

НЕЙРОУПРАВЛЕНИЕ НЕЛИНЕЙНЫМИ ДИНАМИЧЕСКИМИ ОБЪЕКТАМИ НА ОСНОВЕ МНОГОСЛОЙНОГО ПЕРСЕПТРОНА

Введение

Синтез системы управления любым объектом предполагает наличие его математической модели, которая, с одной стороны, должна адекватно отражать его свойства, а с другой – быть достаточно простой и удобной для реализации алгоритма управления. Отсутствие достаточно полной информации о реальном объекте, наличие у него различных нелинейностей, присутствие шумов и помех существенно затрудняет решение задачи управления. Практически все существующие методы управления базируются на идее линеаризации исследуемого объекта, что позволяет получить в ряде случаев приемлемые результаты. Однако изменение свойств объекта или условий его функционирования приводят к необходимости периодической коррекции модели и, соответственно, к изменению закона управления.

Трудности решения реальных задач традиционными методами привели к появлению и развитию методов интеллектуального управления, основой которых являются искусственные нейронные сети (ИНС) и фаззи-логика. Являясь в своей основе нелинейными системами, ИНС оказались весьма эффективными при решении задачи управления нелинейными объектами [1-4]. Простота реализации нейронных сетей и их способность к обучению делают их особенно привлекательными при управлении сложными нелинейными объектами в реальном времени. Структура системы адаптивного управления, построенная на основе ИНС, приведена на рис.1.

Нейросетевая модель объекта

Рассмотрим многомерный нелинейный динамический объект, представленный NARMAX моделью

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

где $\tilde{y}(k)$, $u(k)$ — выходной и входной сигналы объекта в момент времени k соответственно;

m , n — порядки запаздывания по выходному и входному каналам соответственно;

$f(\bullet)$ — неизвестная нелинейная функция;

$\xi(k)$ — помеха измерения выходного сигнала.

Введем вектор обобщенного входного сигнала размерностью $(n+m) \times 1$

© О.Г. Руденко, А.А. Шамраев, 2002

$$x(k) = [y(k-1), \dots, y(k-m), u(k-1), \dots, u(k-n)]. \quad (2)$$

перепишем уравнение (1) в виде

$$\tilde{y}(k) = f[x(k), k] + \xi(k). \quad (3)$$

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

Нейросетевая модель объекта (3) (см. рис.2) имеет вид

$$\widehat{y}(k) = f^L[(W^L)^T f^{L-1}[(W^{L-1})^T f^{L-2}[\dots f^1[(W^1)^T, x(k)]]]], \quad (4)$$

где L — количество слоев в сети;

W^i — вектор весовых параметров нейронов i -го слоя сети;

$f^i(\bullet)$ — функция активации i -го слоя.

Обучение сети, к которому сводится задача идентификации, состоит в настройке ее параметров на основе сравнения выходных сигналов объекта \tilde{y} и модели \widehat{y} и заключается в минимизации обычно квадратичного функционала ошибки [2,5]

$$J_I = r^2(k) = M\{[\tilde{y}(k) - \widehat{y}(k)]^2\}. \quad (5)$$

По аналогии с традиционным подходом к решению задачи идентификации, при котором процесс построения модели разбивается на два этапа – структурную и параметрическую идентификацию, применение ИНС также требует решения двух задач: определения структуры сети и настройки (обучения) ее параметров.

Обычно начальная конфигурация сети выбирается двухслойной с произвольным числом нейронов в слоях.

После проведения ряда экспериментов с различными конфигурациями сети выбирается та, которая дает минимальное значение функционала ошибки или, при достижении заданной точности решения, требует меньших вычислительных затрат.

Если качество обучения недостаточно, увеличивают число нейронов слоя или количество слоев. Если же наблюдается явление переобучения, то следует уменьшить число нейронов в слое или удалить один или несколько слоев.

Обучение сети осуществляется с помощью какого-либо метода минимизации функционала (5). В настоящее время существует большое число таких методов, отличающихся как динамическими свойствами, так и вычислительной сложностью. Одним из наиболее широко используемых является метод Левенберга-Марквардта [5,6,7]

$$w(k+1) = w(k) - [J^T(k)J(k) + \mu(k)I]^{-1} J^T(k)r(k), \quad (6)$$

где $r(k)$ — вектор остатков,

$J(k)$ — матрица Якоби с элементами $J(k)_{ij} = \partial r(k)_i / \partial w(k)_j$,

I — единичная матрица,

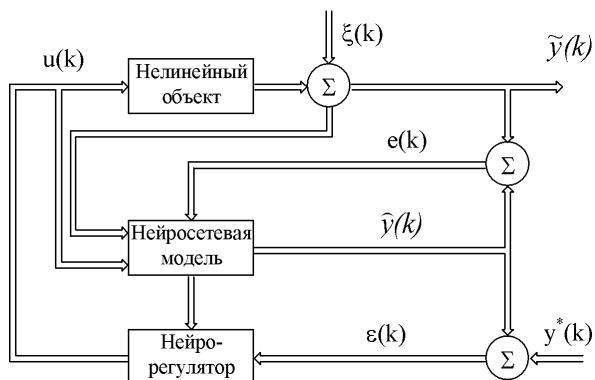


Рис. 1 – Структура системы адаптивного управления

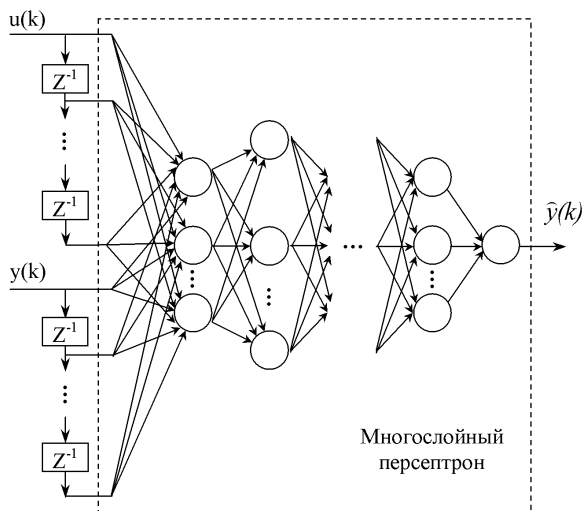


Рис. 2 – Нейросетевая модель объекта

$\mu(k)$ — радиус доверительного интервала, который на каждой итерации вычисляется в соответствии с выражением

$$\mu(k) = \begin{cases} \rho \cdot \mu(k-1), & \text{если } J_I(k) < J_I(k-1), \\ \sigma \cdot \mu(k-1), & \text{если } J_I(k) \geq J_I(k-1). \end{cases}$$

$$0 < \rho < 1, \sigma > 1.$$

При решении задачи идентификации были выбраны следующие начальные значения коэффициентов $\mu(0) = 0.01$, $\rho = 0.9$, $\sigma = 1.1$.

Для обучения МП, по методу Левенберга-Марквардта необходима оценка матрицы Якоби $J(k)$ на каждой итерации k [1]. В терминах архитектуры МП и набора обучения, матрица Якоби имеет $P \cdot N^L \times W$ элементов вида $\partial r_{b,p} / \partial w_{ij}^{lm}$, где P — число образцов в наборе обучения, а N^L — число выходных узлов. Каждый элемент матрицы Якоби $\partial r_{b,p} / \partial w_{ij}^{lm}$ может быть рассчитан согласно выражению

$$\frac{\partial r_{b,p}}{\partial w_{ij}^{lm}} = \delta_{i,b,p}^l y_{j,p}^m, \quad (7)$$

где $\delta_{i,b,p}^l$ рассчитывается при обратном проходе согласно формуле

$$\begin{aligned} \delta_{i,b,p}^L &= f'(a_{i,p}^L), & i &= b \\ \delta_{i,b,p}^L &= 0, & i &\neq b \\ \delta_{j,b,p}^m &= f'(a_{j,p}^m) \sum_{n_i^l \in T_j^m} w_{ij}^{lm} \delta_{i,b,p}^l, & m &< L \end{aligned} \quad (8)$$

Алгоритм управления

Обозначим требуемое значение вектора выходных сигналов объекта как $y^*(k)$. Задача управления заключается в определении значений $u(k)$, обеспечивающих минимум функционалу

$$J_C = r^2(k) = \frac{1}{2} \left\| y^*(k) - \bar{y}(k) \right\|^2. \quad (9)$$

Как и в случае идентификации, для нахождения минимума функционала (9) могут быть использованы различные рекуррентные алгоритмы, в частности, градиентные вида

$$u(k) = u(k-1) + \gamma(k)(\nabla_u r(k)), \quad (10)$$

где $\nabla_u r(k) = \frac{\partial r(k)}{\partial u(k)}$,

$$\gamma(k) > 0.$$

Выбор оптимального значения $\gamma(k)$, обеспечивающего максимальную скорость сходимости алгоритма (10), приводит, как известно, к алгоритму Качмажа [8], который с учетом (2) и (4) и того, что при управлении объектом (1) используется только вектор $u(k-1)$, может быть записан так

$$u(k) = u(k-1) + \nabla_u r(k) [\delta I + \nabla_u^T r(k) \nabla_u r(k)]^{-1} r(k), \quad (11)$$

где $\delta > 0$.

Параметр δ играет в (11) регуляризирующую роль, т.е. используется в алгоритме (11) для повышения его вычислительной устойчивости.

Моделирование

Рассматривалась задача управления нелинейными динамическими объектами, описываемыми различными уравнениями.

Моделирование осуществлялось в среде MatLab 6.1, с использованием ПЭВМ на базе процессора Pentium III-733, с ОЗУ объемом 384 МБайт.

В качестве входных сигналов при обучении сети использовались некоррелированные случайные последовательности с равномерным законом распределения в интервале $[-1, 1]$. Объекты предполагались стационарными. Для обучения сети использовался алгоритм Левенберга-Марквардта, а для управления – (11) с различными значениями δ . Обучающий набор включает 5000 обучающих пар. Требуемая точность идентификации задавалась на уровне $\varepsilon = 0.0001$, а инициализация весов и смещений осуществлялась согласно правилу, при котором каждый вес равномерно распределен в диапазоне $[-\alpha, \alpha]$, где α задается формулой

$$\alpha(w_{ij}^{lm}) = \frac{1}{N^m + 1}.$$

Результаты исследования динамического объекта, описываемого уравнениями

$$\begin{aligned} \tilde{y}_1(k) &= \frac{15u_1(k-1)y_2(k-1)}{2+50[u_1(k-1)]^2} + 0.5u_1(k-1) + 0.25y_2(k-1) + 0.1; \\ \tilde{y}_2(k) &= \frac{\sin(\pi u_2(k-1)y_1(k-1)+2u_2(k-1))}{3}, \end{aligned} \quad (12)$$

приведены на рис. 3-рис. 4.

При построении нейросетевых моделей объектов получалась сеть, состоящая из 26 нейронов, расположенных в трех слоях следующим образом: 12 нейронов во входном слое, 12 нейронов в скрытом слое и 2 нейрона в выходном. В качестве функции активации нейронов входного и скрытого слоев использовался гиперболический тангенс, а выходного слоя – линейная функция.

Результаты работы нейрорегулятора, реализующего алгоритм (11) при $\delta = 0.05$, $\mu(0) = 0.01$ приведены на рис. 3. На всех рисунках пунктирной линией показан требуемый выходной сигнал $y_i^*(k)$, сплошной – реальный сигнал $\hat{y}_i(k)$ на выходе объекта, а линией с кружочками – соответствующее изменение управляющего сигнала $u_i(k)$ ($i = 1, 2$). Требуемые значения выходных сигналов задавались следующие:

$$\begin{aligned} y_1^*(k) &= 0.3 \sin(\pi k/100); \\ y_2^*(k) &= \begin{cases} 0.1 & \text{при } k = \overline{1, 500}; \\ -0.05 & \text{при } k = \overline{500, 1000}. \end{cases} \end{aligned}$$

Рис. 3 отражает работу регулятора при отсутствии помех измерения ($\xi(k) = 0$) (см. рис. 1), а рис. 4 — при наличии случайной помехи $\xi(k)$ равномерно распределенной в интервале $[-0.3, 0.3]$. Появление этой помехи

привело к *незначительному* ухудшению качества управления объектом. Объясняется это тем, что при построении нейросетевой модели использовался алгоритм Левенберга-Марквардта, обладающий высокими сглаживающими свойствами, что позволило сети достаточно точно настроиться.

На рис. 5 приведены результаты работы адаптивного нейрорегулятора при управлении объектом, описываемым уравнениями [8]

$$\begin{aligned} \tilde{y}_1(k) &= \frac{y_1(k-1)}{1+[y_2(k-1)]^2} + u_1(k-1); \\ \tilde{y}_2(k) &= \frac{y_1(k-1)y_2(k-1)}{1+[y_2(k-1)]^2} + u_2(k-1). \end{aligned} \quad (13)$$

Как и при исследовании объекта (12), идентификация (13) осуществлялась на основе алгоритма (6), а при управлении этим объектом использовались та же структура сети 12-12-2 и алгоритм (11) при $\delta = 0.1$. На рис. 5 приведены результаты управления объектом (13) при задании

$$\begin{aligned} y_1^*(k) &= \cos(\pi k/100); \\ y_2^*(k) &= \begin{cases} 1 - 0.004k & \text{при } k = \overline{1, 500}; \\ 1 & \text{при } k = \overline{500, 1000}. \end{cases} \end{aligned}$$

Как видно из рисунка нейросетевой регулятор обеспечивает высокое качество управления. Выходной сигнал объекта $\tilde{y}_1(k)$ практически совпадает с $y_1^*(k)$. Вид требуемого сигнала $y_1^*(k)$ приводит к необходимости выработки периодического сигнала $u_1(k)$. Эта периодичность сказывается на виде как выходного сигнала $y_2(k)$, так и управления $u_2(k)$. Резкое изменение на 501 шаге значения требуемого сигнала $y_2^*(k)$ с -1 до $+1$ (рис. 5б) приводит к возникновению переходного процесса длящегося около 20 тактов. После окончания переходного процесса качество управления достигает прежнего уровня.

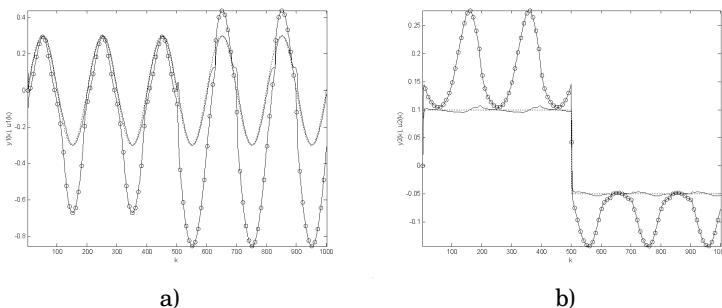


Рис. 3 – Управление объектом, описываемым уравнениями (12), при $\delta = 0.05$: изменение выходного сигнала $y_1(k)$ и управления $u_1(k)$ (а); изменение выходного сигнала $y_2(k)$ и управления $u_2(k)$ (б).

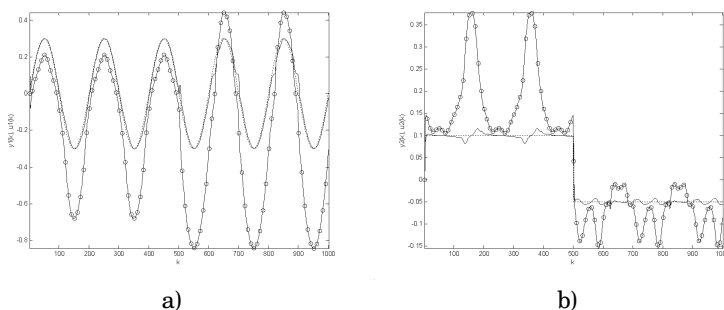


Рис. 4 – Управление объектом, описываемым уравнениями (12), при наличии помехи $\xi(k)$ и $\delta = 0.0001$: изменение выходного сигнала $y_1(k)$ и управления $u_1(k)$ (а); изменение выходного сигнала $y_2(k)$ и управления $u_2(k)$ (б).

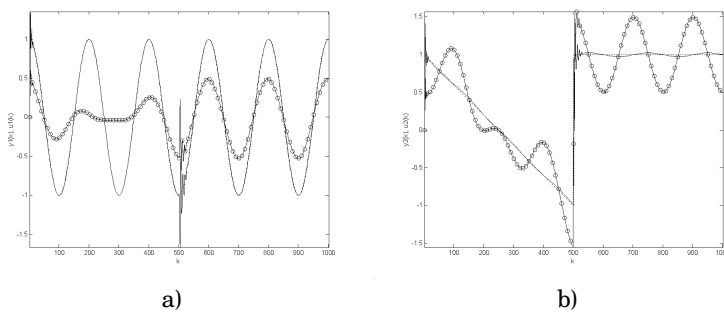


Рис. 5 – Управление объектом, описываемым уравнениями (13), при $\delta = 0.1$: изменение выходного сигнала $y_1(k)$ и управления $u_1(k)$ (а); изменение выходного сигнала $y_2(k)$ и управления $u_2(k)$ (б).

Заключение

Обобщая результаты экспериментов, можно сделать следующие выводы: во-первых, качество управления в первую очередь зависит от точности, с которой нейросетевая модель отражает структуру объекта, во-вторых, получение на выходе объекта требуемого сигнала произвольной формы, при использовании алгоритма (11), не всегда возможно, так как для многомерного объекта равенство нулю градиента функционала по одному из выходов может приводить к резким изменениям управления и, следовательно, к длительному переходному процессу на выходе объекта. Кроме того, остается открытым вопрос выбора значения параметра δ в (11), который зависит как от структуры исследуемого объекта, так и от формы требуемого сигнала и, поэтому, этот параметр должен выбираться индивидуально для каждого случая. Решение задачи идентификации в условиях шума может способствовать повышению качества управления, так как дополнительное возмущение способствует получению более адекватной нейросетевой модели.

Литература

1. Омату С., Халид М., Юсоф Р. Нейроуправление и его приложения. - М.: ИПРЖР, 2000.-272с.
2. Shepherd A.J. Second-Order Methods for Neural Networks.- New York: Springer, 1997. – 146p.
3. Narendra K.S., Parthasarathy K. Identification and control of dynamical systems using neural networks// IEEE Trans. Neural Networks.-1990.-V.1.- №1.-Pp.4-26.
4. Руденко О.Г., Бодянский Е.В. Основы теории искусственных нейронных сетей. - Харьков: ТЕЛЕТЕХ, 2002.-317с.
5. Zhou G., Si J. Advanced neural-network training algorithm with reduced complexity based on Jacobian deficiency// IEEE Trans. Neural Networks.-1998.-V.9.- №3.-Pp.448-453.
6. Медведев В.С., Потемкин В.Г. Нейронные сети. MATLAB 6. - М.: ДИАЛОГ-МИФИ, 2002.-496 с.
7. Haber R., Unbehauen H. Structure identification of nonlinear dynamic systems// Automatica.-1990.-V.26.- №4.-Pp.651-677.
8. Райбман Н.С., Чадеев В.М. Построение моделей процессов производства. – М.: Энергия, 1975.–375 с.

Получено: 26.10.2002