УДК 681.513.7

АЛГОРИТМЫ АДАПТИВНОГО УПРАВЛЕНИЯ С ОГРАНИЧЕНИЯМИ НА ФАЗОВЫЕ ПЕРЕМЕННЫЕ

БОДЯНСКИЙ Е.В., МАНЖАК Е.Л., УДОВЕНКО С.Г.

Предложены алгоритмы адаптивного управления стохастическими объектами с ограничениями на переменные. Обсуждены различные критерии управления с позиций рассматриваемых подходов.

Построение байесовской модели объекта управления может быть интерпретировано как параметризация множества условных распределений вероятностей $p(y(k)\,|\,u(k), d_{(k-1,1)})$, где k – дискретное время, $y(k)$ – вектор выходных параметров объекта, $u(k)$ – вектор управляющих воздействий, $d_{(k-1,1)}=\{u(1), y(1), \ldots, u(k-1), y(k-1)\}$. Элементы этого множества являются скалярными функциями, отражающими информацию об известной части предыстории $d_{(k-1,1)}$:

$$p(y(k)\,|\,u(k), d_{(k-1,1)}) = \Psi(y(k), u(k), d_{(k-1,1)}). \quad (1)$$

Предположим существование некоторой конечномерной векторной величины $x(k-1)$, однозначно связанной с предысторией и определяющей состояние системы в момент $k-1$. Это позволяет представить зависимость (1) в следующем виде:

$$p(y(k)\,|\,u(k), d_{(k-1,1)}) = \Psi(y(k), u(k), x(k-1)). \quad (2)$$

Так как на каждом такте идентификации совокупность данных должна обновляться, то логично предположить наличие детерминированного обновляющего соотношения:

$$x(k) = \Phi(x(k-1), u(k), y(k)). \quad (3)$$

Для построения модели необходимо набрать достаточный объем статистики и осуществить параметризацию скалярной функции Ψ и многомерной функции Φ. Параметризация облегчается, если принять допущение о линейности модели. В этом случае последнюю можно представить следующим образом:

$$y(t)=\bar{\psi}(t, u(k-1), x(k-1)) + e(k), \quad (4)$$

где $e(k)$ – случайная векторная составляющая процесса с нулевым математическим ожиданием и ковариационной матрицей $R_e = M\{e(k)e^T(k)\}$.

При линейности многомерной функции Φ зависимость (3) с учетом (4) преобразуется к виду:

$$x(k+1) = Ax(k) + Bu(k) + e(k),\ t=0,1,2,\ldots, \quad (5)$$

где $x(k)$ – s-мерный вектор состояний объекта в момент времени k, A и B – матрицы параметров объекта.

Представление байесовской модели объекта в форме (5) является удобным при решении задачи адаптивного управления динамическим стохастическим процессом при наличии ограничений на фазовые переменные. Синтез таких алгоритмов можно осуществить, используя метод динамического программирования (терминальное управление) и принципы локальной оптимизации (локально-оптимальное управление) [1].

Применение алгоритма терминального управления усложняется в случае априорно неизвестных значений параметров объекта, задаваемых матрицами A и B. В этом случае общая методика использования стандартной процедуры (2)–(5) состоит в том, что в начале процесса управления решается N-шаговая задача динамического программирования в обратном времени при достаточно произвольно выбранных априорных оценках $A(0)$, $B(0)$ истинных параметров. Далее на объекте (5) реализуется управление $u(0)$, наблюдается реакция $x(1)$ и на основании этих данных с помощью какого-либо адаптивного алгоритма идентификации уточняются оценки. После этого вновь необходимо решить задачу динамического программирования, но уже $(N-1)$-шаговую и на оценках $A(1)$, $B(1)$. После реализации $u(1)$ и наблюдения $x(2)$ уточняются оценки $A(2)$, $B(2)$ и решается $(N-2)$-шаговая задача и т.д. Основные проблемы реализации такого алгоритма подхода связаны с большим объемом вычислений, что усложняет работу в реальном времени и требует применения достаточно мощных управляющих ЦВМ.

Существенного упрощения можно добиться, используя вместо терминального критерия одношаговую целевую функцию

$$J(k)=M\{x^T(k+1)Q(k+1)x(k+1)+u^T(k)R(k)u(k)\}, \quad (6)$$

где $Q(k+1)$, $R(k)$ – штрафные матрицы, $k=0,\ldots,N-1$.

Минимизация (6) на каждом шаге приводит к локально-оптимальному закону управления

$$u^*(k)=-(B^TQ(k+1)B+$$
$$+R(k))^{-1}B^TQ(k+1)Ax(k)= K^*(k)x(k), \quad (7)$$

при этом значение критерия на каждом такте составляет [1]:

$$\min_{u(k)} J(k)=x^T(k)A^TQ(k+1)(I+B(B^TQ(k+1)B+R(k))^{-1}*$$
$$*B^TQ(k+1))Ax(k)+TrQ(t+1)P_w=x^T(k)(A^TQ(k+1)A-$$
$$-K^{*T}(k)(B^TQ(k+1)B+R(k))K^*(k))x(k)+TrQ(k+1)P_w,$$

а суммарные потери –

$$\sum_{k=0}^{N} J(k) = x^T(0)Q(1)x(0)+$$
$$+\sum_{k=0}^{N-1} x^T(k)A^TQ(k+1)(I-B(B^TQ(k+1)B+$$
$$+R(k))^{-1}B^TQ(k+1))Ax(k)+\sum_{k=0}^{N} TrQ(k)P_w.$$

С вычислительной точки зрения алгоритм (7) удобен для реализации в реальном времени, кроме того, на его основе могут быть синтезированы процедуры, пригодные для работы в условиях априорной неопределенности о параметрах объекта. При адаптивном управлении в соответствии с принципом разделения в алгоритме управления (7) неизвестные параметры объекта заменяются своими оценками и управляющее воздействие вычисляется согласно выражению

$$\tilde{u}(k)=-(B^T(k)Q(k+1)B(k)+$$

$$+R(k))^{-1}B^T(k)Q(k+1)A(k)x(k)=-\tilde{K}(k)x(k). \qquad (8)$$

В качестве алгоритма идентификации можно использовать какую-либо процедуру оценивания, например, рекуррентный метод наименьших квадратов или байесовскую процедуру оценивания. Для этого введем составную матрицу $C=(A \vdots B)$ неизвестных параметров объекта (5), матрицу оценок $C(k)=(A(k) \vdots B(k))$ и составной вектор входов $z^T(k)=(x^T(k) \mid (u(k)+\xi(k))^T)$. Тогда алгоритм идентификации будет иметь вид

$$\begin{cases} C(k+1) = C(k) + (x(k+1) - C(k)z(k))z^T(k)D(k) = \\ = C(k) + (x(k+1) - \tilde{x}(k+1))z^T(k)D(k), \quad C(0) = C(\alpha), \\ D(k) = D(k-1) - \dfrac{D(k-1)z(k)z^T(k)D(k-1)}{1 + z^T(k)D(k-1)z(k)}, \\ D(0) = \rho^{-1}I, \ \rho > 0, \end{cases} \qquad (9)$$

где $\xi(k)$ – последовательность независимых случайных возмущений, присутствующих в канале управления, либо вводимая искусственно в целях идентификации, $C(\alpha)$ – априорная оценка матрицы C.

Также следует отметить, что управление (7), оптимальное по критерию, рассчитываемому на каждом шаге, в ряде случаев позволяет получить меньшее значение терминального критерия, чем управление, полученное с помощью динамического программирования [2]. Основным недостатком локально-оптимальных алгоритмов является то, что в процессе управления могут возникать недопустимые перерегулирования управляющих воздействий, т.е. резко ухудшается энергетика системы. Тем не менее, если для набора возможных стандартных ситуаций априорно рассчитать оптимальные траектории, далее их можно реализовать с помощью алгоритма (7). Конечно говорить о стандартных ситуациях в реальных задачах не всегда корректно, поэтому необходимо синтезировать специальные алгоритмы, реализуемые в виде дополнительных контуров адаптации и обеспечивающие требуемые свойства процессу управления.

Известно, что матрицы критериев $Q(k)$ и $R(k)$ задают ограничения на точность управления и энергетику управляющих воздействий. Выбираются эти матрицы обычно эмпирическим путем на стадии проектирования системы управления с помощью многочисленных имитационных экспериментов. Строгих математических процедур для их вычисления нет, хотя в большом числе реальных задач требуется выполнение ограничений (энергетических, экологических, позиционных), особенно на управляющие воздействия.

Рассмотрим задачу терминального управления объектом (5) по критерию

$$J=M\{\sum_{k=0}^{N} x^T(k)Q(k)x(k)\} = M\{x^T(N)Q(N)x(N)+$$
$$\sum_{k=0}^{N-1} x^T(k)Q(k)x(k)\},$$

при дополнительных ограничениях на энергетику управлений, задаваемых системой неравенств

$$\begin{cases} u^T(N-1)Ru(N-1) \le U^2(N-1), \\ u^T(N-2)Ru(N-2) \le U^2(N-2), \\ \vdots \\ u^T(0)Ru(0) \le U^2(0). \end{cases}$$

Для решения задачи сформируем лагранжиан

$$L=J+\sum_{k=0}^{N-1} \lambda(k)\,(u^T(k)Ru(k)-U^2(k)) =$$

$$=M\{x^T(N)Q(N)x(N)+\sum_{k=0}^{N-1} x^T(k)Q(k)x(k)+$$
$$\sum_{k=0}^{N-1} \lambda(k)\,(u^T(k)Ru(k)-U^2(k))\}, \qquad (10)$$

где $\lambda(k)$ — неотрицательные неопределенные множители Лагранжа, удовлетворяющий условиям дополнительной нежесткости.

Для выражения (10) справедливы соотношения:

$$\min\max L=\min_{u(k),k=0,1,\ldots,N-2}\ \max_{\lambda(k)} M\{\ldots$$

$$\{\min_{u(N-1)}\ \max_{\lambda(N-1)} M\{x^T(N)Q(N)x(N)+$$

$$+\sum_{k=0}^{N-1} x^T(k)Q(k)x(k)+$$

$$+\sum_{k=0}^{N-1} \lambda(k)\,(u^T(k)Ru(k)-U^2(k))\}\}\ldots\}.$$

При этом несложно видеть, что

$$\min_{u(N-1)}\ \max_{\lambda(N-1)} L = M\{\sum_{k=0}^{N-1} x^T(k)Q(k)x(k) +$$
$$+\sum_{k=0}^{N-2} \lambda(k)(u^T(k)Ru(k) - U^2(k)) +$$
$$\min_{u(N-1)}\ \max_{\lambda(N-1)} M\{x^T(N)Q(N)x(N) + \qquad (11)$$
$$+\lambda(N-1)(u^T(N-1)Ru(N-1) - U^2(N-1))\}.$$

Оптимизация выражения (11) связана с решением системы уравнений Куна-Такера

$$\begin{cases} \nabla_{u(N-1)} L = 0, \\ \partial L \,/\, \partial\lambda(N-1) \le 0, \quad \lambda(N-1) \ge 0, \end{cases} \text{ или}$$

$$\begin{cases} \nabla_{u(N-1)} L = 2B^T Q(N)Ax(N-1) + 2B^T Q(N)Bu(N-1) + \\ +2\lambda(N-1)Ru(N-1) = 0, \\ \partial L \,/\, \partial\lambda(N-1) = u^T(N-1)Ru(N-1) - U^2(N-1) \le 0. \end{cases}$$

Она может быть проведена с помощью процедуры нелинейного программирования Эрроу-Гурвица-Удзавы [3]. При этом закон управления может быть сразу записан в аналитической форме

$$u(N-1)=-(B^TQ(N)B+\lambda(N-1)R)^{-1}B^TQ(N)Ax(N-1)=$$

$$=-K(N-1)x(N-1), \qquad (12)$$

а для нахождения настраиваемого множителя Лагранжа введем рекуррентную процедуру, работающую в ускоренном масштабе времени так, что между двумя соседними тактами управления реального времени k-1 и k может происходить M машинных итераций ускоренного времени.

Алгоритм управления представлен пунктами:

1. Задается достаточно малое значение $\lambda^o(N-1)$, что соответствует отсутствию ограничений.

2. По (12) вычисляется $u^0(N-1)=f(\lambda^0(N-1))$.

3. Если $u^{0T}(N-1)Ru^0(N-1)\le U^2(N-1)$, то полагается $\lambda^0(N-1)=\lambda(N-1)$ и $u(N-1)=u^0(N-1)$, т.е. вычисляется реализуемое на объекте управление.

4. Если $u^{0T}(N-1)Ru^0(N-1)>U^2(N-1)$, то множитель Лагранжа подстраивается согласно процедуре

$$\lambda^1(N-1)=[\lambda^0(N-1)+\mu(N-1)$$
$$(u^{0T}(N-1)Ru^0(N-1)-U^2(N-1))]_+,$$

где $\mu(N-1)$ – параметр шага градиентного поиска, $[\lambda]_+=\max\{0,1\}$.

5. По (12) вычисляется $u^1(N-1)=f(\lambda^1(N-1))$ и происходит возврат к пункту 3.

Продолжая выкладки по индукции и вводя обозначения $\lambda(k)R=R(k)$, можно записать алгоритм управления

$$u(N-j)=-K(N-j)x(N-j),\quad j=1,2,...,N, \tag{13}$$

$$K(N-j)=(R(N-j)+$$
$$+B^TP(N-j+1)B)^{-1}B^TP(N-j+1)A, \tag{14}$$

$$P(N-j)=Q(N-j)+A^TP(N-j+1)A-K^T(N-j)(R(N-j)+$$
$$+B^TP(N-j+1)B)K(N-j)=Q(N-j)-$$
$$-K^T(N-j)R(N-j)K(N-j)+(A-BK(N-j))^TP(N-j+1)* \tag{15}$$
$$*(A+BK(N-j))=Q(N-j)+A^TP(N-j+1)(I-B(R(N-j)+$$
$$+B^TP(N-j+1)B)^{-1}B^TP(N-j+1)A,$$

$$\lambda^t(N-j)=[\lambda^{t-1}(N-j)+$$
$$\mu(N-j)(\|u^{t-1}(N-j)\|_R^2-U^2(N-j))]_+. \tag{16}$$

Несложно видеть, что алгоритм (13)-(16) есть не что иное, как базовая процедура, дополненная контуром адаптации (16) для настройки матрицы критерия R(k). Введение дополнительного контура позволяет жестко контролировать энергетику на протяжении всего периода управления.

Литература: 1. *Кельманс Г.К., Позняк А.С., Черницер А.В.* Локально-оптимальное управление объектами с неизвестными параметрами // АТ.– 1982.– №10.– С. 80-93. 2. *Изерман Р.* Цифровые системы управления.– М.: Мир.– 1984.– 544 с. 3. *Поляк Б.Т.* Введение в оптимизацию.– М.: Наука.– 1983.– 384 с.

Поступила в редколлегию 12.10.97

Бодянский Евгений Владимирович, д-р техн. наук, профессор кафедры технической кибернетики ХТУРЭ. Научные интересы: теория адаптивных систем, искусственные нейронные сети, техническая диагностика. Увлечения: фелинология, восточные учения, японская поэзия. Служебный адрес: 310726, Харьков-726, пр. Ленина, 14. Домашний адрес: 310145, Харьков-145, ул. Клочковская, 152 “а”, кв. 10.

Манжак Екатерина Леонидовна, системный аналитик НПФ "Вектор". Научные интересы: идентификация стохастических ситем. Увлечения: иностранные языки, современные танцы. Адрес: 310000, Украина, Харьков, ул. акад. Павлова, 134/16, кв. 135, тел. 68-98-29.

Удовенко Сергей Григорьевич, канд, техн, наук, доцент кафедры ЭВМ ХТУРЭ. Научные интересы: идентификация стохастических ситем. Увлечения: иностранные языки, поэзия. Адрес: 310726, Украина, Харьков, пр. Ленина, 14, тел. 38-38-74.

УДК 519.513

FUZZY MODELLING OF NONLINEAR DYNAMIC SYSTEM BY INDUCTIVE LEARNED RULES

PETER OTTO

A fuzzy-modelling method for dynamical systems is presented. The input-output data of the system are changed by fuzzy-sets into linguistical described examples of the systems behaviour. These examples are the base for the knowledge acquisition process by using a machine learning method for inductive learning of production rules (ID3). Contradictory examples will be considered in different knowledge bases and they will be used for defuzzyfization with the center of gravity method. The way this method works is showed by a simulated example.

1. Introduction

There is hardly ever the possibility to optimize directly on the process by creating an optimal fuzzy-controller. Therefore the control design needs the existence of a model on which the optimization can be done. Using classic modelling procedures there are enormeus difficulties to find an appropriate model because of the fact that fuzzy-controllers are mostly used for nonlinear systems.

For nonlinear systems there are two main possibilities to solve this problem. The first possibility is to use artificial neural nets for describing dynamical behaviour of non-linear systems [1], the second one is the design of fuzzy-models in the form of linguistical descriptions of the input-output behaviour of the system. The fuzzy-controller uses linguistical descriptions of process variables as well and therefore a unified description for the whole design process can be found.

When creating a fuzzy-model, one of the most difficult problems is to find the fuzzy-rules for representing the process behaviour. Because of the great complexity of the process it is not possible for one expert to formulate all rules of the model. Therefore approaches to use automatic knowledge acquisition methods have been searched. These approaches are the cluster-method [2], the multistep-method [3] and suggestions to use machine learning methods like ID3 by Quinlan [4,5].

This paper will show the possibility to use the ID3-algorithm for inductive learning of fuzzy-rules for dynamical systems.

2. Rule based modelling

The following form is presupposed to describe the structure of a non-linear dynamical model:

$$y(t)=f((y(t-1),y(t-2),.....,y(t-m),$$
$$u(t-1-k),......,u(t-n-k)). \tag{1}$$