ОПТИМИЗАЦИЯ ПРОМЫШЛЕННЫХ ПРОЦЕССОВ ПУТЕМ АДАПТИВНОГО ПРОГРАММИРОВАНИЯ

Я. КОЧИШ

Кафедра Автоматизации Будапештского политехнического университета

(Поступило в печать 5 сентября 1969 г.)

Представлено проф. Др. Ф. Чаки

За последние годы скачкообразно возрастает применение ЦВМ для решения различных задач управления промышленными процессами. ЦВМ на промышленном объекте может выполнять функцию «классических» регуляторов и, в таком случае, она заменяет систему обычных регуляторов, составленную в соответствии с технологическими требованиями данного промышленного процесса. Возможно также применение ЦВМ для реализации непрерывной оптимизации процесса. Подобная задача является несравнимо сложнее, особенно, если одна и та же ЦВМ комплексно управляет процессом и одновременно оптимизирует его на основе оптимизации некоторого данного технологического или экономического показателя.

Проблематика оптимизации разнообразными промышленными процессами уже долгое время привлекает внимание инженеров-исследователей, работающих в области автоматизации. И так же как уже удалось разработать алгоритмы для решения задач в детерминистических и стохастических системах — в которых статистические данные системы часто произвольно предполагаются неизвестными — уже много сделано в области оптимизации управляемых объектов типа «черного ящика», точнее говоря, — типа «серого ящика». На практике для оптимизации систем, в которых априорная информация содержит в себе необходимые знания о структуре системы, но входные и выходные параметры и даже их статистические характеристики заранее неизвестны, разработаны приемлемые статические методы оптимизации, в то время, как результаты, достигнутые в области динамической оптимизации промышленных процессов, встречают немалочисленные трудности, такие, как ограниченная оперативная память, ограниченное быстродействие ЦВМ, применяемых в настоящее время, условия устойчивости процесса оптимизации, устойчивость численных методов в цифровых системах и т. д.

В настоящей статье предлагается один из возможных вариантов обобщенного алгоритма динамической оптимизации, при помощи которого возможно решать задачу адаптивной оптимизации по функции цели в динамической многомерной нелинейной системе управления. Алгоритм видимо реализуется на управляющих цифровых вычислительных машинах и для его реализации требуется относительно небольшое количество априорной информации.

Постановка задачи

Режим объекта описывается следующим уравнением (см. рис. 1):

$$x = A\,[\mathbf{v}(\mathbf{u}, \mathbf{z})] + \xi, \tag{1}$$

где x — функция цели;
A — некоторый строго выпуклый оператор, связывающий входные параметры системы с функцией цели;
$\mathbf{u}$ — вектор управляющих параметров;
$\mathbf{z}$ — вектор измеряемых, но — из-за технологических условий объекта — не изменяемых параметров (при измерении может оказаться случайная аддитивная помеха с нулевым средним и конечной дисперсией);
$\mathbf{v}$ — вектор состояний системы;
ξ — помеха в измерении функции цели с нулевым средним и конечной дисперсией.

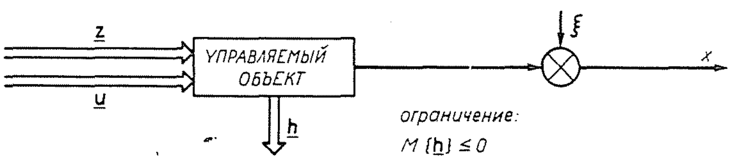

Рис. 1

Предположим, что функция цели измеряется непрерывно, но точная форма уравнения (1) нам неизвестна. Тогда задача состоит в том, чтобы найти вектор оптимальных управляющих параметров таким образом, чтобы функция цели стремилась динамически к экстремальному значению при заранее неизвестных изменениях помех и неизменяемых параметров при небольшом изменении динамических свойств объекта и, наконец, при соблюдении ограничений, наложенных на некоторые параметры объекта, типа неравенств:

$$M\left\{\mathbf{h}(x, \mathbf{u})\right\} \le 0 \tag{2}$$

(здесь и в последующих M обозначает математическое ожидание).

Алгоритм управления

Целью управления является то, чтобы функция цели x стремилась к экстремальному значению. Для этого при выборе квадратичного критерия качества управления типа

$$J_1(\mathbf{u}) = M\left\{x^2\right\} = \min_{\mathbf{u}} \tag{3}$$

необходимо выполнение следующего условия, полученного из функционала (3) путем дифференцирования по неизвестному параметру u:

$$\frac{dJ_1(\mathbf{u})}{d\mathbf{u}} = M\left\{\frac{d}{d\mathbf{u}}\,x^2\right\} = 2\,M\left\{x\,\frac{dx}{d\mathbf{u}}\right\} = 0\,. \tag{4}$$

Для решения такого типа задачи оптимизации известно много различных подходов, некоторые из них — в основном классический подход Понтрягина [1] и Беллмана [2] или варианты этих подходов для стохастических систем (см. например, [3]) — не приемлемы для решения нашей задачи, поскольку по постановке задачи нам неизвестны уравнения, характеризующие режим объекта, как это показано в [4].

Несравнимо больше возможностей обеспечивают относительно новые методы математического программирования, обработанные для анализа и синтеза систем автоматического управления, такие, как квадратичное программирование [5], метод наименьших квадратов [6], метод возможных направлений [7] и как частный случай этого подхода — метод симплекс [8] —, метод максимального правдоподобия [9], байесовский подход [10] и, наконец, метод стохастической апроксимации [11]. И хотя с применением каждого из перечисленных методов с большим или меньшим трудом возможно построить алгоритм решения задачи оптимизации, — именно метод стохастической апроксимации был избран для решения нашей задачи, так как при выборе такого подхода адаптивные алгоритмы получаются довольно просто и они легко поддаются программированию на ЦВМ, как это приведено в работе [4]. Предложенный алгоритм адаптивного программирования можно построить по следующим рассуждениям.

Градиент функции цели x по управляющему воздействию $\mathbf{u}$ в ур. (4) в такой форме нельзя определить, если по постановке задачи нам неизвестны статистические характеристики управляемого объекта. Для приближенного определения градиента предположим, что многомерные, в общем случае нелинейные, процессы в управляемом объекте с достаточной точностью могут быть охарактеризованы полной квадратичной формой относительно функции цели x. И тогда приближенная функция цели $\hat{x}$ будет иметь следующий вид:

$$\hat{x} = \mathbf{c}^{\mathrm{T}}\mathbf{y}\,[\hat{\mathbf{v}}(\mathbf{u}, \mathbf{z})], \tag{5}$$

где

$$\begin{aligned} \mathbf{y}(\hat{\mathbf{v}}) = \{1, v_1, v_2, \ldots, v_N, v_{N+1}, \ldots, v_L, \\ v_1^2, v_1 v_2, \ldots v_1 v_N, v_1 v_{N+1}, \ldots, v_1 v_L, \\ v_2^2, \ldots, v_2 v_N, v_2 v_{N+1}, \ldots, v_2 v_L, \\ \text{— — — — — — — — — — —} \\ v_N^2, v_N v_{N+1}, \ldots, v_N v_L, \\ v_{N+1}^2, \ldots, v_{N+1} v_L, \\ \text{— — — — —} \\ v_L^2\}, \end{aligned} \tag{6}$$

$$v_i[n] = \sum_{m=1}^{n} w_i[m]\, u_i[n-m] \qquad (i=1, 2, \ldots, N), \tag{7}$$

$$v_k[n] = \sum_{m=1}^{n} w_k[m]\, z_{k-N}[n-m] \quad (k=N+1, N+2, \ldots, L), \tag{8}$$

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix};\ \mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_M \end{bmatrix};\ \hat{\mathbf{v}} = \begin{bmatrix} 1 \\ v_1 \\ \vdots \\ v_L \end{bmatrix};\ \mathbf{c} = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_K \end{bmatrix};\ \mathbf{y} = \begin{bmatrix} 1 \\ y_1 \\ \vdots \\ y_K \end{bmatrix}, \tag{9}$$

$$L = M + N; \quad K = \frac{(L+1)(L+2)}{2} - 1. \tag{10}$$

Далее обозначение векторов и параметров системы следующие: $\hat{x}$ — приближенная функция цели, $\mathbf{u}$ — N-мерный вектор управляющих воздействий, $\mathbf{z}$ — M-мерный вектор неизменяемых параметров, $\hat{\mathbf{v}}$ — $(L+1)$-мерный, приближенный вектор состояний, $\mathbf{c}$ — $(K+1)$-мерный вектор неизвестных параметров, $\mathbf{y}$ — $(K+1)$-мерный, измеримый вектор входных параметров объекта.

Если такое приближенное представление функции цели имеет место, то градиент в уравнении регрессии (4) можно также приближать:

$$\frac{dx[n]}{d\mathbf{u}[n-1]} \cong \frac{d\hat{x}[n]}{d\mathbf{u}[n-1]} = \mathbf{D}(\mathbf{w}[1])\,\mathbf{C}[n-1]\,\mathbf{v}[n], \tag{11}$$

где обозначения матриц $\mathbf{D}$, $\mathbf{C}$ представлены в приложении I.

После замены действительного градиента на приближенный градиент (11), уравнение регрессии (4) можно решить методом стохастической апроксимации, предложенным в работе [4] на обобщенный случай адаптивного управления. Если цель адаптивного управления состоит в оптимизации функции цели x путем соответствующего изменения вектора управляющих воздействий $\mathbf{u}$, то приближенная процедура нахождения корня неизвестного уравнения регрессии (4) имеет следующий вид:

$$\mathbf{u}[n]=\mathbf{u}[n-1]-\mathbf{R}_1[n]x[n]\,\mathbf{D}(\mathbf{w}[1])\,\mathbf{C}[n-1]\,\mathbf{v}[n] \tag{12}$$

и процесс сходится с вероятностью, равной единице, к вектору $\mathbf{u}^*$, обеспечивающему минимум функционала (3) при выполнении условий сходимости стохастической апроксимации (см. например [4]). В алгоритме (12) коэффициент сходимости $\mathbf{R}_1$ в общем случае представляет собой заранее известную диагональную матрицу с размерами $N\times N$; известна нам также и диагональная матрица $\mathbf{D}$ $(N\times N)$, зависящая от компонентов постоянного вектора $\mathbf{w}$ [1]. Но алгоритм (12) следует дополнить: с одной стороны необходимо определить влияние ограничений (2), а с другой стороны путем идентификации определить элементы матрицы $\mathbf{C}$, составленной из компонентов вектора неизвестных параметров $\mathbf{c}$ (см. приложение I).

Учет ограничений

Предполагается, что ограничения, наложенные на процессы в объекте, представлены в следующей форме неравенств:

$$\mathbf{g}(\mathbf{u})=M\{\mathbf{h}(x,\mathbf{u})\}\leq 0, \tag{13}$$

где $\mathbf{g}$ представляет собой P-мерный вектор математического ожидания ограниченных параметров, а вектор $\mathbf{h}$ — также P-мерный, измеримый вектор ограниченных параметров. Предполагается, что вектор $\mathbf{h}$ удовлетворяет условию регулярности Слейтера. Тогда условие оптимальности можно определить с помощью теоремы Куна—Такера (см. в работе [12]). Цыпкиным [4] было доказано, что в таком случае множители Лагранжа можно интерпретировать как некоторые оценки влияния ограничений на оптимальное значение вектора неизвестных параметров (в нашей задаче на оптимальный вектор управляющих воздействий $\mathbf{u}^*$) и модифицированная приближенная процедура определения вектора управляющих воздействий $\mathbf{u}$ тогда имеет следующий вид:

$$\mathbf{u}[n]=\mathbf{u}[n-1]-\mathbf{R}_1[n]\left\{x[n]\,\frac{d\hat{x}[n]}{d\mathbf{u}[n-1]}+\mathbf{H}_{\mathrm{u}}\,(x[n],\mathbf{u}[n-1])\,\boldsymbol{\lambda}[n-1]\right\}, \tag{14}$$

где λ представляет собой P-мерный вектор модифицированных множителей Лагранжа, который определяется также рекуррентным соотношением [4]:

$$\boldsymbol{\lambda}[n]=\max\{0;\boldsymbol{\lambda}[n-1]+r[n]\,\mathbf{h}(x[n],\mathbf{u}[n-1])\} \tag{15}$$

$$(\boldsymbol{\lambda}[0]\geq 0).$$

Здесь введено следующее обозначение:

$$\mathbf{H}_u(x[n], \mathbf{u}[n-1]) = \left\| \frac{\partial h_i(x[n], \mathbf{u}[n-1])}{\partial u_j[n-1]} \right\| \quad (i=1, 2, \ldots, P;\ j=1, 2, \ldots, N). \tag{16}$$

Следующая задача состоит в определении элементов матрицы **C** путем идентификации.

Идентификация

Вектор неизвестных параметров **c**, присутствующих посредственно в алгоритме (12), как элементы матрицы **C** определяются с помощью идентификации соответственно принципу дуального управления [13]. Нижеприведенный подход был предложен также Цыпкиным [4] на обобщенный случай адаптивного управления. Для решения нашей конкретной задачи оптимизации такого объекта, режим которого с достаточной точностью можно охарактеризовать режимом квадратичной регрессионной модели объекта, возможно построить конкретные алгоритмы идентификации, реализуемые на универсальной ЦВМ. Целью идентификации является определение оптимального вектора неизвестных параметров $\mathbf{c}^*$ таким образом, чтобы — при выборе здесь также квадратичного критерия качества — следующий функционал достигал минимума:

$$J_2(\mathbf{c}) = M\left\{(x-\hat{x})^2\right\} = \min_{\mathbf{c}} \tag{17}$$

что и выполняется тогда, когда имеет место следующее соотношение, полученное из функционала (17) путем дифференцирования по вектору **c**:

$$\frac{dJ_2(\mathbf{c})}{d\mathbf{c}} = -2M\left\{(x-\hat{x})\frac{d\hat{x}}{d\mathbf{c}}\right\} = 0. \tag{18}$$

Как это показано в приложении 2, приближенный градиент в уравнении регрессии (18) можно представить в следующем виде:

$$\frac{d\hat{x}\ (\mathrm{n})}{d\mathbf{c}[n-1]} = \mathbf{y}[n] + \mathbf{S}[n-1]\,\mathbf{D}(\mathbf{w}[1])\,\mathbf{C}[n-1]\,\mathbf{v}[n], \tag{19}$$

где

$$\mathbf{S}[n-1] = \left\| \frac{du_j[n-1]}{dc_i[n-1]} \right\| = \|S_{ji}[n-1]\| \quad (j=1, 2, \ldots, N;\quad i=0, 1, \ldots, K) \tag{20}$$

матрица коэффициентов чувствительности с размерами $N \times (K+1)$. Для решения уравнения регрессии (18) воспользуемся регулярным беспоисковым, дискретным методом многомерной стохастической апроксимации. В отличие от решения обобщенного алгоритма идентификации, представленного в работе [4], здесь применяется конкретный вид приближенного градиента (19):

$$\mathbf{c}[n]=\mathbf{c}[n-1]+\mathbf{R}_2[n](x[n]-\hat{x}[n])\,\{\mathbf{y}[n]+\mathbf{S}[n-1]\,\mathbf{D}(\mathbf{w}[1])\,\mathbf{C}[n-1]\,\mathbf{v}[n]\}, \quad (21)$$

где коэффициент сходимости $\mathbf{R}_2$ в общем случае представляет собой заранее известную диагональную матрицу с размерами $(K+1)\times(K+1)$.

Для определения элементов матрицы чувствительности S получится также рекуррентное соотношение, если алгоритм управления (12) продифференцировать по компонентам c_j вектора неизвестных параметров $\mathbf{c}$:

$$\mathbf{s}_j[n]=\mathbf{s}_j[n-1]-\mathbf{R}_1[n]\,x[n]\,\mathbf{C}_j^*[n-1]\,\mathbf{D}(\mathbf{w}[1])\,\mathbf{s}_j[n-1] \quad (22)$$
$$(j=0,1,\ldots,K),$$

где значение матрицы $\mathbf{C}_j^*$ объясняется в приложении 3; далее, вектор $\mathbf{s}_j$ представляет собой $(N+1)$-мерный вектор-столбец:

$$\mathbf{s}_j^T=[1\;\; s_{1j}\;\; s_{2j}\;\; \ldots\;\; s_{Nj}] \qquad (j=0,1,\ldots,K). \quad (23)$$

Алгоритмы адаптивного программирования

В конечном итоге алгоритмы (14), (15), (21), (22) показывают алгоритмы адаптивного программирования:

$$\begin{aligned}\mathbf{u}[n]=\mathbf{u}[n-1]-\mathbf{R}_1[n]\,\{x[n]\,\mathbf{D}(\mathbf{w}[1])\,\mathbf{C}[n-1]\,\mathbf{v}[n]+\\ +\mathbf{H}_u(x[n],\mathbf{u}[n-1)\,\lambda[n-1]\},\end{aligned} \quad (14)$$

$$\lambda[n]=\max\{0;\;\lambda[n-1]+r[n]\,\mathbf{h}(x[n],\mathbf{u}[n-1])\}, \quad (15)$$

$$\begin{aligned}\mathbf{c}[n]=\mathbf{c}[n-1]+\mathbf{R}_2[n](x[n]-\hat{x}[n])\,\{\mathbf{y}[n]+\\ +\mathbf{S}[n-1]\,\mathbf{D}(\mathbf{w}[1])\,\mathbf{C}[n-1]\,\mathbf{v}[n]\},\end{aligned} \quad (21)$$

$$\mathbf{s}_j[n]=\mathbf{s}_j[n-1]-\mathbf{R}_1[n]\,x[n]\,\mathbf{C}_j^*[n-1]\,\mathbf{D}(\mathbf{w}[1])\,\mathbf{s}_j[n-1]\,. \quad (22)$$

На основе этих алгоритмов возможно построить принципальную схему адаптивного программирования, представленную на рис. 2., т. е. составить программу оптимизации объекта на ЦВМ, если нам уже известны приближенные передаточные функции отдельных каналов (матрица $\mathbf{D}(\mathbf{w})$, см. в приложении I).

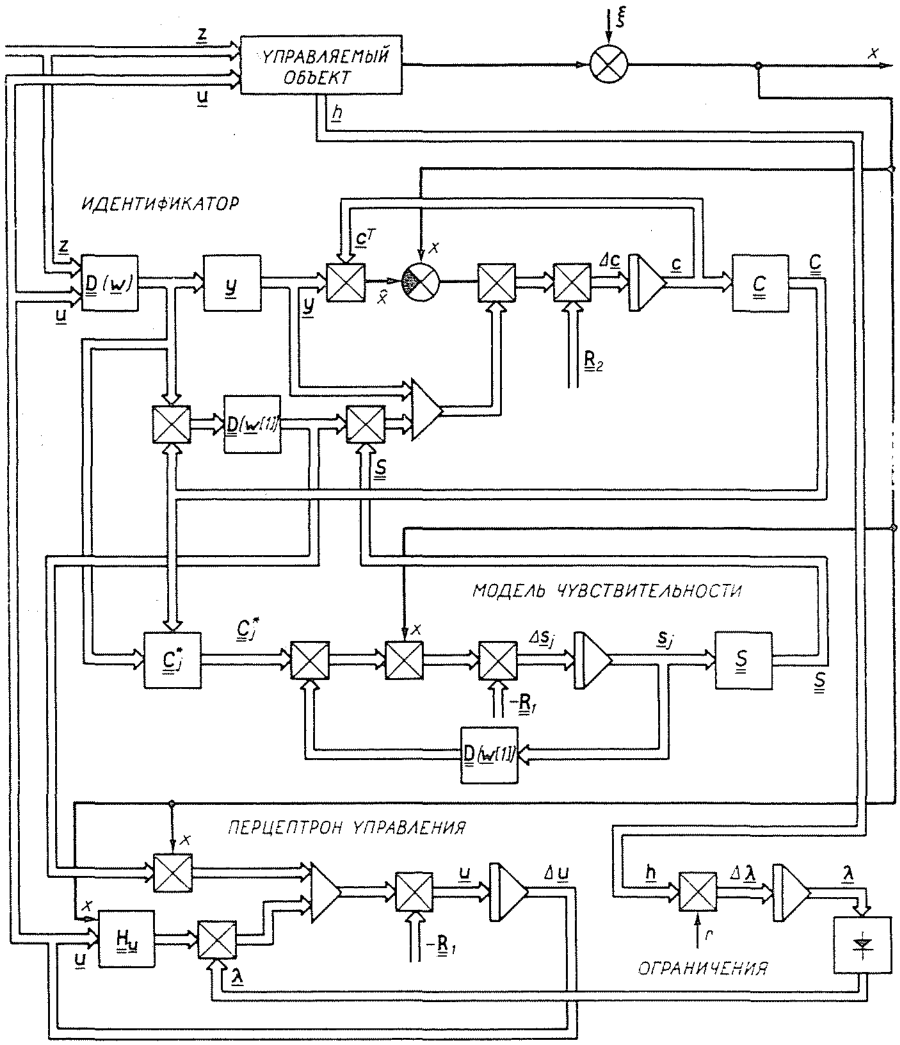

Рис. 2

Определение динамических свойств объекта требует предварительных исследований (см. например, [14]). Но подчеркивается, что при определенной динамике объекта вышеуказанный алгоритм является работоспособным, если даже динамические свойства объекта меняются во времени в небольшой степени внутри некоторых физически возможных границ.

Целесообразно отметить также важную роль выбора коэффициентов сходимости **R**. Ведь устойчивость самого процесса оптимизации зависит, в основном, от выбора этих коэффициентов. Эти коэффициенты могут быть определены оптимально (см. в работе [15]), если применить метод наименьших квадратов. Если коэффициенты сходимости избрать как гармоничный ряд (см. в работе [4]), то предварительно надо определить постоянные величины отдельных гармоничных рядов.

Выводы

В статье представлены алгоритмы адаптивного программирования. Предложенный подход применяется для оптимизации многомерных, динамических нелинейных объектов. Для необходимой априорной информации надо иметь понятие о динамических свойствах объекта, а в ходе оптимизации следует непрерывно измерять величины входных и выходных параметров управляемого объекта. Используя эти данные, на основе вышеуказанных алгоритмов в дискретных точках времени определяется вектор управляющих воздействий, направляющих функцию цели в сторону оптимальной рабочей точки.

Полученные алгоритмы имеют довольно простую конструкцию и их программирование на ЦВМ не представляет особые трудности для инженера, владеющего языком «алгол». Схема реализации алгоритмов представляет собой четверную перцептронную схему; перцептроны воспроизводят гибкую динамическую модель объекта, в ходе процесса обучения их параметры автоматически настраиваются на оптимальные значения, приближающие минимум квадратичных критериев наилучшим образом.

Приложение 1

Приближенная функция цели по (5)—(10) имеет следующий вид:

$$\begin{aligned}\hat{x} &= c_0+c_1 y_1+\ldots+c_K y_K = c_0+c_1 v_1+\ldots+c_K v_L^2 = \\ &= c_0+c_1 \sum w_1 u_1+\ldots+c_N \sum w_N u_N+c_{N+1} \sum w_{N+1} z_1+\ldots+ \\ &+c_L \sum w_L z_M+c_{L+1} \left(\sum w_1 u_1\right)^2+\ldots+c_K \left(\sum w_L z_M\right)^2.\end{aligned} \tag{24}$$

Продифференцируем обе части ур. (24) по вектору параметров u, пользуясь при этом следующим очевидным соотношением:

$$\begin{aligned}\frac{d}{du_i[n-1]}\left(\sum_{m=1}^{n} w_i[m]u_i[n-m]\right) &= \frac{d}{du_i[n-1]}(w_i[1]\,u_i[n-1]+ \\ &+ w_i[n]\,u_i[0]) = w_i[1]\end{aligned} \tag{25}$$

и тогда:

$$
\begin{aligned}
\frac{d\hat{x}[n]}{du_1[n-1]} &= c_1 w_1[1]+2c_{L+1} v_1 w_1[1]+\ldots+c_{2L} v_L w_1[1] = \\
&= w_1[1]\,(c_1+2c_{L+1} v_1+\ldots+c_{2L} v_L) \\
&\vdots \\
\frac{d\hat{x}[n]}{du_N[n-1]} &= w_N[1]\,\left(c_N+c_{L+N} v_1+\ldots+c_{\frac{(N+1)(2L-N+2)}{2}-1}\right)
\end{aligned} \tag{26}
$$

и если ввести следующую матрицу с размерами $(L+1)\times N$

$$
\mathbf{C} = \begin{bmatrix}
c_1 & 2c_{L+1} & c_{L+2} & \ldots & c_{L+N} & \ldots & c_{2L} \\
c_2 & c_{L+2} & 2c_{2L+1} & \ldots & c_{2L+N-1} & \ldots & c_{3L-1} \\
c_3 & c_{L+3} & c_{2L+2} & \ldots & c_{3L+N-1} & \ldots & c_{4L-3} \\
\hdashline
c_N & c_{L+N} & c_{2L+N-1} & \ldots & 2c_{\frac{N(2L-N+3)}{2}} & \ldots & c_{\frac{(N+1)(2L-N+2)}{2}-1}
\end{bmatrix}, \tag{27}
$$

то компоненты приближенного градиента (26) можно выразить в следующей векторной форме:

$$
\frac{d\hat{x}[n]}{d\mathbf{u}[n-1]} = \mathbf{D}\,(\mathbf{w}[1])\,\mathbf{C}[n-1]\,\mathbf{v}[n], \tag{11}
$$

где

$$
\mathbf{w}[1] = [w_1[1]\, w_2[1] \ldots w_N[1]]^T
$$

$$
\mathbf{D}(\mathbf{w}[1]) = \begin{bmatrix}
w_1[1] & 0 & \ldots & 0 \\
0 & w_2[1] & \ldots & 0 \\
\hdashline
0 & 0 & \ldots & w_N[1]
\end{bmatrix}. \tag{28}
$$

Приложение 2

Продифференцируем обе части уравнения (24) по вектору неизвестных параметров $\mathbf{c}$, пользуясь при этом следующим очевидным соотношением:

$$
\begin{aligned}
&\frac{d}{dc_j[n-1]}\left(c_j[n-1]\sum_{m=1}^{n} w_j[m]\,u_j[n-m]\right) = \sum_{m=1}^{n} w_j[m]\,u_j[n-m] + \\
&+ c_j[n-1]\cdot\frac{d}{dc_j[n-1]}\,(w_j[1]\,u_j[n-1]+\ldots+w_j[n]\,u_j[0]) = \\
&= v_{j}[n]+c_j[n-1]\,w_j[1]\,\frac{du_j[n-1]}{dc_j[n-1]} = v_j[n]+c_j[n-1]\,w_j[1]\,s_{jj}[n-1]
\end{aligned} \tag{29}
$$

и тогда

$$
\begin{aligned}
\frac{d\hat{x}[n]}{dc_0[n-1]} &= 1+s_{10}[n-1]\,w_1[1](c_1+2c_{L+1}\,v_1+\ldots+c_{2L}\,v_L)+\\
&+s_{20}[n-1]\,w_2[1](c_2+c_{L+2}\,v_2+\ldots+c_{3L-1}\,v_L)+\ldots+\\
&+\ldots+s_{N0}[n-1]\,w_N[1]\Big(c_N+c_{L+N}\,v_1+\ldots+c_{\frac{(N+1)(2L-N+2)}{2}-1}\,v_L\Big)\\
&\vdots \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (30)\\
\frac{d\hat{x}[n]}{dc_K[n]} &= y_K+s_{1K}(n-1]\,w_1[1](c_1+2c_{L+1}\,v_1+\ldots+c_{2L}\,v_L)+\\
&+\ldots+s_{NK}[n-1]\,w_N[1]\,(c_N+c_{L+N}\,v_1+\ldots).
\end{aligned}
$$

Пользуясь обозначениями, введенными в соотношении (11), градиент (30) также можно выразить в векторной форме:

$$
\frac{d\hat{x}[n]}{d\mathbf{c}[n-1]} = \mathbf{y}[n]+\mathbf{S}[n-1]\,\mathbf{D}(\mathbf{w}[1])\,\mathbf{C}[n-1]\,\mathbf{v}[n]. \qquad (19)
$$

Приложение 3

Если алгоритм управления (12) представим в скалярной форме:

$$
\begin{aligned}
&u_1[n]=u_1[n-1]-r_{11}[n]x[n]\,w_1[1](c_1+2c_{L+1}\,v_1+\ldots+c_{L+N}\,v_N+\ldots+c_{2L}\,v_L)\\
&\vdots \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (20)
\end{aligned}
$$

и продифференцируем обе части по вектору параметров $\mathbf{c}$, то получатся следующие алгоритмы:

$$
\begin{aligned}
s_{10}[n]&=s_{10}[n-1]-r_{11}[n]\,x[n]\,w_1[1](2c_{L+1}\,w_1[1]\,s_{10}[n-1]+\\
&\qquad+\ldots+w_N[1]\,s_{N0}[n-1])\\
s_{11}[n]&=s_{11}[n-1]-r_{11}[n]\,x[n]\,w_1[1](1+2c_{L+1}w_1[1]\,s_{11}[n-1]+\\
\vdots\quad&\qquad+\ldots+w_N[1]\,s_{N1}[n-1]) \qquad\qquad\qquad\qquad (31)\\
s_{NK}[n]&=s_{NK}[n-1]-r_{1N}[n]\,x[n]\,w_N[1](c_{L+N}\,w_1[1]\,s_{NK}[n-1]+\\
&\qquad+\ldots+w_N[1]\,s_{NK}[n-1]).
\end{aligned}
$$

Форма алгоритмов (31) сокращается, если ввести следующие символические обозначения:

$$
\mathbf{C}^*=\begin{bmatrix} 2c_{L+1} & c_{L+2} & \ldots & c_{L+N}\\ c_{L+2} & 2c_{2L+1} & \ldots & c_{2L+N-1}\\ \hline c_{L+N} & c_{2L+N-1} & \ldots & 2\,c_{\frac{N(2L-N+3)}{2}} \end{bmatrix}, \qquad (32)
$$

т. е. матрица $\mathbf{C}^*$ образуется из матрицы $\mathbf{C}$, она является квадратичной симметричной матрицей $N \times N$; далее

$$\mathbf{A}=\|\mathbf{a}_{ij}\|=\left[\begin{array}{l} 0\ 1\ 0\ 0\ \dots\ 0\ \ 0\ \dots\ 0\ \ 2v_1\ v_2\ \dots\ v_N\ \ v_{N+1}\ \dots\ v_L \\ 0\ 0\ 1\ 0\ \dots\ 0\ \ 0\ \dots\ 0\ \ 0\ \ v_1\ \dots\ 0\ \ 0\ \dots\ 0 \\ 0\ 0\ 0\ 1\ \dots\ 0\ \ 0\ \dots\ 0\ \ 0\ \ 0\ \dots\ 0\ \ 0\ \dots\ 0 \\ \hdashline \underbrace{0\ 0\ 0\ 0\ \dots\ 1}_{N}\ \underbrace{0\ \dots\ 0}_{M}\ \underbrace{0\ 0\ \dots\ v_1}_{N}\ \underbrace{0\ \dots\ 0}_{M} \end{array}\right.$$

$$\left.\begin{array}{l} 0\ 0\ \dots\ 0\ \ 0\ \dots\ 0\ \ 0\ \dots\ 0\ \ 0\ \dots\ 0\ \ 0\ \dots\ 0 \\ 2v_2\ v_3\ \dots\ v_N\ v_{N+1}\ \dots\ v_L\ \ 0\ \dots\ 0\ \ 0\ \dots\ 0\ \ 0\ \dots\ 0 \\ 0\ v_2\ \dots\ 0\ \ 0\ \dots\ 0\ \ 2v^3\ .\ 0\ \ 0\ \dots\ 0\ \ 0\ \dots\ 0 \\ \hdashline \underbrace{0\ 0\ \dots\ v_2}_{N-1}\ \underbrace{0\ \dots\ 0}_{M}\ \underbrace{0\ \dots 2v_N}_{\dots}\ \underbrace{v_{N+1}\dots\ v_L}_{M}\ \underbrace{0\ \dots\ 0}_{M(M+1)/2} \end{array}\right] \tag{33}$$

$$(i=1, 2, \dots, N;\ j=0, 1, \dots, K),$$

т. е. матрица с размерами $N \times (K+1)$, далее $\mathbf{a}_j$ обозначает j-тый вектор-столбец матрицы $\mathbf{A}$; тогда при помощи следующей гиперматрицы

$$\mathbf{C}_j^* = [\mathbf{a}_j \vdots \mathbf{C}^*] \tag{34}$$

получится в конечном итоге алгоритм модели чувствительности по (22)

Резюме

Предлагается один из возможных вариантов обобщенного алгоритма динамической оптимизации, при помощи которой возможно решать задачу адаптивной оптимизации по функции цели в динамической многомерной нелинейной системе управления. Алгоритм реализуется на управляющих цифровых вычислительных машинах и для его реализации требуется относительно небольшое количество априорной информации. Схема алгоритмов представляет собой четверную перцептронную схему; перцептроны воспроизводят гибкую динамическую модель объекта; в ходе обучения их параметры автоматически настраиваются на оптимальные значения.

Литература

1. Pontryagin, L.—Boltyanski, V.—Gamkrelidze, R.—Mishchenko, E.: The mathematical theory of optimal processes. John Wiley, 1962.
2. Bellman, R.: Dynamic programming. Princeton University Press, 1957.
3. Ho, Y. C.—Whalen, B.: An approach to the identification of linear dynamic systems with unknown coefficients. AIEE Trans. on Aut. Cont. **AC—8**, No. 3 (1963).
4. Цыпкин, Я. З.: Адаптация и обучение в автоматических системах. Изд. «Наука», 1968.
5. Юдин, Д. Б.: Методы количественного анализа сложных систем. Техническая кибернетика, н. I, 1965.

6. LEE, R. C. K.: Optimal estimation, identification and control. MIT Press, 1964.
7. ZOUTENDIJK, G.: Method of feasible direction. Elsevir, 1960.
8. SPENDLEY, W.—HEXT, G. R.—HIMSWORTH, F. R.: Sequential application of simplex designs in optimisation and evalutionary operation. Technometries **4,** 41 (1962).
9. SAKRISON, D. T.: Efficient recursive estimation, application to estimating the parameters of a covariance functions. Intern. J. of Eng. Sci. **22,** (1965).
10. SWORDER, D.: Optimal adaptive control systems. Academic Press, 1966.
11. ROBBINS, H.—MONRO, S.: A stochastic approximation method. Ann. of Math. Statistics **22,** No. 1 (1951).
12. KÜNZI, H. F.—KRELLE, W. V.: Nichtlineare Programmierung. Unter Mitw. v. Werner Oettli, 1961.
13. Фельдбаум, А. А.: Основы теории оптимальных автоматических систем. Изд. «Наука», 1966.
14. LEVIN, M. T.: Optimum estimation of impulse response in the presence of noise. IRE Trans. on Circuit Theory **CT—7,** No. 1. (1960).
15. ALBERT, A.—SITTLER, R. W.: A method for computing least squares estimators that keep up with the data. J. SIAM Control, Ser. A, **3,** 384—417 (1966).

Янош Кочиш, Будапешт XI, Эгри Йожеф 16, Венгрия