задач в реальном времени, поскольку введение новой БФ требует пересчета всех параметров сети.

Кусочно-линейная аппроксимация может использоваться при выборе БФ другого вида, например, мультикватричной, обратной мультикватричной, сплайн-функции и т.д.

СПИСОК ЛИТЕРАТУРЫ

1. Омату С., Хализ М., Юсоф Р. Нейроуправление и его приложения. — М.: ИПРЖР, 2000. — 272 с.
2. Narendra K. S., Parthasarathy K. Identification and control of dynamical systems using neural networks // IEEE Trans. on Neural Networks. — 1990. — 1, N 1. — P. 4–26.
3. Moody J., Darken C. Fast learning in networks of locally-tuned processing units // Neural Comput. — 1989. — 1. — P. 281–294.
4. Spooner J. T., Passino K. M. Decentralized adaptive control of nonlinear systems using radial basis neural networks // IEEE Trans. on Automat. Contr. — 1999. — 44, N 11. — P. 2050–2057.
5. Yu D. L., Yu D. W. A new structure adaptation algorithm for RBF networks and its application // Neural Comput. & Appl. — 2007. — 16. — P. 91–100.
6. Shilling R. J., Carroll J. J., Al-Ajlouni A. F. Approximation of nonlinear systems with radial basis function neural networks // IEEE Trans. on Neural Networks. — 2001. — 12, N 6. — P. 1–15.
7. Powell M. J. D. Radial basis functions for multivariable interpolation: A review // Mason J. C., Cox M. G. (Eds). Algorithm for approximation. — Oxford: Oxford Un-t Press, 1985. — P. 143–167.
8. Huang G.-B., Saratchandran P., Sundarajan N. An efficient sequential learning algorithm for growing and pruning RBF (GAP-RBF) networks // IEEE Trans. on Systems, Man, and Cybernetics. — 2004. — 34, N 6. — P. 2284–2292.
9. Руденко О. Г., Бессонов А. А. Идентификация нелинейных нестационарных объектов в реальном времени с помощью радиально-базисных сетей // Кибернетика и системный анализ. — 2003. — № 6. — С. 177–185.
10. Webb A., Shannon S. Shape-adaptive radial based functions // IEEE Trans. on Neural Networks. — 1998. — 9, N 6. — P. 1155–1166.
11. Руденко О. Г., Бессонов А. А. Адаптивное управление многомерными нелинейными объектами на основе радиально-базисных сетей // Кибернетика и системный анализ. — 2005. — № 2. — С. 168–176.
12. Хайкин С. Нейронные сети: полный курс. — М.: Изд. дом «Вильямс», 2006. — 1104 с.

Поступила 02.02.2009