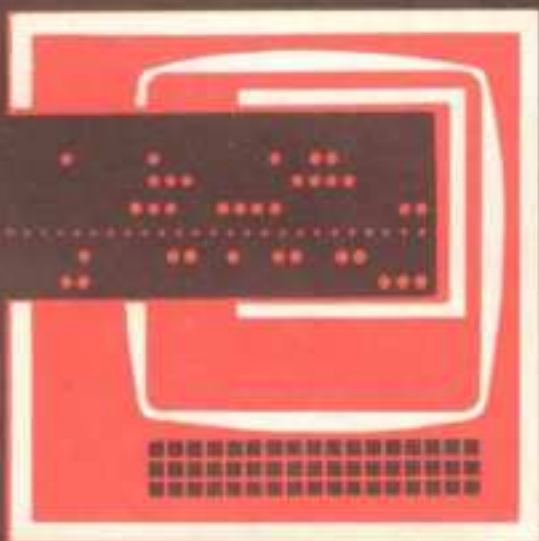


АВТОМАТИЗИ- РОВАННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ И ПРИБОРЫ АВТОМАТИКИ

79|86



681.5(06)
А 22

МИНИСТЕРСТВО ВЫСШЕГО И СРЕДНЕГО
СПЕЦИАЛЬНОГО ОБРАЗОВАНИЯ УССР
ХАРЬКОВСКИЙ ОРДЕНА ТРУДОВОГО КРАСНОГО ЗНАМЕНИ
ИНСТИТУТ РАДИОЭЛЕКТРОНИКИ имени М. К. ЯНГЕЛЯ

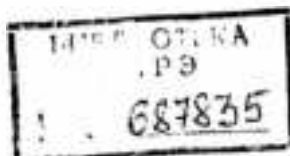
АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ И ПРИБОРЫ АВТОМАТИКИ

Республиканский
межведомственный
научно-технический
сборник

Основан в 1965 г.

В Ы П У С К 79

КС



ХАРЬКОВ
ИЗДАТЕЛЬСТВО ПРИ ХАРЬКОВСКОМ
ГОСУДАРСТВЕННОМ УНИВЕРСИТЕТЕ
ИЗДАТЕЛЬСКОГО ОБЪЕДИНЕНИЯ
«ВИЩА ШКОЛА»
1980



Автоматизированные системы управления и приборы автоматизации: Рес. междувед. науч.-техн. сб. — Х.: Вища шк. Изд-во при Харьк. ун-те, 1986. — Вып. 79. 137 с.

В сборнике представлены материалы по моделированию, идентификации и оптимизации технологических процессов сложных объектов управления. Рассмотрены вопросы математического и программного обеспечения для рационального построения вычислительных и информационно-измерительных систем, алгоритмического обеспечения подсистем автоматизированного проектирования вычислительных структур приборов и средств автоматизации. Изложены результаты исследований в области гибких автоматизированных производств.

Нормативные материалы приведены по состоянию на 1 января 1986 г.

Для научных работников и специалистов.

Редакционная коллегия: Е. П. Пуятин (отв. ред.), В. М. Шостак (зам. отв. ред.), Н. А. Лысенков (отв. секр.), В. И. Васильев, Э. А. Дедиков, В. В. Дубровский, А. Г. Евдокимов, И. В. Кузьмин, С. Ф. Капалал, К. Д. Конопаленко, Е. Н. Лазаренко, В. И. Мелешко, В. М. Михайленко, И. Ф. Огороднейчук, Э. Г. Петров, Л. А. Расстригин, В. В. Свиридов, Ю. Г. Стоян, Н. М. Шнырев

Адрес редакционной коллегии: 310141, Харьков-141, пр. Ленина, 14, институт радиоэлектроники, кафедра применения электронно-вычислительных машин, тел. 40-94-59

Редакция научно-технической литературы

А. Г. ЕВДОКИМОВ, д-р техн. наук, Н. В. ФЕДОРОВ,
канд. техн. наук, С. И. КОЗЫРЕНКО

**ЭФФЕКТИВНЫЙ АЛГОРИТМ ИДЕНТИФИКАЦИИ
СОСТОЯНИЯ МОДЕЛИ УСТАНОВИВШЕГОСЯ
ПОТОКОРАСПРЕДЕЛЕНИЯ В ИНЖЕНЕРНЫХ СЕТЯХ**

При решении различных вопросов, связанных с управлением потоко-распределением в инженерных сетях и их реконструкцией, возникает задача идентификации состояния модели установившегося потоко-распределения (УПР). По замерам давлений и расходов целевого продукта в некоторых узлах сети требуется оценить их значения, а также значения давлений и расходов по остальным участкам инженерной сети. Формальная постановка задачи приводит к необходимости решать задачу нелинейного математического программирования при ограничениях в виде равенств. В статье описан алгоритм, учитывающий специфические свойства, присущие инженерным сетям. Это делает его проще, а программы более быстросействующими по сравнению с использовавшимися ранее [1].

Постановка задачи. Пусть структура модели сети представлена в виде графа $G(V, E)$, где V, E — множества вершин и дуг графа сети. Множество E состоит из трех подмножеств: фиктивных дуг, соответствующих входам сети L и выходам сети N , реальных дуг, соответствующих участкам трубопровода M . Параметры всех реальных дуг графа сети считаем известными. Любая вершина графа — либо вход, либо выход сети. Выберем дерево графа сети. В соответствии с первым и вторым постулатами сетей [2] математическую модель УПР можно представить так:

$$h_r + \sum_{i \in E_1} b_{r,i} h_i = 0, \quad r \in E_2; \quad (1)$$

$$q_i = \sum_{r \in E_1} b_{r,i} q_r, \quad i \in E_1. \quad (2)$$

Здесь E_1, E_2 — множества индексов дуг графа сети, относящихся к ветвям дерева и хордам; q_i, h_i — расход и потеря напора в i -м участке; $b_{r,i}$ — элемент цикломатической матрицы;

$$h_j = c_j |q_j|^n \operatorname{sign} q_j + h_j^{(r)}, \quad j \in M; \quad (3)$$

$$h_j = -P_j, j \in L; \quad (4)$$

$$h_j = P_j, j \in N. \quad (5)$$

где c_j — гидравлическое сопротивление j -го реального участка; α — коэффициент нелинейности; P_j — давление на выходе ($j \in N$) или входе ($j \in L$) j -й фиктивной дуги; $h_j^{(0)}$ — потеря напора, определяемая разностью геодезических высот конца и начала j -го участка.

Пусть в ряде фиктивных дуг, соответствующих входам и выходам сети, измеряются значения давлений и расходов $\bar{P}_i, \bar{q}_i, i \in L \cup N$:

$$\bar{P}_i = P_i + \epsilon_i, i \in L^p \cup N^p; \quad (6)$$

$$\bar{q}_i = q_i + \zeta_i, i \in L^q \cup N^q. \quad (7)$$

В формулах (6), (7) L^p, N^p — множества индексов фиктивных дуг, соответствующих входам и выходам сети, где измерялись давления; L^q, N^q — множества индексов фиктивных дуг, соответствующих входам и выходам сети, где измерялись расходы; ϵ_i, ζ_i — ошибки измерения давлений и расходов. Полагая, что ϵ_i, ζ_i распределены по нормальному закону и нам известны значения дисперсий $\sigma_{P_i}^2, \sigma_{q_i}^2$. Необходимо дать оценки значениям $P_i, q_i, i \in L \cup N$, а также определить $q_i, h_i, i \in M$. В дальнейшем для конкретности (это не меняет общности алгоритма) будем рассматривать задачу идентификации состояния модели УПР водопроводных сетей, для которых зависимость $h_i(q_i), i \in M$ имеет вид

$$h_i = c_i |q_i| q_i + h_i^{(0)}, i \in M. \quad (8)$$

Чтобы получить искомые оценки, воспользуемся методом максимального правдоподобия. Тогда функция максимального правдоподобия

$$\begin{aligned} L = (2\pi)^{\frac{l_1 + n_1 + l_2 + n_2}{2}} \prod_{i \in L^p \cup N^p} \sigma_{P_i}^{-1} \prod_{i \in L^q \cup N^q} \sigma_{q_i}^{-1} \times \\ \times \exp \left[-\frac{1}{2} \sum_{i \in L^p \cup N^p} \frac{1}{\sigma_{P_i}^2} (P_i - \bar{P}_i)^2 \right] \times \\ \times \exp \left[-\frac{1}{2} \sum_{i \in L^q \cup N^q} \frac{1}{\sigma_{q_i}^2} (q_i - \bar{q}_i)^2 \right], \end{aligned} \quad (9)$$

где l_1, l_2, n_1, n_2 — мощность множества L^p, L^q, N^p, N^q соответственно. Максимум функции (9) достигается в случае, если

$$y = \sum_{i \in L^p \cup N^p} \frac{1}{\sigma_{P_i}^2} (P_i - \bar{P}_i)^2 + \sum_{i \in L^q \cup N^q} \frac{1}{\sigma_{q_i}^2} (q_i - \bar{q}_i)^2 \rightarrow \min_{q_i, P_i \in \mathbb{R}}. \quad (10)$$

Отметим, что Ω — область, определяемая системой уравнений (1) — (5).

Итак, задача идентификации состояния модели УПР в рассматриваемой постановке представляет собой задачу минимизации нелинейной функции цели (10) при ограничениях в виде системы равенств (1) — (5).

Алгоритм решения. Выберем дерево графа сети таким образом, чтобы его ветвями стали все фиктивные дуги, соответствующие входам и выходам сети. Тогда все дуги, соответствующие реальным участкам сети, станут хордами. Система (1), (2) с учетом зависимостей (8), (4), (5) примет вид

$$c_r | q_r | q_r + h_r^{(r)} + \sum_{i \in N} b_{1,r} P_i - \sum_{i \in L} b_{1,r} P_i = 0, \quad r \in M; \quad (11)$$

$$q_i = \sum_{r \in M} b_{1,r} q_r, \quad i \in L \cup N. \quad (12)$$

Следовательно, задача идентификации состояния модели УПР является задачей минимизации функции цели (10) при ограничениях (11), (12).

Пусть m , l , n — мощности множеств M , L , N . Ограничения (11), (12) представляют собой систему $m+l+n$ уравнений относительно $m+2l+2n$ переменных q_i , $i \in M \cup L \cup N$; P_j , $j \in L \cup N$. Разделим переменные задачи на зависимые и независимые. К независимым отнесем давления в дугах, соответствующих входам и выходам сети P_j , $j \in L \cup N$, к зависимым — расходы q_i , $i \in M \cup L \cup N$. Предложенное кодирование сети и способ разбиения на зависимые и независимые переменные позволили в отличие от подхода, описанного в работе [1], подключить аналитически первые и вторые условные производные минимизируемой функции (10) и фактически свести задачу минимизации с нелинейными ограничениями в виде равенств к задаче безусловной минимизации. При выбранном таким способом дереве цикломатическая матрица B_1 содержит в каждой своей строке лишь два отличных от нуля элемента, что позволяет отказаться от построения такой матрицы и значительно сократить объем используемой памяти для хранения необходимой информации.

Рассмотрим аналитические выражения для определения условных производных

$$\begin{aligned} & \frac{\partial y}{\partial P_j}, \quad j \in L \cup N; \quad \frac{\partial^2 y}{\partial P_j \partial P_i}, \quad i, j \in L \cup N; \\ & \frac{\partial y}{\partial P_j} = \frac{2}{\sigma_{P_j}^2} (P_j - \bar{P}_j) + 2 \sum_{\substack{k \in L \cup N \\ j \in L^p \cup N^p}} \frac{1}{\sigma_{q_k}^2} (q_k - \bar{q}_k) \frac{\partial q_k}{\partial P_j}, \end{aligned} \quad (13)$$

Если в узле, соответствующем началу или концу j -й дуги, не измерялись давления, то

$$\frac{\partial y}{\partial P_i} = 2 \sum_{k \in L^q \cup N^q} \frac{1}{\sigma_{q_k}^2} (q_k - \tilde{q}_k) \frac{\partial q_k}{\partial P_i}, \quad i \in (L \cup N) / (L^p \cup N^p). \quad (14)$$

Для определения

$$\frac{\partial q_k}{\partial P_i}, \quad k \in L^q \cup N^q, \quad i \in L \cup N$$

используем зависимость (12):

$$\frac{\partial q_k}{\partial P_i} = \sum_{r \in M} b_{1,r} \frac{\partial q_r}{\partial P_i}, \quad i \in L \cup N, \quad k \in L^q \cup N^q. \quad (15)$$

Выражение для $\frac{\partial q_r}{\partial P_i}$, $r \in M$, $i \in L \cup N$ получаем в результате дифференцирования зависимости (11), представляя q_r , $r \in M$ как неявную функцию от P_i , $i \in L \cup N$:

$$\frac{\partial q_r}{\partial P_i} = \frac{\partial_{1,r}}{\partial c_r |q_r|}, \quad i \in L; \quad (16)$$

$$\frac{\partial q_r}{\partial P_i} = -\frac{\partial_{1,r}}{\partial c_r |q_r|}, \quad i \in N. \quad (17)$$

Определим элементы матрицы S вторых условных производных функции цели по независимым переменным

$$\frac{\partial^2 y}{\partial P_i \partial P_j}, \quad i, j \in L \cup N.$$

Диагональные элементы матрицы S для случая, когда измеряется давление в узле, соответствующем началу или концу j -й дуги, вычисляются так:

$$\begin{aligned} \frac{\partial^2 y}{\partial P_j^2} = & \frac{2}{\sigma_{P_j}^2} + 2 \sum_{k \in L^q \cup N^q} \frac{1}{\sigma_{q_k}^2} \left[\left(\frac{\partial q_k}{\partial P_j} \right)^2 + \right. \\ & \left. + (q_k - \tilde{q}_k) \frac{\partial^2 q_k}{\partial P_j^2} \right], \quad i \in L^p \cup N^p. \end{aligned} \quad (18)$$

В противном случае, а также для недиагональных элементов выражение для $\frac{\partial^2 y}{\partial P_i \partial P_j}$ имеет вид

$$\frac{\partial^2 y}{\partial P_i \partial P_j} = 2 \sum_{k \in L^q \cup N^q} \frac{1}{\sigma_{q_k}^2} \left[\left(\frac{\partial q_k}{\partial P_i} \right) \left(\frac{\partial q_k}{\partial P_j} \right) + (q_k - \tilde{q}_k) \frac{\partial^2 q_k}{\partial P_i \partial P_j} \right], \quad (19)$$

$$i, j \in L \cup N.$$

Если $i = j$, то $i \in L^p \cup N^p$. Значения

$$\frac{\partial q_k}{\partial P_i}, \quad \frac{\partial^2 q_k}{\partial P_i^2}, \quad i, j \in L \cup N$$

определяем из формулы (15). Выражение для $\frac{\partial^2 q_k}{\partial P_i \partial P_j}$, $k \in L^0 \cup N^0$, $j \in L \cup N$ получаем в результате дифференцирования зависимости (12):

$$\frac{\partial^2 q_k}{\partial P_i \partial P_j} = \sum_{r \in M} b_{1rk} \frac{\partial^2 q_r}{\partial P_i \partial P_j}, \quad k \in L^0 \cup N^0, \quad j \in L \cup N. \quad (20)$$

Используя зависимость (11) и правила нахождения вторых производных неявных функций, находим

$$\frac{\partial^2 q_i}{\partial P_i \partial P_j} = \pm \frac{b_{1ri} b_{1rj}}{4c_i^2 q_i^3}, \quad r \in M; \quad i, j \in L \cup N. \quad (21)$$

Соотношения между элементами множеств L , N определяют знак в выражении (21): « $\leftarrow$ » соответствует $i \in L$, $j \in L$; $i \in N$, $j \in N$; « $\rightarrow$ » — $j \in N$, $i \in L$, $j \in L$, $i \in N$.

В качестве метода решения рассматриваемой задачи использовались модифицированный обобщенный метод Ньютона, основанный на приведении матрицы Гесса к диагональному виду [3], и метод покоординатного спуска. Основное рекуррентное соотношение для обобщенного метода Ньютона имеет такой вид:

$$\bar{P}^{(k+1)} = \bar{P}^{(k)} + \lambda^{(k)} \Delta \bar{P}^{(k)}, \quad (22)$$

где $\Delta \bar{P}^{(k)}$ — направляющий вектор на k -м шаге (определяет направление движения); $\lambda^{(k)}$ — параметр, характеризующий значение шага в выбранном направлении; $\bar{P}^{(k+1)}$ — вектор значений независимых переменных на $(k+1)$ -м шаге. Направляющий вектор на k -м шаге определяется так:

$$\Delta \bar{P}^{(k)} = - (S^{-1})^{(k)} \left(\frac{\partial y}{\partial \bar{P}} \right)^{(k)}. \quad (23)$$

В формуле (23) $\left(\frac{\partial y}{\partial \bar{P}} \right)^{(k)}$ — вектор первых условных производных.

Используя приведение матрицы вторых условных производных к диагональному виду, можно найти минимум функции цели, когда она не является выпуклой. В этом случае возможно установить направление убывания функции при нахождении в окрестности точки максимума или седловой точки. Кроме того, этот метод позволяет определить направление убывания функции, если матрица S особенная. Параметр $\lambda^{(k)}$, регулирующий значение шага на k -й итерации, вычислялся при минимизации $y(\lambda^{(k)})$ методом дихотомии. Одномерная оптимизация производилась до тех пор, пока не выполнялось условие

$$\max_{j \in L \cup N} (P_j^{k,r+1} - P_j^{k,r}) < \epsilon, \quad (24)$$

где $P_i^{k,r-2}$, $P_i^{k,r-1}$ — значения верхней и нижней границ интервала неопределенности для переменной P_i на k -й итерации после r этапов одномерной оптимизации. Погрешность ϵ выбирается из технологических соображений.

Метод покоординатного спуска используется в модификации, когда основное рекуррентное соотношение для k -го этапа j -го шага имеет вид

$$P_i^{(k+1)} = P_i^{(k)} - \lambda^{(k)} \left(\frac{\partial y}{\partial P_i} \right)^{(k)}, \quad i \in L \cup N. \quad (25)$$

Значение $\lambda^{(k)}$ выбирается так же, как и при использовании метода Ньютона, с той лишь разницей, что

$$P_i^{k,r-2} - P_i^{k,r-1} < \epsilon \quad (26)$$

для рассматриваемой в данный момент времени переменной P_i .

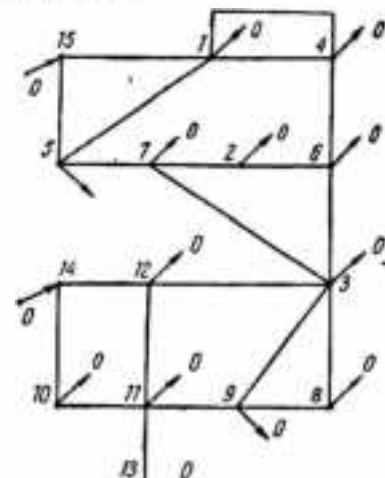
Минимизация функции y производилась до тех пор, пока не выполнялось одно из следующих соотношений:

$$\max_{i \in L \cup N} \left| \frac{\partial y}{\partial P_i} \right| < \delta \quad (27)$$

или

$$\max_{i \in L \cup N} |\Delta P_i| > \epsilon, \quad (28)$$

где δ — некоторая заданная точность. Кроме того, матрица вторых частных условных производных S должна быть положительно определена.



Граф сети

Узел	Давление		Расход	
	намерен- ный	расчет- ный	намерен- ный	расчет- ный
1	69,80	69,35	0,1577	0,1579
2	66,96	67,04	0,0990	0,0991
3	65,92	67,20	0,1099	0,1100
4	67,77	68,09	0,1181	0,1182
5	70,76	69,96	0,1401	0,1402
6	66,72	67,05	0,1198	0,1200
7	66,79	66,24	0,1394	0,1396
8	66,48	66,69	0,1509	0,1511
9	68,15	66,90	0,1186	0,1187
10	69,22	69,00	0,1190	0,1191
11	68,29	68,24	0,0997	0,0997
12	71,80	71,49	0,1421	0,1422
13	57,53	57,21	0,1489	0,1488
14	76,66	77,29	0,9263	0,9185
15	71,60	72,22	0,7522	0,7463

Пример. Задача идентификации состояния модели УПР решена для водопроводной сети, граф которой представлен на рисунке, при измеренных

значениях давлений и расходов на всех входах и выходах. Значения исходных данных и результаты расчета приведены в таблице. Расчеты показали, что время решения задачи идентификации состояния модели УПР методом Ньютона значительно меньше, чем методом покоординатного спуска.

Список литературы: 1. Евдокимов А. Г., Теляшев А. Д. Оперативное управление потокораспределением в инженерных сетях.—Х.: Выща шк., 1980.—144 с. 2. Евдокимов А. Г. Оптимальные задачи на инженерных сетях.—Х.: Выща шк., 1976.—152 с. 3. Евдокимов А. Г. Минимизация функций.—Х.: Выща шк., 1977.—159 с.

Поступила в редколлегию 05.04.85.

УДК 519.6

С. Ю. ИВАНОВ, И. В. СОЛОДОВНИКОВ, канд. техн. наук

**ПРОЦЕДУРЫ ОПТИМАЛЬНОГО ПРОЕКТИРОВАНИЯ
СЛОЖНЫХ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ
ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ**

Рассмотрим методы оптимального проектирования сложных систем с использованием имитационного моделирования. Для решения задач проектирования в систему имитации должна быть включена процедура оптимизации, позволяющая находить рациональные значения параметров. Отметим, что эффективные процедуры для имитационного моделирования практически отсутствуют. Поэтому возникает необходимость в анализе существующих процедур оптимизации и в разработке более эффективных, предназначенных для использования с имитационными моделями. Можно выделить два подхода для выбора оптимального проектного решения, применяющих имитационное моделирование. Первый основан на методах перебора результатов имитационного эксперимента, второй использует экстремальный эксперимент [1, 2]. Процедуры, использующие перебор, предполагают возможность выделения для каждого независимого параметра или фактора ряда уровней, а также применения полных или неполных факторных планов эксперимента. Оптимальным считается такое решение, при котором реакция модели принимает наилучшее значение с точки зрения проектировщика.

Оптимальное решение определяется следующим образом. Пусть задано n прогонов имитационной модели. Каждый из них характеризуется значением вектора независимых параметров или факторов

$$x_i = (x_i^1, \dots, x_i^r), \quad 1 \leq i \leq n$$

и значением вектора реакции

$$y_i = (y_i^1, \dots, y_i^t),$$

где m — число факторов; r — число составляющих вектора реакции. Для выбора оптимального решения на множестве R'

ведем отношение предпочтения] и безразличия между любыми значениями $y_1, y_2 \in R^r$ такое, что всегда можно утверждать, что реализуется один из трех вариантов [3]:

$$y_1 \succ y_2; y_2 \succ y_1; y_2 \sim y_1.$$

Обозначим через Y множество всех реакций, полученных в ходе имитационного эксперимента $y_i \in Y$. Тогда в качестве оптимального можно рассматривать такое значение $y^* \in Y$, для которого не существует $y' \in Y$, причем $y' \succ y^*$. В общем случае таких y^* может быть несколько. Очевидно, что полученному y^* соответствует $x^* \in X$, где X — множество всех значений вектора независимых параметров, использованных в эксперименте,

$$X = \{x_1, x_2, \dots, x_n\}$$

Решения x^* и соответствующие им значения y^* можно называть оптимальными только условно, т. е. в том смысле, что они оптимальные среди значений, полученных в эксперименте. Вероятно, что среди $y \in Y$ можно найти такое y^0 , что $y^0 \succ y^*$. Проверить значения y_i для всех допустимых x_i , как правило, возможно в силу конечности прогонов в эксперименте и непрерывности в общем случае отдельных составляющих вектора независимых параметров. Говоря об оптимальном планировании эксперимента, следует подразумевать оптимизацию функции качества эксперимента, не совпадающую с целевыми функциями проектируемой системы.

Существуют методы планирования экстремальных экспериментов. Большинство из них основано на предположении о существовании функции регрессии и возможности отыскания экстремума этой функции. Функцию регрессии можно найти с помощью имитационного моделирования. Последнее определяет алгоритмы планирования таких экспериментов. Данные алгоритмы являются итерационными [2].

Нахождение функции регрессии заключается в построении с помощью имитационного моделирования эмпирической модели исследуемой системы, которая выражается приближенными зависимостями между независимыми параметрами и реакциями системы. Если проведено n прогонов модели $X = \{x_i\}$, $Y = \{y_i\}$, строится регрессионная или эмпирическая модель

$$y_i = F(X_i, B) + E,$$

где B — неизвестный вектор параметров регрессии; E — вектор ошибок. Эту формулу можно записать так:

$$y_i^j = F(x_1^i, x_2^i, \dots, x_m^i; \beta_{0j}, \beta_{1j}, \dots, \beta_{mj}) + E_{ij}.$$

Здесь y_i^j — g -я составляющая вектора реакции в i -м прогоне; x_i^j — j -я составляющая вектора входных параметров в i -м прогоне.

Процесс построения эмпирической модели можно представить в виде следующей совокупности этапов [1].

1. Выбор математической формы для соотношения между входными и выходными параметрами.
2. Вычисление параметров регрессии.
3. Оценка качества аппроксимации экспериментальных параметров.

На первом этапе выбирается функциональное соотношение между факторами и реакциями проектируемой системы. Первоначально можно выбрать наиболее простое выражение, например линейную модель вида [4]

$$y_k = \beta_0 + \sum_{i=1}^m x_i \beta_{ki} + E_k. \quad (1)$$

Если на такой простой модели не удастся получить удовлетворительных результатов, эмпирическую модель усложняют. Следующий этап заключается в вычислении параметров регрессии. Для линейной модели можно, например, воспользоваться методом наименьших квадратов. Итогом вычисления параметров регрессии является

$$\hat{y}_k = F(x_1, \dots, x_m; b_{k0}, b_{k1}, \dots, b_{kr}), \quad (2)$$

где $b_{k0}, b_{k1}, \dots, b_{kr}$ — оценка параметров регрессии.

Третий этап состоит в оценке качества аппроксимации. Он представляет собой проверку того, насколько каждый параметр регрессии значим в выражении (2). Если параметр не значим, его вклад в вычисление реакции ничтожно мал. Может оказаться, что предлагаемая эмпирическая модель недостаточно точно аппроксимирует имитационный эксперимент. Тогда необходимо увеличить точность эмпирической модели, например перейти от линейной модели первого порядка (1) к линейной квадратичной [5]

$$y_k = \beta_{k0} + \sum_{i=1}^m \beta_{ki} x_i + \sum_{i=1}^m \sum_{j=1}^m \beta_{kij} x_i x_j + E_k.$$

Пусть в результате рассмотрения итерационной процедуры получена эмпирическая модель удовлетворительной точности. Эту модель можно использовать для нахождения оптимального решения классическими методами оптимизации. Считаем также, что при решении оптимизационной задачи найден вектор независимых переменных $\hat{x}^* = (\hat{x}_1^*, \dots, \hat{x}_m^*)$, для которого выполняется условие

$$\hat{y}_k^* = \text{opt } y_k = F(\hat{x}_1^*, \dots, \hat{x}_m^*; b_{k0}, b_{k1}, \dots, b_{kr}).$$

Поскольку $b_{k0}, \dots, b_{kr}$ — оценки, решения и $\hat{y}_k^*$ приближенны. Осушествив прогон имитационной модели со значениями входных параметров $\hat{x}^* = (\hat{x}_1^*, \dots, \hat{x}_m^*)$. В результате получим значение выходного параметра, которое обозначим через $\hat{\hat{y}}_k$.

Сравним y_k^* и $\tilde{y}_k$, оценим качество решения. Пусть качество удовлетворительно. Тогда будем считать, что полученная аппроксимация эмпирической моделью хорошая и найденное решение оптимально. Если качество неудовлетворительно, точность эмпирической модели недостаточна либо в силу неверной оценки параметров регрессии, либо из-за неудачного выбора ее вида. Тогда результаты проверочного прогона присоединением к n пробным прогонам в качестве $(n+1)$ -го, с учетом измененного эксперимента корректируем эмпирическую модель, т. е. повторяем итерационную процедуру с учетом $(n+1)$ -го прогона. Итеративный процесс продолжается до получения удовлетворительных результатов.

Использование итерационной процедуры с применением эмпирической и имитационной моделей позволяет осуществлять направленный поиск оптимального решения. Обоснованность решения можно проверить на имитационной модели. Предлагаемая процедура позволяет строить и уточнять функциональную зависимость между входными и выходными параметрами, подготавливая почву для теоретического описания системы. Рассматриваемую процедуру можно обобщить как вариант экстремального эксперимента, под которым понимается эксперимент, направленный на отыскание экстремума функции регрессии [5, 2]:

$$y = F(x) + E(x),$$

где $E(x)$ — ошибка, причем $M[E(x)] = 0$. Последнее условие верно, если модель откалибрована. Величина Y должна вычисляться и измеряться. В рассматриваемой ситуации величина Y определяется в результате прогона имитационной модели.

Задача планирования экстремального эксперимента состоит в конструировании последовательности точек $x_0, x_1, \dots, x_n \in X$, $n \in 0, 1, \dots$, сходящихся к точке $x^* \in X$, в которой $F(x^*) = \sup_{x \in X} F(x)$, или такой, что $F(x_n) \rightarrow \sup_{x \in X} F(x)$.

Рассмотренная ранее процедура относится к разряду регулярных экстремальных экспериментов, для которых можно определить градиент функции $F(x)$. Однако не всегда можно доверять эмпирической модели. В силу ярко выраженной нелинейности сложной системы каждому небольшому диапазону значений входных параметров соответствует своя эмпирическая модель. В этом случае применять регулярный экстремальный эксперимент нецелесообразно. Для такой ситуации используют поисковые экстремальные эксперименты. На n -м шаге случайным или детерминированным образом выбирается l векторов n в точках

$$x_{n+1,1} = x_n + a_{n+1}q_{n1}; \quad x_{n+1,2} = x_n + a_{n+1}q_{n2}; \quad \dots,$$

$$x_{n+1,l} = x_n + a_{n+1}q_{nl}$$

с помощью имитационного эксперимента определяются значения $y_{gi}^{n+1}, \dots, y_{gi}^{n+1}$. Среди последних выбирается наилучшее:

$$y_{gi}^{n+1} = \text{opt} (y_{gi}^{n+1}, \dots, y_{gi}^{n+1}).$$

Если $y_{gi}^{n+1} | y_{gi}^n$, эксперимент завершается, в противном случае продолжается. В качестве x_{n+1} выбирается $x_{n,i} + \alpha_{n,i} q_{n,i}$, соответствующее y_{gi}^{n+1} .

В качестве модификации поискового алгоритма можно воспользоваться его комбинацией с неполным факторным экспериментом. Пусть неполный факторный эксперимент состоит из n прогонов с векторами входных параметров $x(1), \dots, x(m)$, где $x(i)$ — вектор входных параметров для i -го прогона. Считаем, что в результате моделирования получены $y_g(1), \dots, y_g(m)$, где $y_g(i)$ — значение g -го выходного параметра модели для ее i -го прогона. Определим оптимальное значение среди $y_g(1), \dots, y_g(m)$. Пусть это будет $y_g(j)$. Для каждого $x(i)$, $1 \leq i \leq m$ построим l векторов

$$x(i) + \alpha q_r, \quad 1 \leq r \leq l.$$

С помощью имитационного моделирования найдем

$$y_{gr}(i) = y_g(x(i) + \alpha q_r), \quad 1 \leq r \leq l.$$

Если для любых

$$y_{gr}(i) | y_g(i), \quad y_g(i) | y_g(j),$$

то данное направление эксперимента исключается, т.е. в дальнейшем вектор $x(i)$ изменяться не будет.

Если для каких-то $r_1, \dots, r_k$, $1 \leq k \leq l$

$$y_{gr_d}(i) | y_g(i), \quad 1 \leq d \leq k,$$

то выбирается оптимальное среди $y_{gr_d}^{(i)}$ и следующий шаг будет производиться из точки $x(i) + \alpha_{gr_d}$. Кроме того, среди всех значений $y_{gr_d}(i) | y_g(i)$ и $y_g(j)$ выбирается оптимальное в качестве нового $y_g(j)$. Таким образом, число векторов m после каждого шага остается прежним или сокращается, т.е. в итоге приходим к обычному варианту поискового алгоритма. Значения q_r , α можно задавать, исходя из тех же соображений, которые используются в задачах планирования эксперимента, например $q_r = e_r$, $\alpha = 1, 1$, где e_r — орты. Выбор q_r может задаваться и такими дополнительными условиями, как необходимость сохранять значение вектора входных параметров $x(i)$ внутри некоторого диапазона.

Таким образом, использование имитационного моделирования для поиска оптимальных решений при проектировании сложных систем требует специальным образом организованного эксперимента, а именно, экстремального эксперимента, регулярного в случае систем, для которых в результате моделирования можно

построить эмпирическую модель, и поискового в противном случае.

Имитационная модель служит при поиске оптимума следующим целям. С ее помощью удается определять значения выходных параметров в зависимости от входных, для которых неизвестна функциональная зависимость; контролировать ход эксперимента и проверять соответствие предполагаемого оптимального решения истинному; строить эмпирическую модель проектируемой системы. Если можно построить эмпирическую модель, то она послужит для теоретического осмысления вопросов, связанных с проектируемой системой. Имитационная модель является тем средством, с помощью которого можно осуществить эксперимент, направленный на поиск наилучшего проектного решения для несуществующей системы. Использование имитационного моделирования для построения оптимальных систем одного класса объектов с каждым разом будет давать более точные результаты, поскольку каждую реальную систему можно использовать для калибровки модели, а следовательно, ошибка измерений, проводимых на имитационной модели, должна уменьшаться.

Таким образом, имитационное моделирование улучшает качество проектирования систем и имитационной модели.

Список литературы: 1. Феррари Д. Оценка производительности вычислительных систем.— М.: Мир, 1981.— 125 с. 2. Математическая теория планирования эксперимента / Под ред. С. М. Ермакова.— М.: Наука, 1983.— 178 с. 3. Исследование операций. Методологические основы и математические методы / Под ред. Дж. Моудера, С. Замаграби.— М.: Мир, 1981.— 216 с. 4. Браверман Э. М., Мучник И. Б. Структурные методы обработки эмпирических данных.— М.: Наука, 1983.— 279 с. 5. Адлер Ю. П., Грамовский Ю. В. Планирование эксперимента при поиске оптимальных условий.— М.: Наука, 1976.— 125 с.

Поступила в редакцию 12.01.85.

УДК 62-50

Е. В. БОДЯНСКИЙ, канд. техн. наук

АВТОМАТИЧЕСКИЙ СИНТЕЗ МОДЕЛЕЙ В САПР АДАПТИВНЫХ СИСТЕМ

Задача разработки САПР адаптивных систем управления имеет свою специфику, связанную с высоким уровнем априорной неопределенности, нестационарностью объектов, разнообразием структур адаптивных регуляторов и моделей. Составной частью математического обеспечения САПР является библиотека алгоритмов идентификации, при этом конкретный алгоритм выбирается проектировщиком в зависимости от специфики объекта [1, 2].

В статье предложен алгоритм автоматического синтеза «обобщенной модели», использующий множество математических мо-

делей объекта, которые получены с помощью различных идентификационных процедур.

Пусть объект управления описывается уравнением псевдолинейной регрессии

$$y_n = x_n^T \varphi^* + \xi_n,$$

где y_n — выход объекта в дискретный момент времени n ; x_n — $(m \times 1)$ -обобщенный вектор входов; φ^* — вектор неизвестных коэффициентов, подлежащих определению; ξ_n — помеха на выходе типа дискретного белого шума. Если библиотека алгоритмов идентификации содержит q программных модулей, то модель, полученная с помощью i -го алгоритма, имеет вид

$$\hat{y}_{i,n} = \varphi_{i,n-1}^T x_n, \quad i = \overline{1, q},$$

а блок из q моделей —

$$y_n = \Phi_{n-1}^T x_n.$$

Здесь $\hat{y}_n = (\hat{y}_{1,n}, \hat{y}_{2,n}, \dots, \hat{y}_{q,n})^T$; Φ_{n-1} — матрица коэффициентов отдельных моделей,

$$\Phi_{n-1} = (\varphi_{1,n-1}, \varphi_{2,n-1}, \dots, \varphi_{q,n-1})^T.$$

Введем в рассмотрение «обобщенную модель»

$$\hat{y}_n = \hat{C}_{n-1}^T \hat{y}_n = C_{n-1}^T \Phi_{n-1}^T x_n,$$

где C_{n-1} — $(q \times 1)$ -вектор коэффициентов «обобщения» на $(n-1)$ -й итерации, удовлетворяющий условию несмещенности выхода «обобщенной модели»

$$C_{n-1}^T I = 1.$$

Отметим, что I — $(q \times 1)$ -вектор, состоящий из единиц.

Коэффициенты «обобщения» можно определять в результате оптимизации лагранжиана

$$J_n = \|Y_n - \hat{Y}_n C_n\|^2 + \lambda (C_n^T I - 1), \quad (1)$$

в котором Y_n — $(n \times 1)$ -вектор выхода объекта; $\hat{Y}_n$ — матрица выходов отдельных моделей; λ — неопределенный множитель Лагранжа. Оптимизация выполняется с помощью процедуры Эрроу — Гурвица [3]

$$C_n = C_{n-1} + \alpha \frac{2v_n^2 - \lambda_{n-1} (1 - C_{n-1}^T I)}{\|2v_n y_n - \lambda_{n-1} I\|^2} (2v_n y_n - \lambda_{n-1} I); \quad (2)$$

$$\lambda_n = \lambda_{n-1} + \gamma_n^\lambda (C_{n-1}^T I - 1);$$

$$v_n = y_n - \hat{y}_n, \quad 0 < \alpha < 2,$$

либо с помощью алгоритма [4]

$$C_n = C_{n-1} + \frac{P_{n-1} y_n (y_n - \hat{y}_n^T C_{n-1} + \hat{y}_n^T P_{n-1} \lambda_{n-1} I)}{\hat{p} + \hat{y}_n^T P_{n-1} \hat{y}_n}$$

$$- \beta^{-1} P_{n-1} \lambda_{n-1} I; \\ P_n = \beta^{-1} \left(P_{n-1} - \frac{P_{n-1} \tilde{y}_n \tilde{y}_n^T P_{n-1}}{\beta + \tilde{y}_n^T P_{n-1} \tilde{y}_n} \right); \quad (3)$$

$$\lambda_n = \lambda_{n-1} + \gamma_n^\lambda (C_n^T I - 1), \quad 0 < \beta < 1,$$

где γ_n^λ — параметр шага поиска. Характер процесса настройки вектора коэффициентов «обобщения» в значительной мере определяется выбором параметра γ_n^λ , оптимальное значение которого вычислить невозможно. Этот факт в определенной мере затрудняет использование алгоритмов (2), (3).

Отметим, что решение системы нормальных уравнений

$$\nabla_C J_n = Y_n Y_n - Y_n^T Y_n C_n + \lambda I = 0;$$

$$\frac{\partial J_n}{\partial \lambda} = C_n^T I - 1 = 0$$

приводит к оценке

$$C_n = (\hat{Y}_n^T \hat{Y}_n)^{-1} \left(\hat{Y}_n^T Y_n + \frac{1 - Y_n^T \hat{Y}_n (\hat{Y}_n^T \hat{Y}_n)^{-1} I}{I^T (\hat{Y}_n^T \hat{Y}_n)^{-1} I} I \right) = \\ = \tilde{C}_n + P_n \frac{1 - \tilde{C}_n^T I}{I^T P_n I} I, \quad (4)$$

в которой параметры $\tilde{C}_n$, P_n находятся с помощью стандартного рекуррентного метода наименьших квадратов:

$$\tilde{C}_n = \tilde{C}_{n-1} + P_n (y_n - \tilde{C}_{n-1}^T \hat{y}_n) \hat{y}_n;$$

$$P_n = P_{n-1} - \frac{P_{n-1} \hat{y}_n \hat{y}_n^T P_{n-1}}{1 + \hat{y}_n^T P_{n-1} \hat{y}_n}.$$

Таким образом, алгоритм (4) выгодно отличается от алгоритмов (2), (3) отсутствием контура настройки неопределенного множителя Лагранжа.

Учитывая тот факт, что

$$Y_n - \hat{Y}_n C_n = Y_n I^T C_n - \hat{Y}_n C_n = (Y_n I^T - \hat{Y}_n) C_n = E_n C_n,$$

критерий (1) записываем в форме

$$J_n = C_n^T E_n^T E_n C_n + \lambda (C_n^T I - 1) = C_n^T R_n C_n + \lambda (C_n^T I - 1),$$

где R_n — ковариационная матрица ошибок отдельных моделей. Решая систему нормальных уравнений

$$\nabla_C J_n = R_n C_n + \lambda I = 0;$$

$$\frac{\partial J_n}{\partial \lambda} = C_n^T I - 1 = 0,$$

получаем

$$C_n = \frac{R_n^{-1}I}{I^T R_n^{-1}I}. \quad (5)$$

Выражения (4), (5) эквивалентны, а лагранжиан (1) принимает значение

$$J_n = (I^T R_n^{-1}I)^{-1}.$$

Представим матрицу R_n в виде $R_n = T_n \Lambda_n T_n^T$ [5], где Λ_n — $(q \times q)$ -диагональная матрица; $\lambda_{i,n} > 0$; $T_n^T = T_n^{-1}$. Тогда $R_n^{-1} = T_n \Lambda_n^{-1} T_n^T$, и для любого диагонального элемента матрицы R_n справедливо соотношение $r_{ii,n} = \sum_j \lambda_{j,n} t_{ji,n}^2 > \lambda_{i,n} t_{ii,n}^2$, $i, j = \overline{1, q}$. Введем в рас-

смотрение вектор-столбец $W_n = T_n \Lambda_n^{-1/2} I$, элементы которого удовлетворяют соотношению $W_{i,n} = \sum_j \lambda_{j,n}^{-1/2} t_{ji,n}$. Для квадрата евклидовой нормы этого вектора можно записать выражение

$$\|W_n\|^2 = I^T R_n^{-1} I = \sum_i \left(\sum_j \lambda_{j,n}^{-1/2} t_{ji,n} \right)^2 > t_{ii,n}^2 \lambda_{i,n}^{-1}.$$

Тогда

$$\|W_n\|^{-2} = (I^T R_n^{-1} I)^{-1} = J_n < t_{ii,n}^2 \lambda_{i,n} < r_{ii,n}.$$

Таким образом, «обобщенная модель» обеспечивает предсказание выхода объекта, по точности не уступающее любой из отдельных моделей. В результате использования предложенных процедур можно избежать субъективизма пользователя при выборе конкретного метода, а также существенно повысить качество решения задачи идентификации.

Список литературы: 1. Деревицкий Д. П., Фрадков А. Л. Прикладная теория адаптивных систем управления. — М.: Наука, 1981. — 219 с. 2. Деревицкий Д. П. Функциональная структура САПР адаптивных систем. — В кн.: Тез. докл. Всесоюз. конф. «Теория адаптивных систем и ее применение». М.; Л.: 1983, с. 318—320. 3. Плисс Н., Бодянский Е., Маджаров Н. Адаптивно прогнозирование нестационарных процессов. — Автоматика и вычислительная техника, 1983, № 6, с. 5—12. 4. Бодянский Е. В., Плисс Н. П. Адаптивное обобщенное прогнозирование нестационарных процессов. — Х., 1984. — 12 с. — Рук. деп. в УкрНИИНТИ, № 159 Укр-Д84. Деп. 5. Себер Дж. Линейный регрессионный анализ. — М.: Мир, 1980. — 456 с.

Поступила в редколлегию 12.07.84.

В. Л. КУРИЛИН, М. В. ТРИГУБ
АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ СИНТЕЗА
ПРИБЛИЖЕННО-ОПТИМАЛЬНОГО УПРАВЛЕНИЯ
ДЛЯ НЕЛИНЕЙНОГО ОБЪЕКТА

В работах [1], [2] предложен метод синтеза приближенно-оптимального управления для объекта, движение которого описывается нелинейным дифференциальным уравнением в векторной форме:

$$\dot{x}(t) = \varphi(x) + D u, \quad x(0) = x_0 \in G_0 \subset E_n, \quad (1)$$

где $x(t)$ — вектор-функция состояния объекта, определенная на полуоси $[0, \infty)$, $x(t) = (x_1, \dots, x_n) \in E_n$; $u(t)$ — вектор-функция управления, принадлежащая классу кусочно-непрерывных функций при $t \in [0, \infty)$, $u(t) = (u_1, \dots, u_m) \in E_m$; $D = \|d_{ij}^0\|$ — $(n \times m)$ -матрица. Компоненты вектор-функции $u(t)$ стеснены условиями

$$|u_s(t)| < u_s^*, \quad s = \overline{1, m}. \quad (2)$$

Среди допустимых вектор-функций $u(x)$, удовлетворяющих ограничениям (2), требуется найти такую $u(x)$, которая для любого вектора x_0 из области начальных возмущений G_0 , содержащей начало координат $x = 0$, удовлетворяет неравенству

$$|J[x_0, \bar{u}(x)] - \inf_{(u)} J[x_0, u(x)]| < \epsilon. \quad (3)$$

Здесь

$$J[x_0, \bar{u}(x)] = \int_0^{\infty} \left\{ \omega(x) + \sum_{i=1}^m f_i[u_i(x)] \right\} dt;$$

ϵ — сколь угодно малое наперед заданное положительное число, где $\omega(x)$ — заданная определительно-положительная функция. Функции $f_i[u_i(x)]$ могут принимать такие значения:

$$f_1(u_1) = 0; \quad f_2(u_2) = \delta_2(u_2)^2; \quad f_3(u_3) = \delta_3(u_3).$$

В работах [1, 2] предложено искать $\bar{u}(x)$ в виде

$$\bar{u}_s(x) = \bar{u}_s \left[\sum_{q=1}^n \frac{\partial V(x)}{\partial x_q} d_{sq}^0 \right]. \quad (4)$$

Функция $V(x)$ является решением функционального уравнения Беллмана

$$\inf_{(u)} \left\{ \omega(x) + \left\langle \frac{\partial V(x)}{\partial x}, \varphi(x) \right\rangle + \sum_{i=1}^m F_i[u_i(x)] \right\} = 0, \quad (5)$$

$$V[x(\infty)] = 0,$$

для которого

$$F_1[u, (x)] = f_1[u, (x)] + \sum_{q=1}^n \frac{\partial V(x)}{\partial x_q} d_q^* u_q \quad (6)$$

Подставляя зависимость (4) в (6), получаем минимальные функции

$$F_1^*(\xi_1) = \inf_{[u]} F_1(\xi_1), \text{ где } \xi_1 = \sum_{q=1}^n \frac{\partial V(x)}{\partial x_q} d_q^*.$$

Аппроксимируем неаналитические минимальные функции $F_1^*(\xi_1)$ на конечных интервалах изменения ξ_1 полиномами вида

$$F_1^*(\xi_1) = \lambda_{12}(\xi_1)^2 + \lambda_{13}(\xi_1)^3 + \dots + \lambda_{1n}(\xi_1)^n + R_1'(\xi_1), \quad (7)$$

предполагая, что функции $\omega(x)$, $\varphi_i(x)$ допускают разложение в степенные ряды в некоторой области G_1 , содержащей начало координат $x=0$:

$$\omega(x) = \omega^{(2)}(x) + \dots + \omega^{(n)}(x) + R_\omega'(x); \quad (8)$$

$$\varphi_i(x) = \varphi_i^{(1)}(x) + \dots + \varphi_i^{(n)}(x) + R_{\varphi_i}'(x), \quad i = \overline{1, n}. \quad (9)$$

Функцию $V(x)$ в соответствии с рекомендациями работы [1] отыскиваем единственным образом в виде степенного ряда, сходящегося в заданной области $G \subset E_n$, содержащей $x=0$ и все оптимальные траектории ($G(t)$ — область, содержащая все оптимальные траектории, начиная с момента t):

$$V(x) = V^{(2)}(x) + V^{(3)}(x) + \dots + V^{(n)}(x) + R_V'(x). \quad (10)$$

Подставив ряды (7) — (10) в функциональное уравнение (5) и приравняв нулю коэффициенты при одинаковых степенях, получим соотношения относительно коэффициентов функции Ляпунова — Беллмана $V(x)$. Для решения линейной задачи система уравнений относительно коэффициентов функции $V^{(n)}(x)$ введена в работе [3], а для нелинейной задачи — в работе [1].

Построим конструктивный алгоритм решения задачи синтеза (1) — (3), основанный на методе, предложенном ранее [1]. Пусть области G_2 , G имеют вид

$$G_2: |x_i| \leq x_{im}^i; \quad G: |x_i| \leq x_{i, \max}^i \leq x_{im}^i. \quad (11)$$

Тогда соответствующую G_2 область G_1 в переменных ξ_1 можно записать так:

$$\beta_{11} \leq \xi_1 \leq \beta_{21}.$$

Поскольку минимальные функции $F_1^*(\xi_1)$, соответствующие области G_2 в переменных x_i , симметричны, имеет смысл выбрать β_{11} , β_{21} в виде

$$-\beta_{11} = \beta_{21} = \beta_1 = \max \left\{ \left| \sup_{G_2} \xi_1 \right|, \left| \inf_{G_2} \xi_1 \right| \right\} \quad (12)$$

и аппроксимировать функции $F_s^*(\xi_s)$ на интервалах $[-\beta_s, \beta_s]$ полиномами (7) по четным степеням.

Согласно работе [2] запишем формулу для оценки приближенности построенного управления к оптимальному:

$$\Delta = \int_0^T \left| \left\{ \sup_{\alpha(t)} \left[R'_\omega(x) + \sum_{i=1}^n R'_{\langle x, \varphi \rangle i}(x) \right] - \right. \right. \\ \left. \left. - \inf_{\alpha(t)} \left[R'_\omega(x) + \sum_{i=1}^n R'_{\langle x, \varphi \rangle i}(x) \right] + \right. \right. \\ \left. \left. + \sup_{|\xi_s| < \beta_s(t)} \sum_{i=1}^m R'_i(\xi_s) - \inf_{|\xi_s| < \beta_s(t)} \sum_{i=1}^m R'_i(\xi_s) \right\} \right| dt, \quad (13)$$

где $R'_\omega, \sum_{i=1}^n R'_{\langle x, \varphi \rangle i}(x)$ — остаточные члены рядов для функции $\omega(x)$ и скалярного произведения $\langle \frac{\partial V(x)}{\partial x}, \varphi \rangle$, $R'_i(\xi_s)$ — погрешности аппроксимаций функций $F_s^*(\xi_s)$ полиномами (7). С помощью линейного неособого преобразования [4]

$$x_i = c_{i1}y_1 + \dots + c_{in}y_n,$$

приводящего квадратичную форму $V^{(2)}(x)$ к виду

$$V^{(2)}(y) = y_1^2 + \dots + y_n^2,$$

перейдем к новой системе координат y_i . Тогда области G будет соответствовать область G_y в новых переменных; функциям $\omega(x)$, $\varphi_i(x)$ — функции $\omega(y)$, $f_i(y)$ в переменных y . Выберем для рядов (8), (9) мажорантные в переменных y_i из условий

$$\bar{\varphi}_r \bar{z}_{\max} = \sup_{G_y} |f_i^{(r)}(y)|; \quad \bar{\omega}_r \bar{z}_{\max} = \sup_{G_y} |\omega^{(r)}(y)|.$$

Здесь

$$\bar{z}_{\max} = \max_{G_y} \bar{z}; \quad \bar{z} = \sqrt{\sum_{i=1}^n y_i^2}.$$

Определим постоянные $\bar{c}_r^i (i = \overline{1, n}, r = 2, \dots)$ из условия

$$r \bar{c}_r^i \bar{V}_i \bar{z}_{\max}^{-1} = \sup_{G_y} \left| \frac{\partial V^{(r)}(y)}{\partial y_i} \right|.$$

воспользовавшись известным неравенством

$$\bar{z}(t) \leq \bar{z}(0) e^{-\alpha t}, \quad (-\alpha) = \max_{1 \leq i \leq n} \operatorname{Re} \lambda_i + \delta$$

(λ_i — собственные значения системы первого приближения; δ — сколь угодно малое число). Вычислим последовательно коэффи-

циенты $\tilde{V}_r$ для ряда, являющегося мажорантой (10) в системе координат y_i :

$$\begin{aligned}\tilde{V}_3 &= \frac{1}{3!} \left\{ \sum_{i=1}^n [2\tilde{c}_2^i \tilde{\varphi}_2^i \tilde{V}_2] + \tilde{w}_3 \right\}; \\ \tilde{V}_r &= \frac{1}{r!} \left\{ \sum_{i=1}^n [(r-1)\tilde{c}_{r-1}^i \tilde{V}_{r-1} \tilde{\varphi}_2^i + \dots + 2\tilde{c}_2^i \tilde{\varphi}_{r-1}^i \tilde{V}_r] + \right. \\ &+ \sum_{i=1}^n \left\{ \sum_{j=2}^{r-1} |\lambda_{ij}| \sum_{\substack{k_2+\dots+k_{r-1}=i \\ k_2+2k_3+\dots+(r-2)k_{r-1}=r}} \tilde{C}_i(k_2, \dots, k_{r-1}) \times \right. \\ &\times \left(\sum_{i=1}^n 2\tilde{c}_2^i |\tilde{d}_2^i| \tilde{V}_2 \right)^{k_2} \left(\sum_{i=1}^n 3\tilde{c}_3^i |\tilde{d}_3^i| \tilde{V}_3 \right)^{k_3} \times \dots \times \\ &\times \left(\sum_{i=1}^n (r-1)\tilde{c}_{r-1}^i \tilde{V}_{r-1} |\tilde{d}_1^i| \right)^{k_{r-1}} \Bigg\} + \\ &+ |\lambda_{ir}| \left(\sum_{i=1}^n 2\tilde{c}_2^i |\tilde{d}_2^i| \tilde{V}_2 \right)^r \Bigg\} + \tilde{w}_r.\end{aligned}\quad (14)$$

Здесь $\tilde{d}_i^i$ — коэффициенты матрицы D в переменных y_i . С помощью обратного преобразования $y_i = b_{i1}x_1 + \dots + b_{in}x_n$ найдем коэффициенты $\tilde{V}_r$ ряда, являющегося мажорантой (10) в переменных x_i , определяя и постоянные c_i^i ($i = \overline{1, n}$, $r = 2, \dots$) в этих же переменных. Тогда оценки для остаточных членов $R'_n(x)$, $\sum_{i=1}^n R'_{<x, \varphi> i}(x)$ можно записать так:

$$\begin{aligned}\sup_{\sigma} \left[R'_n(x) + \sum_{i=1}^n R'_{<x, \varphi> i}(x) \right] &< \\ &< \left[\tilde{w}_{r+1} + \sum_{i=1}^n \sum_{j=1}^r (j+1) c_{j+1}^i \tilde{V}_{j+1} \tilde{\varphi}_{r-j+1}^i \right] z_{\max}^{r+1};\end{aligned}\quad (15)$$

$$\begin{aligned}\inf_{\sigma} \left[R'_n(x) + \sum_{i=1}^n R'_{<x, \varphi> i}(x) \right] &= \\ = - \sup_{\sigma} \left[R'_n(x) + \sum_{i=1}^n R'_{<x, \varphi> i}(x) \right].\end{aligned}\quad (16)$$

Отметим, что

$$z = \sqrt{\sum_{i=1}^n x_i^2}, \quad z_{\max} = \max_{\sigma} z.$$

Погрешности аппроксимаций функций $F_i^*(\bar{c}_i)$ полиномами по четным степеням r имеют вид

$$R'_s(\xi_s) = F'_s(\xi_s) - [\lambda_{s2}(\xi_s)^2 + \dots + \lambda_{sr}(\xi_s)^r].$$

Пусть $f_s(u_s) = b_s(u_s)^2$. В этом случае

$$R'_s(\xi_s) = \begin{cases} 0, & \xi_s < \beta_s < 2b_{s1}^*; \\ -\frac{\xi_s^2}{4b_s} - [\lambda_{s2}(\xi_s)^2 + \dots + \lambda_{sr}(\xi_s)^r], & \xi_s < 2b_{s1}^*, < \beta_s; \\ -|\xi_s| u_s^* + b_s(u_s^*) - [\lambda_{s2}(\xi_s)^2 + \dots + \lambda_{sr}(\xi_s)^r], & \beta_s > \xi_s > 2b_{s1}^*; \end{cases}$$

$$\left| \sup_{|\xi_s| < \beta_s} R'_s(\xi_s) - \inf_{|\xi_s| < \beta_s} R'_s(\xi_s) \right| = |R'_s(\xi_{s1}^*) - R'_s(\xi_{s2}^*)|,$$

где ξ_{s1}^* , ξ_{s2}^* — точки максимума и минимума функции $R'_s(\xi_s)$.

Зададим функции $N'_s(|\xi_s(t)|)$ так, чтобы выполнялось условие

$$2|R'_s(\xi_{s1}^*) - R'_s(\xi_{s2}^*)| = N'_s \beta_s. \quad (17)$$

При этом имеет место следующее выражение:

$$N'_s(|\xi_s(t)|) \leq N'_s \sum_{i=1}^n d_i^t [2c_2^t \bar{V}_{2z_{\max}}(t) + \dots + rc_i^t \bar{V}_{rz_{\max}}^{r-1}(t)]. \quad (18)$$

В соответствии с формулами (13), (15) — (18) запишем неравенство для оценки приближенности построенного управления к оптимальному:

$$\Delta \leq \int_0^T \left\{ 2 \left[\omega_{r+1} + \sum_{i=1}^n \sum_{j=2}^{r+1} j c_j^i V_{j\varphi_{r-j+2}} \right] z^{r+1}(t) + \right. \\ \left. + \sum_{i=1}^n N'_i \sum_{j=1}^n d_j^t [2c_2^t V_{2z_{\max}}(t) + \dots + rc_i^t V_{rz_{\max}}^{r-1}(t)] \right\} dt. \quad (19)$$

Пользуясь оценкой $z \leq M e^{-\alpha t}$ (M — константа), устанавливаем, что интеграл (19) сходится.

Задача синтеза (1) — (3) решается в следующем порядке.

1. Задаются число $\varepsilon > 0$, определяющее необходимую точность решения задачи, четная степень r многочленов для функций (7) — (9) (следует начинать решение при $r = 2$), первое приближение интервалов аппроксимаций G_i^{ε} : $[-\beta_i, \beta_i]$, а также области G_2 , G по формуле (11).

2. Минимальные функции $F'_s(\xi_s)$ аппроксимируются (например, методом наименьших квадратов) полиномами (7) степени r по четным степеням. Вычисляются значения коэффициентов аппроксимаций $\lambda_{s2}, \dots, \lambda_{sr}$. Определяется функция $V(x) = V^{(2)} + \dots + V^{(r)}(x)$ согласно рекомендациям работ [1, 3], а также область G_ε по формуле (12). Если $G_\varepsilon \subseteq G_i^{\varepsilon}$, находим Δ по формуле (19) с использованием выражений (15) — (18). При $\Delta \leq \varepsilon$ переход к п. 3. Если $G_\varepsilon \not\subseteq G_i^{\varepsilon}$ или $\Delta > \varepsilon$ пересчитываются интервалы аппроксимации G_i^{ε} с помощью выражения (12) и процедура п. 2

повторяется до совпадения областей G_1, G^6 . При $\Delta \leq \varepsilon$ переход к п. 3. Если при заданном r $\Delta > \varepsilon$, то полагается $r = r + 2$ и процедура п. 2 повторяется до тех пор, пока не будет выполнено условие $\Delta \leq \varepsilon$.

3. Определяется управление $\bar{u}(x)$ минимизацией уравнения (5), принимающее вид (4).

4. Для любого начального состояния $x_0 \in G_0$ интегрированием уравнений (1) при управлении $\bar{u}(x)$ строятся приближенно-оптимальные программные траектории.

Пример. Пусть движение объекта регулирования описывается дифференциальным уравнением

$$\dot{x} = \varphi^{(1)}(x) + \varphi^{(2)}(x) + \varphi^{(3)}(x) + du. \quad (20)$$

Здесь

$$\varphi^{(1)}(x) = \varphi_1 x; \quad \varphi^{(2)}(x) = \varphi_{11} x^2; \quad \varphi^{(3)}(x) = \varphi_{111} x^3,$$

где $\varphi_1 = -0,0173$, $\varphi_{11} = -0,116$, $\varphi_{111} = -0,154$, $d = 0,1049$. На управляющую функцию наложим ограничения

$$|u| < 0,26. \quad (21)$$

Среди допустимых функций $u(x)$, для которых справедливы ограничения (21), требуется найти функцию $\bar{u}(x)$, удовлетворяющую неравенству

$$|J[x_0, \bar{u}(x)] - Q(x_0)| < \varepsilon,$$

в котором

$$J = \int_0^\infty [x^2 + u^2] dt; \quad Q(x_0) = \inf_u J[x, u];$$

$$\varepsilon = 0,0015; \quad x_0 = x(0) \in G_0; \quad |x_0| < 0,24.$$

Области G_2, G заданы в виде $G_2: |x| < 0,26$, $G: |x| < 0,24$; уравнение Беллмана имеет вид

$$\inf_{\{u\}} \left\{ x^2 + u^2 + \frac{\partial V(x)}{\partial x} [\varphi^{(1)}(x) + \varphi^{(2)}(x) + \varphi^{(3)}(x) + du] \right\} = 0.$$

Функцию Ляпунова — Беллмана зададим следующим образом:

$$V(x) = V^{(2)}(x) + V^{(3)}(x) + V^{(4)}(x) + V^{(5)}(x) + V^{(6)}(x) + R_V^6(x),$$

где

$$V^{(2)}(x) = V_{11} x^2; \quad V^{(3)}(x) = V_{111} x^3;$$

$$V^{(4)}(x) = V_{1111} x^4; \quad V^{(5)}(x) = V_{11111} x^5; \quad V^{(6)}(x) = V_{111111} x^6.$$

Введем обозначения

$$\xi = d \frac{\partial V(x)}{\partial x}; \quad F(\xi) = du^2 + d \frac{\partial V(x)}{\partial x} u. \quad (22)$$

Минимизация по ξ функции $F(\xi)$ (22) позволяет записать выражение для оптимального управления:

$$\bar{u}(\xi) = \begin{cases} 0,26, & -\frac{\xi}{2} > 0,26; \\ -\frac{\xi}{2}, & -0,26 < -\frac{\xi}{2} < 0,26; \\ -0,26, & -\frac{\xi}{2} < -0,26. \end{cases} \quad (23)$$

В результате подстановки соотношений (23) в (22) можно найти минимальную функцию $F^*(\xi) = \inf F(\xi)$, которая на отрезке $[-0,53, 0,53]$ аппроксимируется полиномом шестой степени

$$F^*(\xi) = \lambda_2 \xi^2 + \lambda_4 \xi^4 + \lambda_6 \xi^6 + R^6(\xi)$$

с коэффициентами аппроксимации $\lambda_2 = -0,250196$, $\lambda_4 = 0,00263$, $\lambda_6 = -0,007188$. Коэффициенты функции Лапунова — Беллмана принимают следующие значения: $V_{11} = 8,09$; $V_{111} = -5,98$; $V_{1111} = -3,39$; $V_{11111} = 5,74$; $V_{111111} = 1,7$.

Степень приближенности полученного управления к оптимальному $\Delta < 0,0015$.

Список литературы: 1. Тригуб М. В. Оптимальная стабилизация одного класса нелинейных систем. — АСУ и приборы автоматки, 1986, вып. 77, с. 11—16. 2. Александров Е. Е., Тригуб М. В. Оценка точности синтеза приближенно-оптимального управления для нелинейного объекта. — Изв. вузов СССР, Приборостроение, 1982, с. XXV, № 10, с. 46—52. 3. Бахито Р. У., Крайчев Н. Н., Кротов В. Ф. Синтез приближенно-оптимального управления для одного класса управляемых систем. — Автоматика и телемеханика, 1972, № 10, с. 64—72. 4. Курош А. Г. Курс высшей алгебры. — М.: Наука, 1975. — 431 с.

Поступила в редакцию 15.04.85.

УДК 519.2

А. А. ЖИГЛЯВСКИЙ, канд. физ.-мат. наук, М. В. ТЕРЕНТЬЕВА

СТАТИСТИЧЕСКИЕ МЕТОДЫ В ГЛОБАЛЬНОМ СЛУЧАЙНОМ ПОИСКЕ. ОПИСАНИЕ АЛГОРИТМОВ

Рассмотрим задачу поиска глобального максимума алгоритмически заданной функции f на компактном множестве $X \subset R^n$. Эта задача состоит в том, чтобы по результатам вычислений f в точках из множества X найти такую точку $x^* \in X$, в которой $f(x^*) = \max_{x \in X} f(x) = M$.

Решая оптимизационные задачи, обычно принимают во внимание неполноту априорной информации о целевой функции (например, неизвестно, удовлетворяет ли функция f условию Липшица с приемлемой константой, и какое количество локальных максимумов она имеет). При большой размерности n про-

странства X это делает практически невозможным использование любого из детерминированных подходов (достаточно полный обзор современного состояния теории поиска глобального экстремума можно найти в работе [2, § 4.4]), и в указанной ситуации использование методов глобального случайного поиска, как правило, — наилучший способ решения таких задач.

Алгоритмы глобального случайного поиска основаны на конструировании последовательности вероятностных мер, сходящейся к вероятностной мере, сосредоточенной в точке x^* . Алгоритмы используют следующую идею: для надежности и эффективности алгоритма глобального случайного поиска необходимо, чтобы целевая функция вычислялась во всем множестве X согласно функции ценности φ , которая строится адаптивно с использованием оценок f или функционалов от f , и должна обладать тем свойством, что если $f(x_1) > f(x_2)$, то функция ценности в некоторой окрестности x_1 больше, чем в соответствующей окрестности x_2 .

Алгоритмы глобального случайного поиска удобно представлять в виде итерационных, на каждой итерации которых вычисляются следующие операции. На i -й итерации получаем заданное число $N_i (N_i > 1)$ точек $x_1^{(i)}, \dots, x_{N_i}^{(i)}$, как независимые реализации случайного вектора ξ_i с некоторым распределением $P_i(dx)$. Затем анализируем данные $\{x_j^{(i)}, f(x_j^{(i)}), j = \overline{1, N_i}\}$, извлекая информацию об f . Важнейший способ учета информации, получаемой в ходе поиска, — способ выбора распределения $P_{i+1} \times \times (dx)$ на следующей итерации. Строим вероятностную меру $P_{i+1} \times \times (dx)$ таким образом, чтобы условная вероятность попадания в окрестность x^* возрастала.

Вероятностные модели метода ветвей и границ. Рассмотрим алгоритмы, использующие идею широко распространенного в дискретном программировании метода ветвей и границ. Суть метода состоит в отбрасывании некоторых подмножеств X , заведомо не дающих максимума; поиск ведется лишь среди оставшихся подмножеств, являющихся «перспективными». Схематично работу метода можно представить как последовательное осуществление на каждой итерации трех основных этапов.

1. Ветвление исходного множества на дерево подмножеств (разбиение).

2. Нахождение оценок целевой функции на получаемых подмножествах.

3. Определение функции ценности и выбор «перспективных» подмножеств для дальнейшего их ветвления. Метод состоит из нескольких итераций, каждая из которых ведет либо к непосредственному сужению исходного множества, либо подготавливает возможность такого сужения на следующих итерациях. Иными словами, исходное множество разбивается на подмножества, и

на этих подмножествах задается функция ценности, которая оценивает каждое подмножество — вершину дерева ветвления.

В стандартном методе ветвей и границ для построения функции ценности выбираются верхние границы функции f на рассматриваемом подмножестве. При этом все ветви, соответствующие множествам с верхней границей, меньшей или равной максимальному значению для уже рассмотренных множеств, отбрасываются. Таким образом, в этом методе функция ценности $\varphi(Z_{ij})$, $i = \overline{1, s}$, $j = \overline{1, l_i}$ равна либо единице (во множестве Z_{ij} находится глобальный максимум, множество подвергается дальнейшему просмотру (ветвлению)), либо нулю (глобальный максимум в этом множестве достигаться не может, множество отбрасывается).

Рассмотрим алгоритмы, являющиеся вероятностными аналогами метода ветвей и границ. Изучим функцию ценности, одну из наиболее естественных и просто вычисляемых, и опишем порождаемый ею алгоритм. Пусть M_i — наибольшее из полученных на предыдущих итерациях значение функции f . Имеется разбиение X на подмножества Z_{ij} ($j = \overline{1, l_i}$): $X = \bigcup_j Z_{ij}$. В каждом из множеств Z_{ij} вычисляем вероятности p_{ij} того, что $\max_{x \in Z_{ij}} f(x) > M_i$. Функцию ценности φ на подмножествах Z_{ij} определяем следующим образом:

$$\varphi(Z_{ij}) = p_{ij} = 1 - \left[1 - \left(\frac{M_i}{M_i + \gamma_{(N)} - \gamma_{(N-k)}} \right)^{\gamma} \right]^k, \quad (1)$$

где $\gamma_{(N)}$, $\gamma_{(N-k)}$ — порядковые статистики, соответствующие точкам, попавшим в Z_{ij} [1, § 3]. Величины p_{ij} допускают следующую трактовку с точки зрения теории проверки гипотез. Для каждого множества Z_{ij} рассмотрим задачу проверки гипотезы $H_0: \max_{x \in Z_{ij}} f(x) > M_i$ при альтернативе $H_1: \max_{x \in Z_{ij}} f(x) < M_i$. Тогда p_{ij} — ве-

роятность того, чтобы принять гипотезу H_0 при условии, что она верна [1, § 4]. На следующей итерации количество точек в Z_{ij} принимается пропорциональным полученной функции ценности $\varphi(Z_{ij})$. При фиксированном уровне затрат дальнейшему ветвлению подвергаются множества, у которых данная вероятность p_{ij} не меньше $1 - \gamma_i$, где $\gamma_i > 0$.

Формально описанный алгоритм в общем виде выглядит следующим образом.

Алгоритм 1.

1. Полагаем $i = 0$, $X_0 = X$. Если нет априорной информации о положении точки x^* , определяем распределение $p_0(dx)$ как равномерное на множестве X_0 . Моделируем N_0 раз распределение $p_0(dx)$, получаем $\sum_0 = \{x_1^{(0)}, x_2^{(0)}, \dots, x_{N_0}^{(0)}\}$. Полагаем $M_0 = \max(f(x_i^{(0)}), i = \overline{1, N_0})$.

2. Представляем X_i в виде $X_i = \bigcup_j Z_{ij}$, где Z_{ij} — измеримые

односвязные подмножества X (не обязательно непересекающиеся), в каждое из которых попало достаточное для проведения статистических видов количество точек из Σ_t .

3. Для каждого подмножества Z_{it} находим функцию ценности $\varphi(Z_{it})$.

4. Выбираем γ_t . Подмножества Z_{it} , для которых $\varphi(Z_{it}) < 1 - \gamma_t$, — исключаем из рассмотрения, т. е. полагаем $X_{t+1} = \bigcup Z_{it}^*$, где

$$Z_{it}^* = \begin{cases} Z_{it}, & \varphi(Z_{it}) > 1 - \gamma_t \\ \emptyset, & \varphi(Z_{it}) < 1 - \gamma_t \end{cases}$$

5. Пусть

$$\varphi(Z_{it}^*) = \begin{cases} Z_{it}, & Z_{it}^* = Z_{it} \\ 0, & Z_{it}^* = \emptyset \end{cases}$$

Полагаем $P_{t+1}(dx) = \sum_{i=1}^{I_t} p_i^{(t)} Q_{it}(dx)$, где $p_i^{(t)}$ — равномерное распределение на Z_{it} .

6. Моделируя N_{t+1} раз распределение $P_{t+1}(dx)$, получаем E_{t+1} . Полагаем $M_{t+1} = \max(M_t, f(x_1^{(t+1)}), \dots, f(x_{N_{t+1}}^{(t+1)}))$.

7. Переходим к п. 2. Распределение $P_{t+1}(dx)$ моделируется по методу суперпозиции: сначала моделируется дискретное распределение, сосредоточенное в точках $\{1, I_t\}$ с вероятностями $p_i^{(t)}$ (пусть j — полученная реализация), а затем — распределение $Q_{it} \times \times (dx)$.

Видоизменения и модификации алгоритма 1. Сделаем ряд замечаний относительно алгоритма 1 и его модификаций.

1°. В алгоритме 1 на каждой итерации вероятность того, что отбрасывается множество, в котором находится глобальный максимум, меньше γ_t . Таким образом, вероятность не потерять точку глобального максимума за несколько этапов Δ не меньше $1 - \max \gamma_t$. После s итераций алгоритма по формуле (1) можно оценить вероятность того, что глобальный максимум не потерян.

2°. В случае, если все γ_t положить равными нулю, то получим один из вариантов адаптивного случайного поиска, в котором более перспективные области просматриваются тщательнее.

3°. Возможны различные модификации алгоритма 1. Например, в качестве функции ценности можно взять $\varphi(Z_{it}) = r_{it}'$, где r_t — некоторые числа.

4°. На основе различных точечных оценок величины $M_x = \max_x f(x)$ [1, § 2] можно рассмотреть другие функции ценности $\varphi(Z_{it})$ и предложить следующий алгоритм. Отметим, что algo-

ритмы, основанные на функции ценности в виде оценок (получаемых отличным от работы [1] способом) разрабатывались и применялись для решения конкретных практических задач в работах [3, 4].

Алгоритм 2. Пункты 1, 2 такие же как в алгоритме 1. В качестве функции ценности $\varphi(Z_{ij})$ возьмем оценку $\tilde{M}_{ij}$ величины $\max_{Z_{ij}} X \times f(x)$. Например,

$$\varphi(Z_{ij}) = \tilde{M}_{ij} = \left(1 + \frac{2\alpha + 1}{k + 1}\right) \varphi_{(N)} - \frac{2\alpha - 1}{k + 1} \varphi_{(N-k)},$$

где $\varphi_{(N)}$, $\varphi_{(N-k)}$ — порядковые статистики, соответствующие точкам, попавшим в Z_{ij} . Распределение $P_{i+1}(dx)$ на следующей итерации строится аналогично.

5°. Во всех этих алгоритмах ветвления множества X определяется спецификой решаемой задачи. Здесь возможны различные варианты. Например, разбиение X может быть следующее: если X — эллипсоид, параллелепипед, то Z_{ij} могут быть того же вида; если X — произвольное множество, то в качестве Z_{ij} естественно брать шары. Выделяем подобласть Z_{ij} с центром в точке, соответствующей максимальной порядковой статистике. Следующую область выделяем с центром в точке, соответствующей порядковой статистике, максимальной из оставшихся. На следующих итерациях подмножества Z_{ij} , имеющих наибольшую функцию ценности, разбиваем на несколько частей. Отметим, что в отличие от традиционных вариантов метода ветвей и границ подмножества Z_{ij} не обязательно должны быть непересекающимися. Единственное требование на выбор подмножества Z_{ij} следующее: в каждое Z_{ij} должно попасть достаточное для проведения статистических выводов количество точек из E_i [1]. Поэтому на первой итерации не следует разбивать множество X на большое число частей.

Как следует из работы [1], для проведения эффективных статистических выводов достаточно иметь 10 или меньше максимальных порядковых статистик. Поэтому для каждого множества Z_{ij} нужно хранить в памяти не более 10 точек, в которых достигаются максимальные статистики.

6°. В случае, если ресурсы ограничены и нужно проявлять особую осторожность в планировании измерений (каждое измерение — дорогостоящий эксперимент), то разумна следующая стратегия. После того как произведено N измерений функции f , следует извлечь и использовать как можно больше информации о целевой функции f . Чтобы получить эту информацию для каждого множества Z_{ij} , необходимо вычислить вероятность p_{ij} , построить различные точечные оценки величины $M_{Z_{ij}} = \max_{Z_{ij}} f(x)$, например, вида (2), оптимальные оценки в классе линейных оценок,

использующих заданное количество экстремальных статистик исходной выборки [1, § 2], проверить гипотезы о параметре α распределения экстремальных значений [1, § 5], которые показывают, попали ли измерения в окрестность глобального максимума. И только после проведения всевозможных статистических выводов принять решение о «перспективности» данного подмножества Z_{ij} .

7°. На данные алгоритмы следует смотреть как на методы, позволяющие найти окрестность глобального максимума, в которой можно применять различные локальные методы для уточнения точки глобального максимума.

Список литературы: 1. Жигляевский А. А., Терентьева М. В. Статистические методы в глобальном случайном поиске. Математический аппарат. — АСУ и приборы автоматики, 1986, вып. 78, с. 9—18. 2. Ермоков С. М. Математическая теория планирования эксперимента. — М.: Наука, 1983. — 140 с. 3. Яковлев С. В. О некоторых схемах поисковой оптимизации. — АСУ и приборы автоматики, 1985, вып. 75, с. 9—14. 4. Стоян Ю. Г., Яковлев С. В. Исследование сходимости и эффективности метода сужающихся окрестностей. — Х., 1981. — 23 с. — Препринт (АН УССР, Ин-т пробл. машиностроения, № 168).

Поступила в редколлегию 30.04.84.

УДК 681.34

Н. Н. СЕНЧЕНКО, канд. техн. наук, Н. И. ГЛАДЕНКО

ОЦЕНКИ МАКСИМАЛЬНЫХ ЗНАЧЕНИЙ РЕШЕНИЙ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЯ МЕТОДОМ ИНТЕРВАЛОВ

При моделировании динамических систем на аналоговых и аналого-цифровых вычислительных системах возникает необходимость априори оценить максимальные значения решений обыкновенных дифференциальных уравнений, которые в дальнейшем будут использоваться для выбора масштабных и передаточных коэффициентов. Для решения этой задачи авторами были предложены аналитические методы оценки.

Рассмотрим задачу Коши для системы обыкновенных дифференциальных уравнений

$$y'(x) = F(x, y), \quad a \leq x \leq b;$$

$$y(0) = y^0. \quad (1)$$

Здесь $y(x)$ — вектор-функция, которую будем трактовать как элемент пространства $C = C([a, b], R_m)$ — непрерывных на $[a, b]$

функций со значениями в m -мерном пространстве R_m , $y(x) = (y_1(x), \dots, y_m(x))$. Норма в пространстве C вводится естественным образом:

$$\|y(x)\|_C = \max_{a \leq x \leq b} \|y(x)\|_{R_m},$$

где $\|\cdot\|_{R_m}$ — какая-нибудь норма в R_m ; $F(x, y)$ — вектор-функция, определенная на произведении $[a, b] \times R_m$ со значениями в R_m , $F(x, y) = (f_1(x, y), \dots, f_m(x, y))$. Считаем, что функция $F(x, y)$ удовлетворяет следующим условиям. Она непрерывна в некотором $(m+1)$ -мерном параллелепипеде $G = [a, b] \times D$, где $D = \{y \in R_m : |y_i - y_i^0| < A\}$.

В параллелепипеде G функция $F(x, y)$ удовлетворяет условию

$$\|F(x, y^{(1)}) - F(x, y^{(2)})\|_{R_m} < \varphi(x) \|y^{(1)} - y^{(2)}\|_{R_m}.$$

Здесь $\varphi(x)$ — непрерывная на $[a, b]$ функция,

$$\varphi(x) = \max_{1 \leq i \leq m} \max_{y \in D} \sum_{j=1}^m \left| \frac{\partial f_i}{\partial y_j} \right| > 0.$$

Функции, обладающие таким свойством, называются локально липшицевыми. Величина $k = \max_{a \leq x \leq b} \varphi(x)$ представляет собой константу Липшица функции $F(x, y)$ в параллелепипеде G .

Известно*, что если функция $F(x, y)$ удовлетворяет условию Липшица в указанном параллелепипеде, то на интервале $[a, a+h]$, где

$$h = \min \{b-a, A/L\};$$

$$L = \max_G \|F(x, y)\|_{R_m},$$

задача (1) имеет единственное решение $y^*(x)$, для которого верна оценка

$$P_1 = \|y^*(x)\|_C \leq \|y^0\|_{R_m} + \frac{L}{k} (e^{kx} - 1).$$

Введем функции

$$E(x) = \max_{a \leq t \leq x} \left\| \int_a^t F(\sigma, y^0) d\sigma \right\|_{R_m};$$

$$\theta(x) = E(x) \exp \left(\int_a^x \varphi(t) dt \right);$$

$$\mu(x) = \frac{N}{k} [e^{kx-a} - 1];$$

$$k = \max_{a \leq x \leq b} \varphi(x); \quad N = \sup_{a \leq x \leq a+b} E(x)/(x-a).$$

* Гривоми Ф. Дифференциальные уравнения. — М.: Изд-во иностр. лит-ры, 1962. — 351 с.

Функции $E(x)$, $\theta(x)$, $\mu(x)$ неотрицательны, монотонно возрастающие.

Пусть h_1 — корень уравнения $\theta(a+x) = A$, h_2 — корень уравнения $\mu(a+x) = A$. Поскольку функции $\theta(x)$, $\mu(x)$ положительны, монотонно возрастающие, а также $\theta(a) = \mu(a) = 0$, эти уравнения имеют не более одного решения.

Положим

$$h = \max \{h_1, h_2\}, H = \min \{b-a, h_1\}.$$

Тогда при сделанных предположениях были получены следующие оценки максимальных значений решений задачи (1):

$$P_2(H) = \|y^*(x)\|_{C[a, a+H]} \leq \|y^0\|_{R_m} + \frac{N(H)}{k} [e^{k(H-a)} - 1]; \quad (2)$$

$$P_3(H) = \|y^*(x)\|_{C[a, a+H]} \leq \|y^0\|_{R_m} + E \exp \left(\int_a^x \varphi(t) dt \right). \quad (3)$$

Так как $N < L = \|F(x, y)\|_C$, оценка $P_2(H)$ лучше оценки P_1 . Если $y^0 = 0$, то оценка $P_2(H)$ в L/N раз лучше оценки P_1 .

Однако бывают случаи, когда оценки $P_2(H)$, $P_3(H)$ не обеспечивают требуемой точности. Точность этих оценок существенно зависит от свойств функции $F(x, y)$, точнее от константы Липшица этой функции, и от области существования решения. Чем меньше интервал существования решения мы рассматриваем, тем точнее эти оценки. Поскольку при сделанных предположениях решение задачи Коши устойчиво, т. е. малые изменения начальных условий влекут незначительные изменения решения, при реализации вычислительного процесса на аналоговых и аналого-цифровых вычислительных системах можно применять следующий подход.

Пусть известно, что решение задачи (1) существует на интервале $[a, a+H]$ и оно единственно. Для получения более точных оценок максимальных значений решения задачи (1) разобьем интервал $[a, a+H]$ точками $a+h_1, a+h_2, \dots, a+h_n$ на n частей. Используя $P_2(H)$, $P_3(H)$, найдем оценки решения задачи (1) на интервале $[a, a+h_1]$. По ним традиционными методами определим точное решение $y^*(t)$ задачи (1) на интервале $[a, a+h_1]$. Повторим эту процедуру на интервале $[a+h_1, a+h_2]$, где в качестве начальных условий задачи (1) выбирается значение $y^*(a+h_1)$. Выполняя этот процесс n раз, получаем точное решение $y^*(t)$ задачи (1) на всем интервале $[a, a+H]$. Найденные на каждом интервале $[a+h_i, a+h_{i+1}]$ оценки будут незначительно завышены в силу малости чисел h_i .

Пусть ε — допустимое отклонение оценок максимальных значений решений от максимума истинного решения задачи (1), h_1 — корень уравнения $\theta(a+x) = \varepsilon$. Так как $\theta(a+x)$ возрастающая функция, такое решение существует. Из оценок P_2, P_3 следует, что максимум точного решения задачи (1) на интервале $[a, a+h_1]$ не превышает заданное значение ε . Далее выберем h_2 как корень уравнения $\theta(a+h_1+x) = \varepsilon$. Из тех же соображений сле-

дует, что максимум точного решения задачи (1) также не превышает ε . Продолжая этот процесс, получим последовательность интервалов $[a + h_i, a + h_{i+1}]$, и на каждом из них — оценку решения задачи (1)

$$\|y^*(x)\|_{C[a+h_i, a+h_{i+1}]} \leq \|y^*(a+h_i)\|_{R_m} + \varepsilon.$$

Покажем, что конечное число интервалов $[a + h_i, a + h_{i+1}]$ покрывает интервал $[a, a + H]$. Поскольку функция $F(x, y)$ непрерывно дифференцируема в области $G = [a, a + H] \times D$, по построению точек $a + h_i$ имеем

$$\begin{aligned} \varepsilon &= \theta(a + h_{i+1}) = \max_{a+h_i \leq t \leq a+h_{i+1}} \left\| \int_{a+h_i}^{a+h_{i+1}} F(t, y^*(a+h_i)) dt \right\| \times \\ &\times \exp \left(\int_{a+h_i}^{a+h_{i+1}} \varphi(t) dt \right) \leq \max_{a \leq t \leq a+H} \|F(t, y)\| (h_{i+1} - h_i) e^{k(h_{i+1} - h_i)} \leq \\ &\leq M (h_{i+1} - h_i) e^{k(h_{i+1} - h_i)} \leq M (h_{i+1} - h_i) e^{kH}. \end{aligned}$$

Следовательно,

$$h_{i+1} - h_i \geq \varepsilon / e^{kH}.$$

Значит

$$N = 1 + \frac{HMe^{kH}}{\varepsilon}$$

интервалов $[a + h_i, a + h_{i+1}]$ покрывают интервал $[a, a + H]$.

Если сразу разбить интервал $[a, a + H]$ на N частей, то, очевидно, на каждом i -м интервале отклонение функции от i -го начального условия не будет превосходить ε .

Оценить максимальные значения решений задачи (1) на каждом шаге можно с помощью оценок P_2, P_3 . В каждую из последних входит параметр e^R ($R = \int_a^{a+H} \varphi(t) dt$ либо $R = kH$). Поэтому идея интервальных оценок основана на замене e^R на $ne^{R/n}$, где n — число разбиений отрезка $[a, b]$ на части. Очевидно, при $R < 1$ применять интервальные оценки нецелесообразно. Выбор числа разбиений n определяется условием $ne^{R/n} < e^R$, $R > 1$.

Исследуем функцию $z(n) = ne^{R/n}$. Если $1 < n < R$, с ростом n значение $ne^{R/n}$ убывает. В случае $R < n$ функция $ne^{R/n}$ с увеличением n возрастает. Разлагая функцию $e^{R/n}$ в ряд при $n < e^R - R$, получаем

$$ne^{R/n} < e^R.$$

Следовательно, применение интервальных оценок оправданно, если число разбиений отрезка $[a, b]$ находится в пределах

$$1 < n < e^R - R. \quad (4)$$

При $n \rightarrow (e^R - R)$ эффективность интервальных оценок уменьшается.

Априори предположим, что лучшей оценке соответствует лучшая интервальная оценка. Тогда при двух оценках $P_2(H), P_3 \times \times (H)$ лучшей будет оценка, удовлетворяющая условию

$$\min [P_2(H), P_3(H)].$$

Если

$$\frac{N(x)}{k} [e^{k(x-c)} - 1] > (< 0) E(x) e^{\int_c^x \alpha(t) dt} \\ (\forall x \in [c, H], \forall c \in [a, a+H]),$$

то интервальная оценка при P_2 лучше (хуже) интервальной оценки при P_3 .

Проиллюстрируем эффективность интервальных оценок на примере. Пусть для системы

$$\begin{aligned} \dot{y}_1 &= y_2; \\ \dot{y}_2 &= 4y_2 + \sin y_1 + y_4; \\ \dot{y}_3 &= y_4; \\ \dot{y}_4 &= -y_4 - y_3 + y_1 \end{aligned} \quad (5)$$

сформулирована задача Коши с начальными условиями

$$y_1(0) = 1; y_2(0) = 0; y_3(0) = 1; y_4(0) = 0.$$

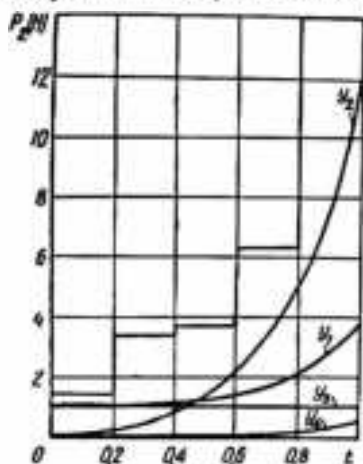
Независимая переменная t изменяется на интервале $[0, 1]$. Предположим, что правые части f_i уравнений (5) определены во всем четырехмерном пространстве, т.е. $f_i(y_1, y_2, y_3, y_4)$ определены в пространстве R_4 . Запишем систему (5) в векторном виде

$$\begin{aligned} Y'(t) &= F(t, Y) = F(Y); \\ Y(0) &= Y^0, t \in [0, 1]. \end{aligned}$$

Функция $F(t, Y)$ локально липшицева на интервале $[0, 1]$. Поэтому на всем этом интервале существует единственное решение задачи Коши. Используя неравенства (2), (3), получаем оценку максимальных значений решений задачи (5). Учитывая, что система уравнений (5) автономна, воспользуемся оценкой $P_2(H)$, которая для таких систем лучше оценки $P_3(H)$. Тогда $H=1$, $k=6$, $N=0,8415$, $E(x)=\sin x$. Отсюда $P_2(H)=57,33$. Если учесть, что максимальное значение решения системы (5) $\|y^*(t)\|$ $\|c=11,712$ (рисунок), то полученная оценка является неточной, отклонения ε на участках разбиения интервала $[a, b]$ от каждого нового начального приближения можно сделать достаточно малыми, если использовать интервальные оценки. Это требует большого количества итерационных шагов. Например, чтобы получить отклонения $\varepsilon=0,1$, для задачи (5) необходимо сделать не менее 110 шагов. На практике такая высокая точность оценок не нужна. Поэтому можно произвольно разделить интервал на несколько частей, выполняя условие (4), и получить соответствующие оценки.

Разобьем интервал $[0, 1]$ на пять равных частей с шагом 0,2. На интервале $[0; 0,2]$ $P_2=1,31$ максимум точного решения

достигается в правой точке и равен 1,02. Используя это значение в качестве начального условия на интервале $[0,2; 0,4]$, получаем $P_2 = 3,27$. Максимум точного решения по-прежнему достигается в правой точке и равен 1,125. В таблице приведены данные расчета по всем интервалам. Отсюда следует, что максимум точного решения задачи (5) отличается от полученной



Решения системы (3), интервальные оценки

Интервал	P_i	Максимум точного решения
$[0; 0,2]$	1,31	1,02
$[0,2; 0,4]$	3,27	1,13
$[0,4; 0,6]$	3,57	2,18
$[0,6; 0,8]$	6,36	5,17
$[0,8; 1]$	14,12	11,71

нами оценки на значение, не превосходящее 2,42. Отметим, что отклонение максимума точного решения от оценки на всем интервале $[0; 1]$ равно 45, 62.

Поступила в редколлегию
23.01.64.

УДК 621.391

Г. Г. ПИСАРЕНОК, канд. техн. наук, С. И. ШМАТКОВ

СРАВНЕНИЕ МЕТОДОВ ОЦЕНИВАНИЯ СРЕДНИХ ДВУХ НОРМАЛЬНЫХ СОВОКУПНОСТЕЙ ПО РЕЗУЛЬТАТАМ ГРУППОВЫХ ИЗМЕРЕНИЙ

При обработке измерительной информации в автоматизированных системах управления не всегда можно однозначно указать принадлежность измерений определенным объектам. Традиционно обработка таких измерений заключается в их предварительной классификации и последующем оценивании параметров объектов по классифицированным выборкам измерений. В этом случае качество оценок существенно зависит от ошибок при классификации.

Решение такой задачи, не зависящее от качества классификации измерений, можно получать при использовании статистик,

инвариантных относительно группы перестановок измерений. Параметры одновременно оцениваются на основе анализа всей совокупности измерений. Классификация не требуется, поскольку значения таких статистик не изменяются при перестановках измерений. Примером могут служить методы, описанные в работах [1, 2].

В статье сравниваются различные методы оценивания по точности получения оценок и вычислительным затратам на примере оценивания средних двух нормальных совокупностей.

Пусть $Q = \{\xi_1, \xi_2\}$ — совокупность двух независимых случайных величин, имеющих нормальное распределение вероятностей с одинаковой дисперсией σ^2 и множеством средних $A = \{a_1, a_2\}$. Имеется n независимых сеансов измерений множества A в виде $Q_1, \dots, Q_n$, где $Q_k = \{x_{1k}, x_{2k}\}$ — множество, содержащее по одному значению каждой случайной величины из Q . Порядок следования элементов множества Q_k произволен. Определить множество оценок $\hat{A} = \{\hat{a}_1, \hat{a}_2\}$ требуется по реализациям случайной совокупности Q .

Рассмотрим традиционные процедуры обработки с предварительной классификацией измерений. Если рассмотреть все возможные способы классификации выборок, то оптимальное по критерию минимума условного риска решение основано на том, что для n выборок существует $k = 2^n$ возможных последовательностей. Для каждого из вариантов классификации вычисляется множество средних $\hat{A}_i = \{\hat{a}_{1i}, \hat{a}_{2i}\}$. Используя $\hat{A}_i$, определяем значения функции правдоподобия для каждого из вариантов. В качестве решения принимается вариант, имеющий максимальное значение функции правдоподобия. Это решение было названо «решением возрастающей сложности». Из-за трудностей практической реализации далее его не рассматриваем.

Известно [3], что линейные классификаторы оптимальны в байесовском смысле только для нормальных распределений с равными ковариационными матрицами. Например, среднее значение выборки измерений двух нормальных совокупностей

$$\bar{a} = \frac{1}{2n} \sum_{i=1}^2 \sum_{k=1}^n x_{ik} \quad (1)$$

является состоятельной оценкой порога $(a_1 + a_2)/2$, который полностью характеризует правило классификации в случае выполнения условий

Таким образом, процедуры линейной классификации предполагают операции сравнения каждого из измерений с оценкой порога (1), а также разделения их на две категории и оценивания средних для каждой из категорий. В этом случае дисперсия оценки $\hat{a}_j$ ($j = 1, 2$) равна σ^2/n , однако оценки имеют смещение.

Квадрат суммарной ошибки, равный сумме дисперсии и квадрата смещения, можно приближенно определить следующим образом:

$$E_s^2 = \frac{\sigma^2}{n} \left(1 + \frac{P_{sk} \Delta^2 n}{16\sigma^2} \right), \quad (2)$$

где

$$P_{sk} = \frac{1}{\sqrt{2\pi}} \int_{-\Delta/2}^{\Delta/2} e^{-x^2/2} dx;$$

$$\Delta = a_2 - a_1.$$

Использование статистик, зависящих от наблюдений через максимальный инвариант относительно группы перестановок [2], приводит к нелинейным оценкам. Таким максимальным инвариантом является вектор базисных функций $f = (f_1, f_2)$, состоящий из симметрических однородных полиномов порядка однородности от одного до двух, в качестве которых можно взять элементарные симметрические функции

$$\sigma_1 = f_1 = x_1 + x_2; \quad \sigma_2 = f_2 = x_1 x_2.$$

В этом случае каждое множество Q_k есть множество корней полинома

$$p(x) = x^2 - \sigma_1(Q_k)x + \sigma_2(Q_k).$$

Итак, можно сделать взаимно однозначное и непрерывное отображение неупорядоченного множества Q_k в упорядоченный (по порядку однородности) вектор базисных функций f .

Оценка $\hat{A} = [\hat{a}_1, \hat{a}_2]$ представляет собой множество корней системы уравнений

$$f(\hat{A}) = F(f(Q_1), \dots, f(Q_n)).$$

В работе [2] показано, что в случае двух нормальных совокупностей с одинаковыми дисперсиями процедура несмещенного оценивания с минимальной дисперсией не зависит от выбора вектора базисных функций. Она включает в себя отыскание корней полинома

$$p(x) = x^2 - \tilde{\sigma}_1(x) + \tilde{\sigma}_2, \quad (3)$$

где

$$\tilde{\sigma}_1 = \langle \sigma_1(Q_k) \rangle = \frac{1}{n} \sum_{k=1}^n (x_{1k} + x_{2k});$$

$$\tilde{\sigma}_2 = \langle \sigma_2(Q_k) \rangle + \frac{1}{4} (\langle \sigma_1(Q_k) \rangle^2 - \langle \sigma_1^2(Q_k) \rangle) =$$

$$= \frac{1}{n} \sum_{k=1}^n x_{1k} \cdot x_{2k} + \frac{1}{4} \left\{ \left[\frac{1}{n} \sum_{k=1}^n (x_{1k} + x_{2k}) \right]^2 - \frac{1}{n} \sum_{k=1}^n (x_{1k} + x_{2k})^2 \right\}.$$

Решив уравнение (3) и выполнив элементарные преобразования, получим

$$\begin{aligned}\hat{a}_1 &= \frac{\sum_{k=1}^n (x_{1k} + x_{2k})}{2n} + \sqrt{\frac{\sum_{k=1}^n (x_{1k} - x_{2k})^2}{4n}}; \\ \hat{a}_2 &= \frac{\sum_{k=1}^n (x_{1k} + x_{2k})}{2n} - \sqrt{\frac{\sum_{k=1}^n (x_{1k} - x_{2k})^2}{4n}}.\end{aligned}\quad (4)$$

Данное решение с точностью до перестановок элементов $\hat{a}_1, \hat{a}_2$ единственно и не зависит от порядка следования измерений. Однако оценки нелинейно зависят от измерений.

Полученные оценки являются асимптотически несмещенными [1, 2]. Приближенное выражение для оценки дисперсии, найденное методом линеаризации, имеет вид

$$\sigma_n^2 = \frac{\sigma^2}{n} \left(1 + \frac{\sigma^2}{\Delta^2} \right). \quad (5)$$

Сравнение точностных характеристик анализируемых методов путем сопоставления выражения (2), (5) показывает, что при

$$\sigma^4/\Delta^4 < P_{\Delta}^2 n/16,$$

т. е. для значений Δ , сравнимых с величиной σ , и при достаточно больших n нелинейный метод оценивания эффективнее линейного классификатора.

Точностные характеристики оценок $\hat{A}$ по формулам [4] и вычислительные затраты на их получение сравнивали с оценками, получаемыми с помощью линейного классификатора. Сравнение выполнено на основе математического моделирования, в процессе которого имитировались измерения x_{ik} ($n = 1000$) при различных значениях Δ, σ . По результатам моделирования для линейного и нелинейного методов оценивались смещения оценок S_{Δ}, S_{σ} , дисперсии оценок $\sigma_{\Delta}^2, \sigma_{\sigma}^2$ и квадраты суммарных ошибок $E_{\Delta}^2 = \sigma_{\Delta}^2 + S_{\Delta}^2$; $E_{\sigma}^2 = \sigma_{\sigma}^2 + S_{\sigma}^2$.

Значения ошибок и времени счета, соответствующие линейному классификатору, приняты за единицу. Тогда относительные суммарные ошибки при $\Delta/\sigma = 0,5$ (0,75; 1,00; 1,25; 1,50; 1,75) равны 0,78 (0,82; 0,92; 1,00; 1,12; 2,07). Затраты времени при нелинейном оценивании уменьшаются в среднем на 21 %.

Таким образом, в условиях наибольшей неопределенности отнесения измерений к одной из совокупностей ($\Delta/\sigma < 1,25$) нелинейный метод обеспечивает более высокую точность оценивания. Однако он проигрывает в точности при $\Delta/\sigma > 1,25$. Аналогично изменяются и отдельные составляющие ошибок. По затратам вычислительного времени нелинейное оценивание экономнее.

Список литературы: 1. Бернштейн А. В., Сидоров А. А. Оценка параметров нормальной совокупности по результатам групповых измерений.— Теория вероятностей и ее применение, 1971, т. XVI, вып. 2, с. 392—395. 2. Бернштейн А. В., Сидоров А. А. Оценки множества средних нормальной совокупности.— Теория вероятностей и ее применение, 1972, т. XVII, вып. 4, с. 768—773. 3. Патрик Э. Основы теории распознавания образов.— М.: Сов. радио, 1980.— 408 с.

Поступила в редколлегию 03.04.85.

УДК 681.14

Б. И. МОРОЗ, Е. В. АНИКИНА

О ВЫБОРЕ КРИТЕРИЯ ЭФФЕКТИВНОСТИ ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНОЙ СИСТЕМЫ ДЛЯ АВТОМАТИЗАЦИИ НАУЧНЫХ ИССЛЕДОВАНИЙ

Задачи, связанные с автоматизацией крупных исследовательских установок, ускорителей, радиотелескопов, гидрофизических судов достаточно актуальны. В работе [1] рассмотрены основные концепции создания микропроцессорных систем с изменяемой (гибкой) структурой для автоматизации научных исследований. Здесь же предложена типовая структурная схема автоматизированной системы научных исследований (АСНИ) на базе серийно выпускаемых технических средств. Сбор информации в такой системе выполняет подсистема нижнего уровня, которая по сути является информационно-измерительной системой (ИИС).

Использование приборного интерфейса на нижнем уровне (уровень связи с объектом) позволяет обеспечить простое сопряжение с серийно выпускаемыми контрольно-измерительными приборами, дает возможность экспериментатору гибко изменять структуру подсистемы по составу и номенклатуре оборудования, программному обеспечению. Разработка рассматриваемой ИИС АСНИ, как и любой технической системы, связана с получением формального вида критерия оценки возможных вариантов построения (критерия эффективности). В качестве таких критериев используются различного рода функционалы, которые в зависимости от целевого назначения системы могут определять экономические показатели, ее технические характеристики и свойства на отдельных этапах функционирования и т. д.

В статье получен формальный вид критерия эффективности ИИС АСНИ. Рассматривая ИИС АСНИ в классе сложных технических систем, правомерно применять общий подход к построению критерия эффективности, учитывая при этом специфические особенности. На этом основании запишем критерий эффективности в виде

$$\Phi = f(P(t), P_d, G, W, V, T, S). \quad (1)$$

Здесь $P(t)$ — функция надежности ИИС АСНИ; P_a — доверительная вероятность невыхода ошибки восстановления за пределы допуска (доверительного интервала); G — масса аппаратуры ИИС АСНИ; W — потребляемая мощность; V — габаритные размеры аппаратуры, $V = \{v_1, \dots, v_i, \dots, v_k\}$; T — время выполнения всех этапов преобразования информации, $T = \{t_1, \dots, t_i, \dots, t_l\}$; t_i — время выполнения i -го этапа преобразования информации; S — стоимость ИИС АСНИ, $S = \{s_1, \dots, s_i, \dots, s_k\}$; $G = \{g_1, \dots, g_i, \dots, g_k\}$, где g_i, v_i, s_i — масса, габаритные размеры и стоимость i -го элемента ИИС АСНИ.

Рассмотрим функцию (1) и входящие в нее варьируемые параметры. Справедливо утверждать, что при выборе явных зависимостей параметров, входящих в выражение (1), можно воспользоваться только эвристическими соображениями. В большинстве случаев так и поступают в подобных ситуациях. В качестве подтверждения можно привести работу [2]. Например, функцию (1) в целях получения конкретного вида критерия эффективности представляют в виде линейной зависимости

$$\Phi = c_1 P(t) + c_2 P_a + c_3 G + c_4 W + c_5 V + c_6 T + c_7 S, \quad (2)$$

где $c_1, \dots, c_7$ — весовые коэффициенты, или в виде

$$\Phi = \frac{P(t) P_a}{G W V T S}. \quad (3)$$

Формы записи (2), (3) эквивалентны [2]. Применять критерий (1) в таком виде для случая выбора оптимального варианта построения ИИС АСНИ затруднительно, а также малоэффективно. Во-первых, определение коэффициентов $c_1, \dots, c_7$, основанное на ряде эвристических процедур, представляется весьма проблематичной и к тому же слабоформализуемой задачей. Эти обстоятельства затрудняют оценку точности решения задачи оптимизации, что в нашем случае нежелательно. Во-вторых, разнотипность варьируемых параметров функции (1) по физическому смыслу, числовым значениям, степени важности и т. д. ставит под сомнение правомерность записи типа (2), (3) для ИИС, поскольку такие системы требуют точных и научно обоснованных расчетов. Исходя из изложенного, критерий эффективности приводим к виду

$$\Phi = P(t) P_a \rightarrow \max. \quad (4)$$

Здесь

$$P_a = \ln f \{P(e(t) < \sigma(t))\}, \quad (5)$$

где t — текущее время; $e(t)$ — ошибка восстановления измеряемой реализации; $\sigma(t)$ — допустимое значение ошибки восстановления; $P(e(t) < \sigma(t))$ — вероятность восстановления измеряемой реализа-

ции на приемном конце в пределах допуска. Если $\sigma(t) = \text{const}$ в пределах $0, T$, то

$$P_d = \inf \{P(\sigma(t) < \delta)\}. \quad (6)$$

Критерий эффективности вида (4) основан на том положении, что функции $P(t)$, P_d являются наиболее важными характеристиками для ИИС АСНИ. Остальные варьируемые параметры выделим в систему ограничений

$$G < G_{\text{доп}}; W < W_{\text{доп}}; V < V_{\text{доп}}; T < T_{\text{доп}}; S < S_{\text{доп}}. \quad (7)$$

Для определения функции надежности можно предложить достаточное количество методов, известных в литературе [3, 4]. Это позволит не останавливаться подробно на определении функции $P(t)$. Однако выбор метода нахождения $P(t)$ в значительной степени влияет на эффективность решения задачи оптимизации ИИС АСНИ. По мнению авторов, учет выводов, сделанных в работе [5], позволяет в необходимой мере исключить возможные ошибки при выборе метода определения $P(t)$. Для нахождения значения необходимо разработать набор специальных алгоритмов прогноза. Анализ существующих методов прогноза в целях применения в нашем случае показывает, что для достижения наибольшего эффекта прогноза целесообразно применять следующие методы [6]: прогноз по последнему значению (экстраполяция нулевого порядка); прогноз по математическому ожиданию; статистические прогнозы по одной, двум и более точкам.

Для случая экстраполяции нулевого порядка

$$e(t + \theta) = X(t + \theta) - \hat{X}(t + \theta) \quad (8)$$

или

$$e(t + \theta) = X(t + \theta) - X(t), \quad (9)$$

где $X(t)$ — измеряемый случайный параметр; $\hat{X}(t)$ — восстановленное (прогнозное) значение измеряемого случайного параметра; θ — время прогноза. Если прогнозировать по математическому ожиданию, то ошибка восстановления

$$e(\theta) = X(t + \theta) - m_x \quad (10)$$

(m_x — математическое ожидание случайного параметра). Статистический прогноз по одной точке дает следующую ошибку восстановления:

$$e(t + \theta) = Y - m_{y/x},$$

Здесь Y — восстановленное значение измеряемого случайного параметра; $m_{y/x}$ — условное математическое ожидание. Ошибка восстановления по двум и более точкам определяется аналогично случаю восстановления значения параметра по одной точке.

Таким образом, получен формальный вид критерия эффективности ИИС АСНИ и определены входящие в него параметры.

Список литературы: 1. Иванов В. А., Иванов В. В. Опыт построения микропроцессорных систем с изменяемой структурой для автоматизации научных исследований.— К.: Наук. думка, 1984.— 20 с. 2. Переверткин С. М. Некоторые проблемы оптимизаций информационных измерительных систем. Автоматическое управление и вычислительная техника.— М.: Машиностроение, 1975, вып. 11, с. 194—207. 3. Гнеденко Б. В., Беляев Ю. К., Соловьев А. Д. Математические методы в теории надежности.— М.: Наука, 1965.— 524 с. 4. Дружинин Г. В. Надежность устройств автоматики.— М.: Энергия, 1967.— 527 с. 5. Долгов М. А., Малаховская Н. Л. Об одном подходе к установлению вида обобщенного закона распределения потока отказов.— Повышение эконом. эффективности обществ. производства, 1975, вып. 2, с. 143—150. 6. Ефимов А. Н. Предсказание случайных процессов.— М.: Знание, 1976.— 64 с.

Поступила в редколлегию 26.02.85.

УДК 621.3.019

С. Ф. ТЕСЛЮК

ОЦЕНКА ПАРАМЕТРИЧЕСКОЙ УСТОЙЧИВОСТИ ЭВМ МЕТОДОМ ГРАНИЧНОГО КОНТРОЛЯ

В цифровых ЭВМ при эксплуатации возникают сбои и параметрические отказы. Последним обычно предшествует увеличение интенсивности сбоев. Сбои в ЭВМ появляются вследствие внутренних и внешних помех в сигнальных цепях и цепях питания, а также из-за недостаточности запаса помехоустойчивости цифровых микросхем по статическим и динамическим помехам. Помехоустойчивость связана с дестабилизирующим влиянием факторов среды (температура, влажность, ионизирующие излучения) и физико-химическими процессами, происходящими в микросхемах и конструкциях ЭВМ при эксплуатации. Поэтому важно не только оценивать правильность функционирования ЭВМ в момент контроля, но и определять запас параметрической устойчивости функционирования, характеризующий надежность ее работы по отношению к сбоям и параметрическим отказам. Систематическое определение запаса устойчивости позволит прогнозировать постепенные отказы и оптимально назначать профилактические работы.

В качестве критерия надежности ЭВМ по отношению к сбоям и параметрическим отказам будем использовать обобщенный показатель устойчивости $K_y(t)$. Под устойчивостью функционирования цифровых схем понимается комплексный запас статической и динамической помехоустойчивости, характеризующий их способность функционировать без сбоев и параметрических

отказов при возмущающих воздействиях в виде помех в различных цепях или изменениях параметров микросхем, снижающих статическую и динамическую помехоустойчивости.

На практике удобно рассматривать и оценивать отдельно статическую и динамическую помехоустойчивости.

Под статической устойчивостью понимается способность схемы работать без сбоев и отказов при статических помехах на входах и в случае изменений параметров, влияющих на запасы ее помехоустойчивости; под динамической устойчивостью — способность системы правильно функционировать при изменениях тактовой частоты и длительности тактовых сигналов. Статическая помехоустойчивость логических элементов определяется известными методами по уровням входных сигналов (с учетом помех) и по характеристике передачи [1]; динамическая устойчивость — с помощью передаточных импульсных характеристик и задержки распространения сигналов на уровне порога переключения [1, 2]. Выражения для показателей статической K_{cy} и динамической K_{dy} устойчивостей приведены в работе [3].

Из-за отсутствия доступа ко входам и выходам схем известные методы прямого определения K_{cy} , K_{dy} для отдельных функциональных частей или устройств ЭВМ не приемлемы. В связи с этим предлагается методика косвенного определения показателей устойчивости с помощью граничных испытаний.

Представим K_{cy} в общем виде:

$$K_{cy} = F_c(\bar{\alpha}, \bar{\beta}). \quad (1)$$

Здесь $\bar{\alpha}$ — вектор независимо изменяющихся параметров схем (изменение параметров компонент в результате старения, дестабилизирующих воздействий факторов среды); $\bar{\beta}$ — вектор параметров электрического режима, к которому относятся питающие напряжения микросхем, частота f , и длительность τ , тактовых сигналов. Аналогично

$$K_{dy} = F_d(\bar{\alpha}, \bar{\beta}). \quad (2)$$

Начальные значения компонентов вектора $\bar{\alpha}$ изменяются во времени под действием различных факторов, они имеют разброс в некоторых пределах:

$$\alpha_{i\min} < \alpha_i < \alpha_{i\max}.$$

Вектор общесхемных параметров $\bar{\beta}$ можно изменять относительно расчетного номинального режима в установленных из физических соображений пределах:

$$\beta_{j\min} < \beta_j < \beta_{j\max}.$$

Разложив функцию K_{cy} в ряд Тейлора в окрестности номиналь-

ной рабочей точки α_n, β_n и ограничившись членами первого порядка, получим

$$K_{cy} = K_{cy_n} + \sum_{i=1}^n \left(\frac{\partial K_{cy}}{\partial \alpha_i} \right)_n \Delta \alpha_{i_c} + \sum_{j=1}^l \left(\frac{\partial K_{cy}}{\partial \beta_j} \right)_n \Delta \beta_{j_c}. \quad (3)$$

Отклонение устойчивости от номинального режима характеризуется параметром

$$\Delta K_{cy} = K_{cy_n} - K_{cy} = \sum_{i=1}^n \left(\frac{\partial K_{cy}}{\partial \alpha_i} \right)_n \Delta \alpha_{i_c} + \sum_{j=1}^l \left(\frac{\partial K_{cy}}{\partial \beta_j} \right)_n \Delta \beta_{j_c}. \quad (4)$$

Аналогично для динамической устойчивости:

$$\Delta K_{dy} = \sum_{i=1}^n \left(\frac{\partial K_{dy}}{\partial \alpha_i} \right)_n \Delta \alpha_{i_d} + \sum_{j=1}^l \left(\frac{\partial K_{dy}}{\partial \beta_j} \right)_n \Delta \beta_{j_d}. \quad (5)$$

Здесь K_{cy_n}, K_{dy_n} — номинальные (начальные) значения показателя статической и динамической устойчивости; $\left(\frac{\partial K_{cy}}{\partial \beta_j} \right)_n, \left(\frac{\partial K_{dy}}{\partial \beta_j} \right)_n$ — коэффициенты влияния отдельных компонентов вектора $\vec{\beta}$ при отклонении их в окрестности номинальной рабочей точки. В начальном состоянии $\Delta \alpha_{i_c} = 0, \Delta \alpha_{i_d} = 0$. Поэтому при граничном контроле устойчивости в момент t_0 , реализуя равенство $K_{cy} = 0, K_{dy} = 0$ изменением $\vec{\beta}$, имеем

$$\Delta K_{cy} = K_{cy_n} - \sum_{j=1}^l \left(\frac{\partial K_{cy}}{\partial \beta_j} \right)_n \Delta \beta_{j_{cr}} [t_0]; \quad (6)$$

$$\Delta K_{dy} = K_{dy_n} - \sum_{j=1}^l \left(\frac{\partial K_{dy}}{\partial \beta_j} \right)_n \Delta \beta_{j_{dr}} [t_0], \quad (7)$$

где $\Delta \beta_{j_{cr}} [t_0], \Delta \beta_{j_{dr}} [t_0]$ — отклонения компонентов вектора $\vec{\beta}$ по антиградиентам K_{cy}, K_{dy} до границ устойчивого функционирования.

Обобщенный критерий устойчивости для текущего момента представим в виде

$$\begin{aligned} K_r [t_1] &= K_{cy_n} (1 - \delta K_{cy_n} [t_1]) K_{dy_n} (1 - \delta K_{dy_n} [t_1]) = \\ &= K_{yn} (1 - \delta K_{cy_n} [t_1]) (1 - \delta K_{dy_n} [t_1]). \end{aligned} \quad (8)$$

Для оценки $[t_1]$ при известном K_{yn} необходимо определить

$$\delta K_{cy} [t_1], \delta K_{dy} [t_1].$$

Из формул (3) — (7) получаем

$$\delta K_{cy}[t_1] = 1 - \frac{\sum_{j=1}^l \left(\frac{\partial K_{cy}}{\partial \beta_j} \right)_* \Delta \beta_{j_{cgp}}[t_1]}{\sum_{j=1}^l \left(\frac{\partial K_{cy}}{\partial \beta_j} \right)_* \Delta \beta_{j_{cgp}}[t_0]}; \quad (9)$$

$$\delta K_{xy}[t_1] = 1 - \frac{\sum_{j=1}^l \left(\frac{\partial K_{xy}}{\partial \beta_j} \right)_* \Delta \beta_{j_{xgp}}[t_1]}{\sum_{j=1}^l \left(\frac{\partial K_{xy}}{\partial \beta_j} \right)_* \Delta \beta_{j_{xgp}}[t_0]}. \quad (10)$$

Таким образом, для определения показателя устойчивости $K_y[t_1]$ при известных значениях коэффициентов влияния

$$\left(\frac{\partial K_{cy}}{\partial \beta_j} \right)_* \cdot \left(\frac{\partial K_{xy}}{\partial \beta_j} \right)_*$$

и начальных отклонениях $\Delta \beta_{j_{cgp}}[t_0]$, $\Delta \beta_{j_{xgp}}[t_0]$ необходимо в моменты t (γ) методом граничного контроля найти текущие граничные отклонения $\Delta \beta_{j_{cgp}}[t_1]$, $\Delta \beta_{j_{xgp}}[t_1]$, изменяя при этом компоненты вектора $\vec{\beta}$ до состояния $K_{cy}[t_1] = 0$, $K_{xy}[t_1] = 0$ соответственно. В этом случае возможны совместный статическо-динамический контроль устойчивости и раздельный контроль по одной из составляющей.

Реализация метода возникла на основе использования в ЭВМ микропроцессорного блока параметрической автодиагностики, который осуществляет граничный контроль, учитывает накопления и анализирует изменения K_y функциональных частей ЭВМ во времени, выдает рекомендации на профилактику и прогнозирует постепенные отказы, оптимизирует периодичность параметрического контроля и стратегию профилактики.

Благоприятные условия для реализации метода создаются применением децентрализованной системы питания ЭВМ с программируемыми вторичными источниками питания, а также построением ЭВМ на БИС и СБИС.

Список литературы: 1. Наунов Ю. Е., Агаев Н. А., Бедревский М. А. Помехоустойчивость устройств на интегральных логических схемах.— М.: Сов. радио, 1975.— 215 с. 2. Саложков К. А., Бершадский А. М., Пац В. Б. Выбор и применение систем логических элементов ЭВМ.— М.: Энергия, 1980.— 73 с. 3. Васильева Н. П., Петрухин Б. П. Проектирование логических элементов автоматики.— М.: Энергия, 1970.— 336 с.

Поступила в редакцию 24.12.84.

**ОБЪЕКТНО-ОРИЕНТИРОВАННЫЙ МЕТОД СИНТЕЗА
КОМПОНЕНТОВ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ**

Высокая стоимость и сложность структуры вычислительных систем (ВС), перераспределение функций между программными (ПС) и аппаратными средствами (АС), увеличение глубины микропрограммного управления АС, а также высокие требования к надежности их функционирования выдвигают задачу разработки методов комплексного проектирования, программных, микропрограммных и аппаратных средств в рамках единой методологии [1, 2]. Основным фактором увеличения сроков разработки и возникновения ошибок при синтезе ВС является сложность проектируемых объектов [2, 3]. Поэтому эффективная методика проектирования должна обеспечивать ограничение сложности объектов на каждом шаге проектирования. При проектировании сложных систем широко используется аналитическая стратегия проектирования, в основе которой лежит принцип от абстрактного к конкретному. Наиболее слабо формализованным этапом синтеза является декомпозиция объектов проектирования.

В данной статье описан метод синтеза компонентов ВС, обеспечивающий в рамках единой объектно-ориентированной модели отражение формальными методами процесса переработки информации на ПС, АС и микропрограммные средства.

Декомпозиция процесса переработки информации. При разработке многоуровневого алгоритмического представления процесса переработки информации в качестве модели его реализации целесообразно использовать концепцию понижения уровня абстракции функционирования многоуровневых виртуальных машин [4]. Аппаратную реализацию ВС можно рассматривать как совокупность двух взаимодействующих объектов: устройства управления и объекта управления [5].

В основе предлагаемого метода лежит разработка объектно-ориентированного описания на базе декомпозиции виртуальных объектов управления (ВОУ). Модели их функционирования представляются на каждом уровне декомпозиции в виде виртуального управляющего объекта (ВУО) данного уровня и ВОУ низшего уровня.

Сущность процесса декомпозиции на управляющую (ВУО) и управляемую части (ВОУ) состоит в следующем. Исходное деление производится путем разработки алгоритмического описания процесса переработки информации первого шага (этапа) синтеза в соответствии с выбранной стратегией. После этого осуществляется его преобразование к объектно-ориентированной форме в результате объединения на различных уровнях абстракции элементов данных и операторов их преобразования в объекты с дополнительными точками входа, которые соответствуют требуемым преобразованиям элемента данных. Заданная последовательность вызовов объектов реализуется в порядке их

естественного следования, а также посредством задания на множестве W вызовов отношений $R^{IF} = \{R_i^{IF} | R_i^{IF} \subset W \times W\}$ селективного следования с помощью конструкции «выбор» и отношения $R^{DO} = \{R_i^{DO} \subset W \times W\}$ циклического следования с помощью конструкции повторения. Предикаты конструкций задаются на множестве элементов данных соответствующего уровня абстракции алгоритма. Объекту, заданному процедурой, ставится в соответствие функциональный блок; множеству входных и выходных параметров процедуры — множество информационных входов и выходов блока; точкам входа и выхода из процедуры — входные и выходные управляющие сигналы, управляющим конструкциям — их стандартная реализация функциональными блоками. Отдельные блоки могут быть виртуальными (физически не реализуемыми). Введение таких блоков позволяет упростить понимание и модификацию функциональной схемы, а также доказывать ее корректность формальными методами. Множество вызовов и конструкций образуют ВУО, а множество объектов — ВОУ. В дальнейшем каждой функции преобразования данных каждого объекта ставится в соответствие ее алгоритмическая детализация. Каждая алгоритмическая детализация преобразуется к объектно-ориентированной форме. Объединение полученных множеств вызовов и конструкций образует ВУО, а объединение множеств полученных объектов — ВОУ низшего уровня.

Процесс декомпозиции ВОУ выполняется до достижения такого уровня спецификаций, который позволяет осуществлять прямую интерпретацию их заданными аппаратными и программными средствами.

Приведем алгоритмические описания и их объектно-ориентированные формы для различных уровней абстракции процесса нахождения наибольшего элемента массива.

УРОВЕНЬ I:

АЛГОРИТМ:

```
DCL (A (10), MAX) BIN FIXED (15,0);
CALL O_M (A, MAX);
O_M: P ROC (A, MAX);
/*НАЙТИ МАКСИМАЛЬНЫЙ*/
END O_M;
```

ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ ФОРМА:

```
DCL (A (10), MAX) BIN FIXED (15,0);
CALL MO (A, MAX);
O_M: PROC (A, MAX);
MO: ENTRY;
/*НАЙТИ МАКСИМАЛЬНЫЙ*/
RETURN; /* RO */ END O_M;
```

УРОВЕНЬ 2:

АЛГОРИТМ:

```

DCL (A (10), MAX) BIN FIXED (15,0);
CALL O_M (A, MAX);
O_M: PROC (A, MAX);
  DCL (A (10), MAX, I) BIN FIXED (15,0);
  I=1;
  MAX=A (I);
  I=2;
  DO WHILE (I = 10);
    IF A (I) > MAX THEN
      MAX=A (I);
    /*ENDIF*/
    I=I+1;
  END; /*ENDW*/
END O_M;

```

ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ ФОРМА:

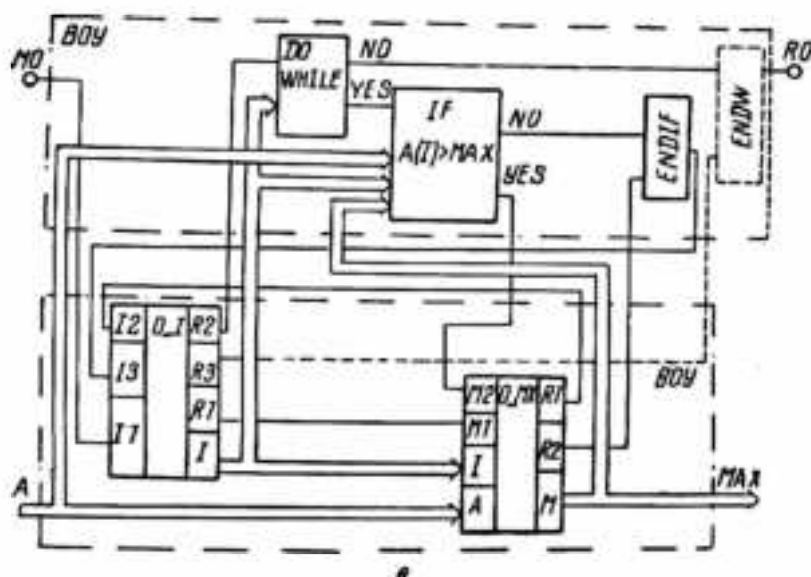
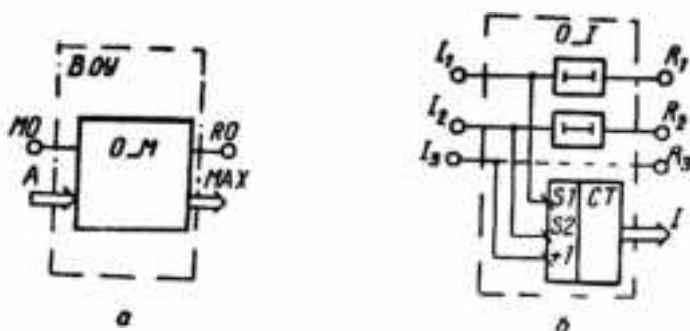
```

DCL (A (10), MAX) BIN FIXED (15,0);
CALL MO (A, MAX);
O_M: PROC (A, MAX); DCL I BIN FIXED (15,0);
MO: ENTRY; CALL I1 (I);
      CALL M1 (A, I, MAX);
      CALL I2 (I);
DO WHILE (I <= 10);
  IF A (I) > MAX THEN
    CALL M2 (A, I, MAX);
  /*ENDI1*/
  CALL I3 (I);
END; /*ENDW*/
RETURN; /*RO*/
O_I: PROC (I);
  DCL I BIN FIXED (15,0);
  I1: ENTRY; I=1; RETURN; /*R1*/
  I2: ENTRY; I=2; RETURN; /*R2*/
  I3: ENTRY; I=I+1; RETURN; /*R3*/
END O_I;
O_MX: PROC (A, I, MAX);
  DCL (A (10), I, MAX) BIN FIXED (15,0);
  M1: ENTRY;
  M2: ENTRY;
  MAX=A (I);
  RETURN; /*R1, R2*/
END O_MX;
END O_M;

```

На рисунке, позиции *a*, *б*, изображены исходный ВОО и его декомпозиция на ВУО и ВОУ низшего уровня. Позиция *в* соответствует реализации элемента ВОУ стандартными узлами. Пунктиром обозначены виртуальные блоки и связи. Приведенный ВУО можно реализовать микропрограммным устройством управления или аппаратно.

Предложенный подход к проектированию ВС позволяет комплексно разработать ВС на основе рационального сочетания программных и аппаратных средств. Достоинством подхода является его общность, обусловленная независимостью от конк-



Исходный BOU и его декомпозиция

ретной технической реализации ВС, упорядоченность и приспособленность к модификациям получаемых структур, а также возможность сквозного контроля их корректности. Это позволяет повысить надежность ВС. Кроме того, с помощью разработанного метода можно отражать полную иерархию абстракций функционирования ВС на программные средства, программируемые устройства или другие аппаратные средства в пропорциях, обеспечивающих заданную эффективность ВС, и доказывать корректность реализации формальными методами. Результаты работы использованы при разработке ВС на микропроцессорах, они могут применяться при проектировании ВС широкого класса.

Список литературы: 1. Рабинович З. Л. Машинный интеллект и структуры ЭВМ пятого поколения.— Кибернетика, 1984, № 3, с. 96—107. 2. Додонов А. Г., Кузнецова М. Г. Об организации программного обеспечения вычислительных структур повышенной живучести.— В кн.: Техническая диагностика электронных систем. К.: Наук. думка, 1982, с. 17—22. 3. Холетед М. Х. Начала науки о программах: Пер. с англ. / Под ред. В. М. Юфы.— М.: Финансы и статистика, 1981.— 128 с. 4. Дал У., Дейкстра Э., Хоор К. Структурное программирование: Пер. с англ.— М.: Мир, 1975.— 248 с. 5. Рабинович З. Л. Структура и развитие процесса переработки информации в ЭВМ.— Кибернетика, 1983, № 4, с. 46—53.

Поступила в редколлегию 10.01.85.

УДК 681.021

В. М. ВОРОНА, А. Г. МУРАШКО, канд. техн. наук,
А. В. АНДРЕЕВА

АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ПОДСИСТЕМЫ ПРОЕКТИРОВАНИЯ СПЕЦИАЛИЗИРОВАННОЙ ВЫЧИСЛИТЕЛЬНОЙ СТРУКТУРЫ

При решении инженерно-технических задач с помощью вычислительной техники (аналоговой, цифровой интегрирующей, микропроцессорной, гибридной) необходимо предварительно задавать стратегии проведения эксперимента [1]. Основные из них следующие.

1. Каждая i -я задача решается за время τ_{i2} с погрешностью δ_{i2} :

$$\tau_i < \tau_{i2}; \delta_i < \delta_{i2},$$

где τ_i , δ_i — время и погрешность решения i -й задачи; τ_{i2} , δ_{i2} — заданные время и погрешность решения.

2. Каждая i -я задача решается с погрешностью $\delta_i < \delta_{i2}$. В этом случае необходимо обеспечить минимально возможное время решения $\tau_{i \min}$ при погрешности δ_{i2} :

$$\delta_i < \delta_{i2}; \tau_i < \tau_{i \min}.$$

3. Каждая i -я задача решается за время $\tau_i < \tau_{i2}$. Следует обеспечить минимально возможную погрешность $\delta_{i \min}$ при времени τ_{i2} :

$$\tau_i < \tau_{i2}; \delta_i < \delta_{i \min}.$$

4. Время и погрешность решения i -й задачи не должны выходить за заданные пределы:

$$\tau_{i2} < \tau_i < \tau_{i2}; \\ \delta_{i2} < \delta_i < \delta_{i2}.$$

При решении задач оптимизации большой размерности [1], связанных с системами обыкновенных дифференциальных и нелинейных алгебраических уравнений, возникает проблема

рационального выбора вычислительной структуры (ВС) высокой производительности из класса однородных вычислительных сред (ОВС). Ставится задача синтеза структуры ОВС, удовлетворяющей требованиям точности, быстродействия решения задач оптимизации и гибкости управления вычислительным процессом. Отметим, что время решения задачи и ошибки в результатах вычислений, влияющие на точность, зависят от всех частных технических параметров вычислительной системы. В этом смысле время решения задачи и точность можно признать обобщенными показателями качества вычислительной структуры. В качестве обобщенного показателя эффективности ВС примем стоимость эксперимента (C_e), которая, как правило, не оптимизируется, а играет роль ограничивающего фактора.

Анализ показывает, что, находясь в рамках первой стратегии, сложно достигать заданных значений τ , δ . Ограничивающий фактор (стоимость эксперимента) играет здесь пассивную роль. Регулирование процесса эксперимента с позиций временных затрат и достижения заданной точности является сложной задачей.

Вторая и третья стратегии предполагают, что фактор стоимости эксперимента будет играть активную ограничивающую роль, направленную в процессе регулирования τ , δ на установление «разумных минимально возможных значений» погрешности и времени решения для заданных $\delta_{1/2}$, $\tau_{1/2}$ соответственно.

Рассматривая четвертую стратегию эксперимента, где критерий C_e также имеет важное значение, отмечаем, что при двух «закрытых границах» по параметрам τ , δ стратегия регулирования будет существенно зависеть от ширины полос $\pm \Delta\tau$, $\pm \Delta\delta$. Этот случай наиболее распространен в практике инженерных расчетов.

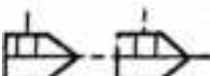
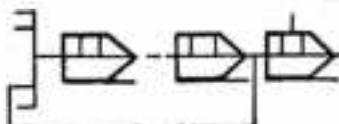
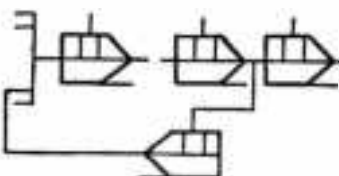
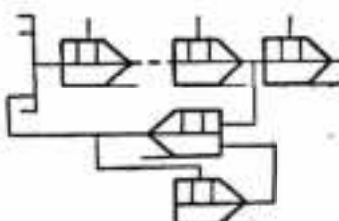
Какие средства моделирования следует выбрать для того, чтобы удовлетворить условиям заданной стратегии? Без тщательного анализа ответить на этот вопрос невозможно. Математические модели динамических систем можно реализовать на аналоговых, цифровых (интегрирующих), микропроцессорных, гибридных (аналого-цифровых), структурно-алгоритмических моделях (системах). Поэтому в каждом случае моделирования возникает задача выбора моделирующей вычислительной структуры. От правильного решения этой задачи во многом зависят суммарный эффект от применения вычислительной техники, результативности эксперимента и внедрения.

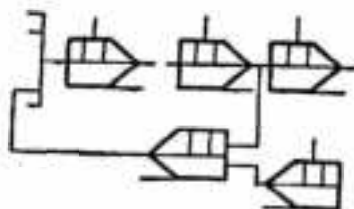
Задача выбора должна решаться системой имитационного моделирования автоматически, путем синтеза и различных вычислительных структур.

Рассмотрим этот процесс подробнее. Пусть есть поток задач — математических описаний решаемых задач (МОРЗ). Для каждой задачи необходимо построить вычислительную структуру, исходя из заданной стратегии эксперимента. Для того чтобы вычислительная модулирующая система осуществ-

для синтеза структур по заданному математическому описанию (МО) задачи, необходимо создать библиотеку алгоритмов отображения МО в вычислительные структуры. В зависимости от требований пользователя структуры могут быть аналоговыми, если требования по точности невелики; цифровыми, когда пользователь не предъявляет высоких требований по быстродействию; гибридные, в случае необходимости компромисса между точностью и временем.

Синтез ВС начинается с представления МО решаемой задачи на входном языке Аналог [2]. Полученная таким образом программа преобразуется в запись на некотором промежуточном языке, удобном для дальнейшей внутримашинной обработки. В результате грамматического разбора данной программы формируется ряд таблиц с информацией об элементах программы и С-структура, используемая непосредственно для построения решающей структуры. С-структурой может быть либо двоячное дерево, либо польская инверсная запись (ПОЛИЗ). Решающая структура получается в результате обработки соответствующим блоком С-структуры и таблиц данных. Этот блок снабжен библиотекой частных алгоритмов построения ВС, которые позволяют перейти от С-структур и таблиц к решающим структурам.

α_i	Цифровая базисная структура	Семантическая программа
0		$C1: 0,13 \quad 13 (i + 1) 01$
const, $ \alpha_i = 1$		$CC1: 013/ \quad 13 (i + 1) 011301/$
const, $ \alpha_i < 1$		$CCC1: 0,13/ \quad 13 (i + 1) 0111m02$ $CCC2: 0,11m1301/$
const, $ \alpha_i > 1$		$CCCC1: 0,13/ \quad 13 (i + 1) 0112s02$ $CCCC2: 0,12s1301/11m02$ $CCCC3: 0,11m13s03$

$a_i(t)$


CCCCC1 : 0,13/ 13 (i +
+ 1) 01 15 q 01
CCCCC2 : a, pr 15 q 02
CCCCC3 : 0,15 q 13 01/

Укрупненная схема синтеза ВС представлена на рис. 1. Для реализации синтеза используется метод непосредственной трансляции [3]. Метод основан на применении семантических программ, составленных на формальном языке для так называемых базисных структур [4]. Из этих программ по определенному алгоритму строится формальное описание вычислительной структуры, представляющее собой программу коммутации. Последняя

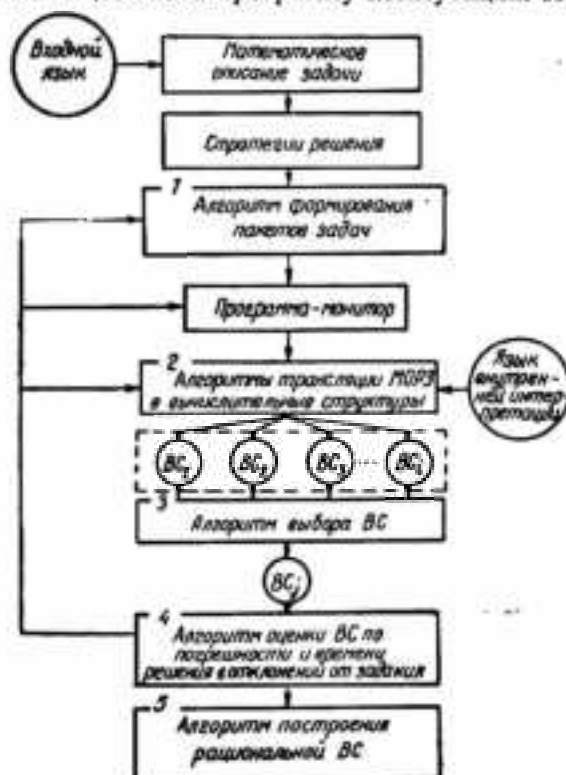


Рис. 1. Структура алгоритмов синтеза ВС

составлена на языке внутренней интерпретации операционных элементов и стандартных процедур вычислительной системы.

Рассмотрим процесс синтеза ВС на примере отображения однородного линейного дифференциального уравнения в цифровую интегрирующую структуру.

При построении ВС для математического описания в виде

$$y^{(n)} = -\alpha_1 y^{(n-1)} - \dots - \alpha_{n-1} y^{(1)} - \dots - \alpha_n y, \quad (1)$$

которое полностью определяется числами n (порядок уравнения) и α_i (коэффициенты), выполняются два основных этапа: составление цепи понижения порядка производной; образование обратных связей в схеме.

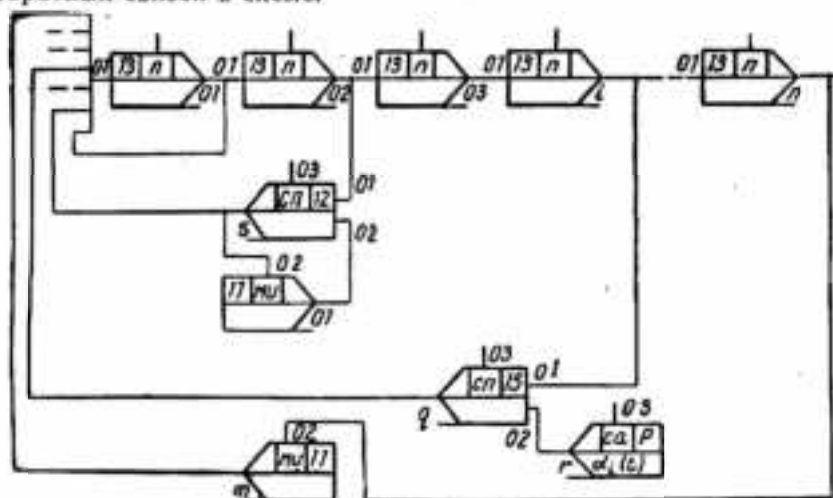


Рис. 2. Цифровая интегрирующая структура для однородного линейного дифференциального уравнения

На рис. 2 в общем виде представлена цифровая интегрирующая структура, построенная для уравнения (1). В таблице даны основные семантические программы, полученные путем анализа цифровых базисных структур. Для рассматриваемого примера выделено лишь пять базисных структур, используя которые совместно со специальным алгоритмом отображения (рис. 3), можно скомпоновать требуемую интегрирующую структуру. Количество цифровых базисных структур возрастает по мере усложнения математического описания решаемой задачи. Однако принципы построения алгоритмов отображения, использующие стандартные семантические программы, неизменны. Их легко автоматизировать с помощью цифровых вычислительных машин.

Аналогично можно построить аналоговую и аналого-цифровую ВС. Автоматизируется не только процесс их построения, но

оценка и выбор ВС, удовлетворяющий пользователя по параметрам τ , δ .

Данный пример частично иллюстрирует работу предлагаемой системы автоматизированного синтеза вычислительной структуры, которая по требованию пользователя в соответствии с заданной стратегией эксперимента синтезирует вычислительные структуры различных типов, оценивает их по точности и быстродействию, осуществляя выбор рациональной структуры. Если по каким-то причинам структура не устраивает пользователя, он может выбрать другую, также предложенную системой.

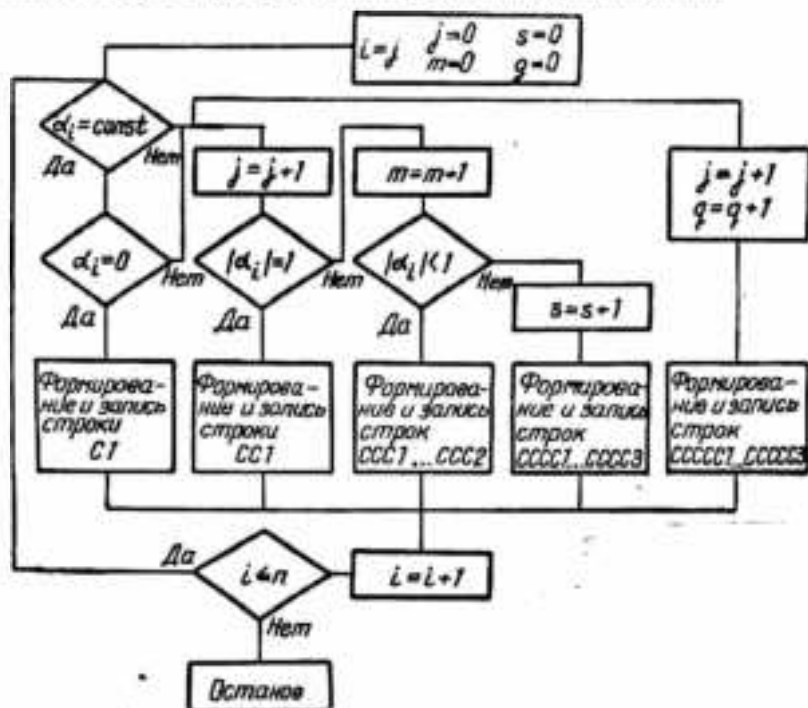


Рис. 3. Алгоритм отображения математического описания в вычислительную структуру

Эту систему можно рассматривать в качестве подсистемы САПР. Она относится к классу проектирующих инвариантных подсистем. Ее можно использовать при проектировании любых объектов, поддающихся описанию дифференциальными, интегральными или интегро-дифференциальными уравнениями, системами уравнений.

Список литературы: 1. Красногорова В. С., Мурашко А. Г., Сенченко Н. И. Структуры и алгоритмы настройки специализированной гибридной вычислительной среды. — Тез. докл. IV Всесоюз. конф. по пробл. однородных вычислит. систем и сред. К.: Наук. думка, 1975. — 45 с. 2. Дюбо Г. Ф.,

Мурашко А. Г., Терентьев М. Ф., Язык Аналог для описания задач, решаемых на аналоговых и аналого-цифровых вычислительных структурах. — В кн.: Неоднородные вычислительные системы. М., 1975.—137 с. 3. Гребенник Л. А., Дюбо Г. Ф., Мурашко А. Г. Отображение математического описания решаемой задачи в решающую структуру. — В кн.: Автоматизация программирования средств АВТ. М., 1971.—112 с. 4. Мурашко А. Г., Сенченко Н. И., Терентьев М. Ф. Об одном способе формального описания решающих структур. — В кн.: Аналоговая и аналого-цифровая вычислительная техника /Под ред. В. Б. Ушакова.—М.: Сов. радио, 1973.—95 с.

Поступила в редколлегию 15.03.85.

УДК 621.3

А. И. БЛИНОВ, В. П. КАЛЯВИН, А. В. МАЗИН,
А. М. МАЛЫШЕВ

ОПТИМИЗАЦИОННАЯ ПОСТАНОВКА ЗАДАЧ ОБОСНОВАНИЯ ТРЕБОВАНИЙ К СИСТЕМАМ ДИАГНОСТИРОВАНИЯ ОБЪЕКТОВ УПРАВЛЕНИЯ

Первый этап построения системы диагностирования (СД) объекта управления, который для обслуживающих его технических средств диагностирования (ТСД) является объектом диагностирования (ОД), заключается в обосновании требований, предъявляемых к ОД, ТСД. Удовлетворив эти требования, можно организовать процесс диагностирования в соответствии с условиями эксплуатации ОД. При организации СД по показателю готовности ОД P_r [1] построение систем диагностирования объектов управления можно сформулировать как решение определенной совокупности задач обоснования требования к СД [2]. Прежде чем решать эти задачи, необходимо дать их математическую постановку.

Приведем математические постановки задач, перечисленных в работе [2]. Обозначим через $X_1, \dots, X_i, \dots, X_k, \dots, X_l, \dots, X_n$ показатели, характеризующие организацию системы диагностирования. В этом случае общая постановка перечисленных в работе [2] задач обоснования требований к СД формулируется следующим образом.

Известны (заданы) значения нескольких характеризующих организацию СД показателей $X_i, i = \overline{1, k}$. Следует так выбрать значения остальных показателей $X_j, j = \overline{k+1, n}$, чтобы показатель готовности ОД

$$P_r = P_r(X_1, \dots, X_i, \dots, X_k, \dots, X_j, \dots, X_n)$$

достигал максимального $P_{r \max}$ (случай 1) и заданного P_{r*} (случай 2) значений.

Опишем математические постановки задач обоснования требований к СД.

Случай 1. Найти

$$\max_{\{X_i\} \in L_1} \Pi_r(X_1, \dots, X_i, \dots, X_k, \dots, X_j, \dots, X_n) \quad (1)$$

при ограничениях L_1 :

$$\begin{aligned} X_i^* &< X_i < X_i^* \\ &\dots \dots \dots \\ X_{n-2}^* &< X_{n-2} < X_{n-2}^* \\ 0 &< X_{n-1} < 1; \\ 0 &< X_n < 1, \end{aligned} \quad (2)$$

где $X_i^*, \dots, X_{n-2}^*, X_n^*, \dots, X_{n-2}^*$ — нижние и верхние граничные (допустимые) значения показателей $X_i, \dots, X_{n-2}$.

Случай 2. Решить уравнение

$$\Pi_r(X_1, \dots, X_i, \dots, X_k, \dots, X_j, \dots, X_n) = \Pi_{r_0} \quad (3)$$

при ограничениях (2) для определения значений X_j .

При конкретном задании показателей X_i может существовать по крайней мере одна комбинация, которая является решением задачи (3), (2). Прямой перебор с дискретными шагами по искомым показателям X_i не всегда обеспечивает требуемую точность решения задачи. Если число показателей X_i $s > 1$, целесообразно рассматривать задачу (2), (3) как оптимизационную. Однако такой подход предполагает введение некоторой целевой функции, отражающей в математической форме цель оптимизации проектируемой СД. Такая функция позволит из множества допустимых вариантов построения СД выбрать оптимальный. Чтобы найти эту функцию, воспользуемся следующими рассуждениями. Обозначим через $X = [X_1, \dots, X_i, \dots, X_k, \dots, X_j, \dots, X_n]^T$ вектор-столбец, принадлежащий n -мерному евклидову пространству $|R^n: X = [X_1, \dots, X_i, \dots, X_k, \dots, X_j, \dots, X_n]^T \in |R^n$.

Рассмотрим функцию

$$\Pi_r = \Pi_r(X_1, \dots, X_i, \dots, X_k, \dots, X_j, \dots, X_n) = \Pi_r(X),$$

множеством задания которой является $|R^n$, а множеством значений — вещественная прямая (вещественные числа) $|R$, т. е. $|R^n \rightarrow |R$. Уравнение

$\Pi_r(X_1, \dots, X_i, \dots, X_k, \dots, X_j, \dots, X_n) = \Pi_r(X) = \Pi_{r_0} = \text{const}$ задает в n -мерном пространстве $|R^n$ $(n-1)$ -мерную поверхность. Пусть

$$X_0 = [X_1, \dots, X_i, \dots, X_k, \dots, X_j^0, \dots, X_n^0]^T \in L_1$$

— вектор, доставляющий максимум функции

$$\Pi_r = \Pi_r(X_1, \dots, X_i, \dots, X_k, \dots, X_j, \dots, X_n) = \Pi_r(X),$$

т. е. решение задачи (1), (2) следующее: $\Pi_r(X_0) = \max_{X \in L_1} \Pi_r(X)$.

Через $X = [X_1, \dots, X_h, \dots, X_k, \dots, \hat{X}_h, \dots, \hat{X}_k]^T \in L_1$ обозначим решение задачи (3), (2), а именно $\Pi_r(\hat{X}) = \Pi_{gr}$. Значения $X_1^0, \dots, X_n^0$ показателей $X_1, \dots, X_n$ являются наилучшими, если учесть готовность ОД. Однако по условию задачи в случае 2 достаточно обеспечить заданный уровень Π_{gr} готовности ОД. В связи с этим необходимо перевести уравнения (3) $\Pi_r(X) = \Pi_{gr}$ в число ограничений. В качестве целевой функции целесообразно использовать функцию близости векторов (точек) $X_0, \hat{X}$ и обеспечить в процессе проектирования СД ее минимум. При такой целевой функции решения задач $X_0, \hat{X}$ (случаи 1, 2) в пределах Π_{gr} будут наиболее близкими. При этом наблюдается некоторое ухудшение каждого из искоемых показателей $X_1, \dots, X_n$ по сравнению с $X_1^0, \dots, X_n^0$ соответственно. Однако такое ухудшение распределяется по всему множеству искоемых показателей X_i , кроме того, оно является минимально возможным.

Таким образом, понятие близость необходимо придать строгий математический смысл. Понятие близости двух векторов

$$U = [U_1, \dots, U_n]^T \in R^n, V = [V_1, \dots, V_n]^T \in R^n$$

в n -мерном пространстве $|R^n$ количественно можно выразить с помощью функции расстояния $d_{2n}(U, V)$ между этими векторами. В этом случае постановка рассматриваемой задачи сводится к описанию функции $d_{2n}(U, V)$, множеством задания которой является пара n -мерных векторов U, V , а множеством значений — вещественная прямая (вещественные числа) $|R$, т. е.

$$|R^n \times |R^n \rightarrow |R.$$

Функция $d_{2n}(U, V)$ должна удовлетворять следующим условиям: для всех

$$U = [U_1, \dots, U_n]^T \in R^n; V = [V_1, \dots, V_n]^T \in R^n$$

справедливо

$d_{2n}(U, V) \geq 0$, т. е. расстояние между векторами U и V неотрицательно;

$d_{2n}(U, V) = 0 \Leftrightarrow U = V$, т. е. два вектора U и V совпадают по длине и направлению тогда и только тогда, когда расстояние между ними равно нулю;

$d_{2n}(U, V) = d_{2n}(V, U)$, т. е. расстояние между векторами U и V обладает свойством симметрии (расстояние от U до V равно расстоянию от V до U).

Поскольку на практике, как правило, метризуемые пространства нормируемы, целесообразно считать, что $d_{2n}(U, V) = d_n(U - V)$, где d_n — функция, множеством задания которой является n -мерное евклидово пространство $|R^n$, а множеством значений — вещественная прямая (вещественные числа) $|R$, т. е. $|R^n \rightarrow |R$. Отметим, что $d_n(\lambda X) = |\lambda| d_n(X)$, $\forall X \in |R^n$, $\forall \lambda \in |R$ (λ — скаляр-

ный множитель). Это свойство отражает независимость близости двух векторов от их положения в пространстве $|R^n$.

Определяя понятие близости, считаем, что для функции $d_{2n}(U, V)$ выполняется неравенство треугольника, т. е.

$$d_{2n}(U, V) \leq d_{2n}(U, W) + d_{2n}(V, W), \quad \forall U, V, W \in |R^n.$$

Следовательно, для функции $d_n(X)$ справедливы такие утверждения:

$$d_n(X + Y) \leq d_n(X) + d_n(Y), \quad \forall X; \quad (4)$$

$$d_n(X) \geq 0; \quad (5)$$

$$d_n(X) = 0 \rightarrow X = 0; \quad (6)$$

$$d_n(\lambda X) = |\lambda| d_n(X), \quad \forall \lambda \in R. \quad (7)$$

Таким образом, функция $d_n(X)$ удовлетворяет аксиомам (4) — (7) нормы в n -мерном евклидовом пространстве $|R^n$. В этом случае функция $d_{2n}(U, V)$ представляет собой расстояние в пространстве $|R^n$, порожденное нормой $d_n(X)$.

Наиболее распространенными нормированными n -мерными пространствами являются пространства l_n^p , представляющие собой множество векторов $X = [X_1, \dots, X_n]^T \in |R^n$, норма которых определяется так:

$$\|X\|_{l_n^p} = \left(\sum_{i=1}^n |X_i|^p \right)^{\frac{1}{p}}.$$

Здесь $n \in |N$; $1 \leq p < \infty$; $|N$ — множество натуральных чисел. Физический смысл числа p состоит в задании формы множества векторов X таких, что $\|X\|_{l_n^p} \leq 1$ в $|R^n$. Это множество геометрически представляет собой шар единичного радиуса в пространстве l_n^p . Например, для $X = [X_1, X_2]^T$ при $p = 1$ шар имеет форму квадрата со сторонами, наклоненными к осям X_1, X_2 под углом 45° (рисунок, позиция а), $p = 2$ — форму круга (позиция б), $p = \infty$ — форму квадрата со сторонами, параллельными осям X_1, X_2 (позиция в).

Функцию $d_n(X)$ удобно рассматривать в пространстве l_n^2 в виде

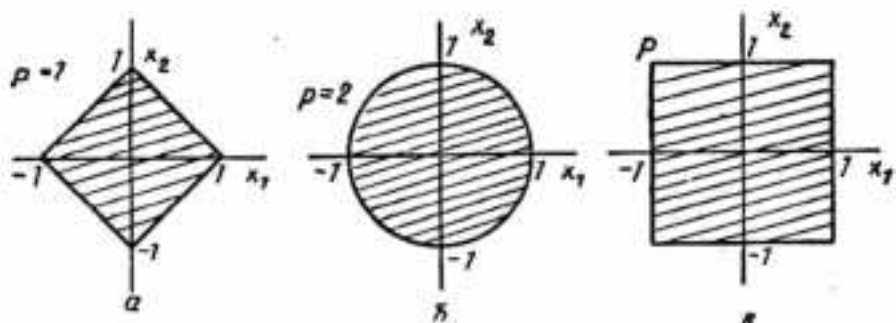
$$d_n(X) = \left(\sum_{i=1}^n |X_i|^2 \right)^{\frac{1}{2}}. \quad (8)$$

Это объясняется тем, что пространство l_n^2 , являясь обычным n -мерным евклидовым пространством, позволяет описать расстояние с помощью методов аналитической геометрии. При вычислениях в l_n^2 можно воспользоваться большим количеством способов, что позволит уменьшить сложность вычислений нормы вектора

$$X = [X_1, \dots, X_n]^T \in |R^n.$$

Понятию близости двух векторов $X_0, \dot{X}$ придадим количественный смысл. Принято, что вектор $\dot{X}$ ближе к вектору X_0 , чем вектор $\dot{Y}$, если $d_n(\dot{X} - X_0) < d_n(\dot{Y} - X_0)$.

В этом случае рассматриваемая задача сводится к нахождению на поверхности, задаваемой уравнением $\Pi_r(X) = \Pi_{r_0}$, вектора, наиболее близкого к вектору X_0 , доставляющему максимум функ-



Единичный шар в пространстве l_1^2 (a), l_2^2 (б), l_∞^2 (в)

ции $\Pi_r = \Pi_r(X)$. Значения показателей $X_{k+1}, \dots, X_{l_1}, \dots, X_n$, характеризующих организацию системы диагностирования объекта управления, вследствие физической природы этих показателей могут различаться на несколько порядков. Поэтому целесообразно изменить вид функции (8), отмасштабировав показатели

$$X_{k+1}, \dots, X_{l_1}, \dots, X_n.$$

Тогда искомой целевой функцией будет функция

$$r(X) = \left[\sum_{j=k+1}^{n-2} \left(\frac{X_j - X_j^0}{X_j^0} \right)^2 + X_{n-1}^2 + X_n^2 \right]^{\frac{1}{2}}. \quad (9)$$

Таким образом, задачу (3), (2) в случае 2 можно свести к следующей оптимизационной задаче. Найти

$$\min_{\{X_j\} \in L_r} r(X) = \left[\sum_{j=k+1}^{n-2} \left(\frac{X_j - X_j^0}{X_j^0} \right)^2 + X_{n-1}^2 + X_n^2 \right]^{\frac{1}{2}} \quad (10)$$

при ограничениях $L_{1*}: X_j^* < X_j < X_j^b; X_{n-2}^* < X_{n-2} < X_{n-2}^b; 0 < X_{n-1} < 1; 0 < X_n < 1; \Pi_r(\bar{X}) = \Pi_{r_0}$ (11).

Задача оптимизации (10), (11) универсальна, поскольку ее можно применять к любой из задач обоснования требований к СД, описанных ранее [2]. Она принципиально не зависит от количества искомых показателей X_j , значения которых следует определять, а также от конкретных требований, предъявляемых

к каждому из показателей X_i , т. е. от необходимости максимизировать или минимизировать данный показатель, входящий в вектор X .

Список литературы: 1. Мозгалецкий А. В., Каляпин В. П., Малышев А. М. Показатель готовности — критерий оптимизации систем диагностирования объектов управления. — В кн.: Электронные системы управления и контроля летательных аппаратов. Уфа: Б.и., 1980, вып. 5, с. 54—61. 2. Каляпин В. П., Малышев А. М., Мозгалецкий А. В. Формирование требований к системе диагностирования РЭА. — В кн.: Конструирование и управление качеством радиоаппаратуры. Л.: Б.и., 1983, вып. 163, с. 51—54.

Поступила в редколлегию 03.04.85.

УДК 621.397.6

В. В. СВИРИДОВ, д-р техн. наук, Д. К. МИХНОВ

ИССЛЕДОВАНИЕ РАЦИОНАЛЬНЫХ МЕТОДОВ ПЕРЕРАБОТКИ ЧАСТОТНЫХ СИГНАЛОВ ДЛЯ ОЦЕНКИ ВЫСОТЫ МУЗЫКАЛЬНЫХ ЗВУКОВ

Решения задач о переработке звуковых сигналов широко применяются в системах обработки информации о вибрационных характеристиках объектов, при цифровом анализе речевых и музыкальных сигналов. Обобщенными показателями исходной информации являются динамические, высотные и спектральные характеристики звуковых сигналов.

В статье рассмотрены методы переработки информации о высотных характеристиках музыкальных звуковых сигналов. Первичной информацией для системы цифровой обработки является акустический звуковой музыкальный сигнал, преобразуемый с помощью датчика (микрофона) в его электрический эквивалент. Остановимся на свойствах, характеризующих музыкальный сигнал. Любой музыкальный сигнал, как сигнал гармонический, можно описать выражением [1]

$$f_x(t) = C + A_1 \cos(2\pi f_{xk}t + \varphi_1) + A_2 \cos(2\pi 2f_{xk}t + \varphi_2) + \dots + A_x \cos(2\pi Zf_{xk}t + \varphi_x) + \dots, \quad (1)$$

где C — некоторая константа; A_x — амплитуды гармоник составляющих сигнала; f_{xk} — частота основной гармоники; φ_x — фазы гармоник составляющих сигнала.

Частотой f_{xk} определяется высота основного тона сигнала; соотношения амплитуд гармоник $A_1, A_2, \dots, A_x, \dots$ — спектральная

характеристика (в совокупности с соотношениями фаз — тембр музыкального сигнала); зависимость сигнала от времени t указывает на его динамический характер.

Высота звукового сигнала — качество, характеризующее различие звуков одного инструмента, воспроизводимых с одинаковой громкостью. Высота звукового сигнала пропорциональна логарифму его частоты. В соответствии с принятым равномерным темперированным строем весь диапазон музыкальных звуков (примерно от 16 до 8000 Гц) разбит на промежутки, граничные частоты которых соотносятся как 1 : 2. Такие интервалы называются октавами. Каждая октава разбита на 12 равных частей (полутонов), соответствующих изменению частоты на 6%. Если каждой октаве (начиная от субконтроктавы) присвоить номера $j=1, 9$, а каждой ноте (начиная от ноты до) — номера $i=1, 12$, то значение частоты f_{ij} любой ноты музыкального звукоряда можно представить в виде матрицы

$$f_{ij} = f_0 \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1j} & \dots & a_{19} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{i1} & a_{i2} & \dots & a_{ij} & \dots & a_{i9} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{91} & a_{92} & \dots & a_{9j} & \dots & a_{99} \end{vmatrix}, \quad (2)$$

в которой f_0 — частота ноты до субконтроктавы, равная 16,351 Гц;

$$a_{ij} = \left(\sqrt[12]{2} \right)^{i-1} \cdot 2^{j-1}.$$

Совокупность значений частот f_{ij} можно представить некоторым множеством эталонных сигналов F , т. е. $f_{ij} \in F$. Анализируемые музыкальные сигналы с частотами f'_{ij} образуют множество измеряемых сигналов $f'_{ij} \in F$:

$$f'_{ij}(t) = f_{ij} \pm \Delta f_{ij}(t), \quad (3)$$

где f_{ij} — эталонный сигнал; Δf_{ij} — абсолютное отклонение сигнала от ближайшего эталонного значения. Отклонения выражают в относительных единицах — центах и комах (0,1 цента) с учетом знака отклонения. При этом отклонение в 1 цент соответствует 10^{-2} логарифма отношения частот, составляющих полутона, т. е. примерно 0,06 % номинала частоты. Мгновенные значения относительных отклонений δ можно оценить по формуле

$$\delta(t) = \frac{f'_{ij}(t) - f_{ij}}{f_{ij}}. \quad (4)$$

Знак относительных отклонений находим следующим образом:

$$F[\text{sign}[\delta f_{ij}(t)]] = \begin{cases} \xi_1, & \forall f'_{ij}(t) - f_{ij} > 0; \\ \xi_2, & \forall f'_{ij}(t) - f_{ij} < 0. \end{cases} \quad (5)$$

Значение, равное примерно шести центам, определяет минимальное изменение высоты тона, ощущаемое человеком [2]. Принимаем его в качестве контрольного при вычислении отклонения измеряемого сигнала от эталонного.

Кроме основной гармоники $f'_{ij}(t)$, музыкальный сигнал содержит ряд кратных гармоник обертонов, причем энергия сигнала распределяется вдоль диапазона неравномерно, она зависит от типа инструмента и громкости звука. Гармоники искажают синусоидальную форму сигнала. Поэтому следует принимать специальные меры для определения периода частоты основного тона, необходимого при решении задачи об отклонении высоты звукового сигнала от эталонного значения.

Таким образом, при анализе методов вычисления отклонения необходим учет двух факторов. Первый связан с тем, что в исследуемом сигнале $f'_{ij}(t)$ амплитуды одной или нескольких гармонических составляющих могут превышать амплитуду основного тона. Второй — с необходимостью вычисления отклонений от множества фиксированных значений эталонных частот F_* . Учитывая эти факторы, делаем вывод о том, что выделение частоты основного тона требует применения устройств частотной селекции перестраиваемых или коммутируемых по числу фиксированных значений F_* .

Известен метод выделения частоты основного тона с использованием перестраиваемых или переключаемых активных фильтров [3]. Использование таких фильтров для обработки звуковых сигналов ограничено перечисленными факторами, учет которых приводит к резкому возрастанию аппаратных затрат на фильтрацию, а следовательно, и повышению стоимости. К этому направлению можно отнести метод временной селекции входного сигнала [1], основанный на априорном знании периодов частот $f_{ij} \in F_*$. Суть метода заключается в выборе диапазонов анализа $T_{ij} \pm \Delta T_{ij}$ и в последующей оценке периода входного сигнала T'_{ij} . При достаточной простоте аппаратной реализации (значения T_{ij} и ΔT_{ij} можно хранить в ОЗУ) метод не исключает ошибочных отсчетов, так как богатые гармониками сигналы содержат «ложные» переходы через нуль, попадающие в диапазон анализа $\pm \Delta T_{ij}$. Сужение диапазона ограничивает возможности анализа звуков, имеющих существенное отклонение от необходимого эталонного значения.

Другое направление основано на амплитудной селекции звуковых сигналов [1]. Здесь период частоты основного тона определяется как временной интервал между моментами превышения амплитудной волны некоторого «высокого» фиксированного уровня $+B$ сразу после пересечения «низкого» фиксированного уровня $-B$. В устройстве фиксированные уровни выбираются равными 0,9 от максимальной амплитуды сигнала $\pm A$. Для успешной работы устройства необходим специальный входной блок с нелинейной характеристикой, позволяющий привести макси-

мальные значения амплитуд сигналов к уровням $\pm A$. Достоинством метода амплитудной селекции является тот факт, что он не требует априорного знания f_H , период сигнала вычисляется автоматически. Кроме того, аппаратные затраты для его реализации невелики. Однако метод нельзя назвать универсальным, так как изменяющиеся во времени соотношения амплитуд составляющих спектра $A_1, A_2, \dots$ и фаз $\varphi_1, \varphi_2, \dots$ приводят к появлению нескольких амплитуд, превышающих уровни $\pm 0,9$ в течение одного периода сигнала T_H . Это может привести к ошибке в результате обработки, если не приняты специальные меры защиты. Приведенные методы выделения частоты основного тона не отвечают критерию минимума аппаратных затрат при заданной точности вычисления и достоверности полученной информации.

Учитывая недостатки рассмотренных методов, предлагаем выделять частоту основной гармоник сигнала с помощью цифрового синхронного фильтра (ЦСФ) с фазовой автоподстройкой частоты. Рациональность ЦСФ обусловлена двумя факторами. Во-первых, амплитудно-частотная характеристика таких фильтров имеет вид «гребенки», т. е. фильтр пропускает все кратные гармоник сигнала, а следовательно, может воспринимать сигнал любой формы. Во-вторых, перестройка фильтра по частоте производится путем изменения частоты внутреннего генератора и не требует сложных цепей коммутации. В момент захвата входной частоты при условии, что входная частота находится в полосе захвата, система фазовой автоподстройки частоты (ФАПЧ) перестраивает генератор из режима биений в режим синхронизма. При этом частоты перестраиваемого генератора (ПГ), коммутации и входная равны. В системе разность фаз и выходное напряжение фазового детектора устанавливаются постоянными. Сигнал управления с фазового детектора подается на перестраиваемый генератор через фильтр нижних частот, который устраняет из спектра сигнала управления нежелательные составляющие побочных частот. Последние, попадая на вход ПГ, вызывают паразитную модуляцию. При условии, что на входе блока существует полезный сигнал, захват по частоте устанавливается автоматически и дальше ФАПЧ начинает работать в режиме слежения за разностью фаз принимаемого сигнала и сигнала ПГ. Частота перестраиваемого генератора выбирается в зависимости от разрядности счетчика: $f_H = f_0 \cdot 2^n$, где n — разрядность счетчика. С помощью входного сигнала дешифратор формирует 2^n равных временных интервалов, на которые делится период резонансной частоты. Чем больше каскадов содержит счетчик, тем точнее выходной сигнал синхронного фильтра повторяет входной. Кроме того, с увеличением коэффициента деления счетчика уменьшается полоса пропускания ЦСФ. Применяя ЦСФ, для дальнейшей обработки можно использовать логический сигнал с выхода делителя, частота кото-

деления мощности его гармонических составляющих. Можно непосредственно получить удобный для дальнейшей обработки логический сигнал и обеспечить простоту управления выбором заданного значения частоты.

Список литературы: 1. Пат. 4217805(США). Determination of pitch/ D. Slepian, Jr., Weldon.— Оpubл. 19.08.80. 2. Волошин В. Н., Федорчук Л. И. Электромузыкальные инструменты. — М.: Энергия, 1971. — 144 с. 3. Пат. 4227437 (США). Frequency detecting apparatus/ T. L. Inloes, D. E. George. — Оpubл. 14.10.80. 4. Иванова З. В., Ишунин В. В., Михов Д. К. Исследование частотных характеристик блока обработки сигналов к цифровому фазоиндикатору. — Радиотехника, 1978, вып. 45, с. 23—26.

Поступила в редакцию 01.10.84.

УДК 621.391

Ю. П. ЖУРАКОВСКИЙ, канд. техн. наук, О. М. МОРГАЛЬ

ОПРЕДЕЛЕНИЕ КОДОВОГО РАССТОЯНИЯ МНОГОПОЗИЦИОННЫХ КОДОВ

Для оценки обнаруживающей и направляющей способностей кодов существенное значение принадлежит расстоянию d . Методика определения кодового расстояния (КР) для двоичных кодов хорошо известна [1—5]. Однако с увеличением основания кода q возрастает сложность нахождения КР. Это объясняется тем, что если для двоичных кодов при определении КР в основном используется метрика Хэмминга, теория которой хорошо разработана, то для многопозиционных МП-кодов она не отражает всего различия кодовых комбинаций КК. Применение других известных метрик (Ли и модульной) ограничено ввиду того, что каждая из них используется лишь для определенного признака МП-кода. Опишем общую методику определения КР МП-кода. Для этого воспользуемся тремя перечисленными метриками.

Рассмотрим данную процедуру на конкретных примерах. Докажем утверждение, что при нахождении КР МП-кодов модульная метрика включает в себя метрики Хэмминга и Ли, в последняя — метрику Хэмминга. Предварительно рассмотрим определение КР во всех трех метриках для двоичного и многопозиционного, содержащего A признаков, кодов, для каждого из которых имеется основание q_n . Основание всего кода — $q = \prod_{n=1}^A q_n$. В качестве примера вычислим КР между кодовыми комбинациями X_a и X_c . Пусть в двоичной системе счисления ($q = 2$) $X_a = 101101$; $X_c = 100011$. Для многопозиционного кода ($q > 2$) $X_a = 201340$; $X_c = 212501$.

Метрика Хэмминга

$$d_{1e}(X_k, X_c) = \begin{cases} 0, & X_k = X_c; \\ 1, & X_k \neq X_c; \end{cases} \quad d_1 = \sum_i d_{1e}.$$

1. $q = 2$ — определение КР сводится к поразрядному сложению кодовых комбинаций по модулю 2, затем вычисляется вес (В) полученной КК, который и будет равен КР между двумя КК:

$$\begin{array}{r} 101101 \\ \oplus 100011 \\ \hline 001110 \end{array}; \quad B = 3; \quad d_1 = 3.$$

2. $q > 2$ — определение КР сводится к поразрядному выполнению логической функции неравнозначности между двумя КК, определению веса полученной двоичной КК, равного КР:

$$\begin{array}{r} 201340 \\ 212501 \\ \hline 011111 \end{array}; \quad B = d_1 = 5.$$

Метрика Ли

$$d_{2e}(X_k, X_c) = \min(|X_k - X_c|; q - |X_k - X_c|), \quad d_2 = \sum_i d_{2e}.$$

1. $q = 2$ — КР определяется так же, как и в метрике Хэмминга. Это доказывается тем, что

$$d_{2e}(X_k, X_c) = \min(|X_k - X_c|; 2 - |X_k - X_c|) = |X_k - X_c|.$$

Так как

$$\max(|X_k - X_c|) = 1, \quad \min(2 - |X_k - X_c|) = 1.$$

Тогда

$$d_{2e} = \min(|X_k - X_c|; 1) = |X_k - X_c|.$$

Поэтому $d_2 = d_1 = 3$.

2. $q > 2$ — определение КР сводится к последовательности алгебраических и логических операций:

$$\begin{array}{r} 201340 - X_k \\ 212501 - X_c \\ \hline 011241 - d_{3e} = |X_k - X_c| \\ 544314 - q_e - d_{3e} \\ \hline 011211 - d_{3e} = \min(d_{3e}; q_e - d_{3e}) \\ B = d_2 = 1 + 1 + 2 + 1 + 1 = 6. \end{array}$$

Модульная метрика

$$d_{3e}(X_k, X_c) = |X_k - X_c|; \quad d_3 = \sum_i d_{3e}.$$

1. $q = 2$ — КР определяется так же, как и в метрике Хэмминга, поскольку

$$d_{3e}(X_k, X_c) = \begin{cases} |X_k - X_c| = 0, & X_k = X_c; \\ |X_k - X_c| = 1, & X_k \neq X_c, \end{cases}$$

что совпадает с результатами d_{1e} . Поэтому $d_3 = d_1 = 3$.

2. $q > 2$ — определение КР сводится к поразрядному вычитанию двух КК, вычислению веса полученной КК, равного КР:

$$\begin{array}{rcccccc} 2 & 0 & 1 & 3 & 4 & 0 \\ 2 & 1 & 2 & 5 & 0 & 1 \\ \hline 0 & -1 & -1 & -2 & 4 & -1 \end{array}; \quad B = d_3 = 9.$$

Найдем зависимости между КР всех трех метрик для МП-кода.

Метрики модульная и Ли:

$$d_{3e}(X_k, X_c) = |X_k - X_c| \geq \min(|X_k - X_c|; q_e - |X_k - X_c|) = d_{2e}.$$

Так как всегда выполняется неравенство

$$d_{3e} \geq \min\{d_{2e}; q_e - d_{3e}\},$$

поэтому

$$d_{3e}(X_k, X_c) \geq d_{2e}(X_k, X_c).$$

Метрики Ли и Хэмминга:

$$d_{2e}(X_k, X_c) = \min(|X_k - X_c|; q_e - |X_k - X_c|).$$

При $X_k = X_c$ имеем

$$d_{2e}(X_k, X_c) = \min\{0, q_e - 0\} = 0 = d_{1e}.$$

Если $X_k \neq X_c$, то $|X_k - X_c| \geq 1$. Докажем, что при этом выполняется неравенство

$$|X_k - X_c| < q_e, \text{ т. е. } q_e - |X_k - X_c| > 1.$$

Параметр k изменяется в пределах от 1 до q_e , для всех этих значений выполняется следующее. Если $1 \leq X_k \leq q_e$, то $|X_k - X_c| < q_e$. Для $0 \leq X_k \leq q_e - 1$ также имеем $|X_k - X_c| < q_e$, так как $\max |X_k| = q_e - 1$. Следовательно,

$$\min(|X_k - X_c|; q_e - |X_k - X_c|) > 1 = d_{1e}$$

(данное значение имеет место при $X_k = X_c$). Итак, доказано, что

$$d_{2e}(X_k, X_c) \geq d_{1e}(X_k, X_c).$$

Таким образом, для всех X_k, X_c выполняется неравенство

$$d_{3e}(X_k, X_c) \geq d_{2e}(X_k, X_c) \geq d_{1e}(X_k, X_c).$$

Поскольку $d = \sum_e d_e$,

$$d_3(X_k, X_c) \geq d_2(X_k, X_c) \geq d_1(X_k, X_c).$$

Исходя из доказанного утверждения, можно сформулировать правило нахождения КР для МП-кода, содержащего признаки, КР которых определяются в различных метриках. Согласно этому правилу сначала необходимо определить КР для каждого

из признаков, затем сложить их в метрике, которую выбирают из встречающихся в данных вычислениях метрик (наиболее удавленную от начала в ряде метрик). Найденное таким образом кодовое расстояние с запасом может служить в дальнейшем для определения количества обнаруживаемых и исправляемых данных МП-кодом ошибок:

$$t_0 = d - 1; t_n = \left\lfloor \frac{d-1}{2} \right\rfloor.$$

Данное правило позволяет КР для всех кодов, имеющих признаки, описываемые одной из трех перечисленных метрик. К таким можно отнести коды, для передачи которых используются сигналы со следующими типами модуляции.

При амплитудной модуляции КР определяют в модульной метрике [1, 5]; при фазовой и время-импульсной — в метрике Ли [1, 2, 5]; при частотной модуляции для ортогональных систем используется метрика Хэмминга [1], для неортогональных — модульная [2].

Передача информации многопозиционными кодами за счет увеличения аппаратных затрат по сравнению со способами, использующими один тип модуляции, позволяет получать выигрыш в информационной скорости и помехоустойчивости, для оценки которой служит приведенная методика определения кодового расстояния.

Список литературы: 1. Берлекемп Э. Алгебраическая теория кодирования. — М.: Мир, 1971.—480 с. 2. Бородин Л. Ф. Введение в теорию помехоустойчивого кодирования. — М.: Сов. радио, 1968.—408 с. 3. Давенпорт В. Б., Рут В. Л. Введение в теорию случайных сигналов и шумов. — М.: Изд-во иностр. лит., 1960.—468 с. 4. Вазенкрант В., Джекобс Дж. Теоретические основы техники связи. — М.: Мир, 1969.—640 с. 5. Касами Т. Теория кодирования. — М.: Мир, 1978.—576 с.

Поступила в редколлегию 03.01.85.

УДК 681.3

Ю. К. КИРЬЯКОВ

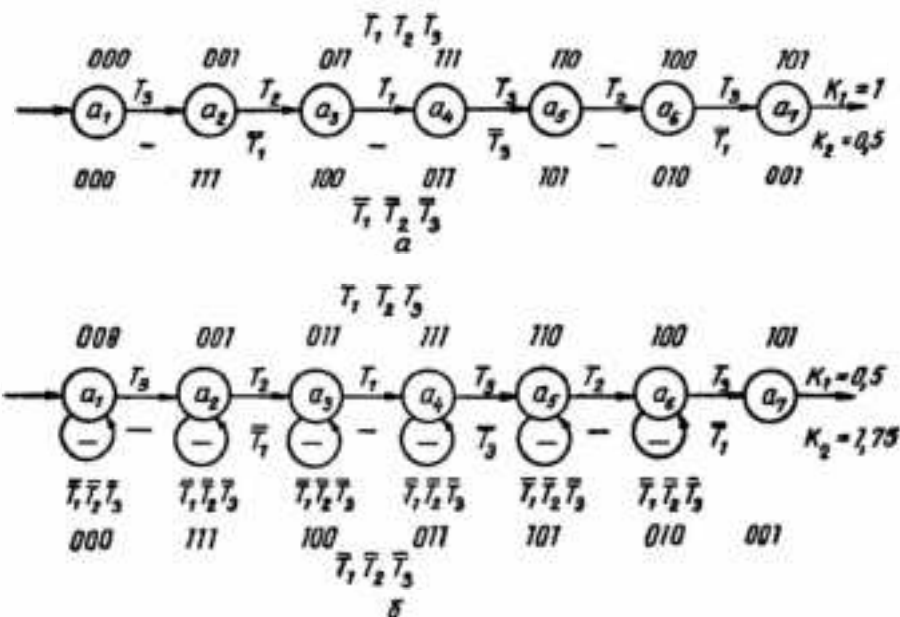
**КОДИРОВАНИЕ ВНУТРЕННИХ СОСТОЯНИЙ
МИКРОПРОГРАММНЫХ АВТОМАТОВ ПО МЕТОДУ
ВКЛЮЧЕНИЯ**

В работах [1—4] показано, что от способа кодирования внутренних состояний микропрограммных автоматов существенно зависит сложность его комбинационной части и всего автомата в целом. Качество полученного кодирования принято оценивать

коэффициентом качества кодирования, представляющим собой среднее число сигналов возбуждения, приходящееся на один переход (дугу) автомата. Если в запоминающей части микропрограммных автоматов используются триггеры T -, RS -, JK -, RST -, R -, S -, E -типов, то коэффициент качества кодирования вычисляется по формуле

$$K = \frac{\sum_{r=1}^{r=R} p_r(i, l) d_r(i, l)}{\sum_{r=1}^{r=R} p_r(i, l)}, \quad (1)$$

где $p_r(i, l)$ — вес перехода из состояния a_i в a_l ; $d_r(i, l)$ — расстояние по Хэммингу между кодовыми комбинациями K_i , K_l для этих состояний; R — общее число различных переходов в микропрограммном автомате. Расстояние по Хэммингу $d_r(i, l)$ для перехода r в случае применения триггеров указанного типа



Отметки сигналов возбуждения и коды состояний для последовательности неустойчивых (а) и устойчивых (б) состояний

совпадает с числом сигналов возбуждения, отмеченных на данном переходе. Стратегия кодирования внутренних состояний микропрограммных автоматов для триггеров T -, RS -, JK -, RST -, R -, S -, E -типов строится таким образом, чтобы суммарное число изменений состояний элементов памяти на всех переходах

автомата было минимальным. В оптимальном случае при соседнем кодировании всех состояний на каждый переход в автомате приходится точно один сигнал возбуждения (рисунок, позиция а).

Коэффициент качества кодирования K_i в этом случае равен единице. Вес перехода $p_r(i, j)$ обычно назначается с учетом сложности (длины) конъюнкции $a_i x$, на переходе r , особенностей структурной реализации микропрограммного автомата, наличия общих частей между функциями возбуждения элементов памяти и функциями выходов [5]. При использовании в запоминающей части автоматов триггеров с инверсными входами $\bar{T}$, $\bar{JK}$, $\bar{JK}JK$ -типов для кодирования внутренних состояний следует руководствоваться стратегией максимизации числа изменений состояний элементов памяти [6]. Коэффициент качества кодирования

$$K = \frac{\sum_{i=1}^{m-R} p_r(i, l) d'_r(i, l)}{\sum_{i=1}^{m-R} p_r(i, l)} \quad (2)$$

Здесь $d'_r(i, l)$ — расстояние между кодовыми комбинациями K_i , K_l , найденное по специальной матрице кодирующих интервалов M_n , учитывающей тип триггера в данном разряде. Матрица кодирующих интервалов M_n имеет размер $4 \times n$, где n — разрядность кодовых комбинаций. Каждой строке матрицы M_n соответствует один из четырех типов переходов $Q(i) \rightarrow Q(i+1)$ элемента памяти. Элемент m_{ij} матрицы M_n представляет собой общее число сигналов возбуждения, которые необходимо подать на входы элемента памяти в разряде j для осуществления перехода $Q(i) \rightarrow Q(i+1)$. Так, для числа состояний $N = 5 \dots 8$ и в случае применения триггеров $\bar{T}$ -типа матрица M_n имеет следующий вид:

$$M_n = \begin{array}{cc|ccc} & Q(i) & \rightarrow Q(i+1) & \bar{T} & \bar{T} & \bar{T} \\ \begin{array}{c} 0 \\ 0 \\ 1 \\ 1 \end{array} & \begin{array}{c} 0 \\ -1 \\ -0 \\ -1 \end{array} & \begin{array}{c} -0 \\ -1 \\ -0 \\ -1 \end{array} & \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \end{array} \quad (3)$$

Коэффициент качества кодирования для микропрограммных автоматов с графом переходов, изображенном на рисунке, позиция а, и триггерами $\bar{T}$ -, $\bar{JK}$ -, $\bar{JK}JK$ -типов равен 0,5, так как половину переходов можно закодировать парами противоположных кодов, для которых $d'_r(i, l) = 0$. Например, для пары противоположных кодовых комбинаций $K_i = 101$, $K_l = 010$ в соответствии с матрицей (3) получаем

$$d'_r(i, l) = \begin{vmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{vmatrix} = m_{10}^1 + m_{01}^2 + m_{10}^3 = 0 + 0 + 0 = 0.$$

Из способа вычисления $d_r(i, j)$, $d'_r(i, j)$ следует их связи

$$d'_r(i, j) = n - d_r(i, j), \quad (4)$$

где $n = \text{int } \log_2 N$. Для участка некоторого графа переходов, представляющего собой последовательность неустойчивых состояний, на рисунке, позиция a , над дугами и вершинами приведены отметки сигналов возбуждения T триггера и коды состояний, а под дугами и вершинами — сигналы возбуждения и коды состояний для случая применения триггера $\bar{T}$ -типа.

Отметим, что выбор типа триггера с прямыми или инверсными входами следует осуществлять с учетом закона функционирования микропрограммного автомата по меньшему значению коэффициента качества кодирования K . Для автоматов, граф переходов которых содержит последовательности устойчивых состояний (рисунок, позиция b), более выгодным является использование триггеров с прямыми входами. В автоматах, где преобладают последовательности неустойчивых состояний (позиция a), меньшие аппаратные затраты достигаются в случае использования в запоминающей части триггеров с инверсными входами. Модификация эвристического алгоритма кодирования внутренних состояний автомата для триггеров с инверсными входами рассмотрена в работе [6]. С учетом формулы (4) коэффициент качества кодирования K можно вычислять по формулам (1), (2), используя любую методику определения кодового расстояния $d_r(i, j)$ или $d'_r(i, j)$. Однако лучшее решение задачи кодирования внутренних состояний микропрограммных автоматов для триггеров с инверсными входами можно получить, если максимизировать число строк матрицы переходов автомата M с состояниями a_i и a_j в строке r , для которых $d'_r(i, j) = 0$.

Рассмотрим один из способов решения этой задачи, называемый методом прямого включения. Подобным методом решается задача распределения микроопераций по полям в автоматах с гибкой логикой [2]. Пусть автомат задан матрицей переходов

$$M = \begin{array}{c} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \left[\begin{array}{ccc} 1 & 2 & 3 \\ 2 & 3 & 2 \\ 3 & 1 & 5 \\ 4 & 1 & 3 \\ 5 & 1 & 4 \\ 6 & 2 & 4 \\ 7 & 3 & 4 \end{array} \right] \begin{array}{c} 3 \\ 2 \\ 2 \\ 1 \\ 1 \\ 1 \\ 1 \end{array} \end{array} \quad (5)$$

Слева от матрицы (5) указан порядковый номер строки, справа — весовой коэффициент $p_r(i, j)$. Для матрицы переходов автомата M составим матрицу совместности строк

$$S = [s_{ij}] = \begin{bmatrix} 0 & s_{12} & \dots & s_{1k} \\ s_{21} & 0 & \dots & s_{2k} \\ \dots & \dots & \dots & \dots \\ s_{m1} & s_{m2} & \dots & s_{mk} \end{bmatrix}. \quad (6)$$

Элемент s_{ij} характеризует совместимость строки y_i и строки y_j матрицы M . Строки $y_i = \{a_i, a_j\}$, $y_j = \{a_i, a_n\}$ считаются совместимыми и $s_{ij} = 1$, если $\{a_i, a_j\} \cap \{a_i, a_n\} \neq \emptyset$. Если же $\{a_i, a_j\} \cap \{a_i, a_n\} = \emptyset$, то $s_{ij} = 0$. Матрица совместимости (6) является булевой и симметричной. Для матрицы (5) получим следующую матрицу совместимости:

$$S = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \end{matrix}.$$

Все множество строк матрицы M разделим на H подмножеств Y_1, Y_2, Y_H , удовлетворяющих свойству полноты и единственности

$$\bigcup_{i=1}^H Y_i = Y; Y_i \cap Y_j = \emptyset; i, j = \overline{1, H}, i \neq j.$$

В подмножество $Y_k = \{y_1, y_2, \dots, y_k\}$ будем включать те из строк матрицы M , которые несовместимы ни с одной строкой этого подмножества. Если среди подмножеств $Y_1, Y_2, \dots, Y_k$ такое подмножество отсутствует, для новой строки матрицы M назначается новое подмножество Y_{k+1} . Состояние процесса разбиения строк на подмножества удобно характеризовать матрицей включения

$$R = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1N} \\ r_{21} & r_{22} & \dots & r_{2N} \\ r_{N1} & r_{N2} & \dots & r_{NN} \end{bmatrix}.$$

Элемент r_{ij} равен единице, если $y_i \in Y_j$, и равен нулю в противном случае. Таким образом, условие включения строки y_i матрицы M в подмножество Y_j сводится к следующему. Строка y_i включается в подмножество Y_j , если строка S_i матрицы совместимости S не пересекается со строкой R_j матрицы R .

Закодируем состояния автомата, заданного матрицей (5), по методу включения.

1. Множество Y_1 пусто. Для строки y_1 назначается подмножество Y_1 :

$$R = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]1.$$

2. Так как $S_2 \cap R_1 \neq \emptyset$, для строки y_2 назначается новое подмножество Y_2 :

$$R = \begin{array}{ccccccc|c} 1 & 2 & 3 & 4 & 5 & 6 & 7 & \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 2 \end{array}$$

3. Так как $S_3 \cap R_1 \neq \emptyset$, $S_3 \cap R_2 = \emptyset$, строка y_3 матрицы M включается в подмножество Y_2 :

$$R = \begin{array}{ccccccc|c} 1 & 2 & 3 & 4 & 5 & 6 & 7 & \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 2 \end{array}$$

4. Так как $S_4 \cap R_1 \neq \emptyset$, $S_4 \cap R_2 \neq \emptyset$, для строки y_4 назначается новое подмножество Y_3 :

$$R = \begin{array}{ccccccc|c} 1 & 2 & 3 & 4 & 5 & 6 & 7 & \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 3 \end{array}$$

5. Так как $S_5 \cap R_1 \neq \emptyset$, $S_5 \cap R_2 \neq \emptyset$, $S_5 \cap R_3 \neq \emptyset$, для строки y_5 назначается новое подмножество Y_4 :

$$R = \begin{array}{ccccccc|c} 1 & 2 & 3 & 4 & 5 & 6 & 7 & \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 4 \end{array}$$

6. Так как $S_6 \cap R_1 \neq \emptyset$, $S_6 \cap R_2 \neq \emptyset$, $S_6 \cap R_3 = \emptyset$, строка y_6 включается в подмножество Y_3 :

$$R = \begin{array}{ccccccc|c} 1 & 2 & 3 & 4 & 5 & 6 & 7 & \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 3 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 4 \end{array}$$

7. Так как $S_7 \cap R_1 = \emptyset$, строку y_7 включаем в подмножество Y_1 :

$$R = \begin{array}{ccccccc|c} 1 & 2 & 3 & 4 & 5 & 6 & 7 & \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 3 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 4 \end{array}$$

Процесс включения строк матрицы M завершен. Матрице включения окончательно соответствует следующее распределение строк матрицы M по подмножествам:

$$Y_1 = \{y_1^3, y_1^4\}; Y_2 = \{y_2^2, y_2^3\}; Y_3 = \{y_4^1, y_4^2\}; Y_4 = \{y_5^1\}.$$

Над y_i проставлены соответствующие данной строке весовые коэффициенты $p_i(i, j)$ из матрицы M . Поскольку в каждое подмножество Y_j ($j = \overline{1, N}$) входят только несовместимые строки, пары состояний из этих строк следует закодировать парами противоположных кодов K_i, K_j для которых $d_i(i, j) = 0$. Для минимизации коэффициента качества кодирования вначале необходимо закодировать пары состояний из подмножества Y_4 . Последнему соответствует максимальная сумма верхних индексов у $y_i \in Y_4$. В нашем случае это подмножество Y_1 или Y_2 . Заменяем строки из Y_j соответствующими им парами состояний:

$$Y_1 = \{(a_1, a_2)^2, (a_3, a_4)^1\}; Y_2 = \{(a_2, a_3)^2, (a_1, a_5)^2\}; \\ Y_3 = \{(a_1, a_3)^1, (a_2, a_4)^1\}; Y_4 = \{(a_1, a_4)^1\}.$$

Для состояний $a_1, a_2, a_3, a_4 \in Y_1$ назначим следующие пары противоположных кодов: $K_1 = 000$ и $K_2 = 111$; $K_3 = 001$ и $K_4 = 110$. Незакодированным осталось состояние a_5 . Дальнейший процесс кодирования можно вести по модифицированному алгоритму кодирования [6]. В соответствии с этим алгоритмом получаем:

$$M_5 = |1 \ 5|2; B_5 = \{1\}; C_1^0 = \emptyset; C_1^1 = \{101, 011\}; D_5^1 = \{101, 011\};$$

$$M_{101} = 2 \begin{vmatrix} 101 \\ 000 \end{vmatrix} = 2(0 + 1 + 0) = 2; W_{011} = 2 \begin{vmatrix} 011 \\ 000 \end{vmatrix} = 2.$$

Для состояния a_1 назначаем $K_3 = 101$, тогда $K = 0,82$.

Достоинством предложенной методики кодирования внутренних состояний автомата по методу включения является наличие программных реализаций этого метода, а также хорошая сходимость.

Список литературы: 1. Баранов С. И. Синтез микропрограммных автоматов. — Л.: Энергия, 1979. — 232 с. 2. Мидоров С. А., Ношков Г. И. Структура электронных вычислительных машин. — Л.: Машиностроение, 1979. — 384 с. 3. Миллер Р. Теория переключаемых схем. — М.: Наука, 1971. — 304 с. 4. Weiner P., Smith E. J. Optimization of Reduced Dependencies for Synchronous Sequential Machines. — IEEE Trans. on Computer, 1967, EC-16, N 6, p. 835—847. 5. Мидоров С. А., Лучко М. В., Скорубский В. И. Модель кодирования внутренних состояний микропрограммных автоматов. — В кн.: Построение управляющих устройств и систем. М.: Наука, 1974, с. 39—43. 6. Кахурин Н. Я., Кирыков Ю. К., Пряницкий А. М. Алгоритм кодирования внутренних состояний микропрограммных автоматов для триггеров с инверсными входами. — Изв. вузов СССР. Приборостроение, 1983, № 6, с. 46—49.

Поступила в редколлегию 27.04.84.

Рассмотрим принцип работы блока преобразования кодов, функциональная схема которого показана на рис. 1. Блок предназначен для преобразования телеграфного кода в 13-элементный код $A_1, A_2, \dots, A_{11}, \dots, A_{12}, B_1$, где A_i — i -я шина, $i = 1, 12$; B_1 — шина выбора постоянного запоминающего устройства буква-цифра. Телеграфный сигнал поступает на вход блока выделения первой паузы 1U2. Последний выделяет первую паузу в коде буквы и в течение этой паузы выдает импульсы заполнения

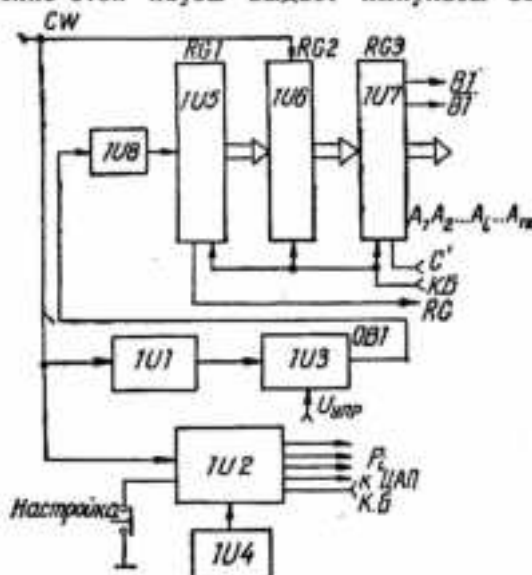


Рис. 1. Общая функциональная схема преобразователя кодов

с генератора прямоугольных импульсов (ГПИ блок 1U4) на цифро-аналоговый преобразователь (ЦАП) в блок 1U5.

Таким образом, по числу импульсов в паузе осуществляется настройка декодера на скорость манипуляции корреспондента. Блок 1U2 также вырабатывает импульсы управления блоком опознавания конца буквы и конца слова 1U3. Одновременно со входа импульсы проходят через формирователь 1U1, представляющий собой блок выделения переднего фронта входного импульса и блок преобразования его в импульс стандартной длительностью 0,1 мс. Этот импульс служит для запуска ждущего мультивибратора (ОВ-1) (блок 1U3). Длительность выходного импульса

ждушего мультивибратора определяется скоростью манипуляции корреспондента и равна $1/2$ длительности «тире». Почему выбрана длительность именно такой, будет обосновано после рассмотрения принципа преобразования телеграфных посылок в 13-элементный код $A_1, \dots, A_{12}, B_1$. Выходной импульс с ОБ-1 поступает на формирователь 1У8, на выходе которого получаются два импульса длительностью до 100 нс. Первый (ТР) соответствует переднему фронту выходного импульса ОБ-1, второй (TZ) — заднему. Эти импульсы управляют работой распределителя RG1 (блок 1У5), представляющего собой сдвиговый регистр на 13D-триггерах с 12 выходами, на которых последовательно появляется логическая единица. Сдвиг на один шаг производится при поступлении на вход импульса (ТР) или (TZ) из блока 1У8. Выходная логическая единица регистра сдвига разрешает запись входной информации в регистр RG2 (блок 1У6), построенный на 12 триггерах. Каждой шине регистра RG1 соответствует вход разрешения записи одного триггера регистра RG2. Таким образом, запись в триггеры регистра RG2 производится последовательно в моменты, определяемые передним и задним фронтами выходного импульса ОБ-1. А это значит, что запись происходит в моменты прихода переднего фронта элементарной телеграфной посылки «точки» или «тире» и в момент, соответствующий $1/2$ длительности «тире». Временные диаграммы изображены на рис. 2 для буквы К. В случае

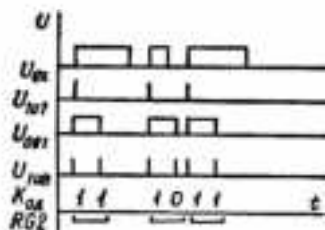


Рис. 2. Временные диаграммы работы время-импульсного преобразователя

прихода «точки» в регистр RG2 (блок 1У6) запишется код 10, так как во время поступления импульса ТР на входе блока декодирования, а следовательно, и на входах D-триггеров регистра RG2 будут присутствовать логические единицы. В соответствующий триггер регистра запишется единица и произойдет переключение на запись в другой триггер. Если пришедший сигнал «точка», то в случае прихода импульса TZ в следующий триггер регистра RG2 запишется ноль; если пришедший сигнал «тире» — единица. Таким образом, «точке» соответствует код 10, а «тире» — 11. Код Морзе шестизначный, следовательно, регистр RG2 должен содержать 12 триггеров. На его выходе при приеме знака появится 12-элементный код $A_1 - A_{12}$, однозначно определяемый пришедшим знаком.

Итак, блок декодирования абсолютно нечувствителен к укорачиванию «точки» и удлинению «тире». Поскольку время задержки импульса TZ выбрано равным $1/2$ длительности «тире», блок нечувствителен к удлинению «точки» менее чем на $1/2$ длительности «точки» и укорачиванию «тире» менее чем на $1/2$ длительности «тире». Таким образом, погрешность длительности

передачи элементарных посылок («точки» и «тире») может достигать 45 %, и это не скажется на качестве декодирования, т. е. не приведет к потере информации. Поэтому такой выбор времени задержки TZ оптимален.

Регистр $RG3$ (блок 1U7) представляет собой буферный регистр, в который заносится 13-элементный код, соответствующий входному знаку. Этот регистр служит для сопряжения блока преобразования с другими блоками системы.

Математическое описание работы преобразователя кодов заключается в следующем. Входной сигнал $U_{вх}$ — скачкообразная функция, принимающая значения 1 либо 0 (см. рис. 2).

Функция $r(t)$, равная единице, во время существования передних фронтов $U_{вх}$, т. е. в моменты, когда выполняется неравенство $dU_{вх}/dt \neq 0$, имеет вид

$$r(t) = \frac{\left[1 + \left| \frac{dU_{вх}}{dt} \right| \cdot \text{sign} \left(\frac{dU_{вх}}{dt} \right) \right] \cdot \frac{dU_{вх}}{dt}}{2} \quad (1)$$

Соответствующие ей моменты времени определяются так:

$$t_r = t_r(t). \quad (2)$$

Моменты времени, отстоящие от t_r на период $T_{0,5}$, равный $\frac{1}{2}$ длительности «тире»,

$$t_{rT} = t_r + T_{0,5}. \quad (3)$$

Таким образом, функция выходного сигнала блока 1U8, существующая в момент прихода переднего фронта входного сигнала и заднего фронта импульса с блока 1U3, отстоящего от переднего фронта на время $T_{0,5}$, примет вид

$$U_{вых}(t) = 1 \cdot t_r + 1 \cdot t_{rT}. \quad (4)$$

Умножив функцию (4) на входную функцию $U_{вх}$, получим выходной 12-элементный код

$$U_{код} = U_{вх} \cdot U_{вых}. \quad (5)$$

В результате прибавления к 12-элементному коду разряда управления регистром буква-цифра приходим к 13-элементному коду. Он может принимать значение 0 или 1 в зависимости от количества пришедших импульсов.

Поступила в редколлегию 28.02.55.

**ИСПОЛЬЗОВАНИЕ МЕТОДОВ И АЛГОРИТМОВ
ДЕДУКТИВНОГО ВЫВОДА ДЛЯ ИНТЕГРАЦИИ
НЕОДНОРОДНЫХ БАЗ ДАННЫХ**

В существующих автоматизированных системах управления, информационно-поисковых, -логических, -расчетных и экспертных системах нередко случаи применения неоднородных баз данных, поддерживаемых рядом разнотипных систем управления базами данных (СУБД). При решении задач, использующих накопленную в различных базах данных информацию, необходимо преодолевать информационную и программную несовместимость, а в ряде случаев — территориальную разобщенность баз данных. В работе [1] выделены следующие основные направления исследований в области организации взаимодействия информационно и программно несовместимых, а также распределенных баз данных: разработка и стандартизация языков определения и манипулирования данными СУБД; разработка методов трансляции данных, позволяющих конвертировать базы данных от представления в рамках одной СУБД к представлению в рамках другой; управление распределенными базами данных (РБД) в целях решения проблемы их территориального удаления; интеграция неоднородных баз данных, под которой понимается реализация одновременного и совместного использования прикладной программой нескольких баз данных, организованных в рамках различных СУБД.

Методы решения проблемы интеграции баз данных, базирующиеся на создании общей модели высокого уровня и преобразовании произвольных моделей данных в общую модель, описаны в работе [2].

Рассмотрим возможность использования алгоритмов дедуктивного вывода для решения некоторых задач интеграции неоднородных баз данных. Предлагаемая методика интеграции баз данных, поддерживаемых рядом разнотипных СУБД, включает следующие этапы: аксиоматизацию баз данных; трансляцию запроса пользователя с ограниченного естественного языка на язык многосортного исчисления предикатов первого порядка; использование алгоритмов дедуктивного вывода.

Первый этап интеграции (аксиоматизация) заключается во введении системы многоместных предикатов, позволяющих адекватно описать имеющиеся схемы баз данных на языке многосортного исчисления предикатов первого порядка, с последующим представлением их в виде совокупности аксиом данного языка.

На втором этапе запрос с ограниченного естественного языка транслируется с учетом ранее введенной системы предикатов в правильно построенную формулу (ППФ) языка многосортного исчисления предикатов. Принципы построения лингвистического процессора, позволяющего осуществить такую трансляцию, рассмотрены в работе [3].

Третий этап заключается в присоединении ППФ запроса к совокупности аксиом, описывающих имеющиеся базы данных, и использовании известных алгоритмов логического вывода [4] для извлечения требуемой информации, которая может находиться в различных базах данных.

Проиллюстрируем использование предлагаемой методики на примере интеграции двух неоднородных баз данных «Успеваемость» и «Кадры», которые могут быть составной частью банка данных «АСУ-ВУЗ». Первая из этих баз данных имеет сетевую структуру, а вторая — иерархическую. Фрагменты баз данных «Успеваемость» и «Кадры» изображены на рис. 1, 2.

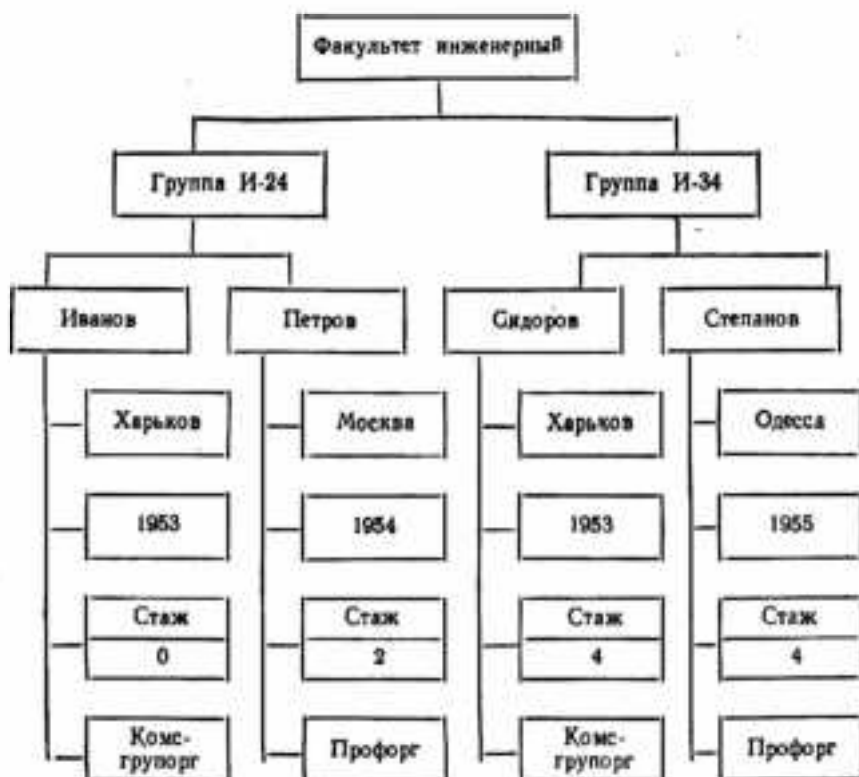


Рис. 1. Фрагменты баз данных «Кадры»

При эксплуатации системы «АСУ-ВУЗ» возможны запросы, ответ на которые требует использования информации, содержащейся в различных базах данных. Рассмотрим, например, запрос следующего вида:

кто из комсгруппиров получил оценку «5» по дисциплине «Д1»?

(1)

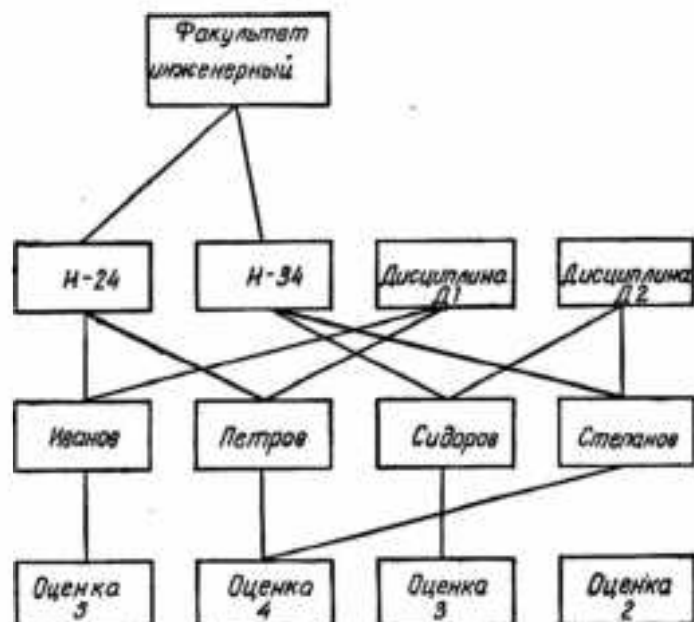


Рис. 2. Фрагменты баз данных «Успеваемость»

Очевидно, для ответа на этот вопрос необходимо совместное использование данных из баз «Кадры», «Успеваемость».

Для описания фрагмента базы данных «Кадры» введем следующие предикаты:

Ф. СТУД (X, Y, Z);
 СТУД (X, α, β, W, V).

Предикат (2) истинен, если студент X учится в учебной группе Y на факультете Z . Областью определения переменной X служит множество, состоящее из фамилий всех студентов вуза; областью определения переменной Y — множество всех учебных групп вуза, а областью определения Z — множество всех факультетов. Второй предикат будет истинным, если студент X родился в населенном пункте α в β -м году и имеет стаж до поступления в вуз W лет, а также занимает выборную общественную должность V . Областью значений переменной α является множество названий населенных пунктов, областью определения

переменной β — множество дат рождения студентов, а областью определения V — множество всех выборных общественных должностей. Тогда с учетом предикатов (2), (3) фрагмент базы данных «Кадры» (рис. 1) можно записать в виде следующей системы аксиом.

- 4.1. Φ . СТУД (ИВАНОВ, И-24, ИНЖ);
- 4.2. Φ . СТУД (ПЕТРОВ, И-24, ИНЖ);
- 4.3. Φ . СТУД (СИДОРОВ, И-34, ИНЖ);
- 4.4. Φ . СТУД (СТЕПАНОВ, И-34, ИНЖ);
- 4.5. СТУД (ИВАНОВ, ХАРЬКОВ, 1953, 0, КОМСГРУПОРГ);
- 4.6. СТУД (ПЕТРОВ, МОСКВА, 1954, 2, ПРОФОРГ);
- 4.7. СТУД (СИДОРОВ, ХАРЬКОВ, 1953, 4, КОМСГРУПОРГ);
- 4.8. СТУД (СТЕПАНОВ, ОДЕССА, 1955, 4, ПРОФОРГ).

Для описания фрагмента базы данных «Успеваемость» введем предикат

$$ОЦ(X, \gamma, \nu), \quad (5)$$

который будет истинным, если студент X по дисциплине γ получил оценку ν . Область определения переменной γ — множество названий дисциплин, изучаемых студентами, а для переменной ν областью определения является множество оценок $A = \{5, 4, 3, 2\}$.

С учетом предикатов (2), (3), (5) фрагмент базы данных «Успеваемость» можно описать совокупностью аксиом.

- 6.1. Φ . СТУД (ИВАНОВ, И-24, ИНЖ);
- 6.2. Φ . СТУД (ПЕТРОВ, И-24, ИНЖ);
- 6.3. Φ . СТУД (СИДОРОВ, И-34, ИНЖ);
- 6.4. Φ . СТУД (СТЕПАНОВ, И-34, ИНЖ);
- 6.5. $ОЦ$ (ИВАНОВ, Д1, 5);
- 6.6. $ОЦ$ (ПЕТРОВ, Д1, 4);
- 6.7. $ОЦ$ (СИДОРОВ, Д2, 3);
- 6.8. $ОЦ$ (СТЕПАНОВ, Д2, 4).

С учетом введенных предикатов (2), (3), (5) запрос (1) можно оттранслировать в следующую ППФ языка многосортного исчисления предикатов:

$$\exists X \text{ СТУД } (X, Y, Z, V, \text{«КОМСГРУПОРГ»}) \ \& \ ОЦ(X, Д1, 5). \quad (7)$$

Таким образом, запрос (1) принимает вид утверждения (7), истинность или ложность которого требуется доказать, исходя из групп аксиом (4), (6). Для поиска решения воспользуемся методом резолюций (4). Отрицание вопроса (7) имеет вид

$$\sim \text{СТУД } (X, Y, Z, V, \text{«КОМСГРУПОРГ»}) \vee \sim ОЦ(X, Д1, 5). \quad (8)$$

К ППФ (8) добавляем специальный предикат ОТВЕТ (X) [5]. В результате получаем выражение

$$\sim \text{СТУД } (X, Y, Z, V, \text{«КОМСГРУПОРГ»}) \vee \sim ОЦ(X, Д1, 5) \vee \text{ОТВЕТ } (X). \quad (9)$$

Доказательство.

1. $\sim \text{СТУД } (X, Y, Z, V, \text{«КОМСГРУПОРГ»}) \vee \sim ОЦ(X, Д1, 5) \vee \text{ОТВЕТ } (X)$ — отрицание вопроса;

2. СТУД (ИВАНОВ, ХАРЬКОВ, 1953, 0, КОМСГРУ-ПОРГ) — аксиома 4.5;

3. ~ОЦ (ИВАНОВ, Д1, 5) V ОТВЕТ (ИВАНОВ) — резолюция 1,2;

4. ОЦ (ИВАНОВ, Д1, 5) — аксиома 6.5;

5. ОТВЕТ (ИВАНОВ) — резолюция 3,4.

В отличие от существующих методов интеграции неоднородных баз данных [2] предлагаемая методика, использующая возможности аппарата математической логики, наряду с решением задачи интеграции баз данных позволяет получать ответы на вопросы, не содержащиеся в явном виде в имеющихся базах данных.

Список литературы: 1. Калиниченко Л. А. Вопросы интеграции неоднородных баз данных. — В кн.: Информационный обмен в вычислительных сетях. М.: Наука, 1980, с. 91—110. 2. Калиниченко Л. А. Методы и средства интеграции неоднородных баз данных. — М.: Наука, 1983. — 424 с. 3. Судаков Б. Н. О формализованной модели языка для взаимодействия пользователя с ЭВМ. — Пробл. бионики, 1982, вып. 29, с. 43—47. 4. Робинсон Дж. А. Машинно-ориентированная логика, основанная на принципе резолюции. — Кибернетич. сб., Н.С. 1970, вып. 7, с. 194—218. 5. Грик К. Доказательство теорем с использованием правила резолюции как основа для построения вопросно-ответной системы. — В кн.: Искусственный интеллект: подход на основе эвристического программирования. М.: Мир, 1973, с. 275—309.

Поступила в редакцию 05.01.84.

УДК 681.325

В. А. КАЛАШНИКОВ, Л. Н. МОЛДАВСКИЙ, В. А. ТИЩЕНКО

**СТРУКТУРА ИСХОДНЫХ ДАННЫХ ДЛЯ ПАКЕТА
ПРИКЛАДНЫХ ПРОГРАММ ВОССТАНОВЛЕНИЯ
СТРУКТУРНЫХ ТАБЛИЦ НЕРЕГУЛЯРНЫХ ПЛМ
ПО ТОПОЛОГИИ БИС**

Применяя программируемые логические матрицы (ПЛМ) для проектирования схем вычислительных устройств, можно значительно сокращать сроки проектирования за счет регулярной структуры и простой «настройки» ПЛМ на выполнение конкретной логической функции. Это особенно важно при проектировании заказных БИС. Основным недостатком ПЛМ — избыточность элементов и, как следствие, увеличение площади кристалла. Если на ПЛМ реализуется только часть схемы, критерий минимума площади в этом случае особенно важен. Для минимизации площади кристалла, занимаемой ПЛМ, топологи деформируют стандартную прямоугольную форму чертежа послойной геометрии (ЧПГ) матрицы. Изгибают шины конъюнкции и входов-выходов, в одной горизонтали располагают несколько шин конъюнкции, произвольно размещают входы-выходы и элементы питания ПЛМ.

При разработке ЧПГ ПЛМ на диалоговых системах машин-

ного проектирования (СПМ) появляются ошибки в чертеже, что требует создания программ контроля.

В данной статье описан пакет прикладных программ (ППП) восстановления структурных таблиц ПЛМ по ЧПГ, а также представлены формы задания информации о правилах поиска и о структуре элементов ПЛМ.

Описание ЧПГ ПЛМ, разработанной на СПМ 15УТ-4-017, задается SOURCE-файлом. Последний задает контуры чертежа списком координат угловых точек контуров, указывает тип контура и номер слоя, к которому контур принадлежит.

ППП восстановления структурных таблиц ПЛМ построен по модульному принципу и содержит следующие модули. *Модуль 1* преобразует SOURCE-файл во внутримашинное представление ЕС ЭВМ, формирует послойное описание ЧПГ. *Модуль 2* объединяет пересекающиеся контуры в слоях металла. Алгоритмы данного модуля реализуют операцию объединения контуров на модифицированной рецепторной модели, что дает линейную зависимость между размерностью задачи и временем решения. *Модуль 3* восстанавливает структуру ПЛМ по описанию ЧПГ. Алгоритмы и программы этого модуля находят и удаляют из описания ЧПГ контуры, реализующие элементы питания ПЛМ, восстанавливают и проверяют правильность реализации входных и выходных транзисторов, определяют их подключение к шинам конъюнкций и входам-выходам ПЛМ. В результате работы программ модуля конструктору выдается таблица, содержащая координаты контуров, реализующих входы-выходы ПЛМ. С помощью таблицы устанавливается соответствие между шинами входов-выходов ПЛМ и реализующими их контурами. *Модуль 4* вводит информацию об именах входов-выходов ПЛМ и печатает структурные таблицы восстановленных из чертежа ПЛМ БИС.

Большое значение при разработке программ восстановления принципиальной схемы (ПС) по топологии имеет способ задания исходных данных о структуре элементов и последовательности восстановления ПС. Успешный выбор формы задания исходной информации позволяет адаптировать программное обеспечение при переходе к новому классу схем или новой топологии. Для решения задачи восстановления необходима следующая информация о связи элементов топологии и ПС.

1. Отличительные признаки элементов и связей ПС (в дальнейшем под терминами «элемент» или «связь ПС» будем понимать их топологическую реализацию). Например, слой, встречающийся только в этом элементе, последовательность слоев ЧПГ, присущая только данному элементу, размер элемента и т. д.

2. Правила выполнения элемента или связи ПС, т. е. правила взаимного расположения слоев, к которым принадлежат контуры чертежа, реализующие элемент.

Целесообразно задавать структуру и правила поиска элементов в виде многосвязных графов:

$G_p(X_p, U_p, G_s(X_s, U_s))$ — граф поиска элементов и их структуры соответственно, где X_p, X_s — множество вершин; U_p, U_s — множество ребер графов G_p, G_s . Каждой вершине $x_p \in X_p$ соответствует слой ЧПГ. Две вершины x_p, x_p^i связаны ребром $u_p^i \in U_p$, если контуры слоев, соответствующих вершинам, пересекаются. Аналогично формируется граф структуры элементов G_s .

Компоненты связности графов G_p, G_s представляют собой корневые деревья, вершины которых объединяются в уровни. Принадлежность к уровню определяется длиной простой цепи, соединяющей вершину с корнем дерева. Корень дерева — нулевой уровень, вершины, смежные с корнем, — первый уровень и т. д.

Каждое дерево в графе поиска определяет последовательность прохождения слоев для поиска пересекающихся контуров, образующих элементы ПЛМ. После просмотра очередной вершины дерева ей в соответствие ставится один или множество контуров ЧПГ. В процессе прохождения по дереву графа G_p строятся деревья, каждой вершине которых соответствуют пересекающиеся контуры ЧПГ. По количеству вершин в этих деревьях и их конфигурации определяются типы элементов ПЛМ, а также контур, от которого восстанавливается структура элемента ПЛМ. Назовем компоненты связности графа $G_p(X_p, U_p)$ деревьями-эталоном типа (ДЭТ) элементов.

Идея определения типа по ДЭТ заключается в следующем. Вершинам ДЭТ поставлены в соответствие веса. При построении деревьев связности контуров суммируются веса вершин для которых найдены пересекающиеся контуры. Каждый тип элемента характеризуется значением суммы весов. Каждое ДЭТ задается списком, строка i которого соответствует вершине x_p^i дерева и содержит следующую информацию: $\langle \text{строка списка ДЭТ} \rangle ::= \langle \text{номер слоя} \rangle \langle \text{начальный адрес списка вершин, смежных из } x_i \rangle \langle \text{признак состояния контура} \rangle \langle \text{вес вершины} \rangle \langle \text{указатель количества контуров} \rangle$, где признак состояния контура задает значение второго элемента строки списка S_2 и указывает контур, от которого восстанавливается структура элемента, по ДЭТ. Указатель количества контуров принимает значения 0, если необходимо поставить в соответствие вершине один контур, и 1, если — множество контуров.

Каждое дерево в графе G_s задает последовательность поиска контуров в слоях ЧПГ для окончательной идентификации и проверки правильности выполнения элемента ПЛМ, которому это дерево соответствует. Компоненты связности графа G_s служат эталоном. По нему формируются деревья, связности контуров, реализующие структуру конкретных элементов ПЛМ.

Назовем компоненты связности графа $G_i(X_i, U_i)$ деревьями-эталоном структуры (ДЭС) элементов ПЛМ. Если дерево связности контуров, построенное при прохождении по вершинам ДЭС, изоморфно эталону, то элемент ПЛМ реализован правильно, иначе выдается сообщение об ошибке. Каждое ДЭС задано списком, в котором строка i соответствует вершине x_i' графа G_i и содержит следующую информацию: <строка списка ДЭС> ::= <номер слоя> <начальный адрес вершин, смежных из x_i > <признак состояния контура> <номер сообщения об ошибке> <указатель количества контуров>. Если для вершины $x_i' \in x_i$ ДЭС не найден контур, то выдается сообщение об ошибке с соответствующим номером.

Алгоритмы поиска по ДЭС и ДЭС аналогичны. Произвольно или по какому-либо условию выбирается контур из слоев, соответствующего корню дерева. Для этого контура в слоях ЧПГ вершин первого уровня отыскиваются контуры, пересекающиеся с исходным. Номера найденных контуров записываются в список $U1$. После просмотра всех вершин первого уровня не пустой список $U1$ перезаписывается в U , и процедура повторяется. Если с контуром m вершины x_i пересекается множество контуров из слоя вершины x_k , то эти контуры и номер вершины запоминаются в списке M . Вершине x_k ставится в соответствие первый контур из M , и алгоритм продолжает работу. Если список $U1$ пуст или просмотрены все вершины дерева, из списка выбирается очередной контур, принадлежащий вершине с максимальным значением уровня, и алгоритм продолжает работу. Поиск пересекающихся контуров по дереву заканчивается, когда списки $U1$ и M пусты. ППП восстановления структурных таблиц ПЛМ реализован на алгоритмическом языке Фортран-4 для ЕС ЭВМ. Общий объем памяти составляет 2790 операторов Фортрана. Время восстановления структурной таблицы ПЛМ по ЧПГ, содержащему 8660 контуров, заданных 20 621 точкой, составляет 1 ч. 40 мин. работы ЭВМ ЕС-1033.

Поступила в редколлегию 06.06.84.

УДК 681.324

С. А. ГУБКА, А. Н. АНИКИН, В. А. ДЕРГАЧЕВ

**ВЫБОР ОПТИМАЛЬНОГО СОСТАВА СИСТЕМЫ
ВСТРОЕННОГО КОНТРОЛЯ ЛОГИЧЕСКИХ МОДУЛЕЙ ГАП**

Основополагающий принцип построения гибких автоматизированных производств (ГАП) — модульность, с помощью которой можно формировать

структуры систем, соответствующих объективным потребностям управляемых объектов, оперативно перестраивать производство при изменении номенклатуры обрабатываемых деталей. Модульная структура обеспечивает автономное управление отдельными модулями ГАП с помощью локальных вычислительных средств, упрощает процесс модификации систем. При проектировании логических модулей для ГАП возникает задача повышения надежности их функционирования. Известные методы программной диагностики требуют значительных затрат времени для выявления отказов. Встроенные средства контроля позволяют в реальном масштабе времени идентифицировать не только отказы, но и сбой логического модуля. Эффективность контроля можно повысить путем оптимального выбора состава устройств встроенного контроля.

Рассмотрим комбинационное устройство с l входами, состояния которых описываются вектором $X = \{x_1, x_2, \dots, x_n\}$, и выходами с функциями $y_i(X)$, $i = \overline{1, m}$. Множество значений вектора X разобьем на t непересекающихся подмножеств (классов)

$$X_1^*, X_2^*, \dots, X_t^*; X_i^* \cap X_j^* = \emptyset; i, j = \overline{1, t}; i \neq j.$$

Для всех X_i , $X_j \in X_i^*$ справедливо $Y_i = Y_j = Y_i^*$, т. е. к одному классу относятся те значения входного вектора, которые совпадают со значениями выходного.

Обозначим через X^* , Y^* текущие значения входного и выходного векторов. Тогда при правильном функционировании контролируемого устройства между значениями X^* , Y^* существует взаимно однозначное соответствие:

$$(X^* \in X_i^*) \leftrightarrow (Y^* \in Y_i^*).$$

Аналогично установим соответствие для объединения нескольких классов:

$$(X^* \in (X_1^* \cup X_2^* \cup \dots \cup X_t^*)) \leftrightarrow (Y^* \in (Y_1^* \cup Y_2^* \cup \dots \cup Y_t^*)).$$

Определение соответствия между значениями входного и выходного векторов для заданного множества классов, участвующих в объединении, назовем проверками. Обозначим их через R_j , где j — номер проверки. Для описания множества классов, участвующих в проверке R_j , введем множество $A = \{a_1^j, a_2^j, \dots, a_l^j\}$, значения элементов которого определяются так:

$$a_i^j = \begin{cases} 1, & \text{если } i\text{-й класс участвует в } j\text{-й проверке;} \\ 0, & \text{в противном случае.} \end{cases}$$

Из-за ошибок на выходах контролируемого устройства значение выходного вектора преобразуется в некоторое ошибочное значение. Ошибочное преобразование можно обнаружить в случае, если ошибочное значение принадлежит классам, не входящим в состав проверки, или множеству запрещенных значений выходного вектора.

Система встроенного контроля состоит из d подсистем, реализующих логические функции соответствующих проверок. Под

структурой системы встроенного контроля будем понимать множество проверок, реализуемых системой встроенного контроля $D = \{R_1, R_2, \dots, R_d\}$. Выбирая различный состав проверок, можно построить множество вариантов и соответствующих им подсистем. При этом возникает задача выбора оптимальной структуры системы встроенного контроля, удовлетворяющей заданным критериям. Такими критериями обычно являются затраты на реализацию системы, а также множество обнаруживаемых ошибок.

Множество вариантов проверок обозначим через $\Gamma = \{R_1, R_2, \dots, R_r\}$, стоимость затрат на их реализацию $S = \{S_1^R, S_2^R, \dots, S_r^R\}$. Пусть E — множество ошибочных преобразований выходного вектора, которое требуется обнаружить, $E = \{e_1, e_2, \dots, e_n\}$; E_i — подмножество ошибочных преобразований, обнаруживаемых i -й проверкой, $E_i = \{e_1^i, e_2^i, \dots, e_{n_i}^i\}$; Ψ_i — количество ошибочных преобразований, обнаруживаемых i -й проверкой. Подмножества $E_i, E_j, i, j = \overline{1, r}$ могут пересекаться. Введем функцию θ , связывающую элементы множества Γ и обнаруживаемые ошибочные преобразования $\theta R_i = E_i$. Пусть S^* — стоимость затрат на реализацию системы контроля. Поставим в соответствие с множеством Γ последовательность $\Omega = \{\omega_1, \omega_2, \dots, \omega_r\}$, где $\omega_i \in \{0, 1\}$:

$$\omega_i = \begin{cases} 1, & \text{если проверка } R_i \text{ реализуется системой} \\ & \text{встроенного контроля;} \\ 0, & \text{в противном случае.} \end{cases}$$

Сформулируем задачу выбора оптимального состава системы встроенного контроля. Необходимо определить подмножество $D \subseteq \Gamma$ такое, что для любого элемента $e_j \in E$ найдется проверка $R_i \in D$, что $e_j \in \theta R_i$, при этом затраты должны быть минимальны:

$$S^* = \sum_{i=1}^r \omega_i S_i^R \rightarrow \min$$

при

$$E \subseteq \bigcup_{i=1}^r E_i \omega_i.$$

Построим двоичную матрицу σ_{ij} размера $r \times \psi$, элементы которой

$$\sigma_{ij} = \begin{cases} 1, & e_j \in E_i; \\ 0, & e_j \notin E_i. \end{cases}$$

Тогда ограничение можно представить в виде

$$\sum_{i=1}^r \sigma_{ij} \omega_i \geq 1, \quad j = \overline{1, \psi}.$$

Если размеры матрицы σ_{ij} небольшие, можно применять известные методы*. Однако с ростом размеров матрицы эффективность методов снижается. В связи с этим была разработана методика выбора оптимальной структуры системы встроенного контроля, позволяющая сократить время решения задачи и требуемый объем памяти. Суть метода заключается в следующем. Обозначим через S_k стоимость контролируемого дискретного устройства, значение которой будем рассматривать как верхнюю границу сложности системы встроенного контроля. Указанное допущение правомерно, так как дублирование контролируемого устройства позволяет обнаружить множество ошибочных преобразований выходного вектора. При выборе оптимальной структуры системы встроенного контроля нас интересуют варианты, обеспечивающие стоимость системы встроенного контроля значительно меньшую, чем дублирование. Верхняя оценка стоимости системы встроенного контроля позволяет отбросить неперспективные варианты проверок. Для этого в порядке возрастания стоимости проверок расположим множество их вариантов. В результате получим упорядоченные множества Γ^* , S^* . Найдем подмножество $R^* \subseteq \Gamma^*$ и соответствующее подмножество S^* , удовлетворяющее неравенству

$$\sum_{i=1}^d S_i^* < S_k. \quad (1)$$

Неравенство (1) представим в виде

$$\sum_{i=1}^{d-1} S_i^* + S_d^* < S_k,$$

откуда

$$S_d^* < S_k - \sum_{i=1}^{d-1} S_i^*,$$

где S_d^* — наибольшее значение стоимости реализации проверок, реализуемых системой встроенного контроля. Стоимость S_d^* максимальна, если

$$\sum_{i=1}^{d-1} S_i^* \rightarrow \min. \quad (2)$$

Условие (2) выполняется в случае

$$\sum_{i=1}^{d-1} S_i^* = \sum_{i=1}^{d-1} S_i^q.$$

Следовательно, перспективными являются варианты проверок, стоимость реализации которых удовлетворяет неравенству

$$S_j^* < S_k - \sum_{i=1}^{d-1} S_i^q, \quad j = \overline{1, m}. \quad (3)$$

Здесь m — количество проверок, удовлетворяющих условию (3).

* Ху Г. Целочисленное программирование и потоки в сетях. — М.: Мир, 1974. — 520 с.

Следующий этап решения задачи — генерация сочетаний проверок из m , по d , а также проверка ограничения для каждого из них. Чтобы уменьшить число проверок, был выбран линейный алгоритм сочетаний — алгоритм лексикографического порождения сочетаний. Последовательность сочетаний на множестве $\{1, 2, \dots, m\}$ представлена в лексикографическом порядке, если она записана в порядке возрастания полученных чисел. Например, лексикографическая последовательность сочетаний трех элементов из пяти имеет вид 123, 124, 125, 134, 135, 145, 234, 235, 345. Сочетание номеров проверок будем обозначать через $\mu^i = (\mu_1^i, \mu_2^i, \dots, \mu_d^i)$, где i — номер сочетания.

Разобьем множество сочетаний на группы. Подмножество μ_b^i , образованное из первых $d-1$ элементов сочетания μ^i , назовем базовым сочетанием. Количество базовых сочетаний N_b . Сочетания, имеющие одинаковые базовые сочетания, отнесем к одной группе. Сочетание μ^i соответствует определенным структурам системы встроенного контроля и стоимости системы. В пределах одной группы сочетаний стоимость варианта системы встроенного контроля не убывает с ростом порядкового номера сочетания в группе.

Алгоритм генерации вариантов структур систем встроенного контроля и выбора оптимального по стоимости состоит из следующих этапов.

1. Полагаем, что минимальное число проверок, необходимых для обнаружения заданного множества ошибок, $d=m$.

2. Формируем подмножество R' перспективных вариантов проверок.

3. Определяем количество базовых сочетаний N_b .

4. Выбираем номер базового сочетания $i=1$.

5. Если $i > N_b$, то увеличивая число проверок, реализуемых системой встроенного контроля: $d=d+1$, переходим к п. 2.

6. Формируем базовое сочетание μ_b^i .

7. Определяем суммарную стоимость проверок, номера которых входят в i -е базовое сочетание.

8. Находим минимальную стоимость проверок, входящих в i -ю группу $S_i^{\min}$.

9. Если $S_i^{\min} > S_g$ и $i=1$, переходим к п. 19. Если $S_i^{\min} > S_g$ и $i > 1$, увеличиваем число проверок, реализуемых системой встроенного контроля: $d=d+1$. Переходим к п. 5.

10. Определяем множество ошибок, обнаруживаемых проверками, номера которых входят в i -е базовое сочетание E_i^b .

11. Вычисляем множество необнаруженных ошибок $E_i^{\text{по}}$.

12. Находим количество проверок, необходимых для обнаружения множества необнаруженных ранее ошибок: d' .

13. Если $d' > 1$, выбираем новое базовое сочетание с номером $i=i+1$, переходим к п. 5.

14. Определяем множество проверок R^* , удовлетворяющих соотношению $E_i^{no} \subseteq E_j$, где $R_j \in R^*$.

15. Выбираем проверку $R_i \in R^*$ с минимальной сложностью S_i^R .

16. Если $S_i^R > S_k - S_i^R$, выбираем следующее базовое сочетание с номером $i = i + 1$, переходим к п. 5.

17. Уменьшаем значение ограничения по сложности $S_k = S_i^R + S_i^R$.

18. Выбираем новое базовое сочетание с номером $i = i + 1$, переходим к п. 5.

19. Конец.

Таким образом, данная методика позволяет оптимально выбирать системы встроенного контроля логических модулей ГАП.

Поступила в редколлегию 26.02.85.

УДК 681.51

Г. В. КОРЕНЕВА

НЕКОТОРЫЕ ВОПРОСЫ УПРАВЛЕНИЯ МОДУЛЕМ ГИБКОГО АВТОМАТИЗИРОВАННОГО ПРОИЗВОДСТВА

Анализируя современные тенденции развития методов управления производством, можно выделить два перспективных направления, интенсивно внедряемых в практику.

1. Жесткое программное управление, обеспечивающее синхронность работы всего производственного механизма.

2. Гибкое программное управление с применением диалоговых систем, исключающих или корректирующих отклонения в производственном процессе, вызванные случайными явлениями.

Часто эти способы применяются совместно в зависимости от целей производства и его специфики. Жесткое программное управление базируется на детерминированном представлении производственных процессов. С его помощью можно синхронизировать производство, рационально использовать технологические возможности оборудования, устанавливать точные сроки окончания выполнения заказа и т. д. Однако ему присущ существенный недостаток, заключающийся в потере управления при сбоях в производственном процессе, вызванных отказами оборудования, несвоевременной поставкой материалов или комплектов изделий и другими факторами. При наличии дестабилизирующих факторов система с жесткой программой управления не может исправить положение и восстановить синхронность работы производственных участков без существенных технико-экономических затрат.

Гибкое программное управление с использованием диалоговых систем позволяет учесть возможные возмущения в производственном процессе и своевременно применить необходимые корректирующие воздействия. Следовательно, в этом управлении содержится элемент прогнозирования, основанный на неформальном анализе сложившейся производственной ситуации. Недостаток такой системы — высокая стоимость системы управления, хотя для систем длительного действия этот недостаток ощущается слабо.

После предварительных замечаний рассмотрим некоторые вопросы управления участком (модулем) гибкого автоматизированного производства (ГАП) в системе с диалоговым управлением. Будем считать, что технические данные используемого оборудования, программа выпуска, номенклатура, а также технико-экономические показатели обслуживающих систем, например материально-технического снабжения и транспортных средств, известны.

На рисунке изображена структурная схема производственного модуля. Здесь введены следующие обозначения: МТС — участок материально-технического снабжения; ТС — участок транспортных средств, например, тележек-роботов; ТО — собственно технологическое оборудование (станки с ЧПУ, роботы); ПДУ — пульт диалогового управления. Подразумеваем, что на ПДУ оператор уполномочен принимать решения. Система укомплектована техническими средствами сбора и обработки информации. Кроме того имеются средства программного управления станками и транспортными средствами, а также математическое обеспечение системы. В исходном состоянии система свободна, т. е. не занята выполнением заказов.

Для организации управления в таких системах необходимо иметь три вида связей или линий передачи информации. На рисунке сплошными линиями показаны потоки программной информации, установленной заранее (перед пуском участка); пунктирными — потоки информации по обратным связям, т. е. информация о фактическом состоянии выполнения задания или фактическом состоянии технологических объектов; волнистыми — потоки информации, предназначенной для коррекции работы модуля ГАП по решению оператора.

Для модуля ГАП основу программной информации составляет программа выпуска

рассмотрим некоторые
Подпись ГАП

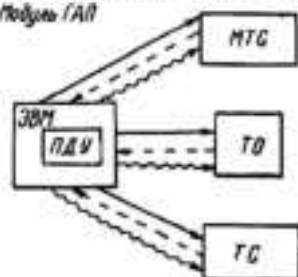


Схема информационных внутримодульных связей

$$P_{\text{ГАП}}(t) = \sum_{i=1}^N \sum_{j=1}^K \tau_{ij}(t),$$

где $\tau_{ij}(t)$ — время производства i -й единицы продукции при выпуске изделий j -й номенклатуры; K — число изделий j -й номенклатуры; N — номенклатура изделий; i — номер изделия в j -й номенклатуре. Программа выпуска представляет собой производственное задание технологическому модулю, который за определенное время должен выпустить заданное количество продукции.

Для транспортных средств основу программной информации составляет траектория и путь, проходимый тележками-роботами за время выполнения задания:

$$L(t) = \sum_{j=1}^N \sum_{i=1}^K l_{ij}(t).$$

Здесь $l_{ij}(t)$ — путь, проходимый тележкой-роботом при транспортировке i -й единицы продукции j -й номенклатуры.

Основой программной информации для МТС является функция расхода материалов по всем изделиям и в отдельности по каждому виду:

$$M = \sum_{j=1}^N \sum_{i=1}^K m_{ij}(t),$$

где $m_{ij}(t)$ — расход материала на i -е изделие j -го вида.

Отметим, что на участке решается задача управления складированием материалов и заготовок. Здесь контролируется состояние занятости складских ячеек и готовность склада обеспечить подачу материала и заготовок для выполнения технологических операций, а также рассматриваются вопросы, связанные со складированием готовой продукции в рамках модуля.

Для управления по жесткой программе характерны следующие ограничения:

$$P_{\text{ГАП}}(t)_{\text{пр}} = P_{\text{ГАП}}(t)_{\text{ф}}; L(t)_{\text{пр}} = L(t)_{\text{ф}}; M_{\text{пр}} = M_{\text{ф}}.$$

Индексы ф, пр означают фактическое и программное значения параметра.

Введением жесткой программы закладывается определенный синхронизм в работу системы. Если в системе нет сбоев, то она будет работать по жесткой программе, обеспечивающей максимальную производительность имеющегося технологического оборудования. Чтобы обеспечить устойчивое управление по жесткой программе, необходимо иметь устройства автоматического сбора информации о выполнении своих функций всеми технологическими автоматами, включая транспортные средства. Эта информация должна передаваться по каналам связи на временное хранение в оперативную часть управляющей программы. Информацию на пульт управления следует передавать безошибочно с темпом, достаточным для исключения

запаздываний информации о фактическом состоянии работы всего технологического модуля. Темп передачи информации определяется сменой состояния каждой единицы оборудования на участках ТО, ТС, МТС модуля ГАП, например переходом из состояния «занят» в состояние «свободен» станка, тележки-робота, складской ячейки. Информация на ПДУ о фактическом состоянии выполнения задания и о состоянии оборудования выдается через заданный интервал времени ΔT . Последний должен определяться характером производственного процесса, а также способностью оператора к восприятию информации. При диалоговом управлении информация может выдаваться по запросу оператора на ПДУ с темпом, аналогичным используемому при управлении по жесткой программе, а также с учетом прогнозирования, позволяющего принять своевременное решение и организовать управление. Это означает, что в случае приближения к заданным граничным значениям резервов, определяющих состояние склада, транспортного участка, технологического оборудования на ПДУ выдается предупреждающее сообщение, информирующее об остающихся резервах.

Итак, рассмотренные вопросы жесткого программного и диалогового управления модулем ГАП могут быть полезны при создании системы управления участком механообрабатывающего цеха в условиях гибкого автоматизированного производства.

Список литературы: 1. Мельцер М. Н. Диалоговое управление производством (модели и алгоритмы). — М.: Финансы и статистика, 1983.—240 с. 2. Гибкое автоматическое производство /Под ред. С. А. Майорова. — Л.: Машиностроение, 1983.—376 с. 3. Алексея В. Г., Гриднев В. Н., Нестеров Ю. И. Технологии ЭВА, оборудование и автоматизация. — М.: Высш. шк., 1984.—392 с.

Поступила в редколлегию 10.07.84.

ОПРЕДЕЛЕНИЕ ОПТИМАЛЬНОГО МЕЖПРОФИЛАКТИЧЕСКОГО ПЕРИОДА ПРИ ЭМПИРИЧЕСКОМ ЗАДАНИИ РАСПРЕДЕЛЕНИЯ ВРЕМЕНИ ДО ОТКАЗА

Рассмотрим сложную систему, в которой с постоянной периодичностью τ проводятся профилактические мероприятия для восстановления ресурса системы до исходного уровня, причем длительность проведения профилактик ξ — величина случайная. При отказе системы, проработавшей время меньшее, чем τ , проводится аварийный ремонт длительностью η , которая также является случайной величиной.

Время τ будем определять по критерию максимума коэффициента полезного использования системы. Для этого необходимо получить формальный вид функции распределения времени безотказной работы. Чтобы использовать методику, предложенную ранее [1, с. 53], следует решить интегральное уравнение, найти аналитическую зависимость для отказов, определить корни трансцендентного уравнения. Для выбранной формы распределения это может оказаться неудобным или даже невозможным.

В статье на основе эмпирической функции распределения времени до отказа изложена методика определения времени τ .

Несгруппированную выборку из n значений реализаций времени безотказной работы можно описать эмпирической кусочно-ступенчатой [2] функцией распределения вида

$$F(x) = P\{\Theta < x\} = \frac{1}{n} \sum_{i=1}^n v(x - a_i),$$

где Θ — значение очередной реализации времени до отказа системы; a_i — упорядоченные по возрастанию значения выборки; $v(x)$ — единичная функция Хевисайда,

$$v(x) = \begin{cases} 0, & x < 0; \\ 1, & x > 0. \end{cases}$$

На временной оси моменты начала функционирования системы после очередной профилактики либо аварийного ремонта являются точками регенерации [3]. В общем случае можно выделить n их типов (в соответствии с длительностью реализации последующего участка нормального функционирования системы). Период времени от точки регенерации i -го типа до последующей определяется так:

$$i_t = \begin{cases} a_t + \eta, & a_t \leq \tau; \\ \tau + \xi, & a_t > \tau. \end{cases}$$

Слагаемые a_t , τ соответствуют отрезкам нормального функционирования системы. Очевидно, период между двумя соседними профилактиками $a_1 \leq \tau \leq a_n$.

Коэффициент использования системы

$$k = \lim_{t \rightarrow \infty} \frac{T(t)}{t},$$

где t — время наблюдения за системой; $T(t)$ — суммарное время нормального функционирования системы. Введем параметр h — номер ближайшего к τ слева элемента упорядоченной по возрастанию выборки $a_1, a_2, \dots, a_n$. Формально

$$a_h < \tau < a_{h+1}; \quad h = \sum_{i=1}^n \gamma(\tau - a_i).$$

Можно записать, что

$$t = \sum_{i=1}^{N_1} (a_1 + \eta_i) + \sum_{i=1}^{N_2} (a_2 + \eta_i) + \dots + \sum_{i=1}^{N_h} (a_h + \eta_i) + \\ + \sum_{i=1}^{N_{h+1}} (a_{h+1} + \xi_i) + \dots + \sum_{i=1}^{N_n} (a_n + \xi_i).$$

Здесь N_i — количество точек регенерации i -го типа за время t ; η_i , ξ_i — длительность аварийного ремонта и плановой профилактики на периоде, соответствующем i -й точке регенерации i -го типа. Аналогично

$$T(t) = \sum_{i=1}^h N_i a_i + \sum_{j=h+1}^n N_j \tau.$$

Вероятность того, что произвольно взятая точка регенерации окажется i -го типа, следующая:

$$b_i = \lim_{t \rightarrow \infty} \frac{N_i}{N},$$

где N — суммарное число точек регенерации за время t ,

$$N = \sum_{i=1}^n N_i.$$

Математическое ожидание длительности аварийного ремонта

$$\eta = \lim_{t \rightarrow \infty} \frac{1}{N} \sum_{i=1}^{N_t} \eta_i$$

не зависит от значения индекса j в последнем выражении. Математическое ожидание длительности плановой профилактики

$$\xi = \lim_{N_f \rightarrow \infty} \frac{1}{N_f} \sum_{i_f=1}^{N_f} \xi_{i_f}$$

Таким образом,

$$k = \frac{\sum_{i=1}^h a_i b_i + \tau \sum_{i=h+1}^n b_i}{\sum_{i=1}^n a_i b_i + \eta \sum_{i=1}^h b_i + \xi \sum_{i=h+1}^n b_i} \quad (1)$$

Максимальное значение k соответствует оптимальной профилактике. Линейный характер изменения k при фиксированном значении h позволяет утверждать, что экстремумы функции (1) достигаются на правых границах интервалов эмпирического распределения. С учетом негруппированности выборки имеем

$$b_i = \frac{1}{n}, \quad i = \overline{1, n}.$$

Окончательное выражение для нахождения периодичности профилактических работ можно записать в виде

$$k^* = \max_{1 \leq h \leq n} \frac{\sum_{i=1}^h a_i + a_{h+1} (n-h)}{\sum_{i=1}^n a_i + \eta h + \xi (n-h)} \quad (2)$$

Тогда $\tau^* = a_{h+1}$. Параметр τ соответствует значению h из выражения (2), на котором достигается глобальный максимум.

Список литературы: 1. Базилевич Е. Ю., Каптанов В. А. Некоторые математические вопросы теории обслуживания сложных систем. — М.: Сов. радио, 1971. — 272 с. 2. Попов В. А., Литвинов М. Л. О решении задач массового обслуживания с использованием кусочных функций распределения. — Кибернетика, 1973, № 3, с. 74–77. 3. Кендалл Д. Стохастические процессы, встречающиеся в теории очередей, и их анализ методом вложенных цепей Маркова. — Математика, 1959, вып. 36, с. 53–56.

Поступила в редколлегию 09.10.84.

**ИНФОРМАЦИОННО-УПРАВЛЯЮЩАЯ СИСТЕМА
АВТОМАТИЗИРОВАННОГО СКЛАДА ДЛЯ ГИБКОГО
АВТОМАТИЗИРОВАННОГО ПРОИЗВОДСТВА**

Важнейшее направление повышения эффективности производства в условиях широкой номенклатуры непрерывно обновляемой продукции — внедрение промышленных роботов, систем программного управления и создание на их основе гибких автоматизированных производств (ГАП), связанных в единую интегрированную систему с САПР, АСУ ТП, АСУП. Составной частью ГАП являются автоматизированные склады. Автоматизированные складские и транспортные системы, предназначенные для процессов накопления и транспортирования, выполняют в условиях ГАП организаторские и диспетчерские функции, обеспечивающие координацию работ технологических модулей, участков подготовки и обеспечения производства. Они в известной степени обуславливают гибкость производства [1]. Система управления автоматизированным складом, реализуемая на современных вычислительных комплексах, связана информационными потоками с системой управления ГАП. Это позволяет своевременно получать объективную информацию о состоянии запасов и движении объектов производства. На основании информации планируется и организуется работа ГАП, корректируется сменно-суточное задание.

Разработку системы управления автоматизированным складом можно свести к решению следующих задач.

1. Выбор и проектирование комплекса технических средств.
2. Создание информационного обеспечения.
3. Разработка программного обеспечения.

Систему управления складом условно разобьем на два уровня: верхний — информационно-поисковый; нижний — управление исполнительными механизмами, к которым относятся один или несколько кранов-штабелеров, осуществляющих загрузку и выгрузку предметов производства, подающие конвейеры, транспортирующие грузы, роботы-манипуляторы, датчики положения кранов-штабелеров. В небольших складах все операции выполняет одна ЭВМ. С возрастанием сложности организации склада краны-штабелеры, как правило, управляются специальными контроллерами, а информационные задачи, управление контроллерами и другие функции возлагаются на ЭВМ. Такая организация системы управления существенно повышает ее надежность за счет возможности перехода на полуавтоматический и ручной режимы работы при отказах ЭВМ.

Основная задача контроллеров состоит в управлении движением кранов-штабелеров по горизонтали и вертикали в целях фиксации у определенной ячейки стеллажа. Последняя определяется координатами, которые выдает ЭВМ. Кроме того, контроллер выполняет одну из двух операций — «Взять» или «Положить». Они непрерывно управляют движением крана-штабелера в функции пройденного пути. Для этого вдоль пути крана-штабелера размещаются число-импульсные или кодовые датчики. Если количество выдаваемых импульсов пропорционально пройденному пути, такие системы называют число-импульсными [2]. Эти системы отличаются простотой реализации, небольшим количеством оборудования. Однако они подвержены сбоям, особенно при потере импульсов.

Наиболее перспективным является реализация контроллеров на микропроцессорах или микроЭВМ, сопрягаемых с основной ЭВМ системы управления. Высокие требования к точности и надежности обеспечиваются позиционными системами управления, когда положение каждой ячейки склада задается кодовыми датчиками. Контроллер крана-штабелера может сравнивать текущее положение с заданным, управлять его движением.

Как правило, в цеховых складах область загрузки совмещена с областью выгрузки. Важная задача при проектировании автоматизированного склада — выбор количества кранов-штабелеров, оценка их производительности. Запросы на прием и выгрузку предметов обычно поступают в случайные моменты времени. Время движения крана-штабелера в зависимости от адреса ячейки находится в определенных пределах и также случайно. Весь склад функционирует в режиме многолинейной системы обслуживания. С точки зрения внешних потребителей основным параметром склада является время от выдачи запроса на ту или иную операцию до окончания выполнения этой операции τ_s . В терминах теории массового обслуживания τ_s — среднее время пребывания запроса в системе [3],

$$\tau_s = \frac{1}{\mu} + \frac{\rho^n p_0}{\mu (n-1)!(n-\rho)^2}.$$

Здесь p_0 — вероятность простоя механизмов склада,

$$p_0 = \left[\sum_{k=0}^{n-1} \frac{\rho^k}{k!} + \frac{\rho^n}{(n-1)!(n-\rho)} \right]^{-1};$$

$1/\mu$ — среднее время выполнения одной операции краном-штабелером [3]; n — количество кранов-штабелеров; $\rho = \lambda/\mu$, где λ — средняя скорость поступления запросов на выполнение основных операций. Знак параметры λ , μ , можно так выбрать значение n , чтобы τ_s находилось в заданных пределах. При

этом не учитывается, что краны-штабелеры обычно закрепляются за определенными зонами обслуживания. Если один штабелер въехал в проем между стеллажами, второму туда въезд запрещен. Эти нюансы можно точнее учесть более сложной моделью или имитационным моделированием. В условиях АСУ каждый предмет производства описывается многозначным классификатором с выделением уровня детализации и указания уровня входимости, например 134.006.028.164, где 134 — код машины; 006 — код узла; 028 — код сборки; 164 — код детали. В автоматизированном складе за каждой деталью обычно закрепляется определенная ячейка. В памяти ЭВМ необходимо хранить адрес этой ячейки, количество деталей в ней, их плановое количество и другую информацию. Если детали мелкие, на поддоне можно хранить несколько типов деталей.

Совокупность информационных данных об объектах производства и их связях образуют базу данных информационно-управляющей системы автоматизированного склада, а программное обеспечение, которое выполняет различные операции над базой данных, образует систему управления базой данных (СУБД) [4]. Последняя решает следующие задачи: быстрый и эффективный поиск нужного элемента, извлечение и запись сведений об этом элементе, внесение новых элементов и исключение ненужных. В случае использования для реализации СУБД языков высокого уровня (Фортран, Бейсик) для организации массивов информации базы данных и установления связей между отдельными элементами можно использовать такой подход. Организуется четыре пула информации, состоящих из отдельных элементов, которые хранят реквизиты номеров машин, узлов, сборок и т. д. Последний пул хранит реквизиты деталей, а также текущую и постоянную информацию, относящуюся к конкретному предмету производства (адрес, количество, плановый показатель). Базы данных строятся по иерархическому принципу, при этом в элемент пула, помимо определенного реквизита предмета производства, вносятся и адреса расположения всех элементов, стоящих ниже по иерархии пула с таким реквизитом. В итоге каждый элемент любого пула, кроме четвертого, содержит три поля: реквизит (номер машины, узла, сборки или детали), начальный адрес (номер) расположения элементов, стоящего ниже по иерархии пула с таким реквизитом, указатель на следующий элемент в пуле, его номер (рис. 1). Третье поле необходимо организовать для выполнения операций сортировки элементов пула по реквизиту, добавления новых элементов, организации связей между элементами при исключении отдельных элементов пула. Последний элемент в этой цепи в третьем поле содержит нуль. Отдельный пул, кроме четвертого, можно строить на основе трех одномерных массивов информации, описываемых оператором DIMENSION. Элемент пула образуется из элементов

этих массивов с одним номером, при этом элемент из первого массива отводится под первое поле элемента пула, элемент второго массива — под второй элемент пула и т. д.

Остановимся на организации четвертого пула, в элементах которого хранится постоянная и текущая информация, относящаяся к конкретным деталям. Количество элементов этого пула должно быть равно возможному количеству деталей, хранящихся на складе. Это количество может достигать нескольких

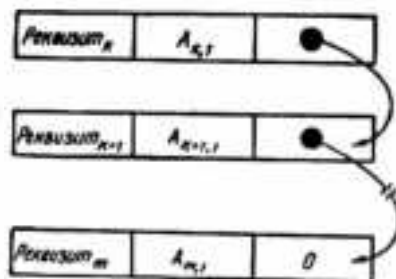


Рис. 1. Формат представления элементов одного пула информации

тысяч. В то же время в некоторых языках программирования, например Бэйсик, накладывается ограничение на размерность массивов информации. Это ограничение можно обойти, используя для организации четвертого пула не одномерные, а двумерные массивы информации, где позиция каждого элемента определяется парой чисел $\{i, j\}$, $1 \leq i \leq N$, $1 \leq j \leq N$. Для перехода от одномерного реквизита детали k к

паре чисел i, j , фактически определяющих положение детали в четвертом пуле, используется следующее преобразование:

$$i = \left[\frac{n}{N} \right] + 1, \quad j = n - \left[\frac{n}{N} \right] N.$$

Здесь $[]$ — целая часть отношения в скобках. При организации четвертого пула используется столько массивов информации, сколько полей отводится для хранения реквизита детали, постоянной и временной информации.

Общая структура программного обеспечения автоматизированного склада изображена на рис. 2. Его основу составляет управляющая программа, осуществляющая диалог с оператором, а также выбор режимов работы и их включение. Режимы подразделяются на информационные и управляющие. Информационные включают в себя инициализацию, печать документов и статистики, внесение корректив в базу данных. Во время инициализации формируются массивы информации, устанавливаются связи между элементами, первоначально заполняются элементы массивов информации. В режиме «Печать документов» извлекается необходимая информация о предметах, хранящихся на складе. Режим «Печать статистики» используется для выдачи сведений о загрузке склада, частоте использования отдельных ячеек, что может быть использовано для рационального размещения элементов в целях минимизации времени доступа к ним. В управляющем режиме по номеру элемента производства осуществляется поиск адреса

расположения этого элемента в базе данных, выдача этого адреса контроллеру крана-штабелера и обработка одной из двух основных операций — «Взять», «Положить». Все операции сопровождаются обращениями к СУБД, выполняющей следующие функции: поиск элемента, модификацию его содержимого, добавление новых элементов, исключение освободившихся и т. д. Если элементы пула упорядочены по значению реквизитов, для выполнения этих операций можно использовать методы ускоренного доступа к данным, в противном случае эти операции

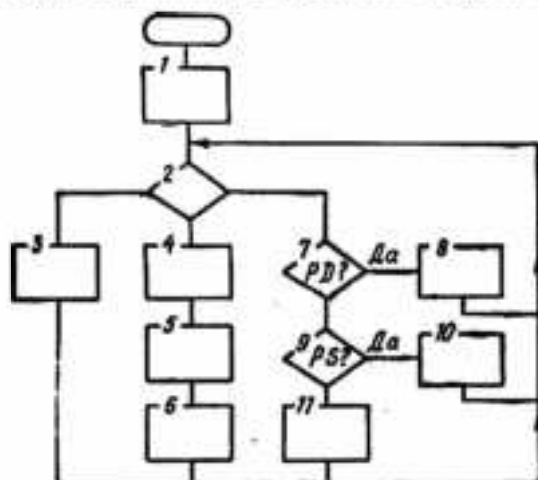


Рис. 2. Структурная схема программного обеспечения автоматизированного склада:

1 — прогон тестов; 2 — выбор режимов работы; 3 — инициализация; 4 — прием кода детали и вида операции от оператора; 5 — поиск в базе данных; 6 — выдача команды в контроллер и обработка операции; 7, 8 — выбор информационных режимов работы; 8 — печать данных; 9 — печать статистики; 10 — другие режимы работы

осуществляются путем последовательного просмотра элементов пула, увязанных в цепь. Для этих целей используются специальные языки, например Кодасил [4].

При усложнении структуры автоматизированного склада, возрастании количества хранимых в нем предметов усложняется и информационно-управляющая система. Она может включать в себя несколько мини-ЭВМ, развитые средства диалога с оператором и связи с системой управления производством, сложные транспортные и погрузоразгрузочные средства, работающие в автоматическом режиме.

Список литературы: 1. Котов Е. П. Основные направления создания автоматизированных складских систем для ГАП. — Обмен опытом в радиопромышленности, 1983, вып. 8, с. 33—35. 2. Снегов А. Л. Автоматизированные склады. — М.: Машинностроение, 1979.—283 с. 3. Саити Т. Л. Элементы теории массового обслуживания и ее приложения. — М.: Сов. радио, 1971.—520 с. 4. Ульман Дж. Основы систем баз данных. — М.: Финансы и ста-

УДК 519.87:71

Л. И. НЕФЕДОВ, канд. техн. наук

**МЕТОД ОПРЕДЕЛЕНИЯ ОПТИМАЛЬНОЙ
ПО ИНСОЛЯЦИИ ПОМЕЩЕНИЯ ОРИЕНТАЦИИ ЗДАНИЯ
В САПР ЖИЛОЙ ЗАСТРОЙКИ**

Улучшение санитарно-гигиенических условий жизни населения — важная задача в деле повышения качества проектирования жилой застройки. Эти условия обеспечиваются, в частности, посредством хорошей инсоляции жилых помещений квартир, детских и классных комнат, садов-яслей и школ. Требуемое время инсоляции помещений достигается в основном соответствующей ориентацией здания. Для зданий прямоугольной формы решение такой задачи не вызывает трудностей. Если здание имеет сложную конфигурацию и внутреннюю планировку, определение его оптимальной ориентации с учетом инсоляции помещений связано со значительным объемом вычислений. Поэтому необходимо разрабатывать математические методы и алгоритмы определения оптимальной по инсоляции помещений ориентации зданий. Это позволит автоматизировать процессы проектирования на базе создания САПР жилой застройки.

Рассмотрим формулировку задачи о выборе оптимальной ориентации здания на примере жилого дома. Известно множество параметров, определяющих форму, размеры, планировку здания и его квартир, а также характеристики территории застройки. Необходимо найти область допустимой ориентации здания Ξ с учетом инсоляционных требований, а также оптимальную ориентацию здания Θ^* по инсоляции жилых комнат всех квартир.

В качестве локального критерия оценки инсоляции жилых помещений дома выбран следующий [1, 2]:

максимальная инсоляция жилых комнат каждой квартиры

$$I(\Theta) = \left\{ \max_{\Theta \in \Xi} I_{\mu}(\Theta), \quad \mu = \overline{1, M} \right\}. \quad (1)$$

Здесь M — количество квартир; $I_{\mu}(\Theta)$ — инсоляция жилых комнат μ -й квартиры,

$$I_{\mu}(\Theta) = \left\{ \max_{\Theta \in \Xi_{\mu}} I_{\mu d}(\Theta) = \overline{d, d_{\mu}} \right\}, \quad (2)$$

где d_μ — количество жилых комнат в μ -й квартире; $I_{\mu d}(\theta)$ — инсоляция d -й жилой комнаты μ -й квартиры; E_μ — допустимый сектор ориентации здания для μ -й квартиры.

Так как все этажи дома чаще всего типовые, достаточно оценить инсоляцию жилых комнат всех квартир только типового этажа. Область допустимых решений E задается такими ограничениями [2].

Не менее t^p часов инсоляции хотя бы одной жилой комнаты в одно-, двух- и трехкомнатной квартирах:

$$\bigvee_{d=1}^{d_\mu} [I_{\mu d}(\theta) - t^p] > 0, \quad d_\mu = \overline{1, 3}. \quad (3)$$

То же для двух жилых комнат в четырех- и пятикомнатной квартирах:

$$\bigvee_{d=1}^{d_\mu-1} \bigvee_{l=d+1}^{d_\mu} [(I_{\mu d}(\theta) - t^p) \wedge (I_{\mu l}(\theta) - t^p)] > 0, \quad d_\mu = \overline{4, 5}. \quad (4)$$

Для того, чтобы решить задачу (1) — (4), необходимо знать аналитическое описание инсоляции каждой жилой комнаты квартиры. Рассмотрим аналитическое описание инсоляции комнаты через окно, если его ориентация совпадает с ориентацией здания. Последнее не имеет лоджий, балконов, карнизов и стен, затеняющих окно [2]. Введем следующие обозначения: Z , E — ширина и высота окна; T — толщина стен дома; $\lambda(t)$, $\omega(t)$ — азимут и угловая высота солнца для дня весеннего (осеннего) равноденствия; t — местное время; t_n , t_s — нижнее и верхнее значения временного интервала, в течение которого оценивается инсоляция в день весеннего (осеннего) равноденствия; $\psi^*(\theta)$, $\psi^*(\theta)$ — нижнее и верхнее значения горизонтального углового сектора, при котором солнечные лучи могут проникать в комнату через окно в зависимости от ориентации здания θ ,

$$\psi^*(\theta) = \theta - \arctg \frac{Z}{T}, \quad \psi^*(\theta) = \theta + \arctg \frac{Z}{T}; \quad (5)$$

$\eta[\theta, \lambda(t)]$ — верхнее значение вертикального углового сектора, при котором солнечные лучи могут проникать в комнату через окно в зависимости от ориентации здания θ и азимута солнца $\lambda(t)$,

$$\eta[\theta, \lambda(t)] = \arctg \left\{ \frac{E}{T} |\cos[\theta - \lambda(t)]| \right\}. \quad (6)$$

Тогда на основании формул (5), (6) инсоляция комнаты через окно во время $t \in [t_n, t_s]$ имеет место, если выполняются следующие условия:

$$\omega(t) < \eta[\theta, \lambda(t)]; \quad \psi^*(\theta) < \lambda(t) < \psi^*(\theta). \quad (7)$$

Учитывая изложенное выше, аналитическое описание инсоляции комнаты в зависимости от ориентации здания при условиях

$$\psi^*(\theta) - \psi^*(\theta) < \lambda(t_s) - \lambda(t_n); \quad (8)$$

$$\varphi[\theta, \lambda(t)]|_{\lambda(t)=\varphi^*(t), \varphi^*(t)} > \max_{t \in [t_n, t_{n+1}]} \omega(t), \quad (9)$$

имеет вид

$$I_{\mu d}(\theta) = \begin{cases} 0, & |\theta| < \lambda(t_n) - \arctg \frac{Z}{T}; \\ \frac{|\theta| - \lambda(t_n) + \arctg \frac{Z}{T}}{15^\circ}, & |\theta| - \lambda(t_n) < \arctg \frac{Z}{T}; \\ \frac{2\arctg \frac{Z}{T}}{15^\circ}, & \lambda(t_n) + \arctg \frac{Z}{T} < \theta < \lambda(t_n) - \arctg \frac{Z}{T}. \end{cases} \quad (10)$$

Если ориентации окна комнаты и здания не совпадают, следует учитывать угол сдвига между расположениями здания и окна. В случае, когда в здании есть элементы, затеняющие окна, например, лоджии, балконы, карнизы, стены и т. д., необходимо принимать во внимание их размеры.

Таким образом, с учетом выражения (10) задача (1) — (4) относится к задачам двухвекторной одномерной оптимизации с кусочно-линейными локальными функциями цели и ограничениями типа «И — ИЛИ». Для ее решения предложен метод, основанный на синтезе обобщенного критерия [3]. При этом строится обобщенный критерий оценки инсоляции жилых комнат каждой квартиры:

$$I'_\mu(\theta^\mu) = \max_{\theta \in \pi_\mu} \left[\frac{1}{d_\mu} \sum_{d=1}^{d_\mu} R_{\mu d}(\theta) \right], \quad (11)$$

где θ^μ — оптимальная ориентация здания с точки зрения инсоляции μ -й квартиры; $R_{\mu d}(\theta)$ — функция полезности d -го локального критерия для μ -й квартиры,

$$R_{\mu d}(\theta) = \frac{I_{\mu d}(\theta) - I_{\mu d}^{\text{наих}}}{I_{\mu d}^{\text{наил}} - I_{\mu d}^{\text{наих}}}; \quad (12)$$

$I_{\mu d}^{\text{наил}}$, $I_{\mu d}^{\text{наих}}$ — наилучшее и наихудшее значения инсоляции d -й комнаты μ -й квартиры.

Критерий (11) является аддитивным, так как при оценке инсоляции квартиры допускается компенсация уменьшения инсоляции одних жилых комнат за счет увеличения инсоляции других. При оценке инсоляции жилых помещений дома необходимо обеспечить выравнивание значений полезности критериев инсоляции квартир, которое достигается при следующем обобщенном критерии оценки инсоляции квартир дома:

$$I'(\theta^*) = \max_{\theta \in \pi} \min_{\mu \in \Gamma, M} I'_\mu(\theta). \quad (13)$$

Опишем алгоритм решения задачи.

1. Используя формулы (5) — (10), строим локальные критерии оценки инсоляции комнат в квартирах.

2. Для каждой квартиры определяем допустимый сектор ориентации здания Ξ_μ . С этой целью решаем неравенства вида (3) или (4).

3. Находим допустимый сектор ориентации здания Ξ с учетом инсоляции всех квартир:

$$\Xi = \bigcup_{\mu=1}^M \Xi_\mu. \quad (14)$$

4. Если $\Xi = \emptyset$, т. е. система неравенств (3), (4) несовместна, то для здания не существует такой ориентации, при которой обеспечивается требуемое время инсоляции всех квартир. В этом случае переходим к п. 5, иначе — к п. 6.

5. Задачу (1) — (4) заменяем задачей поиска такой ориентации здания θ^* , при которой минимизируется максимальное отклонение инсоляционных ограничений от требуемых (3), (4). Для этого решаем задачу безусловной оптимизации по критерию

$$I^*(\theta^*) = \min_{\theta} \max_{\mu \in \overline{1, M}} \left\{ \bigvee_{d_\mu=1}^{d_\mu} [t^{rp} - I_{\mu d}(\theta)], \quad d_\mu = \overline{1, 3}; \quad (15)$$

$$\bigvee_{d_\mu=1}^{d_\mu-1} \bigvee_{l=d_\mu+1}^{d_\mu} \left\{ [t^{rp} - I_{\mu d}(\theta)] \wedge [t^{rp} - I_{\mu l}(\theta)], \quad d_\mu = \overline{4, 5} \right\}.$$

6. С помощью формул (11), (12) строим обобщенный аддитивный критерий оценки инсоляции жилых комнат каждой квартиры $I_\mu^*(\theta)$.

7. По критерию (13) находим оптимальную по инсоляции квартир ориентацию дома.

Разработанный алгоритм реализован на ЭВМ серии ЕС с использованием алгоритмического языка PL/1.

Рассмотрим аналитическое описание инсоляции квартир секции дома и определим ее оптимальную ориентацию. Исходной информацией являются следующие данные: широта г. Харькова — 50° с. ш., временной интервал оценки инсоляции — 8...16 ч; секция дома не имеет затеняющих окна элементов и состоит из двух двухкомнатных и двух трехкомнатных квартир; при ориентации секции дома $\theta = 0$ каждая двухкомнатная квартира имеет две жилые комнаты с окнами на юг, каждая трехкомнатная квартира — две жилые комнаты с окнами на север, и одну жилую комнату с окном на юг; размер окон $1,5 \times 1,5$ м²; толщина стен 0,32 м; ограничение по инсоляции жилых помещений квартир $t^{rp} = 3$ ч.

Нижнее и верхнее значения горизонтального углового сектора, при котором солнечные лучи проникают в комнату через окно, определяются для комнат с окнами на север с помощью

выражений (5). Учитывая коэффициент отражения солнечных лучей, принимаем

$$\psi^*(\theta) = \theta - 60^\circ; \quad \psi^*(\theta) = \theta + 60^\circ.$$

Так как для любых значений $t \in [8; 16]$ выполняются условия (8), (9), аналитическое описание инсоляции комнат северного и южного расположения в зависимости от ориентации секции дома имеет вид

$$I^N(\theta) = \begin{cases} 0, & |\theta| < 60^\circ; \\ \frac{|\theta| - 60^\circ}{15^\circ}, & 60^\circ < |\theta| < 180^\circ; \end{cases}$$

$$I^S(\theta) = \begin{cases} \frac{120^\circ - |\theta|}{15^\circ}, & |\theta| < 120^\circ; \\ 0, & 120^\circ < |\theta| < 240^\circ. \end{cases}$$

Ограничения по инсоляции жилых комнат в квартирах следующие.

Не менее трех часов инсоляции хотя бы одной комнаты в двухкомнатной квартире:

$$[I^S(\theta) - 3] \vee [I^S(\theta) - 3] > 0.$$

То же для трехкомнатной квартиры:

$$[I^N(\theta) - 3] \vee [I^N(\theta) - 3] \vee [I^S(\theta) - 3] > 0.$$

На основании ограничений определяем секторы допустимой ориентации секции дома для двухкомнатной $[-75^\circ, +75^\circ]$ и трехкомнатной квартир $[-75^\circ, +75^\circ]$, $[105^\circ, 255^\circ]$. Допустимый сектор ориентации секции дома Ξ для всех квартир согласно формуле (14) составляет $[-75^\circ, +75^\circ]$.

Воспользуемся формулами (11), (12) для построения обобщенных критериев оценки инсоляции жилых комнат двух- и трехкомнатной квартир:

$$I^{2*}(\theta) = \max_{|\theta| < 75^\circ} \frac{75 - |\theta|}{75};$$

$$I^{3*}(\theta) = \max_i \begin{cases} \frac{|\theta|}{360}, & 60^\circ < |\theta| < 75^\circ; \\ \frac{120 - |\theta|}{360}, & |\theta| < 60^\circ. \end{cases}$$

Оптимальные решения по данным обобщенным критериям совпадают и достигаются при ориентации секции дома $\theta = 0$.

Используя предложенную методику, определяли наилучшую ориентацию жилого дома 5Т-16, имеющего сложные конфигурацию и внутреннюю планировку, лоджии и стены, затеняющие окна жилых комнат. Для этого дома при заданном ограничении по инсоляции жилых помещений квартир $t^p = 2,5$ ч не существовало допустимой ориентации. Поэтому решалась задача

поиска такой ориентации здания Θ^* , при которой достигалось минимаксное отклонение от требуемых инсоляционных ограничений, т. е. выполнялась оптимизация по критерию (15). Минимаксное отклонение от инсоляционных требований получали при ориентации здания $\Theta^* = 10^\circ$. В этом случае только две квартиры имели время инсоляции 2,3 ч, остальные квартиры — более 2,5 ч.

Таким образом, разработанный метод позволяет определять допустимую и оптимальную по инсоляции помещений ориентацию зданий при их размещении на территории застройки.

Список литературы: 1. Нефедов Л. И. Разработка комплекса математических моделей и алгоритмов эффективного управления жилищным строительством города: Автореф. дис. ... канд. техн. наук, Х., 1977. — 24 с. 2. Нефедов Л. И., Петров Э. Г. Размещение жилых домов с помощью ЭВМ. — АСУ и приборы автоматики, 1978, вып. 47, с. 41—49. 3. Автоматизированные системы управления городским хозяйством / И. В. Кузьмин, Э. Г. Петров, И. А. Алферов и др. — К.: Будівельник, 1978. — 144 с.

Поступила в редколлегию 22.01.85.

УДК 621.012

В. И. ТОВЧЕНКО, канд. техн. наук

МОДЕЛЬ ИМИТАЦИИ МНОГОВАРИАНТНЫХ ОРГАНИЗАЦИОННО-ТЕХНОЛОГИЧЕСКИХ СХЕМ СТРОИТЕЛЬСТВА

Планирование и управление в строительных организациях уровня трест-комбинат базируется на организационно-технологических моделях процессов возведения зданий и сооружений. Основа таких моделей — организационно-технологические схемы строительства. Построить эти схемы сложно. Их многовариантность позволяет обеспечить управляемость системы при накладываемых ограничениях.

Многовариантные очередности освоения строительных работ приводят к разным длительностям выполнения всего комплекса работ. Следовательно, существуют варианты, обеспечивающие минимальную продолжительность строительства. Необходимо производить поиск схем с эффективными сроками завершения работ, учитывая при этом ограничения по ресурсам.

Рассмотрим модель построения многовариантных схем строительства, позволяющую осуществлять поиск схем с учетом изложенного.

Трудовые и технические ресурсы типа мощность представлены в модели множеством каналов обслуживания;

$$\bigcup_{i=1}^{I^*} K_i^{(t)} = \{ \langle K_{11}^{(t)}, K_{21}^{(t)} \rangle, \dots, \langle K_{1I^*}^{(t)}, K_{2I^*}^{(t)} \rangle \},$$

где параметр $K_{ij}^{(t)}$ определяет условие занятости или освобождения i -го канала в момент времени t (если $K_{ij}^{(t)} = 0$, то канал свободен, в противном случае занят); I^* — количество каналов обслуживания; $K_{2i}^{(t)}$ — идентифицирует код специализации канала обслуживания. Набор выполняемых работ представлен множеством

$$\bigcup_{j=1}^{J^*} R_j = \{ \langle R_{1j}, R_{2j} \rangle, \dots, \langle R_{1J^*}, R_{2J^*} \rangle \},$$

в котором R_{1j} — наименование j -й работы; R_{2j} — код ресурса, необходимого для выполнения j -й работы; J^* — количество работ.

По дискретной сетке времени в моменты времени t определяется множество $\bigcup_{i=1}^{I^*} G_i^{(t)}$ работ, фронт которых открыт, и для их выполнения имеется хотя бы один свободный ресурс типа мощности по условиям

$$\forall i \in C_i^* = 0;$$

$$\bigcup_{i=1}^{I^*} G_i^{(t)} \cap \bigcup_{j=1}^{J^*} P_j = \emptyset;$$

$$\forall i \forall j [\exists G_{ij}^{(t)} = K_{ij}^{(t)} \& K_{ij}^{(t)} = 0].$$

Здесь C_i^* — признак того, что работа $G_i^{(t)}$ еще не выполнялась, в противном случае работа выполняется или уже выполнена;

$G_{ij}^{(t)}$ — шифр работы с открытым фронтом; $\bigcup_{j=1}^{J^*} P_j$ — множество запрещающих правил выполнения работ; J^* — количество таких правил; $C_{ij}^{(t)}$ — код ресурса, необходимый для выполнения работы $G_{ij}^{(t)}$; I^* — количество работ с открытым фронтом.

Исходя из $\bigcup_{i=1}^{I^*} K_i^{(t)}$, необходимо определить множество каналов обслуживания $D_{\mu}^{(t)}$ при $\mu = 1, \dots, \mu^*$, свободных в момент времени t , т. е. $\forall \mu [D_{\mu}^{(t)} = 0]$. Здесь μ^* — количество каналов обслуживания. Множество $D_{\mu}^{(t)}$ состоит из подмножеств $D_{1\mu}^{(t)}$, $D_{2\mu}^{(t)}$. Первое определяет состояние каналов обслуживания (если $D_{1\mu}^{(t)} = 0$, то канал свободен, в противном случае занят); второе — код специализации канала обслуживания.

Если для каждой работы с открытым фронтом имеются свободные каналы обслуживания, формируется только один вариант схемы строительства. Как правило, трудовых и технических ресурсов для выполнения работ с открытым фронтом в данный момент времени меньше. Следовательно, в очередности выполнения работ с открытым фронтом возникают варианты. Исходя

из $\bigcup_{\mu=1}^{k^*} D_{\mu}^{(i)}$, $\bigcup_{i=1}^{I^*} G_i^{(i)}$, можно построить несколько ветвей реализации схем строительства в данный момент времени t . В дальнейшем при $t > t_0$ (t_0 — начальный момент времени) каждая ветвь порождает свои ответвления схем строительства и т. д.

Таким образом, возникает проблема выбора перспективных вариантов схем строительства в целях обеспечения эффективных сроков $T(K_i^{(i)})$ завершения всего комплекса работ в зависимости от множества $\bigcup_{i=1}^{I^*} K_i^{(i)}$ трудовых и технических ресурсов. Такую задачу решим на основе эвристического метода оценки перспективности вариантов схем строительства. Ветви схем строительства строятся, исходя из множеств $\bigcup_{i=1}^{I^*} G_i^{(i)}$, $\bigcup_{\mu=1}^{k^*} D_{\mu}^{(i)}$, а также с учетом эвристической функции оценки перспективности ветви [1]. Множество работ с открытым фронтом $\bigcup_{i=1}^{I^*} G_i^{(i)}$ раскладывается на k подгрупп. Подгруппа $V_k G_{i/k}^*$ формируется таким образом, что

$$\forall k [C_{2(o_{1k})}^{(i)} = C_{2(o_{2k})}^{(i)} = \dots = G_{2(o_{1k})}^{(i)}],$$

где I — количество работ в k -подгруппе множества $\bigcup_{i=1}^{I^*} G_i^{(i)}$. Для каждой подгруппы $G_{i/k}^*$ из множества $\bigcup_{\mu=1}^{k^*} D_{\mu}^{(i)}$ вычисляют количество каналов обслуживания N , которые можно загружать работами k -й подгруппы. Все подгруппы k^* классифицируются по трем видам. Первый определяется по признаку $N \geq I$. Количество таких подгрупп k' , причем $k' \leq k^*$. В этом случае возможен только один вариант решения — $f(k') = 1$. Второй — исходя из условий $N = 1$, $I > 1$. Количество таких подгрупп удовлетворяет условию $k' \leq k^*$, количество решений

$$f(k^*) = I/N.$$

Третий — по признаку $N < I$ & $N > 1$. Причем $k^* < k^*$,

$$f(k^*) = \frac{(I+1-N)(I+2-N) \dots (I+n-N)}{N!},$$

где $n = N$.

Все возможные варианты решений по множеству $\bigcup_{i=1}^{I^*} G_i^{(i)}$ в данный момент времени определяются так:

$$f^* = \prod_{k=1}^{k^*} f(k).$$

Здесь

$$f(k) = f(k') + f(k'') + f(k''').$$

Каждая ветвь f в следующий момент времени порождает свои варианты. Из-за того, что нельзя реализовать на ЭВМ алгоритм, обеспечивающий просмотр всех возможных ветвей решения (большой размер решаемой задачи, ограниченные возможности ЭВМ по объему памяти), начиная с t_0 следует найти одну или несколько перспективных ветвей. Перспективные ветви назначаются по минимуму функции штрафа, определяемой по множествам $\bigcup_{i=1}^{i^*} G_i^{(n)}$, $\bigcup_{k=1}^{k^*} D_k^{(n)}$:

$$Y_{f(x)} = \sum_{i=1}^{i^*} \sum_{l=1}^{l^*} C_{1il} \cdot X_{il} + \sum_{k=1}^{k^*} \sum_{l=1}^{l^*} C_{2kl} \cdot X_{kl}.$$

Отметим, что $C_{1+i} = 1$, $C_{2+i} = 1$, если работа из множества $\bigcup_{i=1}^{i^*} G_i^{(n)}$ и канал обслуживания из множества $\bigcup_{k=1}^{k^*} K_k^{(n)}$ простаивают; X_{il} — шаг моделирования по времени; l^* — время завершения моделирования. Продолжаются только те ветви, которые обеспечили минимум функции штрафа. В результате такого отбора организационно-технологических решений в диапазоне $\overline{t_0}, \overline{t^*}$ отыскиваются схемы строительства, обеспечивающие эффективные сроки завершения всего комплекса работ.

Рассмотрим алгоритм имитации схем строительства (рисунок).

2. Ввод множеств

$$\bigcup_{i=1}^{i^*} R_i; \quad \bigcup_{k=1}^{k^*} K_k^{(n)}; \quad \bigcup_{j=1}^{j^*} P_j.$$

3. Формирование параметра N и множеств

$$\bigcup_{i=1}^{i^*} G_i^{(n)}; \quad D_k^{(n)}; \quad G_{ik}.$$

4. Формирование вариантов решений по подгруппам k .

5. Формирование множества всех возможных вариантов f^* решений.

6. Установление очередного варианта на просмотр.

7. Загрузка каналов обслуживания по данному f -му варианту.

8. Определение функции штрафа $Y_{f(x)}$ по f -му варианту.

9. Проверка на просмотр всех возможных ветвей решения в данный момент времени.

10. Отмена решения по рассматриваемой ветви.

11. Выбор перспективного варианта решения.

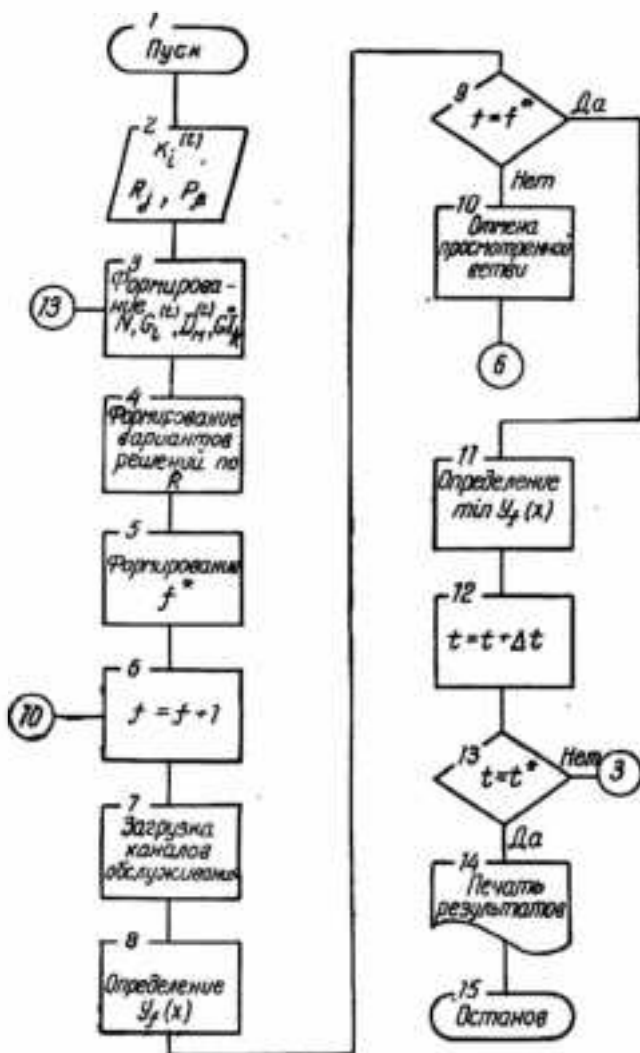
12. Переход на следующий интервал времени.

13. Проверка на завершение времени моделирования.

14. Печать начала и окончания работ по выбранной схеме

строительства, а также графика загрузки ресурсов типа мощности.

По сравнению с альтернативными сетевыми моделями моделирование многовариантных схем строительства на основе систем обслуживания проще. В отличие от моделей построения оптимальной очередности выполнения работ в строительных потоках [3] такое моделирование расширяет область применения этих моделей.



Алгоритм имитации схем строительства

Практическое использование комплекса имитационных моделей многовариантного строительства показало целесообразность их применения в строительных организациях уровня трест — комбинат при планировании и управлении.

Список литературы: 1. Товченко В. И. Автоматизация разработки технологических моделей строительства. — АСУ и приборы автоматики, 1981, вып. 58, с. 84—90. 2. Рыбальский В. И. Автоматизированные системы управления строительством. — К.: Вища шк. — 480 с. 3. Афанасьев В. А. Алгоритм формирования, расчета и оптимизации методов организации работ. — Л., 1980. — 93 с.

Поступила в редколлегию 27.11.84.

УДК 621.438

Г. И. ОНИЩЕНКО, канд. техн. наук, Т. Я. ШИШКИНА,
канд. техн. наук

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ РЕШЕТОК И ЛОПАТОК ТУРБОМАШИН

Профиль канала можно задавать с помощью семейства окружностей переменного радиуса, центры которого находятся на средней линии канала. Кривые, касающиеся этих окружностей, описывают профиль канала. В предлагаемом методе профиль канала описывается дискриминантной кривой однопараметрического семейства. Один из возможных способов задания такого рода семейства линий заключается в следующем.

Рассматривается функция $F(x, y, \alpha)$ трех переменных x, y, α , и для каждого значения параметра α из множества $\{\alpha\}$ определяется кривая семейства с помощью уравнения

$$F(x, y, \alpha) = 0. \quad (1)$$

Характеристической точкой кривой семейства (1), отвечающей данному значению α параметра семейства, называется точка $M(x, y)$, координаты x, y которой удовлетворяют системе уравнений

$$F(x, y, \alpha) = 0;$$

$$F'_\alpha(x, y, \alpha) = 0.$$

Геометрическое место характеристических точек кривых семейства (1) называется дискриминантной кривой этого семейства. Пусть задано уравнение средней линии канала $y = f(x)$ и уравнение проходного сечения [закон изменения радиусов окружностей семейства (1) $R = \varphi(x)$]. Следует найти уравнения наружной и внутренней линий профиля канала, удовлетворяющих

заданным начальным условиям. Решение ищем в виде дискриминантной кривой семейства окружностей переменного радиуса $R = \varphi(x)$ с центрами на средней линии канала, заданной уравнением $y = f(x)$. Такое семейство описывается уравнением

$$(x - \alpha)^2 + [y - f(\alpha)]^2 = \varphi^2(\alpha).$$

Для нахождения дискриминантной кривой этого семейства необходимо решить систему уравнений

$$\begin{aligned} (x - \alpha)^2 + [y - f(\alpha)]^2 &= \varphi^2(\alpha); \\ -x + \alpha - f'_\alpha(\alpha) [y - f(\alpha)] &= \varphi'_\alpha(\alpha) \varphi(\alpha). \end{aligned}$$

Так как при ее решении не всегда удается исключить параметр α , параметрические уравнения дискриминантной кривой имеют вид

$$\begin{aligned} y &= f(\alpha) - \frac{\varphi(\alpha)}{1 + f'^2_\alpha(\alpha)} \left[\varphi'_\alpha(\alpha) f'_\alpha(\alpha) \mp \sqrt{1 + f'^2_\alpha(\alpha) - \varphi'^2_\alpha(\alpha)} \right]; \\ x &= \alpha - \varphi(\alpha) + \frac{\varphi(\alpha) f'_\alpha(\alpha)}{1 + f'^2_\alpha(\alpha)} \left[\varphi'_\alpha(\alpha) f'_\alpha(\alpha) \mp \sqrt{1 + f'^2_\alpha(\alpha) - \varphi'^2_\alpha(\alpha)} \right]. \end{aligned} \quad (2)$$

Исследуем различные классы кривых $y = f(x)$, $R = \varphi(x)$ в целях определения кривых, для которых возможно найти аналитическое решение. Если в системе (2) освободиться от радикала $\sqrt{1 + f'^2_\alpha(\alpha) - \varphi'^2_\alpha(\alpha)}$, то уравнение дискриминантной кривой будет иметь более простой вид. Это можно сделать двумя способами.

1. Выбрать функции $\varphi(x)$, $f(x)$ так, чтобы $f'_x(x) = \varphi'_x(x)$, т. е. $f(x) = \varphi(x) + C$. Тогда $\sqrt{1 + f'^2_\alpha(x) - \varphi'^2_\alpha(x)} = 1$. Но такой выбор не всегда удовлетворяет заданным начальным условиям на входе и выходе канала.

2. При заданном уравнении средней линии канала $y = f(x)$ выбрать функции таким образом, чтобы выражение $[1 + f'^2_\alpha(x) - \varphi'^2_\alpha(x)]$ в системе (2) представляло собой полный квадрат:

$$1 + f'^2_\alpha(x) - \varphi'^2_\alpha(x) = \psi^2(x).$$

Поскольку функция $f(x)$ выбрана в соответствии с определенными требованиями, следует положить

$$\varphi'^2_\alpha(x) = 2f'_x(x).$$

Для определения функции $\varphi(x)$ необходимо решить дифференциальное уравнение

$$\varphi'^2_\alpha(x) = 2f'_x(x).$$

Отметим, что решение этого уравнения не всегда тривиально, для некоторых кривых оно не существует. Можно воспользо-

ваться другим подходом для упрощения параметрических уравнений дискриминантной кривой.

Пример. Считаем функцию $R = \varphi(x)$ известной. Тогда

$$\varphi(x) = \ln(x + \sqrt{x^2 + 1}) + a;$$

$$\varphi'_x(x) = \frac{1}{\sqrt{x^2 + 1}}; \quad f'_x(x) = \frac{1}{2(x^2 + 1)};$$

$$f(x) = \frac{1}{2} \int \frac{1}{x^2 + 1} dx = \frac{1}{2} \arctg x + C;$$

$$1 + f'^2_x(x) - \varphi'^2_x(x) = \psi^2(x);$$

$$1 - 2f'_x(x) + f'^2_x(x) = \psi^2(x); \quad \psi(x) = 1 - f'_x(x).$$

Выберем постоянные a, C так, чтобы удовлетворить требованиям, предъявляемым к функциям $y = f(x), R = \varphi(x)$. Уравнение дискриминантной кривой будет иметь вид

$$y = \frac{1}{2} \arctg a + C - \frac{[\ln(a + \sqrt{1 + a^2}) + a] 2(a^2 + 1)}{[4(a^2 + 1)^2 + 1]} \left\{ \frac{1}{\sqrt{1 + a^2}} \mp \mp (2a^2 + 1) \right\};$$

$$x = a + [\ln(a + \sqrt{a^2 + 1}) + a] \left\{ \frac{\left[\frac{1}{\sqrt{1 + a^2}} \mp (2a^2 + 1) \right]}{4(a^2 + 1)^2 + 1} - \frac{1}{\sqrt{a^2 + 1}} \right\}.$$

Список литературы: 1. Натансон И. П. Краткий курс высшей математики. — М.: Физматгиз, 1963, с. 255—264. 2. Киселев П. Г. Гидравлика. — М.: Госэнергониздат, 1963, с. 294—304. 3. Роч П. Вычислительная гидродинамика. — М.: Мир, 1980, с. 213—258. 4. Мацевитый Ю. М., Малащенко В. А. Моделирование состояния элементов турбомашин. — К.: Наук. думка, 1979, с. 171—219.

Поступила в редколлегию 29.12.84.

УДК 519.676

И. Г. ДЕНИСОВА, канд. техн. наук, Д. Г. ТРУДЛЕР

ВЕРОЯТНОСТНЫЙ МЕТОД ОЦЕНКИ КАЧЕСТВА ТРУДА В ПРОЦЕССЕ СЛОЖНОГО КОМПЛЕКСА РАБОТ

Для современного этапа развития социалистического производства характерны возрастающие требования к вопросам совершенствования во всех его звеньях [1, 2]. Наиболее ощутимые результаты в решении задач повышения эффективности производства на базе комплексного применения научных методов уп-

равления можно получить в области обеспечения качества и высокой производительности труда.

Известно, управление любым процессом предполагает осуществление следующих основных функций управления: контроль, учет, анализ, планирование и регулирование. Особое место принадлежит контролю, который предполагает возможность оценить труд исполнителя, а также группы исполнителей, образованной для решения конкретной производственной задачи.

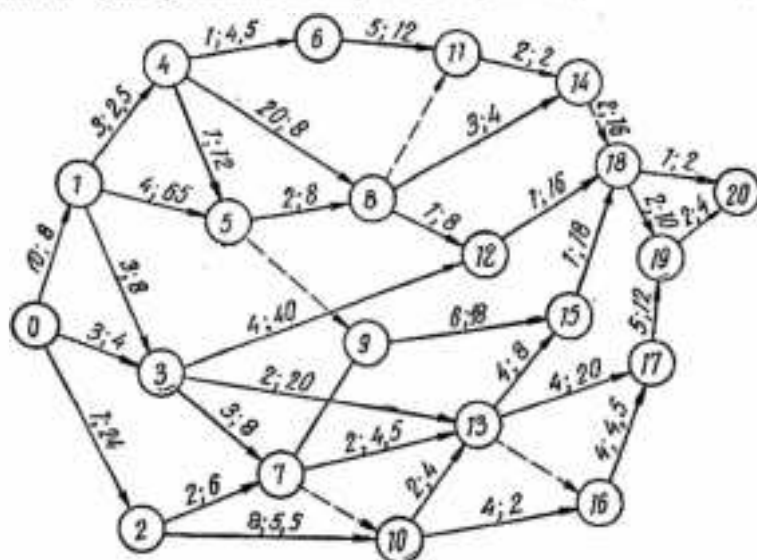


Рис. 1. Сетевой граф комплекса работ

Однако единый подход к определению такой оценки отсутствует. Предлагаемый формализованный подход к оценке качества труда групп исполнителей можно применять при создании системы управления качеством труда, а следовательно, качества конечной продукции, например, в процессах создания и освоения новой техники (СОНТ).

Любой сложный комплекс работ, как известно, можно представить сетевым графом (рис. 1). Количество единичных показателей качества труда L , число работ в сети M (без учета фиктивных работ), количество исполнителей N_0 и длительность каждой работы t_0 считаем известными.

Если рассматривать отдельно любую группу, состоящую из исполнителей, связанных некоторой работой из предлагаемой сети, то модель выполнения этой работы также можно графически представить сетью (рис. 2). Каждая ветвь этой сети соответствует определенному исполнителю. Параллельность ветвей означает совместную работу всех исполнителей. Если каждому

исполнителю работы rs поставить в соответствие вектор, состоящий из единичных показателей качества труда K_{ij} , то они образуют матрицу, каждая строка которой является оценкой качества труда j -го исполнителя (рис. 2):

$$K_{rj} = [K_{1j}, K_{2j}, \dots, K_{Lj}, \dots, K_{Lj}], j = \overline{1, N}, i = \overline{1, L}.$$

В литературе [3, 4], как правило, предложены различные подходы к получению обобщенного показателя труда для испол-

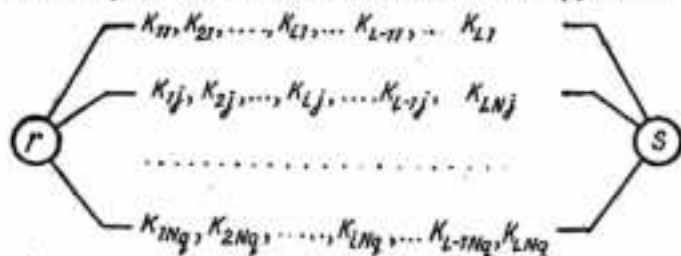


Рис. 2. Модель выполнения одной работы группой специалистов

нителя или групп исполнителей. Таким образом, детальная оценка по локальным показателям исключается, и главная роль принадлежит весовым коэффициентам локальных показателей (т. е. анализ выполняется только по строке матрицы). Однако более ценной является информация о качестве труда, исследуемая по столбцу матрицы, так как она позволяет получать локальные показатели для группы исполнителей, исходя из локальных показателей отдельных исполнителей. Затем можно перейти к оценке качества выполнения работы.

Предлагаемый метод оценки качества труда состоит в следующем. Каждый единичный показатель качества труда K_{ij} характеризует бездефектность труда j -го исполнителя по i -му свойству конкретного труда. Эту вероятную характеристику можно задавать интенсивностью ошибок λ_{ij} или вероятностью безотказной работы P_{ij} по i -му свойству конкретного труда. Единичные показатели качества труда для каждого исполнителя по всем работам сети запланированы на срок $t_{пл} \geq t_{кр}$ (время окончания комплекса работ сети). В течение этого срока такие показатели считаются неизменными.

При анализе качества труда отдельных исполнителей и групп полагаем, что нарушение установленных показателей качества труда (появление дефектов труда) наблюдается во времени под воздействием множества случайных причин. Поэтому поток i -го вида дефектов труда считаем случайным. К расчету i -го показателя качества труда привлекается теория случайных процессов. Будем говорить, что сетевая модель имеет дефект труда по i -му показателю качества труда, если у одного из исполнителей работ сети имеет место дефект труда по этому же показателю. Исполнитель, допустивший ошибку (дефект труда по i -му

показателю), в следующий момент времени работает исправно, т. е. дефект труда не влияет на качество труда исполнителя в последующее время. При этом допущении справедливо такое утверждение: поскольку любые дефекты труда исполнителя не влияют на него самого и на его труд в последующее время, число возможных дефектов труда (и мгновенных восстановлений качества труда) неограничено, все дефекты независимые случайные величины с одной и той же плотностью распределения, моменты дефектов труда образуют простой процесс восстановления. Частым случаем этого процесса является пуассоновский поток, обладающий свойствами стационарности, отсутствия последствия и ординарности.

Стационарность означает, что можно как угодно сдвигать начальную точку отсчета времени без изменения вероятности наступления определенного числа i -го вида дефектов труда для выбранного промежутка времени. Отсутствие последствия заключается в том, что условная вероятность поступления i -го вида дефектов труда за промежуток времени t , $t + \tau$ не зависит от количества дефектов труда, имевших место до момента времени t , и совпадает с безусловной вероятностью. Свойство ординарности потока предполагает невозможность появления двух и более дефектов труда i -го вида за малое время.

Перечисленные требования пуассоновского потока будут автоматически выполняться, если время между появлением i -го вида дефектов труда независимая случайная величина, распределенная по экспоненциальному закону [4].

Среднее время $t_{ср i}$ между появлением i -го вида дефектов труда определяется так:

$$t_{ср i} = 1/\lambda_i,$$

где λ_i — интенсивность появления i -го вида дефектов труда. Вероятность отсутствия i -го вида дефектов труда за время t можно вычислить по формуле

$$P_i(t) = \int_0^{\infty} \lambda_i e^{-\lambda_i t} dt = e^{-\lambda_i t} = e^{-t/t_{ср i}}. \quad (1)$$

Тогда качество труда по i -му виду дефектов характеризуется параметром $P_i(t)$.

Для определения интенсивности ошибок исполнителя, работающего с разным качеством труда на участках времени $t = t_1 + t_2 + t_3 + \dots$, используется формула

$$\lambda^{экв} = (\lambda_1 t_1 + \lambda_2 t_2 + \lambda_3 t_3 + \dots)/t, \quad (2)$$

в которой $\lambda^{экв}$ — эквивалентная интенсивность ошибок исполнителя за время; λ_i — интенсивность ошибок исполнителя на участке t_i .

Оценивая качество труда групп исполнителей, определяем векторы показателей качества труда для каждой работы сети

$$K_{трг} = \{K_{1г}, K_{2г}, \dots, K_{1г}, \dots, K_{Lг}\},$$

где $q = \overline{1, M}$ — одна из работ в сети. Вектор показателей качества труда для всей сети

$$K_{\text{тр.сети}} = \{K_1, K_2, \dots, K_i, \dots, K_L\}.$$

По каждой работе с известными векторами качества труда исполнителей $K_{\text{тр}}$, для определенности будем считать, что единичные показатели K_{ij} представляют собой интенсивности ошибок λ_{ij} по показателю i у j -го исполнителя. Вероятность безотказной работы группы исполнителей работы rs по j -му показателю качества труда зависит от вероятностей безотказной работы каждого из исполнителей по этому же показателю. Если предположить, что при этом каждый из исполнителей выполняет индивидуальное задание, т. е. отсутствует дублирование заданий в процессе работы rs , то вероятность безотказной работы группы исполнителей [5]

$$P_{qi}(t_q) = \prod_{j=1}^{N_q} P_{qij}(t_q). \quad (3)$$

Здесь

$$P_{qij}(t_q) = e^{-\lambda_{qij} t_q}. \quad (4)$$

Отметим, что произведение (3) сомножителей (4) можно привести к виду

$$P_{qi}(t_q) = e^{-\lambda_{qi} t_q}, \quad (5)$$

где λ_{qi} — интенсивность ошибок по i -му показателю качества труда работы rs , полученная суммированием элементов по столбцам матрицы (см. рис. 2), $\lambda_{qi} = \sum_{j=1}^{N_q} \lambda_{qij}$.

Набор значений $P_{qi}(t_q)$ образует искомый вектор

$$K_{\text{тр}q} = \{K_{1q}, K_{2q}, \dots, K_{iq}, \dots, K_{Lq}\},$$

причем

$$K_{1q} = P_{q1}(t_q); K_{2q} = P_{q2}(t_q); K_{iq} = P_{qi}(t_q); K_{Lq} = P_{qL}(t_q).$$

Выражение (5) позволяет найти вектора качества труда $K_{\text{тр}q}$ для всех работ сети. Вероятность безотказной работы всей сети $P_{\text{сети}} \times \times (t)$ по i -му показателю качества труда зависит от вероятности безотказной работы каждой из работ сети $P_{qi}(t_q)$ и, очевидно, также должна определяться их произведением. Однако из-за различия длительности работ t_q и времени t , для которого определяется $P_{\text{сети}}(t)$, непосредственно формула для нахождения произведений значения $P_{qi}(t_q)$, неприменима. Необходимо привести величины $P_{qi}(t_q)$ по всем работам ко времени t . Так как для всех работ справедливо условие $t_q < t$, можно считать, что группа исполнителей по некоторой работе rs за время t работает с переменной интенсивностью ошибок

$$\lambda_{qi} = \begin{cases} \sum_{j=1}^{N_q} \lambda_{qij}, & \text{за время } t_q; \\ 0, & \text{в остальное время } t - t_q. \end{cases}$$

Применяя формулу (2), определяем эквивалентные интенсивности ошибок групп исполнителей за время t :

$$\lambda_{qi}^{мс} = \lambda_{qi} t_q / t. \quad (6)$$

Находим выражения для вычисления вектора

$$K_{тр.сет} = \{K_1, K_2, \dots, K_i, \dots, K_L\}.$$

Отметим, что $K_1 = P_{сет1}(t)$;

$$K_2 = P_{сет2}(t); \dots; K_i = P_{сети}(t); \dots; K_L = P_{сетL}(t).$$

Здесь

$$P_{сети}(t) = \prod_{q=1}^M P_{qi}^{мс}(t);$$

$$P_{qi}^{мс}(t) = e^{-\lambda_{qi}^{мс} t_q};$$

$$P_{сети}(t) = e^{-\lambda_{сети} t} = e^{-\lambda'_{сети} t},$$

где λ_i — эквивалентная интенсивность ошибок по i -му показателю качества труда для всей сети, $\lambda_{сети} = \sum_{q=1}^M \lambda_{qi}^{мс}$;

$$\lambda'_{сети} = \lambda_{сети} t.$$

Алгоритм определения $K_{тр,q}$, $K_{тр.сет}$ составлен с использованием выражений (5) — (7), он включает такую последовательность шагов.

1. Определение векторов $K_{тра}$. Работа $q = 1$, обнуление эквивалентных интенсивностей ошибок сети: $\lambda_{сети} = 0$.

2. Ввод информации для работы q : длительность t_q , количество исполнителей j , $K_{трj} = \{\lambda_{1j}, \lambda_{2j}, \dots, \lambda_{ij}, \dots, \lambda_{Lj}\}$. Показатель качества труда $i = 1$.

3. Интенсивность ошибок по i -му показателю работы q : $\lambda_{qi} = 0$; исполнитель $j = 1$.

4. Накопление интенсивности ошибок по работе q : $\lambda_{qi} = \lambda_{qi} + \lambda_{ij}$.

5. Проверка числа рассмотренных исполнителей. Если $j < N_q$, то $j = j + 1$, переход к п. 4. Если $j \geq N_q$, то переход к п. 6.

6. Вычисление $P_{qi}(t_q) = e^{-\lambda_{qi} t_q}$. Накопление эквивалентной интенсивности ошибок сети. (без деления на t):

$$\lambda'_{сети} = \lambda'_{сети} + \lambda_{qi} t_q.$$

7. Проверка номера текущего показателя качества труда. Если $i < L$, то $i = i + 1$, переход к п. 3. Если $i \geq L$, то переход к п. 8.

8. Печать вектора $K_{тр,q}$, состоящего из значений $P_{qi}(t_q)$.

9. Проверка, все ли работы рассмотрены. Если $q < M$, то $q = q + 1$, переход к п. 2. Если $q \geq M$, то переход к п. 10.

10. Определение вектора $K_{тр.сет}$. Показатель $i = 1$.

ПРОГРАММА РАСЧЕТА ВЕКТОРОВ КАЧЕСТВА ТРУДА (10 ПОКАЗАТЕЛЕЙ) ДЛЯ КАЖДОЙ РАБОТЫ СЕТИ
 $P(T_R)$ И ДЛЯ ВСЕЙ СЕТИ

$P(T_R)$	номер работы 01	0,7250	0,7250	0,7285	0,7655	0,7776	0,6850	0,6586	0,5719	0,6799
	номер работы 02	0,9739	0,9859	0,9631	0,9418	0,8860	0,7860	0,8454	0,8253	0,8253
$P(T_R)$	номер работы 03	0,9194	0,9194	0,9531	0,9763	0,9646	0,9881	0,9531	0,0046	0,9646
	номер работы 1920	0,9570	0,9724	0,9846	0,9881	0,9822	0,9646	0,9920	0,9885	0,9841
		0,99196	0,00020	0,00343	0,00647	0,01144	0,01535	0,02710	0,03481	0,03009

11. Вычисление $P_{\text{сети}}(t) = e^{-\lambda'_{\text{сети}} t}$

12. Проверка номера текущего показателя качества труда. Если $i < L$, то $i = i + 1$, переход к п. 11. Если $i > L$, то переход к п. 13.

13. Печать вектора $K_{\text{тр.сети}}$, состоящего из значений $P_{\text{сети}}(t)$.

Элементы $K_{ij} = \lambda_{ij}$, являющиеся входными данными задачи, можно найти с помощью методов, изложенных в работе [5]. Длина критического пути $t_{кр}$ и планируемый период для показателей λ_{ij} не используются в алгоритме, однако необходимо помнить, что время t , для которого определяются показатели качества труда сети, должно удовлетворять условию $t_{кр} < t < t_{пл}$.

Программа для расчета на ЕС ЭВМ, вычисляющая значения векторов $K_{\text{тр}}$, $K_{\text{тр.сети}}$, реализует предложенный алгоритм. Язык программирования Фортран-4. Программных ограничений на количество работ в сети и длительность каждой работы нет, максимальное число исполнителей по одной работе и количество единичных показателей векторов качества можно произвольно выбрать (на рис. 1 $N_0 = 20$, $L = 10$).

Выходные данные в программе следующие: векторы вероятностей безотказной работы $K_{\text{тр}}$ по 10 показателям для каждой работы с номером, состоящим из номеров начального и конечного событий, количеством исполнителей и длительностью (соответственно первое и второе число над каждой работой на рис. 1); $PS(T_R)$ — вектор вероятностей безотказной работы по 10 показателям для всего комплекса работ $K_{\text{тр.сети}}$. Результаты вычислений печатаются в порядке вычисления по работам, что позволяет экономно использовать память ЭВМ.

В программе показано, в каком виде векторы $K_{трф}$, $K_{тр-сати}$ выводятся на печать.

Список литературы: 1. *Материалы* декабрьского Пленума ЦК КПСС. — М.: Политиздат, 1983. — 12 с. 2. *Материалы* февральского Пленума ЦК КПСС. — М.: Политиздат, 1984. — 9 с. 3. Румянцев А. А. Экономическая оценка НИР и ОКР. — М.: Экономика, 1978. — 135 с. 4. Артамонов В. П. Качество труда. — М.: Изд-во стандартов, 1977. — 147 с. 5. Половое А. М. Сборник задач по теории надежности. — М.: Сов. радио, 1972. — 407 с.

Поступила в редакцию 03.01.85.

УДК 681.3.068

О. П. ЛЫСНЯЯ

**ОБ ОДНОМ СПОСОБЕ ГЕНЕРАЦИИ ТАБЛИЦ
В АВТОМАТИЗИРОВАННОЙ СИСТЕМЕ
ПЕРСПЕКТИВНОГО ПЛАНИРОВАНИЯ**

Любая автоматизированная система обладает достаточно большим количеством выходной информации. Обычно для удобства пользователя она оформляется в виде таблиц, ориентированных на АЦПУ в стандартном формате.

Языки программирования высокого уровня, имеющие развитые средства вывода документов, различные системы автоматического документирования [1, 2], специализированные языки вывода документов [3] призваны облегчить процесс формирования документов любого вида. Но все они не освобождают программиста от непосредственного составления и отладки программ печати, от трудоемкой и механической работы по созданию закодированного документа с последующим формированием упрощенной программы печати. Кроме того, самые незначительные изменения таблицы, в особенности ее головки, заставляют формировать новый шаблон и, как следствие, перерабатывать программу печати. Это требует определенных усилий со стороны программиста. Поэтому автоматизация вывода табличной информации очень важна [4, 5].

В статье описывается способ автоматического генерирования таблиц, суть которого состоит в том, что для каждой части выходного документа формируется с использованием диалогового режима отдельный шаблон, практически не отличающийся от соответствующего фрагмента таблицы. Затем он преобразовывается в группу операторов печати и помещается в раздел библиотеки исходных модулей. Основная программа, осуществляющая постраничную печать документа, включает в свой текст шаблоны по именам библиотечных разделов. Поэтому корректировка любого из шаблонов, входящих в документ, а также

содержимого соответствующего библиотечного раздела не отражается в тексте основной программы.

Предлагаемый способ организации работы с шаблонами любого документа значительно упрощает работу программиста, а надежность программного обеспечения позволяет поручить составление шаблонов непосредственно пользователю.

Подобный подход к формированию выходных документов реализован в автоматизированной системе перспективного планирования научно-исследовательских и опытно-конструкторских разработок (АСПП НИОКР) [6], математическое обеспечение которой написано на языке PL/I. Рассчитывая пяти- и пятнадцатилетние планы для предприятия машиностроительной отрасли, АСПП выдает большое количество документов, оформленных в виде таблиц. К их внешнему виду и читабельности пользователем предъявляются высокие требования: все документы должны быть удобны для последующего применения.

Таблица состоит из следующих элементов.

Тема № 21

Время разработки: 1980—1986
Заказчик: НПО «Алгоритм»
Основание: приказ № 123

Тематический
заголовок

Основные характеристики, минимованные комплекс, системы	Сроки выполнения	Объем работ, шт./тыс. р.		Заголовочная часть
		Пятилетка		
		X	XI	
1	2	3	4	Нумерация граф
Собственные работы Смежные министерства Совсполнители	1980—1983	1 500		Хвостовая часть
	1982—1985	29 480	1780	
			595	
Итого:		30 980	2375	

Боковой

Программа

Специфика выходных документов АСПП НИОКР такова, что в хвостовой части несколько физических строк несут логически завершенную информацию. Для множества подобных групп строк создаются свои шаблоны. Таким образом, вся таблица может быть составлена из различных шаблонов.

Опыт эксплуатации АСПП с 1980 г. показал, что в процессе ее функционирования быстро увеличивалась потребность в новых формах выходных документов, расширялись и становились содержательнее их тематические заголовки, в старых формах добавлялись новые показатели (программы), а также перекомпоновывались имеющиеся. Даже самые незначительные изменения в таблицах заставляли пользователей прибегать к услугам

программистов. Так как формирование и выдача выходных документов, включая возможность корректировки последних, не затрагивает расчет объемных показателей НИОКР, процедуры вывода табличной информации были объединены в независимую подсистему. Работать с ней в диалоговом режиме могут программисты и пользователи.

Подсистема автоматизированного вывода табличной документации в АСПП состоит из следующих каталогизированных процедур: основной процедуры, обрабатывающей шаблон документа и составляющей операторы печати; процедуры формирования таблицы; вспомогательной процедуры создания объединенного шаблона документа.

Для формирования таблиц в работе [5] предлагается любой строке выводимого документа ставить в соответствие строку шаблона. Шаблон создается с помощью диалоговой системы коллективного доступа PRIMUS, которая позволяет наглядно представить формулу будущей таблицы и в случае необходимости быстро и качественно откорректировать ее.

Любой документ можно представить в виде строк, содержащих постоянные и переменные элементы. Переменные элементы до обработки ЭВМ характеризуются положением в строке, максимальным количеством занимаемых позиций, переменным идентификатором и форматом, по которому происходит вывод. В таблице переменными элементами следует считать номер темы, дату разработки, наименования заказчика и документов основания, а также всю хвостовую часть за исключением разделителей. В любой строке шаблона переменные элементы отмечаются знаком «а» в начале и «?» в конце, а постоянные переносятся в шаблоны без изменений. Если ширина таблицы больше 80 символов (ширина экрана дисплея), то шаблон создается из двух частей, размерность каждой из них произвольна (но менее 80). Затем эти части объединяются. Результирующий шаблон представляет собой последовательный набор данных, длина записи которого 120 (стандартная ширина таблицы на АЦПУ).

При обработке шаблона необходимо знать, как будет располагаться на листе таблица, ширина которой меньше стандартной (сдвинута она в крайнее правое или крайнее левое положение, отнесена от левого края на заданное количество позиций или расположена посредине листа). Эта служебная информация помещается в первой строке шаблона. Следом за ней располагаются данные о высоте заголовочной части. Далее заполняется первая строка любыми пустыми символами. Основная каталогизированная процедура подсистемы подсчитывает их количество и берет его в качестве ширины таблицы.

Каждой строке шаблона за исключением первой программно ставится в соответствие один или несколько операторов печати языка PL/1, переменные элементы в которых обозначены

соответствующими идентификаторами. Полученная группа операторов печати оформляется в виде раздела библиотеки исходных модулей. Когда при формировании выходного документа возникает необходимость вставки фрагмента таблицы, соответствующие операторы печати включаются в текст основной программы с помощью предпроцессорного оператора включения, использующего в качестве параметра имя раздела библиотеки исходных модулей. В этом случае всем идентификаторам переменных, встречающихся в библиотечном разделе, должны быть присвоены фактические значения.

Обычно текст выходного документа не помещается по длине на одном стандартном листе, поэтому на последующих листах, кроме их порядковых номеров, помещается нумерация характеристик. Чтобы не создавать самостоятельный шаблон для нумераций характеристик, каталогизированная процедура, входящая в подсистему автоматизированного вывода документации, использует информацию о высоте нумерации характеристик, выделяя группу последних вновь созданных операторов печати в отдельный раздел библиотеки исходных модулей. Отдельными разделами оформляются строка — подчеркивание (при печати в конце листа) и пустая строка, содержащая только разделители колонок. По мере необходимости они вызываются в основную программу.

Описанный способ работы с шаблонами выходных документов возможен не только в автоматизированной системе перспективного планирования, но и в другой системе с документами любого вида. Обширный архив выходных документов АСПП, состоящий из шаблонов и соответствующих библиотечных разделов и находящийся на внешнем носителе, нагляден и введение его не представляет трудностей. Все это облегчает процесс формирования таблиц и делает его более творческим.

Список литературы: 1. Собельман В. И., Старчеве Г. М. Средства печати в системе обработки данных. — Алгоритмы и организация решений экон. задач, 1978, вып. 12, с. 12—24. 2. Меснякина В. В., Жарносек Л. М. Пакет прикладных программ автоматизации документирования выходной информации на базе ОС ЕС ЭВМ. — Алгоритмы и организация решений экон. задач, 1978, вып. 12, с. 67—77. 3. Коченов А. И. Специализированный алгоритмический язык документирования СЛД. — Алгоритмы и организация решений экон. задач, 1973, вып. 1, с. 5—49. 4. Безруков Н. Н. Генерация программ печати документов по описанию структуры шапки. — Программирование, 1979, № 6, с. 92—95. 5. Рыбаков А. В., Борисова Е. А. Система автоматизированной обработки программ вывода документов. — Управляющие системы и машины, 1980, № 2, с. 76—80. 6. Автоматизированная система перспективного планирования НИОКР /И. П. Коц, В. И. Мелешко, Ю. А. Кудрявцева, В. И. Жулид, Т. И. Башмакова, З. Я. Красовская. — Управляющие системы и машины, 1980, № 2, с. 106—109.

Поступила в редколлегию 22.05.84.

**МЕТОДИКА ОПРЕДЕЛЕНИЯ ТЕХНИЧЕСКОГО УРОВНЯ
ПРОИЗВОДСТВА РАДИОАППАРАТУРЫ**

Задача объективной научно обоснованной комплексной оценки технического уровня производства радиоаппаратуры сложна и актуальна. Сложность ее решения связана со специфическими особенностями предприятий, занятых производством радиоаппаратуры. Специфика в большинстве случаев выражается в многономенклатурном мелкосерийном характере производства. Создаваемую на этих предприятиях аппаратуру по своим конструктивно-технологическим характеристикам, техническим и эксплуатационным требованиям, предъявляемым к ней, часто нельзя сравнить с аппаратурой, изготавливавшейся на этом же предприятии ранее или изготавливаемой в настоящее время на другом родственном предприятии. Все это не позволяет абсолютно или сравнительно оценивать технические уровни отдельных предприятий путем взаимного сопоставления таких интегральных показателей производства как себестоимость и трудоемкость изготовления единицы аппаратуры, выработку в натуральном выражении на одного работающего, съем аппаратуры с единицы производственной площади и т. д. Применявшиеся методики для оценки технического уровня производства в различных отраслях не позволяли выполнить его комплексную количественную оценку. Например, ранее была разработана методика, предусматривавшая определение всего двух показателей — коэффициента механизации работ, равного отношению трудоемкости механизированных работ к общей трудоемкости изделия; коэффициента организации, вычисляемого как отношение расчетной трудоемкости по технологическим процессам к общим трудовым затратам. Важность этих показателей не вызывает сомнения. Однако они не дают комплексной оценки технического уровня предприятия, исходя из которого можно выполнять следующее.

1. Сравнивать технические уровни отдельных предприятий и отрасли в целом.

2. Устанавливать степень соответствия технических уровней производств и предприятий объективным возможностям, вытекающим из масштабов производства и степени его специализации.

3. Определять экономическую целесообразность выбранных направлений технического совершенствования производства,

способствующих снижению трудоемкости изготовления радиоаппаратуры.

4. Выявлять резервы повышения технического уровня.

5. Определить оптимальные затраты на механизацию и автоматизацию производства для поднятия технического уровня.

В целях устранения указанных недостатков для внедрения на предприятиях, занятых изготовлением радиоаппаратуры, можно рекомендовать методику определения технического уровня производства, основанную на таком принципе. Весь производственный процесс разделяется на 15 видов производств, каждый из которых, в свою очередь, разделяются на пять видов работ. Последние подразделяются на определенное количество операций. Составляются соответствующие таблицы по видам производств и работ, а также по операциям. Для каждой операции по видам работ устанавливаются характеристики, определяющие количественно технический уровень ее выполнения. Таким характеристикам дано условное название определители. Шифр определителей состоит из шести цифр: первые две цифры обозначают вид производства, вторые — вид работ; последние — номер операции, причем первая цифра соответствует коэффициенту технического уровня выполнения этой операции, вторая — номеру данного определителя. Например шифр определителя 11-01-53 следует читать так: 11 — вид производства — производство микросхем; 01 — вид работ — выполнение заготовительных операций; 53 — номер операции данного вида работ, выполняемой с коэффициентом технического уровня 0,500 и имеющей номер определителя 3.

Определители каждого вида работ составлены таким образом, что коэффициент технического уровня 0,100 характеризует производство с самым низким техническим уровнем, а коэффициент 0,900 — самый высокий технический уровень, который можно достигнуть на данной операции, исходя из современного отечественного и зарубежного опыта автоматизации производства. Общий коэффициент технического уровня по виду работ определяется так.

При относительно равной трудоемкости операций (равное удельное значение определителей) в данном виде работ как среднеарифметическая величина коэффициентов технического уровня отдельных операций, входящих в данный вид работ,

$$Y_p = \frac{1}{n} \sum_{i=1}^n Y_{оп. i} \quad (1)$$

где Y_p — коэффициент технического уровня вида работ; $Y_{оп. i}$ — коэффициент технического уровня операции; n — число операций или определителей.

В случае значительных отклонений удельных значений определителей в данном виде работ как средневзвешенная величина

коэффициентов технических уровней отдельных операций (определителей), входящих в данный вид работ:

$$Y_n = \sum_{i=1}^n \frac{Y_{onf} K_{onf}}{100}, \quad (2)$$

где K_{onf} — удельное значение операции (определителя) в данном виде работ, выраженное в % от общей трудоемкости вида работ.

Коэффициент технического уровня вида производства определяется так же, как средневзвешенная величина коэффициентов технического уровня видов работ, входящих в данный вид производства в зависимости от удельного объема каждого из видов работ:

$$Y_{np} = \sum_{i=1}^m \frac{Y_{pfi} K_{pfi}}{100}, \quad (3)$$

где Y_{np} — коэффициент технического уровня вида производства; K_{pfi} — удельный объем вида работ в общей трудоемкости данного вида производства; m — количество видов работ.

Коэффициент технического уровня изготовления изделий на данном предприятии определяется как средневзвешенная величина коэффициентов видов производств с учетом удельного объема каждого вида производства в изготовлении данного изделия:

$$Y_{изд} = \sum_{i=1}^p \frac{Y_{npfi} K_{npfi}}{100}, \quad (4)$$

где $Y_{изд}$ — коэффициент технического уровня изготовления изделия; K_{npfi} — удельный объем вида производства в общей трудоемкости изделия; p — количество видов производств, участвующих в изготовлении данного изделия.

Коэффициент технического уровня по предприятию, изготавливающему одновременно несколько изделий, определяется как средневзвешенная величина коэффициентов технических уровней основных изделий с учетом удельного объема работ по каждому из них в общем объеме производства:

$$Y_z = \frac{\sum_{i=1}^k Y_{издfi} K_{издfi}}{\sum_{i=1}^k K_{издfi}}, \quad (5)$$

где Y_z — коэффициент технического уровня предприятия; $K_{издfi}$ — удельный объем трудоемкости изготовления данного изделия от общей трудоемкости всех изделий, изготавливаемых

на предприятии, %; S — количество изделий, по которым рассчитывается технический уровень предприятия.

Данная методика позволяет определять коэффициент технического уровня группы родственных предприятий или в целом отрасли. Руководствуясь описанной методикой и таблицей определителей по всем видам производств и работ, можно решить обратную задачу: при директивно заданном росте коэффициента технического уровня по годам выявить резервы повышения технического уровня, определить наиболее перспективные направления механизации и автоматизации производства. Выполняя необходимые экономические оценки возможных направлений механизации и автоматизации производства, можно найти оптимальный вариант затрат для достижения заданного коэффициента технического уровня.

Поступила в редколлегию 22.06.84.

УДК 621. 529

Ю. И. ТРОФИМОВ, В. С. ФЛАНЧИК, канд. техн. наук

ЭФФЕКТИВНОСТЬ АЛГОРИТМОВ КРУГОВОЙ ИНТЕРПОЛЯЦИИ НА ОСНОВЕ ЦДА В СИСТЕМАХ ЧИСЛОВОГО ПРОГРАММНОГО УПРАВЛЕНИЯ ОБОРУДОВАНИЕМ

В системах числового программного управления (ЧПУ) металлообрабатывающим оборудованием одним из основных алгоритмов функционирования является интерполяция заданного контура участками прямых линий или дугами окружностей. В аппаратных устройствах ЧПУ функции интерполирования реализованы с помощью специальных устройств-интерполаторов. Широко распространены интерполаторы, основанные на вычислении и анализе знака оценочной функции [1]. Эти устройства реализуют помпультное управление.

В системах ЧПУ, использующих микроЭВМ в качестве центрального вычислителя, интерполяция осуществляется программными средствами. Поскольку быстрдействие вычислителя для обеспечения высоких скоростей перемещения подвижных органов недостаточно (особенно при малой дискретности измерительных преобразователей и высоких скоростях перемещений с использованием помпультного управления), необходимо реализовать алгоритмы интерполяции в многоразрядных приращениях. Вычисляемые микроЭВМ многоразрядные приращения можно непосредственно обрабатывать приводом или преобразовывать в унитарный код с помощью микроинтерполатора.

Так как управление осуществляется в реальном масштабе времени, для нормального функционирования системы ЧПУ важен учет времени реализации алгоритмов управления, которое определяется производительностью вычислителя и эффективностью алгоритмов управления.

Из известных алгоритмов интерполяции в современных системах ЧПУ применяются алгоритмы цифрового интегрирования (алгоритмы ЦДА), позволяющие реализовать постоянство контурной скорости. Они описываются аналитическими зависимостями, относительно несложными при программной реализации. Алгоритмы ЦДА обеспечивают высокую точность при линейной интерполяции. В случае круговой интерполяции точность, достигаемая в ре-

зультате функционирования алгоритмов ЦДА, представляет предмет самостоятельного исследования.

Известно, существенный недостаток алгоритмов ЦДА заключается в методической погрешности, обусловленной заменой интеграла конечной суммой. В результате поисков путей и методов уменьшения этого недостатка был найден ряд алгоритмов. Их можно разбить на две основные группы. К первой относятся те алгоритмы, которые представляют собой модификации базового классического алгоритма ЦДА. В нем осуществляется однократное интегрирование. Ко второй — базовый алгоритм, отличающийся двойным интегрированием.

В связи с тем, что в дальнейшем будут повышаться точность обработки на станках с ЧПУ, увеличиваться скорости подач, вопросы, связанные с определением путей дальнейшего повышения точности интерполяции (особенно круговой) актуальны.

В работах [2, 4] изучены вопросы точности различных алгоритмов ЦДА, приведены выражения для оценки точности интерполяции.

Однако общая оценка пределов точности, а также некоторая предельная форма, ограничивающая развитие структуры алгоритмов на основе ЦДА, в данных работах не приведены.

В статье предложена такая форма. Она получена на основе чисто геометрического подхода без привлечения дифференциальных форм.

При движении по дуге окружности в первом квадранте против часовой стрелки из точки M_0 в M_1 справедливо уравнение

$$(x_0 + \Delta x)^2 + (y_0 - \Delta y)^2 = R^2. \quad (1)$$

Здесь R — радиус окружности; x_0, y_0 — координаты начальной точки дуги; $\Delta x, \Delta y$ — приращения по координатам. Учитывая, что $\Delta x^2 + \Delta y^2 = \lambda^2$, где λ — шаг интерполяции, после преобразований получаем уравнение относительно Δx :

$$\Delta x^2 + x_0 \frac{\lambda^2}{R^2} \Delta x + \frac{\lambda^2}{4R^2} (\lambda^2 - 4y_0^2) = 0. \quad (2)$$

Решение уравнения (2) имеет следующий вид:

$$\Delta x = -\frac{\lambda^2}{2R^2} x_0 + \frac{\lambda}{2R} \cdot 2y_0 \sqrt{1 - \frac{\lambda^2}{4R^2}} \quad (3)$$

Используя обозначение $q = \lambda/R$, приводим выражение (3) к виду

$$\Delta x = -\frac{q^2}{2} x_0 + \frac{q}{2} \cdot 2y_0 \sqrt{1 - \frac{q^2}{4}}. \quad (4)$$

Аналогично можно получить выражение для приращения Δy :

$$\Delta y = \frac{q^2}{2} y_0 + qx_0 \sqrt{1 - \frac{q^2}{4}}. \quad (5)$$

Значения координат точек окружности на i -м шаге интерполяции определяются так:

$$x_i = x_{i-1} + \frac{q^2}{2} x_{i-1} + qy_{i-1} \sqrt{1 - \frac{q^2}{4}} = \left(1 - \frac{q^2}{2}\right) x_{i-1} + qy_{i-1} \sqrt{1 - \frac{q^2}{4}};$$

$$y_i = y_{i-1} - \frac{q^2}{2} y_{i-1} - qx_{i-1} \sqrt{1 - \frac{q^2}{4}} = \left(1 - \frac{q^2}{2}\right) y_{i-1} - qx_{i-1} \times \\ \times \sqrt{1 - \frac{q^2}{4}}. \quad (6)$$

Уравнения (6) являются математическим описанием алгоритма круговой интерполяции в многоразрядных приращениях при заданном значении q . Нетрудно убедиться, что эти уравнения имеют нулевую методическую погрешность.

Покажем, что форма (6) служит обобщением алгоритмов ЦДА при некоторых допущениях.

1. Поскольку на практике шаг интерполяции значительно меньше шага окружности, т. е. q — малая величина, пренебрегая значением q^2 , получаем

$$x_i = x_{i-1} + qy_{i-1}; \\ y_i = y_{i-1} - qx_{i-1}. \quad (7)$$

Выражения (7) определяют классический алгоритм ЦДА (алгоритм I).

2. Следующую ступень приближения получают, если считать, что $q \neq 0$, $\sqrt{1 - q^2/4} \approx 1$. В этом случае уравнения (6) приводятся к следующему виду:

$$x_i = x_{i-1} + q[y_{i-1} - (q/2)x_{i-1}]; \\ y_i = y_{i-1} - q[x_{i-1} + (q/2)y_{i-1}]. \quad (8)$$

Зависимости (8) описывают алгоритм ЦДА с двойным интегрированием (алгоритм II).

3. Если $\sqrt{1 - \frac{q^2}{4}}$ разложить в биномиальный ряд, ограничившись вторым членом разложения, получим

$$\left(1 - \frac{q^2}{4}\right)^{1/2} \approx 1 - \frac{q^2}{8}.$$

Учитывая это выражение, из уравнений (6) находим

$$x_i = x_{i-1} + q[y_{i-1} - (q/2)[x_{i-1} + (q/4)y_{i-1}]]; \\ y_i = y_{i-1} - q[x_{i-1} + (q/2)[y_{i-1} - (q/4)x_{i-1}]]. \quad (9)$$

По аналогии с зависимостями (8) алгоритм, описанный уравнениями (9), будем называть алгоритмом ЦДА с тройным интегрированием (алгоритм III). Последний в литературе не рассматривался.

Выполним анализ погрешностей рассмотренных алгоритмов. В качестве критерия сопоставления точности алгоритмов будем рассматривать погрешность по радиусу на один шаг интерполяции, а также накопленную погрешность при обработке окруж-

ности. Погрешность по радиусу является геометрической суммой погрешностей по каждой из управляемых координат и определяется разностью между вычисленным после выполнения i -го шага алгоритма и заданным радиусами.

Для каждого из рассмотренных алгоритмов погрешности вычисления радиуса на шаг интерполяции имеют следующий вид.

Для алгоритма, описываемого уравнениями (6),

$$\delta^* R = 0.$$

Применяя алгоритмы круговой интерполяции, можно решать вопрос о погрешности интерполяции. Поскольку вычисление корня в уравнениях (6) требует больших затрат времени микроЭВМ, поэтому в системах ЧПУ нецелесообразно использовать эти уравнения.

Для алгоритмов I, II, III имеем

$$(\delta^* R)_I = \frac{\delta R}{R} = \sqrt{1 - q^2} - 1 \simeq \frac{q^2}{4}; \quad (10)$$

$$(\delta^* R)_{II} = \sqrt{1 + \frac{q^4}{4}} - 1 \simeq \frac{q^4}{8}; \quad (11)$$

$$(\delta^* R)_{III} = \sqrt{1 + \frac{q^6}{64}} - 1 \simeq \frac{q^6}{128}. \quad (12)$$

Накопленная погрешность ΔR при отработке полной окружности для каждого из перечисленных выше алгоритмов определяется разностью между радиусом, найденным на последней итерации алгоритма, и заданным радиусом окружности.

Если R — радиус окружности, а λ — шаг интерполяции, то число шагов интерполяции i при отработке полной окружности

$$i = \frac{2\pi R}{\lambda} = \frac{2\pi}{q}.$$

Для алгоритма I

$$R_i = R_0 \left(1 + \frac{q^2}{2}\right)^{2\pi/q}.$$

Тогда $\Delta R = R_0 \left[\left(1 + \frac{q^2}{2}\right)^{2\pi/q} - 1\right]$. Разложим $\left(1 + \frac{q^2}{2}\right)^{2\pi/q}$ в биномиальный ряд. Ограничиваясь третьим членом разложения, получаем

$$\Delta^* R = \frac{\Delta R}{R_0} = \frac{\pi q}{4} [4 + q(2\pi - q)]. \quad (13)$$

Для алгоритма II имеем

$$\Delta^* R = \frac{\pi q^3}{64} [16 + q^3(2\pi - q)], \quad (14)$$

а для алгоритма III

$$\Delta^* R = \frac{\pi q^5}{16384} [256 + q^5 (2\pi - q)]. \quad (15)$$

При сопоставлении указанных алгоритмов для обоснования рациональной области применения будем исходить из сравнения точности воспроизведения контура и продолжительности вычислительной процедуры. Результаты сопоставления относительной погрешности $\Delta^* R$ как функции параметра q для каждого из рассмотренных алгоритмов показывают, что алгоритм III обеспечивает более высокую точность. Сравнивая погрешности алгоритмов II, III, делаем вывод, что при $q=0,125$ погрешность алгоритма III на три порядка меньше погрешности алгоритма II. При дальнейшем уменьшении q отношение погрешностей алгоритмов II, III значительно возрастает.

Таким образом, алгоритм III является конечным в развитии базовых структур алгоритмов ЦДА.

При практическом использовании алгоритма III в системах ЧПУ представляет интерес оценка продолжительности вычислительной процедуры. Для этого приведем уравнения (9) к виду

$$x_i = Ax_{i-1} + By_{i-1}; \quad y_i = Ay_{i-1} - Bx_{i-1}, \quad \text{где } A = \left(1 - \frac{q^2}{2}\right); \quad (16)$$

$$B = \left(1 - \frac{q^2}{8}\right).$$

Отсюда следует, что по сравнению с алгоритмом II в выражении коэффициента B добавляется сомножитель $\left(1 - \frac{q^2}{8}\right)$.

Поэтому вычисление B в отличие от A отличается двумя командами сдвига. Что касается операции умножения на сомножитель B , дополнительные затраты времени при наличии аппаратного расширителя для умножения, который применяется в УЧПУ 2С85, исчисляются десятками микросекунд.

Таким образом, продолжительность вычислений согласно алгоритму III по сравнению с алгоритмом II возрастает незначительно. Точность алгоритма III для практически встречающихся значений на несколько порядков превосходит точность алгоритма II.

Предложенный алгоритм ЦДА с тройным интегрированием целесообразно применять в системах ЧПУ высокоточным оборудованием, а также для обеспечения повышения скоростей обработки при сохранении точности.

Поступила в редакцию 08.02.85.

ОБ ЭФФЕКТИВНОСТИ ДВУХ АЛГОРИТМОВ
ОЦЕНИВАНИЯ, РЕАЛИЗУЕМЫХ НА ЭВМ

В теории оценивания получены основополагающие результаты, позволяющие выявить потенциальные возможности (точность, скорость сходимости) различных алгоритмов. Отдельные вопросы точности и сходимости алгоритмов оценивания с учетом погрешностей, возникающих при их реализации на ЭВМ, рассмотрены в работах [1—4]. Однако данные, с помощью которых можно в явном виде определять эффективность алгоритмов и оптимальное количество обрабатываемых измерений в зависимости от разрядности ЭВМ и параметров оцениваемых сигналов, отсутствуют.

В статье исследованы два наиболее часто реализуемых на ЭВМ алгоритма оценивания постоянного сигнала s :

алгоритм усреднения

$$y[N] = \left(\sum_{n=1}^N x[n] \right) / N \quad (1)$$

и его рекуррентный вариант

$$y[n] = \frac{1}{n} x[n] + \frac{(n-1)}{n} y[n-1], \quad (2)$$

где $x[n] = s + \xi[n]$; $\xi[n]$ — аддитивная помеха в n -м измерении ($n = \overline{1, N}$); $y[n]$ — оценка сигнала s по n измерениям. Алгоритм (2) превосходит (1) по быстродействию (позволяет получать несмещенную оценку сигнала s для каждого n -го измерения). Пусть справедливы следующие допущения:

$$M\{\xi[n]\} = 0; M\{\xi^2[n]\} = \sigma_\xi^2; M\{\xi[n]\xi[m]\} = 0. \quad (3)$$

Здесь M — символ математического ожидания; $m = \overline{1, N}$ ($n \neq m$). Эффективность алгоритмов будем характеризовать дисперсией оценки сигнала

$$\sigma^2 = M\{[y[n] - s]^2\}.$$

Найдем дисперсию оценки сигнала s для алгоритма (1) и (2) соответственно. Считаем погрешность округления не зависящей от $x[n]$. Тогда дисперсия оценки после обработки N измерений сигнала s будет равна сумме двух составляющих. Первая обусловлена остаточной помехой из-за эффекта усреднения с дисперсией σ_ξ^2/N , вторая — погрешностями округления. Выражения для

среднеквадратического отклонения ошибки округления получены в работе [1]:

$$\sigma_1 = \frac{2^{-q}}{3} \sqrt{NM[x^2[n]]}; \quad (4)$$

$$\sigma_2 = (3N + 7) 2^{-(q+1)} \sqrt{M[x^2[n]]}. \quad (5)$$

При этом полагаем, что $x[n]$, N нормализованы и числа двоичных разрядов их мантисс равны q .

Воспользуемся допущениями (3), откуда

$$M[x^2[n]] = s^2 + \sigma_s^2.$$

Вводя отношение сигнал/шум $\rho = s^2/\sigma_s^2$, получаем дисперсии оценок сигнала в виде

$$\mu_1^2 = \sigma_s^2 \left[\frac{1}{N} + \frac{N(\rho+1)}{9 \cdot 2^{2q}} \right]; \quad (6)$$

$$\mu_2^2 = \sigma_s^2 \left[\frac{1}{N} + \frac{(3N+7)^2(\rho+1)}{9 \cdot 2^{2(q+1)}} \right]. \quad (7)$$

Из формул (6), (7) следует, что алгоритмы (1), (2) при реализации на ЭВМ не обеспечивают сходимости оценок к истинному значению сигнала в случае неограниченного увеличения числа измерений. Из условия $\partial \mu_1^2 / \partial N = 0$ найдем оптимальное число измерений $N_{\text{опт}}$, при котором дисперсия оценки сигнала μ_1^2 минимальна:

$$N_{\text{опт}} = \frac{3 \cdot 2^q}{\sqrt{1+\rho}}. \quad (8)$$

Аналогично

$$N_{2\text{опт}} = \sqrt[3]{\frac{2^{2q+1}}{9(\rho+1)}}. \quad (9)$$

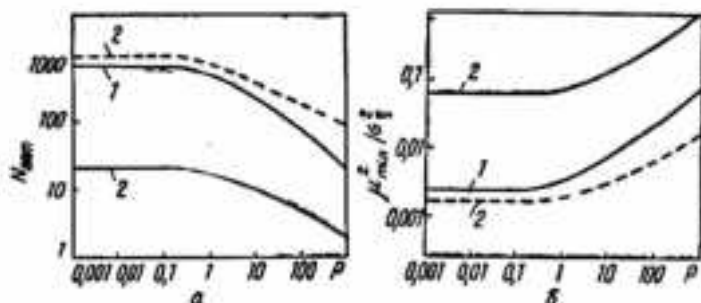
Формулы (6) — (9) справедливы для разрядности $q \geq 8$ (ошибки округления с высокой достоверностью можно считать случайными [4]).

На рисунке изображены кривые, построенные с помощью формул (6) — (9). Кривая 1 соответствует алгоритму (1), 2 — алгоритму (2), сплошная линия — разрядности $q=8$, пунктирная — $q=16$. Из анализа рисунка следует, что при одинаковой разрядности вычислений алгоритм (1) эффективнее алгоритма (2) по точности. В частности, оценка сигнала по алгоритму (1) для разрядности $q=8$ незначительно отличается по точности от оценки, получаемой на 16-разрядной ЭВМ с помощью алгоритма (2).

При малых отношениях квадрата сигнала к дисперсии помехи ($\rho < 0,1$) число оптимально обрабатываемых измерений, а также возможность повышения точности оценки ограничи-

ваются разрядностью вычислений. В предельном случае из формул (8), (9) получаем

$$\lim_{p \rightarrow 0} N_{1\text{опт}} = 3 \cdot 2^q; \quad \lim_{p \rightarrow 0} N_{2\text{опт}} = \sqrt[3]{\frac{2^{2q+1}}{9}}.$$



Кривые изменения $N_{\text{опт}}$ (а) и $\mu^2_{\text{min}}/\sigma^2$ (б) для различных значений p, q

С увеличением отношения сигнал/шум p сокращается число измерений, необходимое для получения минимально возможной для заданной разрядности вычислений дисперсии оценки. В предельном случае $\lim_{p \rightarrow \infty} N_{1\text{опт}} = 0$; $\lim_{p \rightarrow \infty} N_{2\text{опт}} = 0$.

Если задано число обрабатываемых измерений N_0 , то разрядность вычислений согласно формулам (8), (9) должна удовлетворять соотношениям

$$q_1 > \log_2 \left(\frac{N_0 \sqrt{1+p}}{3} \right) \quad (10); \quad q_2 > \frac{3 \log_2 (N_0 \sqrt[3]{9(p+1)}) - 1}{2}. \quad (11)$$

При данной разрядности в N_0 отсутствуют те измерения, обработка которых вызывает лишь увеличение дисперсии оценки.

Рассмотренные алгоритмы оценивания при их реализации на ЭВМ не обеспечивают сходимости оценок в случае увеличения числа измерений. Для них существует оптимальное число обрабатываемых измерений, когда дисперсия оценки минимальна. Из соотношений (10), (11) следует, что разрядность ЭВМ целесообразно выбирать тем меньше, чем меньше отношение сигнала к помехе и число обрабатываемых измерений.

Список литературы: 1. Коржова В. Н. Оптимизация алгоритмов оценки математического ожидания случайной величины в различных режимах обработки данных. — Кибернетика, 1978, № 5, с. 96—101. 2. Лим Д., Сейдж Э. Алгоритмы дискретного последовательного максимума правдоподобия для оценки смещения и анализ соответствующих ошибок. — Экспресс-информ. Техн. кибернетика, 1972, № 12, с. 1—26. 3. Симсарьян Р. А. О вычислительной погрешности адаптивного алгоритма оценки среднего. — Автоматика и телемеханика, 1979, № 4, с. 156—161. 4. Соренков Э. И., Телига А. И., Шаталов А. С. Точность вычислительных устройств и алгоритмов. — М.: Машиностроение, 1976.—200 с.

Поступила в редколлегию 12.01.85.

Евдокимов А. Г., Федоров Н. В., Козыренко С. И. Эффективный алгоритм идентификации состояния модели установившегося потока-распределения в инженерных сетях	3
Иванов С. Ю., Солодовников И. В. Процедуры оптимального проектирования сложных систем с использованием имитационного моделирования	9
Бодянский Е. В. Автоматический синтез моделей в САПР адаптивных систем	14
Курилин В. Л., Тригуб М. В. Алгоритм решения задачи синтеза приближенно-оптимального управления для нелинейного объекта	18
Жигляевский А. А., Терентьева М. В. Статистические методы в глобальном случайном поиске. Описание алгоритмов	24
Сенченко Н. И., Гладенко Н. И. Оценки максимальных значений решений дифференциальных уравнений методом интервалов	29
Писаренко Г. Г., Шматков С. И. Сравнение методов оценивания средних двух нормальных совокупностей по результатам групповых измерений	34
Мороз Б. Н., Аникина Е. В. О выборе критерия эффективности информационно-измерительной системы для автоматизации научных исследований	38
Теслюк С. Ф. Оценка параметрической устойчивости ЭВМ методом граничного контроля	41
Ткаченко С. Н., Козий С. П. Объектно-ориентированный метод синтеза компонентов вычислительных систем	45
Ворона В. М., Мурашко А. Г., Андреева А. В. Алгоритмическое обеспечение подсистемы проектирования специализированной вычислительной структуры	49
Блинов А. И., Калявин В. П., Мазин А. В., Малышев А. М. Оптимизационная постановка задач обоснования требований к системам диагностирования объектов управления	55
Свиридов В. В., Михнов Д. К. Исследование рациональных методов переработки частотных сигналов для оценки высоты музыкальных звуков	60
Журавковский Ю. П., Моргалъ О. М. Определение кодового расстояния многопозиционных кодов	65
Кирьяков Ю. К. Кодирование внутренних состояний микропрограммных автоматов по методу включения	68
Филиппенко О. И. Преобразователь кодов	75
Старченко А. Н., Теплов С. А. Использование методов и алгоритмов дедуктивного вывода для интеграции неоднородных баз данных	78
Каланников В. А., Молдавский Л. Н., Тищенко В. А. Структура исходных данных для пакета прикладных программ восстановления структурных таблиц нерегулярных ПЛМ по топологии БИС	82
Губка С. А., Аникин А. Н., Дергачев В. А. Выбор оптимального состава системы встроенного контроля логических модулей ГАП	85
Коренева Г. В. Некоторые вопросы управления модулем гибкого автоматизированного производства	90
Литвинов М. Л. Определение оптимального межпрофилактического периода при эмпирическом задании распределения времени до отказа	94
Литвинов А. Л. Информационно-управляющая система автоматизированного склада для гибкого автоматизированного производства	97
Нефедов Л. И. Метод определения оптимальной по изоляции помещений ориентации зданий в САПР жилой застройки	102
Товченко В. И. Модель имитации многовариантных организационно-технологических схем строительства	107
Онищенко Г. И., Шапкина Т. Я. Математическое моделирование решеток и лопаток турбомашин	112
Денисова Н. Г., Трудлер Д. Г. Вероятностный метод оценки качества труда в процессе сложного комплекса работ	114

Лысяная О. П. Об одном способе генерации таблиц в автоматизированной системе перспективного планирования	121
Бреславец А. В., Хижняк Ю. Г. Методика определения технического уровня производства радиоаппаратуры	125
Трофимов Ю. Н., Фланчик Б. С. Эффективность алгоритмов круговой интерполяции на основе ЦДА в системах числового программного управления оборудованием	128
Родионов Ю. В. Об эффективности двух алгоритмов оценивания, реализуемых на ЭВМ	133

АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ И ПРИБОРЫ АВТОМАТИКИ

Выпуск 79

Редактор Т. Н. Водолажская
Художественный редактор Т. П. Короленко
Технический редактор Л. Т. Ена
Корректоры В. Л. Светличная, Е. В. Сергина

Информ. бланк № 10336

Сдано в набор 06.09.86. Подл. в печать 17.12.86. Бл 09463. Формат 60×90/16. Бумага типогр. № 3. Лит. гарн. Вис. печати 8,5 печ. л. 8,75 кр.-отт. 9,9 уч.-изд. л. Тираж 800 экз. Изд. № 1426. Зак. 5-370. Цена 1 р. 40 к.

Издательство при Харьковском государственном университете издательского объединения «Вища школа», 310068, Харьков-3, ул. Университетская, 16

Харьковская книжная фабрика «Коммунист», 310012, Харьков-12, ул. Зигельска, 11

т. р. 40 к.

АВТОМАТИЗИ- РОВАННЫЕ СИСТЕМЫ УПРАВЛЕНИЯ

И ПРИБОРЫ АВТОМАТИКИ

**РЕСПУБЛИКАНСКИЙ
МЕЖДУВЕДОМСТВЕННЫЙ
НАУЧНО-ТЕХНИЧЕСКИЙ
СБОРНИК**

Основан в 1965г.

АСС и приборы автоматизации, ISSN, вып. 29, 1--238

