

## РОБАСТНАЯ ИДЕНТИФИКАЦИЯ НЕЛИНЕЙНЫХ ОБЪЕКТОВ С ПОМОЩЬЮ ЭВОЛЮЦИОНИРУЮЩЕЙ РАДИАЛЬНО-БАЗИСНОЙ СЕТИ

**Ключевые слова:** *нейронная сеть, обучение, идентификация, эволюционный алгоритм, робастность.*

### ВВЕДЕНИЕ

Задача получения математических моделей реальных объектов, адекватно отображающих их свойства, не только представляет самостоятельный интерес, но и является неотъемлемой частью задачи оптимизации функционирования данных объектов (управление, предсказание их поведения и т.д.). Основные трудности качественного решения задачи идентификации обусловлены нелинейностью и нестационарностью характеристик исследуемых объектов, наличием различного рода помех, отсутствием достаточно полной априорной информации как о самих объектах, так и об условиях их функционирования. Если теория идентификации линейных стационарных объектов разработана достаточно полно, то идентификация нелинейных объектов осуществляется наиболее часто субъективно и использует в основном аппроксимацию нелинейностей различными рядами (Вольтерры, Гаммерштейна, Винера и т.п.) либо полиномами. Однако эти классические модели являются непараметрическими, что существенно затрудняет решение задачи идентификации.

Трудности, связанные с идентификацией нелинейных динамических объектов традиционными методами, привели к появлению и развитию альтернативного, нейросетевого, подхода к решению этой задачи. Поскольку с математической точки зрения задача идентификации представляет собой задачу аппроксимации (или восстановления) некоторой в общем виде сложной нелинейной функции, для ее решения используют искусственные нейронные сети (ИНС), образуемые нейронами с нелинейными функциями активации и являющиеся хорошими аппроксиматорами.

Следует отметить, что при исследовании нелинейных объектов с помощью ИНС в фундаментальной роли выступают объекты вида NARMAX (Nonlinear Auto-Regressive Moving Average with eXogeneous inputs) или NARX (Nonlinear Auto-Regressive eXogeneous with inputs) моделей, имеющих соответственно вид [1–3]

$$y(k) = f[y(k-1), \dots, y(k-K_y), u(k-1), \dots, u(k-K_u), \xi(k-1), \dots, \xi(k-K_\xi)] + \xi(k), \quad (1)$$

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

где  $y(i)$ ,  $u(i)$  — выходной и входной сигналы соответственно;  $K_y$ ,  $K_u$ ,  $K_\xi$  — порядки запаздывания по выходным, входным сигналам объекта и помехе соответственно;  $f[\cdot]$  — нелинейная функция;  $\xi(k)$  — помеха.

Для моделей (1), (2) задача идентификации состоит в получении оценки функции  $f[\cdot]$  по измерениям входных и выходных переменных.

По аналогии с традиционным подходом к решению задачи идентификации, при котором процесс построения модели подразделяется на два этапа: структурную и

параметрическую идентификацию, применение ИНС также требует решение двух задач: определения структуры сети и настройки ее параметров путем обучения.

Простота структуры и наличие большого количества алгоритмов обучения радиально-базисных сетей (РБС) обеспечили им широкое применение при идентификации нелинейных динамических объектов [4–9].

#### СТРУКТУРА РБС

РБС имеет двухслойную структуру. Скрытый слой состоит из нейронов, каждый из которых вычисляет некоторое расстояние между его центром  $\mu$  и входным вектором сети  $\mathbf{x}(k)$ . Ниже используется NARX модель, для которой

$$\mathbf{x}(k) = [y(k-1), y(k-2), \dots, y(k-K_y), u(k-1), u(k-2), \dots, u(k-K_u)]^T.$$

Затем каждый нейрон скрытого слоя преобразует результат с помощью определенной нелинейной базисной функции (БФ)  $\Phi_i(\mathbf{x}(k), \boldsymbol{\mu}, \sigma) = f(\|\mathbf{x} - \boldsymbol{\mu}\|, \sigma)$  (здесь  $\sigma$  — радиус БФ).

Модель, представленная радиально-базисной сетью, имеет вид

$$\hat{y}[k] = a_0 + \sum_{i=1}^N w_i \Phi_i(\mathbf{x}(k), \boldsymbol{\mu}, \sigma), \quad (3)$$

где  $a_0$  — смещение нейрона выходного слоя,  $w_i$  — вес связи  $i$ -го нейрона скрытого слоя с нейроном выходного слоя,  $N$  — число нейронов в скрытом слое.

Наиболее часто в качестве БФ выбираются функции, приведенные в табл. 1.

**Таблица 1**

Номер функции	Наименование функции	Вид функции
1	Гауссовская	$\Phi(x) = \exp\left\{-\frac{(x-\mu)^2}{\sigma^2}\right\}$
2	«Мексиканская шляпа»	$\Phi(x) = \left(1 - \frac{(x-\mu)^2}{\sigma^2}\right) e^{-\frac{(x-\mu)^2}{\sigma^2}}$
3	Лапласа	$\Phi(x) = \exp\left\{-\frac{ x-\mu }{\sigma}\right\}$
4	Релея	$\Phi(x) = \frac{2(x-\mu)}{\sigma} \exp\left\{-\frac{(x-\mu)^2}{\sigma^2}\right\}$
5	Обобщенная гауссовская	$\Phi(\mathbf{x}) = e^{-(\mathbf{x}-\boldsymbol{\mu})^T R^{-1}(\mathbf{x}-\boldsymbol{\mu})}$ , где $R^{-1} = [r_{ij}^k]$ , $i, j = \overline{1, L}$ , $k = \overline{1, N}$ , — масштабирующая матрица ( $L$ — размерность входного сигнала, $N$ — количество нейронов)
6	Косинусоидальная	$\Phi(x) = \cos\left\{\frac{2\pi(x-\mu)}{\sigma}\right\}$
7	Параболическая	$\Phi(x) = 1 - \frac{(x-\mu)^2}{\sigma^2}$

Вопрос выбора структуры РБС (количества и вида базисных функций отдельных нейронов) является чрезвычайно важным, поскольку от этого зависит и точность, и сложность решения поставленной задачи. В настоящее время не существует единой или хотя бы достаточно эффективной методологии определения структуры сетей, поэтому выбор топологии сети является эмпирическим,

основанным на опыте проектировщика. Если в работах [10, 11] рассматривалось последовательное усложнение структуры РБС путем введения нового нейрона на основании определенного критерия, то впоследствии во многих публикациях (см., например, библиографию к [12–14]) для определения структуры сети использовался эволюционный подход на основе генетического алгоритма (ГА).

#### ТРАДИЦИОННОЕ ОБУЧЕНИЕ РБС

Обучение РБС с базовыми функциями, приведенными в табл. 1, заключается в определении вектора параметров скрытого слоя  $\theta = (a_0, \mathbf{w}, \boldsymbol{\mu}, \sigma)^T$  на основании предъявления сети обучающих пар  $\{\mathbf{x}(k), y(k)\}$ ,  $k = 1, 2, \dots$

При выборе критерия

$$F_k(\theta) = \frac{1}{k} \sum_{i=1}^k \rho(e(i, \theta)), \quad (4)$$

где  $\rho(e(i, \theta))$  — некоторая функция потерь;  $e(i, \theta) = y(i) - \hat{y}(i, \theta)$  — ошибка, обучение сводится к поиску оценки  $\hat{\theta}_k = \arg \min_{\theta} F_k(\theta)$ , определяемой как решение системы уравнений

$$\nabla F(\theta_j) = \frac{\partial F(\theta)}{\partial \theta_j} = \sum_{i=1}^k \psi(e(i, \theta)) \frac{\partial e(i, \theta)}{\partial \theta_j} = 0. \quad (5)$$

Здесь  $\psi(e(i, \theta)) = \frac{\partial \rho(e(i, \theta))}{\partial e(i, \theta)}$  — функция влияния.

Большинство используемых в настоящее время алгоритмов обучения основано на гипотезе принадлежности к нормальному закону распределения помех  $\xi$ , они представляют собой различные модификации метода наименьших квадратов, минимизирующего квадратичную функцию потерь  $\rho(e(i, \theta))$  и обеспечивающего в этих условиях решение, асимптотически оптимальное с минимальной дисперсией в классе несмещенных оценок.

Если распределение помех отлично от нормального, имеет выбросы или длинные «хвосты», то МНК-оценка оказывается неустойчивой, что и явилось предпосылкой для развития альтернативного, робастного оценивания в статистике с целью исключения влияния больших ошибок.

Среди основных типов робастных оценок, M-, L-, и R-оценок, являющихся соответственно оценками максимального правдоподобия, линейными комбинациями порядковых статистик и оценками, получаемыми в ранговых критериях, в задачах обучения наиболее часто используется предложенная Хьюбером M-оценка [15].

M-оценка также представляет собой оценку  $\hat{\theta}$ , определяемую как решение экстремальной задачи (4) или как решение системы уравнений (5), однако функция потерь  $\rho(e_i)$  выбирается отличной от квадратичной. Исследование различных классов распределений помех позволило получить для этих классов наименее благоприятные, т.е. минимизирующие информацию Фишера, распределения, использование которых, в свою очередь, определяет вид функции потерь и обеспечивает получение робастных оценок, являющихся работоспособными практически для любых распределений помех.

Классические робастные методы ориентированы на симметричность засорения, когда выбросы одинаково часто появляются как в интервале отрицательных, так и в интервале положительных значений. Робастные алгоритмы обучения РБС рассматривались в работах [17–22].

Указанные методы позволяют эффективно бороться с помехами, описываемыми моделью Тьюки–Хьюбера [15, 16],

$$\rho(\xi) = (1 - \varepsilon)\rho_0(\xi) + \varepsilon q(\xi), \quad (6)$$

где  $\rho_0(\xi)$  — плотность соответствующего основного распределения;  $q(\xi)$  — плотность засоряющего распределения;  $\varepsilon \in [0, 1]$  — параметр, характеризующий степень засорения основного распределения. В этом случае основное и засоряющее распределения являются гауссовскими с нулевыми математическими ожиданиями и дисперсиями  $\sigma_1^2$  и  $\sigma_2^2$ ,  $\sigma_1^2 \ll \sigma_2^2$ .

В более общей ситуации произвольного вида засорения (например, когда гауссовское засоряющее распределение имеет ненулевое математическое ожидание или засоряющее распределение является несимметричным (Релея, логнормальное, Гамма-распределение, Вейбулла–Гнеденко и др.)), оценки, полученные на основании этих методов, будут смещенными. Необходимость учета асимметрии распределений обуславливает целесообразность выбора асимметричных функционалов [23].

Таким образом, если в настоящее время разработано и хорошо изучено достаточно большое количество алгоритмов настройки параметров сети, то вопросы выбора ее оптимальной структуры остаются открытыми.

### НЕЙРОЭВОЛЮЦИОННЫЙ АЛГОРИТМ ОБУЧЕНИЯ РБС

Для алгоритмов, использующих модели механизмов естественной эволюции, обычно применяют обобщенное название — эволюционные алгоритмы. Существует множество разновидностей подобного рода алгоритмов, отличающихся способом использования определенных механизмов, а также представлением особей. Одним из наиболее распространенных видов эволюционных алгоритмов являются генетические алгоритмы, предложенные Дж. Холландом.

В ГА каждая особь кодируется сходным с ДНК методом — в виде строки (хромосомы), содержащей определенный набор генов. Длина хромосомы постоянна. Популяция, состоящая из некоторого количества особей, подвергается процессу эволюции с использованием операций скрещивания и мутаций.

Классический ГА содержит следующие шаги.

1. Создание начальной популяции.
  - 1.1. Инициализация хромосомы каждой особи.
  - 1.2. Оценивание начальной популяции.
2. Этап эволюции — построение нового поколения.
  - 2.1. Отбор кандидатов на скрещивание (селекция).
  - 2.2. Скрещивание, т.е. порождение каждой парой отобранных кандидатов новых индивидов.
  - 2.3. Мутация.
  - 2.4. Оценивание новой популяции.
3. Проверка критерия завершения; если не выполнено — переход к п. 2.

В начале работы нейроэволюционного алгоритма случайным образом инициализируется популяция  $P_0$ , состоящая из  $S$  особей (РБС сетей),  $P_0 = \{H_1, H_2, \dots, H_S\}$ . При этом каждая особь в популяции 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,  $w_{\max}$  — максимальное допустимые значения соответственно). Формат хромосомы и соответствие между генами и параметрами РБС представлен на рис. 1. Здесь  $c_1$  и  $c_2$  — матожидание соответственно основной и

Общие параметры сети					$i$ -й нейрон						
$s_1$	$s_2$	$c_1$	$c_2$	$a_0$	...	1/0	BF	$w_i$	$\mu_i$	$\sigma_i$	...

засоряющей популяции;  $s_1$  и  $s_2$  — соответственно дисперсия основной и

Рис. 1

засоряющей помех. Следует отметить, что длина хромосомы постоянна и ограничивается максимально допустимым количеством нейронов.

Как видим из рисунка, каждая хромосома состоит из генов, в которых хранится информация о соответствующих параметрах сети. Первые гены хромосомы хранят информацию о параметрах помехи и являются активными лишь в случае идентификации зашумленных объектов. Следующий ген кодирует информацию о параметре  $a_0$  — смещении нейрона выходного слоя сети. Затем следуют блоки генов, кодирующие параметры соответствующих нейронов скрытого слоя. Первый ген каждого такого блока (1/0) определяет присутствие соответствующего нейрона в структуре сети, т.е. участие (или неучастие) его в вычислении выходной реакции сети на поступивший входной сигнал. Ген BF определяет вид базисной функции из заданного набора БФ, которая используется для вычисления реакции нейрона.

Далее в хромосоме следует группа генов, кодирующая непосредственно параметры соответствующего нейрона, а именно:  $w$  — его весовой параметр,  $\mu$  — центр базисной функции,  $\sigma$  — радиус базисной функции. 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.

#### ОЦЕНИВАНИЕ ПОПУЛЯЦИИ

После формирования начальной популяции оценивается приспособленность каждой особи, входящей в популяцию, исходя из анализа функции приспособленности (фитнесс-функция). В качестве такой функции в случае обучения off-line и при наличии полной выборки входных-выходных сигналов объекта имеем

$$f_i(x_j) = \frac{1}{M} \sum_{j=1}^M |y_j^*(x_j) - \hat{y}_j(x_j)|, \quad (7)$$

где  $y^*(k)$  — желаемая реакция сети;  $\hat{y}(k)$  — реальный выходной сигнал;  $M$  — размер выборки.

Для упрощения дальнейших операций сортировки популяции обычно функция приспособленности нормализуется как

$$f_i^N(x_j) = \frac{f_i(x_j)}{\sum_{j=1}^N f_j(x_j)}. \quad (8)$$

Следует отметить, что для обеспечения робастности получаемого решения в качестве фитнес-функции может быть использована неквадратичная функция потерь  $\rho(i, \theta)$ , применяемая для М-обучения РБС. В этом случае нет необходи-

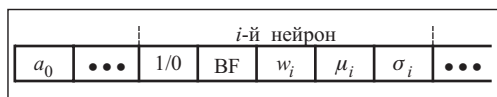


Рис. 2

мости оценивать параметры помехи для устранения смещения получаемого решения, а структура хромосомы принимает вид, показанный на рис. 2.

Таким образом, чтобы определить приспособленность сети, необходимо выполнить ее симуляцию на всей выборке и затем сравнить реакции сети с реальным выходным сигналом объекта.

## СЕЛЕКЦИЯ

После вычисления для каждой особи (сети) ее функции приспособленности в популяции необходимо провести отбор особей, хромосомы которых будут принимать участие в формировании нового поколения. Для этого проводится вычисление среднего значения фитнес-функции  $f_{av}$  популяции как среднего арифметического значения фитнес-функций всех входящих в нее особей:

$$f_{av} = \frac{1}{N} \sum_{j=1}^N f_j.$$

Затем для каждой особи вычисляется следующее соотношение:

$$P_s(j) = \frac{f_j}{f_{av}},$$

и в зависимости от значения  $P_s(j)$  формируется массив особей, хромосомы которых будут участвовать в процессе скрещивания.

Возможен также вариант селекции с использованием порогового значения функции приспособленности. При этом проводится сортировка особей по убыванию нормированного значения функции приспособленности (8), задается порог  $\theta \in [0, 1]$ . Для участия в процедуре скрещивания отбираются лишь те особи, для которых выполняется условие  $f_j \leq \theta$ .

## СКРЕЩИВАНИЕ

После отбора родительских особей методом селекции осуществляется их скрещивание (кроссовер). Кроссовер применяется с целью воспроизведения потомства для обмена генетической информацией между родительскими особями. Пусть родительские особи описываются выражениями

$$H^{(1)} = \{h_1^{(1)}, \dots, h_i^{(1)}, \dots, h_L^{(1)}\};$$
$$H^{(2)} = \{h_1^{(2)}, \dots, h_i^{(2)}, \dots, h_L^{(2)}\},$$

а их потомки описываются как

$$Y^{(1)} = \{y_1^{(1)}, \dots, y_i^{(1)}, \dots, y_L^{(1)}\};$$
$$Y^{(2)} = \{y_1^{(2)}, \dots, y_i^{(2)}, \dots, y_L^{(2)}\}.$$

Количество родителей и потомков зависит от выбора оператора кроссовера и может меняться. В настоящее время существует достаточно большое число различных операторов скрещивания: точечное, однородное, равномерное, сравнительное, диагональное, нечеткое и т.д. Наиболее простым оператором, который был использован в настоящей статье, является оператор одноточечного скрещивания, при котором две родительские особи:  $H^{(1)}$  и  $H^{(2)}$ , формируют хромосомы двух потомков следующим образом:

$$Y^{(1)} = \{h_1^{(1)}, \dots, h_i^{(1)}, h_{i+1}^{(2)}, \dots, h_L^{(2)}\},$$
$$Y^{(2)} = \{h_1^{(2)}, \dots, h_i^{(2)}, h_{i+1}^{(1)}, \dots, h_L^{(1)}\},$$

где  $i$  — случайная величина в интервале  $[1, L]$ .

## МУТАЦИЯ

Мутация позволяет создать новый генетический материал в популяции для сохранения ее разнообразия. Мутация — это не что иное как изменение случайной части хромосомы, представляющей отдельную особь. Количество мутаций в популяции регулируется параметром  $p_m$ , который определяет вероятность мутации. Таким образом, только  $p_m \times N$  случайных хромосом в популяции могут мутировать.

Оператор мутации осуществляет возможные мутации на определенных генах некоторой хромосомы. Если хромосома перед мутацией имела вид

$$H_j = \{h_{1j}, \dots, h_{ij}, \dots, h_{Lj}\},$$

где  $h_{ij}$  — ген, в котором должна произойти мутация, то после мутации ее можно записать следующим образом:

$$H'_j = \{h_{1j}, \dots, h'_{ij}, \dots, h_{Lj}\}.$$

В данной статье использовалась неравномерная мутация, при которой ген  $h'_{ij}$  создается из гена  $h_{ij} \in [h_{\min}, h_{\max}]$ , где  $h_{\min}$  и  $h_{\max}$  — минимально и максимально допустимые значения для данного гена, следующим образом:

$$h'_{ij} = \begin{cases} h_{ij} + \Delta(k, h_{\max} - h_{ij}), & \text{если } \tau = 0; \\ h_{ij} - \Delta(k, h_{ij} - h_{\min}), & \text{если } \tau = 1, \end{cases}$$

здесь  $\tau$  — случайная бинарная величина с равномерным распределением. Значение функции  $\Delta$  вычисляется формулой

$$\Delta(k, y) = y(1 - \alpha \left(1 - \frac{k}{T}\right)^b),$$

где  $\alpha$  — случайная величина, равномерно распределенная в интервале  $[0, 1]$ ,  $T$  — максимальное число итераций алгоритма,  $k$  — текущая итерация,  $b$  — параметр, определяющий степень неравномерности распределения.

Оператор  $\Delta(k, y)$  может принимать значения в диапазоне  $[0, y]$ , причем вероятность того, что это значение стремится к нулю, возрастает с увеличением  $k$ . Таким образом, на начальной стадии работы ГА неравномерная мутация позволяет существенно изменять значение мутирующего гена, а на более поздних стадиях осуществляются лишь небольшие уточняющие мутации, которые позволяют увеличить точность уже полученного решения.

Следует отметить, что метод неравномерной мутации применяется лишь для генов, кодирующих параметры сети. Для генов, отвечающих за активацию нейронов и вид базисной функции, используется случайная замена:

$$h_{ij} = \text{rand}[h_{\min}, h_{\max}],$$

где  $\text{rand}[x, y]$  — случайное целое число в интервале  $[x, y]$ , распределенное по равномерному закону. Так, для гена, отвечающего за активацию нейрона,  $x = 0, y = 1$ , а для гена, определяющего вид базисной функции,  $x = 1, y = P$ , где  $P$  — количество используемых базисных функций.

Следует отметить, что, с одной стороны, мутации могут привести к ухудшению приспособленности данной особи, а с другой — они являются единственным способом внесения новой информации в ее хромосому.



**МОДЕЛИРОВАНИЕ**

**Эксперимент 1.** Проводилась идентификация нелинейного стационарного объекта

$$y(k) = 0,725 \sin \left( \frac{16u(k-1) + 8y(k-1)}{3 + 4u^2(k-1) + 4y^2(k-1)} \right) + 0,2u(k-1) + 0,2y(k-1) + \xi(k) \quad (9)$$

при наличии помехи  $\xi(k)$ , описываемой моделью

$$\xi(k) = (1 - \varepsilon)q_1(k) + \varepsilon q_2(k), \quad (10)$$

где  $\varepsilon = 0,1$ ;  $q_1(k)$ ,  $q_2(k)$  — нормально распределенные помехи с математическими ожиданиями  $m_1 = m_2 = 0$  и дисперсиями  $\sigma_1 = 0,6$ ;  $\sigma_2 = 12$  соответственно.

Для коррекции результатов, связанной с необходимостью устранения смещения, вызванного действием помехи (10), в [21–23] параметры помех оценивались с помощью рекуррентных алгоритмов. В данном эксперименте для оценки параметров модели помехи (9) в хромосоме, кодирующей структуру РБС, были использованы четыре дополнительных гена (см. рис. 1), в которых хранятся параметры  $s_1$ ,  $s_2$  (оценки  $\sigma_1$  и  $\sigma_2$ ),  $c_1$ ,  $c_2$  (оценки  $m_1$  и  $m_2$ ). Функция приспособленности (7) также была модифицирована следующим образом:

$$f_i = \begin{cases} \frac{1}{M} \sum_{j=1}^M \left| \frac{y_j^*(x_j) - \hat{y}_j(x_j) - c_1}{\sigma_1^2} \right|, & \text{если } y_j^*(x_j) - \hat{y}_j(x_j) - c_1 \leq 3\sigma_1; \\ \frac{1}{M} \sum_{j=1}^M \left| \frac{y_j^*(x_j) - \hat{y}_j(x_j) - c_2}{\sigma_2^2} \right| & \text{в противном случае.} \end{cases} \quad (11)$$

Результаты идентификации объекта (9), зашумленного помехой (10), представлены на рис. 3. Здесь показаны гистограмма помехи (рис. 3, а) и восстановленная поверхность (рис. 3, б). После 2000 эпох обучения были получены следующие оценки параметров помех:  $\sigma_1 = 0,5860$ ;  $\sigma_2 = 12,5818$ ;  $c_1 = 0,0333$ ;  $c_2 = -0,1646$ . Сеть состояла из 13 нейронов (семь нейронов с БФ вида (5) и шесть нейронов с БФ вида (6)).

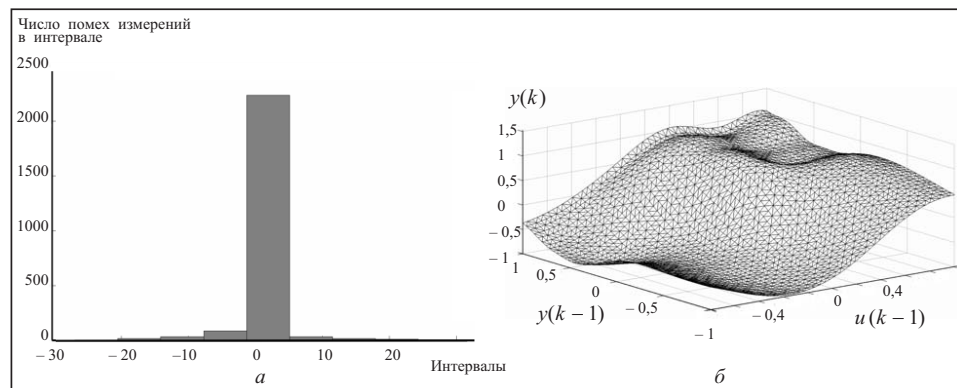


Рис. 3

**Эксперимент 2.** Рассматривалась задача идентификации объекта (9) при наличии помехи  $\xi(k)$ , имеющей равномерное распределение и имеющей распределение Релея (Ray(1,6)). На рис. 4, а приведены гистограммы помехи, а на рис. 4, б — виды восстановленных поверхностей.

Предполагая, что реальное распределение помехи неизвестно, в данном эксперименте использовалась применяемая в [21, 22] аппроксимация распределений



помех моделью Тьюки–Хьюбера (10), в которой и основное, и засоряющее распределения являются нормальными с оценками  $m_1, m_2 \neq 0$ . В результате обучения РБС были получены следующие параметры засоренной помехи для равномерного распределения:  $\sigma_1 = 12,60$ ;  $\sigma_2 = 1,83$ ;  $c_1 = 0$ ;  $c_2 = -0,17$ ; для распределения Релея:  $\sigma_1 = 0,9981$ ;  $\sigma_2 = 0,8555$ ;  $m_1 = 1,9702$ ;  $m_2 = 5,2842$ , которые использовались для коррекции выходных сигналов сети. При этом сеть содержала 14 нейронов (пять нейронов с БФ вида (5) и девять нейронов с БФ вида (6)).

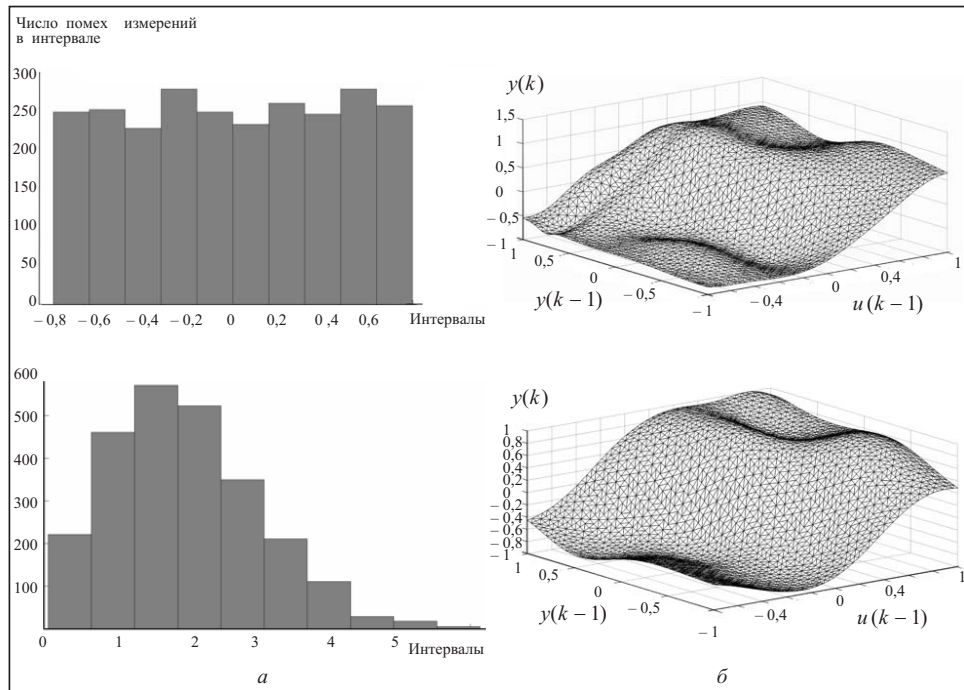


Рис. 4

**Эксперимент 3.** Решалась задача идентификации многомерного объекта (MIMO), который описывался следующими уравнениями:

$$y_1(k) = \frac{15u_1(k-1)y_2(k-1)}{2 + 50[u_1(k-1)]^2} + 0,5u_1(k-1) - 0,25y_2(k-1) + 0,1 + \xi_1(k);$$

$$y_2(k) = \frac{\sin(\pi u_2(k-1))y_1(k-1) + 2u_2(k-1)}{3} + \xi_2(k), \quad (12)$$

где  $u_1, u_2$  — входные сигналы;  $y_1$  и  $y_2$  — выходные сигналы;  $\xi_1$  и  $\xi_2$  — помехи измерений.

Помеха  $\xi_1$  описывалась моделью (10) и имела те же параметры, что и в эксперименте 1. Помеха  $\xi_2$  имела распределение Релея (Ray(1,6)). Таким образом, выходные сигналы объекта были зашумлены различными помехами. Кроме того, поскольку данный объект является многосвязным, окончательный вид распределений установить весьма сложно. После 2000 эпох обучения были получены следующие оценки параметров помех (независимые по каждому выходу объекта):  $\sigma_1^1 = 1,2042$ ;  $\sigma_2^1 = 7,4612$ ;  $m_1^1 = 0,0$ ;  $m_2^1 = 0,6765$  — для первого выхода;  $\sigma_1^2 = 2,5593$ ;  $\sigma_2^2 = 1,5492$ ;  $m_1^2 = 0,0$ ;  $m_2^2 = 2,0294$  — для второго выхода.

Сеть содержала 17 нейронов (девять нейронов с БФ № 1 (см. табл.1) и восемь нейронов с БФ № 2). Результаты моделирования приведены на рис. 5. Эталонные поверхности, описываемые выражениями (12), представлены на рис. 5, а; восстановленные поверхности — на рис. 5, б; график изменения значения фитнес-функции особи-победителя — на рис. 5, в; график изменения количества активных нейронов особи-победителя — на рис. 5, г.

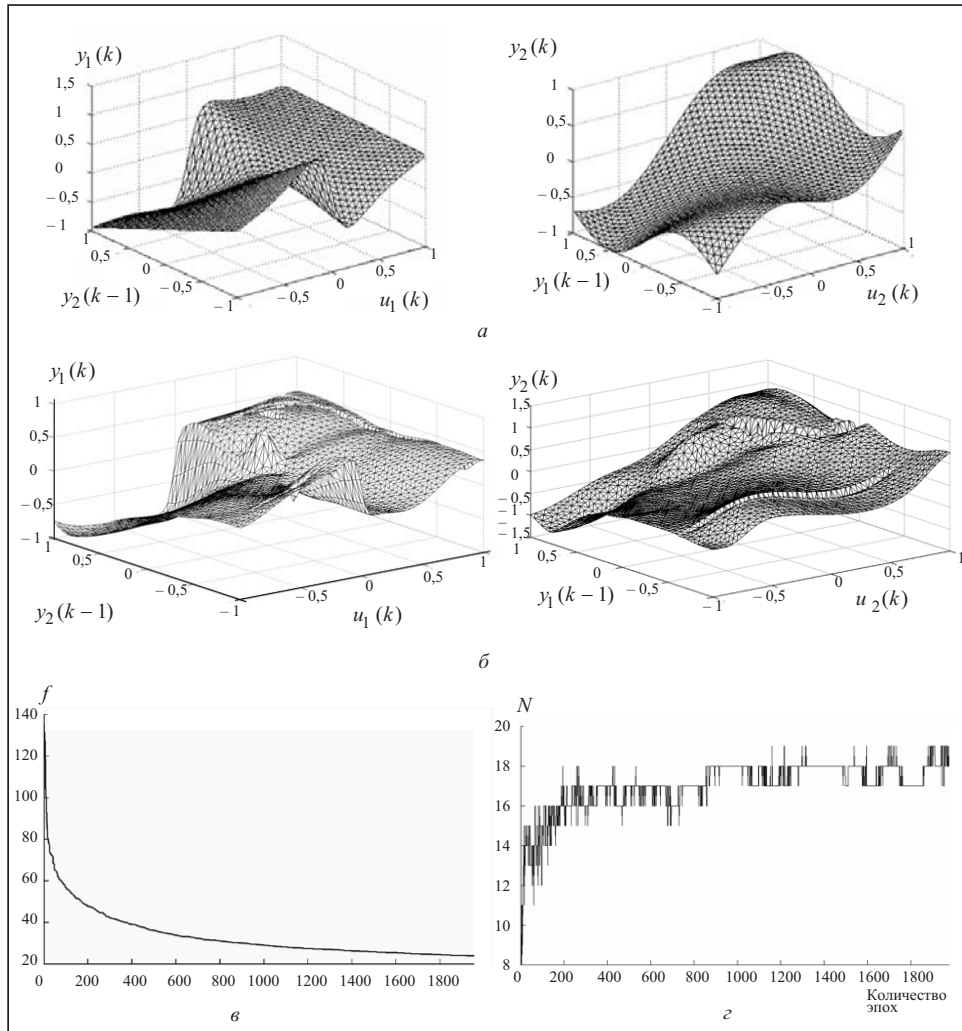


Рис. 5

**Эксперимент 4.** Выполнялась идентификация объекта (9) с параметрами помехи такими же, как и в эксперименте 1, и с использованием фитнес-функции вида (7), где в качестве функционала  $\rho$  использовалось выражение

$$\rho[e(k)] = \frac{e^2(k)}{1 + e^2(k)}. \quad (13)$$

На рис. 6, а показан график изменения значения фитнес-функции (7) для особи-победителя, на рис. 6, б — график изменения количества активных нейронов. Восстановленная поверхность приведена на рис. 6, в. Результаты моделирования свидетельствуют о том, что использование фитнес-функции (7) с функционалом (13) позволяет эффективно устранять помеху, используя более простой формат хромосомы.

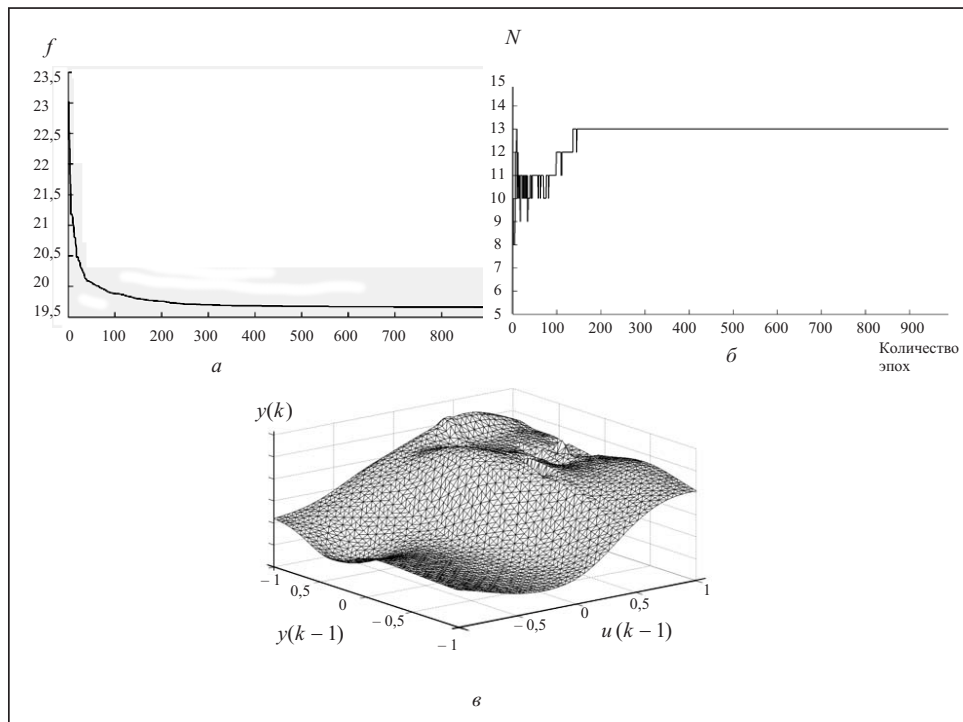


Рис. 6

#### ЗАКЛЮЧЕНИЕ

Как видно из результатов моделирования, эволюционирующая РБС, использующая ГА как для определения структуры нейросетевой модели, так и оценивания ее параметров, обладает высокой степенью робастности и способна решать задачу идентификации сильно зашумленных объектов. При этом возможны два подхода к устранению влияния помехи. Первый базируется на использовании модели Тьюки–Хьюбера и заключается в оценивании параметров помехи, второй подход основан на использовании М-обучения и позволяет несколько упростить структуру хромосомы, так как не требует хранения дополнительных параметров. Результаты моделирования демонстрируют эффективность обоих подходов.

#### СПИСОК ЛИТЕРАТУРЫ

1. Leontaritis I.J., Billings S.A. Input-output parametric models for non-linear systems. Part I: Deterministic non-linear systems // *Inf. J. of Control*. — 1985. — **41**. — P. 303–308.
2. Leontaritis I.J., Billings S.A. Input-output parametric models for non-linear systems. Part II: Stochastic non-linear systems // *Int. J. of Control*. — 1985. — **41**. — P. 309–344.
3. Chen S., Billings S.A. Representations of nonlinear systems: the NARMAX model // *Int. J. of Control*. — 1983. — **49**(3). — P. 1013–1032.
4. 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.
5. Chen S., Billings S.A., Grant P.M. Recursive hybrid algorithm for non-linear system identification using radial basis function networks // *Int. J. of Control*. — 1992. — **55**. — P. 1051–1070.
6. Хайкин С. Нейронные сети: полный курс. — М.: Изд. дом «Вильямс», 2006. — 1104 с.

7. Spooner J.T., Passino K.M. Decentralized adaptive control of nonlinear systems using radial basis neural networks // IEEE Trans. on Automatic Control. — 1999. — **44**, N 11. — P. 2050–2057.
8. 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.
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. Li Y., Sundararajan N., Saratchandran P. Analysis of minimal radial basis function network algorithm for real-time identification of nonlinear dynamic systems // IEEE Proc. — Control Theory Appl. — 2000. — **147**, N 4. — P. 476–484.
11. 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.
12. Maillard E.P., Gueriot D. RBF neural network, basis functions and genetic algorithm // Proc. Int. Conf. Neural Networks, Houston, TX. — 1997. — **4**. — P. 2187–2192.
13. Ding S., Xu L., Zhu H. Studies on optimization algorithms for some artificial neural networks based on genetic algorithm (GA) // J. Computers. — 2011. — **6**, N 5. — P. 939–946.
14. Buchtala O., Klimek M., Sick B. Evolutionary optimization of radial basis function classifiers for data mining applications // IEEE Trans. on Systems, Man, and Cybernetics, Part B. — 2005. — **35**, N 5. — P. 928–947.
15. Хьюбер П. Робастность в статистике. — М.: Мир, 1984. — 304 с.
16. Hampel F.R., Ronchetti E.M., Rousseeuw P.J., Stahel W.A. Robust statistics. The approach based on influence functions. — N.Y.: John Wiley and Sons, 1986. — 526 p.
17. Chen D.S., Jain R.C. A robust back-propagation learning algorithm for function approximation // IEEE Tran. on Neural Networks. — 1994. — **5**. — P. 467–479.
18. Liano K. A robust approach to supervised learning in neural network // Proc. ICNN. — 1994. — **1**. — P. 513–516.
19. Lee Ch.-Ch., Chung P.-Ch., Tsai J.-R., Chang Ch.-I. Robust radial basis function neural networks. Part B: Cybernetics // IEEE Trans. on Systems, Man, and Cybernetics. — 1999. — **29**, N 6. — P. 674–685.
20. Руденко О.Г., Бессонов А.А. Робастное обучение вэйвлет-нейросетей // Проблемы управления и информатики. — 2010. — № 5. — С. 66–79.
21. Руденко О.Г., Бессонов А.А. Робастное обучение радиально-базисных сетей // Кибернетика и системный анализ. — 2011. — № 6. — С. 38–46.
22. Rudenko O., Bezsonov O. Function approximation using robust radial basis function networks // J. of Intelligent Learning Systems and Appl. — 2011. — **3**. — P. 17–25.
23. Руденко О.Г., Бессонов А.А. М-обучение радиально-базисных сетей с использованием асимметричных функций влияния // Проблемы управления и информатики. — 2012. — № 1. — С. 79–93.

*Поступила 15.01.2012*