

**ПРОГНОЗИРУЮЩЕЕ УПРАВЛЕНИЕ
НЕЛИНЕЙНЫМИ ОБЪЕКТАМИ НА ОСНОВЕ
ЭВОЛЮЦИОНИРУЮЩИХ НЕЙРОННЫХ СЕТЕЙ
ПРЯМОГО РАСПРОСТРАНЕНИЯ**

Введение

Отличительной особенностью получивших в последнее время широкое распространение методов прогнозирующего управления динамическими объектами [1–3] является способность построенных на данном принципе контроллеров работать достаточно длительное время без вмешательства оператора. Из различных методов проектирования систем управления на основе концепции прогнозирующего управления с моделью (ПУМ) наиболее часто используются такие стратегии, как Dynamic Matrix Control (DMC), Model Algorithmic Control (MAC), Predictive Functional Control (PFC), Extended Prediction Self-Adaptive Control (EPSAC), Extended Horizon Adaptive Control (ЕНАС) и Generalized Predictive Control (GPC) [4].

ПУМ явно использует точную модель процесса для прогнозирования его поведения для расчета оптимальной траектории управления (последовательности управляющих сигналов) за счет оптимизации некоторой целевой функции в пределах заданного горизонта предсказания.

В настоящее время достаточно хорошо разработаны методы прогнозирующего управления с линейной моделью, которые применимы и для процессов, не являющихся существенно нелинейными. Действительно, в некоторых случаях оправдывает себя замена нелинейной модели линейной и нахождение параметров регулятора, которые обеспечивали бы наилучшее управление в некоторой компромиссной точке. Хотя такой подход и позволяет в ряде случаев существенно уменьшить априорную неопределенность и реализовать достаточно эффективное управление, ограничение линейными моделями не всегда обеспечивает получение требуемого результата. Поэтому более эффективным представляется разработка систем управления на основе адаптивного подхода в сочетании с методами теории искусственных нейронных сетей (ИНС), что позволяет, с одной стороны, построить довольно простые нелинейные нейросетевые модели, а с другой — адаптивно корректировать параметры как самих моделей, так и регулятора в соответствии с изменяющимися условиями. Однако при использовании ИНС возникает проблема выбора ряда характеристик сети (тип сети, ее структура, количество нейронов и вид их базисных функций), существенно влияющих на качество решаемой задачи. Если задача определения структуры сети дискретно оптимизационная (комбинаторная), то поиск оптимальных параметров осуществляется в непрерывном пространстве с помощью классических методов оптимизации. Кроме того, существенным недостатком применения ИНС для построения систем управления является их низкая скорость реакции, обусловленная необходимостью использования для обучения сети и реализации управления ресурсоемких градиентных алгоритмов оптимизации. В связи со сложностью получаемых таким образом контроллеров гибкими и наиболее подходящими для управления нелинейными объектами являются неградиентные методы оптимизации [5]. С учетом сказанного целесообразно применение эволюционного подхода, позволяющего не только осуществить выбор параметров сети, но и их настройку.

© О.Г. РУДЕНКО, А.А. БЕССОНОВ, 2015

Основным преимуществом использования эволюционных алгоритмов (ЭА) в качестве алгоритмов обучения является то, что многие параметры ИНС могут быть закодированы в геноме и определяться параллельно. Более того, в отличие от большинства алгоритмов оптимизации, предназначенных для потактового решения задачи, ЭА оперируют множеством решений — популяцией, что позволяет достичь глобального экстремума, не застревая в локальных. При этом информация о каждой особи популяции кодируется в хромосоме (генотипе), а получение решения (фенотипа) осуществляется после эволюции (отбора, скрещивания, мутации) путем декодирования.

Среди ЭА, являющихся стохастическими и включающими эволюционное программирование, эволюционные стратегии, генетические алгоритмы, генетическое программирование, в частности программирование с экспрессией генов, одними из наиболее распространенных являются генетические алгоритмы (ГА) [6]. С помощью простых арифметических операций они позволяют достаточно быстро не только вычислять значения управляющих сигналов, но и осуществлять адаптивную настройку внутренних параметров контроллера.

В данной работе предлагается схема прогнозирующего управления нелинейным объектом, являющаяся дальнейшим развитием схем, предложенных в работах [7, 8]. Она включает эволюционирующую нейросетевую модель и контроллер, реализованный на основе ГА.

Задача прогнозирующего управления

Рассмотрим задачу управления нелинейным объектом в дискретном времени, описываемом уравнением

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

где $y(k)$, $u(k)$ — выходные и управляющие сигналы соответственно; $f(\bullet)$ — неизвестная нелинейная функция; $\xi(k)$ — помеха.

Традиционная схема управления объектом (1) с прогнозирующей моделью с предсказывающей процедурой (пунктирная линия) показана на рис. 1. В текущий момент времени модель используется для предсказания влияния будущего управляющего сигнала \tilde{u} (или «манипулирующей переменной») на выходной сигнал объекта \tilde{y} . Предсказание производится на несколько шагов вперед. Учитывая будущие значения эталонного сигнала \tilde{r} и выходного сигнала \tilde{y} , будущее изменение управляющего сигнала \tilde{u} определяется с помощью оптимизационного процесса в целях уменьшения предсказанной ошибки.

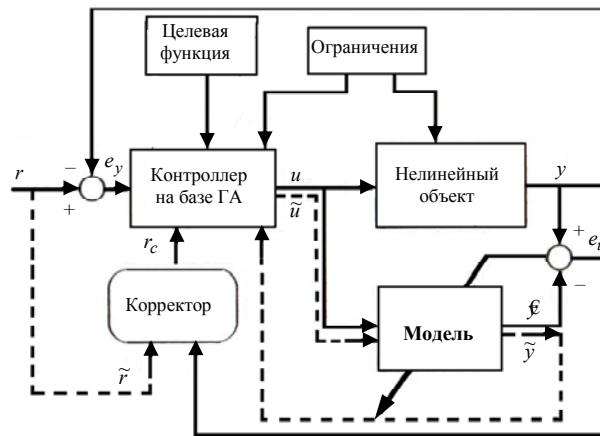


Рис. 1

Принцип работы прогнозирующего контроллера представлен на рис. 2. На основе модели, предположительно известной, определяются будущие значения выходов объекта $y(k+l)$, $l=1, 2, \dots, N_H$, где N_H — горизонт предсказания, которые зависят от его текущих состояний и будущих значений управляющих сигналов, $u(k+m)$, $m=1, 2, \dots, N_C$, где N_C — горизонт управления и $N_C \leq N_H$. Прогнозирующий контроллер вычисляет возможные будущие значения сигналов управления таким образом, чтобы значения будущих выходных сигналов были наиболее близкими к требуемым значениям $r(k+l)$, $l=1, 2, \dots, N_H$.

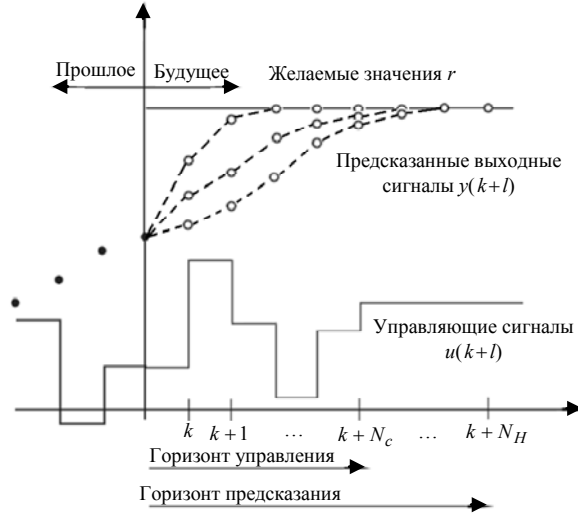


Рис. 2

Задача управления при этом заключается в точном отслеживании эталонной траектории $r(k)$ при отсутствии значительных колебаний управляющего сигнала и соблюдении при этом всех наложенных ограничений на диапазон управляющего сигнала, градиент управляющего сигнала и допустимый диапазон состояний процесса:

$$\begin{aligned} u_{\min} &\leq u(k) \leq u_{\max} \quad \forall k, \\ -\Delta u_{\max} &\leq \Delta u(k) \leq \Delta u_{\max} \quad \forall k, \\ y_{\min} &\leq y(k) \leq y_{\max} \quad \forall k. \end{aligned} \quad (2)$$

Математически цель процесса управления можно сформулировать как минимизацию некоторой целевой функции. В качестве целевой функции при прогнозирующем управлении широко используется функционал вида

$$J(N_0, N_p, N_c) = \sum_{i=N_0}^{N_p} \lambda_p(i) [y(k+i|k) - r(k+i)]^2 + \sum_{j=1}^{N_c} \lambda_c(j) [\Delta u(k+j-1)]^2, \quad (3)$$

где N_0 — начало интервала предсказания, λ_p и λ_c — некоторые весовые параметры.

Прогнозирующее нейруправление

При реализации нейросетевого прогнозирующего управления в качестве модели объекта (1) используется нейросетевая модель, для построения которой наиболее часто применяются многослойный персептрон (МП) и радиально-базисная сеть (РБС).

То обстоятельство, что обе сети позволяют аппроксимировать с любой заданной точностью любую непрерывную функцию, обеспечило их широкое при-

менение в задачах идентификации нелинейных объектов, а сочетание хороших аппроксимирующих свойств со способностью быстро обучаться позволяет использовать их для управления нелинейными динамическими объектами в реальном времени. Обе сети используют аппроксимацию нелинейного оператора объекта $f(\bullet)$ некоторой системой базисных функций, реализуемой нейронами, образующими слой сети, а задача идентификации сводится к обучению сети, заключающемся в настройке параметров нейронов на основе предъявления обучающих пар (измеряемых значений входных и соответствующих выходных переменных) обычно путем минимизации квадратичного функционала ошибки

$$J = e_u^2(k) = M \{ [\tilde{y}(k) - \mathfrak{f}(k)]^2 \}, \quad (4)$$

где $\tilde{y}(k)$ — вектор выходных сигналов модели.

Обучение многослойного персептрона, содержащего несколько (чаще всего не более двух) скрытых слоев, осуществляется обычно с помощью алгоритма обратного распространения ошибки, реализация которого связана с существенными вычислительными трудностями. Более простой архитектурой обладают РБС (они состоят из одного слоя нейронов). Однако для их обучения также используются градиентные алгоритмы, требующие значительных вычислительных затрат, что усложняет их применение в системах управления в реальном времени. В связи с этим возникла новая область, объединяющая нейронные и эволюционные парадигмы [9, 10], в основном направленная на объединение алгоритмов обучения с теорией эволюции (эволюция весов ИНС, архитектуры, алгоритмов обучения и активационных функций).

Использование в эволюционирующих искусственных нейронных сетях (ЭИНС) двух форм адаптации — эволюции и обучения, позволяющих изменять структуру сети, ее параметры и алгоритмы обучения без внешнего вмешательства, делает данные сети наиболее приспособленными для работы в нестационарных условиях и при наличии неопределенности относительно свойств исследуемого объекта и условий его функционирования.

Подробно процесс построения моделей нелинейных объектов на основе ЭИНС прямого распространения рассматривается в работах [11–13].

После окончания процесса обучения сеть используется для реализации алгоритма управления с предварительной коррекцией эталонной траектории.

Как и в случае идентификации, для нахождения минимума функционала (2) могут использоваться различные рекуррентные алгоритмы, в частности градиентные вида

$$u(k) = u(k-1) + \gamma(k)(\nabla_u e_y(k)), \quad (5)$$

где $\nabla_u e_y = \frac{\partial e_y(k)}{\partial u(k)}$; $e_y = r(k) - y(k)$; $\gamma(k) > 0$.

Следует отметить, что наличие помех измерений приводит к тому, что решения, получаемые как с помощью алгоритмов управления, использующих вычисления производных, так и с помощью эволюционных алгоритмов, обычно неудовлетворительны [14]. Более того, отсутствие информации о виде распределения помех зачастую приводит к тому, что возникает необходимость дополнительной коррекции эталонной траектории. При этом одним из наиболее эффективных подходов является применение модели Тьюки–Хьюбера для ε -засоренных вероятностных распределений [15]

$$p(x) = (1 - \varepsilon) p_0(x) + \varepsilon q(x), \quad (6)$$

где $p_0(x)$ — плотность соответствующего основного распределения $N(0, \sigma_1^2)$; $q(x)$ — плотность засоряющего (произвольного) распределения $N(0, \sigma_2^2)$, $\sigma_1^2 \ll \sigma_2^2$; $\varepsilon \in [0, 1]$ — параметр, характеризующий степень засорения основного распределения.

При использовании модели засорения (6) с помощью некоторых процедур можно настроить величины s_1^2 и s_2^2 , являющиеся оценками σ_1^2 и σ_2^2 , и осуществить коррекцию эталонной траектории управления следующим образом:

$$r_c(k+h) = r(k+h) - \mathcal{E}^2(k), \quad (7)$$

где $h = N_0, \dots, N_p$;

$$\mathcal{E}^2(k) = \begin{cases} \frac{(y_j^*(x_j) - \mathfrak{F}_j(x_j))^2}{s_1^2(k)}, & \text{если } |y_j^*(x_j) - \mathfrak{F}_j(x_j)| \leq 3s_1(k); \\ \frac{(y_j^*(x_j) - \mathfrak{F}_j(x_j))^2}{s_2^2(k)} & \text{в противном случае.} \end{cases} \quad (8)$$

Скорректированная с помощью процедуры (7) эталонная траектория подается на контроллер, который минимизирует функционал (3).

Прогнозирующее управление на основе ГА

В связи с высокими вычислительными затратами, связанными с вычислением градиента для вычисления возможных будущих сигналов управления, целесообразно использовать ГА. Общую схему ГА можно описать следующими шагами.

1. Инициализация популяции хромосом (набор случайно сгенерированных хромосом).
 2. Оценка фитнес-функции для всех особей.
 3. Выбор родительских особей.
 4. Кроссовер и мутации родительских особей и их потомков.
 5. Завершение формирования нового поколения из новых потомков и отдельных особей предыдущего поколения.
 6. Проверка критерия останова. Если не выполняется, то переход к шагу 2.
- Отдельные шаги алгоритма рассмотрим более подробно.

Инициализация популяции хромосом. Каждая особь популяции кодируется хромосомой $\{o_i; i = 1, \dots, N_{\text{pop}}\}$, содержащей набор генов, определяющих последовательность управляющих сигналов и внутренние параметры корректора (s_1^2 и s_2^2) и контроллера. Формат хромосомы представлен на рис. 3. Следует отметить, что помимо указанных на рисунке параметров, хромосома может содержать также гены, характеризующие активационные функции нейронов, оценки σ_1 , σ_2 и т.д. (это целесообразно при построении нейросетевой модели).



Рис. 3

Оценка фитнес-функции. Фитнес-функцией оптимизатора является целевая функция системы интеллектуального управления (3).

Кроссовер. С его помощью осуществляется обмен генетической информацией между двумя особями популяции. В данной работе используется одноточечный оператор, модифицированный таким образом, что он выполняется только в том случае, если после данной операции не будут нарушены ограничения управляющих сигналов (2). **Мутация.** Мутация — единственный способ ввода новой информации в популяцию. В данном алгоритме оператор мутации применяется для

случайно выбранного значения управляющего сигнала особи $\{o_i; i = 1, \dots, N_{\text{pop}}\}$ с некоторой вероятностью P_{mut} . В работе используется модифицированная гауссовская мутация [16], которая отсекает измененные значения, лежащие вне допустимого интервала, что позволяет удовлетворить условиям, ограничивающим допустимые значения управляющего сигнала (5).

Моделирование

Эксперимент 1. Решалась задача управления нелинейным объектом, описываемым уравнением

$$y(k+1) = \sin(u(k)) + \frac{y(k)}{1+y^2(k)}, \quad (9)$$

где $u(k)$ — сигнал управления.

Вначале производилась идентификация объекта с помощью эволюционирующей РБС. Начальная популяция состояла из 128 особей (ИНС). На каждой итерации алгоритма каждой особи предъявлялось $M = 5000$ обучающих пар.

После окончания процесса идентификации производилось прогнозирующее управление объектом (9) с помощью контроллера, использующего для вычисления управляющего воздействия ГА. Популяция состояла из 20 особей. Каждая особь была закодирована хромосомой, содержащей информацию о будущих N_H управляющих воздействиях и о внутренних параметрах контроллера и предиктора.

Желаемый закон изменения выходного сигнала объекта представлял собой пилообразный сигнал, изменяющийся в диапазоне $[-0,5; 0,5]$. Результаты управления объектом (9) при различных значениях горизонта предсказания приведены на рис. 4. Так для рис. 4, а $N_H = 10$, для рис. 4, б $N_H = 20$, а для рис. 4, в $N_H = 30$. На всех рисунках сплошной линией обозначен желаемый выходной сигнал, пунктирной — реальный выходной сигнал, сплошной с кружками — управляющий сигнал.

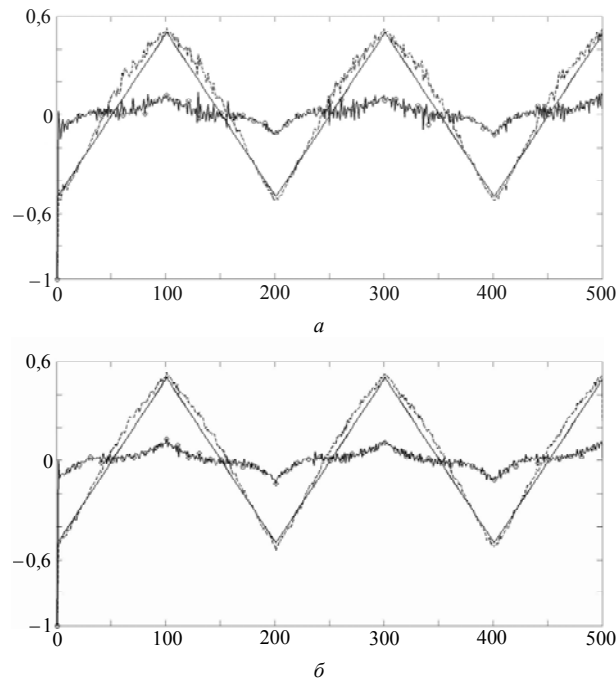
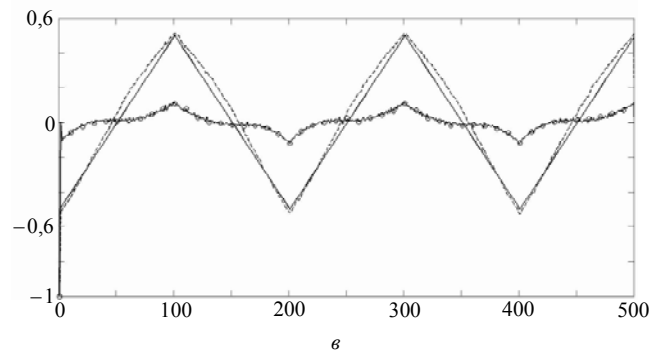


Рис. 4



Как следует из рис. 4, с увеличением горизонта предсказания происходит сглаживание результатов и повышается точность управления.

Эксперимент 2. Решалась задача идентификации зашумленного объекта, который описывался уравнением

$$y(k+1) = \frac{u(k)[\cos(y(k)) + 0,5]}{2 + y^2(k)} + \xi(k), \quad (10)$$

где $\xi(k)$ — нормально распределенная помеха с $\sigma^2 = 0,6$.

Желаемое значение выходного сигнала определялось по формуле

$$r(k+1) = 0,6 + 0,07 \sin(\pi k / 200) - 0,05 \cos(\pi k / 100). \quad (11)$$

Для идентификации объекта (10) использовалась популяция, состоящая из 150 особей. На каждом из 500 этапов эволюции каждой особи предъявлялось 5000 обучающих пар. В ходе работы алгоритма управления оценивались параметры помехи в соответствии с моделью (6). При этом были получены следующие значения: $s_1^2 = 0,6074$, $s_2^2 = 2,128$, $\varepsilon = 0,0021$.

Результаты управления объектом приведены на рис. 5. Здесь, как и в предыдущем эксперименте, сплошной линией обозначен желаемый выходной сигнал, пунктирной — реальный выходной сигнал, сплошной с кружками — управляющий сигнал.

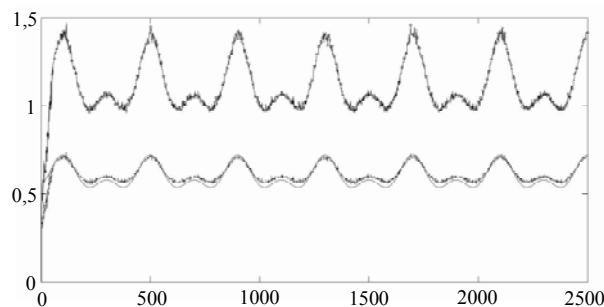


Рис. 5

Результаты моделирования свидетельствуют о том, что рассматриваемый подход позволяет эффективно решать задачу управления при наличии помех измерений.

Эксперимент 3. Изучалась задача прогнозирующего управления многомерным нелинейным динамическим объектом, описываемым уравнениями

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

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

где $\xi_1(k)$ и $\xi_2(k)$ — помехи измерений.

В качестве входных сигналов при обучении сети использовались некоррелированные случайные последовательности с равномерным законом распределения в интервале $[-1, 1]$. Для обучения сети использовалась популяция, состоящая из 250 особей. На каждом из 500 этапов эволюции каждой особи предъявлялось 10 тыс. обучающих пар. По окончании процесса идентификации осуществлялось управление многомерным объектом (12). Задавались требуемые значения выходных сигналов:

$$y_1^*(k) = \begin{cases} -0,5 + k/100, & k = \overline{1, 100}; \\ 1,5 - k/100, & k = \overline{101, 200}; \\ -2,5 + k/100, & k = \overline{201, 300}; \\ 3,5 - k/100, & k = \overline{301, 400}; \\ -4,5 + k/100, & k = \overline{401, 500}; \\ 0,2\sin(\pi k/100) + 0,05\cos(\pi k/200), & k = \overline{501, 1000}; \end{cases}$$

$$y_2^*(k) = \begin{cases} 0,2, & k = \overline{1, 500}; \\ 0,1, & k = \overline{501, 1000}. \end{cases}$$

Результаты работы нейрорегулятора отражены на рис. 6. На всех рисунках пунктирной линией показан требуемый выходной сигнал $r_i(k)$, сплошной — реальный $f_i(k)$, а линией с кружками — соответствующее изменение управляющего сигнала $u_i(k)$, $i=1, 2$. Рисунки 6, а, б отражают результаты моделирования при отсутствии помех измерений, т.е. $\xi_1(k) = 0$, $\xi_2(k) = 0$. На рис. 6, в, г приведены результаты моделирования при наличии помех измерений по обоим каналам управления, при этом $\xi_1(k)$ являлась нормально распределенной помехой с $\sigma^2 = 0,6$, а $\xi_2(k)$ — с $\sigma^2 = 0,8$. Оценки параметров помех в соответствии с моделью (6) дали следующие значения: $s_1^2 = 0,5912$, $s_2^2 = 1,9558$, $\varepsilon = 0,0023$ для первого канала управления и $s_1^2 = 0,7982$, $s_2^2 = 2,5847$, $\varepsilon = 0,0028$ для второго.

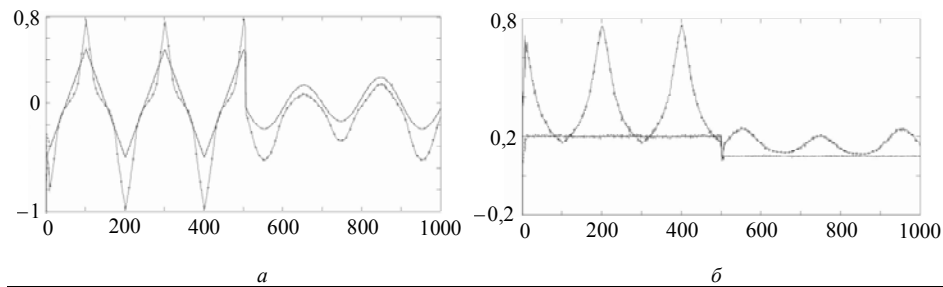
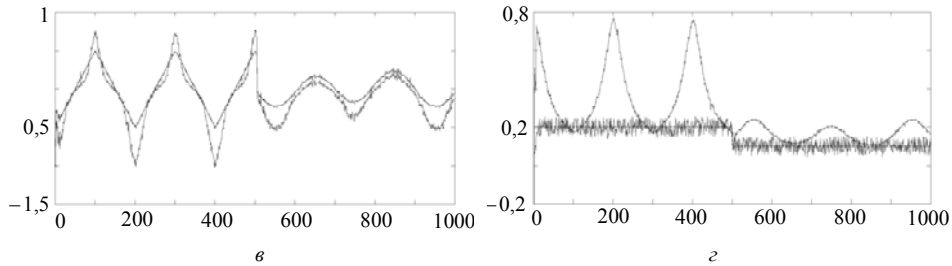


Рис. 6



Результаты моделирования свидетельствуют о том, что развиваемый подход позволяет синтезировать нейросетевой контроллер для управления многомерными нестационарными объектами при наличии помех измерений.

Заключение

Как следует из результатов моделирования, рассматриваемый подход к синтезу системы прогнозирующего управления нелинейными объектами на основе эволюционирующих нейронных сетей прямого распространения позволяет существенно ускорить процесс синтеза модели объекта за счет применения неградиентных процедур построения нейросетевой модели и вычисления управляющего сигнала. Результаты имитационного моделирования подтверждают эффективность применения предложенных методов для управления одно- и многомерными нелинейными объектами в реальном времени при наличии помех измерений.

Несмотря на ряд очевидных преимуществ такого подхода (отсутствие необходимости в обучающем множестве данных при построении нейросетевой модели объекта, независимость от структуры используемой ИНС и возможность одновременной настройки топологии сети и ее параметров и т.д.), при практической его реализации возникает ряд специфических для каждого конкретного случая проблем, например:

- выбор способа кодирования информации об используемой сети;
- выбор используемых генетических операторов кроссовера и мутации.

Способ кодирования влияет на скорость решения задачи, так как с увеличением сложности кодирования возрастают сложность декодирования и размерность пространства поиска, что приводит к росту вычислительных затрат на его реализацию. В свою очередь выбор проблемно-ориентированных генетических операторов, зависящий от способа кодирования, существенно влияет на качество получаемого решения. В этих условиях представляется целесообразным распараллеливание эволюционного поиска путем использования, например, коэволюционного подхода [17, 18].

О.Г. Руденко, О.О. Безсонов

ПРОГНОЗУЮЧЕ КЕРУВАННЯ НЕЛІНІЙНИМИ ОБ'ЄКТАМИ НА ОСНОВІ ЕВОЛЮЦІОНУЮЧИХ НЕЙРОННИХ МЕРЕЖ ПРЯМОГО РОЗПОВСЮДЖЕННЯ

Розроблено метод керування нелінійними об'єктами за допомогою еволюційних штучних нейронних мереж прямого розповсюдження. Нейронні мережі застосовуються для побудови нелінійної моделі об'єкта, яка в подальшому використовується для рекурсивного передбачення поведінки об'єкта в системі керування з прогнозуючою моделлю. Для навчання нейронної мережі

запропоновано застосування генетичних алгоритмів, що дозволяє значно прискорити процес навчання. Наводяться результати імітаційного моделювання, що підтверджують ефективність запропонованого методу керування.

O.G. Rudenko, A.A. Bezsonov

PREDICTIVE CONTROL OF NONLINEAR OBJECTS USING EVOLVING FEEDFORWARD NEURAL NETWORKS

The development of a method of nonlinear objects control with the evolving feedforward neural networks is considered. The neural networks are used to build a nonlinear model of the object, which is then utilized for recursive prediction of the object's behavior in the model predictive control. Genetic algorithms, which are used for the neural network's training, significantly speed up the training process. The simulation results confirm the effectiveness of the proposed control method.

1. *Qin S.J., Badgwell T.* A survey of industrial model predictive control technology // *Control Engineering Practice*. — 2003. — **11**, — N 7. — P. 733–764.
2. *Yu D.L., Yu D.W., Gomm, J.B.* Neural model adaptation and predictive control of a chemical rig // *IEEE Transaction on Control Systems Technology*. — 2006. — **14**, N 5. — P. 828–840.
3. *Lawrynczuk M.* An efficient nonlinear predictive control algorithm with neural models based on multipoint on-line linearization // *EUROCON. The International Conference on Computer as a Tool*. Warsaw. — 2007. September 9-12. — P. 777–784.
4. *Rusnak V., Fikar M., Najim K., Meszaros A.* Generalized predictive control based on neural networks // *Neural Process. Lett.* — 1996. — N 4. — P. 107–112.
5. *Potocnik P., Grabec I.* Nonlinear model predictive control of a cutting process // *Neurocomputing*. — 2002. — N 43. — P. 107–126.
6. *Holland J.* *Adaptation in natural and artificial systems*: 2nd edn. — Cambridge: MIT Press, 1992. — 228 p.
7. *Ichikawa Y., Sawa T.* Neural network application for direct feedback controllers // *IEEE Trans. Neural Networks*. — 1992. — N 3. — P. 224–231.
8. *Sette S., Boullart L., Van L.* Langenhove, Optimizing a production process by a neural network: genetic algorithm approach // *Eng. Appl. Artif. Intell.* — 1996. — N 9. — P. 681–689.
9. *Yao X.* Evolving artificial neural networks // *Proc. of the IEEE*. — 1999. — **87**, N 9. — P. 1423–1447.
10. *Yao X., Lin Y.* A new evolutionary system for evolving artificial neural networks // *IEEE Trans. on Neural Networks*. — 1997. — N 3. — P. 694–713.
11. *Руденко О.Г., Бессонов А.А.* Робастная нейроэволюционная идентификация нелинейных нестационарных объектов // *Кибернетика и системный анализ*. — 2014. — № 1. — С. 11–20.
12. *Руденко О.Г., Бессонов А.А.* Идентификация нелинейных нестационарных объектов с помощью эволюционирующей радиально-базисной сети // *Международный научно-технический журнал «Проблемы управления и информатики»*. — 2012. — № 4. — С. 5–14.
13. *Руденко О.Г., Бессонов А.А.* Многокритериальная оптимизация эволюционирующих сетей прямого распространения // *Там же*. — 2014. — № 6. — С. 29–41.
14. *Jin Y., Branke J.* Evolutionary optimization in uncertain environments — a survey // *IEEE Tr. Evolutionary Computation*. — 2005. — **5**, N 3. — P. 303–317.
15. *Хьюбер П.* Робастность в статистике. — М.: Мир, 1984. — 304 с.
16. *Бессонов А.А.* Обучение радиально-базисных сетей с помощью генетических алгоритмов с адаптивной мутацией // *Системи обробки інформації*. — 2012. — Вип. 3(2). — С. 177–180.
17. *Potter M.A., De Jong K.A.* Cooperative coevolution: An architecture for evolving coadapted sub-components // *Evolutionary Computation*. — 2000. — N 8. — P. 1–29.
18. *Moriarty D.E., Miikkulainen R.* Forming neural networks through efficient and adaptive coevolution // *Ibid.* — 1997. — N 5. — P. 373–399.

Получено 19.05.2015