Если значения $\{u_i(0)\}_1^l$ различны и неизвестны, то вектор η определяется только с точностью до множителя. Для этого достаточно вычислить значение собственного вектора матрицы W^TW, соответствующего ее единственному нулевому собственному числу. Значение (с точ-

УДК 681.513.6

А. Н. БЕНДИЧ, Н. Н. БЕНДИЧ

(Иркутск)

АДАПТИВНЫЕ АЛГОРИТМЫ СТАБИЛИЗАЦИИ ОДНОГО КЛАССА И ИХ ПРАКТИЧЕСКОЕ ИСПОЛЬЗОВАНИЕ

В настоящей работе рассматриваются алгоритмы, позволяющие в режиме нормального функционирования по наблюдаемым входу и выходу подбирать такие параметры управляющего воздействия, которые обеспечивают экстремум заданного функционала. Предлагаемые алгоритмы применимы в задачах управления [1], сочетающих в себе методы идентификации и подстройки входных воздействий одновременно, в задачах слежения за объектом и сопротивления его, в задачах, связанных с изучением и регулированием биологических систем и т. п. Остановимся на задаче управления, все остальные сведутся к ней переосмысливанием пространств состояния, наблюдения и управления объектом.

Недоступность измерения некоторых параметров, их постоянные изменения, наличие некоторого дрейфа относительно оптимального режима работы создают предпосылки для создания системы управления, обеспечивающей постоянный поиск оптимального режима.

Пусть предполагаемый объект управления — сложный технологический объект, т. е. находится под воздействием различного рода стохастических возмущений, является многомерным, а его параметры связаны нелинейными соотношениями, которые, как правило, не могут быть выявлены. Предположим также, что качество работы на определенном этапе характеризуется стабильностью некоторых наблюдаемых параметров. Критерием управления предлагается рассматривать максимум вероятности того, что текущие значения на выходе не превысят нормативные показатели.

Считаем, что имеем дело с дискретным (или рассматриваемым в дискретные моменты времени) объектом, который можно определить совокупностью замкнутых множеств в евклидовых пространствах $\{X, Y, U\}$. Элементы этих множеств — соответственно значения векторов состояния $\mathbf{x}$, допустимых векторов управления $\mathbf{u}$ и выходных векторов $\mathbf{y}$.

Все множество Y разобьем на два (Y_1 и $Y_2 = Y \setminus Y_1$) таким образом, чтобы $\mathbf{y} \in Y_1$ являлся желаемым в каком-либо смысле. Тогда существует ненулевая условная вероятность

$$P(\mathbf{y}_{t+1} \in Y_1/\mathbf{y}_t,\ \mathbf{u}_t).$$

Таким образом, каждой паре $\mathbf{u}_t$, $\mathbf{y}_t$ ставим в соответствие функцию вероятности $P(\mathbf{y}_t,\ \mathbf{u}_t)\colon Y \cup U \to R^1$. Тогда оптимальное с точки зрения выбранного критерия управление на каждом шаге является функцией от наблюдаемого вектора $\mathbf{y}_t$, т. е. $\mathbf{u}^*(\mathbf{y}_t)$. Значит, без ограничения общности можно считать, что определяется управляющее воздействие, доставляющее безусловный максимум функции $P(\mathbf{u}(\mathbf{y}_t))\colon U \to R^1$. Следовательно, отыскание точки, в которой функция достигает условного экстремума, заменяется на отыскание безусловного экстремума, что существенно облегчает задачу. Пусть с достаточной точностью зависимость вектора управлений от вектора наблюдений может быть выражена в виде

$$\mathbf{u}_t^* = C\boldsymbol{\varphi}(\mathbf{y}_t). \tag{1}$$

Здесь $\boldsymbol{\varphi}(\mathbf{y}_t)$ — вектор из линейно независимых функций (в дальнейшем будем считать их ортонормированными полиномами); $C=\{c_{ij}\}-(m\times M)$-матрица неизвестных коэффициентов, которую требуется идентифицировать.

Рассмотрим характеристическую функцию

$$\Theta(\mathbf{y}_t, C)=\begin{cases}1, & \text{если} \quad \mathbf{y}_{t+1}\in Y_1;\\ 0, & \text{если} \quad \mathbf{y}_{t+1}\overline{\in} Y_1.\end{cases} \tag{2}$$

Тогда выбранный критерий качества стабилизации можно переписать таким образом:

$$I(C)=\max_{\mathbf{u}} P(\mathbf{u}(\mathbf{y}_t))=\max_{C} M_{\mathbf{y}_t}\{\Theta(\mathbf{y}_t, C)\}. \tag{3}$$

Такая запись позволяет обойтись без измерений неизвестной функции вероятности. Предлагается алгоритм для работы в реальном масштабе времени на действующем объекте. В этом случае коррекция будет происходить через каждые $2m$ шагов:

$$\mathbf{u}_t^{2s+2m}=\mathbf{u}_t^{2s}+\frac{\gamma_s}{2a_s}\left[\boldsymbol{\Theta}_+\left(\mathbf{y}_t^{2s},\mathbf{u}_t^{2s},a_s\right)-\boldsymbol{\Theta}_-\left(\mathbf{y}_t^{2s},\mathbf{u}_t^{2s},a_s\right)\right]\boldsymbol{\varphi}^T\left(\mathbf{y}_t^{2s}\right)\boldsymbol{\varphi}(\mathbf{y}_t). \tag{4}$$

Здесь $\mathbf{u}_t^0$ — начальное приближение; $\{\gamma_s\}$, $\{a_s\}$ — последовательности положительных чисел, регулирующие длину шагов адаптации;

$$\boldsymbol{\Theta}_+(\cdot)=\Big(\Theta\big(\mathbf{y}_t^{2s},\mathbf{u}^{2s}(\mathbf{y}_t^{2s})+a_s\mathbf{e}_1\big),\ldots,\Theta\big(\mathbf{y}_t^{2s+2(i-1)},\mathbf{u}^{2s}\big(\mathbf{y}_t^{2s+2(i-1)}\big)+a_s\mathbf{e}_i\big),\ldots$$
$$\ldots,\Theta\big(\mathbf{y}_t^{2s+2(m-1)},\mathbf{u}^{2s}\big(\mathbf{y}_t^{2s+2(m-1)}\big)+a_s\mathbf{e}_m\big)\Big);$$
$$\boldsymbol{\Theta}_-(\cdot)=\Big(\Theta\big(\mathbf{y}_t^{2s+1},\mathbf{u}^{2s}(\mathbf{y}_t^{2s+1})-a_s\mathbf{e}_1\big),\ldots,\Theta\big(\mathbf{y}_t^{2s+2i-1},\mathbf{u}^{2s}\big(\mathbf{y}_t^{2s+2i-1}\big)-a_s\mathbf{e}_i\big),\ldots$$
$$\ldots,\Theta\big(\mathbf{y}_t^{2s+2m-1},\mathbf{u}^{2s}\big(\mathbf{y}_t^{2s+2m-1}\big)-a_s\mathbf{e}_m\big)\Big),$$

где $\{\mathbf{e}_i\}$ — набор векторов, образующих базис в пространстве E_m. Алгоритм сходится почти наверное, если функция $P(\mathbf{u}(\mathbf{y}_t))$ непрерывна и всюду дифференцируема по $\mathbf{u}$. Если же $P(\mathbf{u}(\mathbf{y}_t))$ не является дифференцируемой и имеет разрывы, то удобно рассмотреть алгоритм, полученный из метода минимизации широкого класса функций вида $P(\mathbf{u}_t)=M\Theta(\mathbf{u}_t)$ [2], где $\Theta(\mathbf{u}_t)$ удовлетворяет лишь локальному условию Липшица. Алгоритм (4) можно представить в виде детерминированной и стохастической составляющих:

$$\mathbf{u}_t^{2s+2m}=\mathbf{u}_t^{2s}+\gamma_s\mathbf{R}_s\left(\mathbf{y}_t^{2s},\mathbf{u}_t^{2s},\mathbf{y}_t\right)+\gamma_s\boldsymbol{\xi}^s\left(\mathbf{y}_t^{2s},\mathbf{u}_t^{2s},\mathbf{y}_t\right). \tag{5}$$

Здесь $\boldsymbol{\xi}^s\left(\mathbf{y}_t^{2s},\mathbf{u}_t^{2s},\mathbf{y}_t\right)$ выступает в роли ошибки измерения $P\left(\mathbf{u}_t^{2s}\left(\mathbf{y}_t^{2s}\right)\right)$, причем $M\boldsymbol{\xi}^s=0$, а $\mathbf{R}_s(\cdot)=\frac{1}{2a_s}\{\mathbf{P}_+\left(\mathbf{y}_t,\ \mathbf{u}_t^{2s},\ a_s\right)-\mathbf{P}_-\left(\mathbf{y}_t^{2s},\ \mathbf{u}_t^{2s},\ a_s\right)\}\times$ $\times\boldsymbol{\varphi}\left(\mathbf{y}_t^{2s}\right)\boldsymbol{\varphi}(\mathbf{y}_t)$ — детерминированная составляющая процесса адаптации.

Распределение ошибки трудно аппроксимировать приближенно для получения оптимального алгоритма, однако необходимо обеспечить малую чувствительность решений к априорной неопределенности и к принятым допущениям. Так, в [3] предлагается для многомерных алгоритмов при отыскании безусловного экстремума дифференцируемой функции в случае наличия случайных помех в вычислении ее градиента для процедуры вида (5) вводить дополнительное преобразование $\Psi: R^m\to R^m$:

$$\mathbf{u}_t^{2s+2m}=\mathbf{u}_t^{2s}+\gamma_s\boldsymbol{\Psi}\left[\left(\nabla P\left(\mathbf{u}_t^{2s}\right)+\boldsymbol{\xi}^s\right)\boldsymbol{\varphi}^T\left(\mathbf{y}_t^{2s}\right)\boldsymbol{\varphi}(\mathbf{y}_t)\right],$$

Здесь $\nabla P(\cdot)$ — градиент $P(\cdot)$; $\nabla P(\cdot)+\xi^s$ — приближенное значение градиента, определяемое как

$$\frac{1}{2a_s}\left[\Theta_+\left(\mathbf{y}_t^{2s}, \mathbf{u}_t^{2s}, a_s\right)-\Theta_-\left(\mathbf{y}_t^{2s}, \mathbf{u}_t^{2s}, a_s\right)\right].$$

Вид $\Psi(\cdot)$ зависит от множества факторов, а в нашем случае свойство «робастности» алгоритмов будет обеспечено введением ограничений, налагаемых на аргумент $\Psi(\cdot)$ и вычисляемых эмпирически в каждом отдельном случае [4].

Рассмотрим теперь условия, влияющие на сходимость и устойчивость алгоритмов в связи с полиномиальным представлением управляющего воздействия. Чебышевская система функций позволяет провести выбор порядка полиномов l и получить единственное разложение $\mathbf{u}_t=C\boldsymbol{\varphi}(\mathbf{y}_t)$. Казалось бы, при увеличении порядка полиномов увеличивается точность представления $\mathbf{u}(\mathbf{y}_t)$, хотя за эту точность и приходится платить возрастанием объема используемой машинной памяти и времени, но оказывается, что если на вход объекта действует аддитивная помеха $\mathbf{v}$, то для каждой координаты v_j, $j=\overline{1,\ m}$, условия робастности выразятся в виде [4]

$$v_j \leqslant T\sin^2[\pi/2(l+2)], \tag{6}$$

где $T=\min\limits_i T_i$; T_i — интервал изменения y_i.

Далее, для ускорения скорости сходимости алгоритмов следует использовать априорную информацию об объекте, о помехах и возможном виде аппроксимируемой функции. Так, последовательность $\{\gamma_j\}$ будем выбирать, считая, что $\gamma_j=1/Aj$, где коэффициент A оценивается максимальной дисперсией полиномов разложения $\boldsymbol{\varphi}(\mathbf{y}_t)$. Отмечается, что для непрерывной максимизируемой функции при обычном выборе $a_j=1/Bj^\nu$ лучший результат получается при $\nu=1/4$, если скорость сходимости оценивается среднеквадратическим критерием $M\|\mathbf{u}_t^s-\mathbf{u}_t^*\|^2$. Параметр B определяется величиной первых пробных шагов a_0.

На первых шагах отыскания оптимального управления целесообразно ввести несколько шагов с рандомизированной стратегией управления, содержащей случайный механизм выбора. Можно также осуществить несколько шагов случайного поиска для «нащупывания» достаточно широкой окрестности экстремума, это должно некоторым образом восполнить недостаток априорной информации.

Начальные значения коэффициентов разложения c_{ij}^0 полезно выбирать, привлекая к этому простейшее описание из возможного вида функции $\underline{\mathbf{u}_t^*=C\boldsymbol{\varphi}(\mathbf{y}_t)}$, в крайнем случае можно положить $c_{ij}^0=0$ для всех $i=\overline{1,\ m}$, $j=\overline{1,\ M}$. Если же управляемый объект не удовлетворяет требованиям стационарности и имеется небольшой дрейф параметров или воздействий, тогда весь отрезок управления делится на участки, на которых дрейф пренебрежительно мал, и на каждом таком участке идентификация проводится заново, при этом используется результат, полученный на предыдущем участке в качестве начального приближения c_{ij}^0.

Работоспособность алгоритмов была проверена на модели нелинейного процесса, соответствующего некоторому химическому производству. В ходе экспериментов выяснилось, что немаловажным вопросом, относящимся к выбору параметров, является вопрос о соотношении между длиной пробного шага a_s и величиной ε, характеризующей ширину желаемого интервала стабилизации. Однако если значения ε последовательно уменьшать в процессе работы, удается добиться попадания в сколь угодно малую окрестность заданной траектории, дисперсия выхода уменьшается при этом на порядок. При оценке вероятности попадания выхода объекта в заданную область она увеличилась с 0,59 до 0,74.

ЛИТЕРАТУРА

1. **Бендич Н. Н.** Об одном алгоритме адаптивной стабилизации // Автометрия.— 1980.— № 3.
2. **Бендич Н. Н., Кондратьев В. В.** Один подход к максимизации функции вероятности в задачах стабилизации // Приближенные методы решений операторных уравнений и их приложения.— Иркутск: СО АН СССР, 1982.
3. **Поляк Б. Т., Цыпкин Я. З.** Робастные псевдоградиентные алгоритмы адаптации // АиТ.— 1980.— № 10.
4. Бендич Н. Н. Робастная процедура определения стабилизирующей обратной свя-

УДК 519.68 : 007.52

О. А. БАШКИРОВ, Ю. Г. ВАСИН, С. Б. РУДОМЕТОВА

(Горький)

МАТЕМАТИЧЕСКИЕ МОДЕЛИ СТРУКТУРИРОВАННОГО ОПИСАНИЯ ГРАФИЧЕСКИХ ИЗОБРАЖЕНИЙ

В работе рассматриваются математические модели структурированного описания сложных графических изображений типа карт, чертежей, схем и т. п. Предложены контурная и линейно-контурная модели описания, рассмотрены алгоритмы построения моделей, способы представления их в памяти ЭВМ и возможности применения для решения задач обработки и распознавания.

Основные понятия. Бинарным изображением будем называть прямоугольную матрицу (растр) $\{a_{i,j}\}$, где

$$a_{i,j} = 0,\ 1;\quad 1 \leqslant i \leqslant n;\quad 1 \leqslant j \leqslant n.$$

Для записи растра используем более экономичную форму: «штрихи» или «хорды» [1]. Штрих есть тройка $(x_н, x_к, y)$, где

$$(x_н = j_1) \wedge \overline{a}_{i,j_1-1} \wedge a_{i,j_1};\quad (x_к = j_2) \wedge a_{i,j_2} \wedge \overline{a}_{i,j_2+1};\quad y = i,$$

т. е. штрих задает последовательность единичных («черных») клеток в строке растра; $x_н$ — номер первой клетки последовательности; $x_к$ — номер последней клетки.

Определим контур связной фигуры в штриховой форме [1] следующим образом. Рассмотрим всевозможные варианты расположения штрихов в двух соседних строках растра и найдем для каждого из них предыдущую и последующую точки контура так, чтобы «черное» оставалось справа. Все варианты сводятся к восьми, изображенным на рис. 1. Последовательность предыдущих и последующих точек контура образует контур, всегда замкнутый, без пересечений и разветвлений, что следует из определения. Контур может быть внешним (упорядочен по часовой стрелке) и внутренним (против часовой стрелки).

Под математической моделью изображения будем понимать тройку $M = (E, R, O)$, где E — множество непроизводных элементов $\{e_i\}$; R — множество отношений между непроизводными элементами $\{r_j\}$; O — множество объектов модели $\{o_к\}$.

Контурная модель изображения. Пусть $К = \{x_1, y_1, x_2, y_2, \ldots, x_n, y_n\}$ — последовательность точек контура связной фигуры. Тогда контурная модель изображения $КМ = (E_1, R_1, O_1)$ [2, 3] состоит из непроизвод-