

**В О П Р О С Ы  
ПРОГРАММИРОВАНИЯ  
И  
АВТОМАТИЗАЦИИ  
ПРОЕКТИРОВАНИЯ**

цади...» (К. Маркс и Ф. Энгельс. Соч., т. 25, ч. II, 11, 12)  
ное ведение сельского хозяйства, как правило, связа  
нительных капитальных вложений на единицу зе  
эти затраты обеспечивают значительный рост п  
ее себестоимости.

Анализируя развитие капитализма в селе, аналитич  
нин убедительно показал, что интенсификация произв  
мерный, а не эпизодический процесс развития сель  
это утверждали сторонники антинаучного «революцион  
зающего плодородия почв», не учитывая  
ния техники, состояния производства  
статистическом материале показал, что  
редств в сельское хозяйство, как п  
и, с введением новых, улучшенн  
роизводства. В работе «Аграрны  
исал: «В сущности ведь самое  
ельные) вложения труда и ка  
ов производства, преобраз  
ельных размерах количеств  
добести новые машины,

ии сельск

ное направ

ИФИКАЦИЯ



# ВОПРОСЫ ПРОГРАММИРОВАНИЯ И АВТОМАТИЗАЦИИ ПРОЕКТИРОВАНИЯ

(Выпуск 3)



ИЗДАТЕЛЬСТВО ТОМСКОГО УНИВЕРСИТЕТА

Томск — 1977

Статьи, представленные в настоящем сборнике, посвящены проблемам автоматизации исследования цепей, проектирования технологических процессов, а также автоматизации программирования. Сборник предназначен для специалистов в области системного программирования и теории цепей.

Зуб 2472

Ответственный редактор — профессор,  
доктор технических наук Л. М. Ананьев

3-3-14

© Издательство Томского университета, 1977 г.

2472



## **СИМВОЛИЧЕСКОЕ МОДЕЛИРОВАНИЕ И АНАЛИЗ ЭЛЕКТРОННЫХ СХЕМ НА ЭВМ**

**Е. А. АРАИС, В. М. ДМИТРИЕВ**

### **Введение**

К настоящему времени создано несколько систем автоматического моделирования и анализа радиоэлектронных схем [1, 2]. Методы, положенные в их основу, можно квалифицировать как численные, поскольку все основные операции в них производятся над численными матрицами и векторами. Отметим наиболее характерные недостатки численных методов.

1. Ограничен класс элементов, составляющих схемы.
2. Вычислительный процесс на ЭВМ производится с некоторой погрешностью, которую трудно оценить и зачастую невозможно устранить.
3. Довольно сложной является задача ввода в исследуемую схему элементов с новыми характеристиками.
4. При изменении первичных параметров схемы весь процесс моделирования и анализа приходится повторять вновь, что связано с большими затратами машинного времени, особенно при оптимизации и статистическом исследовании схем.

Благодаря подстановке символов вместо чисел удается частично или полностью устранить указанные недостатки. Кроме того, символическое представление цепи позволяет конструктору получить ее модель в наглядной форме. В такой модели легко выделить взаимосвязи между параметрами, ее можно также использовать при моделировании более сложных систем.

Символический анализ схем уже рассматривался в ряде работ [3, 6]. Однако его использование ограничивалось пока анализом линейных схем, а предлагаемые алгоритмы явля-

ются громоздкими и неудобными для программирования на ЭВМ.

Одной из основных причин низкой эффективности разработанных методов является, по-видимому, отсутствие специальных систем программирования аналитических выкладок на ЭВМ, которые могли бы использоваться в качестве базовых. Хотя затраты на разработку таких систем являются значительными, впоследствии они окупаются простотой и универсальностью алгоритмов решения ряда прикладных задач.

Цель настоящей статьи — дать общее описание метода символического анализа, реализованного в системе программирования Авто-Аналитик, на ЭВМ БЭСМ-6, структура и возможность которой описаны в [4, 6]. Приводимые ниже алгоритмы являются универсальными и могут быть использованы как для системы Авто-Аналитик, так и для любой другой, эквивалентной ей по своим возможностям.

## 1. МОДЕЛИРОВАНИЕ ЦЕПЕЙ

### § 1. Символические модели компонентов цепи

Прежде всего введем определения основных понятий, использованных при описании предлагаемой методики.

Цепью будем называть произвольную совокупность многополюсников, выводы которых, именуемые ветвями цепи, соединены в точки, именуемые узлами цепи. Совокупности выводов (ветвям цепи с номерами  $0, 1, 2, \dots, m$ ) соответствует вектор токов  $J = \{J_0, J_1, \dots, J_m\}$ . Узлам цепи  $N_0, N_1, \dots, N_k$  соответствует вектор напряжений  $U = \{U_0, U_1, \dots, U_k\}$ . Вектор  $V = \{U, J\}$  будем называть в дальнейшем вектором переменных цепи.

Каждый компонент цепи представляет собой произвольный многополюсник (рис. 1). Все его полюса и соответствующие им выводы нумеруются в произвольном порядке. Каждому выводу с номером  $i$  соответствует ток  $J$  и напряжение  $U$ .

Математическая модель  $n$ -полюсника, имеющего выводы  $B_0, B_1, \dots, B_{n-1}$ , есть система  $n$  уравнений, связывающая токи  $J_0, J_1, \dots, J_{n-1}$  и напряжения  $U_0, U_1, \dots, U_{n-1}$ , которые будем называть внешними параметрами многополюсника. Совокуп-



ность внешних параметров всех входящих в цепь многополюсников образует переменные цепи. Внутренними параметрами многополюсника (обозначим их  $a_0, a_1, \dots, a_{b-1}$ ) могут быть его электрические, электрофизические, конструктивно-технологические параметры.

Многополюсник (рис. 1) будем называть автономным, если его внешние параметры могут быть выражены через внутренние параметры, и неавтономным в противном случае. Модели неавтономного многополюсника соответствует недоопределенная система уравнений. Всякую цепь, очевидно, можно отождествить с автономным многополюсником.

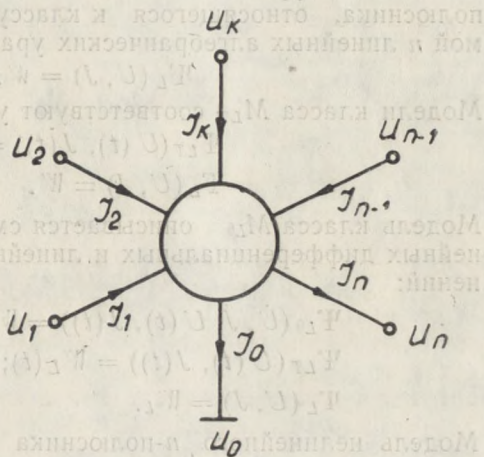


Рис. 1

На основании введенных определений произведем классификацию моделей компонентов. Принадлежность модели к тому или иному классу определяется видом входящих в нее уравнений (линейные или нелинейные), физическими свойствами компонента (инерционный или безынерционный) и характером протекающих в нем процессов (постоянного или переменного тока). Для последующего использования введем обозначения:  $L$  — признак линейности модели,  $N$  — нелинейности,  $T$  — зависимости от времени,  $\Theta$  — инерционности. Тогда классификация моделей многополюсников будет выглядеть следующим образом:

- $M_L$  — линейные безынерционные постоянного тока;
- $M_{LT}$  — линейные безынерционные переменного тока;
- $M_{L\Theta}$  — линейные инерционные переменного тока;
- $M_N$  — нелинейные безынерционные постоянного тока;
- $M_{NT}$  — нелинейные безынерционные переменного тока;
- $M_{N\Theta}$  — нелинейные инерционные переменного тока.

Необходимо отметить, что такая классификация рассматривает задающие источники тока и напряжения наравне

с другими компонентами цепи. Так, например, источник постоянного тока имеет модель класса  $M_L$ , а переменного тока — класса  $M_{LT}$ .

Запишем уравнения моделей каждого класса. Модель  $n$ -полюсника, относящегося к классу  $M_L$ , описывается системой  $n$  линейных алгебраических уравнений вида

$$\Psi_L(U, J) = W. \quad (1.1)$$

Модели класса  $M_{LT}$  соответствуют уравнения:

$$\begin{aligned} \Psi_{LT}(U(t), J(t)) &= W(t); \\ \Psi_L(U, J) &= W. \end{aligned} \quad (1.2)$$

Модель класса  $M_{L0}$  описывается смешанной системой из линейных дифференциальных и линейных алгебраических уравнений:

$$\begin{aligned} \Psi_{L0}(\dot{U}, J, U(t), J(t)) &= W_L(t, U(t), J(t)); \\ \Psi_{LT}(U(t), J(t)) &= W_L(t); \\ \Psi_L(U, J) &= W_L. \end{aligned} \quad (1.3)$$

Модель нелинейного  $n$ -полюсника класса  $M_N$  описывается системой из нелинейных и линейных алгебраических уравнений:

$$\begin{aligned} \Psi_N(U, J) &= W_N(U, J); \\ \Psi_L(U, J) &= W_L. \end{aligned} \quad (1.4)$$

Уравнения модели класса  $M_{NT}$  имеют вид:

$$\begin{aligned} \Psi_{NT}(U(t), J(t)) &= W_N(t, U(t), J(t)); \\ \Psi_N(U, J) &= W_N(U, J); \\ \Psi_L(U, J) &= W_L. \end{aligned} \quad (1.5)$$

Модель класса  $M_{N0}$  описывается системой линейных относительно производных дифференциальных уравнения плюс уравнения моделей всех остальных классов:

$$\begin{aligned} \Psi_{N0}(\dot{U}, J, U(t), J(t)) &= W_N(t, U(t), J(t)); \\ \Psi_{NT}(U(t), J(t)) &= W_N(t, U(t), J(t)); \\ \Psi_N(U, J) &= W_N(U, J); \\ \Psi_{LT}(U(t), J(t)) &= W_L(t); \\ \Psi_{L0}(\dot{U}, J, U(t), J(t)) &= W_L(t, U(t), J(t)); \\ \Psi_L(U, J) &= W_L. \end{aligned} \quad (1.6)$$



В уравнениях (1.1) — (1.6)  $\Psi$  — линейная форма от  $U$  и  $J$  или их производных,  $W_L$  и  $W_N$  — соответственно линейная и нелинейная вектор-функции от токов и напряжений. Иерархическая взаимосвязь моделей приведена на рис. 2.

В заключение данного раздела отметим, что предлагаемый способ представления математических моделей компонентов допускает ввод моделей элементов с произвольной внутренней структурой и произвольным числом

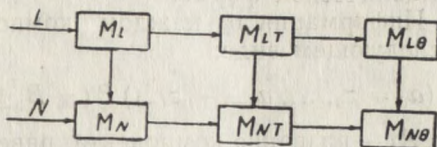


Рис. 2

выводов. Уравнения таких моделей могут быть получены чисто эмпирически, путем аппроксимации характеристик, снятых с реальных приборов.

## § 2. Построение модели цепи

Для удобства формального описания моделей цепей введем понятие компонентных подцепей и матриц. Компонентной подцепью типа  $A$  (обозначим  $P_A$  — подцепь) будем называть часть цепи, содержащую только компоненты типа  $A$ . Так как любая компонентная подцепь является в то же время и многополюсником, то приведенная выше классификация для моделей компонентов применима и к компонентным подцепям.

Таким образом, мы можем зафиксировать шесть типов подцепей:  $P_L$ ,  $P_{LT}$ ,  $P_{L\theta}$ ,  $P_N$ ,  $P_{NT}$ ,  $P_{N\theta}$ . Матрицы, образованные линейными формами в левых частях уравнений подцепи, будем называть компонентными матрицами и обозначать соответственно символами  $\Psi_L$ ,  $\Psi_{LT}$ ,  $\Psi_{L\theta}$ ,  $\Psi_N$ ,  $\Psi_{NT}$ ,  $\Psi_{N\theta}$ . Число строк каждой из этих матриц равно общему числу уравнений, из которых состоят входящие в компонентную подцепь модели, а число столбцов равно  $(k+m)$ , где  $k$  — число узлов, а  $m$  — число ветвей цепи.

Теперь любую цепь можно определить как некоторую совокупность компонентных подцепей. Состав цепи определяется набором входящих в нее компонентных подцепей. Класс цепи определяется многополюсником, который является наиболее сложным с точки зрения иерархической взаимосвязи моделей (рис. 2). Очевидно, что классификация цепей также не отличается от классификации многополюсников, поскольку

любую цепь мы можем трактовать как автономный многополюсник. Для построения модели цепи требуется исходная информация, которая содержит принципиальную схему, заданную перечнем элементов (многополюсников), способ их соединения и параметры элементов, заданные в численной или символической форме.

Информация о каждом компоненте цепи записывается в следующем виде:

$$A_i(a_0 = z_0, \dots, a_{l-1} = z_{l-1}) B(\pm B_1 \pm B_2 \dots \pm B_m) N(n_1 n_2 \dots n_\kappa),$$

где  $A_i$  — название компонента, равенства  $a_i = z_i$  ( $i = 0, \overline{l-1}$ ) задают численные или символические значения его внутренних параметров,  $(\pm b_i)$ , ( $i = 1, m$ ) — номера выводов элемента. Предполагается, что  $(+b)$  фиксирует ток, втекающий в узел,  $(-b)$  — ток, вытекающий из узла.  $N_i$  ( $i = 1, \kappa$ ) — номера узлов, с которыми соединен каждый вывод. Узлы перечисляются в том же порядке, что и выводы.

Методика построения модели цепи весьма проста и состоит из следующих этапов:

1. Преобразуются модели элементов цепи путем ввода в них вместо токов и напряжений на выводах токов и напряжений в цепи (индекс у символа  $I$  совпадает с номером ветви, у символа  $U$  — с номером узла). Преобразование осуществляется простой заменой индексов в соответствующих уравнениях. Параметры модели заменяются их численными или символическими значениями, заданными в исходной информации.

2. Для каждого узла, кроме нулевого, расписывается закон Кирхгофа для токов. Вводится уравнение  $U_0 = 0$ .

3. Производится переход к единой переменной  $U_0 = V_0, U_1 = V_1, \dots, U_{\kappa-1} = V_{\kappa-1}, I_0 = V_\kappa, \dots, I_{m-1} = V_{\kappa+m-1}$ . Из полученной системы уравнений исключаются все уравнения вида  $V_i = V_j$ . Тем самым количество уравнений значительно сокращается. В моделях двухполюсников остается одно уравнение.

Дальнейшая обработка зависит от класса цепи.

## 2.1. Линейные цепи постоянного тока

Линейные цепи постоянного тока содержат в себе только подцепи  $P_L$ -типа.  $P_L$ -цепь характеризуется матрицей  $\Psi_L$  размерности  $\{m_L \times n\}$ , где  $m_L$  — число ветвей в цепи, а  $n = k + m$ . Закон Кирхгофа в матричной форме имеет вид

$$\Psi_L \cdot V = 0,$$



где  $\Psi_I$  — матрица, образованная из нулевых и единичных коэффициентов при соответствующих токах ветвей.

Объединяя матричные уравнения  $P_L$ -подцепи и закона Кирхгофа, получим модель линейной цепи постоянного тока в виде системы линейных алгебраических уравнений:

$$\begin{bmatrix} \Psi_L \\ \Psi_J \end{bmatrix} V = \begin{bmatrix} W_L \\ 0 \end{bmatrix}, \quad (1.7)$$

где  $W_L$  — вектор задающих источников, который включает в себя и правые части уравнений остальных элементов. Обращением к соответствующему оператору Авто-Аналитика система (1.7) решается. Тем самым все токи и напряжения в цепи выражаются явным образом.

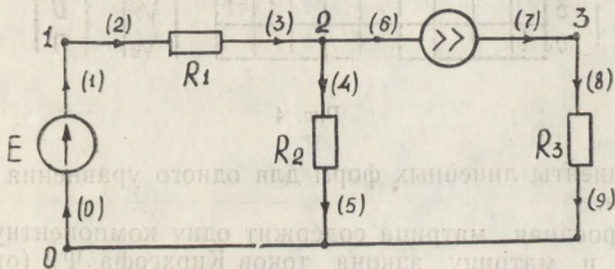


Рис. 3

Рассмотрим пример моделирования линейной цепи постоянного тока (рис. 3). Цепь состоит из линейных безынерционных двухполюсников (источников тока, напряжения и резисторов), каждый вывод у которых пронумерован. Исходная информация к схеме имеет следующий вид:

|               |           |          |
|---------------|-----------|----------|
| $E (E=10)$    | $B(-0,1)$ | $N(0,1)$ |
| $R_1 (R_1=1)$ | $B(-2,3)$ | $N(1,2)$ |
| $R_2 (R_2=2)$ | $B(-4,5)$ | $N(2,0)$ |
| $I (I=1)$     | $B(-6,7)$ | $N(2,3)$ |
| $R_3 (R_3=3)$ | $B(-8,9)$ | $N(3,0)$ |

Модель данной цепи, соответствующая уравнению (1.7), имеет вид (рис. 4).

После замены индексов у переменных в моделях построение компонентной матрицы сводится к выписыванию коэффи-

коэффициентов моделей элементов при соответствующих переменных цепи. Номер переменной цепи соответствует номеру столбца, а каждая строка компонентной матрицы содержит в себе

$$\begin{array}{c}
 \begin{array}{c} \Psi_L \\ \Psi_I \end{array} \left\{ \begin{array}{c} E \\ I \\ R_1 \\ R_2 \\ R_3 \\ 1 \\ 2 \\ 3 \\ 0 \end{array} \right. \begin{array}{c} 0 \quad 1 \quad 2 \quad 3 \quad 0 \quad 2 \quad 4 \quad 6 \quad 8 \\ \begin{array}{|c|c|c|c|c|c|c|c|c|} \hline 1 & -1 & & & & & & & \\ \hline & & & & & & -1 & & \\ \hline & -1 & 1 & & & -R_1 & & & \\ \hline & & & & & -R_2 & & & \\ \hline & & & & & & & -R_3 & \\ \hline & & & 1 & -1 & & & & \\ \hline & & & & 1 & -1 & -1 & & \\ \hline & & & & & & & 1 & -1 \\ \hline 1 & & & & & & & & \\ \hline \end{array} \end{array} \cdot x = \begin{array}{c} V_0 \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \\ V_7 \\ V_8 \end{array} = \begin{array}{c} E \\ \mathcal{I} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array}
 \end{array}$$

Рис. 4

коэффициенты линейных форм для одного уравнения модели элемента.

Построенная матрица содержит одну компонентную матрицу  $\Psi_L$  и матрицу закона токов Кирхгофа  $\Psi_I$  (отделены пунктиром).

В полученной модели  $V_0=U_0$ ;  $V_1=U_1$ ;  $V_2=U_2$ ;  $V_3=U_3$ ;  $V_4=I_0$ ;  $V_5=I_2$ ;  $V_6=I_4$ ;  $V_7=I_6$ ;  $V_8=I_8$ . Токи  $I_1$ ,  $I_3$ ,  $I_5$ ,  $I_7$  исключены.

## 2.2. Линейная безынерционная цепь переменного тока

Данные цепи содержат в себе компонентные подцепи  $P_L$   $P_{LT}$ -типов. Модель такой цепи имеет вид

$$\begin{array}{c} \Psi_L \\ \Psi_{LT} \\ \Psi_I \end{array} V = \begin{array}{c} W_L \\ W_{LT} \\ 0 \end{array}. \quad (1.8)$$

Здесь  $W_{LT}$  — вектор, включающий в себя задающие источники, зависящие от времени, и правые части остальных уравнений  $P_{LT}$ -подцепи. Примером данной цепи может служить схема на рис. 5. Предполагается, что емкости конденсаторов  $C_1$ ,  $C_2$  выбраны настолько большими, что их влиянием на параметры усилителя на средних частотах можно пренебречь.



Кроме того, влияние собственных емкостей транзистора в среднем диапазоне частот также несущественно, поэтому для транзистора может быть выбрана модель класса  $M_{LT}$ .

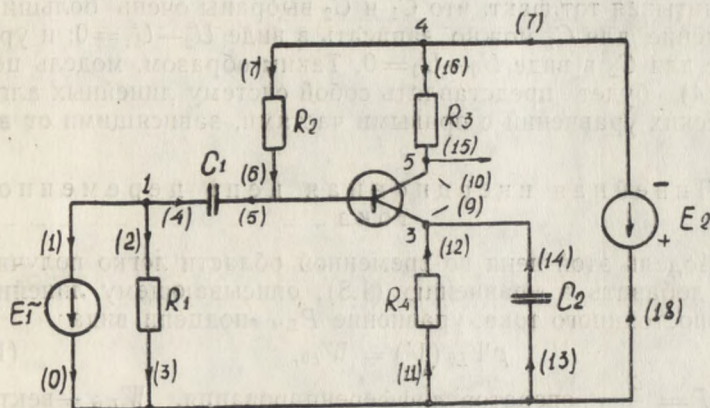


Рис. 5

Выбор линейной модели обусловлен малой амплитудой входного сигнала.

С учетом изложенного запишем уравнение модели цепи. Ввиду большой разреженности компонентных матриц все приводимые ниже модели будут представлены в виде систем уравнений с линейными формами относительно переменных цепи в левых частях:

$$\begin{aligned}
 U_1 - U_0 &= E_1; & U_0 &= 0; \\
 U_1 - U_0 - J_2 R_1 &= 0; & J_4 - J_2 - J_0 &= 0; \\
 \frac{d(U_2 - U_1)}{dt} &= \frac{1}{C_1} J_4; & J_8 - J_6 - J_4 &= 0; \\
 J_9 + J_{10} - J_8 &= 0; & J_{12} + J_{14} - J_9 &= 0; \\
 U_3 - U_2 - J_9 \gamma_3 - J_8 \gamma_6 &= 0; & J_{18} - J_{16} - J_6 &= 0; \\
 -U_3 + U_5 - \alpha \gamma_k J_9 - J_8 \gamma_6 - J_{10} \gamma_k &= 0; & J_{16} - J_{10} &= 0; \\
 U_4 - U_2 - J_8 R_2 &= 0; \\
 U_4 - U_5 - J_{16} R_3 &= 0; \\
 U_0 - U_3 - J_{12} R_4 &= 0;
 \end{aligned}$$

$$\frac{d(U_0 - U_3)}{dt} = \frac{1}{C_2} J_{14};$$

$$U_0 - U_4 = E_2.$$

Учитывая тот факт, что  $C_1$  и  $C_2$  выбраны очень большими, уравнение для  $C_1$  можно записать в виде  $U_2 - U_1 = 0$ ; и уравнение для  $C_2$  в виде  $U_0 - U_3 = 0$ . Таким образом, модель цепи (рис. 4) будет представлять собой систему линейных алгебраических уравнений с правыми частями, зависящими от времени.

### 2.3. Линейная инерционная цепь переменного тока

Модель этой цепи во временной области легко получить, если добавить к уравнению (1.8), описывающему линейную цепь постоянного тока, уравнение  $P_{L\theta}$ -подцепи вида

$$p\Psi_{L\theta}(V) = W_{L\theta}, \quad (1.9)$$

где  $P = \frac{d}{dt}$  — оператор дифференцирования,  $W_{L\theta}$  — вектор-функция из правых частей уравнений  $P_{L\theta}$ -подцепи. Поскольку для линейных цепей допустимо операторное изображение, уравнение (1.9) может быть записано и в другой форме

$$\Psi_{L\theta}(p) \cdot V = W_{L\theta}(p). \quad (1.10)$$

Объединяя уравнения (1.9) и (1.10) с уравнением (1.8), получим модели линейной инерционной цепи соответственно во временной (1.11a) и операторной форме (1.11б).

$$\begin{bmatrix} p\Psi_{L\theta} \\ \Psi_L \\ \Psi_{LT} \\ \Psi_J \end{bmatrix} \cdot V = \begin{bmatrix} W_{L\theta} \\ W_L \\ W_{LT} \\ 0 \end{bmatrix}; \quad (1.11a)$$

$$\begin{bmatrix} \Psi_{L\theta}(p) \\ \Psi_L \\ \Psi_{LT} \\ \Psi_J \end{bmatrix} \cdot V = \begin{bmatrix} W_{L\theta}(p) \\ 0 \\ W_{LT}(p) \\ 0 \end{bmatrix}. \quad (1.11б)$$

Модель, соответствующая уравнению (1.11a), обрабатывается следующим образом. После решения подсистемы линейных алгебраических уравнений ( $P_L$ ,  $P_{LT}$ -подцепи) и подстановки решения в прочие уравнения получается система дифференциальных уравнений с некоторыми линейными фор-



мами относительно  $V_i$  под производными. Производим дифференцирование и решение системы линейных относительно производных алгебраических уравнений. В результате получается система дифференциальных уравнений в нормальном виде. Размерность этой системы  $P \leq \kappa + m - l$ , где  $l$  — число линейных алгебраических уравнений. Знак неравенства появляется вследствие возможных вырождений дифференциальных уравнений в алгебраические, что связано

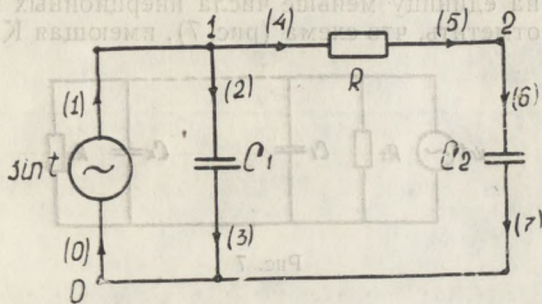


Рис. 6

с появлением под знаком производных известных решений. Рассмотрим случай вырождения на простом примере. Модель схемы (рис. 6) имеет следующий вид:

$$\begin{aligned} \frac{dU_1}{dt} &= \frac{1}{C_1} J_2; & U_0 &= 0; \\ \frac{dU_2}{dt} &= \frac{1}{C_2} J_6; & J_0 - J_2 - J_4 &= 0; \\ & & J_4 - J_6 &= 0. \\ U_0 - U_1 &= \sin t; \\ U_1 - U_2 - RJ_4 &= 0; \end{aligned}$$

Для приведения дифференциальных уравнений к нормальному виду решим систему линейных алгебраических уравнений относительно  $U_1$  и  $J_6$ :

$$\begin{aligned} U_1 &= -\sin t; \\ J_6 &= J_4 = \frac{U_1 - U_2}{R}. \end{aligned}$$

Подставляя найденные значения в дифференциальные уравнения, получим:

$$\frac{dU_1}{dt} = \cos t = \frac{1}{C_1} J_2;$$

$$\frac{dU_2}{dt} = -\frac{\sin t}{RC_2} - \frac{U_2}{RC_2}.$$

Из этих двух уравнений дифференциальным является только второе.

Таким образом, для данного примера порядок системы уравнений на единицу меньше числа инерционных элементов. Интересно отметить, что схема (рис. 7), имеющая  $K$  инерцион-

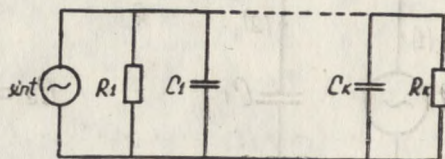


Рис. 7

ных элементов, описывается системой алгебраических уравнений.

Перейдем теперь к рассмотрению линейной цепи переменного тока в частотной форме. Для этого представим переменные цепи в виде  $U = U' + jU''$ ,  $I = I' + jI''$ , где  $U'$  и  $I'$  — действительные;  $U''$  и  $I''$  мнимые составляющие переменных цепи.

Оператор дифференцирования  $P = \frac{d}{dt}$ .

Далее получим частотную модель для  $P_{L\theta}$ -подцепи. Имея в виду, что данная подцепь содержит индуктивности и емкости, запишем их модели в частотной форме:

$$\left( \frac{1}{\omega L} U' + J'' \right) + j \left( \frac{1}{\omega L} U'' - J' \right) = 0;$$

$$\left( \frac{1}{\omega C} J' + U'' \right) + j \left( \frac{1}{\omega C} J'' - U' \right) = 0.$$

После разделения мнимой и действительной части получим модель  $P_{L\theta}$ -цепи в матричной форме:

$$\begin{vmatrix} W_L & 0 \\ 0 & W_C \end{vmatrix} \begin{vmatrix} U' \\ J' \end{vmatrix} + \begin{vmatrix} 0 & \Gamma_L \\ \Pi_\theta & 0 \end{vmatrix} \begin{vmatrix} U'' \\ J'' \end{vmatrix} = 0,$$



$$\begin{vmatrix} \Pi_L & 0 \\ 0 & \Gamma_c \end{vmatrix} \times \begin{vmatrix} U' \\ J' \end{vmatrix} - \begin{vmatrix} W_L & 0 \\ 0 & W_c \end{vmatrix} \begin{vmatrix} U'' \\ J'' \end{vmatrix} = 0,$$

или в более компактном виде

$$\begin{vmatrix} W_{L0} & \Psi_{L0} \\ \Psi_{L0} & -W_{L0} \end{vmatrix} \begin{vmatrix} V' \\ V'' \end{vmatrix} = 0,$$

где  $V' = \begin{vmatrix} U' \\ J' \end{vmatrix}$ , а  $V'' = \begin{vmatrix} U'' \\ J'' \end{vmatrix}$ .

Модели  $P_{LT}$ ,  $P_L$ -подцепей и закона Кирхгофа в частотной форме также строятся путем разделения переменных. Окончательно модель цепи переменного тока в частотной форме имеет вид:

$$\begin{vmatrix} \Psi_{L0} & W_{L0} \\ \Psi_{LT} & 0 \\ \Psi_L & 0 \\ \Psi_J & 0 \\ -W_{L0} & \Psi_{L0} \\ 0 & \Psi_{LT} \\ 0 & \Psi_L \\ 0 & \Psi_J \end{vmatrix} \times \begin{vmatrix} V' \\ V'' \end{vmatrix} = \begin{vmatrix} 0 \\ W_{LT} \\ 0 \\ 0 \\ 0 \\ W_{LT}'' \\ 0 \\ 0 \end{vmatrix}. \quad (1.12)$$

Рассмотрим пример моделирования цепи (рис. 8) в частот-

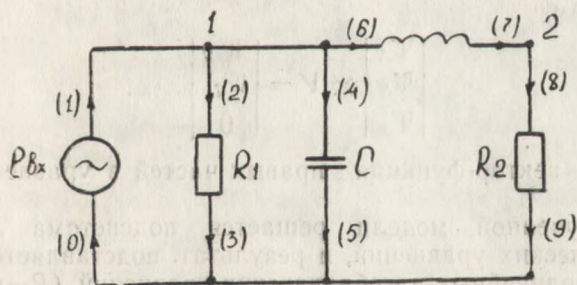


Рис. 8

ной области. Модель этой цепи в матричной форме будет иметь следующий вид (рис. 9):

|             | 1' | 2' | 0' | 2' | 4'               | 6' | 8'     | 1''    | 2'' | 0'' | 2'' | 4''              | 6'' | 8'' |                   |             |
|-------------|----|----|----|----|------------------|----|--------|--------|-----|-----|-----|------------------|-----|-----|-------------------|-------------|
| $\Psi_{L6}$ | +1 | -1 |    |    |                  |    |        |        |     |     |     |                  |     |     | $U_1'$            | 0           |
|             |    |    |    |    | $\frac{1}{2}w_0$ |    |        | -1     |     |     |     |                  |     |     | $U_2'$            | 0           |
| $\Psi_{L7}$ | -1 |    |    |    |                  |    |        |        |     |     |     |                  |     |     | $\mathcal{I}_0'$  | $\varrho^L$ |
| $\Psi_L$    | +1 |    |    |    | $-R_1$           |    |        |        |     |     |     |                  |     |     | $\mathcal{I}_2'$  | 0           |
|             |    | +1 |    |    |                  |    |        | $-R_2$ |     |     |     |                  |     |     | $\mathcal{I}_4'$  | 0           |
| $\Psi_y$    |    |    | 1  | -1 | -1               | -1 |        |        |     |     |     |                  |     |     | $\mathcal{I}_6'$  | 0           |
|             |    |    |    |    |                  | 1  | -1     |        |     |     |     |                  |     |     | $\mathcal{I}_8'$  | 0           |
|             |    |    |    |    |                  |    | $-w_0$ | +1     | -1  |     |     |                  |     |     | $U_1''$           | 0           |
|             | 1  |    |    |    |                  |    |        |        |     |     |     | $\frac{1}{2}w_0$ |     |     | $U_2''$           | 0           |
|             |    |    |    |    |                  |    |        | -1     |     |     |     |                  |     |     | $\mathcal{I}_0''$ | $\varrho^L$ |
|             |    |    |    |    |                  |    |        | +1     |     |     |     | $-R_1$           |     |     | $\mathcal{I}_2''$ | 0           |
|             |    |    |    |    |                  |    |        |        | +1  |     |     |                  |     |     | $\mathcal{I}_4''$ | 0           |
|             |    |    |    |    |                  |    |        |        |     | 1   | -1  | -1               | -1  |     | $\mathcal{I}_6''$ | 0           |
|             |    |    |    |    |                  |    |        |        |     |     |     |                  | 1   | -1  | $\mathcal{I}_8''$ | 0           |

x

Рис. 9

## 2.4. Нелинейная цепь постоянного тока

Общего вида нелинейная цепь постоянного тока содержит в себе подцепи типа  $P_L$  и  $P_N$ . Модель такой цепи описывается уравнением

$$\begin{bmatrix} \Psi_N \\ \Psi_L \\ \Psi_J \end{bmatrix} \times V = \begin{bmatrix} W_N \\ W_L \\ 0 \end{bmatrix}, \quad (1.13)$$

где  $W_N$  — вектор-функция правых частей в уравнениях  $P_N$ -подцепи.

В полученной модели решается подсистема линейных алгебраических уравнений, и результат подставляется в подсистему нелинейных алгебраических уравнений ( $P_N$ -подцепь). В системе нелинейных алгебраических уравнений приводятся подобные.



Необходимо отметить, что и здесь возможно сокращение размерности подсистемы нелинейных алгебраических уравнений за счет появления известных решений в векторе правых частей  $W_N$ .

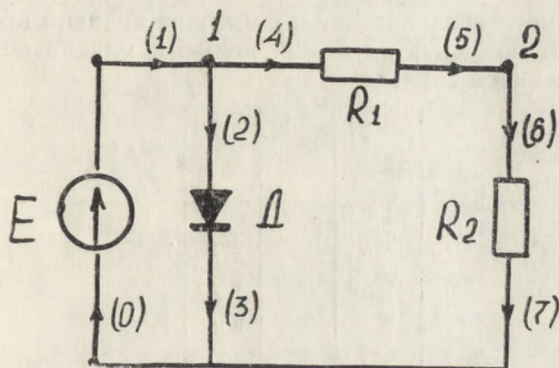


Рис. 10

Рассмотрим пример моделирования нелинейной цепи постоянного тока (рис. 10). Модель этой схемы описывается следующими уравнениями:

$$\begin{aligned} J_2 &= K_1 (I^{K_2} U_1 - 1); & U_0 &= 0; \\ U_1 - U_2 - R_1 J_2 &= 0; & J_0 - J_2 - J_4 &= 0; \\ U_2 - R_2 J_3 &= 0; & J_4 - J_6 &= 0. \\ U_1 &= E; \end{aligned}$$

После подстановки  $U_1 = E$  в уравнение диода оно становится линейным, и модель схемы описывается системой линейных алгебраических уравнений.

## 2.5. Модель нелинейной безынерционной цепи переменного тока

К данному классу относятся некоторые цепи большого сигнала, работающие в среднем диапазоне частот. От предыдущего класса их отличает появление подцепи типа  $P_{NT}$ , уравнение которой имеет вид:

$$\Psi_{NT} \cdot V = W_{NT}. \quad (1.14)$$

Объединяя его с уравнением (1.13), получим модель нелинейной безынерционной цепи.

## 2.6. Модель нелинейной цепи переменного тока

Нелинейные цепи переменного тока являются самыми сложными по своему составу и содержат подцепи всех типов. Комбинируя модели этих цепей, получим модель нелинейной цепи переменного тока.

$$V = \begin{pmatrix} p\Psi_{N0} \\ p\Psi_{L0} \\ \Psi_{NT} \\ \Psi_N \\ \Psi_{LT} \\ \Psi_L \\ \Psi_J \end{pmatrix} = \begin{pmatrix} W_{N0} \\ W_{L0} \\ W_N \\ W_{L0} \\ W_{LT} \\ W_L \\ 0 \end{pmatrix}. \quad (1.15)$$

Если в исследуемой цепи отсутствует  $P_N$ -подцепь, то дальнейшее моделирование производится так же, как и для линейной цепи переменного тока. В противном случае выдается смешанная система из нелинейных алгебраических и дифференциальных уравнений. Процесс моделирования заканчивается.

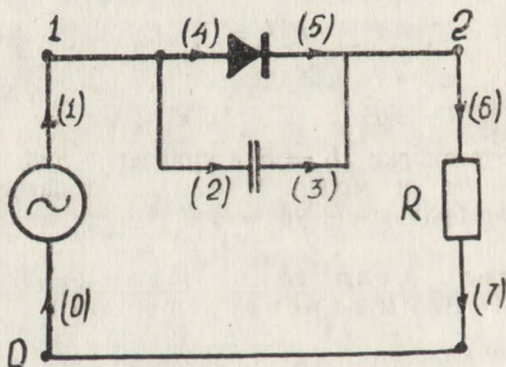


Рис. 11

Рассмотрим пример моделирования нелинейной цепи переменного тока (рис. 11). Уравнения модели имеют следующий



вид:

$$\begin{aligned} \frac{d}{dt}(U_1 - U_2) &= \frac{1}{C} J_2; & U_0 &= 0; \\ J_4 &= K_1 (e^{K_2 (U_1 - U_2)} - 1); & J_0 - J_2 - J_4 &= 0; \\ U_0 - U_1 &= E; & J_2 + J_4 - J_6 &= 0. \\ U_2 - U_0 - R J_6 &= 0; \end{aligned}$$

Если два элемента — диод и емкость рассматривать как один двухполюсник (рис. 11), то модель цепи будет содержать

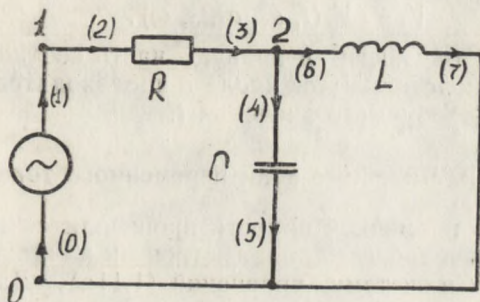


Рис. 12

одно дифференциальное уравнение с нелинейной правой частью

$$\frac{d}{dt}(U_1 - U_2) = \frac{U_2}{RC} - \frac{K_1}{C} (e^{K_2 (U_1 - U_2)} - 1).$$

Нелинейные алгебраические уравнения в этом случае отсутствуют.

## II. АНАЛИЗ ЦЕПЕЙ

Задачей анализа является получение решения цепи, то есть вектора переменных  $V$ , удовлетворяющего моделям компонентов и законам Кирхгофа. Как и при моделировании, вид анализа определяется классом исследуемой цепи и задачей, поставленной разработчиком.

## § 1. Линейные цепи постоянного тока

В результате моделирования получилась система уравнений (1.7), которая решается затем символически и получается решение в виде:

$$V_i = V_i(a_0, a_1, \dots, a_l), \quad \overline{i} = 1, n. \quad (2.1)$$

Здесь  $a_0, a_1, \dots, a_l$  — внутренние параметры цепи. Если все эти параметры задавались в числовом виде, то и  $V_i$  примет числовые значения.

При анализе линейных безынерционных цепей переменного тока решение имеет вид:

$$V_i = V_i(t, a_0, a_1, \dots, a_l). \quad (2.2)$$

При необходимости в правую часть соотношения (2.2) вместо  $t$  и  $a_i$  подставляются числа и производятся вычисления для каждого требуемого значения  $t$ .

## § 2. Линейные цепи переменного тока

Анализ во временной области производится подстановкой численных значений всех параметров и начальных данных для вектора  $V$  в систему уравнений (1.11а). Далее оператором численного интегрирования производится построение решения на заданном промежутке с нужной точностью.

Для анализа в частотной области может быть предложено два варианта.

Рассмотрим систему уравнений (1.12). Если анализ производится для какой-то конкретной частоты,  $i$ -я компонента вектора переменных имеет вид:

$$V'_i = V'_i(a_1, a_2, \dots, a_l, \omega_k); \quad (2.3)$$

$$V''_i = V''_i(a_1, a_2, \dots, a_l, \omega_k).$$

Анализ осуществляется решением системы линейных алгебраических уравнений. Амплитуда и фаза искомой переменной определяется по известным формулам:

$$A_{V_i} = \sqrt{(V'_i)^2 + (V''_i)^2};$$

$$Y_{V_i} = \arctg \frac{V''_i}{V'_i} + Y_0, \quad (2.4)$$

где  $Y_0$  — начальная фаза сигнала.



В том случае, когда необходимо построение частотной характеристики, вместо  $a_i$  и  $\omega_i$  подставляются числа, и далее производятся вычисления по формулам (2.3) и (2.4) для каждой частоты.

Частотные характеристики цепи могут быть легко найдены с помощью передаточной функции. Для этого перепишем уравнение (1.116) в следующем виде:

$$\Psi(p) V_{\text{вых}} = W_{\text{вх}}.$$

Решая это уравнение относительно  $V_{\text{вых}}$ , получим

$$V_{\text{вых}} = \Psi^{-1}(p) W_{\text{вх}}.$$

Назовем  $\Psi^{-1}(p)$  матрицей передаточных функций. Из этой матрицы может быть получена передаточная функция цепи относительно любого ее входа и выхода, а также любой набор передаточных функций относительно всех интересующих выходов. Если необходимо получить передаточную функцию от  $i$ -го входа к  $k$ -му выходу цепи, то удобно воспользоваться правилом Крамера, при этом решение уравнения (1.116) имеет вид

$$V_{\text{вых } k} = \frac{\Delta \Psi_k(p)}{\Delta \Psi(p)} W_{\text{вх } i}.$$

Здесь  $\Delta \Psi_k(p)$  — определитель матрицы  $\Psi(p)$ , в котором  $k$ -й столбец заменен столбцом свободных членов.

Отношение  $\frac{\Delta \Psi_k(p)}{\Delta \Psi(p)}$  и является искомой передаточной функцией цепи (обозначим ее  $K(p)$ ). Для нахождения частотных характеристик в  $K(p)$  вместо  $p$  подставим  $j\omega$ . Выделяя мнимую и действительную составляющие передаточной функции, находим  $K'(\omega)$  и  $K''(\omega)$ . Далее эти значения подставляются в выражения (2.4) вместо  $V_i$  и  $V_i^*$ , из которых сразу получаются амплитуда и фаза передаточной функции.

### § 3. Нелинейные цепи постоянного тока

Одним из операторов оптимизации [5] или методом Ньютона производится поиск решения системы нелинейных алгебраических уравнений до выполнения условий достижения требуемой точности.

Анализ нелинейной безынерционной цепи переменного тока заключается в решении модели этой цепи, содержащей уравнения (1.13) и (1.14), для каждого требуемого значения  $t$ .

#### § 4. Нелинейные цепи переменного тока

Анализ этих цепей аналогичен анализу линейных цепей в том случае, когда в составе цепи отсутствует  $P_N$ -подцепь. В противном случае при моделировании появляется смещенная система дифференциальных и нелинейных алгебраических уравнений, решение которой связано с чрезмерными затратами машинного времени. Чтобы обойти это препятствие, модели нелинейных компонентов дифференцируются по времени, в результате чего образуется система дифференциальных уравнений с нелинейными правыми частями, решение которой аналогично решению системы (1.11 а).

Отметим, что при анализе динамического режима приходится предварительно определять начальные условия в цепи, для этого в системе (1.15) заменяются уравнения  $P_{N\theta}$  - и  $P_{L\theta}$ -подцепей уравнениями для начальных условий:

$$0 = W_{N\theta};$$

$$0 = W_{L\theta}.$$

Далее решается модель нелинейной цепи постоянного тока, из которой и определяются начальные данные для интегрирования.

#### § 5. Анализ допусков

Анализ допусков линейных цепей в методе символического анализа выполняется по следующим простым формулам. Вектор переменных линейной цепи находится из соотношения вида:

$$V = \Psi^{-1}(a) \cdot W, \quad (2.5)$$

где  $a$  — значения входящих в матрицу  $\Psi$  параметров. Полагая, что  $\Delta a$  — допускá на эти параметры, а  $\Delta W$  — допускá на параметры источников, получим:

$$V + \Delta V = W^{-1}(a + \Delta a) (W + \Delta W). \quad (2.6)$$

Тогда простым вычитанием (2.5) из (2.6) получается выражение для допуска в векторе переменных

$$\Delta V = [\Psi^{-1}(a + \Delta a) - \Psi^{-1}(a)] \cdot W + \Psi^{-1}(a + \Delta a) \Delta W. \quad (2.7)$$

Выражение (2.7) удобно для статистического анализа цепей при больших приращениях  $\Delta a$  и  $\Delta W$ . При использовании метода статистических испытаний его необходимо получить только один раз, далее в правую часть уравнения (2.7) подставляются числа, и вычисления производятся значительно бы-



стрее, нежели обращение матрицы на каждом шаге метода статистических испытаний, как это имеет место, например, в методе узловых потенциалов.

Для малых приращений  $\Delta a_i$  приращение  $\Delta V$  вычисляется в окрестности точки  $V_0$  по известной формуле

$$\Delta V = \sum_{i=1}^U \frac{dV}{da_i} \bigg|_{V=V_i} \Delta a_i$$

в которой частные производные определяются с помощью символического дифференцирования.

Статистическое исследование нелинейных цепей в символическом анализе также обладает перед численными методами анализа тем преимуществом, что построение системы уравнений, описывающих нелинейную цепь, производится только один раз. Дальнейшие расчеты связаны с простой подстановкой случайных значений параметров компонентов в полученные уравнения и вычислениями по формулам.

### Заключение

В данной статье дано общее описание методики символического моделирования и анализа электронных схем.

Предпринята попытка системного изложения вопроса использования преимуществ аналитических выкладок на ЭВМ для целей машинного проектирования.

Предлагаемые выше алгоритмы сочетают простоту структуры с универсальностью применения и могут быть использованы в других системах машинного проектирования, основанных на символическом представлении параметров цепей. Выбор в качестве базовой системы Авто-Аналитик позволил с минимальными затратами на программирование и отладку построить развитый комплекс программ машинного проектирования схем.

Дальнейшее развитие предлагаемого метода будет производиться в области символического анализа нелинейных цепей. Решение этой проблемы позволит отказаться от цикличности анализа при оптимизации и статистическом исследовании цепей и значительно расширит рамки практического применения системы.

### ЛИТЕРАТУРА

1. Калахан Д. Методы машинного расчета электронных схем. М., «Мир», 1970.

2. Ильин В. Н. Машинное проектирование электронных схем. М., «Энергия», 1972.
  3. Трохименко Я. К. Метод обобщенных чисел и анализ линейных цепей. М., «Советское радио», 1972.
  4. Арайс Е. А., Сибиряков Г. В. Основы системы программирования Авто-Аналитик.— Вопросы программирования и автоматизации проектирования, вып. 1. Изд-во Томского университета, 1971.
  5. Арайс Е. А., Дмитриев В. М. Многопараметрическая оптимизация в машинном проектировании цепей. Там же.
  6. Арайс Е. А. Символический анализ радиоэлектронных схем в системе Авто-Аналитик.— Вопросы программирования и автоматизации проектирования, вып. 2. Изд-во Томского университета, 1972.
-



## АНАЛИЗ НЕЛИНЕЙНЫХ ЦЕПЕЙ В СИСТЕМЕ МАРС

Е. А. АРАЙС

Одним из способов решения проблемы разброса постоянных времени является использование схем неявного интегрирования. В настоящей работе рассматривается организация анализа нелинейных цепей в системе автоматизированного проектирования радиоэлектронных схем МАРС на основе модельного метода [1]. Отличительной особенностью предлагаемого подхода является точное (аналитическое) вычисление частных производных матрицы Якоби, что позволяет повысить точность расчетов и быстродействие. Допускается также включение в исследуемую цепь произвольных многополюсников (без построения и ввода в ЭВМ эквивалентных схем, как это принято в большинстве существующих программ).

### § 1. Общая схема алгоритма

Составными частями алгоритма анализа нелинейных цепей являются метод интегрирования системы дифференциальных уравнений и метод решения нелинейных алгебраических уравнений. В рамках комплекса программ численного анализа цепей МАРС используется неявная формула Эйлера для интегрирования и одношаговый метод Ньютона для решения алгебраических уравнений. Необходимая точность обеспечивается выбором шага по времени. Распишем работу алгоритма анализа по шагам.

Шаг 1. Путем опроса моделей компонентов [2] и анализа топологии цепи строим матрицу  $\Phi$  и вектор  $W$ , которые задают систему уравнений:

$$\Phi \times V = W. \quad (1)$$

Здесь  $V = (U_0, U_1, \dots, U_{n-1}, I_0, I_1, \dots, I_{m-1})$  — решение цепи, имеющей  $n$  узлов и  $m$  ветвей.  $\Phi$  — прямоугольная матрица, построенная из коэффициентов линейных алгебраических уравнений моделей и законов Кирхгофа для токов.  $W$  — вектор правых частей.

Ясно, что система линейных алгебраических уравнений (1) является недоопределенной.

Шаг 2. Определяем автоматически либо строим вручную вектор  $I = (i_1, i_2, \dots, i_k)$ , задающий индексы нелинейных переменных при сплошной нумерации  $V = (V_0, V_1, \dots, V_{m+n-1})$ . Под нелинейными будем понимать такую совокупность переменных, которая вместе с системой (1) полностью определяет решение  $V$  исследуемой цепи. В этом случае выбор  $I$  не может быть произвольным, так как для разрешимости объединенной системы уравнений

$$\begin{cases} V_1 = C, \\ \Phi \times V = W, \end{cases} \quad (2)$$

где  $V_1$  — вектор нелинейных переменных;  $C$  — вектор их значений; требуется невырожденность матрицы из коэффициентов левых частей системы (2).

Приведем физическую интерпретацию способа выбора вектора индексов  $I$ . Будем считать, что в (2) нелинейная переменная (напряжение или ток) задает компонент цепи, являющийся источником соответственно напряжения или тока, заменяя реальный компонент. Источники напряжения или тока могут определяться также и уравнениями (1). Предъявляя обычные требования отсутствия в цепи колец из источников напряжения и звезд из источников тока, получаем практический критерий для построения вектора  $I$  индексов нелинейных переменных.

С математической точки зрения требуется невырожденность матрицы, полученной из  $\Phi$  после вычеркивания всех столбцов, соответствующих нелинейным переменным. Доказательство существования индексного вектора  $I$  следует из невырожденности матрицы  $\Phi$ . Обычно для радиоэлектронных схем можно строить довольно обширное семейство векторов  $I$ , однако легко представить ситуацию, когда выбор является единственным.

Шаг 3. Преобразуем (1) к виду

$$V_1 = W_1 - AV_2, \quad (3)$$

где  $V_1$  — нелинейные переменные,  $V_2$  — прочие переменные из  $V$ , которые мы будем называть линейными. Требования



к выбору индексного вектора  $I$  обеспечивают существование квадратной матрицы  $A$ .

Шаг 4. Реализация шага интегрирования производится в моделях компонентов следующим образом. Рассмотрим уравнение

$$\frac{d\Psi(\bar{V})}{dt} = f(\bar{V}, t). \quad (4)$$

Данное уравнение может входить в модель произвольного многополюсника,  $\bar{V}$  — переменные на выводах этого многополюсника,  $\Psi\bar{V}$  — линейная форма с постоянными коэффициентами. Распишем шаг интегрирования по неявной формуле Эйлера и приведем полученное выражение к виду нелинейного алгебраического уравнения

$$F = \Psi(\bar{V}(t)) - \Psi(\bar{V}(t+h)) + hf(\bar{V}(t+h), t+h) = 0. \quad (5)$$

Здесь  $t$  — текущее время,  $h$  — шаг интегрирования. Формула (5) входит в машинную реализацию модели. По заданным переменным на выводах обеспечивается вычисление невязки  $F$ . В модель включается также вычисление производных от  $F$  по переменным на выводах в точке  $(t+h)$ .

Алгоритм, использующий описанное представление моделей, осуществляет решение нелинейных алгебраических уравнений вида (5) и обеспечивает выбор шага  $h$ .

Шаг 5. На данном шаге совершается шаг методом Ньютона. Как указывалось выше, повторных шагов для уточнения решения в точке  $(t+h)$  мы не производим. Необходимая точность обеспечивается выбором достаточно малой величины  $h$ . Для непрерывных процессов при малых  $h$  хорошим начальным приближением будет решение в точке  $t$ . В связи с этим вид формулы (5) упрощается. Итерационная формула Ньютона имеет вид в данном случае

$$V_1(t+h) = V_1(t) - J^{-1}F(t).$$

Для упрощения вычислительного процесса приведем ее к виду

$$JV_1(t+h) = JV_1(t) - F(t) \quad (6)$$

Решением системы линейных алгебраических уравнений (6) реализуется шаг методом Ньютона. Рассмотрим методику построения матрицы Якоби, которая обозначена в (6) буквой  $J$ .  $F(t)$  — вектор невязок, вычисляемый в моделях. Система (6) расписывается относительно нелинейных уравнений

модели цепи. Переменные на выводах нелинейных многоплюсников, по которым производится дифференцирование нелинейных форм (5), могут быть и нелинейными и линейными. Если переменная является нелинейной, то соответствующая частная производная из модели пересылается в матрицу  $Y$ . Для линейных переменных используется правило дифференцирования сложных функций. При этом значения производных от линейных переменных по нелинейным выбираются из соответствующих позиций матрицы  $A$  выражения (3). Таким образом, удастся достаточно просто строить матрицу Якоби без численного дифференцирования.

После построения матрицы Якоби вычисляется вектор правых частей в (6) и решается система линейных алгебраических уравнений. Получаем значение нелинейных переменных в точке  $(t+h)$ . Для вычисления значений линейных переменных используем формулу (3) и объединяем полученные данные в общее решение цепи  $V$ . На этом основной цикл вычислительного процесса заканчивается. Перевычислять весь вектор решений цепи  $V$  необходимо на каждом шаге интегрирования, так как для вычисления (5) может потребоваться произвольная комбинация переменных.

Шаг 6. Выбираем значение  $h$  для следующего шага интегрирования. В практически реализованной программе решение осуществляется с абсолютной точностью  $\epsilon$ , которая устанавливается путем пересчета с половинным шагом. По-видимому, возможны и другие критерии выбора  $h$ . После выбора шага интегрирования обрабатываем полученное решение  $V$  и обращаемся к шагу 4. Алгоритм работу заканчивает при достижении заданного времени.

Описанный выше алгоритм является в высокой степени универсальным, так как при его помощи можно производить анализ нелинейных статических и динамических цепей. Анализ динамических цепей описан подробно. Статические модели оформляются аналогично динамическим. Формула (5) при этом принимает вид

$$F(V, t) = 0. \quad (7)$$

Для простоты организации алгоритма и соответствующей программы модели источников тока и напряжения вводятся также в виде (7) и решаются как нелинейные алгебраические уравнения с использованием значений производных по переменным на выводах.

Если переменных (зависящих от времени) источников в цепи не имеется, то при любом исходном шаге  $h$  приведен-



ный алгоритм эквивалентен многошаговому методу Ньютона. В силу того, что сходимость метода Ньютона существенно зависит от точности начального приближения, построить которое обычно не удастся, удобно заменять в модели цепи источники постоянного тока или напряжения на источники с плавным возрастанием внешних параметров до номинальных от нулевых начальных значений. Автоматическая выборка шага  $h$  гарантирует сходимость вычислительного процесса, дополнительные итерации при установившихся номинальных значениях источников обеспечат заданную точность полученного решения. Время установления номинальных значений источников выбирается на основании физических соображений. При этом требуется, чтобы при промежуточных вычислениях схема не выходила на критические режимы. Так, например, источники питания должны выходить на номинальный режим значительно быстрее источников выходных сигналов.

Обычно при интегрировании моделей радиоэлектронных цепей от нулевых начальных данных матриц Якоби является невырожденный и вычислительный процесс можно производить по приведенному выше алгоритму, как для динамических, так и для статических цепей. Однако в некоторых случаях, особенно при исследовании цепей неэлектрического характера, шаг от нулевых начальных данных произвести не удастся вследствие вырожденности матрицы Якоби. Практически это затруднение удастся обойти, задавая малый исходный шаг  $h$  и произвольные, близкие к нулю, значения переменных цепи. Затем решение в точке  $t=h$  уточняется до заданной точности последовательными итерациями методом Ньютона. Сходимость в данном случае обеспечивается малостью шага  $h$ . Для статических цепей этот путь является корректным, так как значения начальных данных не влияют на поведение решения в дальнейшем. При анализе неустойчивых динамических цепей внесенная в начальные данные погрешность может существенно исказить результат. Чтобы избежать этого, один или несколько первых шагов интегрирования совершаем по любой явной схеме с достаточно малым шагом. Используя полученное решение в качестве начальных данных, продолжаем процесс по схеме неявного интегрирования.

## § 2. Структура моделей

Опишем процесс подготовки и программирования моделей для алгоритма неявного интегрирования. В качестве базовой будем использовать библиотеку моделей, приведенную в [2].

Первым этапом подготовки модели является построение нелинейных форм вида (5) или (7) и вычисление производных от этих форм по переменным на выводах. Дифференцирование производится сначала по напряжениям, затем по токам в последовательности, заданной фиксированной нумерацией выводов моделируемого многополюсника.

Исходной информацией для работы модели является значение времени  $t$ , шага  $h$  и переменных на выводах многополюсника. Эти значения заносятся общей программой анализа в соответствии с заданным распределением памяти. Адрес начала вектора исходных данных при обращении к модели заносится в регистр адреса РА и по второму адресу А2 ячейки 0006. В качестве результата модель выдает значение нелинейной формы (5) или (7) и значения производных с ячейки, адрес которой указан по А2 ячейки 0005. Начало вычислительного блока модели для очередного нелинейного уравнения по А2 ячейки 0007. В целом структура моделей неявного интегрирования совпадает с описанной в [2]. Отметим лишь, что в данном случае линейные формы в левых частях дифференциальных и нелинейных алгебраических уравнений отсутствуют.

Программная реализация вычислительных блоков моделей неявного интегрирования имеет особенность, связанную с тем, что выражения для нелинейных форм и производных оказываются очень большими. Поэтому объем моделей может оказаться чрезмерным, что приводит к ограничению размерности исследуемой цепи из-за нехватки памяти. Рекомендуется выносить вычисление больших выражений в операторную часть системы. Если объем памяти ограничен, операторная часть модели включается в динамический обмен с магнитными барабанами. Описанный прием позволяет сократить постоянно хранящуюся в оперативном запоминающем устройстве модель цепи и избежать многократной записи одинаковых вычислительных блоков для однотипных элементов с различными внутренними параметрами.

Приведем примеры подготовки моделей компонентов радиоэлектронных цепей для неявного интегрирования.

Пример 1. Модель емкости  $C$ .  
Основное уравнение имеет вид

$$\frac{d(U_1 - U_2)}{dt} = \frac{1}{C} J.$$

Распишем шаг по неявной схеме Эйлера.

$$F = U_1(t) - U_2(t) - U_1(t+h) + U_2(t+h) - h/c J(t+h) = 0.$$



Напомним, что в алгоритме используется одношаговый метод Ньютона с начальным приближением из предыдущего шага. Поэтому после приведения к общему знаменателю и приведения подобных получаем нелинейную форму  $F = -hI = 0$ . Производные по напряжениям и токам на выводах имеют соответственно вид  $-C, C, -h, 0$ .

В силу особенности организации программы вторую производную по току для двухполюсников полагаем равной нулю. Отметим также, что приведение к общему знаменателю обязательно для моделей всех компонентов. В противном случае за счет появления малых делителей процесс интегрирования будет неустойчивым.

**Пример 2.** Модель источника синусоидального сигнала. Основное уравнение имеет вид

$$U_1 - U_2 = A \sin(Bt + C).$$

Нелинейная форма имеет вид

$$F = U_1 - U_2 - A \sin(Bt + C) = 0$$

с производными 1, -1, 0, 0.

**Пример 3.** Статическая модель диода. Основное уравнение имеет вид

$$I_2 = I_{\text{тз}} [\exp(U_1 - I_2 R_0 - U_2) Q].$$

Нелинейная форма и производные от нее вычисляются соответственно по формулам:  $F = I_2 - I_{\text{д}}$ , где

$$J_{\text{д}} = J_{\text{тз}} [\exp(U_1 - J_2 R_0 - U_2) Q];$$
$$\frac{\partial F}{\partial U_1} = -J_{\text{д}} Q; \quad \frac{\partial F}{\partial U_2} = J_{\text{д}} Q; \quad \frac{\partial F}{\partial J_2} = 1 + J_{\text{д}} \cdot R_0 \cdot Q.$$

### § 3. Программная организация алгоритма

Описанный выше алгоритм анализа нелинейных цепей реализован несколькими операторами системы МАРС и общим компилирующим оператором НИМ (Неявное Интегрирование в Модельном методе). Разбиение общего алгоритма на ряд операторов позволяет предъявлять менее жесткие требования к размерам рабочего поля РП интерпретирующей системы Ридер, упрощает программирование и отладку операторов. Кроме того, отдельные операторы имеют и самостоятельное значение. Их можно использовать при реализации других алгоритмов с идентичными этапами.

Обращение к оператору неявного интегрирования имеет вид: НИМ, ТИП, СХ, Я, А, Z T, Z H, Z E, O, K, V 1, V 2, O, I, V, K 1; здесь СХ — название исследуемой схемы либо адрес начала ее модели. Я — рабочий массив в 1-м кубе МОЗУ размерности  $(6 \times L + 320 + k(k+1))$ , ячеек, если нужно выдавать графическую информацию и  $(6 \times L + 50 + k(k+1))$  ячеек в противном случае. Здесь  $L$  — размерность цепи (суммарное количество узлов и ветвей),  $k$  — количество нелинейных переменных цепи.

А — адрес начала рабочего массива в 0-м кубе МОЗУ для размещения  $\Phi$ -матрицы.

T — время интегрирования, служит критерием окончания счета и выхода из оператора. Начальное время всегда получается нулевым. Буква Z имеет служебный характер, связанный с особенностями входного языка системы МАРС. Она обозначает, что следующий за ней аргумент занимает целую ячейку.

H — начальный шаг интегрирования. Рекомендуется задавать его заведомо малым, так как программа обеспечивает автоматическую выборку шага.

E — абсолютная точность шага интегрирования. В программе сравнивается с невязкой в решениях, полученных с заданным и с половинным шагом. При задании величины E следует учитывать, что в цепи обычно напряжения и токи резко различаются по величине.

I — адрес начала вектора, задающего количество и индексы нелинейных переменных.

V — решение цепи. В эти же ячейки следует занести начальные данные перед обращением к оператору.

Аргументы ТИП, K, V 1, V 2 K 1 управляют выдачей информации на печать. Представим восьмеричную триаду ТИП в двоичной форме 000 000  $a_1 a_2 a_3$ . Если  $a_3 = 1$ , то через K шагов ( $K \geq 1$ ) на печать выдается решение цепи V по СП-22 из ИС-22. Если  $a_2 = 1$ , та же информация выдается на печать при помощи специального оператора ТАБ печати таблиц. Отметим, что не печатается  $U_0 \equiv 0$ , вместо него выдается текущее значение времени. Если  $a_1 = 1$ , то через K1 шагов на график откладывается значение времени t и переменные V1, V2 (в обращении указываются их индексы). Выдача времени необходима для правильного масштабирования графиков, так как шаг интегрирования является переменным. Если необходима пошаговая обработка результатов нестандартного характера, рекомендуем осуществить ее специальным опера-



тором, присвоив ему название НПМ и сохранив структуру обращения к этому оператору.

Рассмотрим структуру программы. Непосредственно оператор НИМ осуществляет автоматическую выборку шага интегрирования и управляет режимом печати. Прочие действия выполняются во вспомогательных операторах, к которым обращается НИМ. Начинается оператор НИМ подготовительным блоком, в котором при помощи оператора ФТ в 0-м кубе формируется прямоугольная  $\Phi$ -матрица. Затем оператор ФМ (Формирование Матрицы) выражает линейные переменные через нелинейные. Основной цикл вычислений производится операторами ЯКОБИ и ФРС (Формирование и Решение Системы). Оператор ЯКОБИ опрашивает модели компонентов и, используя  $A$ -матрицу, строит матрицу частных производных. Далее оператор ФРС совершает шаг методом Ньютона и пересчитывает значения линейных переменных по формуле (3). Полученные значения нелинейных и линейных переменных рассылаются в соответствующие позиции вектора  $V$ . Возвращаемся в оператор НИМ. Известно, что скорость сходимости метода Ньютона связана с выбором нелинейных переменных. Выше показано, что вместе с тем выбор этого вектора не может быть произвольным. Предполагая, что исследуемая цепь мало насыщена нелинейными элементами, мы можем реализовать режим работы программы с переменными вектором нелинейных переменных. Для этого в модели встраиваются дополнительные переключательные блоки, которые выдают индекс нелинейной переменной для каждого уравнения модели в зависимости от значений переменных на выводах. Критерии выбора нелинейных переменных для основных компонентов радиоэлектронных цепей хорошо известны, и здесь мы на них останавливаться не будем.

Дополнительный оператор  $\Phi I$  (формирование вектора  $I$ ) опрашивает переключательные блоки моделей и строит новый вектор  $I$ . Пользователь может установить режим работы оператора  $\Phi I$ , обеспечивающий контроль вырожденности  $\Phi$ -матрицы или отсутствие этого контроля. В первом случае отвергаются изменения в векторе  $I$ , если они ведут к вырожденности  $\Phi$ -матрицы. При отсутствии контроля счет прерывается в операторе ФМ и решение задачи прекращается. Выбор режима легко осуществляется на основе визуального изучения исследуемой схемы. Отметим, что далеко не всегда полезно пользоваться переменным вектором  $I$ . Дело в том, что величина шага интегрирования определяется наихудшим случаем, поэтому

частичная вариация бесполезна, но приводит к большим потерям времени, так как после изменения вектора  $I$  необходимо новое обращение к ФМ и ФТ. Даже небольшой опыт практической работы с системой МАРС позволяет подбирать вектор  $I$ , обеспечивающий удовлетворительное быстродействие.

В заключение отметим, что возможно построение формул интегрирования более высоких порядков, однако это связано с большими потерями памяти, что недопустимо для ЭВМ типа М-220. Существенным ограничением на практическое использование оператора НИМ является его низкое быстродействие, связанное с необходимостью обращения матрицы Якоби на каждом шаге интегрирования. Достоинством методики является сравнительная простота и универсальность. Можно сделать вывод, что применение методов неявного интегрирования позволяет производить однократный анализ довольно сложных цепей на ЭВМ средней мощности. Ограниченное быстродействие не позволяет осуществлять многократный анализ цепей, необходимый при комплексном решении задач проектирования, в частности, при статистическом исследовании и параметрической оптимизации радиоэлектронных цепей.

#### ЛИТЕРАТУРА

1. Арайс Е. А. Модельный метод анализа радиоэлектронных схем. В сб.: Вопросы программирования и автоматизации проектирования. Вып. 2, Изд-во Томского университета, 1972.
  2. Арайс Е. А., Дмитриев В. М. Математические модели основных компонентов радиоэлектроники. Там же.
-



## ГРАФИЧЕСКАЯ ФОРМА МОДЕЛЬНОГО МЕТОДА

В. М. ДМИТРИЕВ

В исследовании характеристик и свойств физических систем важную роль играет построение математических моделей. В инженерной практике такие модели строятся в алгебраической либо в графической форме. В машинном проектировании цепей методы составления уравнений модели цепи делят на алгебраические, топологические и смешанные [1]. К первым относятся методы построения передаточных функций линейных систем, основанные на теоремах об определителе суммы и произведения матриц. Топологические методы решают ту же задачу с помощью теории графов.

Как те, так и другие методы связаны со сложными комбинаторными процедурами и в вычислительном отношении считаются малоэффективными. Следует, однако, отметить, что в настоящее время топологические методы получили свое дальнейшее развитие в связи с появлением теоретико-множественных методов представления линейных систем [2], [3].

Наиболее эффективное использование в современных программах машинного анализа цепей находят смешанные методы [4], в которых топологическое исследование цепи ограничивается лишь построением матриц соединений, характеризующих структуру системы. Модельный метод [5] и табличные методы анализа цепей [6] вообще исключают в явном виде исследование топологии цепи.

Несмотря на отмеченную тенденцию методов автоматического моделирования уменьшить или исключить анализ топологических свойств цепи и тем самым упростить процедуру построения ее модели, применение линейных графов к анализу цепей оправдано следующими причинами.

1. Графическая модель цепи позволяет легко выявить топологическую структуру. С ее помощью можно выделить ряд фундаментальных связей в цепи, а также произвести необходимые упрощения. Уравнения равновесия систем выражают лишь количественные соотношения между переменными.

2. Построение линейных и других графов цепей может служить универсальным аппаратом для формального описания систем, содержащих разные по физическому характеру компоненты (электрические, механические, гидравлические и др.).

3. Решение задач анализа цепей большой размерности а также автоматического синтеза наиболее удобно с привлечением графических моделей.

В данной работе рассматривается графическая модель модельного метода. Построен соответствующий этой форме линейный граф и граф связей.

### Обобщенный граф модельного метода

Уравнения модельного метода строятся в полном координатном базисе, переменными которого являются токи ветвей ( $I$ ) и напряжения узлов схемы ( $U$ ). Законом связей переменных является первый закон Кирхгофа. Для линейных цепей постоянного и переменного тока общая форма этих уравнений имеет следующий вид:

$$\begin{bmatrix} \Pi & -Z\Gamma \\ 0 & A^t \end{bmatrix} \begin{bmatrix} U \\ J \end{bmatrix} = \begin{bmatrix} Q \\ 0 \end{bmatrix}$$

или

$$\begin{bmatrix} Y\Pi & -\Gamma \\ 0 & A^t \end{bmatrix} \begin{bmatrix} U \\ J \end{bmatrix} = \begin{bmatrix} Q \\ 0 \end{bmatrix}$$

Здесь  $\Pi$  и  $\Gamma$  — матрицы задающих напряжений и токов соответственно;  $Q = \begin{bmatrix} E \\ I \end{bmatrix}$  — вектор задающих источников сигнала,  $A$  — матрица инцидентий,  $Z$  и  $Y$  — матрицы импедансов и проводимостей соответственно.

Обобщенный граф для модельного метода, соответствующий этим уравнениям, приведен на рис. 1.

Структуру обобщенного графа можно отобразить следующими блоками (подграфами):

$$\Pi\text{-граф} \Rightarrow H\text{-граф} \iff K\text{-граф} \iff T\text{-граф}$$



Здесь  $\Pi$  — потенциальный граф,  $H$  — граф напряжений на ветвях,  $K$  — компонентный граф,  $T$  — токовый граф.

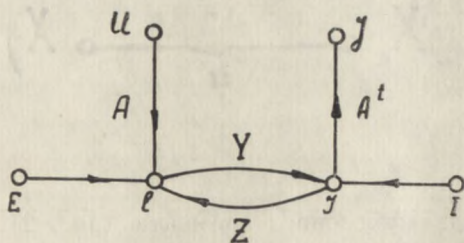


Рис. 1

Легко установить изоморфизм матричного и графического представления уравнений модельного метода. В табл. 1 приведены отдельные подграфы обобщенного графа (М-графа) и соответствующие им матричные соотношения.

Таблица 1

| Структура М-графа    | Матричное соотношение                                                                                                      |
|----------------------|----------------------------------------------------------------------------------------------------------------------------|
| ( $\Pi$ — $H$ )-граф | $AU = e$                                                                                                                   |
| $K$ -граф            | $\begin{vmatrix} e \\ J \end{vmatrix} = \begin{vmatrix} Z & 0 \\ 0 & Y \end{vmatrix} \begin{vmatrix} J \\ e \end{vmatrix}$ |
| $T$ -граф            | $A^t J = 0$                                                                                                                |

### Анализ структуры М-графа

Вершинами М-графа являются переменные цепи: основные (потенциальные  $U$  и токовые  $I$ ) и промежуточные ( $e, j$ ). Поскольку модельный метод не использует промежуточные переменные в явном виде, при построении графа они могут не участвовать. Ребрами М-графа являются коэффициенты уравнений, описывающих компоненты схемы. Для  $\Pi$ - $H$ -графа и  $T$ -графа это единичные коэффициенты. Общее число вершин М-графа равно  $K + M$ , где  $K$  — число узлов, а  $M$  — число ветвей в цепи. Число ребер равно  $L$ , где  $L$  — общее число нулевых коэффициентов в моделях компонентов цепи.

Прежде чем перейти к обсуждению правил построения М-графа, введем понятие элементарного графа или ячейки

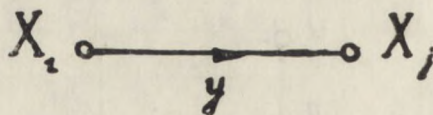


Рис. 2

графа. Для двухполюсного элемента (рис. 2). общее уравнение связи  $y$  с  $(x_i, x_j)$  имеет вид

$$y_i = T_i \Pi_i x.$$

Различные формы такой связи образуют набор разных по физическому характеру элементов (табл. 2).

Таблица

| Название            | Уравнение                                                                                                                                                                                                                                           | Элементарный граф |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| Унистор             | $y_i = T_i \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} \begin{vmatrix} x_i \\ x_j \end{vmatrix}$                                                                                                                                                   |                   |
| Гиристор            | $y_i = T_i \begin{vmatrix} 1 & 1 \\ 0 & 1 \end{vmatrix} \begin{vmatrix} x_i \\ x_j \end{vmatrix}$                                                                                                                                                   |                   |
| Униполярный элемент | $y_i = \begin{cases} T_i \begin{vmatrix} 1 & -1 \\ 0 & 0 \end{vmatrix} \begin{vmatrix} x_i \\ x_j \end{vmatrix}, & x_i > x_j \\ T_i \begin{vmatrix} 0 & 0 \\ 0 & 0 \end{vmatrix} \begin{vmatrix} x_i \\ x_j \end{vmatrix}, & x_i < x_j \end{cases}$ |                   |
| Биполярный элемент  | $y_i = T_i \begin{vmatrix} 1 & -1 \\ 0 & 1 \end{vmatrix} \begin{vmatrix} y_i \\ x_j \end{vmatrix}$                                                                                                                                                  |                   |



Ниже будут рассматриваться графы, содержащие преимущественно биполярные элементы.  $T_i$  имеет здесь характер проводимости и соответственно  $1/T_i$  характер импеданса. Всякий элементарный граф будем далее обозначать  $\mu$ -графом.

Для двухполюсных цепей каждый  $\mu$ -граф изоморфен соответствующей строке  $\Phi$ -матрицы. Структура такого изоморфизма имеет следующий вид:

$$\{\text{Строка } \Phi\text{-матрицы}\} \Leftarrow \{\text{Уравнение элемента}\} \Rightarrow \{\mu\text{-граф}\}$$

$M$ -граф можно представить через  $\mu$ -графы следующим образом:

$$M = \sum_{i=1}^N \mu_i + T \quad (4)$$

Здесь  $T$  — токовый граф, а  $N$  — число элементов в цепи.

Уравнение для токового графа удобно переписать в следующем виде:

$$[E - A'_1] \begin{vmatrix} J_1 \\ J_2 \end{vmatrix} = 0, \quad (5)$$

где  $E$  — единичная матрица размерности  $(k \times k)$ ; размерность вектора  $I_1$  равна  $k$ , а вектора  $I_2$  —  $m - k$ . Такое представление исключает из  $T$ -графа промежуточные переменные ( $j$ ) и упрощает структуру графа.

Теперь процедуру построения  $M$ -графа можно описать следующим образом.

1. Наносится поле потенциальных величин — узловые напряжения цепи.
2. Наносится поле токовых величин — токи ветвей цепи.
3. В соответствии с формулой (4) строится вся совокупность  $\mu$ -графов, которая связывает поле токовых и потенциальных величин. При этом образуются  $\Pi$ -и  $K$ -графы.
4. По формуле (5) строится  $T$ -граф. На этом построение невозбужденной части  $M$ -графа заканчивается.

Для того чтобы построить возбужденный  $M$ -граф, необходимо ввести узлы, соответствующие источникам тока и напряжения, и объединить их с соответствующими узлами невозбужденного  $M$ -графа.

Для схемы (рис. 3) дан пример построения  $M$ -графа, содержащего и не содержащего промежуточные переменные.

Сравним число узлов и число ребер построенных графов. Для графа (рис. 3а) число узлов может быть найдено по формуле

$$K_1 = 2(k-1+m),$$

а для графа (рис. 3б) оно равно

$$K_2 = k-1+m.$$

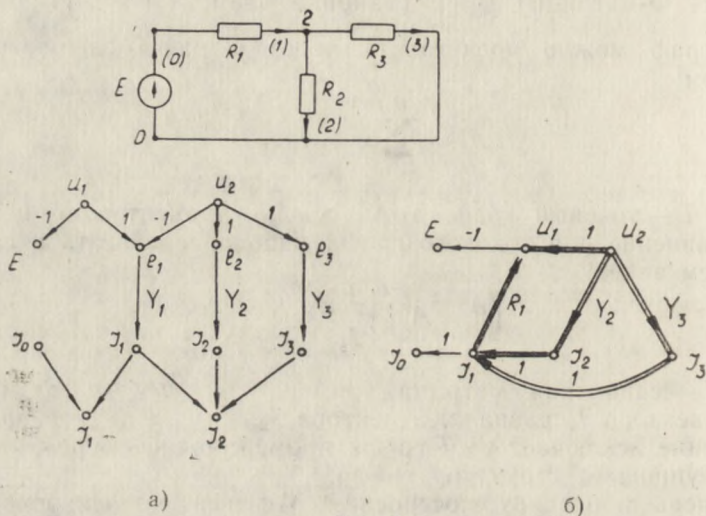


Рис. 3

Соответственно число ребер для первого и второго графов определяется следующим образом:

$$M_1 = 2(2m - m_0) + N;$$

$$M_2 = (m - m_0) + k + N,$$

где  $m_0$  — число ветвей, инцидентных с нулевым узлом,  $N$  — число элементов в цепи.

Разница числа узлов и ребер построенных графов определяется из соотношений:

$$\Delta K = k - 1 + m;$$

$$\Delta M = 3m - m_0 - k.$$

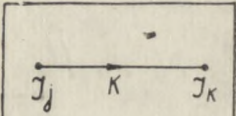
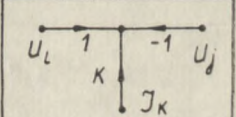
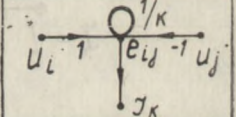
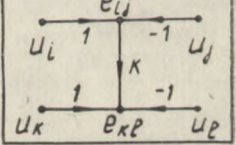
Для графов (рис. 3а и 3б)  $\Delta k = 5$  и  $\Delta M = 5$ .



## М-графы, содержащие зависимые источники и многополюсники

Появление зависимых источников в цепи не вызывает изменений в методике построения М-графа. Всякий зависимый источник рассматривается наравне с другими компонентами цепи и своим  $\mu$ -графом вписывается в общую структуру М-графа. В табл. 3 приведены элементарные графы для оси основных типов зависимых источников.

Таблица 3

| Название                                               | Уравнение                                                         | $\mu$ -граф                                                                        |
|--------------------------------------------------------|-------------------------------------------------------------------|------------------------------------------------------------------------------------|
| Зависимый источник тока, управляемый током             | $J_K = KJ_J$                                                      |   |
| Зависимый источник напряжения, управляемый током       | $e_{ij} = U_i - U_j = KJ_K$                                       |   |
| Зависимый источник тока, управляемый напряжением       | $J_K = 1/K (U_i - U_j)$                                           |   |
| Зависимый источник напряжения, управляемый напряжением | $K e_{ij} = e_K l$<br>$e_{ij} = U_i - U_j$<br>$e_K l = U_K - U_l$ |  |

Для построения М-графа, содержащего многополюсные компоненты с неизвестной структурой, необходимо иметь либо математическую модель многополюсника, либо схему его замещения. Наличие этих данных позволяет построить  $\mu$ -граф подсхемы или многополюсника. Для этого переменные подсхемы разделяются на внешние и внутренние. Строится соответствующий  $\mu$ -граф и из него исключаются узлы,

соответствующие внутренним переменным. Своими внешними узлами полученный после упрощения  $\mu$ -граф навешивается на соответствующие узлы  $M$ -графа.

### Упрощение $M$ -графа

Правила упрощения  $M$ -графа ничем не отличаются от общеизвестных правил упрощения линейных графов.

В качестве примеров упрощения  $M$ -графа рассмотрим переход от графа модельного метода к графу уравнений метода узловых потенциалов и передаточной функции.

Так, для того чтобы от графа (рис. 3а) перейти к графу метода узловых потенциалов, достаточно исключить в  $T$ -графе токи ветвей. При этом граф (рис. 3а) примет вид (рис. 4).

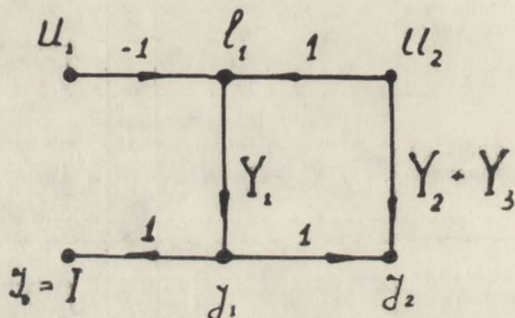


Рис. 4

Для него легко записать уравнения узловых потенциалов

$$I = Y_1 (U_1 - U_2);$$

$$0 = Y_1 (U_1 - U_2) - (Y_2 + Y_3) U_2.$$

Чтобы построить передаточную функцию  $T = U/U_1$  для схемы (рис. 3) воспользуемся графом (рис. 3б). В этом графе необходимо просуммировать передачи всех путей, ведущих из  $U_2$  к  $U_1$ , исключая в них все промежуточные переменные. На рис. 3б эти пути обозначены двойными стрелками.

В результате такого суммирования получим

$$U_1 = R_1 (Y_2 + Y_1) U_2 + U_2$$



или после преобразования

$$\frac{U_2}{U_1} = \frac{R_2 \cdot R_3}{R_1 R_2 + R_1 R_3 + R_2 R_3}.$$

### Модельные графы связей

Модельные графы связей можно получить с помощью перехода от алгебраической диаграммы, связывающей переменные графов связей, к алгебраической диаграмме *М*-графа. При этом необходимо использовать основные понятия графов связи [8].

Графом связи (ГС) называется в общем случае линейный граф, вершинами которого являются многосвязные элементы, а дугами — его связи.

Переменными полусвязей (свободных связей) многосвязного элемента определены следующие шесть величин. Усилие  $e(t)$  и поток  $f(t)$ , которые непосредственно относятся к данной полусвязи, называются переменными мощности полусвязи. Мощность  $P(t)$  определяется как скалярное произведение усилия на поток.

$$P(t) = e(t) \cdot f(t) \quad (6)$$

Кроме этого, используются три интегральные величины: момент

$$p(t) = p(t_0) + \int_{t_0}^t e(\lambda) d\lambda; \quad (7)$$

перемещение

$$q(t) = q(t_0) + \int_{t_0}^t f(\lambda) d\lambda; \quad (8)$$

и энергия

$$E(t) = E(t_0) + \int_{t_0}^t P(\lambda) d\lambda. \quad (9)$$

Обычно для ГС изображают только усилие и поток следом за полусвязью или связью. Остальные величины определяются из приведенных выше формул.

Основные многосвязные элементы делятся на четыре категории в соответствии с их энергетическими характеристиками.

1. Источники: а) усилия ( $SE$ ),  
б) потока ( $SF$ ).

2. Аккумуляторы: а) емкость ( $C$ ),  
б) инерционность ( $I$ ).
3. Потери: сопротивление ( $R$ ).
4. Узлы: а) трансформатор ( $TF$ ),

$$e_1 = m e_2; \quad m f_1 = f_2; \quad (10)$$

- б) гиратор ( $GY$ ),

$$e_1 = r \cdot f_2; \quad e_2 = r \cdot f_1; \quad (11)$$

- в) узел общего усилия ( $Y_0$ -узел)

$$e_1 = e_2 = e_3; \quad f_1 + f_2 - f_3 = 0; \quad (12)$$

- г) узел общего потока ( $Y_I$ -узел)

$$f_1 = f_2 = f_3; \quad e_1 + e_2 - e_3 = 0. \quad (13)$$

Узлами эти элементы названы потому, что они сохраняют мощность. В графах связей легко совершается переход от многосвязных элементов к многосвязным полям, когда появляется необходимость рассмотрения элементов с  $n$  полусвязями ( $n > 2$ ).

Исходя из рассмотренных выше основных понятий ГС, легко построить алгебраическую диаграмму связи переменных физических систем и характеризующих соотношений. На рис. 5а такая диаграмма представлена для переменных ГС, а на рис. 5б для переменных электрических цепей.

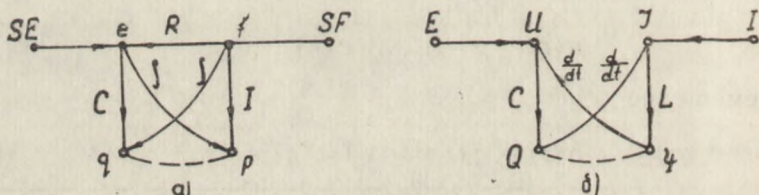


Рис. 5

Из диаграмм (рис. 5) можно установить систему аналогий между переменными ГС и переменными электрических цепей (табл. 4).

Теперь нетрудно построить обобщенную структурную схему ГС. Она примет следующий вид (рис. 6).

Отметим далее сходства и различия, которые имеются в моделировании схем с помощью ГС и модельных графов. К сходствам этих двух подходов можно отнести следующие.



| Переменные ГС   |             | Переменные ЭЦ                       |             |
|-----------------|-------------|-------------------------------------|-------------|
| название        | обозначение | название                            | обозначение |
| Усилие          | $e$         | Напряжение                          | $U$         |
| Поток           | $f$         | Ток                                 | $I$         |
| Перемещение     | $q$         | Заряд                               | $Q$         |
| Момент          | $p$         | Потокоцепление                      | $\Psi$      |
| Источник усилия | $SE$        | Источник напряжения                 | $E$         |
| Источник потока | $SF$        | Источник тока                       | $I$         |
| Сопротивление   | $R$         | Резистор                            | $R$         |
| Инерционность   | $I$         | Индуктивность                       | $L$         |
| Емкость         | $C$         | Емкость                             | $C$         |
| Нулевой узел    | $Y_0$       | Параллельное соединение выводов     | —           |
| Единичный узел  | $Y_1$       | Последовательное соединение выводов | —           |

1. Как М-графы, так и графы связей используют много-связный элемент (многополюсник) в качестве основного элементарного компонента цепи.

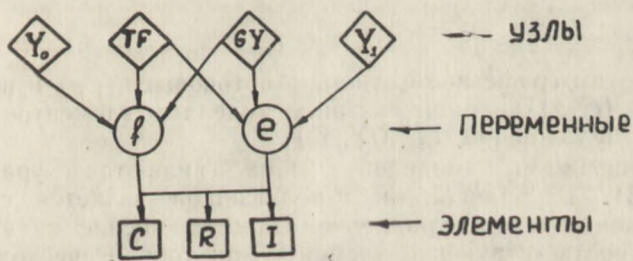


Рис. 6

2. Выводам многополюсника в М-графе соответствуют полусвязи многосвязного элемента в ГС.

3. Переменными полусвязей в ГС являются потоковые ( $f$ ) и потенциальные ( $e$ ) величины. Аналогичными перемен-

ными являются ток ( $I$ ) и потенциалы ( $U$ ) на выводах многополюсника.

Различия этих двух подходов состоят в следующем.

1. Модели элементов в ГС разграничены на четыре основные категории, с помощью которых, очевидно, можно представить любую цепь. Модельный метод допускает любые модели, например, составные, которые содержат в себе элементы всех или некоторых категорий.

2. ГС имеет явно выраженную узловую структуру, которая соответствует узловой структуре цепи. Узлами же М-графа являются переменные модельного метода.

На основе такого взаимного сопоставления может быть предложена обобщенная структура модельного графа связей (МГС), которая приведена на рис. 7.

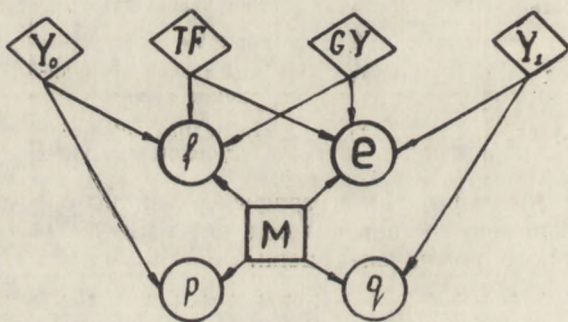


Рис. 7

На этом графе полный набор потоковых ( $f$ ,  $p$ ) и потенциальных ( $e$ ,  $g$ ) величин связан с моделями элементов ( $M$ ) и моделями узлов ( $Y_0$ ,  $TF$ ,  $GY$ ,  $Y_1$ ).

Уравнения моделей узлов являются уравнения (10—13), а уравнения  $n$ -полюсников является система, содержащая  $n = 1$  уравнение. Одно уравнение из этой системы, соответствующее элементу как топологическому узлу (обычно это  $Y_0$ -узел) выносится, и в узловой структуре графа появляется дополнительный элемент. Таким образом, общее уравнение модели элемента МГС, может быть записано в виде

$$\Psi(p, q, f, e) = W(t);$$

$$\sum_{i=1}^l f_i + \sum_{i=1}^n p_i = 0, \quad (14)$$



где  $\Psi$  — линейная форма от переменных  $p, q, f, e$ , зависящих от  $t$ , которая может содержать в себе операции дифференцирования,  $W(t)$  — линейная вектор-функция правых частей.

Модели элементов в МГС в наиболее общем случае могут быть связаны со всеми типами переменных. При этом лишь требуется, чтобы размерности произведений соответствующих переменных обеих групп были физически совместимыми.

Модели  $Y_0$ -узлов (поточковых узлов) и  $Y_1$ -узлов (узел усилий) связывают соответственно поточковые и потенциальные величины без преобразования мощности полусвязей (выводов). Модели  $TF$  и  $GY$  связывают поточковые и потенциальные величины между собой, также без преобразования мощности выводов.

Рассмотрим структуру МГС для линейной схемы переменного тока (рис. 8).

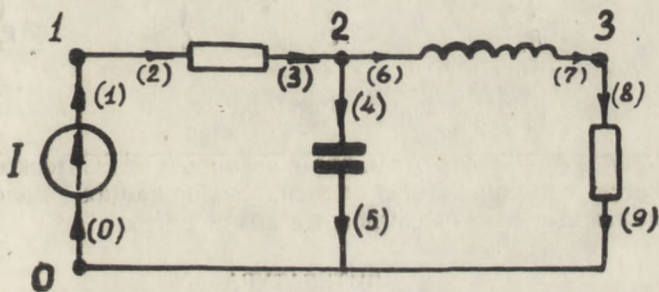


Рис. 8

Узлы межсоединений

1:  $Y_1$ —узел;

2:  $Y_0$ —узел.

Узлы компонентов

$R_1 \div R_3$  — все типа  $Y_0$

Модели узлов

а) межсоединений

1:  $I_0 - I_1 = 0$ ;

2:  $I_0 - I_1 - I_2 = 0$ ;

б)  $I_0 - I_1 = 0$  — для  $R_1, R_2, L$  и  $C$ .

Модели элементов

$$R_1; U_0 - U_1 - JR_1 = 0;$$

$$R_2; U_0 - U_1 - JR_2 = 0.$$

$$C: \frac{d}{dt}(U_0 - U_1) = \frac{1}{C} J;$$

$$L: \frac{dJ}{dt} = \frac{1}{L} (U_0 - U_1)$$

Схема графа (рис. 9)

По схеме графа (рис. 9) легко составить модель цепей (рис. 8), которая соответствует символической форме модельного метода [7].

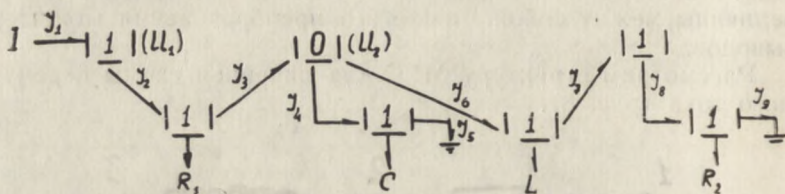


Рис. 9

В заключение отметим, что с помощью МГС легко осуществляется моделирование цепей, содержащих нелинейные двухполюсные и многополюсные компоненты.

#### ЛИТЕРАТУРА

1. Сигорский В. П. Алгоритмы анализа электронных схем. — «Радиоэлектроника», 1968, № 11.
2. Трохименко Я. К. Метод обобщенных чисел и анализ линейных цепей. М., «Советское радио», 1972.
3. Беллерт С. Топологический анализ и синтез линейных систем. — «Зарубежная радиоэлектроника», 1963, № 6.
4. Брэнин Ф. Методы анализа цепей с помощью ЭЦВМ ТИИЭ. т. 55, 1967, № 11.
5. Арайс Е. А. Модельный метод анализа цепей. — В сб.: Вопросы программирования и автоматизации проектирования. Изд-во ТГУ, 1973.
6. Weeks W. T. et al. Network Analysis Using a Sparse Tableau. Proc. of the 1973. Int. Symposium on Circuit Theory. April. 1973. Toronto.
7. Арайс Е. А., Дмитриев В. М. Символическое моделирование и анализ электронных схем на ЭВМ. В настоящем сборнике.



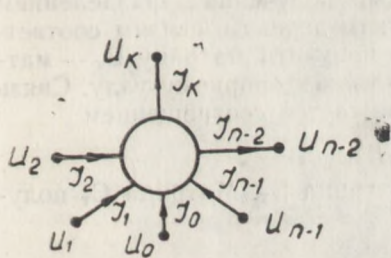
## СВЯЗЬ МОДЕЛЬНОГО МЕТОДА С ДРУГИМИ МЕТОДАМИ АНАЛИЗА ЦЕПЕЙ

В. М. ДМИТРИЕВ

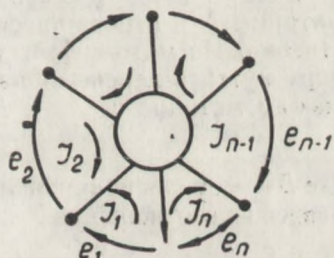
Для установления такой связи отметим прежде всего те характерные особенности, которые отличают один метод анализа от другого. К таким особенностям можно причислить следующие:

- а) используемый координатный базис;
- б) форму представления моделей компонентов;
- в) форму представления информации о функциях схемы.

Координатным базисом метода принято считать ту систему физических переменных, относительно которых записывается решение цепи по этому методу. К таким переменным относятся  $U$  — напряжения узлов,  $I$  — токи ветвей,  $e$  — напряжение вет-



а) относительно  $U$  и  $J$



б) относительно  $e$  и  $J$

Рис. 1

вей,  $j$  — контурные токи и некоторые другие, производные от этих величин, образуемые при топологическом исследовании цепи.

Форма представления модели компонента может быть общего вида двухполюсником или  $n$ -полюсником. Для двухполюсников используются обычно универсальные ветви типа  $Z$  или  $Y$ .  $YZ$ -ветвей уравнения выражают напряжения, а у  $Y$  — ветвей — токи. Для многополюсников, в соответствии с используемым координатным базисом, возможны два вида представлений (рис. 1).

В соответствии с формой представления информации о схеме методы анализа можно разделить на топологические, алгебраические и комбинированные. Ниже рассматриваются только последние.

### § 1. Связь модельного метода с методами одной переменной

Методы одной переменной имеют однородный координатный базис, содержащий переменные одного типа. Задачей установления связи между ними и модельным методом является переход от координатного базиса модельного метода к координатному базису интересующего нас метода.

Уравнения цепи в любом из методов одной переменной получаются с помощью набора топологических матриц. Основной из них является матрица инцидентий  $A$ , из которой могут быть получены матрица контуров  $C$  и матрица сечений  $D$ . Конкретный набор этих матриц зависит от выбранного метода анализа (рис. 2).

Здесь  $A_T$  и  $A_L$  — субматрицы, полученные разделением матрицы  $A$  и относящиеся к ветвям дерева и связям соответственно. Из матрицы  $A_T$  можно получить матрицу  $B_T$  — матрицу путей в дереве от каждого узла к опорному узлу. Связь между матрицами  $A_T$  и  $B_T$  выражается соотношением

$$A_T^{-1} = B_T^t,$$

где  $B_T^t$  — транспонированная матрица  $B_T$ . Матрица  $C_T$  получается из соотношения

$$C_T = -B_T \cdot A_L^t,$$

а матрица  $C_L$  — единичная матрица.

В матрице сечений  $D$  единичной является субматрица  $D_T$ , а матрица  $D_L$  находится из равенства

$$D_L = -C_T^t.$$

Подробные алгоритмы построения этих матриц здесь не рассматриваются, их можно найти в [1].



Приведем некоторые соотношения, которые будут использоваться ниже, а именно:

$$A^t C = 0; \quad (1)$$

$$D^t C = 0, \quad (2)$$

где  $A^t$  и  $D^t$  — транспонированные матрицы  $A$  и  $D$ .

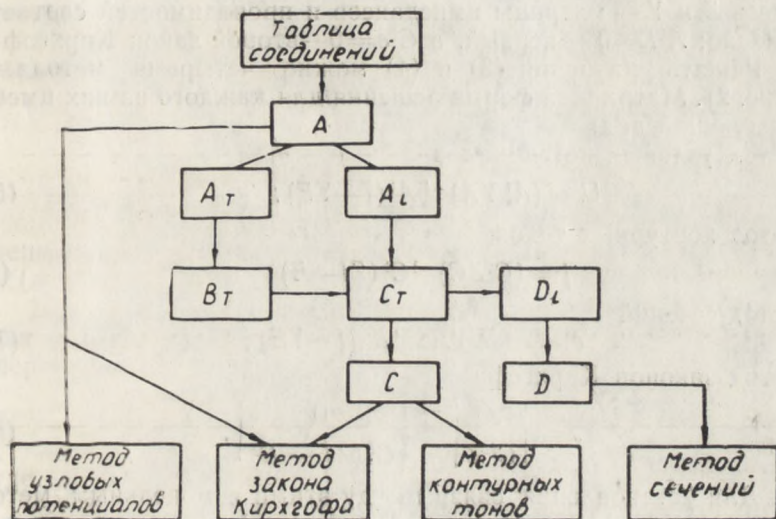


Рис. 2

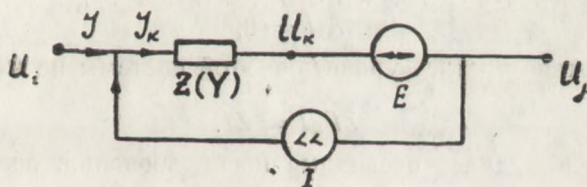


Рис. 3

Для линейных цепей постоянного тока будем полагать элементарным компонентом цепи универсальную ветвь (рис. 3).

Обозначим  $U_0 - U_1 = e$ . Тогда задача анализа схемы, структура которой содержит только ветви (рис. 3), будет заключаться в нахождении вектора переменных этой схемы, удовлетворяющих следующим уравнениям:

$$\begin{aligned} E + e &= Z(J + I); & I + J &= Y(e + E); \\ A^t I &= 0 & C^t e &= 0 \end{aligned} \quad (3) \quad (4)$$

Здесь  $Z$  и  $Y$  — матрицы импедансов и проводимостей соответственно;  $A^t I = 0$  — первый, а  $C^t e = 0$  — второй закон Кирхгофа.

Решать уравнения (3) и (4) можно четырьмя методами (рис. 2). Матричная форма решения для каждого из них имеет следующий вид:

метод узловых потенциалов

$$U = (A^t Y A)^{-1} A^t (I - Y E); \quad (5)$$

метод контурных токов

$$j = (C^t Z C)^{-1} C^t (Z I - E); \quad (6)$$

метод сечений

$$e_t = (D^t Y D)^{-1} D^t (I - Y E); \quad (7)$$

метод законов Кирхгофа

$$J = \begin{vmatrix} A^t & 0 \\ C^t Z & C^t (Z I - E) \end{vmatrix}^{-1} \begin{vmatrix} 0 \\ C^t (Z I - E) \end{vmatrix}. \quad (8)$$

Для установления связи между этими и модельным методом запишем модель цепи, содержащей только элементы (рис. 3), в модельном методе:

$$\begin{aligned} \Pi_R U - Z \Gamma_R J &= 0; \\ \Pi_Q U + 0 &= E; \\ 0 + P_Q J &= I; \\ A_t J &= 0. \end{aligned}$$

Умножая третье уравнение этой системы на матрицу  $Z$ , имеем

$$Z \Gamma_Q J = Z I.$$

Совершая далее очевидные преобразования, получим:

$$[\Pi_R - \Pi_Q] U - Z [\Gamma_R + \Gamma_Q] J = Z I - E.$$

Исключая из полученной системы внутренние переменные ветвей  $U_k$  и  $I_k$ , где  $k = \overline{1, m}$ , получим окончательно

$$\begin{aligned} \Pi_{RQ} U - Z \Gamma_{RQ} J &= -E + Z I; \\ A^t J &= 0. \end{aligned}$$



Здесь матрица  $\Pi_{RQ}$  совпадает с матрицей  $A$ , а  $Z\Gamma_{RQ} = Z$ , так как  $\Gamma_{RQ}$  — единичная матрица. В результате последнее уравнение можем переписать в следующем виде:

$$AU - ZI = -E + ZI; \quad (9)$$

$$A^t I = 0,$$

которое аналогично по форме уравнению (3), так как  $AU = e$ .

Решение уравнения (9) может быть записано следующим образом:

$$\begin{vmatrix} U \\ J \end{vmatrix} = \begin{vmatrix} A & -Z \\ 0 & A^t \end{vmatrix}^{-1} \begin{vmatrix} ZI - E \\ 0 \end{vmatrix}. \quad (10)$$

Матрица в правой части уравнения  $\begin{vmatrix} A & -Z \\ 0 & A^t \end{vmatrix}^{-1}$  — матрица решения в модельном методе. Ее размерность  $\{(m + \kappa) \times (\kappa + m)\}$ , где  $\kappa$  — число узлов, а  $m$  — число ветвей в схеме.

Для модельного метода может быть предложена и другая форма решения. Обратимся к уравнению (4). Его можно переписать в следующем виде:

$$Ye - Cj = I - YE; \quad (11)$$

$$C^t e = 0,$$

где  $Cj = I$ . Решение уравнения (11) ищется в виде

$$\begin{vmatrix} e \\ j \end{vmatrix} = \begin{vmatrix} Y & -C \\ C^t & 0 \end{vmatrix}^{-1} \begin{vmatrix} I - YE \\ 0 \end{vmatrix}. \quad (12)$$

Матрица  $\begin{vmatrix} Y & -C \\ C^t & 0 \end{vmatrix}^{-1}$  также является матрицей решения в модельном методе. Ее размерность  $\{(m + l) \times (l + m)\}$ , где  $l$  — число контуров в схеме.

Вторая форма модельного метода требует использования моделей компонентов, переменными в которых являются контурные токи и напряжения ветвей (рис. 16). Получение и использование таких моделей более затруднительно, чем моделей, переменными которых являются напряжения полюсов и токи ветвей.

Сравнивая две формы модельного метода, можно заметить, что наиболее удобной является форма (10), так как используемые в ней топологические матрицы  $A$  и  $A^t$ , а также  $Z$  могут быть получены прямо из исходной информации о схеме.

Перейдем теперь последовательно от уравнения (9) к уравнениям для методов одной переменной.

К методу узловых потенциалов

Умножая сначала (9) на матрицу  $Y$  слева, получим

$$YAU - I = J - YE;$$

$$A^t I = 0. \quad (13)$$

От уравнения (13) к решению по методу узловых потенциалов можно перейти двумя способами.

В первом случае домножим последнее уравнение на матрицу  $A^t$  слева

$$(A^t Y A) U - A^t J = A^t (I - YE)$$

и, учитывая, что  $A^t J = 0$ , получаем решение по методу узловых потенциалов

$$U = (A^t Y A)^{-1} \cdot A^t (I - YE).$$

К такому же уравнению можем прийти и простым исключением токовых переменных из уравнения (13). Так как последнее утверждение не вполне очевидно, рассмотрим простой пример схемы делителя напряжения (рис. 4). Система уравнений для этой схемы, соответствующая (13), имеет вид:

$$Y_1 (U_1 - U_2) - J_1 = 0;$$

$$Y_1 U_2 - J_2 = 0;$$

$$J_0 = I;$$

$$J_0 - J_1 = 0;$$

$$J_1 - J_2 = 0.$$

Здесь  $Y_1 = 1/R_1$ , а  $Y_2 = 1/R_2$ . Исключая все токи из этой системы, получим уравнение вида:

$$Y_1 (U_1 - U_2) = I;$$

$$Y_2 U_2 - Y_1 (U_1 - U_2) = 0,$$

которые легко приводятся к виду (5):

$$\begin{vmatrix} U_1 \\ U_2 \end{vmatrix} = \begin{vmatrix} Y_1 & -Y_1 \\ -Y_1 & Y_1 + Y_2 \end{vmatrix} \times \begin{vmatrix} I \\ 0 \end{vmatrix}.$$



## К методу контурных токов

Умножим уравнение (3) на матрицу  $C^t$  слева:

$$C^t A U - C^t Z J = C^t (Z I - E).$$

Учитывая, что  $C^t A U = 0$ ,  $J = C j$ , запишем решение по методу контурных токов:

$$j = (C^t Z C)^{-1} C^t (Z I - E)$$

Таким же образом можно прийти к методу сечений и методу законов Кирхгофа.

## § 2. Связь модельного метода с методом переменных состояния

Обобщенное представление цепи при составлении уравнений состояния имеет вид (рис. 4).

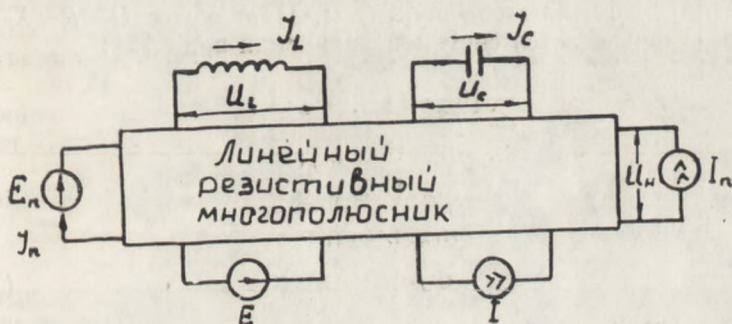


Рис. 4

Ему соответствует стандартная форма уравнений переменных состояния [2]:

а) для линейной цепи ( $E_n$  и  $I_n$  отсутствуют)

$$\frac{d}{dt} V_x = A V_x + B V_q; \quad (14)$$

б) для нелинейной цепи

$$\begin{aligned} \frac{d}{dt} V_x &= A V_x + B V_q + B' V_n; \\ W_n &= C V_x + D V_q + D' V_n. \end{aligned} \quad (15)$$

Здесь  $A, B, B', C, D, D'$  — матрицы линейных форм при соответствующих переменных. Метод использует следующий состав переменных:

$V_x = \begin{bmatrix} V_C \\ I_L \end{bmatrix}$  — вектор переменных состояния, в котором  $V_C$  — напряжение на емкости, а  $I_L$  — ток индуктивности;

$V_q = \begin{bmatrix} E \\ I \end{bmatrix}$  — вектор задающих источников;

$V_n = \begin{bmatrix} V_{nI} \\ J_{nE} \end{bmatrix}$  — вектор независимых переменных у нелинейных источников, в котором  $V_{nI}$  — напряжения на источниках тока, а  $J_{nE}$  — токи в источниках напряжения.

$W_n$  — вектор правых частей у нелинейных источников.

В модельном методе цепь (рис. 4) является  $Q-R-X-N$  цепью и описывается системой уравнений вида [3]:

$$\frac{d}{dt} [\Phi_x \cdot V] = W_x \cdot V;$$

$$\Phi_N \cdot V = W_n;$$

$$\Phi_Q \cdot V = W_q;$$

$$\Phi_R \cdot V = 0;$$

$$\Phi_J \cdot V = 0.$$

(16)

Для установления связи между уравнениями (16) и (14), (15) исходная система уравнений, полученная с помощью модельного метода, перестраивается. В ней дифференциальные уравнения инерционных источников и нелинейные алгебраические уравнения нелинейных источников заменяются линейными алгебраическими с дополнительными переменными. Замена производится таким образом, что уравнения источников тока заменяются уравнениями связи для их аргументов (напряжения на источниках тока), а уравнения источников напряжения — уравнениями связи для функций (напряжения на источниках напряжения). В результате образуется однородная линейная система алгебраических уравнений, которую легко разрешить относительно зависимых переменных (функций) источников  $V_f$ , выражая их через переменные состояния

$$V_f = F(V_x, V_q, V_n), \quad (17)$$



где  $F$  — линейная функция. Найденные переменные и представляют собой правые части уравнений состояния, то есть инерционных и нелинейных источников.

Ниже предлагается формальное изложение процедуры перехода от уравнений модельного метода к уравнениям состояния.

Процесс перехода от уравнений (16) к уравнениям переменных состояния складывается из следующих этапов.

Этап I. Поскольку модельный метод оперирует узловыми напряжениями, а метод переменных состояния — напряжениями ветвей, то на данном этапе вводятся дополнительные переменные  $V_L, V_C, V_n, V_{nE}, V_E$  — напряжения ветвей источников. Каждая из этих переменных удовлетворяет равенству:

$$\Pi_j U - V_j = 0, \quad (18)$$

в котором  $\Pi_j$  — матрица координат узлов для подцепи из источников типа  $j$ ,  $V_j$  — вектор дополнительных переменных для этих же источников. Введение дополнительных переменных расширяет координатный базис схемы, соответственно увеличивается размерность основной матрицы  $\Phi$ .

Этап II. На этом этапе выделяются независимые переменные источников:  $V_x, V_q, V_n$ , то есть те, которые использует метод переменных состояния.

Здесь же выделяются зависимые переменные инерционных нелинейных источников тока и напряжения.

$$V_p = \begin{bmatrix} J_C \\ V_L \end{bmatrix}, \quad V_m = \begin{bmatrix} J_{nI} \\ V_{nE} \end{bmatrix},$$

где  $W_L$  — напряжения индуктивных ветвей, а  $V_{nE}$  — напряжения ветвей источников напряжения.

Нужные токи выделяются с помощью  $\Gamma$ -матриц, матриц координат ветвей (токов), а нужные напряжения с помощью  $\Pi$ -матриц и заменяются дополнительными переменными на основании равенств вида (18).

Этап III. На этом этапе производится преобразование исходной системы уравнений (16) в систему определяющих уравнений и систему связывающих соотношений. Для этого перепишем систему уравнений (16) в следующем виде:

$$\frac{d}{dt} [\Phi_x \cdot V] = W_x \cdot V;$$

$$W_n = \Phi_n \cdot V;$$

$$W_q = \Phi_q \cdot V;$$

$$0 = \Phi_R \cdot V;$$

$$0 = \Phi_j \cdot V.$$

Вычтем далее из левой и правой части уравнений, описывающих инерционные, нелинейные и независимые источники, соответствующие им зависимые переменные  $V_n$ ,  $V_m$ ,  $V_q$ . В результате образуется две системы уравнений. Первая из них имеет вид:

$$\begin{aligned}\frac{d}{dt}[\Phi_x \cdot V] &= V_p; \\ W_n &= V_n; \\ W_q &= V_q\end{aligned}\tag{19}$$

и будет в дальнейшем называться системой определяющих уравнений, поскольку эти уравнения определяют состояние источников в цепи.

Вторая система уравнений записывается так:

$$\begin{aligned}0 &= W_x \cdot V - V_p; \\ 0 &= \Phi_N \cdot V - V_m; \\ 0 &= \Phi_Q \cdot V - V_q; \\ 0 &= \Phi_R \cdot V; \\ 0 &= \Phi_J \cdot V.\end{aligned}$$

Ее необходимо дополнить уравнениями связи для независимых переменных, инерционных и нелинейных источников тока и напряжения:

$$\begin{aligned}0 &= \Phi_x \cdot V - V_x; \\ 0 &= \Phi_n \cdot V - V_n,\end{aligned}$$

где матрица  $\Phi_n$  имеет вид

$$\Phi_n = \begin{vmatrix} 0 & \Pi_l \\ \Gamma_E & 0 \end{vmatrix}.$$

Здесь  $\Pi_l$  — матрица координат узлов у источников тока,  $\Gamma_E$  — матрица координат ветвей у источников напряжения.

В итоге получаем систему связывающих соотношений, которая представляет собой однородную систему линейных алгебраических уравнений, содержащих все переменные заданной цепи, включая и дополнительные. Совместно с системой уравнений (19) она определяет решение цепи. Система связывающих соотношений содержит  $n-l$  уравнений, где



$n$  — общее число переменных в системе, а  $l$  — число уравнений источников.

Этап IV. Решаем линейную систему алгебраических уравнений, исключая в ней все переменные, кроме переменных состояния  $V_x$ ,  $V_q$  и  $V_n$ , а также  $V_p$  и  $V_m$ . После исключения эта система принимает вид

$$|\Phi_s| \times \begin{vmatrix} V_p \\ V_m \\ V_x \\ V_n \\ V_q \end{vmatrix} = 0, \quad (20)$$

где  $\Phi_s$  — матрица линейных форм при оставшихся после исключения переменных.

Теперь для получения первого уравнения из (15) достаточно в уравнении (20) исключить переменные  $V_m$  и разрешить его относительно  $V_p$ . При этом оно принимает следующий вид:

$$V_p = A \cdot V_x + B V_q + B' V_n.$$

Объединяя полученное уравнение с первым уравнением из (19), получаем

$$\frac{d}{dt} [\Phi_x \cdot V] = A V_x + B V_q + B' V_n,$$

то есть искомое уравнение из метода переменных состояния. Для получения второго уравнения из (15) исключаем в уравнении (20) переменные  $V_p$  и разрешаем его относительно вектора переменных  $V_m$ :

$$V_m = C V_x + D V_q + D' V_n.$$

Полученное уравнение объединяем со вторым уравнением из (19), в результате чего получаем уравнение

$$W_n = C V_x + D V_q + D' V_n,$$

которое также является уравнением метода переменных состояния.

Решение уравнения (20) относительно переменных  $V_p$  и  $V_m$  производится нормализацией строк матрицы относительно коэффициентов при этих переменных и последующим разделением общего вектора переменных на переменные состояния  $V_x$ ,  $V_q$  и  $V_n$ . Матрицы линейных форм, соответ-

ствующие переменным состояниям, образуются выделением соответствующих столбцов.

Заметим, что с помощью описанной методики могут быть получены уравнения состояния как нелинейной, так и линейной цепи. Во втором случае в рассмотрение не входят нелинейные источники и связанные с ними переменные.

Для пояснения предложенной методики рассмотрим примеры линейной и нелинейной цепи.

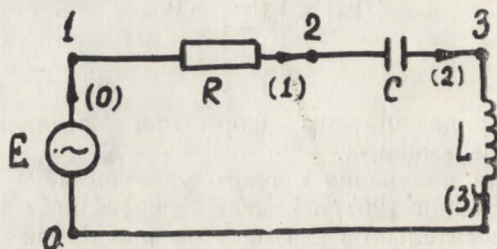


Рис. 5

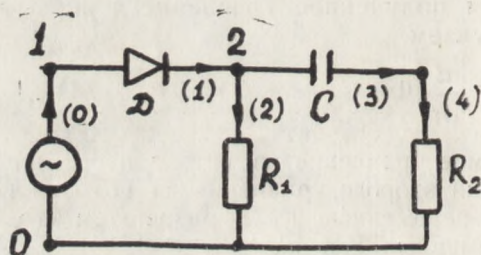


Рис. 6

На рис. 5 приведена линейная  $RLC$ -цепь. Ее модель в модельном методе записывается так:

$$C \frac{d(U_2 - U_3)}{dt} = J_2,$$

$$L \frac{dJ_3}{dt} = U_3,$$

$$U_1 = E,$$



$$\begin{aligned}U_1 - U_2 - J_1 R &= 0, \\J_0 - J_1 &= 0, \\J_1 - J_2 &= 0, \\J_2 - J_3 &= 0.\end{aligned}$$

Выделяем в построенной системе уравнений переменные состояния. Для этого вводим предварительно дополнительные переменные — напряжения ветвей источников:

$$U_1 - V_1 = 0, \quad U_2 - U_3 - V_2 = 0, \quad U_3 - V_3 = 0.$$

Независимыми переменными источников являются здесь следующие переменные:

$$V_x = \left| \frac{V_2}{J_3} \right|, \quad V_q = V_1.$$

Перестраиваем исходную систему уравнений, разделяя ее на определяющие уравнения и уравнения связи.

Определяющие уравнения:                      Уравнения связи:

$$\begin{aligned}C \frac{d}{dt} (U_2 - U_3) &= J_2, & U_3 - V_3 &= 0; \\L \frac{dJ_3}{dt} &= V_3, & U_2 - U_3 - V_2 &= 0; \\E &= V_1, & U_1 - V_1 &= 0; \\& & U_1 - U_2 - J_1 R &= 0; \\& & J_0 - J_1 &= 0; \\& & J_1 - J_2 &= 0; \\& & J_2 - J_3 &= 0.\end{aligned}$$

Исключая в системе уравнений связи все переменные, кроме  $V_q$ ,  $V_x$  и  $V_p$ , и разрешая их относительно  $V_p$ , получим

$$\begin{aligned}J_2 - J_3 &= 0, \\V_3 &= V_1 - V_2 - J_3 R.\end{aligned}$$

Подставляя найденные значения в определяющие уравнения, получаем

$$\frac{d}{dt} \left| \frac{U_2 - U_3}{J_3} \right| = \begin{vmatrix} 0 & \frac{1}{C} \\ -\frac{1}{L} & -\frac{R}{L} \end{vmatrix} \cdot \left| \frac{V_2}{J_3} \right| + \begin{vmatrix} 0 \\ \frac{1}{L} \end{vmatrix} E,$$

что соответствует форме уравнения (14).

На рис. 6 приведена нелинейная цепь переменного тока. Уравнения модели этой цепи имеют вид:

$$C \frac{d(U_2 - U_3)}{dt} = J_3;$$

$$J_1 = I_T [\exp \Theta (U_1 - U_2) - 1],$$

$$U_0 - U_1 = E,$$

$$U_2 - J_2 R_1 = 0,$$

$$U_3 - J_4 R_2 = 0,$$

$$J_0 - J_1 = 0,$$

$$J_1 - J_2 - J_3 = 0,$$

$$J_3 - J_4 = 0.$$

Вводим дополнительные переменные:

$$U_1 - V_1 = 0, \quad U_1 - U_2 - V_2 = 0, \quad U_2 - U_3 - V_3 = 0.$$

Выделяем независимые переменные источников:

$$V_x = V_3, \quad V_g = V_1, \quad V_n = V_2,$$

а также зависимые переменные этих источников

$$V_p = I_3, \quad V_m = I_1.$$

После этого перестраиваем исходную систему.

Определяющие уравнения:

$$C \frac{d(U_2 - U_3)}{dt} = J_3,$$

$$I_T [\exp (\Theta (U_1 - U_2) - 1)] = J_1;$$

$$E = V_1.$$

Уравнения связи:

$$U_2 - U_3 - V_3 = 0,$$

$$U_1 - U_2 - V_2 = 0,$$

$$U_1 - V_1 = 0,$$

$$U_2 - J_2 R_1 = 0,$$

$$U_3 - J_4 R_2 = 0,$$

$$J_0 - J_1 = 0,$$

$$J_1 - J_2 - J_3 = 0,$$

$$J_3 - J_4 = 0.$$

Исключая из системы уравнений связи все переменные, кроме выделенных переменных источников, получаем:

$$V_1 - V_2 - I_1 R_1 + I_3 R_1 = 0,$$

$$-V_3 + I_1 R_1 - I_3 (R_1 + R_2) = 0.$$



Из образованной системы исключаем зависимые переменные нелинейных источников ( $J_1$ ) и разрешаем ее относительно переменной  $V_p = J_3$ .

$$J_3 = \frac{V_3 + V_1 - V_2}{R_2}.$$

Аналогичным способом разрешаем эту систему относительно переменной  $V_m = J_1$ :

$$J_1 = -\frac{1}{R_2} V_3 + \frac{R_1 + R_2}{R_1 R_2} V_1 - \frac{R_1 + R_2}{R_1 R_2} V_2.$$

Подставляя найденные значения в определяющие уравнения, получим уравнения переменных состояния цепи (рис. 6):

$$\frac{d(U_2 - U_3)}{dt} = -\frac{1}{R_2} V_3 + \frac{1}{R_2} V_1 - \frac{1}{R_2} V_2;$$

$$I_T [\exp \Theta (U_1 - U_2) - 1] = -\frac{1}{R_2} V_3 + \\ + \frac{R_1 + R_2}{R_1 R_2} V_1 - \frac{R_1 + R_2}{R_1 R_2} V_2.$$

Заметим, что в приведенных примерах можно было использовать и матричную запись модели цепи и всех промежуточных преобразований, однако ввиду разряженности матриц такие преобразования становятся громоздкими и малонаглядными.

Данная методика пригодна и для цепей с зависимыми источниками, которые описываются однородными линейными алгебраическими уравнениями. Эти уравнения входят в систему уравнений связи и участвуют в дальнейших преобразованиях.

### § 3. Построение передаточной функции из уравнений модельного метода

Здесь, очевидно, могут быть построены различные алгоритмы, позволяющие от уравнений модельного метода перейти к уравнению вида

$$V_S = \frac{W_1(p)}{W_2(p)} \cdot V_q,$$

где дробно-рациональное выражение  $W_1(p)/W_2(p)$  представляет собой передаточную функцию линейной цепи.

Один из возможных алгоритмов построения передаточной функции состоит в следующем. Положим, что от уравнений модельного метода с помощью описанной выше процедуры совершен переход к уравнениям переменных состояния:

$$\begin{aligned}\frac{d}{dt} V_x &= AV_x + BV_q; \\ V_s &= CV_x + DV_q.\end{aligned}\tag{21}$$

Здесь  $V_s$  — вектор выходных переменных цепи, который удовлетворяет соотношению

$$P_s U - V_s = 0.$$

Это соотношение вводится в систему связывающих соотношений, которая решается относительно переменных  $V_s$ . Выражая эти переменные через переменные состояния, получаем второе уравнение из (21).

Применим преобразование Лапласа для уравнений (21), в результате получим:

$$\begin{aligned}pV_x(p) - V_x(0) &= AV_x(p) + BV_q(p); \\ V_s(p) &= CV_x(p) + DV_q(p).\end{aligned}$$

Выражая  $V_x(p)$  из первого уравнения и подставляя во второе, определяем выходной вектор

$$V_s(p) = C\Phi(p)V_x(0) + [C\Phi(p) \cdot B + D] \cdot V_q(p), \quad [(22)$$

где  $\Phi(p) = [pU - A]^{-1}$ ,  $U$  — единичная матрица.

Выходной сигнал является суперпозицией характеристик, обусловленных начальными условиями ( $V_x(0)$ ) и входными сигналами ( $V_q(p)$ ).

Таким образом, для вычисления передаточной функции достаточно получить матрицы  $A$ ,  $B$ ,  $C$ ,  $D$ , а также  $\Phi(p) = [pU - A]^{-1}$ . Эффективный алгоритм вычисления этой матрицы рассмотрен в [2].

Рассмотрим пример получения передаточной функции схемы (рис. 5). Выходной переменной в этой схеме является напряжение  $U_3$ . Вводя дополнительную переменную для вы-



ходной переменной  $V_3 = U_3$  и выражая ее через переменные состояния  $V_1, V_2, J_3$ , можем записать

$$V_3 = |-1 - R| \left| \frac{V_2}{J_3} \right| + |1| \cdot V_1.$$

Таким образом, получены все необходимые матрицы метода переменных состояния для этой схемы:

$$A = \begin{bmatrix} 0 & \frac{1}{C} \\ -\frac{1}{L} & -\frac{1}{R} \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix}; \quad C = |-1 - R|; \quad D = |1|,$$

теперь необходимо вычислить матрицу  $[pU - A]^{-1} = \Phi(p)$ .

$$\Phi(p) = \frac{1}{\Delta(p)} \begin{bmatrix} p + \frac{R}{L} & \frac{1}{C} \\ -\frac{1}{L} & p \end{bmatrix},$$

где  $\Delta(p) = p^2 + \frac{R}{L} \cdot p + \frac{1}{LC}.$

Полагая начальные условия для этой схемы нулевыми, то есть  $V_x(0) = 0$ , подставим найденные матрицы в уравнение (22). В результате получим

$$V_3 = \left\{ |-1 - R| \cdot \frac{1}{\Delta} \begin{bmatrix} p + \frac{R}{L} & \frac{1}{C} \\ -\frac{1}{L} & p \end{bmatrix} \cdot \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} + |1| \right\} V_1;$$

или после преобразования

$$V_3 = \frac{p^2 LC}{p^2 LC + pRC + 1} V_1.$$

Дробно-рациональное выражение в правой части последнего уравнения и представляет собой передаточную функцию схемы.

#### ЛИТЕРАТУРА

1. Брэнин Ф. Методы анализа цепей с помощью ЭЦВМ.— ТИИЭР, т. 55, № 11, 1967.
2. Ильин В. И. Машинное проектирование электронных схем. М., «Энергия», 1972.
3. Арайс Е. А., Дмитриев В. М. Матричная интерпретация модельного метода анализа цепей.— В сб.: Вопросы программирования, вып. II, Томск, Изд-во ТГУ, 1972.

## ПРИМЕНЕНИЕ МОДЕЛЬНОГО МЕТОДА К АНАЛИЗУ ЦЕПЕЙ С РАСПРЕДЕЛЕННЫМИ ПАРАМЕТРАМИ

Е. А. АРАЙС, В. Ф. ШАДРИН

### Введение

Автоматизация анализа и проектирования различного рода цепей — актуальная задача, выдвигаемая практикой. Система МАРС позволяет проводить анализ радиоэлектронных цепей с сосредоточенными параметрами [1]. Анализ основывается на модельном подходе к рассматриваемой цепи. Ниже решается задача распространения модельного метода на цепи с распределенными параметрами. Примерами таких цепей являются гидравлические цепи, системы магистральных нефтегазопроводов, большие энергосистемы, интегральные схемы, устройства СВЧ-электроники и некоторые другие.

Математически задача сводится к решению системы дифференциальных уравнений в частных и обыкновенных производных, алгебраических уравнений на ориентированных графах. В системном подходе к анализу цепей одним из основных моментов является определение компонентов цепи и построение математических моделей компонента. В общем случае такой моделью для компонента с распределенными параметрами (КРП) будет система дифференциальных уравнений в частных производных, с заданными начальными условиями на компоненте, граничными условиями и условиями сопряжения в узлах цепи.

В качестве условий сопряжения выступают законы сохранения, характерные для рассматриваемой цепи. Например, в энергетических системах — это закон Кирхгофа, а в гидравлических цепях — законы сохранения расхода, массы, теплоты. Граничные условия моделируются отдельно, представляя собой самостоятельный компонент цепи. Примером такого компонента будет источник напряжения, если



воспользоваться терминологией электрических цепей с сосредоточенными параметрами. Таким образом, начальные условия, условия сопряжения и заданные источники определяют существование и единственность решения системы дифференциальных уравнений в частных производных на ориентированных графах.

Решение поставленной задачи, даже в простейших случаях, возможно лишь численными методами. Естественно для создания универсального алгоритма использовать разностные методы. Тогда КРП моделируется системой линейных или нелинейных алгебраических уравнений, а это, в свою очередь, вызывает усложнение в машинной реализации модели, ведет к нежелательному увеличению памяти. Возникают и трудности вычислительного характера, связанные с необходимостью решать систему алгебраических уравнений высокого порядка, даже в случае, если цепь состоит только из нескольких КРП. Методы, разработанные специально для решения задач на ориентированных графах, позволяют создать достаточно экономичные и в вычислительном отношении устойчивые алгоритмы [2—6]. Применение варианта метода прогонки дает возможность систему разностных уравнений, моделирующих компоненты, свести к системе уравнений значительно меньшего порядка и связывающей только узлы графа, такая модель КРП позволяет эффективно применить операторы системы МАРС для определения решения в узловых точках. На следующем этапе находится решение во внутренних точках компонента.

## § 1. Построение математической модели КРП

Математической моделью компонента с сосредоточенными параметрами будет некоторая система уравнений относительно значений неизвестной функции только в узлах графа цепи [1]. Для КРП такую модель записать сразу нельзя. В КРП значение неизвестной функции меняется непрерывным образом от узла к узлу. Поэтому подход к построению модели КРП должен иметь свои особенности. Чтобы выявить эти особенности и наметить путь построения модели, рассмотрим простой пример. Возьмем в качестве цепи с распределенными параметрами (ЦРП) систему диффузных резисторов (или теплопроводных стержней) с источником напряжения. На рис. 1 изображен граф этой цепи.

За математическую модель резистора примем линейное, одномерное и однородное параболическое уравнение:

$$\frac{\partial V}{\partial t} = \kappa \frac{\partial^2 V}{\partial x^2}, \quad (1)$$

где  $V$  — напряжение;  $x$  — координата;  $t$  — время;  $\kappa$  — некоторый коэффициент, принятый нами за постоянный. Условия

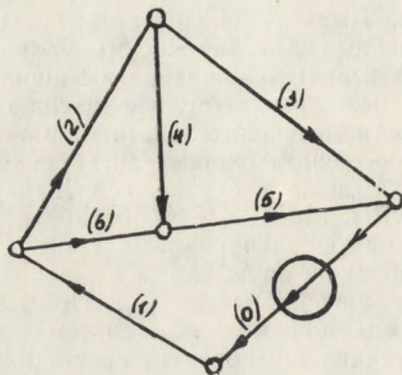


Рис. 1

сопряжения представляют собой закон Кирхгофа, который в случае  $p$  компонент, примыкающих к узлу, запишется в виде:

$$\sum_{j=1}^p J_j = 0, \quad (2)$$

где  $I_j$  — ток в  $j$ -той компоненте.

Учитывая закон Ома, уравнение (2) можно записать через напряжение:

$$\sum_{j=1}^p c_j \frac{\partial V_j}{\partial x} = 0, \quad (3)$$

где  $C_j$  — известные коэффициенты.

Аппроксимируем уравнение (1) с помощью обычной неявной разностной схемы. На каждой ветви графа получим систему разностных уравнений вида:

$$av_{n-1} + bv_n + av_{n+1} = d_n, \quad n = 1, 2, \dots, N,$$



где  $v$  — сеточный аналог функции  $V$ ;  $N$  — число внутренних точек на ветви графа, тогда  $V_0, V_{N+1}$  — значения функции, соответствующие начальному и конечному узлу ориентированной ветви графа.

$$a = \frac{T\kappa}{H^2}, \quad b = -(1 + 2a),$$

$d_n$  — значение функции на предыдущем слое;  
 $T$  — шаг по времени;  
 $H$  — шаг по координате.

Используя метод, предложенный в [6], получим для КРП математическую модель из двух линейных алгебраических уравнений относительно неизвестных значений функции в узловых и ближайших к ним точках компоненты:

$$\begin{aligned} v_1 &= x_N v_{N+1} + y_N v_0 + z_N; \\ v_N &= x_N v_{N+1} + y_N v_0 + z_N, \end{aligned} \quad (5)$$

где переменные, в общем случае, коэффициенты при неизвестных и свободные члены находятся по рекуррентным формулам:

$$\begin{aligned} X_n &= -\frac{a}{b + aX_{n-1}}, \quad x_n = x_{n-1} X_n, \\ Y_n &= -\frac{aY_{n-1}}{b + aX_{n-1}}, \quad y_n = y_{n-1} + x_{n-1} Y_n, \\ Z_n &= \frac{d_n - aZ_{n-1}}{b + aX_{n-1}}, \quad z_n = z_{n-1} + x_{n-1} Z_n, \\ n &= 1, 2, \dots, N. \end{aligned} \quad (6)$$

$$X_0 = Z_0 = y_0 = z_0 = 0, \quad Y_0 = x_0 = 1.$$

Ток  $I$  на концах компоненты приближенно запишем

$$\begin{aligned} J_0 &= c_1(v_1 - v_0), \\ J_{N+1} &= c_1(v_{N+1} - v_N), \end{aligned} \quad (7)$$

где  $c_1$  — известный коэффициент.

Подставим  $v_1, v_N$  из (5) в (7) и получим:

$$\begin{aligned} s_{11} v_0 + s_{12} v_{N+1} - J_0 &= q_1, \\ s_{21} v_0 + s_{22} v_{N+1} + J_{N+1} &= q_2, \end{aligned} \quad (8)$$

где

$$\begin{aligned} s_{11} &= c_1 (y_N - 1), & s_{12} &= c_1 x_N, \\ s_{21} &= c_1 Y_N, & s_{22} &= c_1 (X_N - 1), \\ q_1 &= -c_1 z_N, & q_2 &= -c_1 Z_N. \end{aligned} \quad (9)$$

Уравнения (8) соответствуют определению математической модели компонента цепи, принятому в системе МАРС. Уравнение источника возьмем в виде

$$v = v(t) \quad (10)$$

Тогда уравнения (3), (8), (10) составят замкнутую систему уравнений, решая которую можно найти значения тока и напряжения в узлах графа цепи. Значения напряжений во внутренних точках компонента легко определить обратной прогонкой:

$$v_n = X_n v_{n+1} + Y_n v_0 + Z_n, \quad n = N, N-1, \dots, 1. \quad (11)$$

Устойчивость используемой разностной схемы исследована в [6]. Относительно точности получаемого решения заметим, что она зависит от порядка аппроксимации дифференциальных операторов и равна в нашем случае  $o(T+H)$ . Для графа в целом точность решения будет определяться точностью решения на компоненте с наихудшей аппроксимацией.

Из рассмотренного простого примера можно сделать важные для построения модели КРП выводы.

Уравнения в частных производных, описывающих КРП, нужно так аппроксимировать в каждой внутренней точке компонента линейными алгебраическими уравнениями, чтобы было возможно получение уравнений (5). Для линейных одномерных уравнений параболического, гиперболического и эллиптического типа получение уравнений (5) не вызывает особых затруднений. Причем случай эллиптического уравнения модели сводится к параболическому, если применить для получения решения метод установления.

Не рассматривая детально случай плоского или объемного компонента, заметим лишь, что и для него можно получить уравнения, подобные уравнениям (5), если воспользоваться схемами расщепления исходных уравнений.

Использование условия сопряжения позволило нам в примере получить замкнутую систему уравнений. В иных случаях потребуются конструирование дополнительных граничных соотношений [2], [3], [4]. Эти соотношения можно получить, например, аппроксимируя исходное уравнение на границе [7].



Предыдущие выводы справедливы и в случае квазилинейной математической модели КРП. Нелинейные же исходные уравнения КРП приводят к нелинейным алгебраическим уравнениям во внутренних точках. Общего алгоритма получения уравнений модели КРП, аналогичных уравнениям (8), указать нельзя.

## § 2. Реализация модели КРП и организация анализа ЦРП в системе МАРС

Машинную реализацию модели КРП рассмотрим на примере линейного одномерного параболического уравнения. Ниже представлены случаи задания исходного уравнения:

1.  $\frac{\partial V}{\partial t} = \kappa \frac{\partial^2 V}{\partial x^2},$
2.  $\frac{\partial V}{\partial t} = \kappa \frac{\partial^2 V}{\partial x^2} + f(x, t),$
3.  $\frac{\partial V}{\partial t} = \frac{\partial}{\partial x} \left( \kappa(x, t) \frac{\partial V}{\partial x} \right),$
4.  $\frac{\partial V}{\partial t} = \frac{\partial}{\partial x} \left( \kappa(x, t) \frac{\partial V}{\partial x} \right) + f(x, t).$

Каждому типу задания исходного уравнения присвоен идентификатор. Модель оформлена как стандартная подпрограмма, к которой происходит обращение из основной программы анализа цепи. Например:

ЛПУ2 (T, H, c, l, N, K, УЧ),

что означает линейное параболическое уравнение тип 2. В скобках перечислены задаваемые параметры: l — длина компонента; УЧ — условное число, характеризующее функцию  $f(x, t)$ ; функции  $K(x, t)$ ,  $f(x, t)$  задаются таблично пользователем и оформляются как подпрограммы в модели. Учитывая, что получение уравнений (9) содержит однотипные вычисления для всех компонент, вычисление  $a, b, x_N, y_N, z_N, X_N, Y_N, Z_N$  оформлено как отдельный оператор, обращение к которому происходит из модели на каждом шаге по времени. Заметим, что хотя машинная реализация математической модели зависит от метода, избираемого для нахождения решения цепи, структура модели неизменна и состоит из информационной строки, информационного массива

и вычислительного блока. В информационную строку заносятся длина информационного массива и длина всей модели, исключая информационную строку. В некоторых ячейках информационного массива находятся адреса, указывающие на начало отдельных числовых массивов и подпрограмм в модели. В вычислительном блоке находятся обращение к оператору вычисления коэффициентов и подпрограммы вычисления функций  $\kappa(x, t)$ ,  $f(x, t)$ , начальное значение  $V(x, 0)$ .

Опишем кратко организацию анализа ЦРП в системе МАРС. Разработчиком устанавливается и ориентируется граф схемы. На входном языке анализа и проектирования цепей пишется информация о цепи [1]. Транслятор переводит информацию о цепи во внутренний язык системы. Модели КРП вызываются в МОЗУ. На первом этапе происходит обращение к оператору вычисления коэффициентов модели, которые после вычисления заносятся в модель. Вычисляются функции  $\kappa(x, t)$ ,  $f(x, t)$ . На втором этапе решается система уравнений относительно значений искомых функций в узлах цепи. Используются операторы статического или динамического анализа из библиотеки операторов системы МАРС. Затем происходит возврат на первый этап.

### Заключение

Отметим в заключение некоторые выводы по данной работе и преимущества системного подхода к анализу ЦРП. В работе устанавливается возможность реализации анализа ЦРП в системе МАРС. На примере показано применение модельного подхода к анализу. Строится математическая модель КРП и показана ее машинная реализация.

Основными достоинствами системного подхода являются, во-первых, быстрота и эффективность анализа при наличии развитой библиотеки операторов и моделей, во-вторых, использование математических аналогий позволяет вести анализ цепей, различных по своей физической природе и содержащих компоненты как с распределенными, так и с сосредоточенными параметрами.

Авторы благодарны А. Ф. Воеводину за обсуждение задачи.

### ЛИТЕРАТУРА

1. Вопросы программирования и автоматизации проектирования. Вып. 1, Вып. 2, Томск, Изд-во Томского ун-та, 1971, 1972.



2. Шугрин С. М. Численный расчет неустановившегося движения воды в системах речных русел или каналов.— «Изв. СО АН СССР, Сер. техн. наук», 1969, № 3, вып. 1, 25—31.

3. Воеводин А. Ф. О применении метода прогонки для численного интегрирования гиперболических систем дифференциальных уравнений, определенных на комплексе.— В сб.: Динамика сплошной среды, вып. III. Новосибирск, Ин-т Гидродинамики СО АН СССР, 1969, 91—110.

4. Рябенский В. С. Расчет теплопроводности на системе стержней.— Ж. вычисл. матем. и матем. физ., 1970, 10, № 1, 236—239.

5. Фрязинов И. В. Алгоритм решения разностных задач на графах.— Ж. вычисл. матем. и матем. физ., 1970, 10, № 2.

6. Воеводин А. Ф. Метод прогонки для разностных уравнений, определенных на комплексе.— Ж. вычисл. матем. и матем. физ., 1973, 13, № 2, 494—497.

7. Алалыкин Г. Б., Годунов С. К., Киреева И. Л., Плинер Л. А. Решение одномерных задач газовой динамики в подвижных сетках. М., «Наука», 1970.

---

## ИССЛЕДОВАНИЕ ЛИНЕЙНЫХ ЦЕПЕЙ В СИСТЕМЕ МАРС

Е. А. АРАЙС, Л. А. ШАПИРО

При анализе и моделировании радиоэлектронных схем разработчику необходимо знать конкретные значения параметров схемы. Для решения задачи нахождения значений параметров схемы разработан модельный метод анализа цепей [1], в котором процесс нахождения параметров цепи сводится к решению модели цепи.

Общим моментом при исследовании любых цепей является построение модели цепи, состоящей из  $\Phi$ -матрицы и вектора  $W$ . Рассмотрим цепь с  $N$  узлами (с номерами  $0, 1, \dots, N-1$ ) и  $M$  ветвями (с номерами  $0, 1, \dots, M-1$ ). Полагаем, что каждый вывод многополюсника образует ветвь (за исключением двухполюсника, образующего одну ветвь между узлами, к которым он подключен). Каждой ветви соотносится произвольное направление тока в ней.

Естественной формой хранения информации о схеме в памяти ЭВМ является таблица, в которой перечислены названия элементов, номера узлов, в которые ведут ветви. Эта таблица может иметь вид (индексы записываются в квадратных скобках):

[illegible]

Здесь  $M_1, \dots, M_K$  — названия компонентов схемы, имеющих соответственно  $n_1, \dots, n_K$  выводов. Индекс у  $B$  задает номер ветви, образованной выводом компонента; индекс у  $N$  задает



номер узла, в который ведет ветвь. Знак плюс соответствует ветви, ток в которой направлен в компоненту.

В модельном методе анализ линейных цепей постоянного тока и линейных цепей переменного тока в частной форме осуществляется решением системы линейных алгебраических уравнений  $\Phi \times V = W$ .  $\Phi$ -матрица размерности  $(M + N) \times (M + N)$  расписывается из коэффициентов линейных форм в левых частях уравнений моделей компонентов, составляющих исследуемую схему. Каждому уравнению модели соответствует строка  $\Phi$ -матрицы. Первые  $N$  элементов каждой строки соответствуют узловым напряжениям, последующие  $M$  — токам в ветвях. В вектор  $W$  заносятся соответствующие правые части уравнений. Кроме того, в  $\Phi$ -матрицу и вектор  $W$  включаются коэффициенты линейных алгебраических однородных уравнений, задающих закон Кирхгофа для всех узлов, кроме нулевого, а также коэффициенты уравнения  $U_0 = 0$ , задающего точку отсчета для напряжений. Буквой  $V$  в дальнейшем будем обозначать вектор-столбец

$$\begin{matrix} U \\ J \end{matrix} \text{ с компонентами } (V_0, V_1, \dots, V_{N-1}, V_N, \dots, V_{N+M-1}).$$

Основными этапами работы алгоритма анализа линейных цепей постоянного тока и линейных цепей переменного тока в частотной форме является роспись уравнений моделей, законов Кирхгофа и решение построенной системы уравнений.

Каждому компоненту цепи соответствует математическая модель. Линейная модель для  $n_i$ -полюсника задается системой  $n_i$  линейных алгебраических уравнений. Потребуем, чтобы в уравнениях модели переменные были упорядочены в соответствии с нумерацией выводов компонента. Напряжения должны предшествовать токам. Линейная модель хранится в виде матрицы коэффициентов линейных алгебраических уравнений модели. Будем расписывать матрицу в памяти ЭВМ построчно вместе с правыми частями, используя сплошную нумерацию элементов матрицы. Обозначив общий член буквой  $a$ , получим:

$$\begin{array}{ccccccc} a[0] & a[1] & \cdots & a[2n_i - 1] & a[2n_i] & & \\ a[2n_i + 1] & a[2n_i + 2] & \cdots & a[4n_i] & a[4n_i + 1] & & \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a[2n_i \times n_i - n_i - 1] & a[2n_i \times n_i - n_i] & \cdots & a[2n_i \times n_i + n_i - 2] & & & \\ & & & a[2n_i \times n_i + n_i - 1] & & & \end{array}$$

Роспись уравнений модели цепи заключается в переносе коэффициентов из матриц, задающих модели компонентов, в матрицу, задающую модель всей цепи. Размерность матрицы, задающей модель цепи (т. е.  $\Phi$ -матрицы), равна  $L \times L$ , где  $L = N + M$  — суммарное количество ветвей и узлов в цепи. Нумерацию элементов  $\Phi$ -матрицы и вектора  $W$  будем начинать с нуля.

Распишем алгоритм построения  $\Phi$ -матрицы и вектора  $W$  по шагам, используя для обозначения индекса в названиях компонентов цепи букву  $r$ ; для обозначения нижнего индекса в номерах ветвей и узлов букву  $q$ ; букву  $p$  — для элементов матриц моделей. Компоненты  $\Phi$ -матрицы будем обозначать буквой  $\varphi$ , а компоненты вектора  $W$  — буквой  $\omega$ . Индекс при  $\varphi$  обозначим буквой  $s$ , при  $\omega$  — буквой  $t$ .

Шаг 1. Устанавливаем начальные значения индексов для рассылки коэффициентов в  $\Phi$ -матрицу и вектор  $W$ :

$$s := 0; t := 0.$$

Шаг 2. Устанавливаем начальное значение индекса для названий компонентов цепи:

$$r := 1.$$

Шаг 3. Приступаем к рассылке коэффициентов модели компонента в модель цепи. Рассматриваем компонент  $M_r$ . Устанавливаем начальное значение индекса для коэффициентов модели:

$$p := 0.$$

Шаг 4. Начало цикла просмотра уравнений модели (строк матрицы, задающей модель). Устанавливаем начальное значение индекса для номеров ветвей и узлов, указанных в информации о схеме для компонента  $M_r$ :

$$q := 1.$$

Шаг 5. Рассылаем коэффициенты из модели в  $\Phi$ -матрицу:

$$\varphi[s + j'_q] := a^r[p];$$

$$\varphi[s + i'_q + Q] := Q^r[p + n_r].$$

Здесь буквой  $a$  с верхним индексом  $r$  обозначен коэффициент модели, соответствующий компоненту  $M_r$ . Первые  $n_r$  коэффициентов для рассматриваемого  $n_r$ -полюсника соответствуют напряжениям на выводах. В  $\Phi$ -матрице напряжениям соответствуют первые  $Q$  (число узлов) столбцов.



Прочие столбцы задают коэффициенты при токах в цепи. Значения величин  $j_q^r$  и  $l_q^r$  выбираются из информации о цепи.

Шаг 6. Переход к пересылке следующего коэффициента модели:

$$p := p + 1;$$

$$q := q + 1.$$

Переходим к шагу 5, если  $q < n_r + 1$ . В противном случае выполняется шаг 7.

Шаг 7. Закончено формирование очередной строки  $\Phi$ -матрицы. Последний элемент строки матрицы модели следует переслать в вектор  $W$  со своим знаком.

$$\omega[t] := a^r[p].$$

Устанавливаем индекс  $p$  на начало следующей строки в матрице модели. Индекс  $s$  устанавливается на начало следующей строки в  $\Phi$ -матрице. Индекс  $t$  подготавливается для записи очередного коэффициента в  $W$ .

$$p := p + 1; s := s + L; t := t + 1.$$

Возвращаемся к шагу 4 ( $n_r - 1$ ) раз. Затем шаг 8.

Шаг 8. Закончена пересылка коэффициентов модели очередного компонента цепи  $M_r$ . Устанавливаем индекс  $r$  для следующего компонента цепи

$$r := r + 1.$$

Возвращаемся к шагу 3  $\kappa$  раз (т. е. переходим к шагу 3, если  $r < \kappa + 1$ ), после чего алгоритм росписи уравнений моделей работу заканчивает. Далее распишем алгоритм росписи уравнений законов Кирхгофа для токов и уравнения, задающего точку отсчета для напряжений  $U_0 = 0$ .

Шаг 9. Засылаем в  $\Phi$  и  $W$  коэффициенты уравнения  $U_0 = 0$ . Перестраиваем индексы  $s$  и  $t$  для записи следующего уравнения цепи.

$$\varphi[s] := 1; s := s + L; t := t + 1.$$

Шаг 10. Приступаем к росписи уравнений модели цепи, задающих закон Кирхгофа для всех узлов цепи, кроме нулевого. Устанавливаем начальное значение номера узла  $N$

$$N := 1.$$

Шаг 11. Начало цикла по росписи уравнения закона Кирхгофа для очередного узла  $N$ . Устанавливаем начальное значение индекса для названий компонентов цепи

$$r := 1.$$

Шаг 12. Устанавливаем начальное значение индекса для номеров ветвей и узлов, указанных в информации о схеме для  $M_r$

$$q := 1.$$

Шаг 13. Проверяем, ведет ли очередная ветвь, образованная выводом компонента  $M_r$ , в узел  $N$ . Если выполняется условие  $j_q^r = N$ , ветвь с номером  $l_q^r$  ведет в узел  $N$ . Предполагая, что ток, вытекающий из узла, имеет знак минус, осуществляем засылку коэффициента в  $\Phi$ -матрицу

$$\varphi[s + i_q^r + Q] := -1 \times \text{sign } B[i_q^r].$$

Если условие  $j_q^r = N$  не выполняется, то переходим к шагу 14.

Шаг 14. Перестраиваем индекс  $q$  для рассмотрения очередного вывода компонента  $M_r$

$$q := q + 1.$$

Возвращаемся к шагу 13  $n_r$  раз (т. е. если  $q < n_r + 1$ ).

Шаг 15. Просмотрены все выводы очередного компонента. Перестраиваем индекс  $r$  на следующий компонент.

$$r := r + 1.$$

Организуем цикл для поиска всех ветвей, ведущих в узел  $N$ . Переход к шагу 12, если  $r < \kappa + 1$ .

Шаг 16. Расписано уравнение для узла  $N$ . Переходим к рассмотрению следующего узла

$$N := N + 1.$$

Организуем цикл для росписи уравнений для каждого узла. Переход к шагу 11, если  $N < Q$ .

Если все узлы перебраны, алгоритм формирования системы уравнений, задающих модель линейной цепи постоянного тока, работу заканчивает.

Следующим этапом анализа линейных цепей является решение построенной системы уравнений. Поскольку  $\Phi$ -матрица является весьма большой (размерность ее  $L \times L$ ), то для задач, подлежащих исследованию на ЭЦВМ типа М-220,  $L$  не превышает пятидесяти.



Если  $\Phi$ -матрица расписана полностью (вместе с нулевыми коэффициентами), то для решения построенной системы уравнений можно применять любой метод исключения типа метода Гаусса. Однако решение уравнений математической модели цепи даже в этом простейшем случае не является тривиальной задачей из-за больших ошибок вычислительного характера. В силу характерных свойств РЭС применение метода Гаусса или другого обычно используемого алгоритма не гарантирует получения правильного результата. Отметим некоторые свойства РЭС, приводящие к упомянутым ошибкам, возникающим в процессе вычислений.

1. Большой разброс коэффициентов модели, который может достигать двадцати и более порядков. Под порядком разброса здесь понимается порядок отношения максимального коэффициента к минимальному. Легко видеть, что такой разброс чисел далеко выходит за рамки машинной точности и в ряде случаев может привести к результату, весьма отличному от истинного.

2. Большой разброс решений цепи, характерный для многих цепей, ведет к большим ошибкам, особенно для малых решений.

Максимальная размерность цепей, равная пятидесяти, в ряде случаев оказывается недостаточной при анализе реальных цепей. Значительно повысить размерность решаемых задач позволяет разреженность (большое количество нулевых элементов)  $\Phi$ -матрицы. Легко заметить, что количество ненулевых элементов растет линейно (пропорционально размерности) при росте размерности  $\Phi$ -матрицы. Число же ненулевых элементов растет пропорционально квадрату размерности. Коэффициент пропорциональности количества ненулевых элементов невелик и для большинства цепей не превышает трех.

В силу приведенных соображений удобным оказывается хранение в памяти ЭЦВМ  $\Phi$ -матрицы в плотном виде, т. е. без нулевых элементов. Можно предложить много способов хранения в памяти машины плотных матриц. Остановимся здесь на следующем. Представим плотную  $\Phi$ -матрицу в виде двух матриц:  $\Phi_1$ , называемой числовой, и  $\Phi_2$ , называемой топологической. Матрица  $\Phi_1$  содержит все ненулевые элементы матрицы  $\Phi$ . Топологическая матрица  $\Phi_2$  содержит информацию о размещении ненулевых элементов в матрице  $\Phi$ . Каждый элемент матрицы  $\Phi_2$  будем задавать

парой чисел  $(l, A)$ , где  $l$  — количество ненулевых элементов в строке  $\Phi$ -матрицы, содержащей ненулевой элемент, номер которого относительно начала строки равен  $A$  (нумерация ведется с нуля). Таким образом,  $\Phi$ -матрица

$$\begin{pmatrix} 0 & 1 \\ 1 & 2 \end{pmatrix}$$

запишется в виде числовой и топологической матриц

$$\Phi_1 = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}, \quad \Phi_2 = \begin{pmatrix} (1, 1) \\ (2, 0) \\ (2, 1) \end{pmatrix}.$$

Вектор  $W$  не требует для хранения большого объема памяти, поэтому будем расписывать его со всеми нулевыми элементами.

Таким образом, выше описаны два представления  $\Phi$ -матрицы, которые мы будем называть соответственно полной и плотной записью  $\Phi$ -матрицы.

Если при анализе цепей с полной  $\Phi$ -матрицей размерность не должна превышать пятидесяти, то при использовании плотной матрицы допустимая размерность резко возрастает. Решение задач линейной алгебры с плотными матрицами в принципе не отличается от аналогичных задач с полными матрицами. С точки зрения программной реализации наиболее удобны для плотных матриц итерационные методы, не изменяющие структуру матриц. Однако при этом возникают новые проблемы точности решения и скорости сходимости.

Организация программ, реализующих методы исключения для плотных матриц, является более сложной, чем для полных матриц, что объясняется тем, что методы исключения изменяют от шага к шагу числовую и топологическую матрицы  $\Phi_1$  и  $\Phi_2$ : многие нулевые элементы матрицы  $\Phi$  переводятся в ненулевые, увеличивая тем самым размерность матриц  $\Phi_1$  и  $\Phi_2$ . Такой непредсказуемый рост матриц  $\Phi_1$  и  $\Phi_2$  является одним из основных ограничений применения методов исключения.

Рассмотрим некоторые алгоритмы решения систем линейных уравнений с плотными матрицами. Исходной информацией являются числовая и топологическая матрицы  $\Phi_1$  и  $\Phi_2$ , вектор правых частей  $W$ , записанный вместе со всеми нулями.



## 1. Итерационные методы

### а) Метод проекций.

Решение системы  $\Phi \times V = W$ ,  $\Phi = (\Phi_1, \Phi_2, \dots, \Phi_n)^T$  с заданным начальным приближением  $V = V_0$  находится по формуле:

$$V_n = V_{n-1} + \frac{W_i - (\Phi_i, V_{n-1})}{(\Phi_i, \Phi_i)} \Phi_i, \quad i = 1, \dots, n.$$

Итерация считается законченной, когда  $i$  пробежит все значения от 1 до  $n$ . Вычисления продолжаются до тех пор, пока  $|V_n - V_{n-1}| \geq \varepsilon$ , где  $\varepsilon$  — заданная величина.

Программа, построенная на основе данного алгоритма, дает решение в ячейках  $V$ , матрицы  $\Phi_1$  и  $\Phi_2$  и вектор  $W$  не портятся. Для работы требуется массив рабочих ячеек длиной  $3n$ .

### б) Метод проекций с двойной точностью.

Все арифметические действия выполняются с удвоенным количеством разрядов. По достижении заданной точности  $\varepsilon$  вычисления прекращаются, и результат уплотняется до одинарной точности. Матрицы  $\Phi_1$ ,  $\Phi_2$  и вектор  $W$  не портятся. Программа использует  $5n$  рабочих ячеек.

### в) Метод скорейшего спуска.

Решение системы линейных алгебраических уравнений  $\Phi \times V = W$  с заданным начальным приближением  $V = V_0$  находится по формулам:

$$r_i = \sum_{j=1}^n \varphi_{ij} \omega_j - \sum_{j=1}^n \sum_{\kappa=1}^n \varphi_{\kappa i} \varphi_{\kappa j} V_j^{(n-1)}, \quad i = 1, \dots, n,$$

$$\alpha = \frac{\sum_{i=1}^n r_i^2}{\sum_{i=1}^n \left( \sum_{j=1}^n \varphi_{ij} r_j \right)^2},$$

$$V_i^{(n)} = V_i^{(n-1)} + \alpha \times r_i, \quad i = 1, \dots, n.$$

Вычисления заканчиваются при  $|\alpha r_i| < \varepsilon$ , где  $\varepsilon$  — заданная точность вычислений.

Программа, реализующая данный метод, получает решение в ячейках  $V$ ; матрицы  $\Phi_1$  и  $\Phi_2$ , а также вектор  $W$  не портятся. Для работы требуется массив рабочих ячеек длиной  $4n$ .

## 2. Методы исключения

Предлагается два варианта метода исключения Гаусса с выбором главного элемента по строке. Будем полагать, что вслед за матрицами  $\Phi_1$  и  $\Phi_2$  имеется некоторый резерв памяти, так как при подстановках размерность матриц может увеличиваться. Распишем алгоритмы по шагам.

А. Шаг 1. Устанавливаем начальное значение номера строки  $i := 1$ .

Шаг 2. В  $i$ -той строке матрицы  $\Phi_1$  находим максимальный по модулю элемент  $\varphi_1[m_i]$ . Строку матрицы  $\Phi_1$  с номером  $i$  делим на  $\varphi_1[m_i]$ , запоминаем номер  $m_i$  максимального элемента в рабочем массиве с ячейки  $Q$ . Делим  $\omega_i$  на главный элемент.

Шаг 3. Просматриваем последовательно строки  $j = i + 1, \dots, n$ . Если в них есть элемент с номером  $m_i$ , то преобразуем  $j$ -тую строку, вычитая из нее  $i$ -тую строку, умноженную на элемент с номером  $m_i$   $j$ -той строки. Вычисляем  $\omega_j = \omega_j - \omega_i \times \varphi[j, m_i]$ .

Шаг 4. Увеличиваем  $i$  на единицу:  $i := i + 1$ . Если  $i \leq n$ , то переходим к шагу 2. В противном случае выполняется шаг 5.

Шаг 5. Начинаем обратный ход. Просматриваем матрицу, начиная с последней строки. Устанавливаем  $Vm_n = \omega_n$ ,  $i := n - 1$ .

Шаг 6. Просматриваем  $i$ -тую строку. Если  $\varphi_1[i, j] \neq 1$ , то вычисляем  $\omega_i = \omega_i - \varphi_1[i, j] \times V_j$ ,  $j = 1, \dots, p_i$  где  $p_i$  — число ненулевых элементов в  $i$ -той строке.

Шаг 7. Присваиваем значение  $Vm_i = \omega_i$ .

Шаг 8. Уменьшаем  $i$  на единицу:  $i := i - 1$ . Если  $i \geq 0$  то переходим к шагу 6. В противном случае в  $V$  получим решение системы.

Программа, составленная на основе данного алгоритма, строит решение системы в ячейках  $V$ ; матрицы  $\Phi_1$ ,  $\Phi_2$  и вектор  $W$  портятся. Промежуточные вычисления производятся в другом кубе МОЗУ.

Б. Шаг 1. Устанавливаем начальное значение индекса для номера строки:

$$q := 1.$$



Шаг 2. Из  $q$ -той строки матрицы  $\Phi_1$  выбирается максимальный по модулю элемент  $\varphi_1[\max]$ . Все элементы  $q$ -той строки делятся на  $\varphi_1[\max]$ ;  $\omega_q := \omega_q / \varphi_1[\max]$ . Эта операция не изменяет структуру  $\Phi_2$ . Запоминаем индекс  $\max$  в рабочем массиве  $Q$  в ячейке  $Q + q - 1$ . Увеличиваем  $q$  на единицу:  $q := q + 1$ .

Шаг 3. Просматриваем строки  $q, \dots, n$  до тех пор, пока не встретится строка, содержащая элемент с номером  $\max$ . Если такой строки нет, переходим к шагу 5.

Шаг 4. Найдена строка  $s$ , из которой нужно исключить неизвестное с номером  $\max$ . Выбираем коэффициент при неизвестном и умножаем на него  $q$ -тую строку, запоминая результат умножения  $p$  в рабочих ячейках. Далее следует вычесть строку  $p$  (включающую элемент из  $W$ ) из строки  $s$ . В силу плотной записи матриц  $\Phi_1$  и  $\Phi_2$  количество элементов в строке  $\Phi_1$  и  $\Phi_2$  может изменяться, в том числе и увеличиваться. Поэтому строка  $s - p$  на месте строки  $s$  не разместится. Распишем новую строку в  $\Phi_1$  и  $\Phi_2$  вслед за этими матрицами. При этом коэффициенты с одинаковыми индексами вычитаются; коэффициенты из  $s$  с индексами, которых нет в  $p$ , переписываются в  $\Phi_1$  и  $\Phi_2$  без изменений; коэффициенты из  $p$ , не входящие в  $s$ , переписываются с обратным знаком. Коэффициент с индексом  $\max$  опускается. Подсчитывается число элементов в новой строке и корректируется соответствующим образом строка в  $\Phi_2$ . Уплотняем  $\Phi_1$ ,  $\Phi_2$  и  $W$ , аннулируя строку  $s$ . Продолжаем просмотр матриц для поиска элементов, подлежащих исключению.

Шаг 5. Если  $q > n$ , то переходим к шагу 6; в противном случае выполняется шаг 2.

Шаг 6. Закончено приведение матрицы к псевдотреугольному виду. Приступаем к подстановке, то есть обратному ходу метода Гаусса. Устанавливаем начальное значение индекса  $i := n$ . Индекс в ячейке  $Q + i - 1$  в рабочем массиве  $Q$  указывает на номер неизвестного, полученного в последней ячейке вектора  $W$ .

Шаг 7. Просматриваются строки матрицы с номерами  $1, 2, \dots, i - 1$ . Если они включают вычисленное на предыдущем шаге значение неизвестного (индекс в ячейке  $Q + i - 1$ ), то это значение умножается на свой коэффициент в строке и вычитается из соответствующего элемента  $W$ . Тем самым в ячейке  $W + i - 2$  получается следующее неизвестное. Далее  $i := i - 1$  и продолжается подстановка. Если  $i = 0$ , переходим к шагу 8.

Шаг 8. Вычислены все неизвестные системы. Упорядочиваем их в соответствии с индексами, указанными в массиве  $Q$ .

Работа алгоритма закончена. Решение получается на месте вектора  $W$ . Матрицы  $\Phi_1$  и  $\Phi_2$  портятся. Для работы требуется  $2n$  рабочих ячеек.

Работу описанных алгоритмов проиллюстрируем на примере следующих схем: однозвенный фильтр  $RC$  (рис. 1),

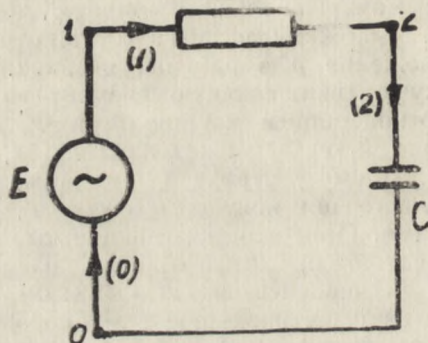


Рис. 1

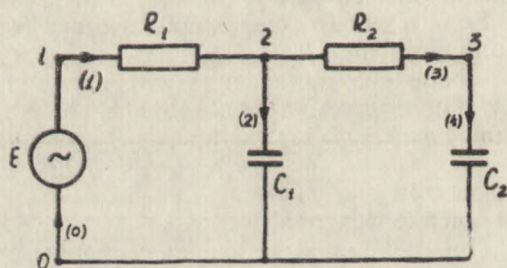


Рис. 2

двухзвенный фильтр ФИЛЬТР (рис. 2), малосигнальный усилитель МСУ (рис. 3), усилитель промежуточной частоты УПЧ (рис. 4).



По результатам применения этих методов можно сделать следующие выводы об их применимости для анализа цепей.

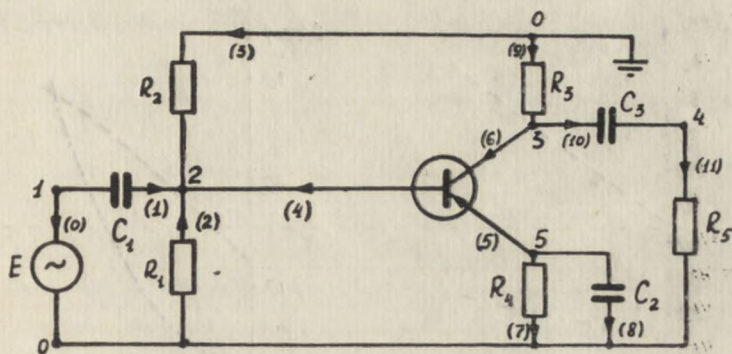


Рис. 3

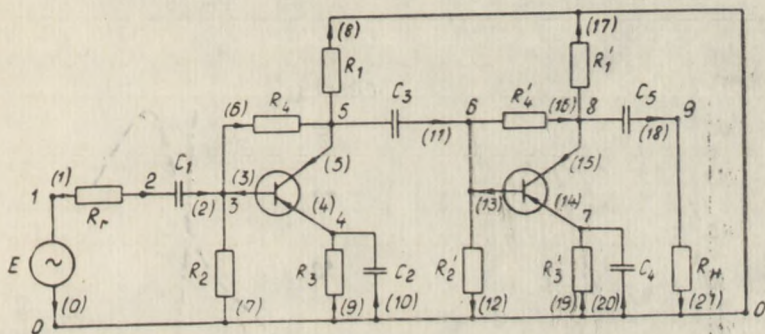


Рис. 4

1. Метод скорейшего спуска практически непригоден для решения задач теории цепей. Так, например, для  $RC$ -цепи при  $R = 1$  ом,  $C = 10^{-6}$  ф,  $\omega = 10^9$  гц,  $\varepsilon = 10^{-9}$  получена максимальная относительная погрешность  $\Delta = 5 \cdot 10^{-2}$ , что неприемлемо на практике. Кроме того, метод имеет плохую сходимость. На рис. 5, 6, 7, 8, 9, 10 (штрих-пунктирная линия) показана зависимость времени счета, соответственно,

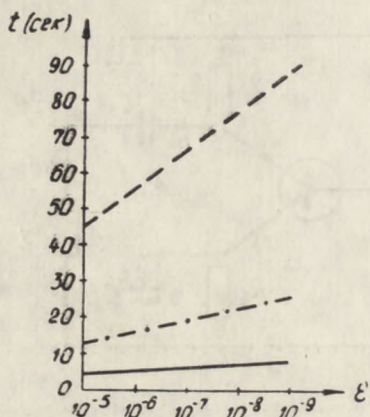


Рис. 5

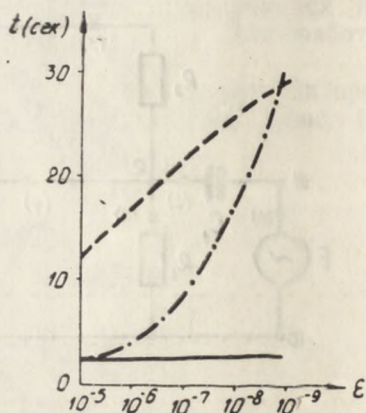


Рис. 6

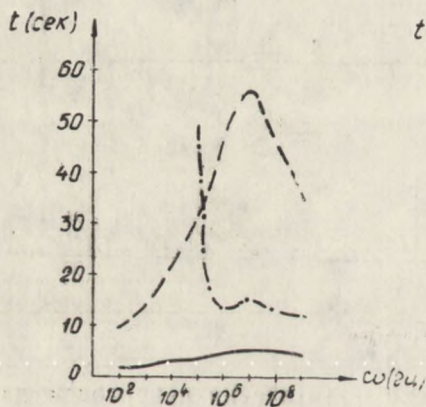


Рис. 7

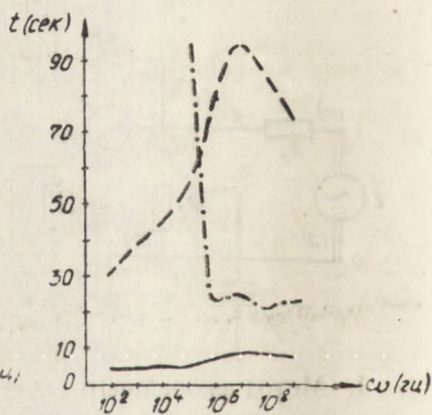


Рис. 8



от точности  $\varepsilon$  для  $RC$ -цепи, от  $\varepsilon$  для ФИЛЬТРА, от частоты  $\omega$  для  $RC$ -цепи при  $\varepsilon = 10^{-5}$  и  $\varepsilon = 10^{-8}$ , от  $\omega$  для ФИЛЬТРА при  $\varepsilon = 10^{-5}$  и  $\varepsilon = 10^{-8}$ . Здесь под точностью  $\varepsilon$  понимается верхняя граница модуля разности двух соседних итераций. Для более сложных схем этот метод вообще не сходится.

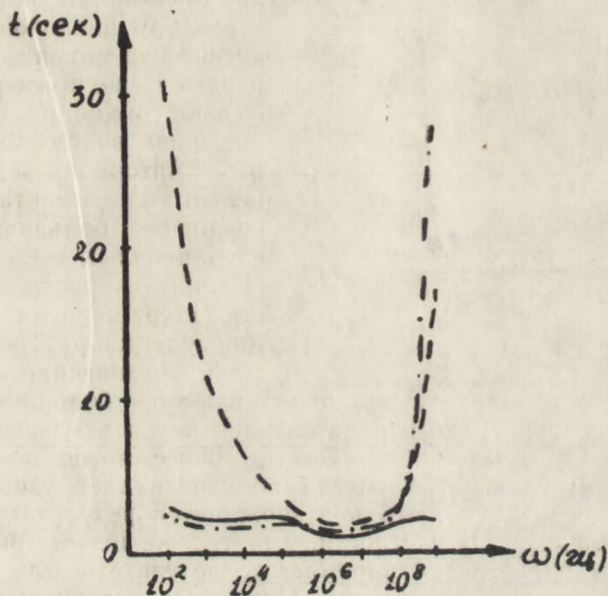


Рис. 9

2. Несколько лучшие результаты дает метод проекций. Удовлетворительная точность получается при анализе схем  $RC$  и ФИЛЬТР. На тех же рисунках (5—10) сплошной линией показаны зависимости времени счета от заданных точности и частоты. Однако для более сложных схем (например, МСУ) сходимость метода обеспечивается лишь при точности, не превосходящей  $10^{-8}$ . Результаты анализа при такой точности неудовлетворительны. При дальнейшем повышении точности итерационный процесс теряет сходимость за счет ошибок округления.

Организация вычислений с двойной точностью (метод проекций с двойной точностью, пунктирная линия на рис.

5—10) не позволяет получить решение, достаточно близкое к истинному в силу медленной сходимости метода.

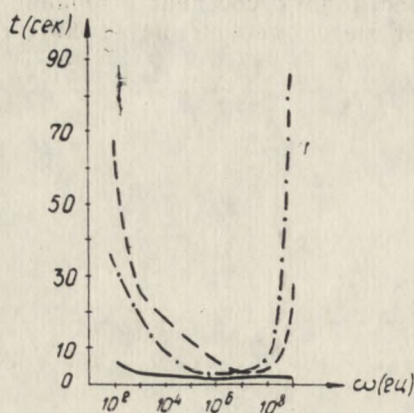


Рис. 10

который в задачах теории цепей является большим.

Известно [2], что в ряде случаев метод исключения дает непредсказуемую ошибку. Тем не менее метод исключения с выбором главного элемента по строке дает удовлетворительные результаты, что подтверждается результатами анализа ряда схем. В частности, для тех же цепей (рис. 1—4) получены практически приемлемые результаты для всех частот ( $10^1 \leq \omega \leq 10^9$ ). На рис. 11 приведена зависимость времени счета от размерности цепи (размерность цепи есть сумма числа узлов и числа ветвей цепи).

Отметим, что время счета при работе с плотными матрицами превосходит время счета при работе с разреженными матрицами. Теоретические оценки показывают, что работа с плотными матрицами становится эффективной не только по объему памяти, но и по быстродействию, когда разреженная матрица не вмещается в МОЗУ. Поэтому экспериментальные сравнительные оценки провести невозможно. Применение метода исключения позволяет анализировать цепи без применения метода подсхем размерностью до 500.

Одним из существенных неудобств применения метода исключения для анализа цепей является непредсказуемое

3. Несмотря на то, что при работе с плотными матрицами более удобными являются итерационные методы, так как они не изменяют структуру числовой и топологической матриц, рассмотренные выше примеры показывают их непригодность к решению задач теории цепей. Авторы не задавались целью экспериментально исследовать большой объем итерационных методов. Обычно скорость сходимости итерационных методов определяется разбросом собственных значений матрицы,



изменение размерности матриц (числа ненулевых элементов), которое может привести к прерыванию работы системы вследствие нехватки памяти. Одним из способов обхода этой трудности является применение алгоритмов предварительного упорядочения строк матрицы. По-видимому, невозможно построить точный алгоритм, обеспечивающий максимальное сокращение размерности матрицы. Рассмотрим следующие приближенные алгоритмы предварительного упорядочения строк.

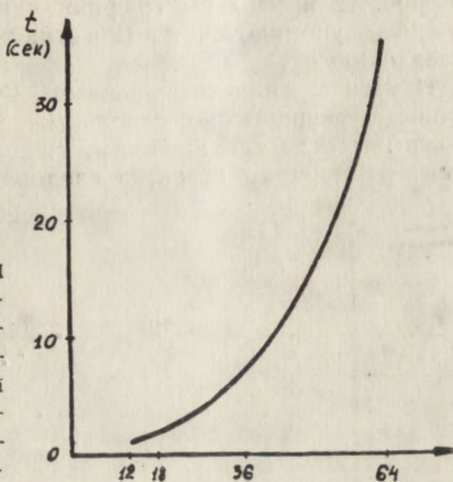


Рис. 11

### Способ 1

Шаг 1. Вычисляем вектор  $L = (l_1, l_2, \dots, l_n)$ , где  $l_j$  равно числу ненулевых элементов в  $j$ -й строке.

Шаг 2. Вычисляем вектор  $M = (m_1, m_2, \dots, m_n)$ ,  $m_i$  равно числу ненулевых элементов  $i$ -го столбца.

Шаг 3. Вычисляем вектор  $P = (p_1, p_2, \dots, p_n)$ ,  $p_j = \max_{i \in I_j} m_i$ , где  $I_j$  — множество индексов ненулевых элементов  $j$ -й строки.

Шаг 4. Упорядочиваем строки по возрастанию  $l_j$ . Строки с равными признаками  $l_j$  упорядочиваем по убыванию соответствующих признаков  $p_j$ .

### Способ 2

Строки упорядочиваются по возрастанию  $(s_1 + s_2)$ , где  $s_1$  — число ненулевых элементов в строке,  $s_2$  — число нулей в столбцах, соответствующих ненулевым элементам строки.

Практика показывает, что строить более точные алгоритмы, например связанные с упорядочением строк на каждом шаге исключения, нецелесообразно, так как уже любой из описанных приемов удовлетворяет основному требованию:

невозрастанию объема матрицы по сравнению с исходной. На рис. 12 приведены графики изменения размерности матриц без упорядочения (крестиками) и с упорядочением (звездочками).

Изучение значения невязок  $\Phi V - W$  показывает, что точность решения по отдельным уравнениям меняется, но максимум остается прежним, т. е. точность решения не зависит от порядка строк, а следовательно, от алгоритма.

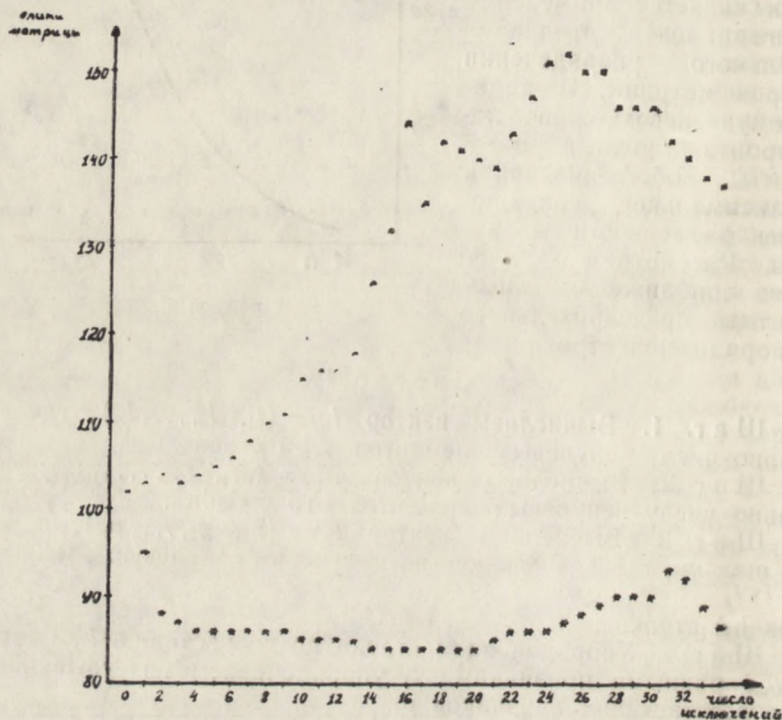


Рис. 12

В заключение на примере цепи МСУ покажем, как готовится исходная информация для анализа цепи на ЭВМ. В системе МАРС есть операторы исследования линейных цепей ИЛ1 и ИЛ2, которые скомплексированы с учетом вышеизложенного, т. е. используют метод исключения. Опера-



тор ИЛ1 производит анализ линейных цепей в частотной форме: определяет переменные цепи (токи в ветвях и напряжения узлов), амплитудно-частотную и фазочастотную характеристики для каждого значения частоты из заданного массива частот. Оператор ИЛ2 определяет частотную чувствительность и групповую задержку по фазе. Вычисленные характеристики схемы выдаются на печать в виде таблиц или графиков.

Программа анализа схемы МСУ имеет вид:

СХЕМА : МСУ;

0 : И ( $N3 = 2_{10} - 2$   $\Phi 4 = 0.$ ) В  $(-0,0)$  N  $(1,0)$ ;

1 : Ц ( $C3 = 5_{10} - 6$ ) В  $(-1,1)$  N  $(1,2)$ ;

2 : Р ( $R3 = 1000.$ ) В  $(-2,2)$  N  $(0,2)$ ;

3 : Р ( $R3 = 6000.$ ) В  $(-3,3)$  N  $(0,2)$ ;

4 : Р ( $R3 = 2000.$ ) В  $(-9,9)$  N  $(0,3)$ ;

5 : Р ( $R3 = 500.$ ) В  $(-7,7)$  N  $(5,0)$ ;

6 : Р ( $R3 = 10000.$ ) В  $(-11,11)$  N  $(4,0)$ ;

7 : Ц ( $C3 = 6_{10} - 5$ ) В  $(-8,8)$  N  $(5,0)$ ;

8 : Ц ( $C3 = 5_{10} - 6$ ) В  $(-10, 10)$  N  $(3,4)$ ;

9 : ТНЭ N ( $N3 = 50., RE4 = 0., RB5 = 350., RC6 = 11_{10}$

3) В  $(5, -4, 6)$  N  $(5, 2, 3)**$ .

READ : ИЛ1,  $|603|$ ,  $|3700|$ ,  $|4600|$ ,  $|400|$ , 17,  $|40|$ ,  $|6000|$ ,  
 $10 - 3$ ,  $|100|$ , Q;

СТОП\*\*

МАССИВ : МА, 40;

$10 1$ ,  $5_{10} 1$ ,  $10 2$ ,  $5_{10} 2$ ,  $10 3$ ,  $5_{10} 3$ ,  $10 4$ ,  $5_{10} 4$ ,  $10 5$ ,  $5_{10} 5$ ,  $10 6$ ,  $5_{10} 6$ ,  
 $10 7$ ,  $5_{10} 7$ ,  $10 8$ ,  $5_{10} 8$ ,  $10 9**$

МАССИВ : Q, 6740;

5, 4, 5, 4, 5, 4, 5, 4, \*\*

АРП : 700\*\*

ЛРП : 3000\*\*

АСХЕМ : 3700\*\*

ЛСХЕМ : 700\*\*

АИС : 7000\*\*

END : \*\*

Здесь строка СХЕМА : МСУ задает цепь (компоненты, значения числовых параметров, номера ветвей и узлов) на входном языке. Строка READ : ИЛ1 есть обращение к оператору ИЛ1. Строки МАССИВ : МА и МАССИВ : Q задают частоты, на которых исследуется цепь, и номера узлов, для которых строятся графики. Строки АРП и ЛРП задают соответственно начало и длину рабочего поля, отводимого для

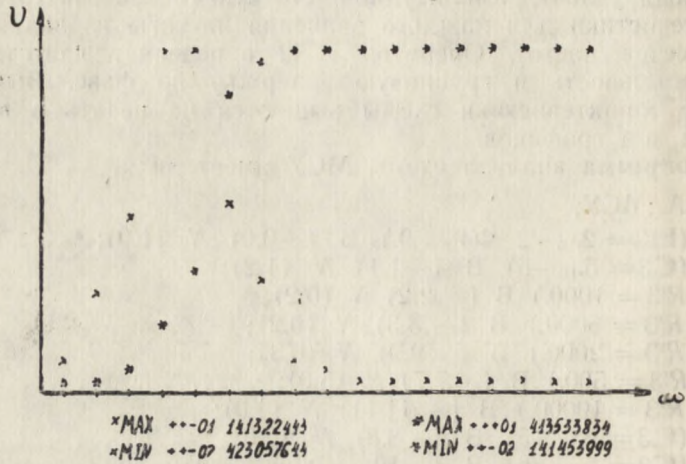


Рис. 13

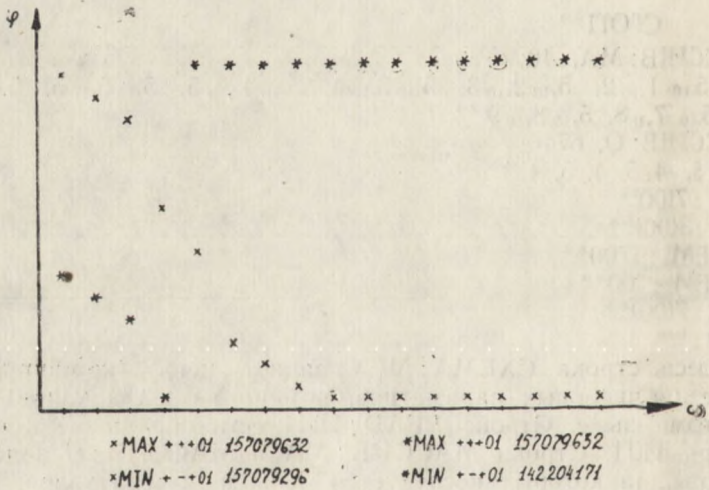


Рис. 14



операторов; АСХЕМ и ЛСХЕМ — начало и длину резерва памяти для размещения модели цепи. Строка АИС:7000\*\* указывает, что рабочее поле интерпретирующей системы ИС-22 начинается с ячейки 7000. END\*\* означает конец информации на входном языке.

В результате работы программы получены амплитудно-частотные и фазочастотные характеристики для значений частот из массива МА. На рис. 13 и 14 изображены соответственно амплитудно-частотная и фазочастотная характеристики для пятого (крестиками) и четвертого (звездочками) узлов. По оси абсцисс точками отмечены значения частоты.

#### ЛИТЕРАТУРА

1. Арайс Е. А. Модельный метод анализа радиоэлектронных схем.— В сб.: Вопросы программирования и автоматизации проектирования. Вып. 11. Томск, Изд-во Томского университета, 1972.
  2. Форсайт Дж., Молер К. Численное решение систем линейных алгебраических уравнений. М., Изд-во «Мир», 1969.
-

## ВЫЧИСЛЕНИЕ ПРОИЗВОДНЫХ ОТ ВЫХОДНЫХ ПАРАМЕТРОВ ЛИНЕЙНЫХ ЦЕПЕЙ

С. А. РАЗИН

При анализе радиоэлектронных схем иногда полезным является знание частных производных от выходных параметров схемы по внутренним параметрам элементов этой схемы. Производные первого порядка (коэффициенты влияния) обычно используются для оценки степени влияния внутренних параметров на выходные параметры. Если отклонения значений внутренних параметров невелики и приемлемо линейное приближение, то коэффициенты влияния можно использовать для вычисления допусков на выходные параметры схемы. В случае больших отклонений внутренних параметров от номиналов и недопустимости линейного приближения, для вычисления допусков на выходные параметры могут использоваться производные более высокого порядка. Ниже предлагается точный способ вычисления частных производных любого порядка для цепей, описываемых линейной алгебраической системой уравнений. Рассматривается приложение этого способа к модельному методу анализа радиоэлектронных схем [1, 2].

### 1. Общий случай

Пусть схема описывается системой линейных алгебраических уравнений

$$AV = W, \quad (1.1)$$

где  $V$  — вектор выходных параметров схемы; элементы матрицы  $A$  и вектора  $W$  являются достаточное число раз диф-



ференцируемыми функциями внутренних параметров схемы  $X = (x_1, \dots, x_m)$ . Дифференцируя (1.1) по  $x_i$   $n$  раз, имеем

$$AV_i^{(n)} = W_i^{(n)} - \sum_{j=0}^{n-1} C_n^j A_i^{(n-j)} V_i^{(j)}. \quad (1.2)$$

Здесь  $V_i^{(n)}$  означает  $n$ -ю производную вектора  $V$  по параметру  $x_i$ ,  $A_i^{(n-j)}$  — производная порядка  $n-j$  от матрицы  $A$  по  $x_i$ . Формула (1.2) определяет рекуррентную процедуру вычисления производной  $n$ -го порядка, если известны все младшие производные. Матрица системы (1.2) та же, что и в (1.1). Это обстоятельство позволяет существенно сократить объем вычислений при нахождении производных по формуле (1.2), если систему (1.1) решать путем обращения матрицы  $A$ . Смешанные частные производные вычисляются аналогично. Дифференцируя (1.1) по  $x_i$  и  $x_j$ , имеем

$$AV_{ij}^{(2)} = W_{ij}^{(2)} - A_i^{(1)} V_j^{(1)} - A_j^{(1)} V_i^{(1)} - A_{ij}^{(2)} V. \quad (1.3)$$

В случае, когда элементы матрицы  $A$  зависят линейно от  $x_i$ , а вектор  $W$  от  $x_i$  не зависит, формула (1.2) перепишется в виде

$$AV_i^{(n)} = -n A_i^{(1)} V_i^{(n-1)},$$

или, вводя обозначение  $D_i = A^{-1} A_i^{(1)}$ , получим

$$V_i^{(n)} = n! (-D_i)^n V. \quad (1.4)$$

Запишем разложение в ряд Тейлора для  $V(x_i)$  в окрестности  $\bar{x}_i$ :

$$V(x_i) = V(\bar{x}_i) + \sum_{j=1}^{\infty} [-D_i(x_i - \bar{x}_i)]^j V(\bar{x}_i). \quad (1.5)$$

Необходимым и достаточным условием сходимости этого матричного ряда будет сходимость ряда

$$\sum_{j=1}^{\infty} [-\lambda_i(x_i - \bar{x}_i)]^j,$$

где  $\lambda_i$  — максимальное по модулю собственное значение матрицы  $D_i$ . Таким образом, ряд (1.5) сходится в интервале

$$|x_i - \bar{x}_i| < \frac{1}{|\lambda_i|}. \quad (1.6)$$

Сумма этого ряда вычислится по формуле

$$V(x_i) = \{E - D_i(x_i - \bar{x}_i)[E + D_i(x_i - \bar{x}_i)]^{-1}\} V(\bar{x}_i), \quad (1.7)$$

где  $E$  обозначает единичную матрицу. Основным объемом вычислений по формуле (1.7) определяется размерностью обрабатываемой матрицы  $H = E + D_i(x_i - \bar{x}_i)$ . Как правило, от параметра  $x_i$  зависит относительно небольшое число элементов матрицы  $A$ . Обозначим через  $g$  общее число столбцов матрицы  $A$ , содержащих такие элементы. Если  $g < N$ , где  $N$  — порядок системы (1.1), то размер обрабатываемой матрицы в (1.7) можно снизить до  $g$ . В этом случае в матрице  $D_i$  ненулевыми столбцами являются только  $g$  столбцов с номерами, равными номерам ненулевых столбцов матрицы  $A$ . Следовательно, для нахождения произведения  $D_i H^{-1}$  достаточно знать только  $g$  соответствующих строк матрицы  $H^{-1}$ . В матрице  $H$  нулевым столбцам матрицы  $D_i$  будут соответствовать столбцы, содержащие лишь одну единицу на диагонали матрицы  $H$ . При вычеркивании строки матрицы  $H$  с номером, равным номеру любого нулевого столбца матрицы  $D_i$ , в оставшейся матрице появляется ненулевой столбец, имеющий тот же номер. Таким образом, алгебраическое дополнение недиагональных элементов этой строки равно нулю, а алгебраическое дополнение диагонального элемента равно определителю матрицы  $H$ . Отсюда, вычисляя элементы матрицы  $H^{-1}$  как отношение алгебраических дополнений элементов матрицы  $H$  к определителю этой матрицы, приходим к выводу, что столбцы матрицы  $H$ , соответствующие нулевым столбцам матрицы  $D_i$ , при обращении  $H$  не меняются. Так как скалярные произведения строк матрицы  $D_i$  и эти столбцы равны нулю, то матрица  $D_i H^{-1}$  имеет те же нулевые столбцы, что и матрица  $D_i$ . Таким образом, вычислению подлежат только элементы матрицы  $H^{-1}$ , стоящие на пересечении столбцов и строк с номерами, равными номерам ненулевых столбцов матрицы  $D_i$ . Эти элементы образуют

матрицу размера  $g \times g$ , которую обозначим  $\tilde{H}^{-1}$ . Аналогичную ей матрицу из коэффициентов матрицы  $H$  обозначим  $\tilde{H}$ . Разлагая определитель матрицы  $H$  по строкам, соответствующим нулевым столбцам матрицы  $D_i$ , и учитывая, что алгебраические дополнения недиагональных элементов этих

строк равны нулю, убеждаемся, что матрицы  $H$  и  $\tilde{H}$  имеют равные определители. Рассматривая алгебраическое дополнение

произвольного элемента матрицы  $\tilde{H}$  и алгебраическое



дополнение этого же элемента в матрице  $H$ , аналогичным образом обнаруживаем их равенство. Отсюда следует, что  $\tilde{H}^{-1} = (\tilde{H})^{-1}$ .

Проведенные выше рассуждения позволяют переписать формулу (1.7) в виде

$$V(x_i) = V(\bar{x}_i) - (x_i - \bar{x}_i) D'_i [E + \tilde{D}_i (x_i - \bar{x}_i)]^{-1} V'(\bar{x}_i), \quad (1.8)$$

где  $D'_i$  — матрица размера  $N \times g$ , полученная из  $D_i$  вычеркиванием всех нулевых столбцов,  $\tilde{D}_i$  — матрица размера  $g \times g$ , полученная из  $D'_i$  вычеркиванием строк с номерами, равными номерам нулевых столбцов матрицы  $D_i$ ,  $V'(x_i)$  —  $g$  — мерный вектор, составленный из соответствующих ненулевым столбцам матрицы  $D_i$  элементов вектора  $V(x_i)$ . Таким образом, размерность обращаемой матрицы снижена до  $g$ . Вместо (1.4) в рассматриваемом случае удобнее использовать формулу

$$V_i^{(n)} = -n! D'_i (-\tilde{D}_i)^{n-1} V'. \quad (1.9)$$

Отметим также, что спектры собственных значений матриц  $D_i$  и  $\tilde{D}_i$  могут отличаться только на значение  $\lambda = 0$ . Это следует из разложения характеристического определителя матрицы  $D_i$  по столбцам с номерами нулевых столбцов этой матрицы. Рассматриваемое разложение имеет вид  $|D - \lambda E| = (-\lambda)^{N-g} |\tilde{D} - \lambda E|$ . Следовательно, для определения интервала сходимости ряда (1.5) с помощью неравенства (1.6) достаточно найти максимальное по модулю собственное значение матрицы  $\tilde{D}_i$ .

Неравенство (1.6) определяет окрестность, внутри которой функции  $V(x_i)$  могут быть разложены в сходящийся ряд Тейлора, при этом сумма ряда вычислится по формуле (1.7). Само же выражение (1.7) справедливо во всем интервале изменения параметра  $x_i$ . Для доказательства предположим, что (1.7) справедливо не везде, то есть найдется интервал  $M$ , может быть, более широкий, чем окрестность (1.6), вне которого по крайней мере одна зависимость  $v_j(x_i) \in V(x_i)$  не совпадает с истинной зависимостью  $\tilde{v}_j(x_i)$  параметра  $v_j$  от  $x_i$ . Зависимости  $v_j(x_i)$  и  $\tilde{v}_j(x_i)$  являются дробно-рациональными функциями и их значения совпадают внутри  $M$ . Разность  $v_j(x_i) - \tilde{v}_j(x_i)$  также является

дробно-рациональной функцией, то есть представляет отношение двух полиномов. Числитель этого отношения не является тождественным нулем, так как вне  $M$  есть точки, где  $v_j(x_i) - \tilde{v}_j(x_i) \neq 0$ . Внутри  $M$  эта разность равна нулю и, следовательно, многочлен, являющийся числителем разности, должен иметь континуальное число корней. Полученное противоречие доказывает утверждение.

Если параметр  $x_i$  входит линейно в вектор свободных членов  $W$  системы (1.1), то  $n$ -я производная от вектора  $V$  по  $x_i$  вычислится по формуле

$$V_i^{(n)} = n! (-D_i)^{n-1} (A^{-1} W_i^{(1)} - D_i V),$$

а к правой части формулы (1.7) добавится слагаемое вида

$$(x_i - \bar{x}_i) [E + D_i (x_i - \bar{x}_i)]^{-1} A^{-1} W_i^{(1)}.$$

При этом, если  $x_i$  не входит в матрицу  $A$ , то  $D_i = 0$  и имеет место линейная зависимость

$$V(x_i) = V(\bar{x}_i) + (x_i - \bar{x}_i) A^{-1} W_i^{(1)}.$$

Рассмотрим  $n$ -ю частную производную от вектора  $V$  по системе параметров  $x_{i_1}, x_{i_2}, \dots, x_{i_n}$ , предположив, что эти параметры входят линейно в матрицу  $A$  и не входят в вектор  $W$ . Дифференцируя систему (1.1)  $n$  раз по такой системе параметров, получим:

$$V_{i_1, \dots, i_n}^{(n)} = - \sum_{k=1}^n D_{i_k} V_{i_1, \dots, i_{k-1}, i_{k+1}, \dots, i_n}, \quad (1.10)$$

где по-прежнему  $D_{i_k} = A^{-1} A_{i_k}^{(1)}$ . Осуществляя последовательную подстановку вычисляемых аналогичным образом младших производных в формулу (1.10), приходим к выражению

$$V_{i_1, \dots, i_n}^{(n)} = (-1)^n \sum_{J \in P_n} D_{j_1} D_{j_2} \dots D_{j_n} V, \quad (1.11)$$

где  $J = (j_1, j_2, \dots, j_n)$ , а  $P_n$  обозначает множество всех перестановок индексов  $(i_1, i_2, \dots, i_n)$ . Формула (1.4) является частным случаем формулы (1.11), при  $i_1 = i_2 = \dots = i_n$ . Формулу (1.10) целесообразно использовать для последовательного нахождения всех частных производных до порядка  $n$  включительно. Для нахождения отдельных частных производных удобнее применять формулу (1.11).



## 2. Линейные резистивные цепи

Рассмотрим линейную резистивную цепь, содержащую в себе  $k$  узлов,  $m$  ветвей,  $q_E$  источников напряжения,  $q_I$  источников тока и  $r$  резисторов. Согласно [2], модель такой цепи будет описываться системой

$$\begin{vmatrix} \Pi_Q & 0 \\ 0 & \Gamma_Q \\ A_1 & -R \\ 0 & A^t \end{vmatrix} \begin{vmatrix} U \\ J \end{vmatrix} = \begin{vmatrix} E \\ I \\ 0 \\ 0 \end{vmatrix}, \quad (2.1)$$

где  $U$  — вектор узловых напряжений;  $I$  — вектор токов в ветвях схемы;  $E$  — вектор задающих напряжений размерности  $q_E$ ;  $I$  — аналогичный вектор токов размерности  $q_I$ ;  $R$  — матрица сопротивлений размерности  $r \times k$ , построенная в координатах: резисторы — ветви схемы;  $A_1$  — матрица инцидентностей для  $R$ -цепи размерности  $r \times k$ ;  $\Pi_Q$  — топологическая матрица задающих напряжений, построенная в координатах: источники напряжения — узлы схемы, размерности  $q_E \times k$ ;  $\Gamma_Q$  — топологическая матрица задающих токов, построенная в координатах: источники тока — ветви схемы, размерности  $q_I \times m$ ;  $A^t$  — матрица топологического уравнения 1-го закона Кирхгофа, размерности  $k \times m$ . Внутренние параметры схемы входят в матрицу  $R$  и векторы  $E$  и  $I$ . Матрицы  $\Pi_Q$ ,  $\Gamma_Q$ ,  $A_1$  и  $A^t$  состоят из нулей и взятых с соответствующими знаками единиц. Систему (2.1) кратко будем записывать в виде  $AV = W$ .

Рассмотрим зависимость выходных параметров (напряжений и токов) схемы от параметров источников напряжения и тока. Продифференцировав (2.1) по  $E_i$ , получим

$$AV_i^{(1)} = W_i^{(1)},$$

где в  $W_i^{(1)}$  отличным от нуля элементом является лишь одна единица в  $i$ -й позиции.

Таким образом,

$$V_i^{(1)} = A^{-1} W_i^{(1)} = B_i,$$

где  $B_i$  —  $i$ -й столбец матрицы  $A^{-1}$ . Следовательно, первые  $q_E$  столбцов матрицы  $A^{-1}$  представляют коэффициенты влияния источников напряжения на выходные параметры  $V$ . Аналогично следующие за ними  $q_I$  столбцов матрицы  $A^{-1}$  являются коэффициентами влияния источников тока. Легко

видеть, что все последующие производные порядка от вектора  $V$  по параметрам источников равны нулю. Таким образом, зависимости компонент вектора  $V$  от этих параметров линейны и выполняется принцип наложения. Если обозначить  $A^{-1}_Q$  матрицу из первых  $q_E + q_I = q$  столбцов матрицы  $A^{-1}$ ,  $W_Q$  — вектор-столбец задающих напряжений и токов, то для выходных параметров схемы получаем набор зависимостей

$$V = A_Q^{-1} W_Q. \quad (2.2)$$

Перейдем теперь к исследованию зависимости выходных параметров от величин резисторов цепи. Пусть резистор  $R$  включен в  $j$ -ю ветвь, тогда его величина будет стоять в матрице  $A$  на пересечении столбца  $k + j$  со строкой  $q + i$ , а матрица  $A_i^{(1)}$  будет иметь на этом месте  $(-1)$ . Все остальные элементы матрицы  $A_i^{(1)}$  равны нулю, поэтому матрица  $D_i = A^{-1} A_i^{(1)}$  будет содержать только один ненулевой столбец с номером  $k + j$ , который представляет взятый со знаком минус столбец  $q + i$  матрицы  $A^{-1}$ , то есть равен  $(-B_{q+i})$ . Таким образом,

$$D'_i = -B_{q+i}, \quad \tilde{D}_i = b_{k+j, q+i}, \quad V'(\bar{x}_i) = v_{k+j}(\bar{x}_i) = J_j.$$

Формула (1.9) в этом случае сведется к виду

$$V_i^{(n)} = n! b_{k+j, q+i}^{n-1} J_j B_{q+i}.$$

Отсюда при  $n = 1$  получаем выражение для коэффициентов влияния

$$Y_i = V_i^{(1)} = J_j B_{q+i}. \quad (2.3)$$

Таким образом, для нахождения коэффициента влияния резистора  $R_i$  на выходной параметр  $v_g$  нужно ток, протекающий через этот резистор, умножить на элемент  $b_{g, q+j}$  матрицы  $A^{-1}$ .

Согласно (1.8),

$$V(R_i) = V(\bar{R}_i) + (R_i - \bar{R}_i) B_{q+i} (1 - b_{k+j, q+i})^{-1} J_j, \quad (2.4)$$

а из (1.6) получаем окрестность

$$|R_i - \bar{R}_i| < \frac{1}{|b_{k+j, q+i}|},$$

внутри которой вектор  $V(R_i)$  может быть разложен в следующий ряд Тейлора.



Основываясь на формуле (2.4), можно предложить экспериментальный способ определения зависимости выходных параметров  $V$  от величины некоторого резистора  $R$ . Формулу (2.4) через приращения параметров перепишем, опуская индексы, в виде

$$\Delta V = Y \frac{\Delta R}{1 - b \Delta R}, \quad (2.5)$$

где  $Y$ , согласно (2.3), есть вектор коэффициентов влияния. Задавая два приращения  $\Delta R_1 \neq \Delta R_2$  величины  $R$  и измеряя соответствующие приращения  $\Delta v_1$  и  $\Delta v_2$  некоторого выходного параметра  $v \in V$ , можно вычислить  $y$  и  $b$  из полученной системы двух уравнений. Для вычисления остальных коэффициентов влияния достаточно измерять по одному приращению выходных параметров. Таким образом, для получения  $n$  зависимостей выходных параметров от величины  $R$  нужно произвести всего  $n + 1$  измерение. Полученная при этом погрешность будет определяться только погрешностью измерений и может быть уменьшена путем выбора достаточно больших приращений  $\Delta R_1$  и  $\Delta R_2$ . Если в схеме допустимо отключение и замыкание накоротко резистора  $R$ , то, обозначая через  $\Delta V_\infty$  и  $\Delta v_0$  соответствующие приращения вектора  $V$  и некоторого параметра  $v \in V$ , приходим к формулам:

$$b = - \frac{\Delta v_0}{(\Delta v_0 - \Delta v_\infty) R}, \quad Y = - b \Delta V_\infty.$$

Остановимся на вычислении смешанных производных. Так как матрицы, входящие в (1.10), в рассматриваемом случае содержат по одному ненулевому столбцу, то их произведение сводится к произведению ненулевого столбца первой матрицы на элементы ненулевых столбцов последующих матриц, при этом из каждого столбца берется по одному элементу с номером, равным номеру ненулевого столбца предыдущей матрицы. Результирующий столбец получается на месте ненулевого столбца последней матрицы произведения.

При  $n = 2$  из (1.11) получаем

$$V_{i_1 i_2} = B_{q+i_1} b_{k+j_1, q+i_2} J_{j_2} + B_{q+i_2} b_{k+j_2, q+i_1} J_{j_1},$$

где  $j_1$  и  $j_2$  — номера ветвей, содержащих резисторы  $R_{i_1}$  и  $R_{i_2}$ . После того, как система (2.1) решена методом обратной

матрицы, для нахождения вектора смешанных производных по формуле (1.10) требуется дополнительно  $n!(n+N)-1$  арифметических операций. Если эти же производные оценивать по разностной формуле путем вычисления  $n$  приращений вектора  $V$  и для решения системы (2.1) использовать, например, метод Гаусса, то потребуется  $2n \left[ \frac{N^3}{3} + \frac{N(N+1)}{2} \right] + N(3n-1)$  арифметических операций. Так как обычно  $n \ll N$ , то вычисление по формуле (1.10) требует гораздо меньшего числа операций. Например, при  $N=10$  эта формула сохраняет свое преимущество до значений  $n=5$ . Кроме того, следует учитывать, что формула (1.10) дает точное значение производной.

### 3. Линейные цепи переменного тока (частотная область)

Рассмотрим некоторую линейную цепь переменного тока, которая содержит в себе  $k$  узлов,  $m$  ветвей,  $q_E$  источников напряжения,  $q_i$  источников тока,  $r$  резисторов,  $s$  емкостей и  $l$  индуктивностей. Согласно [2], модель этой цепи описывается системой:

$$\begin{vmatrix}
 \Pi_Q & 0 & 0 & 0 \\
 0 & \Pi_Q & 0 & 0 \\
 0 & 0 & \Gamma_Q & 0 \\
 0 & 0 & 0 & \Gamma_Q \\
 A_1 & 0 & -R & 0 \\
 0 & A_1 & 0 & -R \\
 0 & \Pi_C & \Gamma_C \frac{1}{\omega C} & 0 \\
 \Pi_C & 0 & 0 & -\Gamma_C \frac{1}{\omega C} \\
 0 & \Pi_L & -\Gamma_L \omega L & 0 \\
 \Pi_L & 0 & 0 & \Gamma_L \omega L \\
 0 & 0 & A^t & 0 \\
 0 & 0 & 0 & A^t
 \end{vmatrix}
 \begin{vmatrix}
 U' \\
 U'' \\
 J' \\
 J''
 \end{vmatrix}
 =
 \begin{vmatrix}
 E' \\
 E'' \\
 I' \\
 I'' \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0
 \end{vmatrix} \quad (3.1)$$



где  $\Pi_C$ ,  $\Pi_L$  и  $\Gamma_C$ ,  $\Gamma_L$  — матрицы, аналогичные матрицам  $\Pi_Q$  и  $\Gamma_Q$ , но записанные для емкостей и индуктивностей; множители  $\frac{1}{\omega C}$  и  $\omega L$  при матрицах  $\Gamma_C$  и  $\Gamma_L$  означают, что строки этих матриц умножены на соответствующие им реактивные сопротивления;  $U'$ ,  $E'$ ,  $J'$ ,  $I'$  и  $U''$ ,  $E''$ ,  $J''$ ,  $I''$  обозначают действительные и мнимые составляющие напряжений и токов. Как и прежде, система (3.1) кратко будет записываться в виде  $AV = W$ .

Аналогично случаю линейных резистивных цепей зависимость вектора  $V$  от параметров источников ( $E'$ ,  $E''$ ,  $I'$ ,  $I''$ ) линейна с коэффициентами пропорциональности, занимающими первые  $2q = 2(q_E + q_I)$  столбцов матрицы  $A^{-1}$ .

Рассмотрим зависимость вектора  $V$  от величины резистора  $R_i$ . Пусть ветвь, содержащая этот резистор, имеет номер  $j$ , тогда матрица  $A_i^{(1)}$  содержит лишь две  $(-1)$  на пересечении столбцов  $k+j$  и  $k+m+j$  со строками  $q+i$  и  $q+r+i$ . В матрице  $D_i$  отличными от нуля столбцами будут столбцы  $2k+j$  и  $2k+m+j$ , которые соответственно равны взятым со знаком минус столбцам  $2q+i$  и  $2q+r+i$  матрицы  $A^{-1}$ . Формула (1.9) в рассматриваемом случае примет вид

$$V_i^{(n)} = n! B_i' \tilde{B}_i^{n-1} J_j,$$

где матрица  $B_i' = -D_i'$  состоит из столбцов  $2q+i$  и  $2q+r+i$  матрицы  $A^{-1}$ ,  $J_j$  — вектор-столбец из действительной и мнимой составляющих тока  $j$ -й ветви и

$$\tilde{B}_i = (-\tilde{D}_i) = \begin{vmatrix} b_{2k+j, 2q+i} & b_{2k+j, 2q+r+i} \\ b_{2k+m+j, 2q+i} & b_{2k+m+j, 2q+r+i} \end{vmatrix}.$$

При  $n = 1$  получаем выражение для вектора коэффициентов влияния

$$Y_i = V_i^{(1)} = B_i' J_j.$$

Формуле (1.8) будет соответствовать выражение

$$V(R_i) = V(\bar{R}_i) + (R_i - \bar{R}_i) B_i' [E - \tilde{B}_i (R_i - \bar{R}_i)]^{-1} J_j,$$

которое после обращения матрицы второго порядка сводится к виду

$$V(R_i) = V(\bar{R}_i) + (R_i - \bar{R}_i) \det [E - \tilde{B}_i(R_i - \bar{R}_i)]^{-1} \tilde{B}_i' [E - \tilde{B}_i'^T (R_i - \bar{R}_i)] J_j = V(\bar{R}_i) + \frac{R_i - \bar{R}_i}{1 - h_i(R_i - \bar{R}_i) + p_i(R_i - \bar{R}_i)^2} [Y_i - (R_i - \bar{R}_i) F_i]. \quad (3.2)$$

Здесь  $\tilde{B}_i'^T$  означает транспонированную относительно второй диагонали матрицу  $\tilde{B}_i$ , вектор  $F_i = \tilde{B}_i' \tilde{B}_i'^T J_j$ , а коэффициенты  $h_i$  и  $p_i$  вычисляются по формулам:

$$h_i = \text{tr}(\tilde{B}_i) = b_{2k+j, 2q+i} + b_{2k+m+j, 2q+r+i}, \quad p_i = \det(\tilde{B}_i).$$

Матрица  $\tilde{D}_i$  имеет характеристическое уравнение

$$\lambda^2 + h_i \lambda + p_i = 0.$$

Если корни этого уравнения  $\lambda_1$  и  $\lambda_2$  действительны то

$$|\lambda_i| = \max(|\lambda_1|, |\lambda_2|) = \frac{|h_i| + \sqrt{h_i^2 - 4p_i}}{2},$$

в противном случае  $|\lambda_i| = \sqrt{p_i}$ . Полученное значение  $|\lambda_i|$  позволяет найти по формуле (1.6) окрестность, внутри которой вектор  $V(R_i)$  может быть разложен в сходящийся ряд Тейлора.

Исследование влияния реактивных сопротивлений  $X_C = \frac{1}{\omega C}$  и  $X_L = \omega L$  аналогично предыдущему. Единственным отличием является то, что в случае емкости первый, а в случае индуктивности второй столбец матрицы  $\tilde{B}_i$  следует взять со знаком минус. Таким образом, вид зависимости вектора  $V$  от  $X_C$  или  $X_L$  полностью совпадает с (3.2). Эту зависимость перепишем для приращений параметров в виде

$$\Delta V = \frac{\Delta Z}{1 - h\Delta Z + p\Delta Z^2} (Y - \Delta Z F), \quad (3.3)$$

где  $\Delta Z$  — приращение сопротивления  $Z$  (активного или реактивного). Зависимости (3.3) можно определить экспериментально. Для этого выделив некоторый комплексный



параметр  $v \in V$ , действительная  $v'$  и мнимая  $v''$  части которого принадлежат вектору  $V$ , и задав три различных приращения  $Z$ , на основании трех измеренных приращений величин  $v'$  и  $v''$  из простой системы шести уравнений можно вычислить коэффициенты  $h, p, y', y'', f', f''$ . Для получения каждой последующей зависимости достаточно производить два измерения, так как  $h$  и  $p$  уже известны. Таким образом, для получения  $n$  зависимостей  $v(Z)$  требуется  $2n + 1$  измерение. Получать приращения  $\Delta Z$  реактивных сопротивлений путем изменения частоты  $\omega$  нельзя, так как от  $\omega$  зависят коэффициенты формулы (3.3). Поэтому (3.7) не следует рассматривать как формулу, определяющую зависимость  $V$  от  $\omega$ .

Для получения зависимости  $V(\omega)$  умножим на  $\omega$  строки матрицы  $A$ , содержащие величины  $\frac{1}{\omega C}$ . Новую матрицу обозначим  $A_\omega$ . Соответствующая ей обратная матрица получается из  $A^{-1}$  путем умножения столбцов с номерами  $2(q+r)+1, \dots, 2(q+r)+s$  на  $\omega^{-1}$ . Число  $g$  столбцов матрицы  $A_\omega$ , содержащих частоту  $\omega$ , равно удвоенной сумме числа индуктивностей и числа узлов ненулевого потенциала, включающих в себя емкости. Таким образом,  $2(s+l) \leq g \leq 2(2s+l)$ . Согласно (1.8) и (1.9)

$$V(\omega) = V(\omega_0) - (\omega - \omega_0) D'_\omega [E - \tilde{D}_\omega (\omega - \omega_0)]^{-1} V'(\omega_0), \quad (3.4)$$

$$V_\omega^{(n)} = n! (-D'_\omega) (-\tilde{D}_\omega)^{n-1} V'. \quad (3.5)$$

Если  $g \ll N$ , то при построении АЧ и ФЧ характеристик вычислением  $V(\omega)$  в дискретной последовательности значений частоты  $\omega$  формула (3.4) требует значительно меньшего объема вычислений, чем непосредственное решение системы (3.1). Формула (3.4) упрощается, если интересующие выходные параметры принадлежат только вектору  $V'(\omega)$ , то есть являются напряжениями в узлах, содержащих емкости, и токами в индуктивных ветвях. В этом случае  $(V')_\omega^m = n! (-\tilde{D}_\omega)^n V'$  и

$$V'(\omega) = [E + \tilde{D}_\omega (\omega - \omega_0)]^{-1} V'(\omega_0).$$

Производные (3.5) можно использовать для аппроксимации зависимости  $V(\omega)$  в окрестности  $\omega_0$  отрезком ряда Тейлора.

Размер окрестности, вне которой такое приближение недопустимо, определится из (1.6). При этом отметим, что если  $g > 2(s + l)$ , то  $g$  ненулевых столбцов матрицы  $D\omega = A_{\omega}^{-1} A_{\omega}^{(1)}$  являются линейной комбинацией  $2(s + l)$  столбцов матрицы  $A_{\omega}^{-1}$  и, следовательно, матрица  $\tilde{D}_{\omega}$  имеет ранг не выше  $2(s + l)$ . Поэтому число отличных от нуля собственных значений матрицы  $\tilde{D}_{\omega}$  не превосходит  $2(s + l)$ .

Вычисление  $n$ -й смешанной производной по параметрам  $R, C, L$  с помощью (1.11) сводится к нахождению суммы  $n!$  слагаемых, являющихся произведениями матрицы размера  $N \times 2$  на матрицы второго порядка. Каждая такая матрица получается из соответствующей матрицы  $D'$  извлечением строк с номерами ненулевых столбцов предыдущей матрицы. Результирующие столбцы произведения образуются на месте ненулевых столбиков последнего сомножителя.

Например, при  $n = 2$

$$V_{i_1 i_2}^{(2)} = D'_{i_1} \tilde{D}_{i_1 i_2} J_{j_2} + D'_{i_2} \tilde{D}_{i_1 i_2} J_{j_1}.$$

Здесь  $j_1, j_2$  — номера ветвей, содержащих элементы цепи  $i_1, i_2$ ;  $I_{j_1}$  и  $I_{j_2}$  — вектор-столбцы из действительной и мнимой составляющих токов этих ветвей; матрица  $\tilde{D}_{i_1 i_2}$  состоит из двух строк матрицы  $D'^{-1}$  с номерами ненулевых столбцов матрицы  $D_{i_{\varepsilon_2}}$ ; матрица  $\tilde{D}_{i_1 i_2}$  строится аналогично.

Исследование влияния внутренних параметров на выходные параметры малосигнальных цепей переменного тока ничем существенным не отличается от предыдущего. При анализе этих цепей в частотной области получаются результаты, аналогичные рассмотренному здесь случаю линейных цепей переменного тока. Безынерционные малосигнальные цепи, описываемые системой линейных уравнений  $AV(t) = W(t)$ , в каждый фиксированный момент времени  $t$  анализируются аналогично изложенному в п. 2 случаю линейных резистивных цепей. Все формулы п. 2, учитывающие влияние источников и резисторов на выходные параметры, справедливы в рассматриваемом случае, если в них под  $V$  и  $W$  подразумевать вектор-функции времени. Так, формула (2.2) определяет решение системы во временной области. Формулы, учитывающие влияние параметров зависимых источников, по виду также полностью совпадают с соответствующими формулами п. 2.



В заключение рассмотрим пример применения изложенной здесь методики к анализу линейной цепи переменного тока, изображенной на рис. 1.

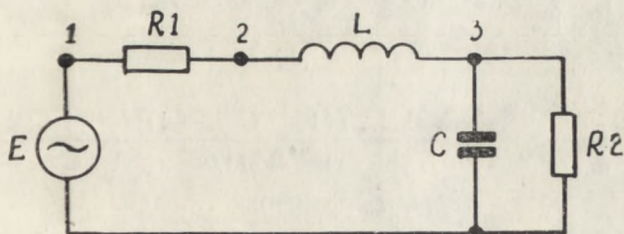


Рис. 1

Для расчетов были выбраны следующие параметры элементов:  $E = 1\text{В}$ ,  $\omega = 10^6$  рад/сек,  $R_1 = 10$  Ом,  $R_2 = 100$  кОм,  $C = 1000$  пФ,  $L = 1$  мГн. Используя вычисленные коэффициенты обратной матрицы системы вида (3.1), описывающей данную цепь, были получены следующие зависимости напряжения  $\dot{U}_3$  в узле 3:

$$\dot{U}_3(E) = 0,2499938E - j49,99875E,$$

$$\dot{U}_3(R_1) = 0,2499938 -$$

$$- \frac{[0,624989 \cdot 10^{-6} + 0,624967 \cdot 10^{-3} (R_1 - 10)] (R_1 - 10)}{1 + 0,1000025 (R_1 - 10) + 0,250019 \cdot 10^{-2} (R_1 - 10)^2} +$$

$$+ j \left\{ -49,99875 + \right.$$

$$\left. + \frac{[2,5000625 + 0,125006 (R_1 - 10)] (R_1 - 10)}{1 + 0,1000025 (R_1 - 10) + 0,250019 \cdot 10^{-2} (R_1 - 10)^2} \right\},$$

$$\dot{U}_3(R_2) = 0,2499938 +$$

$$+ \frac{[0,624985 \cdot 10^{-10} + 0,625031 \cdot 10^{-11} (R_2 - 10^5)] (R_2 - 10^5)}{1 + 0,999975 \cdot 10^{-5} (R_2 - 10^5) + 0,249994 \cdot 10^{-10} (R_2 - 10^5)^2} -$$

$$- j \left\{ 49,99875 + \right.$$

$$\left. + \frac{[0,250006 \cdot 10^{-3} + 0,125000 \cdot 10^{-8} (R_2 - 10^5)] (R_2 - 10^5)}{1 + 0,999975 \cdot 10^{-5} (R_2 - 10^5) + 0,249994 \cdot 10^{-10} (R_2 - 10^5)^2} \right\},$$

$$\begin{aligned}
 \dot{U}_3(C) &= 0,2499938 + \\
 &+ \frac{[2500,0625 + 1875,0313 (10^{-9} C^{-1} - 1)] (10^{-9} C^{-1} - 1)}{1 + 1,500012 (10^{-9} C^{-1} - 1) + 2500,6875 (10^{-9} C^{-1} - 1)^2} + \\
 &+ j \left\{ -49,99875 + \right. \\
 &+ \left. \frac{[0,624968 \cdot 10^{-3} + 125006,25 (10^{-9} C^{-1} - 1)] (10^{-9} C^{-1} - 1)}{1 + 1,500012 (10^{-9} C^{-1} - 1) + 2500,6875 (10^{-9} C^{-1} - 1)^2} \right\}, \\
 \dot{U}_3(L) &= 0,2499938 - \\
 &- \frac{[2500,0625 + 625,0312 (10^3 L - 1)] (10^3 L - 1)}{1 + 0,499999 (10^3 L - 1) + 2500,1875 (10^3 L - 1)^2} - \\
 &- j \left\{ 49,99875 + \right. \\
 &+ \left. \frac{[0,624969 \cdot 10^{-3} - 125006,25 (10^3 L - 1)] (10^3 L - 1)}{1 + 0,499999 (10^3 L - 1) + 2500,1875 (10^3 L - 1)^2} \right\}.
 \end{aligned}$$

Ниже записаны соответствующие окрестности разложения в ряд Тейлора:

$$-10 < R_1 < 30, \quad -10^5 < R_2 < 3 \cdot 10^5$$

$$980,4 < C < 1020,4, \quad 0,98 < L < 1,02.$$

Из двух последних неравенств следует, что при вычислении допуска на выходное напряжение данной цепи использовать разложение в ряд Тейлора можно при допуске на отклонение  $C$  и  $L$  от номинала не более 2%.

### Заключение

В данной статье предложен способ вычисления частных производных любого порядка от выходных параметров линейных цепей. Значения производных получаются с помощью легко реализуемых на ЭВМ операций над элементами обратной матрицы системы уравнений, описывающих рассматриваемую цепь. Наиболее просто этот способ реализуется в модельном методе моделирования и анализа радиоэлектронных схем. В этом случае получен также общий вид одномерных зависимостей выходных параметров от внутренних параметров цепи. Коэффициенты этих зависимостей



связаны простыми соотношениями с элементами обратной матрицы. Найден точный способ определения размера окрестностей, внутри которых рассматриваемые зависимости могут быть разложены в сходящийся ряд Тейлора. Получена формула, которая позволяет сократить объем вычислений при построении АЧ и ФЧ характеристик линейной цепи.

По сравнению с методом приращений предлагаемый способ вычисления производных при существенно меньших вычислительных затратах обеспечивает значительно более высокую точность, которая определяется только погрешностью округления при промежуточных вычислениях.

#### ЛИТЕРАТУРА

1. Арайс Е. А. Модельный метод анализа радиоэлектронных схем.— В сб.: Вопросы программирования и автоматизации проектирования. Вып. 2, Томск, 1972, с. 26—34.
  2. Арайс Е. А., Дмитриев В. М. Матричная интерпретация модельного метода анализа цепей.— В сб.: Вопросы программирования и автоматизации проектирования. Вып. 2, Томск, 1972, с. 35—66.
-

## РЕАЛИЗАЦИЯ ВХОДНОГО ЯЗЫКА АВТО-АНАЛИТИКА НА ЭВМ БЭСМ-6

Б. Ш. ГЕЛЬФМАН

Для облегчения общения пользователя с системой программирования Авто-Аналитик [1] был разработан входной язык. В настоящей статье описывается транслятор с входного языка системы программирования Авто-Аналитик (ТАА).

Транслятор переводит текст, написанный на входном языке, в программу, записанную в машинном коде ЭВМ БЭСМ-6. ТАА сделан в виде двух пакетов — пакета стандартных операторов (ПСО) системы программирования Авто-Аналитик и пакета стандартных программ (ПСП).

В ПСО входят управляющая программа, операторы, осуществляющие трансляцию строк, оператор печати и различные вспомогательные операторы, используемые во время трансляции.

В ПСП входят стандартные программы, которые широко используются большинством операторов из ПСО и оформление которых в виде операторов существенно снизило бы скорость трансляции.

Во время трансляции используется оперативная память в 32 листа (1 лист — 1024 48-разрядных слова) и память на МБ в размере 64 трактов (1 тракт — 1024 48-разрядных слова).

Управляющая программа (УП) сделана в виде стандартного оператора номер 0040 системы программирования Авто-Аналитик. Обращение к VII имеет вид:

0040          ТИП           $A_1$            $A_2$

Здесь  $A_1 \times 10\,000 + A_2$  — адрес начала входного текста. Распишем восьмеричную тетраду ТИП в двоичной форме:



ТИП =  $V_1V_2V_3$        $V_4V_5V_6$        $V_7V_8V_9$        $V_{10}V_{11}V_{12}$

Если  $V_{12} = 1$ , то печатаются формулы,

$V_{11} = 1$ , —,— массивы,

$V_{10} = 1$ , —,— параметры системы,

$V_9 = 1$ , —,— ручные программы,

$V_8 = 1$ , —,— операторы,

$V_7 = 1$  — оператором АРАП печатается транслированный эквивалент формул,

$V_6 = 1$  — печатаются транслированные массивы,

$V_5 = 1$  — —,— —,— параметры системы,

$V_4 = 1$  — —,— —,— ручные программы,

$V_3 = 1$  — —,— —,— операторы,

$V_2 = 1$  — печатается таблица меток и слов (см. ниже) каждого оператора входного текста,

$V_1 = 1$  — печатается таблица операторов (см. ниже).

Адрес начала входного текста должен быть больше или равен  $6000_8$ . В противном случае входной текст не транслируется, печатается текст

**АДРЕС НАЧАЛА ВХОДНОГО ТЕКСТА МЕНЬШЕ 6000**  
и происходит прерывание.

Распределение памяти машины производится самим пользователем. При этом необходимо учесть, что 0, 1, 2 листы ОЗУ БЭСМ-6 заняты БСП;  $36_8$  и  $37_8$  листы ОЗУ — под ПСП и лист обмена с внешними запоминающими устройствами. Длина РП должна быть не меньше  $30000_8$  ячеек.

Если обращение к управляющей программе произошло не из программы пользователя, то память БЭСМ-6 на время трансляции распределяется следующим образом:  $A^{\circ}_{\text{орп}} = 46000_8$ ;  $A^{\circ}_{\text{ОАС}} = A^{\circ}_{\text{ТП}} = A^{\circ}_{\text{срп}} = 72400_8$ ;  $A^{\circ}_{\text{МО}} = 72000_8$ ; 32 тракта МБ отводятся под БАС и БРП и еще 32 тракта МБ отводятся под транслированные массивы и ручные программы, которые во время трансляции засылаются на МБ.

Рассмотрим работу транслятора в целом.

На время трансляции на ОРПІІ заводятся таблицы:

а) таблица операторов (ТО);

б) таблица массивов (ТМ).

Каждый элемент ТО состоит из двух ячеек: в первой ячейке находится имя оператора, во второй ячейке находится восьмеричный номер, сопоставленный транслятором данному имени. Каждый элемент ТМ также состоит из двух ячеек:

в первой находится имя массива, во второй — адрес ячейки, в которую заносится первый элемент массива.

Управляющая программа определяет тип строки и передает управление на оператор, осуществляющий трансляцию данной строки. Сначала транслируются формулы, параметры системы, ручные программы, массивы и каждому оператору, именем которого служит слово, сопоставляется свободный номер; затем транслируются операторы.

Такой способ трансляции вызван тем, что в операторах могут использоваться элементы массивов. Поэтому до начала трансляции операторов необходимо выявить адреса этих элементов в памяти ЭВМ.

Номер оператора может находиться в следующих пределах:  $100_8 \leq N \leq 777_8$ . Если в адресном столе прописано больше  $700_8$  операторов, то печатается текст

**В ПРОГРАММЕ БОЛЬШЕ 700 ОПЕРАТОРОВ**  
и происходит прерывание.

Заметим, что во входном тексте формулы с номерами меньше  $1000_8$  и операторы, именами которых являются восьмеричные номера, должны быть расположены раньше любого из операторов, именем которого является слово. В противном случае может возникнуть следующая ситуация: транслятор присвоит транслируемому оператору, именем которого является слово, номер  $100_8 \leq N \leq 777_8$ ; затем во время трансляции формулы (оператора) с этим же номером  $N$  транслированный ранее оператор будет аннулирован в ОЗУ или МБ.

По окончании трансляции строки в зависимости от значения аргумента ТИП в обращении к УП происходит печать строки или ее транслированного эквивалента.

Если пользователем были заданы какие-либо параметры системы, то по окончании трансляции входного текста происходит распределение памяти согласно этим параметрам.

Если во входном тексте не было ошибок, то при наличии соответствующих указателей происходит передача управления на запуск системы или в противном случае выход из УП.

Если в какой-либо строке входного текста была допущена ошибка, то печатается информация о характере ошибки, кусок текста, в котором была допущена ошибка, и сама строка. Дальнейшая трансляция строки не производится. В случае, когда строка, в которой допущена ошибка, является оператором, печатается предложение, в котором была допущена ошибка, имя оператора и транслятор переходит к трансляции следующего предложения.



Рассмотрим назначение и работу важнейших стандартных программ и операторов из пакета транслятора.

К СП группы чтения относятся СП чтения символа входного текста (ЧВС), СП чтения слова (ЧВСЛ), СП чтения восьмеричных чисел (ЧВС8). Если при чтении встретился символ — (пробел), то он пропускается. Поэтому пробел можно использовать для вычеркивания какой-нибудь части входного текста.

ЧВС производит чтение по настройке символа входного текста в рабочие ячейки транслятора.

ЧВСЛ читает по настройке в рабочую ячейку транслятора до тех пор, пока не будет считан символ, не являющийся буквой или цифрой. Так как слово идентифицируется по первым шести символам, то в рабочую ячейку считывается максимум шесть символов.

ЧВС8 производит чтение по настройке восьмеричных чисел в рабочую ячейку транслятора. Чтение продолжается до тех пор, пока не будет считан символ, не являющийся восьмеричной цифрой, или пока данное восьмеричное число больше  $2^{48}$ . Последний случай квалифицируется как ошибка в написании входного текста и, следовательно, печатается информация об этом.

Ряд стандартных программ служит для перевода из символов входного языка на внутренний язык Авто-Аналитика. Эти СП используются при трансляции констант.

Оператор «Трансляция чисел» читает по настройке число входного текста, производит перевод этого числа из десятичной системы в двоичную и в рабочей ячейке указывает, какое это число — вещественное или целое. На выходе настройка указывает на символ входного текста, непосредственно следующий за данным числом.

Оператор «Трансляция формул» производит перевод формул, записанных на входном языке, на внутренний язык, при этом используется вспомогательный оператор ПТФ, который собственно и осуществляет перевод тела строки, являющейся формулой. Этот же оператор осуществляет трансляцию предложений второго рода, т. е. трансляцию формул, вводимых в программу оператора.

Оператор «Трансляция параметров системы» определяет, является ли имя строки идентификатором какого-либо параметра системы, затем транслирует этот параметр. По окончании трансляции входного текста УП устанавливает распреде-

ление памяти и шкалы коммутативных и ассоциативных связей согласно этим параметрам.

Для трансляции ручных программ и массивов отводится 36<sub>8</sub> лист ОЗУ БЭСМ-6, который впредь будем называть листом обмена (ЛО). Ручные программы и массивы транслируются на ЛО в виде списка, состоящего из: а) заголовка списка; б) тела списка. Заголовок списка есть ячейка, имеющая вид:

в 48—33 разрядах —  $l$  — длина списка,

в 32 разряде —  $m$  — вид списка,  $m = 0$  — массив,  $m = 1$  — ручная программа, в 15—1 разрядах —  $A$  — адрес начала массива или ручной программы в ОЗУ БЭСМ-6.

Телом списка является транслированная ручная программа или массив.

Когда ЛО полностью заполнится, он переписывается на МБ. По окончании трансляции входного текста все транслированные ручные программы и массивы переписываются с МБ в ОЗУ согласно адресам  $A$ , указанным в заголовках соответствующих списков.

Трансляцию операторов осуществляет оператор под тем же названием «Трансляция операторов» (ТОП). Рассмотрим, как работает ТОП.

Так как максимальная длина оператора равна 7777<sub>8</sub>, то для трансляции какого-либо оператора ТОП отводит часть ОЗУ с ячейки  $A_{236}$  по ячейку  $A_{236} + 7777_8$ , причем с ячейки  $A_{236}$  в сторону возрастания адресов ТОП пишет программу реализующую данный оператор (допустим, до ячейки  $B$ ), а с ячейки  $A_{236} + 7777_8$  в сторону уменьшения адресов (допустим, по ячейку  $C$ ) ТОП отводит под слова, числа, константы и рабочие ячейки, используемые в этом операторе. По окончании трансляции оператора ТОП стыкует массивы ячеек  $C \div A_{236} + 7777_8$  с массивом ячеек  $A_{236} \div B$  и полученную таким образом программу переписывает на СРМ и прописывает в адресном столе.

Во время трансляции оператора должно выполняться условие  $B < C$ . В противном случае печатается текст ДЛИНА ОПЕРАТОРА  $> 7777$ , прекращается трансляция данного оператора и печатается строка, задающая этот оператор.

На время трансляции оператора ТОП заводит таблицу слов и меток. Каждый элемент этой таблицы состоит из двух ячеек: в первой из них находится слово (метка), во второй ячейке находится относительный адрес, сопоставленный дан



ному слову (соответственно относительный адрес данной метки).

Первоначально ТОП просматривает все предложения и каждую метку, соотнесенную предложению или какой-либо ячейке кодового участка, заносит в таблицу слов и меток. Если двум различным предложениям или двум различным ячейкам в кодовом участке соотнесена одна и та же метка М, то ТОП квалифицирует этот случай как ошибку и печатает текст В ОПЕРАТОРЕ ДВЕ ОДИНАКОВЫХ МЕТКИ М; затем прекращается трансляция данного оператора и печатается строка, задающая этот оператор.

На время трансляции операторов создается информационный массив. Этот массив служит информацией о том, какие ячейки транслированной программы оператора являются командами, а какие — константами. Такая информация необходима ТОПу для корректировки адресов меток, рабочих ячеек и ячеек, содержащих константы и числа, при стыковке массива ячеек  $C \div A_{236} + 7777_8$  с массивом ячеек  $A_{236} \div B$  транслированной программы оператора.

После занесения всех меток в таблицу слов и меток ТОП начинает собственно трансляцию операторов. Из [1] известно, что строка, задающая оператор, состоит из последовательности предложений трех родов. ТОП определяет, какого рода является каждое предложение, и передает управление на подпрограмму, транслирующую это предложение. Если данное предложение является предложением второго рода, то его трансляцию осуществляет оператор ПТФ. Если же данное предложение является предложением первого рода, то ТОП определяет, каким оператором оно задается (логическим, внутренним или оператором из БСО) и передает управление на трансляцию этого оператора. При трансляции предложений первого рода широко используется оператор «Трансляция выражений» (ТВ). Оператор ТВ осуществляет трансляцию выражений. Его работу в самом общем виде можно описать так: ТВ переписывает данное выражение в бесскобочную запись особого вида — Польскую инверсную запись [2]. Затем, исходя из этой записи, транслирует данное выражение. При этом если в написании выражения была допущена ошибка, то печатается характер ошибки и само выражение.

Для печати строк, операторов, меток и слов используется оператор «Печать». В зависимости от значения аргумента ТИП в обращении к УП оператор «Печать» печатает строку

того или иного типа, ее транслированный эквивалент, ту или иную таблицу.

Так как набор строк может модифицироваться пользователем, то необходимо было предусмотреть возможность присоединения к ТАА новых блоков. Это легко сделать: достаточно новый блок оформить в виде стандартного оператора системы программирования Авто-Аналитик и включить в ПСО.

В заключение отметим, что качество транслированных ТАА программ можно сравнить с качеством программ, написанных программистом без особой тщательности.

#### ЛИТЕРАТУРА

1. Е. А. Арайс, Г. В. Сибиряков. Авто-Аналитик. НГУ, 1974.
  2. Б. Ренделл, Л. Рассел. Реализация АЛГОЛа-60, М., Изд-во «Мир», 1967.
-



## ЯЗЫК МАРС

Е. А. АРАИС, А. В. ШУТЕНКОВ

Входной язык является составной частью системы программирования МАРС (моделирование и анализ радиоэлектронных схем). Теоретические основы данной системы приведены в работах [1, 2]. Язык позволяет упростить процесс подготовки исходной информации, а также программирование новых операторов системы и комплексацию уже имеющихся программ. Во время разработки входного языка системы МАРС были учтены недостатки находящегося в эксплуатации варианта входного языка [3]. В связи с этим были расширены старые объекты входного языка, либо введены новые.

### § 1. Основные понятия

Вся исходная информация записывается при помощи символов, которые разбиты на три группы: буквы, цифры и ограничители.

В качестве букв используются заглавные буквы русского и латинского алфавитов. Буквы применяются при построении идентификаторов, слов, встречаются в тексте комментария.

К группе символов, названных цифрами, относятся 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, причем цифры с 0 по 7 называются восьмеричными.

Ограничители делятся на знаки операций и разделители. Среди знаков операций различаются:

знаки арифметических операций:

$+$  (сложение),  
 $-$  (вычитание или знак минус),

$\times$  (умножение).

$/$  (деление),

$\uparrow$  (возведение в степень);

знаки логических операций:

$\vee$  (логическое сложение);

$\wedge$  (логическое умножение);

знаки операций отношения:

$=$  (знак равенства);

$<$  (знак меньше);

знак функциональной зависимости  $*$  и знак операции сдвига  $\div$ .

Употребление этих знаков в большинстве случаев очевидно. Они служат для построения выражений.

К числу разделителей относятся следующие символы:  
, . <sup>10</sup> : ; — ' ( ) [ ] |

Символ — (пробел) во время трансляции не воспринимается и может быть использован для облегчения чтения программ.

Разделитель ' (кавычка) служит для выделения комментария. В начале и в конце текст комментария ограничивается двумя кавычками. Комментарии используются для пояснения текста программы и не влияют на ее работу. Текст комментария может состоять из произвольного набора символов, за исключением " (кавычек), и располагаться в любом месте программы.

Использование остальных разделителей станет ясно из дальнейшего описания входного языка.

## 1. Числа и константы

Числа делятся на условные (целые), восьмеричные и вещественные. Запись чисел на входном языке мало отличается от обычной записи.

**Определение 1.** Произвольная конечная последовательность цифр со знаком минус или без знака называется условным (целым) числом.

Условные числа при трансляции переводятся из десятичной в двоичную систему счисления и записываются в младшие разряды ячеек. Отрицательное условное число, то есть со знаком минус, переводится в код дополнительный к положительному значению числа.

**Определение 2.** Произвольная конечная последовательность восьмеричных цифр называется восьмеричным числом.



Восьмеричные числа используются при формировании констант, записи номеров ячеек, номеров стандартных операторов.

Определение 3. Вещественным числом называется конечная последовательность символов  $A_1 A_2 \dots A_n$  со знаком минус или без знака, удовлетворяющая четырем условиям:

(1). Один из символов  $A_i$  ( $1 \leq i \leq n$ ) является разделителем . (точка) или  $_{10}$  (подстрочная десятка). При наличии двух этих разделителей точка расположена левее  $_{10}$ .

(2). Если разделитель  $_{10}$  не является первым символом числа, не считая знака, то наличие точки в числе обязательно.

(3). После разделителя  $_{10}$  должно быть условное число, не превышающее по абсолютной величине 19.

(4). Символы, отличные от знака минус и указанных в (1) разделителей, есть цифры.

В записи вещественного числа разделитель точка играет роль десятичной запятой, разделяющей целую и дробную части числа. Условное число после разделителя  $_{10}$  обозначает десятичный порядок.

Определение 4. Константой называется конечная последовательность символов  $A_1 A_2 \dots A_n$ , из которых  $A_1$  и  $A_n$  являются разделителем | (вертикальная черта), остальные  $A_i$  ( $i = 2, 3 \dots, n-1$ ) есть восьмеричные цифры.

Во время трансляции константы записываются в младшие разряды ячеек, старшие разряды обнуляются. Если в записи константы присутствует более 15 восьмеричных цифр, то в ячейку заносятся первые 15 цифр считая справа налево.

Примерами записи чисел и констант являются последовательности

192 — 29 0 5.123 .023<sub>10</sub> — 3

10<sup>5</sup> |7777| |67000|

## 2. Слова и выражения

Определение 5. Словом называется произвольная конечная последовательность букв и цифр, начинающаяся с буквы.

При составлении программ слова используются для обозначения идентификаторов постоянных и переменных величин, массивов, меток, функций и так далее. Два слова тождественны, если они содержат одинаковое количество символов и символы, стоящие на одном и том же месте в этих словах, совпадают.

## Примеры записи слов:

МЕТКА 1

ППВ 2

А 24

В языке МАРС вводятся обозначения для некоторых основных функций:  $\text{mod}$ ,  $\ln$ ,  $\sin$ ,  $\cos$ ,  $\exp$  (запись на входном языке соответственно  $\text{MOD}$ ,  $\text{LN}$ ,  $\text{SIN}$ ,  $\text{COS}$ ,  $\text{EXP}$ ). Для разделения идентификатора функции и ее аргумента служит знак функциональной зависимости  $*$ .

При помощи знаков операций и скобок из чисел, констант и идентификаторов различных величин составляются выражения. Правила записи выражений мало отличаются от общепринятых.

Будем называть выражение арифметическим, если в его построении участвуют какие-либо операции, за исключением операций отношения. Два арифметических выражения, связанные одним знаком операции отношения, будем называть условным выражением.

При реализации программы все выражения вычисляются до числа. Вследствие этого они могут принимать либо вещественное, либо условное значение. Выражения, которые должны иметь условное (целое) значение, необходимо записывать в квадратных скобках. Знаки операций, встречающиеся в квадратных скобках, реализуются транслятором как операции над условными числами.

Порядок выполнения действий в выражениях, не содержащих скобок, зависит от старшинства операций. Ниже приведен список операций в порядке возрастания их старшинства. Операции, расположенные в одной строке, равноправны. Очередность выполнения таких операций, например, сложения и вычитания, определяется порядком их записи в выражении (слева направо).

$:=$

$<$

$=$

$\vee$

$\wedge$

$+$

$-$  (вычитание)

$\times$

$/$

$-$  (изменение знака)

$\uparrow$

$*$

$\div$



Соответствующей расстановкой скобок можно задать необходимый порядок выполнения действий. Символ — (минус) применяется для обозначения как операции вычитания, так и операции изменения знака. Операция изменения знака выражения является одноместной, все остальные — двуместные операции.

Описание использования знака  $=$  (оператор присваивания) приведено в § 5. Выражения, построенные с помощью операции сдвига, имеют вид

$$A \div [B]$$

При этом сдвигу подвергается весь код, определяющий значение выражения  $A$ . Величина и направление сдвига определяется условным числом, либо выражением  $B$ , принимающим условное значение. При положительном значении условного числа происходит сдвиг кода влево, при отрицательном — вправо. Использование остальных знаков операций общезвестно.

Для знака функциональной зависимости  $*$  изменен порядок выполнения операций. В выражениях вида

$$F1 * F2 * A,$$

где  $F1$  и  $F2$  — идентификаторы каких-либо функций,  $A$  — выражение, вычисляется первой функция  $F2$ , затем  $F1$ . Такое отклонение от обычного порядка выполнения операций позволяет избежать записи лишних скобок, так как необходимо было бы писать  $F1 * (F2 * A)$ .

Приведем примеры записи выражений.

$$R \times (-LN * A)$$

$$B \div 14 = N \div [-(A + C)]$$

$$MOD * SIN * (A + C \uparrow 3./B)$$

$$A < B \wedge |7777|$$

### 3. Структура исходной информации

Вся информация, вводимая в ЭВМ, представляет собой совокупность строк.

Определение 6. Строкой называется последовательность символов входного языка МАРС, имеющая вид

$$\langle \text{заголовок строки} \rangle : \langle \text{тело строки} \rangle **$$

Здесь и далее в угловках  $\langle \rangle$  выделяется некоторая структура входного языка. Заголовком является слово,

как правило, отражающее смысл объекта, вводимого строкой. Заголовок отделяется от прочих символов разделителем: (двоеточие). Признаком конца информации в строке являются два ограничителя \*\*. Телом строки может быть, вообще говоря, любая последовательность символов, отличных от символов конца строки. Системой в настоящее время допускаются строки со следующими заголовками: АРП, ЛРП, АИС, АСХЕМ, LСХЕМ, СХЕМА, МАССИВ, ОПЕРАТОР, ПОДПРОГРАММА, ПРОГРАММА, PROGRAM, ВХОД, READ, END.

## § 2. Параметры системы

При решении задач исследования РЭС обычно велик объем используемой памяти ЭВМ. Автоматическое распределение памяти поэтому представляется нецелесообразным.

Под параметрами системы будем понимать совокупность констант, задающих распределение памяти. Вводятся параметры при помощи строк вида:

<название параметра> : <значение параметра> \*\*  
В данном случае заголовком строки является название параметра.

Перечислим заголовки строк и дадим описание соответствующих значений параметров. В скобках указаны стандартные значения параметров.

АРП — адрес начала рабочего поля стандартных операторов системы Ридер, значение которого, как и значения всех последующих параметров, указывается восьмеричным числом. (Стандартное значение параметра — 1000).

ЛРП — количество ячеек, отводимых под рабочее поле (2000).

АИС — адрес начала рабочего поля интерпретирующей системы ИС-22 (6000).

АСХЕМ — адрес начала массива ячеек, где следует расположить после трансляции информацию о вводимых схемах (3000).

LСХЕМ — количество ячеек в этом массиве (5000).

При отсутствии во входной информации строк указанного вида в систему заносятся стандартные значения параметров. Параметрами задается распределение 1-го куба оперативной памяти ЭЦВМ.

Пример. Зададим распределение памяти строками

АРП : 1600\*\*

ЛРП : 1400\*\*



АИС : 6200\*\*

АСХЕМ : 3400\*\*

LSХЕМ : 2000\*\*

Согласно этим строкам транслятор установит начало рабочего поля в оперативной памяти ЭЦВМ с ячейки 1600 1-го куба МОЗУ, длину рабочего поля — 1400 ячеек. Рабочее поле ИС-22 будет расположено с ячейки 6200. Для размещения информации о схемах отводится массив длиной в 2000 ячеек. Начало этого массива расположено в ячейке с номером 3400. Если объем информации о схемах превысит отведенные 2000 ячеек, то произойдет прерывание работы системы.

### § 3. Схемы

Информация о схеме вводится в ЭЦВМ строкой вида  
СХЕМА : <название схемы>;

0 : <идентификатор модели> (<значения параметров компонента>)

В (<номера ветвей>) N (<номера узлов>);

1 : <идентификатор модели> (<значения параметров компонента>)

В (<номера ветвей>) N (<номера узлов>);

К : <идентификатор модели> (<значения параметров компонента>)

В (<номера ветвей>) N (<номера узлов>)\*\*.

Тело строки СХЕМА представляет собой последовательность предложений, отделенных друг от друга разделителем: (точка с запятой).

Первое предложение состоит из названия схемы, которым может быть произвольное слово.

Каждое последующее предложение несет в себе информацию об элементах схемы. Перечисляются номер элемента по порядку, идентификатор модели данного элемента, числовые значения его параметров. Указываются также номера ветвей, образованных выводами элемента, с направлением тока в них, и номера узлов, с которыми соединен элемент. Порядок перечисления элементов произвольный.

Порядковый номер элемента выражается условным числом (без знака) и отделяется от остальных символов предложения разделителем : (двоеточие). Перечисление начинается с нуля. Номера элементов используются только для контроля записи информации о схеме.

Идентификатор модели элемента есть слово (название), под которым эта модель включена в библиотеку системы. После идентификатора модели в круглых скобках перечисляются значения параметров элемента. В случае, когда используется модель элемента без параметров, после идентификатора модели, вместо круглых скобок с параметрами элемента ставится разделитель , (запятая).

Числовые значения параметров элемента записываются в виде равенств. В левой части равенства указывается название параметра и его номер. Каждому параметру в модели элемента отведена определенная ячейка. Номер этой ячейки относительно начала модели (это и есть номер параметра) задается восьмеричным числом. Считается, что ячейки в модели перенумерованы в порядке 0, 1, 2, ... Последнее перед знаком равенства восьмеричное число воспринимается как номер параметра. В качестве названия параметра указывается произвольное слово, но оно может и отсутствовать. Тогда в левой части равенства будет находиться только номер параметра. Название параметра рекомендуется писать для наглядности вводимой информации.

Правая часть равенства представляет собой значение параметра элемента схемы и может быть выражена произвольным числом или константой.

Равенства, задающие значения параметров, отделяются друг от друга разделителем, (запятая).

При записи параметров наиболее часто встречающихся элементов схем наличие номера параметра и его названия не обязательно. В этом случае указывается только значение параметра. К таким относятся элементы, модели которых обозначены идентификаторами *R*, *L*, *C*.

Информация о ветвях, образуемых выводами элемента, задается в круглых скобках, перед которыми ставится буква *B*. Номера ветвей перечисляются в виде условных чисел через разделитель, (запятая). Считается, что условным числом без знака задается положительное направление тока в данной ветви, а условным числом со знаком минус — отрицательное.

Аналогично информации о ветвях перечисляются номера узлов, с которыми соединен элемент. В этом случае указателем перечисления узлов является наличие перед круглыми скобками буквы *N* (вместо *B*), а условные числа не должны иметь знака минус.

Для каждого элемента порядок перечисления номеров



узлов должен соответствовать порядку записи информации о номерах ветвей.

Рассмотрим пример занесения информации о схеме. Для выпрямителя (рис. 1) она будет иметь вид:

СХЕМА: ВЫПРЯМИТЕЛЬ;

0: ES ( $H4 = 0,5$ ,  $W5 = 50$ ,  $\Phi 6 = 0$ ) В ( $-0,0$ ) N ( $1,0$ );

1: R ( $100$ .) В ( $-2,2$ ) N ( $2,0$ );

2: ДБ ( $T10 = 300$ ,  $12 = 10 - 4$ ,  $13 = 2 \cdot 10$ ,  $6,14 = 1,2$ ,  
 $15 = 10 - 3$ ,  $11 = 30$ ,  $21 = -30$ ,  $F23 = 1$ .) В ( $-$   
 $-1,1$ ) N ( $1,2$ )\*\*

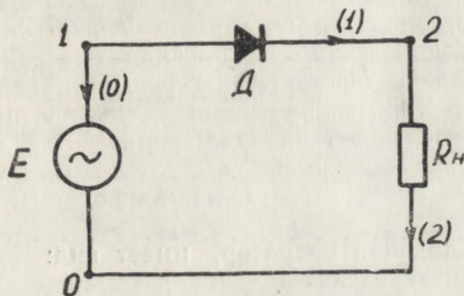


Рис. 1

#### § 4. Массивы

Входной язык системы допускает ввод одномерных массивов с помощью строк вида

МАССИВ: <название массива>, <адрес>; <элемент массива>, ..., <элемент массива>\*\*.

Элементами массива могут быть числа и константы. Разделителем элементов массива при записи является запятая.

Так как автоматического распределения памяти не производится, то следует указывать адрес, начиная с которого будут расположены элементы массива. Адрес записывается восьмеричным числом.

Названием массива является произвольное слово. Для указания какого-либо элемента массива служит переменная с индексом. Запись переменной с индексом имеет вид

<название массива> [ $i$ ],

где  $i$  — условное число или выражение, определяющее индекс элемента массива. Элементы в массиве нумеруются, начиная с нуля.

Название массива можно опускать, тогда строка примет вид:

**МАССИВ**: <адрес>; <элементы массива>\*\*.

Переменную с индексом можно также использовать для образования новых массивов и работы с ними. Начало таких массивов задается присвоением значения (адреса массива) какой-либо переменной. В этом случае идентификатор переменной выполняет роль названия массива.

## § 5. Операторы

Программа, записанная на входном языке, транслируется в стандартный оператор интерпретирующей системы Ридер. При решении какой-либо задачи или класса задач можно выделить отдельно общие этапы решения, некоторые частные задачи. Алгоритмы решения таких частных задач целесообразно оформлять на входном языке в виде операторов. Достаточно обширная библиотека операторов позволяет облегчить процесс составления программ для решения новых задач.

Строка, задающая оператор, имеет вид:

**ОПЕРАТОР**: <название оператора>; <предложение>; ...; <предложение>\*\*.

Названием оператора может быть либо произвольное слово, либо восьмеричное число, обозначающее номер этого оператора. После трансляции оператор включается в библиотеку системы Ридер под любым свободным номером, если указано название оператора, или заданным номером.

Во избежание путаницы в терминологии будем называть в дальнейшем последовательность предложений, реализующих данный стандартный оператор, программой оператора или просто программой. Каждое предложение состоит из названия оператора входного языка, его аргументов и заканчивается разделителем; (точка с запятой). Внутри оператора аргументы перечисляются через запятую. Таким образом, предложение имеет вид:

<название оператора>, <аргумент>, ..., <аргумент>;

Операторы входного языка являются указаниями о выполнении определенных действий, с их помощью организуется логика программы. Имеются следующие операторы:

- 1) **АРГ** оператор выдачи аргументов,
- 2) **:** оператор присваивания,
- 3) **НА** оператор безусловного перехода,



- 4) КОНЕЦ оператор выхода из подпрограммы,
  - 5) ЕСЛИ оператор условного перехода,
  - 6) КОД оператор КОД,
  - 7) ТЕКСТ оператор печати текста,
  - 8) СТОП оператор останова,
  - 9) ЦИКЛ оператор цикла,
  - 10) ВЫХОД оператор выхода из программы.
- Рассмотрим подробнее их работу и использование.

## 1. Оператор выдачи аргументов

Так как в описываемой системе в основном используется операторная структура выполнения программ, то необходима передача информации из одного стандартного оператора в другой. Этой цели служит предложение вида

АРГ, <слово>, ..., <слово>;

которое обычно ставится в начале программы оператора. Считается, что аргументы стандартного оператора могут быть расписаны либо поячеечно (например, аргумент число занимает целую ячейку), либо по позициям. Ячейка ЭЦВМ М-220, состоящая из 45 разрядов, разбивается на четыре позиции соответственно с 45-го по 37-й, с 36-го по 25-й, с 24-го по 13-й, с 12-го по 1-й разряды. Разряды в ячейке нумеруются справа налево. Легко видеть, что позиции неравнозначны. Первая позиция на три разряда короче остальных. Такое разбиение было принято с целью экономии памяти машины, более полного использования ячеек, а также для удобства стыковки новых программ с ранее написанными стандартными операторами системы Ридер.

Оператор АРГ осуществляет запись аргументов стандартного оператора. Каждому аргументу в программе отводится ячейка, и все записанные по позициям аргументы переносятся в младшие разряды соответствующих ячеек. Каждой такой ячейке ставится в соответствие идентификатор — слово, взятое из обращения к оператору АРГ. Если в обращении какое-либо слово состоит из одной буквы Н, то это значит, что соответствующая позиция в массиве аргументов безразлична. В этом случае ячейка в программе оператора не отводится. Признаком того, что аргумент занимает целую ячейку, является буква Z перед словом.

Назначение и работу оператора АРГ рассмотрим на примере обращения к какому-либо оператору. Пусть обращение к оператору СТРОФ записано в виде (см. ниже обращение к стандартным операторам).

СТРОФ, 30, 4095, [430], ZM [2], 0, 3, 7.,

В программе оператора СТРОФ должно быть обращение к оператору АРГ, например, вида

АРГ, А, В, С, Z число, Н, С4, ZK;

Оператор АРГ производит запись аргументов из обращения в тело оператора СТРОФ и сопоставит переменным А, В, С и С4 ячейки, в младших разрядах которых соответственно записаны восьмеричные коды 36, 7777 (десятичные условные числа 30 и 4095 переводятся в восьмеричную систему), 430 и 3. Переменной ЧИСЛО будет присвоено значение второго элемента массива с названием М. Переменная К получит вещественное значение 7. Буква Н в обращении к оператору АРГ означает, что соответствующая позиция для оператора СТРОФ безразлична.

## 2. Оператор присваивания

Данный оператор служит для вычисления значения выражения и присваивания этого значения одной или несколькими переменным.

Оператор присваивания состоит из левой части, символов  $:=$  (знак присваивания) и арифметического выражения, образующего правую часть оператора. В левой части указывается идентификатор переменной. Если необходимо присвоить одно и то же значение нескольким переменным, то их следует перечислить в круглых скобках, разделяя запятыми.

Например, предложение :

(А, РЕЗУЛЬТАТ [5])  $:=$  В + С/В;

вычислит значение выражения  $В + С/В$  и присвоит его переменной А и пятому элементу массива РЕЗУЛЬТАТ.

## 3. Метки и операторы перехода

Обычно в программе операторы выполняются в естественной последовательности, то есть в порядке следования предложений. Операторы перехода используются для изменения этой последовательности и указания места, с которого должна выполняться программа дальше. Такое указание осуществляется с помощью меток. Меткой может быть произвольное слово. Для того, чтобы пометить любое предложение, достаточно поставить перед ним метку, отделив ее от самого предложения разделителем: (двоеточие). Примерами меток могут служить слова



## МЕТКА 1 ВЫХОД ИЗ ПРОГРАММЫ СУММА

Оператор безусловного перехода состоит из указательного слова НА и метки, следующей за ним через запятую:

НА, <метка> ;

Такой оператор вызывает всегда переход на предложение с указанной меткой.

Изменение порядка выполнения программы при выполнении некоторых условий производится оператором условного перехода, обращение к которому имеет вид:

ЕСЛИ, <условное выражение> ,..., <условное выражение> , ТО, <метка>;

Если отношения < (меньше) или = (равно) выполняются во всех условных выражениях, перечисленных в обращении к оператору, то происходит переход на заданную метку. При нарушении хотя бы одного условия будет выполняться предложение, следующее за обращением к условному оператору.

Примеры операторов перехода.

НА, М5;

ЕСЛИ,  $A < B$ , ТО, ВЫХОД;

ЕСЛИ,  $[5 - CxV] \wedge |37| = A$ ,  $B = 5$ ,  $E <_{10} - 3$ , ТО, СТОП;

Оператор НА используется также для обращения к подпрограммам, включенным в программу оператора.

### 4. Организация подпрограмм

Подпрограммой будем называть последовательность предложений, первое из которых помечено меткой специального вида, начинающейся с символов ПП. Последним должно быть предложение

КОНЕЦ;

Обращение к подпрограммам производится предложением вида

НА, <метка>;

Если аргументом оператора НА является обычная метка, то он выполняется как оператор безусловного перехода. Если же указана метка специального вида, то есть начинающаяся

с символов ПП, то происходит обращение к подпрограмме, помеченной данной меткой. Оператор КОНЕЦ (выход из подпрограммы) осуществляет возврат на предложение, следующее за оператором обращения к этой подпрограмме.

Метками специального вида следует пользоваться только при оформлении подпрограмм. Случаи, когда в программе имеется предложение с такой меткой, но нет оператора КОНЕЦ, или, наоборот, имеется оператор КОНЕЦ, но нет предложения со специальной меткой, воспринимаются транслятором как ошибки.

## 5. Оператор КОД

Для включения кодов машинного языка ЭЦВМ М-220 в программу оператора служит предложение вида

КОД, <кодовый участок>;

Запись кодового участка сходна по форме с записью программ на Автокоде ЭЦВМ М-220. В адресную часть команд кодового участка могут входить слова, являющиеся идентификаторами переменных и метками. Метки заключаются в круглые скобки. Выражения, а также переменные с индексами писать запрещается. Части команды (код операции и адреса) и сами команды разделяются запятыми. Код операции и каждый адрес команды состоит соответственно не более чем из трех и четырех восьмеричных цифр. Незначащие слева нули в частях команд можно не указывать.

Любая команда кодового участка может быть помечена. Для этого достаточно поставить метку, отделив ее от команды разделителем: (двоеточие). Недостающие до полной ячейки части в последней команде считаются нулевыми.

Оператором КОД удобно пользоваться для написания программ в условных адресах.

Рассмотрим пример. Пусть требуется получить значение выражения  $A + B/C$  и присвоить его переменной А. После чего следует перейти на предложение с меткой М1.

В качестве рабочей будем использовать ячейку, помеченную меткой R. Тогда, используя оператор, КОД можно записать

КОД, 4, В, С, (R),  
1, А, (R), (R),  
56, (R), (M1), А,  
R: ;



## 6. Оператор печати текстовой информации

Предложение вида

ТЕКСТ, <текстовая информация>;

осуществляет печать текста, указанного в обращении. Текст может состоять из произвольного набора символов, исключая точку с запятой, две звездочки и кавычки.

## 7. Оператор останова

Обращение к оператору имеет вид

СТОП, <текст>;

Оператор используется для прерывания работы программы. На печать выдается информация

## РАБОТА СИСТЕМЫ ПРЕРВАНА ОПЕРАТОРОМ СТОП.

Далее распечатывается текст, являющийся аргументом оператора. Аналогично оператору ТЕКСТ текстовая информация может состоять из произвольного набора символов, за исключением точки с запятой, двух звездочек и кавычек.

## 8. Оператор цикла

Обращение к оператору имеет вид

ЦИКЛ, <метка>, <слово>, <выражение>;

Оператор выполняет последовательность предложений, следующих за ним, до предложения с указанной меткой. Помеченное предложение в состав цикла не включается. Цикл повторяется  $K$  раз, где  $K$  — значение выражения, вычисленное перед организацией цикла. Значением выражения должно быть условное число.

Во время работы оператора слово (параметр цикла) принимает значения  $1, 2, \dots, K$ . В теле цикла можно изменить значение параметра, что соответственно скажется на числе повторений цикла. Изменение значений слов, входящих в выражение, на число повторений цикла не влияет.

Оператор цикла можно следующим образом записать с помощью других операторов входного языка:

$K := \langle \text{выражение} \rangle;$

$\langle \text{слово} \rangle := 0;$

НА, МЕТКА2;

Метка 1:  $\langle \text{слово} \rangle := [\langle \text{слово} \rangle + 1];$  <тело цикла>;

МЕТКА 2: ЕСЛИ,  $[\langle \text{слово} \rangle < K],$  ТО, МЕТКА 1;

Приведем примеры обращений к оператору цикла.  
ЦИКЛ, МЗ, В,  $[A + B]$ ;  
ЦИКЛ, МЕТКА, К, 10;  
ЦИКЛ, L, I,  $|103|$

## 9. Обращение к стандартным операторам

Предложение вида  $\langle \text{название стандартного оператора} \rangle$ ,  $\langle \text{аргумент} \rangle, \dots, \langle \text{аргумент} \rangle$ ; используется для обращения к стандартным операторам системы Ридер. При этом указывается название стандартного оператора, под которым он включен в библиотеку системы, и его аргументы. Вместо названия можно писать восьмеричный номер этого оператора.

Аргументы перечисляются через запятую и записываются по позициям (аналогично оператору АРГ). Если перед аргументом указана буква Z, то он занимает целую ячейку. Считается, что записанное в качестве аргумента вещественное число также занимает целую ячейку и символ Z можно не указывать. Названия схем и массивов заменяются при трансляции адресами начала этих объектов в памяти машины. Встречающиеся выражения вычисляются, и их значения, если нет признака Z, заносятся из младших разрядов ячейки результата в соответствующую позицию аргумента.

Примерами обращений к стандартным операторам являются следующие предложения:

ТАБ,  $|200|$ , N, V, 0;  
ЧАН, 0, M, M  $[1]$ ,  $|5400|$ , 0, V,  $|22|$ , ВУ, Z100;  
100, Z, ТЕСТ, Z  $|2000|$ , Z  $|6000|$ ;

## 10. Оператор выхода из программы

Последним выполняемым предложением программы оператора должно быть

**ВЫХОД;**

Данное предложение осуществляет возврат по цепочке стандартных операторов системы Ридер.

## 11. Пустой оператор

Пустой оператор ничем не обозначается и не связан ни с какими действиями. Им удобно пользоваться, например, для организации циклов или выделения комментария.

ЦИКЛ, M1, N,  $[A + B]$ ;

ЦИКЛ, M2, N1,  $[A - B]$ ;



«ПОЛУЧЕНИЕ СУММЫ»;  
 $S := S + P [N + N1 - 2];$   
 $M2 :=$ ;  
 $M1 : S := MOD * S;$

## § 6. Подпрограммы

В некоторых случаях, например для вычисления значений функции, может возникнуть необходимость не оформлять программу стандартным оператором. Для этого используется строка вида ПОДПРОГРАММА: <адрес> ; <предложение> ; ... ; <предложение> \*\*.

Подпрограмма, аналогично строке ОПЕРАТОР, состоит из последовательности предложений и после трансляции располагается в памяти машины с указанного адреса. Адрес записывается восьмеричным числом.

Обращение к таким подпрограммам необходимо писать в кодах машины или с помощью оператора КОД. При этом следует учитывать, что ячейка возврата имеет номер, равный <адрес> — 1. Последнее исполняемое предложение в подпрограмме должно иметь вид

КОНЕЦ;

Оператор КОНЕЦ осуществляет переход в ячейку <адрес> — 1.

## § 7. Программы

Для ввода программ, написанных в кодах машины, используются строки вида

ПРОГРАММА: <адрес> ; <коды> \*\* и  
PROGRAM: <адрес> ; <коды> \*\*

Адрес указывается восьмеричным числом и означает адрес ячейки памяти машины, начиная с которой будут располагаться вводимые коды.

Строкой первого вида пользуются при вводе информации, отперфорированной в машинных кодах. Недостающие до полной ячейки символы после точки с запятой в этом случае могут быть любыми и транслятором не учитываются.

Если коды расписаны и отперфорированы на УПП в алфавитно-цифровом режиме, то используется строка второго вида. Между собой коды никакими символами не разделяются.

Пример:  
ПРОГРАММА : 5400 ;  
015 5431 5401 0000  
036 0000 0175 0000  
256 5460 0010 5431  
\*\*

Первая и последняя строчки перфорируются в кодировке УПП, остальные — в кодах машины М-220. После трансляции информация будет расположена в ячейках 5400, 5401 и 5402 первого куба МОЗУ.

Те же самые коды можно ввести строкой

PROGRAM : 5400 ; 015 5431 5401 0000  
036 0000 0175 0000 256 5460 0010 5431 \*\*

В этом примере считается, что вся информация перфорируется в семиразрядной кодировке УПП.

## § 8. Начало работы системы

Указанием на способ запуска системы являются строки вида ВХОД : <адрес> \*\*  
или

READ : <последовательность предложений> \*\*

При наличии строки ВХОД после трансляции управление передается в ячейку с указанным адресом (восьмеричное число).

Тело строки READ в общем случае представляет собой программу оператора и транслируется аналогично предложениям в строке ОПЕРАТОР. Программа также оформляется стандартным оператором, которому присваивается свободный номер системы. После окончания трансляции работа системы начинается с выполнения последовательности предложений, заданных в строке READ.

Если во входной информации имеется несколько строк с заголовками ВХОД и READ, то к исполнению принимается последнее указание.

Примерами указаний для запуска системы являются следующие строки:

ВХОД : 20 \*\*

READ : A := M[0]; B := K[0];

ЦИКЛ, ВЫХОД, N, 10;

ЕСЛИ, A — K[N] < 2., B — M[N] < 1., ТО, ВЫХОД;

A := M[N], B := K[N];

ВЫХОД : ВЫХОД;

\*\*



READ : 100, Z ТЕСТ, Z |2600|, Z |5000|;

ВЫХОД;

\*\*\*

## § 9. Конец информации

Строка

END : \*\*

является указанием о конце входной информации. При встрече такой строки транслятор заканчивает свою работу.

## ЛИТЕРАТУРА

1. Арайс Е. А. Модельный метод анализа радиоэлектронных схем.— «Вопросы программирования и автоматизации проектирования». Вып. 2, Томск, Изд-во Томского университета, 1972.
  2. Арайс Е. А., Дмитриев В. М. Матричная интерпретация модельного метода анализа цепей. Там же.
  3. Арайс Е. А., Дмитриев В. М., Шутенков А. В. Входной язык системы автоматического проектирования РЭС. Там же.
-

## ТРАНСЛЯТОР С ВХОДНОГО ЯЗЫКА СИСТЕМЫ МАРС

А. В. ШУТЕНКОВ

В [1] приведено описание входного языка системы, предназначенной для автоматизации проектирования радиоэлектронных схем. Настоящая работа посвящена описанию транслятора с входного языка. В качестве вычислительной машины используется ЭВМ М-220 с двумя кубами оперативной памяти. Транслятор реализован в виде пакета стандартных операторов, включенных в библиотеку интерпретирующей системы Ридер [2]. Данный пакет включает в себя управляющий оператор, операторы для трансляции строк каждого вида и различные вспомогательные операторы, такие как чтения, печати исходной информации, составления информационного массива для стандартного оператора и другие.

Обращение к оператору (управляющий оператор имеет номер 0133) имеет вид:

0133, T, L, A, I.

Аргумент T обозначает тип обращения. Транслятором предусматриваются следующие типы обращений:

T = 000 — трансляция без печати информации;

T = 001 — печать информации на входном языке;

T = 002 — печать информации после трансляции;

T = 003 — печать как входного текста, так и транслированной информации. В последнем случае сначала печатается строка на входном языке, затем та же информация в кодах машины.

Вся входная информация располагается в 0-м кубе памяти машины. В обращении к транслятору указывается A — адрес начала входного текста.



Аргументы  $L$  и  $l$  обозначают длины массивов ячеек, отводимых соответственно для размещения операторов после трансляции и для таблиц меток и переменных. Если  $L$  и  $l$  в обращении указаны нулевыми, то транслятор устанавливает  $L = 1000$ ,  $l = 200$ .

Вся входная информация перфорируется в семиразрядной кодировке УПП, за исключением кодового участка в строке ПРОГРАММА, который перфорируется в кодах машины. Транслированная информация размещается в 1-м кубе оперативной памяти. Части программы, вводимые строками МАССИВ, ПРОГРАММА, PROGRAM, ПОДПРОГРАММА, располагаются в 1-м кубе с указанных в этих строках адресов ячеек. Стандартные операторы, полученные после трансляции строк ОПЕРАТОР и READ, записываются на магнитный барабан и прописываются в таблице характеристик интерпретирующей системы Ридер.

Рассмотрим распределение памяти ЭВМ во время трансляции. Распределение 1-го куба производится в соответствии с заданными параметрами системы. В 0-м кубе с ячейки А, указанной в обращении к транслятору, расположена входная информация. Далее транслятор отводит место для размещения таблиц названий операторов системы Ридер, вводимых схем и массивов, таблиц, встречающихся в программе меток и переменных. Длина последней таблицы равна  $l$  ячейкам. Следующий массив из  $L$  ячеек отводится для записи информации после трансляции строк ОПЕРАТОР, READ, ПОДПРОГРАММА.

Во время трансляции очередного оператора или подпрограммы управляющий оператор аннулирует старую таблицу меток и переменных и эта таблица составляется заново. Все метки и переменные считаются локализованными в той строке, где они встречаются. Поэтому нельзя осуществить переход по метке, например, из одного стандартного оператора в другой. Для обозначения переменных величин запрещается использовать идентификаторы, которыми обозначены метки, а также совпадающие с названиями схем и массивов. Таблица с названиями вводимых схем и массивов сохраняется до окончания трансляции.

Управляющий оператор транслятора во время работы использует некоторые ячейки из 1-го куба памяти машины, состояние которых он восстанавливает по окончании трансляции.

Работу транслятора в целом можно представить следующим образом. После распределения памяти 0-го куба ЭВМ управляющий оператор производит обращение к оператору загрузки параметров системы. Этот оператор выбирает информацию из строк, задающих распределение 1-го куба памяти машины, и заносит ее в соответствующие ячейки. Так, параметры АРП и ЛРП, указывающие размеры рабочего поля Ридера, заносятся в ячейки 0415—0417; АИС — в ячейки 7541 и 7615 интерпретирующей системы ИС-22. Параметры АСХЕМ и L СХЕМ сохраняются в управляющем операторе на время трансляции. Этот оператор также очищает старое рабочее поле Ридера для того, чтобы переписать на магнитный барабан операторы, не закончившие свою работу.

После загрузки параметров происходит трансляция имеющихся во входной информации схем и массивов. Составляется таблица, включающая в себя название схемы или массива, начало расположения в оперативной памяти и количество занимаемых ячеек. Далее производится трансляция оставшихся строк. При встрече строки *END*, извещающей об окончании входного текста, управляющий оператор подготавливает систему к работе и осуществляет запуск системы в соответствии с имеющимся указанием во входной информации. Если указания о способе запуска системы не было, то происходит с помощью интерпретирующей системы Ридер возврат по цепочке вложенных операторов.

Управляющий оператор производит распределение работы по трансляции входной информации между различными операторами, входящими в транслятор, и обработку поступающей от них информации. Такой способ организации транслятора позволяет без особых осложнений производить как усовершенствование существующего варианта, так и расширение транслятора при вводе новых объектов во входном языке. Транслятор представляет собой, вообще говоря, стандартный оператор системы Ридер, что дает возможность обращаться к нему не только в начале работы, но и по ходу выполнения какой-либо программы. Это позволяет также производить ввод, трансляцию и обработку информации по частям, что удобно при решении больших задач.

Опишем теперь кратко работу операторов, входящих в пакет транслятора.

Оператор чтения. Программы чтения отдельных символов и слов входного текста оформлены одним опера-



тором. Одна из рабочих ячеек является настройкой для оператора чтения. В ней хранится адрес ячейки и номер читаемого символа (в ячейке расположено по 6 символов). В зависимости от обращения оператор производит чтение очередного символа по настройке или слова до разделителя. После чего настройка перестраивается на символ, следующий за последним считанным, а результат чтения засылается в рабочие ячейки.

Оператор перевода числа. Осуществляет перевод встречающихся во входном тексте констант, условных и вещественных чисел в двоичную систему. Исходной информацией является настройка на начало числа. Результатом работы оператора является само переведенное число и его признак. В зависимости от того, что обрабатывал оператор — константу, условное или вещественное число, в одну из рабочих ячеек засылается соответствующий признак. Наличие нуля в ячейке признака означает, что оператор выявил ошибку в исходной записи числа.

Оператор загрузки массивов. Данный оператор производит с помощью оператора перевода числа чтение элементов массива и их засылку в соответствующие ячейки 1-го куба оперативной памяти ЭВМ. В рабочие ячейки заносится информация, содержащая идентификатор массива, адрес первой ячейки массива и его длину, а также информация о наличии ошибок, если таковые встретились при обработке строки МАССИВ.

Оператор трансляции программ. Для обработки строк ПРОГРАММА и PROGRAM используется один оператор. Он формирует ячейки по пятнадцать восьмеричных цифр и располагает их, начиная с указанного в строке адреса. При отсутствии в строке ошибок в рабочую ячейку заносится адрес начала и длина полученного массива ячеек.

Оператор загрузки схем. Данный оператор использует входящую в систему Ридер библиотеку моделей. Модель каждого компонента схемы вызывается с магнитного барабана в оперативную память и в нее заносятся параметры, указанные в исходной информации. После этого модель записывается на свободный участок магнитного барабана. При этом составляется таблица, содержащая информацию о числе выводов каждого элемента схемы, направления токов в них и о месте размещения модели во внешней памяти ЭВМ. После обработки всей строки СХЕМА происходит обращение к оператору, составляющему по указанной таб-

лице общую модель схемы. Исходной информацией для загрузки схем являются величины *A SCHEM* и *L SCHEM*. Они выбираются из управляющего оператора и изменяются при составлении модели схемы. По окончании работы в рабочие ячейки засылаются идентификатор схемы, адрес начала модели схемы в оперативной памяти и количестве ячеек, занятых моделью. В соответствии с этим управляющий оператор уменьшает величину *LCHEM* и настраивает параметр *A SCHEM* на следующий свободный участок оперативной памяти.

Оператор трансляции выражений. Этот оператор используется как вспомогательный при обработке строк *ОПЕРАТОР*, *ПОДПРОГРАММА* и *READ*. Он производит трансляцию встречающихся в этих строках арифметических и условных выражений. Вообще говоря, данный оператор можно разделить на два основных оператора. Первый из этих операторов расписывает заданное выражение в виде *ПОЛИЗ* (ПОльская Инверсная Запись) [3]. Магазином для накопления и пересылки операций служит массив ячеек внутри транслируемой программы. На выходе из оператора ячейки, отведенные под магазин, очищаются *ПОЛИЗ* записывается в ячейках, отводимых в каждой программе под рабочие. Обработка выражения производится до первого ограничителя, не являющегося скобкой или знаком операции. Считается, что такими разделителями могут быть: (точка с запятой) или , (запятая). При этом учитывается возможность использования запятой для перечисления переменных в операторе присваивания. Если арифметическое выражение представляет собой либо вещественное число, либо символ одной переменной, то обращение ко второму оператору не происходит. В этом случае в рабочую ячейку заносится адрес этой переменной или вещественного числа и происходит выход из оператора.

Второй из операторов трансляции выражений производит составление машинных команд по заданной *ПОЛИЗ*. На выходе из оператора в рабочие ячейки транслятора заносится адрес результата выполнения арифметического выражения и адрес последней из занятых рабочих ячеек. Эти данные используются для ограничения массива рабочих ячеек в каждой транслируемой программе. Для трансляции встречающихся в выражениях операций возведения в степень и идентификаторов функций, за исключением *MOD*, используются



обращения к стандартным программам интерпретирующей системы ИС-22.

Транслятор операторов входного языка. Данный оператор составляет основу группы операторов, предназначенных для трансляции строк ОПЕРАТОР, READ, ПОДПРОГРАММА. Транслированная программа размещается в 0-м кубе памяти. Для нее отводится массив в  $L$  ячеек ( $L$  задано в обращении к управляющему оператору). В начале массива расположена сама программа, в конце — ее рабочие ячейки и константы. Во время трансляции составляется таблица меток и переменных. Все истинные адреса в операторе входного языка КОД запоминаются и в дальнейшем не подвергаются никаким изменениям. В полученной программе производится уплотнение за счет сдвига массива констант и рабочих ячеек и коррекция всех адресов в соответствии с таблицей меток и переменных.

Информация, полученная после трансляции строки ПОДПРОГРАММА, настраивается по указанному в строке адресу и пересылается в 1-й куб оперативной памяти.

В остальных случаях программа оформляется в виде стандартного оператора. В соответствии с требованиями интерпретирующей системы Ридер составляется информационный массив и стандартному оператору присваивается произвольный свободный номер системы. Затем идентификатор этого оператора заносится в таблицу названий операторов, сам оператор записывается на свободный участок магнитного барабана и прописывается в таблице характеристик Ридера.

Оператор печати. Для распечатки информации как на входном языке, так и транслированной служит оператор печати. В зависимости от типа обращения к транслятору производится соответствующая печать. Этот оператор используется также при распечатке ошибок, выявленных в ходе трансляции. При печати транслированной информации в машинных кодах может быть использована стандартная программа печати из ИС-22.

Транслятор выдает сообщения о встречающихся во входной информации ошибках. Если ошибка была в заголовке строки, то далее до конца этой строки информация не обрабатывается. При наличии ошибок в теле строки выдается на печать сообщение об ошибке и предложение, ее содержащее. Затем производится обработка следующего предложения.

По окончании строки, содержащей ошибки, печатается текст

## НЕВЕРНА СТРОКА

и заголовок строки. При обнаружении ошибок прекращается печать транслированной информации, если она была указана в обращении к транслятору. В этом случае, независимо от типа обращения, распечатывается весь входной текст, а по окончании обработки всей информации происходит прерывание работы системы. Если ошибки допущены в задании параметров системы, то прерывание работы системы производится сразу, без трансляции остальных строк.

Список диагностических текстов, выдаваемых на печать при наличии ошибок, и краткое описание характера каждой ошибки приведены в приложении.

Пакет операторов трансляции прописан в системе Ридер под номерами со 114 по 133. В настоящее время общий объем транслятора составляет около 10 000 машинных слов.

Время работы транслятора зависит от величины рабочего поля и особенностей задания. С увеличением рабочего поля время работы сокращается. В таблице приведена зависимость времени трансляции от длины рабочего поля входного текста, содержащего распределение памяти, массив из 20 элементов, схему из 4 компонентов и два стандартных оператора, включающих в себя в общей сложности 150 операторов входного языка, 60 переменных величин и 40 меток.

Т а б л и ц а

| Длина РП<br>(количество ячеек) | Время работы<br>(минуты) |
|--------------------------------|--------------------------|
| 1400                           | 19                       |
| 3400                           | 10                       |
| 6000                           | 6                        |

Удлинняют работу транслятора алгоритмы, связанные с просмотром большой информации.

Эффективность составленных транслятором программ, оформляемых стандартными операторами, также зависит от особенностей исходного алгоритма и способов описания на входном языке. Сравнение программ, составленных трансля-



тором и вручную, систематически не проводилось. Однако для программ средней сложности имеющийся материал позволяет считать, что длина и скорость работы программ, составленных транслятором, как правило, не будет существенно превосходить длину и скорость работы программы, составленной вручную без особой тщательности.

Время, которое потребовалось для составления входного языка и реализацию транслятора с него, составляет около 3 человеко-лет.

В дальнейшем будет проводиться работа по улучшению работы транслятора в направлении уменьшения времени трансляции и длины составляемых программ.

## ПРИЛОЖЕНИЕ

1. НЕВЕРНО УКАЗАН РАЗДЕЛИТЕЛЬ <разделитель>.

2. НЕВЕРЕН СИМВОЛ <символ>.

Вместо указанного разделителя или символа должен стоять какой-либо другой разделитель или символ в соответствии с требованиями входного языка. Например, после заголовка строки указан разделитель, отличный от двоеточия, или в записи информации о схеме отсутствует символ *N* для обозначения перечисления номеров узлов какого-либо элемента.

3. НЕТ ЗАГОЛОВКА СТРОКИ <слово>

Указан заголовок строки, которого нет во входном языке, либо в имеющемся заголовке допущена ошибка.

4. НЕВЕРНО ЗАДАНЫ ПАРАМЕТРЫ СИСТЕМЫ

Допущены ошибки при задании какого-либо параметра системы. Сами ошибки распечатываются раньше данного текста.

5. НЕХВАТКА РЕЗЕРВА ДЛЯ СХЕМ

При трансляции очередной схемы заполнен весь массив ячеек, заданный параметром *LSXEM*. Необходимо увеличить значение этого параметра.

6. МАЛО МЕСТА ДЛЯ ТРАНСЛЯЦИИ.

Во время распределения памяти в 0-м кубе для трансляции операторов или подпрограмм не остается достаточного количества ячеек. Необходимо уменьшить либо объем входной информации, либо адрес ее начала в памяти машины.

7. ОШИБКА В АДРЕСЕ (НЕТ;)

Допущена ошибка в записи символов адреса либо после адреса отсутствует разделитель — точка с запятой.

#### 8. АДРЕС + ДЛИНА $> 7777$

Длина подпрограммы больше количества отведенных для нее ячеек, то есть подпрограмма не может быть размещена, начиная с указанного адреса и до последней ячейки памяти машины.

#### 9. НЕТ НАЗВАНИЯ ОПЕРАТОРА (НЕТ;)

Отсутствует название после заголовка строки ОПЕРАТОР либо после названия нет разделителя точки с запятой.

#### 10. ОШИБКА В НОМЕРЕ ОПЕРАТОРА ( $N < 5$ , $N > 133$ ).

Допущена ошибка в символах номера оператора, либо указанный номер меньше 5, либо больше 133.

#### 11. НЕТ СВОБОДНОГО НОМЕРА ДЛЯ ОПЕРАТОРА $< \text{название} >$

Заполнена вся таблица названий операторов, то есть в системе нет свободных номеров для записи новых стандартных операторов. В этом случае можно использовать названия или номера уже имеющихся в системе операторов, но не требующихся для выполнения вводимой программы.

#### 12. ЗАПОЛНЕНА ВСЯ ТАБЛИЦА МЕТОК И ПЕРЕМЕННЫХ.

В программе много меток и переменных, и в отведенной таблице не хватает места. Требуется увеличить величину  $L$  в обращении к транслятору.

#### 13. ДЛИНА ОПЕРАТОРА (ПОДПРОГРАММЫ) $> 1000$ ЯЧЕЕК.

Происходит заполнение всего массива ячеек, отведенных в 0-м кубе для размещения стандартных операторов и подпрограмм после трансляции. Необходимо в обращении к транслятору увеличить величину  $L$  (стандартное значение  $L = 1000$ ).

#### 14. НЕТ СООТВЕТСТВИЯ МЕЖДУ КОНЕЦ И МЕТОКОЙ.

Допущена ошибка в оформлении подпрограмм, включенных в состав стандартного оператора. Встречается оператор КОНЕЦ, но не было перед этим предложения со специальной меткой. Если после текста указан идентификатор метки, то в подпрограмме, помеченной данной меткой, либо отсутствует оператор КОНЕЦ, либо еще какое-нибудь предложение помечено меткой специального вида.

#### 15. ОШИБКА В ОПЕРАТОРЕ КОД

Данный текст печатается при неверной записи хотя бы одной из команд в операторе КОД. Ошибками могут быть,



например, большее чем четыре количество цифр в записи адреса (больше трех в коде операции) или неверное указание разделителей.

**16. ОПРЕДЕЛЕНО БОЛЕЕ ОДНОЙ МЕТКИ <идентификатор>.**

Одной и той же меткой помечено несколько предложений. Ошибкой также является использование в качестве метки идентификатора какой-либо переменной.

**17. НЕТ АРГУМЕНТА ОПЕРАТОРА.**

В операторе АРГ отсутствует аргумент между двумя разделителями.

**18. НОМЕР ОПЕРАТОРА > 133 (<5).**

В обращении к стандартному оператору указан номер больше чем 133 или меньше чем 5.

**19. НЕТ ОПЕРАТОРА <название>.**

В обращении к стандартному оператору указано название несуществующего в системе оператора.

**20. НЕВЕРНА ЗАПИСЬ (НЕТ РАЗДЕЛИТЕЛЯ ; ).**

Допущена ошибка в записи предложения. Такими ошибками могут быть, например, предложение, начинающееся с квадратной скобки, или пропущенный разделитель после метки.

**21. НЕТ ЗНАКА ОПЕРАЦИИ.**

В выражении пропущен знак операции.

**22. НЕТ ФУНКЦИИ <название>.**

В выражении встречается запись функции, не определенной во входном языке системы.

**23. НАРУШЕНО ПРАВИЛО СКОБОК.**

В записи выражения встречается неравное количество открывающих и закрывающих скобок.

**24. ЛИШНИЙ ЗНАК ОПЕРАЦИИ.**

Выражение содержит несколько знаков операций, записанных подряд.

**25. СКОБКИ [ ] В ЗАПИСИ ЭЛЕМЕНТА МАССИВА**

Ошибкой является запись квадратных скобок для обозначения условного характера части выражения в уже имеющихся квадратных скобках, определяющих индекс элемента массива.

**26. НЕВЕРНА ЗАПИСЬ ОПЕРАТОРА: =**

Встречается знак: = (оператор присваивания) в скобках или в левой части этого оператора имеется арифметическое выражение, не определяющее индекс элемента массива.

## 27. В ВЫРАЖЕНИИ $> 1$ ОПЕРАТОРА: =

В выражении знак оператора присваивания должен встречаться не более одного раза.

## 28. НЕПОНЯТНА ОПЕРАЦИЯ $< \text{символ} >$ .

В записи выражения встречается символ, не являющийся знаком операции.

## 29. НЕТ ПАРАМЕТРА LRP.

После установки параметров системы транслятором адрес ячейки начала рабочего поля получился больше адреса ячейки конца рабочего поля.

## 30. ОШИБКА В ПАРАМЕТРЕ.

Неверна запись строки, задающей какой-либо параметр системы. Значение параметра задано не восьмеричным числом, либо нет признака конца строки (двух звездочек).

## 31. ОШИБКА В КОДАХ.

В записи тела строк ПРОГРАММА и PROGRAM встречаются символы, отличные от восьмеричных цифр.

## 32. ЗАПРЕЩЕННЫЙ НАБОР ЦИФР X14 4XXX.

В строке ПРОГРАММА не должно встречаться кода, начало которого имеет вид X14 4XXX, где символом X обозначена произвольная цифра. Такая комбинация означает в семиразрядной кодировке УПП символ \*, который употребляется для обозначения конца строки.

## 33. ЧИСЛО НЕ УСЛОВНОЕ.

Отсутствует условное число в каком-либо месте входного текста, где оно должно быть в соответствии с описанием входного языка.

## 34. Нет НАЗВАНИЯ СХЕМЫ.

Введена схема, не имеющая названия.

## 35. НЕВЕРЕН НОМЕР ЭЛЕМЕНТА.

В схеме нарушен порядок нумерации элементов.

## 36. НЕТ МОДЕЛИ $< \text{название модели} >$ .

В схеме указан идентификатор модели, не включенный в библиотеку системы.

## 37. N ПАРАМЕТРА $>$ ДЛИНЫ МОДЕЛИ.

Номер параметра какого-либо элемента схемы либо отсутствует, либо превосходит длину самой модели.

## 38. ЧИСЛО ВЕТВЕЙ $\neq$ ЧИСЛУ УЗЛОВ.

У какого-либо элемента схемы нарушено равенство между количеством ветвей и узлов.

## 39. ОШИБКА В СИМВОЛАХ ЧИСЛА.

Первый символ числа отличен от цифры или символов, употребляемых в записи вещественных чисел.



#### 40. КОНСТАНТА БЕЗ | (ЦИФРЫ $> 7$ ).

В записи константы нет закрывающей вертикальной черты, либо встречаются цифры, отличные от восьмеричных.

#### 41. НЕТ ПОКАЗАТЕЛЯ СТЕПЕНИ.

В вещественном числе после разделителя  $10$  отсутствует показатель степени.

#### 42. ЧИСЛО $>_{10} 19$ (МНОГО ЗНАКОВ).

Показатель степени вещественного числа больше 19 или само число больше чем  $10^{19}$ . Ошибкой также является большое количество цифр в записи условного числа, что при переводе этого числа вызывает переполнение разрядов ячейки.

#### 43. НЕТ МЕТКИ $<\text{слово}>$ .

В программе оператора встречается передача управления на метку, которой не помечено ни одно из предложений.

### ЛИТЕРАТУРА

1. Арайс Е. А., Шутенков А. В. Язык МАРС. Настоящий сборник.
2. Арайс Е. А., Сибиряков Г. В., Шутенков А. В. Система интерпретации для решения больших задач. Вопросы программирования и автоматизации проектирования. Вып. 1, Изд-во Томского университета, Томск, 1971.
3. Б. Ренделл, Л. Рассел. Реализация АЛГОЛА-60. М., Изд-во «Мир», 1967.

## О ЗАДАЧЕ ОПТИМАЛЬНОГО ВЫБОРА ИНСТРУМЕНТА

Е. А. АРАЙС, А. К. МАРТЫНОВ, Л. А. ШАПИРО

В настоящей работе вводятся основные понятия, необходимые для построения и описания алгоритмов оптимального выбора режущего инструмента, принципы программной реализации этих понятий. В основу системы понятий положена терминология модельного метода машинного проектирования радиоэлектронных схем МАРС [1].

Решением задачи оптимального выбора инструмента будем считать совокупность алгоритмических, программных и модельных средств, обеспечивающих автоматический (при помощи ЭВМ) выбор инструмента для выполнения заданной последовательности операций в соответствии с установленным критерием оптимальности.

Поскольку данная задача решается на основе комплекса МАРС, следует формализовать некоторые понятия для приведения их в соответствие с уже существующими понятиями и определениями комплекса. Будем в дальнейшем считать, что деталь в рамках задачи оптимального выбора инструмента есть установленный технологом порядок преобразования заготовки. Элементарный акт преобразования в данном технологическом процессе назовем переходом. Здесь и далее будем обозначать детали буквой  $D$  с индексами или без них, а переходы — буквами  $P$  с индексами или без индексов.

Каждый переход задается своей математической моделью. Математическая модель перехода включает в себя символ  $P_i$  (название перехода) и числовые параметры  $Z_1, Z_2, \dots, Z_k$ , которые могут иметь вещественные или целые значения. Модель перехода будем записывать в виде  $P_i(z)$



Математической моделью детали будем называть соответствующую детали последовательность переходов. Модель детали записывается в виде

$$D_i = \Pi_1^i(z) \times \Pi_2^i(z) \times \dots \times \Pi_n^i(z). \quad (1)$$

Такую модель будем называть  $n$ -мерной, по числу входящих в нее переходов.

Рассмотрим далее программную реализацию введенных понятий. При этом будем исходить из того, что комплекс программ решения задачи автоматического выбора инструментов является частью более общего комплекса автоматизации проектирования технологии.

Каждый переход будем задавать как ветвь графа. Номер ветви соответствует порядковому номеру перехода. Состояния детали после выполнения каждого перехода нумеруются; эти состояния будем считать узлами графа. Предполагается, что производство всегда осуществляется по замкнутому циклу. Обработка начинается в 0-м узле и там же заканчивается. Таким образом, нулевой узел служит как бы «заземлением», т. е. представляет собой внешнее по отношению к рассматриваемому производству. Процесс обработки детали при решении задачи выбора инструмента можно графически представить в виде кольца.

Программное задание каждого перехода осуществляется его названием, числовыми параметрами, номером ветви (порядковым номером перехода), номерами начального и конечного узлов. Предложение, задающее переход, имеет вид:  $\langle \text{номер предложения} \rangle : \langle \text{название перехода} \rangle (\langle \text{числовые значения параметров} \rangle)$  В ( $-\langle \text{номер ветви} \rangle$ ,  $\langle \text{номер ветви} \rangle$ ). N ( $\langle \text{номер начального узла} \rangle$ ,  $\langle \text{номер конечного узла} \rangle$ );

Рассмотрим структуру предложения подробнее. Номер предложения (номера начинаются с нуля) записывается с контрольной целью и в решении задачи не участвует.

Название перехода — произвольная последовательность букв и цифр, начинающаяся с буквы. Названия распознаются по первым шести символам. Таким образом, переходы АБВГДЕІ, АБВГДЕ, АБВГДЕЖЗИ, АБВГДЕ750 не различаются в рамках данного комплекса.

Числовые значения параметров записываются в виде равенств. Правая часть равенства задает числовое значение параметра, левая часть — название параметра и его номер в модели (номер в модели задается из программных сообщений и пользователя не касается). Названием параметра

может быть любая последовательность букв. Указывать название параметра не обязательно, для правильной работы программы достаточно номера в модели. Название вводится для облегчения визуального контроля информации.

Номер ветви и номера начального и конечного узлов задаются целыми десятичными числами.

Программное задание детали осуществляется ее названием и последовательностью переходов, переписанных в порядке выполнения. Завершается задание детали символом \*\*.

Пример. Рассмотрим программное задание детали с названием «Шайба», обработка которой осуществляется фрезерованием контура со снятием 5 мм металла, глубина — 10 мм, расточкой отверстия диаметром 30 мм, фрезерованием выступа со снятием металла 3 мм, глубина 4 мм, диаметр выступа 150 мм.

ДЕТАЛЬ: ШАЙБА;

О: ФРЕЗК (ДИАМЕТР2 = 8, МАТЕРИАЛ5 = 2, ШИРИНА4 = 5, ГЛУБИНА3 = 10), В (—0,0) N (0,1);

1: ФРЕЗОК (ДИАМЕТР2 = 30, МАТЕРИАЛ5 = 2)  
В (—1,1) N (1,2);

2: ФРЕЗВК (ДИАМЕТР2 = 20, МАТЕРИАЛ5 = 2, ШИРИНА4 = 3, ГЛУБИНА3 = 4, ДЛИНА6 = 150) В (—2,2) N (2,0)\*\*.

Введем следующее основное понятие — понятие инструмента, которое будем строить на основе обобщенной стоимости использования инструмента для выполнения того или иного перехода. Стоимость использования инструмента определяется не только типом перехода, но и временем смены инструмента, т. е. предыдущим переходом и типом инструмента.

Стоимость  $S$  применения инструмента  $V$  для выполнения данного перехода можно записать как

$$S(V) = S_1 + S_2 + S_3.$$

Здесь  $S_1$  — стоимость снятия предыдущего инструмента,  $S_2$  — стоимость установки данного инструмента,  $S_3$  — стоимость использования инструмента непосредственно для выполнения перехода.

В общем случае, по-видимому, можно выделить следующие типы задач, связанные с определением стоимости использования инструмента:



(А). Стоимость инструмента определяется непосредственно типом реализуемого перехода и не зависит от предыдущих переходов (стоимость без памяти).

(Б). Стоимость инструмента определяется реализуемым переходом и типом предшествующего перехода (только одного) — стоимость с единичной памятью.

(В). Стоимость инструмента определяется реализуемым переходом и типом инструмента, выбранного для реализации предшествующего перехода.

(Г). Стоимость инструмента определяется реализуемым переходом и всей совокупностью предшествующих переходов либо типов инструментов — стоимость с полной памятью.

Будем предполагать, что в задачах машинного проектирования технологии стоимость использования инструмента может зависеть лишь от предшествующих переходов и не зависит от последующих.

Определим стоимость использования инструмента для  $j$ -го перехода модели детали  $D_i$  (1) следующим образом.

(А). Если стоимость инструмента не зависит от предыдущих переходов (стоимость без памяти), будем полагать, что инструмент задается функцией

$$F_j = F_j(\Pi_j^i(z)). \quad (2)$$

Применение инструмента для реализации конкретных переходов будем впредь обозначать этой же буквой  $F$  с индексами или без них.

(Б). Для стоимости с единичной памятью инструмент задается функцией

$$F_j = F_j(\Pi_j^i(z), \Pi_{j-1}^i(z)). \quad (3)$$

После выбора инструментов для реализации каждого из переходов в (1) модель детали можно записать в виде

$$D_i = F_1^i \times F_2^i \times \dots \times F_n^i. \quad (4)$$

Здесь  $F_1^i, F_2^i, \dots, F_n^i$  — названия соответствующих инструментов. В некоторых ситуациях, возможно, удобным будет использование смешанной записи детали, включающей в себя элементы (1) и (4).

(В). Если стоимость инструмента определяется реализуемым переходом и типом инструмента для предшествующего перехода, то его можно задать функцией

$$F_j = F_j(\Pi_j^i(z), F_{j-1}). \quad (5)$$

(Г). Для стоимости с полной памятью инструмент задается функцией вида

$$F_j \equiv F_j(\Pi_1^i(z), \dots, \Pi_j^i(z)) \quad (6)$$

либо

$$F_j = F_j(\Pi_j^i(z), F_1, F_2, \dots, F_{j-1}). \quad (7)$$

Таким образом, определено понятие стоимости инструмента как некоторой функции, заданной на последовательности переходов (на математической модели детали).

Рассмотрим программное задание (модель) инструмента. В общем случае это некоторая подпрограмма, вычисляющая по заданному отрезку математической модели детали стоимость применения инструмента для реализации последнего перехода этого отрезка.

На первом этапе разработки системы машинного проектирования технологии предлагается следующее задание стоимости использования инструмента. Рассматривается только стоимость с единичной памятью. Предполагаем, что стоимость использования инструмента для выполнения  $j$ -го перехода модели определяется типом перехода  $\Pi_j$  и типом инструмента  $F_{j-1}$ , использованного для предыдущего перехода.

Задавать модели инструментов будем в виде операторов системы МАРС. Структура такого оператора в целом имеет вид:

ОПЕРАТОР: <название инструмента>;  
АРГ, <аргументы оператора>;  
<подпрограмма вычисления стоимости>;  
ВЫХОД \*\*

Аргументами оператора являются материал режущей части инструмента, его размеры, параметры перехода.

На основании введенных понятий детали, перехода, инструмента и стоимости его использования можно ввести основные понятия стоимости производства и оптимального выбора инструмента.

Стоимостью производства детали или просто стоимостью детали (1) в смысле задачи оптимального выбора инструмента будем называть функцию

$$S = \sum_{j=1}^n F_j,$$

где  $F_j$ ,  $j = 1, \dots, n$  — зафиксированный набор инструментов для выполнения каждого из переходов.



Оптимальным набором инструментов для обработки детали (1) будем называть набор инструментов

$F_1^i, F_2^i, \dots, F_n^i$ , доставляющий минимум стоимости детали

$$S_{\min} = \sum_{j=1}^n F_j^l = \min_{l \in L} \sum_{j=1}^n F_j^l,$$

где  $F_j^l, j = 1, \dots, n, l \in L$  — весь исходный набор инструментов [2].

Рассмотрим структуру моделей переходов инструментов.

Модель перехода содержит значения параметров перехода, расположенных в определенном фиксированном порядке, и допускает внесение дополнительных параметров, характеризующих инструмент, применяемый для выполнения данного перехода.

Моделью инструмента является оператор, который по заданным параметрам перехода и инструмента вычисляет оперативное время.

Программа-оптимизатор имеет операторную структуру. Исходной информацией для нее служат схема, задающая деталь, и таблицы имеющихся в наличии инструментов (табл. 1) и соответствия между материалами деталей и материалами режущей части инструмента, примененного для обработки этого материала (табл. 2).

Оператор оптимизации основан на простой процедуре упорядоченного перебора, так как полный перебор невозможен.

С помощью вспомогательных операторов определяются допустимый материал режущей части инструмента, стоимость инструмента (используются табл. 1 и 2), ширина и глубина фрезерования, стойкость инструмента, которые в качестве дополнительных параметров вносятся в модели соответствующих переходов. Тем самым задается набор режущих инструментов для обработки детали.

Специальным оператором вычисляется обобщенная стоимость  $F$  применения полученного набора инструментов, которая принимается за начальное приближение к искомому минимальному значению.

Операторы оптимизации варьируют параметры переходов и инструментов (ширину и глубину фрезерования, диаметр и материал режущей части инструмента) и заносят новые значения в модель (схему). После этого снова произ-

водится обращение к оператору, вычисляющему обобщенную стоимость применения уже нового набора инструментов.

Таблица 1

| Материал<br>р. и. | Диаметр<br>р. и. | Стоимость<br>р. и. | Число<br>допустимых<br>переточек |
|-------------------|------------------|--------------------|----------------------------------|
|                   |                  |                    |                                  |
|                   |                  |                    |                                  |
|                   |                  |                    |                                  |
|                   |                  |                    |                                  |
|                   |                  |                    |                                  |

Таблица 2

| Материал<br>р. и.  | 1 | 2          | 3 |
|--------------------|---|------------|---|
| Материал<br>детали |   |            |   |
| 1                  |   |            |   |
| 2                  |   | $a_{ij}^*$ |   |
| ...                |   |            |   |

$$*a_{ij} = \begin{cases} 0, & \text{если обработка возможна} \\ 1, & \text{в противном случае} \end{cases}$$

При вычислении обобщенной стоимости  $F$  учитывается оперативное время — время непосредственной обработки детали, время смены инструментов и амортизация инструментов. Обобщенная стоимость  $F = T + \alpha A$ , где  $T = t_0 + t_{см}$  (сумма оперативного времени и времени смены инструмента) — показатель производительности.  $A$  — амортизация инструмента  $A = \frac{t_0 \cdot c}{n_1 \cdot T_1}$  ( $t_0$  — оперативное время,  $c$  — стоимость инструмента,  $T_1$  — стойкость инструмента,  $n_1$  — число переточек) — показатель экономичности,  $\alpha$  — весовой коэффициент.

Операторный принцип программирования позволяет достичь существенной экономии оперативной памяти. Существенным удобством при решении практических задач является то, что библиотеку операторов можно легко расширить в нужном направлении, т. е. в направлении совершенствования методики решения задачи, несложно вводить модификации. Можно легко изменять набор имеющихся в наличии инструментов, не перестраивая программ оптимизации.



## ЛИТЕРАТУРА

1. Арайс Е. А., Шутенков А. В. Язык МАРС. Настоящий сборник.
  2. Этин А. О., Шумяцкий Б. Л., Складьевская Е. И. Выбор режущего инструмента при оптимизации операционной технологии на автоматизированных станках.—«Вестник машиностроения», № 4, 1974.
-

## СИСТЕМА ИНТЕРПРЕТАЦИИ ГРАВИТАЦИОННЫХ И МАГНИТНЫХ АНОМАЛИЙ (СИГМА)

А. А. НИКОЛЬСКИЙ

### 1. Введение и постановка задачи

Обработка геофизических полей (в том числе магнитных и гравитационных) на ЭВМ характеризуется следующими особенностями:

а) большое количество исходной информации (до  $10^5$  бит);

б) объемы промежуточной и конечной информации соизмеримы с объемом исходной информации;

в) большое разнообразие способов интерпретации.

Первые две особенности приводят к тому, что время, затрачиваемое на решение задач, определяется в основном временем ввода и вывода информации. Третья особенность предопределила накопление большого количества различных программ, каждая из которых реализует тот или иной метод интерпретации. Однако при комплексной интерпретации геофизических полей на ЭВМ интерпретатор сталкивается с целым рядом факторов, затрудняющих применение данных программ, основные из которых сводятся к следующему:

а) большинство программ реализует либо один способ обработки, либо весьма узкий круг таких способов;

б) невозможность оперативного обмена информацией между различными программами;

в) для различных программ различные способы задания исходной информации;

г) невозможность одновременной обработки двух и более исходных полей.

Все это не позволяет интерпретатору выбирать произвольную схему интерпретации и он вынужден решать поставленную задачу по частям или составлять свою программу.



В обоих случаях значительно увеличивается время, затрачиваемое на интерпретацию.

Учитывая вышесказанное, очевидна целесообразность такой системы, которая позволяла бы, во-первых, задавать произвольную схему интерпретации, во-вторых, включать в ее состав новые программы.

Ниже описывается система СИГМА, которая отвечает поставленным требованиям. Система реализована на ЭВМ типа М-220.

## 2. Структура системы

Составными частями системы СИГМА являются:

- 1) библиотека стандартных операторов (БСО);
- 2) административная система (АС);
- 3) интерпретирующая система Ридер;
- 4) входной язык;
- 5) транслятор с входного языка.

Система Ридер, входной язык и транслятор с входного языка заимствованы из системы МАРС [1] — [4].

## 3. Библиотека стандартных операторов (БСО)

БСО включает в себя комплекс программ, каждая из которых реализует определенный метод (или комплекс методов) интерпретации того или иного геофизического поля. Все программы оформлены в виде операторов интерпретирующей системы Ридер [2].

В состав БСО после соответствующей переделки могут быть включены любые программы как в кодах М-220, так и на различных языках (автокод, АЛГОЛ, АЛЬФА и др.). В первом случае перед программой следует поставить информационный массив и в соответствующих местах обращения к программам административной системы. Во втором случае — переписать программу на входном языке.

На данном этапе разработки системы в БСО имеется 10 операторов, которые позволяют пересчитывать гравимагнитные поля в верхнее и нижнее полупространства, вычислять вторую производную на любом уровне, находить энтропию, радиус автокорреляции, дисперсию, число пересечений со средним, функцию пересеченности и другие функции. Кроме того, в БСО имеется ряд вспомогательных операторов, которые подготавливают исходные данные к обработке. К таким операторам, например, относится оператор ПАЛ 1, который

выбирает из матрицы исходных значений значения, необходимые для вычисления интеграла Пуассона. Обращение к нему на входном языке имеет вид

ПАЛ1, ШИФР, С, Э;

где ШИФР — шифр массива, который требуется обрабатывать;

С — номер столбца и Э — номер элемента в столбце, для которых производится преобразование.

После окончания работы оператора ПАЛ1 в МОЗУ-1 будет сформирована матрица значений для работы операторов ВВЕРХ, ВНИЗ и VZZ, которые позволяют получать значения поля в верхнем, нижнем полупространствах и вычислять вторую производную на любом заданном уровне. Обращение к данным операторам на входном языке имеет вид:

ВВЕРХ, ТИП, Z; ВНИЗ, ТИП, Z; VZZ, ТИП, Z;

где Z — вертикальная координата, ТИП — условное число (см. ниже).

Не останавливаясь подробно на описании операторов, отметим лишь, что все операторы, кроме вышеприведенных, в качестве аргументов имеют следующие значения:

ШИФР, ТИП, С, Э, ΔX, ΔY, NЭ, NC;

где ШИФР, С, Э — то же, что и выше;

ТИП — условное число, указывающее на режим работы АС (см. ниже);

ΔX и ΔY шаг по горизонтали и вертикали соответственно (через сколько элементов и столбцов выбирать элементы);

NЭ и NC — количество элементов и столбцов соответственно, которые одновременно подвергаются обработке при вычислении той или иной функции.

По этим аргументам административная система производит выборку необходимых элементов из обрабатываемого массива и формирование массивов по результатам вычисления.

### 3. Административная система

При решении геофизических задач исходная, промежуточная и конечная информации обычно представляются в виде матриц, размеры которых достигают  $200 \times 500$  и более. Поэтому при использовании ЭВМ для обработки таких



массивов одним из основных является вопрос рационального распределения памяти.

Учитывая, что при определенных схемах интерпретации может возникнуть необходимость одновременного хранения нескольких исходных и промежуточных массивов, для их хранения отводятся следующие ЗУ: часть МОЗУ-0, часть МОЗУ-1, 4(8) секции МБ и МФ-0. Указанные ЗУ распределяются под массивы административной системой, которая включает в себя комплекс программ, предназначенных для автоматического распределения памяти и ввода-вывода информации.

На каждый массив, находящийся в памяти машины, заводится паспорт, в котором отражены основные характеристики массива (количество элементов в столбце, номера начального столбца и элемента в столбце и расположение его во всех ЗУ). Паспорт занимает  $10_8$  ячеек (табл. 1). Паспорта всех массивов хранятся в паспортном столе (ПС), который во время работы систем записывается в МОЗУ-0 с 7300 по 7657 ячеек.

Перед ПС находится адресный стол, который занимает  $10_8$  ячеек и записывается с 7270 ячейки МОЗУ-0. В адресном столе хранится информация о распределении ЗУ (табл. 2).  $A_{\text{МОЗУ-1}}$ ,  $A_{\text{МОЗУ-0}}$ ,  $A_{\text{МБ}}$  — начальные ячейки свободного места соответствующих ЗУ.

Основными программами являются программы записи (ПЗАП) и обмена (ПОБ), первая из которых осуществляет накопление массивов с автоматическим распределением памяти и вывод информации на печать, а вторая — выборку из ЗУ необходимых для обработки элементов.

Обращение в ПЗАП имеет вид:

|        |      |      |      |
|--------|------|------|------|
| 057    | 0100 | 0000 | М    |
| 074    | 7752 | ШИФР | 7200 |
| 000    | Э    | 0000 | 7201 |
| 000    | С    | 0000 | 7202 |
| 013    | 7711 | НЭ   | 7203 |
| 000    | ТИП  | 0000 | 7204 |
| М: 000 | 0000 | 0000 | 0000 |
| 050    | 0030 | 6574 | 7160 |

где ШИФР — номер ячейки, содержащий условное трехзначное восьмеричное число — номер накапливаемого массива;

Таблица 1

| Разряды<br>№ ячейки |             | 45-43           | 42-40           | 39-37           | 36-34           | 33-31           | 30-28           | 27-25           | 24-22           | 21-19           | 18-16           | 15-13           | 12-10           | 9-7           | 6-4 | 3-1 |               |  |
|---------------------|-------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|---------------|-----|-----|---------------|--|
| 7300 + К            | ШИФР        | Ic              |                 |                 | Lмб (пзп)       |                 |                 | Lмб (пзп)       |                 |                 | Lмозу-1 (пзп)   |                 |                 | Lмозу-1 (пзп) |     |     | Lмозу-1 (пзп) |  |
| 7301 + К            | Снач        | № зоны МЛ       |                 |                 | Lмб (пзп)       |                 |                 | Lмб (пзп)       |                 |                 | Lмозу-1 (пзп)   |                 |                 | Lмозу-1 (пзп) |     |     | Lмозу-1 (пзп) |  |
| 7302 + К            | Энач        | N Эс            |                 |                 | α мб (пзп)      |                 |                 | Lмозу-1 (пзп)   |                 |                 | Lмозу-1 (пзп)   |                 |                 | Lмозу-1 (пзп) |     |     | Lмозу-1 (пзп) |  |
| 7303 + К            | NN          | δ <sub>1</sub>  | δ <sub>2</sub>  | δ <sub>3</sub>  | δ <sub>4</sub>  | δ <sub>5</sub>  | δ <sub>6</sub>  | δ <sub>7</sub>  | δ <sub>8</sub>  | δ <sub>9</sub>  | δ <sub>10</sub> | δ <sub>11</sub> | δ <sub>12</sub> |               |     |     |               |  |
| 7304 + К            | Σδ          | δ <sub>13</sub> | δ <sub>14</sub> | δ <sub>15</sub> | δ <sub>16</sub> | δ <sub>17</sub> | δ <sub>18</sub> | δ <sub>19</sub> | δ <sub>20</sub> | δ <sub>21</sub> | δ <sub>22</sub> | δ <sub>23</sub> | δ <sub>24</sub> |               |     |     |               |  |
| 7305 + К            | a   b       | δ <sub>25</sub> | δ <sub>26</sub> | δ <sub>27</sub> | δ <sub>28</sub> | δ <sub>29</sub> | δ <sub>30</sub> | δ <sub>31</sub> | δ <sub>32</sub> | δ <sub>33</sub> | δ <sub>34</sub> | δ <sub>35</sub> | δ <sub>36</sub> |               |     |     |               |  |
| 7306 + К            | Энач МОЗУ-0 | Снач МОЗУ-0     |                 |                 | α мозу-0 (поб)  |                 |                 | Снач МБ         |                 |                 |                 |                 |                 |               |     |     |               |  |
| 7307 + К            | Экон МОЗУ-0 | Скон МОЗУ-0     |                 |                 | α мб (поб)      |                 |                 | Скон МБ         |                 |                 |                 |                 |                 |               |     |     |               |  |



Таблица 2

| № ячейки | Разряды |       | 36—28          | 27—26 | 24—13                          | 12—1        |
|----------|---------|-------|----------------|-------|--------------------------------|-------------|
|          | 45—40   | 39—37 |                |       |                                |             |
| 7270     |         |       | A <sub>2</sub> |       | A <sub>2</sub> —A <sub>1</sub> | АМОУ-I      |
| 7271     |         |       | Лмб (пзп)      |       | A <sub>1</sub>                 | ЛМОУ-I      |
| 7272     |         |       |                |       | аМБ (ПЗАП) ТИП-1.2             | № зоны МЛ   |
| 7273     |         |       |                |       | аМБ (ПЗАП) ТИП-3               |             |
| 7274     |         |       |                |       | аМБ (ПОБ)                      |             |
| 7275     |         |       | Лмб (поб)      |       | АМОУ-0                         | ЛМОУ-0      |
| 7276     | Шифр    |       | ΔX             |       | Снач МОЗУ-I                    | Энач МОЗУ-I |
| 7277     |         |       | ΔY             |       | Скон МОЗУ-I                    | Экон МОЗУ-I |

С и Э — ячейки, содержащие номер столбца, для которого производились вычисления, и номер элемента в столбце соответственно;

№ — ячейка, содержащая число элементов в столбце

ТИП — ячейка, содержащая условное число, в зависимости от которого изменяется режим работы программы;

ТИП-0 — обращения к ПЗАП не происходит;

ТИП-1 — происходит накопление массива;

ТИП-2 — накопление массива и выдача его на печать

ТИП-3 — выдача массива на печать без его запоминания.

Перед обращением к ПЗАП засылаемый элемент должен быть записан в 0001 ячейку МОЗУ-1.

Во время работы системы ПЗАП располагается в МОЗУ-1 в 6574—7156 ячейках. Обращение к ней на входном языке имеет вид

ПЗАП, ШИФР, Э, С, №, ТИП;

Программа обмена (ПОБ) предназначена для поиска требуемой информации в ЗУ в записи ее в МОЗУ-1. Обращение к ней имеет вид

|        |      |      |      |
|--------|------|------|------|
| 057    | 0100 | 0000 | М    |
| 074    | 7752 | ШИФР | 7200 |
| 000    | Э    | 0000 | 7201 |
| 000    | С    | 0000 | 7202 |
| 013    | 7711 | №    | 7203 |
| 013    | 7711 | №С   | 7204 |
| 013    | 7711 | ΔХ   | 7205 |
| 013    | 7711 | ΔУ   | 7206 |
| М: 000 | 0000 | 0000 | 0000 |
| 050    | 0030 | 6000 | 7160 |

где ШИФР — номер обрабатываемого массива; С и Э — координаты (столбец, элемент) точки, для которых производятся вычисления; №С и № — количество столбцов и элементов в столбце из которых нужно получить матрицу МОЗУ-1;

ΔХ и ΔУ — интервал выбора столбцов и элементов соответственно.

После окончания работы ПОБ в МОЗУ-1 с ячейки, задан



ной оператором ВВОД (см. ниже) в  $NC \times NЭ$  последовательных ячейках расписывается матрица вида

$$\begin{pmatrix} X_{CЭ}, & X_{CЭ+\Delta X}, & \dots, & X_{CЭ+NЭ \times \Delta X} \\ X_{C+\Delta Y Э}, & X_{C+\Delta Y Э+\Delta X}, & \dots, & X_{C+\Delta Y Э+NЭ \times \Delta X} \\ X_{C+2\Delta Y Э}, & X_{C+2\Delta Y Э+\Delta X}, & \dots, & X_{C+2\Delta Y Э+NЭ \times \Delta X} \\ \dots & \dots & \dots & \dots \\ X_{C+NC \times \Delta Y Э}, & X_{C+NC \times \Delta Y Э+\Delta X}, & \dots, & X_{C+NC \times \Delta Y Э+NЭ \times \Delta X} \end{pmatrix}$$

Обращение в ПОБ на входном языке имеет вид  
ОБМЕН, ШИФР, Э, С, NC, NЭ, ТИП  $\Delta X$ ,  $\Delta Y$ .

Кроме программ записи и обмена в АС входят программы ВВОД, ФАС1, ФАС2 (Формирование адресного стола) и КН (конец накопления).

Программа ВВОД оформлена в виде оператора интерпретирующей системы Ридер и предназначена для ввода исходной информации. На входном языке обращение к оператору имеет вид

ВВОД, ШИФР, ТИП, A1, A2, 3, N3, NN3, ДОП, M, N, NN, S, B1, B2;

где ШИФР — номер вводимого массива; ТИП — условное число, ТИП-1 — ввод чисел, записанных в машинной форме, ТИП-2 — записанных по два числа в ячейку, ТИП-11 и ТИП-12 — довод чисел в машинном виде и по два числа в ячейке соответственно; A1 — начальный адрес МОЗУ-1 для накапливаемых массивов (если A1 = 0, то массивы расписываются с 4000 ячейки); A2 — начальный адрес МОЗУ-1 для обрабатываемого массива (если A2 = 0, то матрица ПОБ расписывается с 5500 ячейки); 3 — номер начальной зоны для данного массива; N3 — количество зон, требуемое для вводимого массива; NN3 — общее число отведенных под массив зон с учетом возможного довода; ДОП — количество зон, требуемых для довода; M и N — количество столбцов и элементов в столбце соответственно во вводимом массиве; NN — количество столбцов, размещаемых в одной зоне; S — интервал группирования для построения гистограммы; B1 и B2 номера МБ (по 4 секции), отведенных под массивы (если B2=0, то отводится только B1). После окончания работы оператора ВВОД вводимый массив располагается в N3 последовательных зонах, начиная с 3, и на него заводится паспорт. На печать выдается максимальное, минимальное, сред-

нее значения исходного поля и гистограмма распределения значений последнего с интервалом группирования  $S$  в графическом и числовом видах.

Операторы ФАС1 и ФАС2 производят формирование адресного стола — первый в случае, если отведено 4 секции второй — если отведено 8 секций. Обращение к ним на входном языке имеет вид

ФАС1, К, К1; ФАС2, К, К1;

где К и К1 — число одновременно обрабатываемых и накапливаемых массивов соответственно.

Оператором КН завершается накопление массивов. После его срабатывания все накапливаемые массивы записываются на МЛ и на перфорацию выдаются их паспорта.

При использовании системы следует учитывать ограничения, накладываемые на исходные и промежуточные массивы.

1. Число массивов не должно превышать 30.

2. Число элементов в накапливаемом массиве не должно превышать

$$NЭ_{\max} \leq \frac{113331}{\text{entier} \left( \frac{2n + \kappa}{\kappa} \right)},$$

где  $n$  — число одновременно накапливаемых массивов;  
 $\kappa$  — число отведенных секций МБ.

3. Число элементов в столбце не должно превышать

$$NЭ_{\text{ст max}} \leq \frac{\text{entier} \left( \frac{4096 \times \kappa}{n} \right)}{N_{\text{C max}} \times \Delta X},$$

где  $N_{\text{C max}}$  — максимальное число столбцов, которые могут одновременно потребоваться при вычислениях.

4. Число одновременно вводимых элементов не должно превышать 3063.

В случае невыполнения хотя бы одного из условий происходит прерывание работы системы с указанием характера ошибки.

#### 4. Пример программы

Пусть требуется вычислить и вывести на печать энтропию и дисперсию исходного поля, пересчитанного в верхнее полупространство на уровень 200 метров, и получить значения  $V_{zz}$



на уровнях 50 и 100 метров без вывода на печать. Тогда в общем виде программа на входном языке будет иметь вид

READ: ВВОД, ШИФР, ТИП, A1, A2, 3, N3, NN3, ДОП, M, N, NN, S, B1, B2;

ФАС2, 1, 3;

ЦИКЛ, M, I, [NC — 20]; ЦИКЛ, M1, J, [NЭ — 20];

ПАЛ1, ШИФР, I, J;

ВВЕРХ, 1, 200; VZZ, 1, 50; VZZ, 1, 100;

M: ; M1 : КН;

ЦИКЛ, M2, I, [NC — LC]; ЦИКЛ, M3, J, [NЭ — LЭ];

ENTR, ШИФР, 2, C, Э, ΔX, ΔY, LЭ, LC;

DS, ШИФР, 2, C, Э, ΔX, ΔY, LЭ, LC;

M2: ; M3 : КН; СТОП, \*\*.

Здесь LC и LЭ — параметры палетки для вычисления энтропии и дисперсии.

На данном этапе разработки система СИГМА позволяет решать весьма ограниченный круг геофизических задач, что связано лишь с недостаточно развитой библиотекой стандартных операторов. После накопления БСО система станет весьма гибким и универсальным аппаратом, который позволит ускорить интерпретацию гравимагнитных полей и повысить ее качество.

#### ЛИТЕРАТУРА

1. Арайс Е. А., Дмитриев В. М. Программа моделирования и анализа радиоэлектронных схем (МАРС-1). Вопросы программирования и автоматизации проектирования. Вып. 1. Томск, Изд-во ТГУ, 1971.

2. Арайс Е. А., Сибиряков Г. В., Шутенков А. В. Система интерпретации для решения больших задач. Вопросы программирования и автоматизации проектирования. Вып. 1. Томск, Изд-во ТГУ, 1971.

3. Арайс Е. А., Шутенков А. В. Язык МАРС. Настоящий сборник.

4. Шутенков А. В. Транслятор с входного языка системы МАРС. Настоящий сборник.

## **АНАЛИЗ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ВЫПОЛНЕНИЯ ГРАВИРОВАЛЬНЫХ РАБОТ НА ФРЕЗЕРНЫХ СТАНКАХ С ЧИСЛОВЫМ ПРОГРАММНЫМ УПРАВЛЕНИЕМ**

**А. С. БАТУРИН, В. И. ЛИВШИЦ, А. К. МАРТЫНОВ, В. П. СУШКО**

Панели с нанесенной на них графической информацией в виде букв, цифр и знаков (символов), а также другие детали с надписями являются неперменной частью современных машин, приборов и аппаратов. Наибольшая долговечность графической информации обеспечивается при нанесении ее на твердый материал методом гравирования, т. е. вырезания концевой конической фрезой [1].

Выполняется эта операция в настоящее время чаще всего на копировально-фрезерных станках с пантографом модели 6Л463. В качестве программносителя используется копир в увеличенном масштабе, изготовленный слесарным способом вручную.

Существенно вырастает производительность гравировальных работ при переводе их на фрезерные станки с числовым программным управлением (ЧПУ). Кроме того, отпадает необходимость в ручном изготовлении копиров. Рабочий-оператор освобождается от утомительного ручного труда и может обслуживать несколько станков. Подготовка производства переносится в сферу инженерного труда.

Конечно, наибольший эффект был бы достигнут при наличии в номенклатуре фрезерных станков с ЧПУ станка, специально спроектированного для выполнения гравировальных работ и имеющего прежде всего требуемые числа оборотов шпинделя ( $30 \div 35$  тысяч оборотов в минуту). Однако такие станки пока что промышленностью не выпускаются, и для гравировальных работ используются вертикально-фрезерные станки с ЧПУ 1-го габарита, например, модели 6М11ФЗ.

Вместе с тем некоторые факторы, связанные со станками с ЧПУ, ведут к дополнительным затратам: необходимость использовать труд высококвалифицированных специалистов —



наладчика и электроника; высокая стоимость станков с ЧПУ; использование комплекса записи программ на магнитную ленту; использование универсальной ЭВМ для расчета управляющих программ. Поэтому необходима экономическая оценка эффективности сопоставимых вариантов выполнения гравировальных работ на станках с ручным управлением и на станках с ЧПУ.

При определенных условиях внедрение станков с ЧПУ обеспечивает:

1. Высвобождение производственной площади.
2. Рост фондоотдачи.
3. Повышение производительности труда.
4. Улучшение качества продукции.
5. Сокращение затрат на оснастку.
6. Высвобождение рабочей силы.
7. Сокращение длительности производственного цикла.
8. Сокращение незавершенного производства.
9. Улучшение условий труда рабочих.
10. Снижение брака.

Подготовка программ для станков с ЧПУ вызывает необходимость использования дорогостоящего оборудования: универсальных ЭВМ, интерполяторов, координатографов. Чтобы иметь высокую эффективность всего комплекса, необходимо возможно более полное использование упомянутого оборудования, так как применение ЭВМ и интерполятора для одного станка не обеспечивает их полную загрузку во времени. Оборудование для подготовки программ должно использоваться по времени минимум в течение двух смен. В дальнейших расчетах будет предполагаться выполнение этого условия, и на подготовку конкретной программы будет относиться только часть затрат пропорционально времени использования оборудования.

Важнейшее значение для оценки экономической эффективности перевода гравирования на станки с ЧПУ имеет метод подготовки и расчета управляющих программ (УП). Графическая информация при гравировании имеет весьма сложные контуры (существенно превосходящие по сложности очертания деталей машин и приборов), хотя и составленные из тривиальных элементов — прямых и дуг окружностей.

Ручной расчет УП в этом случае совершенно неприемлем. Низкую эффективность дают и существующие способы автоматизации расчета на ЭВМ с помощью проблемно-ориентированных машинных языков САП-2 и СППС.

Причина этого заключается в том, что указание на детали и детали требуют разбивки на отдельные составляющие элементы объектов графической информации — букв, цифр и знаков, а затем описания этих элементов одним из известных способов для определения точек их пересечения или сопряжения. Ввиду сложности объектов графической информации работа по разбивке их на элементы и описанию, выполняемая технологом-программистом или расчетчиком, является весьма трудоемкой и сопровождается многими ошибками, затрудняющими внедрение УП на станке. Эта работа по своей трудоемкости становится соизмеримой с теми операциями расчета, которые выполняются в САП-2 и СППС с помощью ЭВМ препроцессором, процессором и постпроцессором. Поэтому этот метод подготовки и расчета управляющих программ может быть назван полуматематизированным.

Для увеличения эффективности подготовки и расчета УП в случае гравирования на станках с ЧПУ следует стремиться к созданию полностью автоматизированного метода, при котором объект графической информации задается целиком одним кодовым знаком в программе исходной информации для ЭВМ. Такая система автоматизации и расчета УП для гравировальных работ создана в Томском политехническом институте и получила название система «Гравер».

При экономическом сравнении вариантов выполнения работ необходимо располагать критерием эффективности. Экономическим критерием эффективности являются годовые приведенные затраты [2], определяемые по формуле:

$$Z_i = C_i + E_n \cdot K_i, \quad (1)$$

где  $Z_i$  — годовые приведенные затраты по  $i$ -му варианту, руб/год;

$C_i$  — себестоимость годового выпуска продукции по  $i$ -му варианту, руб/год;

$E_n$  — нормативный коэффициент сравнительной экономической эффективности, руб/руб. год. Для машиностроения  $E_n = 0,10 \div 0,15$  [2];

$K_i$  — единовременные капитальные вложения по  $i$ -му варианту, руб.

Анализируя варианты по формуле годовых приведенных затрат (1), считают экономически наиболее выгодным тот вариант, у которого эта сумма будет иметь минимальное значение.

В тех случаях, когда капитальные вложения  $K$  используются частично, только на время обработки изделия (детали),



формула приведенных затрат может быть представлена в следующем виде [3]:

$$Z_i = C_i + E_n \cdot K_i \frac{T_{шк}}{\Phi_o K_u}, \quad (2)$$

где  $T_{шк}$  — штучно-калькуляционное время;

$\Phi_o$  — годовой действительный фонд времени работы станка;

$K_u$  — коэффициент использования станка по штучному времени.

В приведенной формуле (2) капитальные вложения включаются пропорционально времени использования станка.

Так как формула приведенных затрат учитывает не только себестоимость, но и капитальные вложения, она используется как количественный критерий сравнительной экономической эффективности. Разность годовых приведенных затрат по сравниваемым вариантам дает годовой экономический эффект  $\mathcal{E}$ :

$$\mathcal{E} = Z_i - Z_{i+1} = (C_i + E_n \cdot K_i) - (C_{i+1} + E_n \cdot K_{i+1}). \quad (3)$$

Годовая экономия сравниваемых вариантов определяется разностью годовых эксплуатационных расходов (годовой себестоимости):

$$\mathcal{E}_z = C_i - C_{i+1}, \quad (4)$$

где  $\mathcal{E}_z$  — годовая экономия, руб/год.

При определении экономической эффективности различных вариантов выполнения гравирования за базу для сравнения принимаются показатели лучших имеющихся методов осуществления этих работ. В настоящее время наиболее механизированным методом выполнения гравировальных работ является гравирование на копировально-фрезерных станках с пантографом [1]. Поэтому в дальнейших расчетах этот метод принят за базовый.

Сравниваемые варианты выполнения гравировальных работ должны быть сопоставимыми, то есть различия между ними должны сводиться только к различию затрат. Сопоставимость должна обеспечиваться во времени, по ценам и тарифным ставкам заработной платы, которые используются для расчетов, по элементам затрат. Все экономические показатели должны определяться по действующим на момент расчета оптовым ценам, тарифам на электрическую энергию и ставкам заработной платы.

Рассматриваемые варианты выполнения работ должны быть приведены в сопоставимый вид также по объему выполняемых работ, качеству и срокам осуществления.

## Определение величины капитальных вложений по вариантам

К капитальным вложениям относятся единовременные затраты, связанные с внедрением конкретного варианта выполнения работ. В общем виде величина капитальных затрат может быть определена по формуле:

$$K = K_n + K_0 \pm \Delta O_c + K_a - K_{\text{выс}} + K_{\text{осн}} + K_{\text{инв}} + K_{\text{см}}, \quad (5)$$

- где  $K$  — капитальные затраты, связанные с внедрением определенного метода гравирования, руб.;
- $K_n$  — предпроизводственные затраты на разработку САП для гравировальных работ, руб.;
- $K_0$  — затраты на оборудование и производственные здания, руб.;
- $\Delta O_c$  — изменение величины оборотных средств («+» увеличение, «—» сокращение), руб.;
- $K_{\text{выс}}$  — остаточная стоимость высвобожденного оборудования, устройств, зданий, сооружений, которые будут использованы или реализованы на сторону, руб.;
- $K_{\text{осн}}$  — капитальные вложения в оснастку, руб.;
- $K_a$  — остаточная стоимость ликвидируемого оборудования, устройств, зданий, сооружений, которые не нашли применения и реализации, руб.;
- $K_{\text{инв}}$  — капитальные вложения в инвентарь, руб.;
- $K_{\text{см}}$  — капитальные вложения в смежные производства, руб.

Предпроизводственные затраты составляют расходы на научные исследования, направленные на разработку САП «Гравер», включая разработку, отладку и внедрение программ для ЭВМ, на составление инструкций и других руководящих документов. Эти затраты должны распределяться между всеми станками, которые будут переведены на работу по программам, подготовленным с помощью системы «Гравер». Затраты по разработке управляющих программ для конкретного изделия, на котором необходимо выполнить гравировальные работы, относятся не к капитальным, а к эксплуатационным расходам, т. е. входят в себестоимость.

Затраты на оборудование включают цену приобретения, транспортные расходы и затраты на монтаж.

Изменение нормируемых оборотных средств определяют по формуле:

$$\Delta O_c = O_{c1} - O_{c2}, \quad (6)$$



где  $Oc_1$  — объем оборотных средств в первом варианте гравирования, руб.;

$Oc_2$  — объем оборотных средств во втором сравниваемом варианте, руб.

Остаточная стоимость высвобождаемого (ликвидируемого) оборудования, зданий, сооружений определяется по формуле:

$$K_{\text{выс}} = B_c (1 - H_a \cdot T_a), \quad (7)$$

где  $B_c$  — восстановительная стоимость действующего производственного оборудования, зданий, сооружений, руб.;

$H_a$  — годовая норма амортизации на реновацию (полное восстановление) оборудования;

$T_a$  — продолжительность эксплуатации, год.

Капитальные вложения целиком входят в состав годовых приведенных затрат (1) в том случае, когда они осуществляются лишь для данного варианта гравирования и не могут быть использованы для других целей. Однако зачастую капитальные вложения, используемые для конкретного варианта, наряду с этим могут применяться и для других целей. Например, при расчете управляющих программ выполнения гравирования на станках с ЧПУ ЭВМ «Минск-32» используется только частично для этих целей, и потому в составе капитальных вложений варианта гравирования стоимость ЭВМ учтена пропорционально времени ее использования. Интерполятор также обслуживает несколько станков. Поэтому в дальнейших расчетах предполагается полное использование интерполятора по времени.

### Определение себестоимости гравирования по вариантам

Для анализа вариантов гравирования целесообразно использовать технологическую, а не хозрасчетную себестоимость по бухгалтерскому методу. Одинаковые затраты для всех методов не включаются в технологическую себестоимость. Благодаря этому удастся более точно выяснить экономические различия сравниваемых вариантов гравирования.

В общем виде технологическая себестоимость может быть представлена:

$$C = Z_a + Z_p + Z_m + Z_k + Z_p + Z_{np} + Z_{nl} + Z_{ny} + Z_c, \quad (8)$$

где  $C$  — годовая технологическая себестоимость изготовления изделий, руб/год;

$Z_3$  — годовая заработная плата рабочих (основная и дополнительная), включая отчисления на социальное страхование, руб./год;

$$Z_3 = \left( \frac{H_{ст}}{C_{ст}} + \frac{H_n}{C_n} + \frac{H_э}{C_э} \right) H_{см}, \quad (9)$$

где  $H_{ст}$  — среднегодовая заработная плата одного рабочего (станочника или оператора) в расчете на год работы станков, руб./год;

$C_{ст}$  — количество станков, обслуживаемых одним станочником или оператором;

$H_n$  — среднегодовая заработная плата одного наладчика, руб./год;

$C_n$  — количество станков, обслуживаемых одним наладчиком в смену;

$H_э$  — среднегодовая заработная плата одного электрика, руб./год;

$C_э$  — количество станков, обслуживаемых одним электриком;

$H_{см}$  — сменность работы оборудования;

$Z_э$  — годовые затраты на силовую электроэнергию, руб./год;

$Z_c$  — годовые затраты на межремонтное обслуживание оборудования, руб./год;

$Z_T$  — годовые затраты на текущий ремонт оборудования, руб./год;

$Z_k$  — годовые затраты на капитальный ремонт оборудования, руб./год;

$Z_p$  — годовые амортизационные отчисления на реновацию, руб./год;

$Z_{пр}$  — годовые затраты на амортизацию приспособлений, руб./год;

$Z_{пл}$  — годовые затраты на содержание и амортизацию здания, занимаемого оборудованием, руб./год.

Все перечисленные составляющие вычисляются по обычным формулам, приводимым в справочно-методической литературе (например, [2]).

$$Z_{ny} = H_{ny} \cdot a, \quad (10)$$

где  $Z_{ny}$  — годовые затраты на подготовку управляющих программ для станков с ЧПУ, руб./год;

$H_{ny}$  — затраты на подготовку управляющих программ для одного наименования детали, руб.;



*a* — число наименований обрабатываемых деталей за год, закрепленное за данным станком.

### **Учет затрат на подготовку управляющих программ**

При полуавтоматизированном методе подготовки гравировальных программ, когда, как указывалось выше, ЭВМ используется для расчета координат опорных точек траектории фрезы и координатных разностей на основе ранее существовавших САП (например, САП-2, СППС), выполняются следующие этапы работы:

1. Технологическая подготовка: подбор номенклатуры деталей для выполнения гравировальных работ на станках с ЧПУ; вычерчивание гравлируемых элементов в увеличенном масштабе (выкладка чертежа); определение последовательности выполнения операций; разработка траектории движения инструмента; оформление расчетно-технологических карт (РТК); разбивка объектов гравирования на типовые составляющие элементы; описание типовых элементов и траекторий на машинных языках САП; выбор инструмента и режимов резания; проектирование приспособления.

2. Математическая подготовка: перфорация данных для ввода в ЭВМ, расчет координат опорных точек и координатных разностей, формирование программ по кадрам и их кодирование в коде интерполятора на ЭВМ, выдача результатов расчета на печать, перфорация результатов на бумажную ленту для ввода в интерполятор.

3. Контроль правильности расчета на ЭВМ и перфорации на ленте.

4. Интерполирование и запись на магнитную ленту.

5. Контрольное вычерчивание траектории на координатографе.

6. Испытание магнитной ленты на станке с ЧПУ, корректировка программ по результатам испытания.

7. Внедрение программ.

При автоматизированном методе подготовки гравировальных программ на основе системы «Гравер» состав работы существенно уменьшается. Исчезают такие трудоемкие работы, как выкладка чертежа, разработка траектории движения инструмента, оформление РТК, разбивка знаков на типовые элементы. Описание информации на языке «Гравер» выполняется очень просто и быстро, а в целом ряде случаев — при гравировании буквенно-цифровой информации — прямым текстом гравлируемой надписи.

Общие работы для сравниваемых вариантов (подбор номенклатуры деталей для гравирования; определение последовательности выполнения операций; выбор инструмента и режимов работы, проектирование приспособлений) исключены из анализа, так как в обоих вариантах они примерно одинаковы по затратам. В случае существенной разницы затрат общие работы должны учитываться в составе затрат на подготовку программ соответствующих вариантов.

В общем виде затраты на подготовку программы полуавтоматизированным методом для одного наименования детали могут быть определены:

$$H_{пу} = Z_{вч} + Z_{ртек} + Z_{эвм} + Z_{к} + Z_{пк} + Z_{мл} + Z_{ц} + Z_{ко}, \quad (11)$$

где  $Z_{вч}$  — затраты на вычерчивание элементов гравирования (выкладку чертежа):

$$Z_{вч} = \frac{Ч_ч \cdot H_ч \cdot Z_{мт}}{\Phi_{\partial т}}, \quad (12)$$

где  $Ч_ч$  — число чертежей нормального формата и определенной категории сложности;

$H_ч$  — норма времени на один чертеж данной категории сложности, час.;

$Z_{мт}$  — месячная заработная плата техника-чертежника соответствующей квалификации, руб./мес.;

$\Phi_{\partial т}$  — месячный фонд времени работы техника-чертежника, час./мес.;

$Z_{ртек}$  — затраты по разработке технологии и расчетно-технологической карты (РТК), включающей в себя следующие работы: разработку траектории движения инструмента; оформление РТК; разбивку знаков на типовые элементы; описание информации на языке САП.

$$Z_{ртек} = \frac{T_{отк} \cdot Z_{м}}{\Phi_{\partial}}, \quad (13)$$

где  $T_{ртек}$  — трудоемкость в нормо-часах указанных работ;

$Z_{м}$  — месячная зарплата технолога соответствующей квалификации, руб./мес.;

$\Phi_{\partial}$  — месячный фонд времени работы технолога, час./мес.;

$Z_{эвм}$  — затраты по расчету координат опорных точек и координатных разностей на ЭВМ, руб.:



$$Z_{эвм} = \chi_{эвм} \cdot T, \quad (14)$$

где  $\chi_{эвм}$  — цена одного часа работы ЭВМ, руб./час.;  
 $T$  — машинное время расчета координат опорных точек и координатных разностей на ЭВМ, час.;  
 $Z_k$  — затраты на формирование программ по кадрам и их кодирование в коде интерполятора на ЭВМ:

$$Z_k = \chi_k \cdot T, \quad (15)$$

где  $\chi_k$  — машинное время формирования программ по кадрам и их кодирования на ЭВМ, час.;  
 $Z_{пк}$  — расходы, связанные с перфорацией программноносителя для ЭВМ:

$$Z_{пк} = Z_{уп} \cdot \kappa_{вн} \cdot T_{пер} \cdot \kappa_{\partial}, \quad (16)$$

где  $Z_{уп}$  — часовая основная зарплата перфораторщика, руб./час;  
 $\kappa_{вн}$  — коэффициент выполнения норм;  
 $T_{пер}$  — трудоемкость перфорирования, час;  
 $\kappa_{\partial}$  — коэффициент, учитывающий дополнительную зарплату и начисления на социальное страхование.  
 $Z_{мл}$  — расходы на магнитную ленту, руб.;

$$Z_{мл} = T_z \cdot C_n \cdot \zeta_{л}, \quad (17)$$

где  $T_z$  — длительность записи управляющей программы, мин.;  
 $C_n$  — скорость протягивания магнитной ленты, м/мин.;  
 $\zeta_{л}$  — стоимость 1 м магнитной ленты, руб./м.;  
 $Z_u$  — расходы, связанные с работой интерполятора с устройством записи на магнитную ленту:

$$Z_u = T_z \cdot C_u, \quad (18)$$

где  $T_z$  — машинное время записи управляющей программы, час.;  
 $C_u$  — стоимость одного часа работы интерполятора, руб./час.

Часовая стоимость работы интерполятора определяется по формуле:

$$C_u = \frac{Z + A_{инт} + Z_u + P_{уп}}{\Phi_{\partialи} \cdot \kappa_z}, \quad (19)$$

где  $Z$  — зарплата электроников, обслуживающих интерполятор, руб./год:

$$Z = \chi_z \cdot Z_z \cdot \kappa_{\partial} \cdot \chi_m, \quad (20)$$

где  $Ч_э$  — численность электроников, чел.;  
 $З_э$  — месячный оклад электроника, руб./мес.;  
 $к_д$  — коэффициент, учитывающий премии и отчисления в соцстрах;  
 $Ч_м$  — число месяцев в году;  
 $A_{инт}$  — амортизационные отчисления от стоимости интерполятора, руб./год;

$$A_{инт} = \frac{Ц_{инт} \cdot H_a}{100}, \quad (21)$$

где  $Ц_{инт}$  — балансовая стоимость интерполятора, руб.;  
 $H_a$  — норма амортизационных отчислений, равная 10% в год;  
 $\Phi_{ди}$  — действительный годовой фонд работы интерполятора, час./год;  
 $к_з$  — коэффициент загрузки интерполятора по времени;  
 $Э_и$  — затраты на силовую электроэнергию, которые определяются:

$$Э_и = M_{и} \cdot \Phi_{ди} \cdot Ц_э, \quad (22)$$

где  $M_{и}$  — потребляемая мощность интерполятора, квт;  
 $\Phi_{ди}$  — действительный годовой фонд времени работы интерполятора, час./год;  
 $Ц_э$  — стоимость 1 квтч электроэнергии по двухставочному тарифу, руб./квтч;  
 $P_{ир}$  — расходы на текущие ремонты интерполятора, которые можно укрупненно принять 3% от стоимости интерполятора.  
 $З_{ко}$  — затраты по контролю записи на перфоленте и контролю программносителя на координатографе и на станке:

$$З_{ко} = \frac{T_{ко} \cdot З_{мк}}{\Phi_{дк}}, \quad (23)$$

где  $T_{ко}$  — общая трудоемкость контроля, корректировки и отладки программ, час.;  
 $З_{мк}$  — месячная заработная плата инженера по контролю программы, руб./мес.;  
 $\Phi_{дк}$  — месячный фонд времени инженера по контролю программ, час./мес.

При разработке программ на основе системы „Гравер“ с более широким использованием ЭВМ (автоматизиро-



ванный метод) затраты на подготовку программы  $H_{пу(a)}$  определяются по формуле:

$$H_{пу(a)} = Z_{оп} + Z_{пк} + Z_{эвм} + Z_{к} + Z_{мл} + Z_{и} + Z_{ко}, \quad (24)$$

где  $Z_{оп}$  — затраты по описанию информации о конкретной детали, подлежащей гравированию, на языке системы „Гравер“:

$$Z_{оп} = \frac{Z_{мт} \cdot T_{оп}}{\Phi_{дм}}, \quad (25)$$

где  $T_{оп}$  — трудоемкость описания технологом-программистом конкретной детали, час.;

$Z_{мт}$  — месячная зарплата технолога-программиста, руб./мес.;

$\Phi_{дм}$  — месячный фонд времени работы технолога-программиста, час./мес.

Остальные составляющие затрат в формуле (24) аналогичны составляющим в формуле (11), однако их значения могут отличаться при подготовке гравировальных программ полуавтоматизированным и автоматизированным методами.

Производственные испытания показали, что подготовка гравировальных программ с использованием систем «Гравер» обеспечивает экономию затрат труда технолога-программиста в 4 ÷ 5 раз, а снижение трудоемкости перфорирования данных для ЭВМ «Минск-32» — в 6 раз против затрат труда на те же работы при использовании системы СППС.

### Условия сравнения вариантов по производительности

Немаловажное значение в оценке эффективности станков с ЧПУ имеет сравнение вариантов выполнения работ по производительности. При сопоставлении производительности различных станков для выполнения гравировальных работ необходимо выдерживать следующие условия:

1. Должны быть правильно подобраны детали-представители. Эти детали по своим геометрическим и технологическим параметрам выбираются таким образом, чтобы они позволяли в полной мере использовать эксплуатационные возможности станков. По удельному весу в производстве они должны составлять не менее 70—80% общего объема всех деталей, обрабатываемых на данных станках. Важно при этом также, чтобы детали-представители охватывали

крайние типоразмеры обрабатываемых изделий. Обычно выбираются три детали-представителя.

2. Технологические процессы на детали-представители должны разрабатываться по одной форме в пооперационном разрезе.

3. Техническое нормирование должно осуществляться по одной и той же методике и на основе одних и тех же нормативов времени.

4. Производительность станков с ЧПУ определяется по снижению трудоемкости:

$$C_m = 100 - \frac{T_{шк\ cр\ чпу}}{T_{шк\ cрп}} \cdot 100, \quad (26)$$

где  $C_m$  — снижение трудоемкости на станке с ЧПУ, %;  
 $T_{шк\ cр\ чпу}$  — средневзвешенное штучно-калькуляционное время обработки деталей на станке с ЧПУ;

$T_{шк\ cрп}$  — то же на станке с пантографом.

Рост производительности труда может быть определен по формуле:

$$P_{пт} = \frac{C_m \cdot 100}{100 - C_m}, \quad (27)$$

где  $P_{пт}$  — рост производительности труда (повышение нормы выработки), %.

В расчетах производительности следует учесть, что станок с ЧПУ наряду с оператором обслуживают наладчик и электроник. Если принять, что один оператор обслуживает 2 станка, а наладчик и электроник — по 5 станков, то на один станок приходится:  $1/2 + 1/5 + 1/5 = 0,9$ , т. е. около одного рабочего, как и на станке-аналоге. Только в этом случае расчеты производительности будут верными.

### Обсуждение результатов сравнения вариантов

Разработанная методика комплексной экономической оценки эффективности различных вариантов выполнения гравировальных работ базируется на положениях о том, что применение новых методов должно обеспечивать все главные экономические показатели [4]:

экономии на годовых приведенных затратах;

окупаемость дополнительных капитальных вложений за нормативные сроки;



рост производительности труда;  
экономия производственной площади;  
увеличение фондоотдачи.

В таблице приведены результаты сравнения по экономической эффективности четырех вариантов выполнения гравировальных работ для изготовления детали-представителя (станочной панели средней сложности).

Во всех рассматриваемых вариантах гравирование производится однопроходным фрезерованием буквенной и цифровой информации, которая наносится на металлические пластинки (панели или платы) небольшой толщины. Существенное различие вариантов состоит в способах осуществления гравирования, режимах работы станка и методах подготовки гравировальных программ.

В первом варианте гравирование производится вручную на копировально-фрезерном станке с пантографом модели 6Л463.

Во втором варианте гравирование осуществляется автоматически от заранее подготовленной программы на вертикально-фрезерном станке с ЧПУ модели 6М11Ф3. Этот станок по своим техническим показателям не обеспечивает оптимальных режимов гравирования, так как имеет недостаточное число оборотов шпинделя (предел 2000 об/мин, а необходимо по расчету 34000 об/мин.) и малую скорость рабочих перемещений стола (500 мм/мин., а необходимо 3400 мм/мин.).

Поскольку подготовка гравировальных программ для станков с ЧПУ может осуществляться различными системами САП, рассмотрены возможности применения двух систем. Во втором варианте используется система подготовки программ СППС (полуавтоматизированная подготовка программ). В третьем варианте применяется также станок модели 6М11Ф3 с ЧПУ, но программы разрабатываются с использованием системы «Гравер», которая специально предназначена для подготовки управляющих программ гравирования буквенной и цифровой информации (автоматизированная подготовка программ). В четвертом варианте применены расчетные режимы гравирования и подготовка программ по системе «Гравер». Этот вариант предполагает модернизацию станка 6М11Ф3 или создание нового станка на его основе.

Экономическое сравнение рассматриваемых вариантов проведено с учетом сопоставимости. В данном случае вари-

Т а б л и ц а

Результаты расчета экономической эффективности по вариантам

| п/п № | Наименование показателей              | Ед. изм.       | I вариант         |          |          | II вариант | III вариант | IV вариант |
|-------|---------------------------------------|----------------|-------------------|----------|----------|------------|-------------|------------|
|       |                                       |                | без коэф. привед. | K = 2,25 | K = 17,4 |            |             |            |
| 1     | Годовые приведенные затраты           | руб. год       | 4478              | 10900    | 78000    | 7170       | 7000        | 7000       |
| 2     | Экономия на приведенных затратах:     | руб./год       | —                 | —        | —        | 3730       | —           | —          |
|       | а) II вар. относительно I варианта    | "              | —                 | —        | —        | —          | 3900        | —          |
|       | б) III вариант относительно I вар.    | "              | —                 | —        | —        | —          | 170         | —          |
|       | в) III вар. относительно II вар.      | "              | —                 | —        | —        | —          | —           | —          |
|       | г) IV вар. относительно I вар.        | "              | —                 | —        | —        | —          | —           | 71000      |
| 3     | Окупаемость дополн. капиталовложений: | год            | —                 | —        | —        | 2,0        | —           | —          |
|       | а) II вар. относительно I вар.        | "              | —                 | —        | —        | —          | 2,0         | —          |
|       | б) III вар. относительно I вар.       | "              | —                 | —        | —        | —          | 1,5         | —          |
|       | в) III вар. относительно II вар.      | "              | —                 | —        | —        | —          | —           | —          |
| 4     | Рост производительности труда:        | %              | —                 | —        | —        | 112        | —           | —          |
|       | а) II вар. относительно I вар.        | "              | —                 | —        | —        | —          | 112         | —          |
|       | б) III вар. относительно I вар.       | "              | —                 | —        | —        | —          | —           | 1540       |
|       | в) IV вар. относительно I вар.        | "              | —                 | —        | —        | —          | —           | 25         |
| 5     | Занимаемая производственная площадь   | м <sup>2</sup> | 5,5               | 11       | 95       | 25         | 25          | 70         |
| 6     | Экономия произв. площади              | "              | —                 | —        | —        | —          | —           | —          |
| 7     | Фондоотдача                           | шт. руб. год   | 9,2               | 9,2      | 9,2      | 2,98       | 2,95        | 22,4       |



анты различаются производительностью, поэтому капитальные вложения и себестоимость годового объема гравировальных работ приводятся с помощью коэффициента приведения затрат, отражающего сравнительную производительность вариантов относительно базового первого варианта. При этом число рабочих на одном станке принято одинаковым во всех вариантах.

Во всех расчетах предполагается, что запись программ на магнитную ленту производится на комплексе «Программа Л-68» [5]. Затраты по записи на магнитную ленту включены через себестоимость одного часа работы комплекса. Так как один комплекс «Программа Л-68» может обслуживать большое число станков с ЧПУ, капитальные вложения его не учитывались в расчетах приведенных затрат. Дополнительная проверка показала, что даже в том случае, если капитальные вложения по комплексу «Программа Л-68» отнести к одному станку с ЧПУ, четвертый вариант по-прежнему оказывается эффективным, но при этом фондоотдача возрастает всего на 10% в сравнении с базовым первым вариантом.

Второй вариант (станок 6М11Ф3 и подготовка программ по СППС) обеспечивает в сравнении с 1-м вариантом годовую экономию на приведенных затратах 3730 руб., окупаемость дополнительных капитальных затрат за два года, рост производительности труда на 112%, но требует большей производственной площади (24,6 м<sup>2</sup> против 12 м<sup>2</sup>) и снижает фондоотдачу (2,98 шт./руб. год против 9,2 в 1-м варианте). Таким образом, вариант не является наилучшим.

Третий вариант характеризуется аналогичными данными: подготовка программ по системе «Гравер» дает экономию на приведенных затратах в сравнении со вторым вариантом в 170 руб. в год.

Четвертый вариант (станок с расчетными режимами гравирования и подготовка программ по системе «Гравер») обладает наивысшей экономичностью. Годовой экономический эффект составляет по расчету 71000 руб. на станок. Однако на данной стадии разработки нельзя считать полученную цифру абсолютно достоверной, так как станок с необходимыми параметрами еще не спроектирован. В связи с этим, воспользовавшись рекомендациями [6], можно скорректировать полученный годовой экономический эффект с учетом вероятности успеха на данной стадии разработки ( $P = 0,4$ ), то есть уменьшить значение годового эффекта

до  $0,4 \cdot 71050 = 28000$  руб./год, что, округляя, можно приближенно принять за 30000 руб./год. Следует отметить, что по IV варианту обеспечивается значительный рост производительности труда, высвобождение производственной площади и увеличение фондоотдачи.

На эффективность применения станков с ЧПУ существенное влияние оказывает применяемая система подготовки программ, так как затраты по подготовке программ достигают 15 и более процентов себестоимости годового объема гравирования. Сопоставление годовых затрат на подготовку программ позволяет установить, что для гравировальных работ выгодно применять систему «Гравер».

Таким образом, наибольшую эффективность дает IV вариант за счет того, что в этом случае на станке возможно реализовать расчетные режимы резания. Однако даже обработка по III варианту приводит к существенному росту производительности труда и экономии на приведенных затратах. Следует отметить, что при этом резко улучшаются условия труда, становится более творческим его содержание.

Необходимо также подчеркнуть, что эффективность обработки по III и IV вариантам существенно возрастает с увеличением числа гравлируемых знаков на панели. Все расчеты выполнены для двухсменной работы станков с ЧПУ при высоком коэффициенте загрузки — до 0,85.

Выполненный анализ позволяет сделать вывод о том, что внедрение станков с ЧПУ для автоматизации производства изделий должно сочетаться с внедрением в технологическую подготовку производства методов автоматизации инженерного труда с помощью ЭВМ. Лишь такое сочетание может дать экономический эффект по всем основным показателям.

#### ЛИТЕРАТУРА

1. Улановский О. О. Механизация граверных работ. Л., «Машиностроение», 1971.
2. Методика определения экономической эффективности металлорежущих станков и автоматических линий, руководящие материалы. МСИП, ЭНИМС, М., 1971.
3. Булкин А. П., Догаев Ю. М. Расчет эффективности перспективной технологии на предприятиях. М., «Экономика», 1972.
4. Типовая методика определения экономической эффективности капитальных вложений. М., «Экономика», 1969.
5. Никитенко В. Д. Подготовка программ для станков с программным управлением. М., «Машиностроение», 1973.
6. Вегер Л. Л. Расчет экономической эффективности и ее обеспечение в условиях неопределенности. — В сб.: Проблемы управления научными исследованиями. М., «Наука», 1973.
7. Мирошников Л. П. Применение станков с программным управлением. Харьков, «Прапор», 1972.



## **АВТОМАТИЗАЦИЯ ПОДГОТОВКИ УПРАВЛЯЮЩИХ ПРОГРАММ ДЛЯ ВЫПОЛНЕНИЯ ГРАВИРОВАЛЬНЫХ РАБОТ НА ФРЕЗЕРНЫХ СТАНКАХ С ЧПУ**

**М. Н. БОГОЛЮБОВА, В. И. ЛИВШИЦ, А. К. МАРТЫНОВ**

Во многих отраслях машиностроения и приборостроения гравировальные работы занимают значительное место. В настоящее время гравирование производится на копировально-фрезерных станках с пантографами. Однако этим станкам свойственны недостатки, присущие механическим способам копирования, главными из которых являются весьма большие усилия, необходимые для перемещения инструмента относительно заготовки; большие давления, возникающие на малой поверхности контакта копира и щупа и ведущие к износу соприкасающихся поверхностей, снижению точности обработки; применение копиров, шаблонов, трафаретов, первоосновой создания которых является ручное гравирование; зависимость качества гравирования от квалификации гравера [1].

С увеличением объема гравировальных работ возникает необходимость в применении более эффективных способов гравирования, свободных от указанных недостатков.

В настоящее время основным направлением автоматизации механической обработки является внедрение станков с числовым программным управлением (ЧПУ). Автоматизация гравирования путем перевода на фрезерные станки с ЧПУ существенно повышает производительность этого трудоемкого процесса.

Подготовка управляющих программ (УП) для станков с ЧПУ состоит из ряда этапов. Наиболее трудоемкими из

них являются этапы проектирования траектории движения режущего инструмента, расчета координат опорных точек траектории и кодирования числовой информации

Сложность подготовки УП для гравирования на станках с ЧПУ обусловлена следующими причинами:

- а) большой объем и разнообразие типов исходных знаков (буквы русского, латинского и других алфавитов, цифры, символы, шкалы и т. д.);
- б) высокая плотность расположения знаков на панели;
- в) сложность и многоконтурность структуры знаков;
- г) переменная толщина линий в знаке.

В работе [2] описана находящаяся в эксплуатации специальная установка, включающая гравировальный станок с программным управлением. Подготовка УП сводится здесь к перезаписи подпрограмм стандартных знаков с ленты-библиотеки на рабочую ленту. На выбор и перезапись одного знака затрачивается в среднем около двух минут. При большом количестве знаков этот способ подготовки УП требует существенных затрат времени.

С точки зрения повышения уровня автоматизации расчетов и ускорения процесса подготовки гравировальных программ более целесообразно осуществлять его с помощью универсальных ЭВМ. В настоящее время разработаны и применяются системы автоматизации выполнения некоторых этапов подготовки УП с помощью универсальных САПС-М22, СППС и других [3].

Попытки применить эти системы для подготовки гравировальных УП показали, что использование универсальных ЭВМ в данном случае не дает высокого эффекта вследствие непригодности САП для описания сложной исходной информации чертежа — форма записи исходных данных здесь сложна и неудобна.

Требуется разработка специфических методов кодирования информации и расчета гравировальных УП, имеющих возможность представления исходной информации чертежа в экономной и удобной для последующей обработки на ЭВМ форме. Такие методы разработаны в системе «Гравер».

Система «Гравер» состоит из двух подсистем: «Подготовка УП гравирования текстовых надписей» и «Подготовка УП гравирования символов».

Подготовка УП гравирования текстов и символов осуществляется по схеме, приведенной на рис. 1.



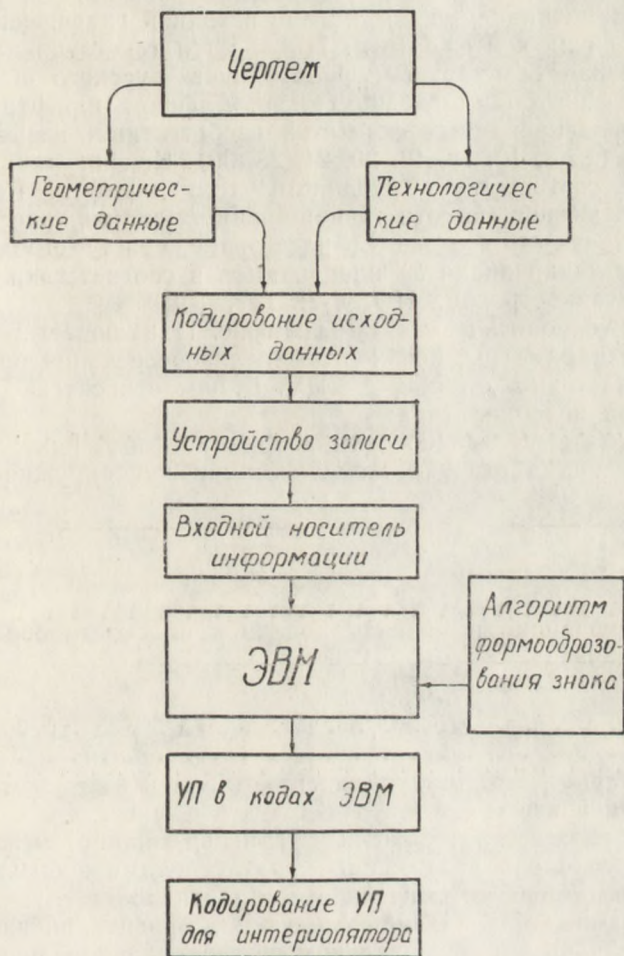


Рис. 1. Структурная схема системы подготовки УП гра-  
вирования текстов и символов

## I. Подготовка УП гравирования текстовых надписей

### 1. Характеристика подсистемы

В рассматриваемой подсистеме исходной графической информацией на чертеже (рис. 1) является текстовая надпись, выполненная стандартным рядом букв русского и латинского алфавитов и арабских цифр. Алфавит шрифта имеет восемь фиксированных форматов изображения, называемых кеглями (№ 8, 10, 12, 16, 20, 24, 28, 36). Конфигурация букв и цифр соответствует рубленому типу шрифта. Размеры шрифтов кеглей (высота  $H$  и ширина  $B$  знаков, расстояние между знаками в слове  $C$ , расстояние между строками  $f$ , толщина линии знака  $b$ ) назначаются в соответствии с нормалью станкостроения НОЗ-1-72 [4].

Блок технологических данных (рис. 1) включает технологические параметры, необходимые для обеспечения процесса гравирования на станках с ЧПУ. К ним относятся:

- а) код интерполятора;
- б) цена импульса по осям координат —  $h_{xyz}$  [мк];
- в) величина стрелки аппроксимации дуг окружностей —  $\delta$  [мк];
- г) ускоренная и рабочая подачи на оси  $Z$  —  $S_{z\text{ уск}}, S_{z\text{ раб.}}$   
 $\left| \frac{\text{мм}}{\text{мин.}} \right|$ ;

- д) ускоренная и рабочая подачи в плоскости обработки  $XOY$  —  $S_{\text{уск}}, S_{\text{раб.}}$   
 $\left| \frac{\text{мм}}{\text{мин.}} \right|$ ;

К данной подсистеме предъявляется ряд требований, определяющих ее функциональные возможности:

- а) задание исходной надписи «открытым» текстом (буквами и цифрами);
- б) возможность изменения размеров знаков, межзнаковых расстояний и расстояний между строками в большую и меньшую стороны относительно стандартных;
- в) возможность задания следующих ориентаций надписи: расположение в горизонтальном и вертикальном направлениях; симметричное расположение относительно осей координат  $X, Y$ ; поворот по квадрантам; расположение по дуге окружности;
- г) перемещение инструмента от буквы к букве на высоте 1 мм над плоскостью;



д) расчет необходимых перемещений инструмента от буквы к букве, от слова к слову, от строки к строке без возврата в исходное положение;

е) возможность дополнительного внесения знаков.

В соответствии с приведенными требованиями данная подсистема обеспечивает выполнение следующих функций:

1) автоматический ввод и преобразование исходных данных;

2) расчет координат опорных точек траектории обработки знаков в соответствии с типоразмерами знаков и ориентацией их на детали;

3) формирование выходной геометрической и технологической информации.

## 2. Кодирование исходной информации

Кодирование исходной информации состоит в задании геометрической и технологической информации, определяющей траекторию движения режущего инструмента и режимы резания при воспроизведении надписи на станках с ЧПУ.

Исходная геометрическая информация в данной подсистеме состоит из двух существенно различающихся частей. К первой части относится постоянная информация, которая повторяется при программировании обработки различных текстов. К постоянной информации относится исходный алфавит буквенно-цифровых знаков. Ко второй части относится переменная информация, различная для каждой исходной надписи. Она включает содержание надписи и параметры ориентации ее на панели.

В соответствии с разделением геометрической информации на две части разработаны специальные методы кодирования постоянной и переменной составляющих геометрической информации с целью независимого представления их в памяти ЭВМ.

### *А. Кодирование постоянной геометрической информации*

Буквам и цифрам исходного алфавита присвоены семи-разрядные двоичные коды системы символического кодирования ЭВМ «Минск-32» [5]. В соответствии с кодами знаков в памяти ЭВМ хранятся описания, определяющие конфигурацию и размеры знаков и последовательность обработки их контуров на станках с ЧПУ. Контур букв и цифр может быть выражен аналитически, графически или в виде таблиц. На основании анализа конфигураций знаков исходного на-

бора в данной подсистеме разработаны приемы формализованного описания геометрической информации, что позволило сократить объем и ограничить диапазон изменения вводимых в ЭВМ исходных данных.

Разнообразие исходных знаков ограничено сравнительно узким классом букв русского и латинского алфавитов и арабских цифр с некоторыми общими свойствами:

1. Буквы русского и латинского алфавитов имеют в своем составе ряд одинаковых по начертанию и размерам букв (*A, B, C, H, P* и др.), что сокращает общее количество знаков исходного алфавита.

2. Ряд буквенно-цифровых знаков алфавита входит в состав более сложных знаков. Так, например, буква *B* является частью букв *Ъ, Ъ*; из цифры *3* может быть получена цифра *8*, и т. д.

3. Некоторые геометрические элементы (прямые и дуги окружностей, а также их сочетания) многократно повторяются в знаках исходного алфавита.

Формализация буквенно-цифровой информации была произведена в следующей последовательности: выбор локальной системы координат; совмещенное вычерчивание контуров знаков в выбранной системе координат; анализ конфигураций знаков; выделение типовых геометрических элементов; синтез знаков из типовых элементов; назначение траектории обработки контура знака; составление описаний контуров знаков.

На диаграмме совмещенного представления буквенно-цифровых знаков (рис. 2), которая принята за исходную, типовыми элементами являются прямые и дуги окружностей, дающие в целом 87 номеров опорных точек. По данным диаграммы производится синтез знаков из типовых элементов. В соответствии с оптимальным маршрутом траектории инструмента составляются описания контуров знаков в виде последовательности номеров опорных точек с указанием типа линии между ними. В случае прямой между точками ставится знак «—», в случае дуги окружности — знак «*R*». Так, например, формализованное описание букв *A, G* и цифры *5* представлено в следующем виде: *A (1—2—3—79—4);*

*G (70—65—24R 25—31R 30—29R 28—8R 27);*

*3 (30R 31—25R 24—68R 67—66—5—7—8).*

Диаграмма типовых элементов выполнена для максимального кегля. Переход к другим типоразмерам шрифтов осу-



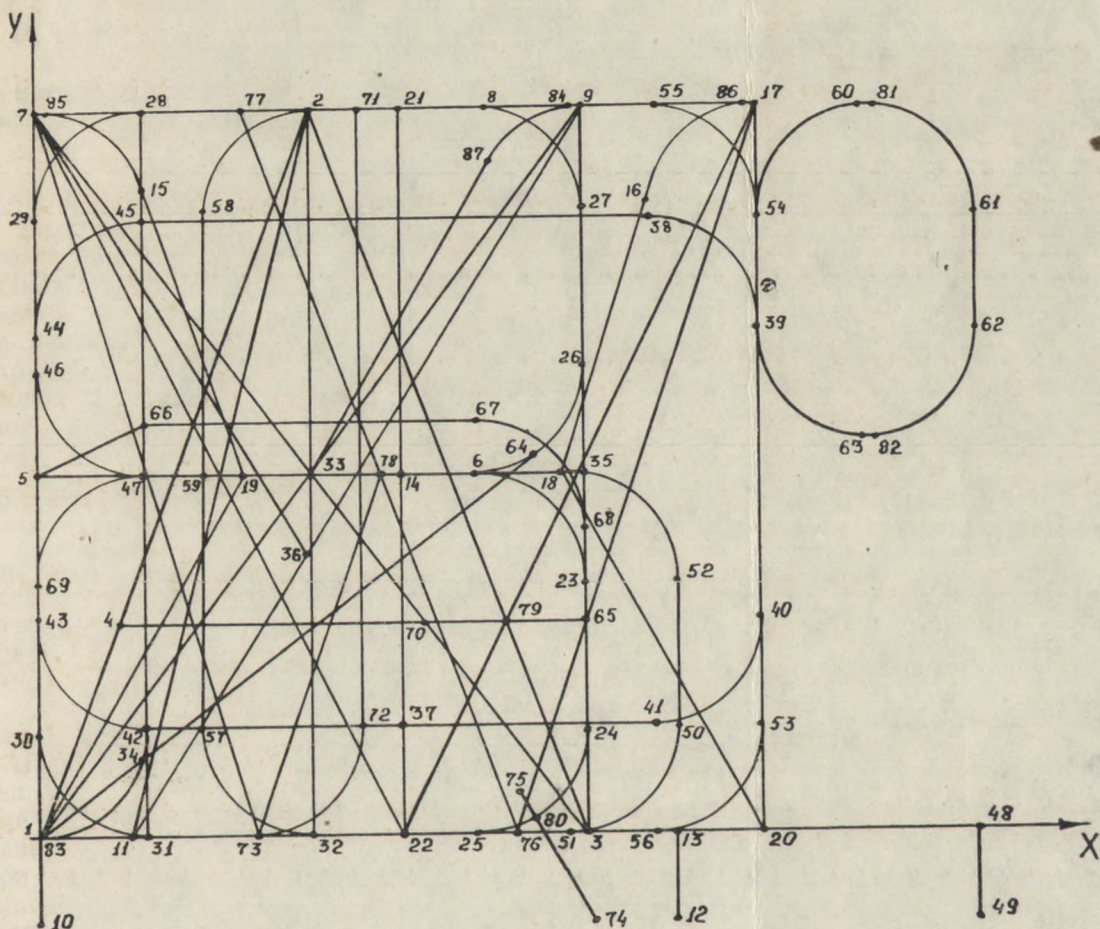
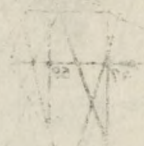


Рис. 2. Диаграмма типовых элементов контуров буквенно-цифровых знаков





ществляется с помощью масштабных коэффициентов по двум осям, координат  $K_x$  и  $K_y$ .

$$K_x = \frac{B - b}{B_i - b_i} \quad K_y = \frac{H - b}{H_i - b_i},$$

где  $B$  и  $H$  — ширина и высота знаков максимального кегля;

$B_i$  и  $H_i$  — ширина и высота знаков  $i$ -го кегля;

$b$  и  $b_i$  — толщина линий знаков соответствующих кеглей.

Расчет радиусов дуг окружностей производится по коэффициенту:

$$K = \frac{K_x + K_y}{2}.$$

Рассмотренный метод формализованного описания контуров буквенно-цифровых знаков предусматривает ввод новых знаков. Новый знак синтезируется из типовых элементов исходной диаграммы (рис. 2). При необходимости на диаграмме вводятся новые геометрические элементы: прямые, дуги окружностей и опорные точки. Описания контуров вновь вводимых знаков составляются по вышеприведенной методике, заносятся в массив исходных данных и вводятся в память ЭВМ.

### *Б. Кодирование переменной геометрической информации*

Содержание надписи и параметры ее ориентации изменяются от текста к тексту и вводятся в ЭВМ каждый раз заново в качестве исходных для расчета УП. В системах подготовки УП существуют различные методы кодирования, основными из которых являются: табличный метод и языковая запись.

Примером табличного метода кодирования может служить система СППС, примером языковой записи — система САП-2. Табличный метод достаточно универсален и экономичен, но требует соблюдения некоторых условий: формализации входных данных в виде таблиц; однозначности входных данных, т. е. отсутствия повторений одних и тех же строк при повторном проходе опорной точки. Системы языкового описания исходной информации более гибки в обращении (так как расчленение системы на части позволяет быстро компоновать нужную программу), но имеют, как правило, большое количество слов входного языка, что усложняет структуру транслятора и увеличивает объем вычислительных программ для ЭВМ.

В рассматриваемой подсистеме на основании использования типовых подпрограмм описания исходных данных разработан блочный способ задания входной информации, что существенно упрощает подготовку исходной информации и сокращает объем вводимых в ЭВМ исходных данных. Исходные данные — содержание надписи и параметры ее ориентации — записываются на специальном бланке (форма «Гравер 1») в соответствии с чертежом изделия. Форма «Гравер 1» выполнена в виде таблицы, в которой для каждой входной величины отведена отдельная графа. Такой вид таблицы обеспечивает наглядность записи информации и ее удобный контроль. Форма «Гравер 1» приведена в табл. 1.

### Правила записи исходных данных в форме «Гравер 1»

1. В первых двух графах указываются координаты  $x_n$ ,  $y_n$  начальной точки надписи относительно начала выбранной системы координат. За начальную точку принята нижняя левая точка прямоугольника, описывающего первую букву надписи.

2. В графе 3 записывается номер кегля  $N$ , определяющий типоразмер шрифта.

3. В графе 4 записывается значение масштабного коэффициента  $M$  в случае изменения стандартных размеров знаков в большую ( $M > 1$ ) или меньшую ( $M < 1$ ) стороны.

4. В графе 5 указывается расстояние между словами  $L$ .

5. В графах 6 и 7 записываются расстояние между буквами  $C$  и расстояние между строками  $f$ . Эти данные указываются только в том случае, если необходимо изменить стандартные значения  $C$  и  $f$ , определяемые номером кегля.

6. В графе 8 указывается признак симметричного расположения надписи относительно оси  $X$ . Координаты центра симметрии записываются в первых двух графах таблицы  $x_n$ ,  $y_n$ .

7. Симметричное расположение надписи относительно оси  $Y$  задается в графах 1, 2, 3 аналогично пункту 6.

8. В графе 10 указывается признак вертикального расположения надписи. Наличие признака соответствует вертикальному расположению надписи, отсутствие — горизонтальному расположению.

9. В графе 11 проставляется значение угла  $\alpha$  поворота надписи по квадрантам относительно положительного направления оси  $X$ .



Таблица 1

Форма «Гравер 1»

| Координаты<br>начальной<br>точки |            | Номер клетки N | Масштаб M | Расстояние между<br>словами L, мм |   | Специаль-<br>ные данные                |                                         | Симметричное рас-<br>полож. надписи<br>относит. оси X | Симметричное рас-<br>полож. надписи<br>относит. оси Y | Расположение над-<br>писи по вертикали | Поворот надписи<br>по квадратам $\alpha$ ,<br>град. | Расположение надписи<br>по дуге окружности |     |    |    | Содержание<br>надписи |
|----------------------------------|------------|----------------|-----------|-----------------------------------|---|----------------------------------------|-----------------------------------------|-------------------------------------------------------|-------------------------------------------------------|----------------------------------------|-----------------------------------------------------|--------------------------------------------|-----|----|----|-----------------------|
| $X_n$ , мм                       | $Y_n$ , мм | 3              | 4         | 5                                 | 6 | расстояние<br>между буква-<br>ми C, мм | расстояние<br>между стро-<br>ками f, мм | 8                                                     | 9                                                     | 10                                     | 11                                                  | 12                                         | 13  | 14 | 15 |                       |
| 100                              | 80         | 8              | —         | —                                 | — | —                                      | —                                       | —                                                     | —                                                     | —                                      | —                                                   | —                                          | —   | —  | —  | Станок**              |
| 100                              | 40         | 24             | —         | 3                                 | — | —                                      | —                                       | —                                                     | 1                                                     | —                                      | —                                                   | —                                          | —   | —  | —  | Деталь —<br>№ 123**   |
| —                                | —          | 20             | —         | —                                 | — | —                                      | —                                       | —                                                     | —                                                     | —                                      | —                                                   | 40                                         | 210 | 40 | 60 | Пуск***               |

10. Ориентация надписи по дуге окружности определяется параметрами дуги: радиус дуги  $R_g$ , начальный угол дуги  $\alpha_{нач}$ , координаты центра дуги  $x_c$ ,  $y_c$ . Эта информация заносится в графы 12, 13, 14, 15.

11. В графе 16 записывается содержание исходной надписи. Текст записывается в строку. Конец слова обозначается знаком —, конец строки исходного текста — знаком \*. Конец текста заданного типоразмера и заданной ориентации обозначается знаком \*\*.

Обязательному заполнению в каждой строке таблицы подлежат графы 1, 2, 3, 16 (за исключением ориентации по дуге окружности, где 1 и 2 графы не заполняются). Остальные графы заполняются выборочно, в зависимости от типоразмера шрифта и способа ориентации надписи. При отсутствии данных в какой-либо из граф в ней прочеркивается линия. Конец информации обозначается знаком \*\*\*, который записывается в конце таблицы. Примеры записи исходных данных приведены в табл. 1.

### *В. Кодирование технологической информации*

Технологические данные процесса гравирования остаются неизменными для обработки любых текстов. Они не зависят от содержания надписи и параметров ее ориентации. Технологические данные представлены в памяти ЭВМ самостоятельным числовым массивом и используются при формировании выходных данных.

## **II. Подготовка УП гравирования символов**

Исходной графической информацией в данной подсистеме являются условные знаки машиностроения, называемые символами согласно ОСТ-2-НО6-72. Каждый символ исходного набора представлен в локальной системе координат с указанием всех необходимых размеров, определяющих конфигурацию символа.

Анализ конфигураций и размеров символов исходного набора показал, что многие из них состоят из типовых геометрических элементов таких, как прямоугольники, стрелки, окружности и т. д. С учетом этого обстоятельства составлены формализованные описания символов с целью представления их в памяти ЭВМ.



В исходном наборе символов выделены типовые элементы (ТЭ). Типовым элементам присвоены коды, соответствующие их порядковым номерам в исходном наборе ТЭ. Например: ТЭ001, ТЭ025 и т. д. В соответствии с кодами в памяти ЭВМ хранятся массивы описаний ТЭ в виде совокупности координат опорных точек, определяющих траекторию обработки этих элементов на станках с ЧПУ.

Описание типового элемента начинается с кода ТЭ, далее следуют координаты  $X$  и  $Y$  опорных точек обработки контура ТЭ с указанием типа линии между ними. В случае дуги окружности дополнительно указывается направление обхода дуги и координаты центра.

В конце описания стоит признак конца типового элемента — константа КНЦТЭ.

Формализованное описание символов составлено по принципу синтеза из набора типовых элементов. Каждому символу присвоен код согласно его обозначению по ОСТ-2-Н06-72. В соответствии с присвоенными кодами в постоянном блоке памяти ЭВМ хранятся описания символов. Описания символов состоят из кодов ТЭ, входящих в состав символа, и координат начала локальной системы координат ТЭ. Описание геометрических элементов контуров символов, не вошедших в набор ТЭ, составляется аналогично описаниям ТЭ.

Таблица 2

| Код символа | Координата $X$ , мм | Координата $Y$ , мм | Признак |
|-------------|---------------------|---------------------|---------|
| 1           | 2                   | 3                   | 4       |
| 1.37        | 35.80               | 1.000               | —       |
| 1.1A        | 00000               | 10.00               | —       |
| 3.10        | 135.0               | 150.0               | КОНЕЦ   |
| КОНЕЦ       |                     |                     |         |

Исходные данные для расчета УП обработки символов записываются на специальном бланке — форме «Гравер 2». Форма «Гравер 2» выполнена в виде таблицы и состоит из четырех граф. В первой графе записывается код обрабатываемого символа. Во второй и третьей графах записываются координаты  $X$  и  $Y$  начала локальной системы координат символа. В четвертой графе записывается признак «КОНЕЦ»

в случае необходимости отвода инструмента в исходное положение по окончании обработки символа. Признаком конца исходных данных является слово «КОНЕЦ» в графе «Код символа». В табл. 2 приведены примеры записи исходных данных на бланке (форма «Гравер 2»).

### III. Формирование выходной информации

Результаты расчета УП в системе «Гравер» формируются в машинных кодах в виде двух массивов выходных данных — геометрического и технологического.

Для применения на имеющихся в производстве типах интерполяторов и станков с ЧПУ выходные данные с помощью специального программного блока преобразуются в форму, входа в систему СППС (форма № 1 и форма № 2 СППС), что позволяет использовать комплект постпроцессоров, разработанных для данной системы. Входная информация для системы СППС формируется в алфавитно-цифровых кодах ЭВМ «Минск-22». Результаты выводятся на перфоленту и на печать.

### Заключение

1. Разработанный метод формализации описаний буквенно-цифровых знаков и символов по способу синтеза из набора типовых элементов позволил существенно упростить их описания и сократить объем вводимой в ЭВМ исходной информации.

2. Блочный метод задания исходной информации в формах «Гравер 1» и «Гравер 2» повышает уровень автоматизации подготовки УП для гравирования по сравнению с подготовкой УП с помощью универсальных систем автоматизации программирования обработки деталей на станках с ЧПУ.

3. Приведенные испытания и внедрение системы «Гравер» при подготовке УП с помощью ЭВМ «Минск-32» для гравирования текстов и символов показали высокую эффективность системы и существенное снижение трудоемкости технологической подготовки производства гравировальных работ на станках с ЧПУ.

### ЛИТЕРАТУРА

1. Улановский О. О. Механизация граверных работ. Л., «Машиностроение», 1971.



2. Равва Ж. С. и Руманов Б. А. Программное управление процессом гравирования.— «Станки и инструмент», 1965, № 4.
  3. Никитенко В. Д. Подготовка программ для станков с программным управлением. М., «Машиностроение», 1973.
  4. Символы для таблиц и панелей. Отраслевой стандарт ОСТ2 НОЗ-5-72, М., 1972.
  5. Кушнерев Н. Т., Неменман М. Е., Цагельский В. И. Программирование для ЭВМ «Минск-32». «Статистика», 1973.
-

## **СТРУКТУРА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ «ГРАВЕР» ПОДГОТОВКИ ПРОГРАММ ДЛЯ СТАНКОВ С ЧИСЛОВЫМ ПРОГРАММНЫМ УПРАВЛЕНИЕМ**

**М. Н. БОГОЛЮБОВА, В. И. ЛИВШИЦ, А. К. МАРТЫНОВ**

В данной статье рассматривается структура математического обеспечения системы «Гравер», описанной в [1]. Математическое обеспечение системы включает комплекс алгоритмов и программ, разработанный для ЭВМ «Минск-32» и предназначенный для решения задач автоматизации подготовки и расчета управляющих программ (УП) гравирования букв, цифр и знаков на станках с числовым программным управлением (ЧПУ).

В соответствии со спецификой решаемых задач [1] математическое обеспечение системы «Гравер» разделено на две части — для подготовки управляющих программ гравирования текстов и символов.

### **1. Алгоритм процесса подготовки управляющих программ**

#### **§ 1. Постановка задачи**

Основная исходная геометрическая информация, необходимая для составления текстовых надписей, постоянно хранится в памяти ЭВМ. Эта информация включает следующие компоненты:

- 1) библиотеку формализованных описаний контуров буквенно-цифровых знаков исходного алфавита;
- 2) типоразмеры шрифтов соответствующих кеглей;
- 3) значения координат  $X$ ,  $Y$  опорных точек контуров знаков максимального кегля;
- 4) масштабные соотношения  $K_x$  и  $K_y$  между размерами знаков максимального и других типов кеглей.



Вторая составляющая часть исходной геометрической информации, представленная формой «Гравер 1» [1], является переменной. Она вводится в ЭВМ в качестве исходной для расчета соответствующей управляющей программы.

Технологические данные процесса гравирования остаются неизменными для обработки текстов всех типоразмеров и ориентаций. Они хранятся в памяти ЭВМ в виде самостоятельного массива и используются при формировании выходных данных.

Математическое обеспечение предназначено для выполнения следующих основных операций:

1. Ввод и преобразование исходной информации.
2. Формирование текста из составляющих знаков с учетом ориентации надписи:
  - а) горизонтальное расположение;
  - б) вертикальное расположение;
  - в) симметричное расположение относительно оси  $X$ ;
  - г) симметричное расположение относительно оси  $Y$ ;
  - д) поворот по квадрантам на угол  $\alpha$  относительно оси  $X$ ;
  - е) расположение по дуге окружности.
3. Формирование выходной информации.

## § 2. Блок-схема алгоритма

На основании анализа исходных данных и требований к математическому обеспечению, рассмотренных в § 1, разработана блок-схема алгоритма подсистемы подготовки УП для гравирования текстов (рис. 1). Алгоритм состоит из ряда частных алгоритмов-блоков, каждый из которых полностью решает определенную задачу.

### *1. Блок ввода и преобразования исходной информации*

Данный блок осуществляет ввод в память ЭВМ с внешних носителей исходных данных формы «Гравер 1» в виде числовых и текстовых величин. Исходные данные представлены двумя массивами: текстовая информация, определяющая содержание надписи, и числовая информация, определяющая типоразмер шрифта и параметры ориентации надписи. Числа переводятся в двоичную форму с плавающей запятой. Преобразованная информация пересылается в рабочую область памяти, где осуществляется ее анализ и расшифровка.

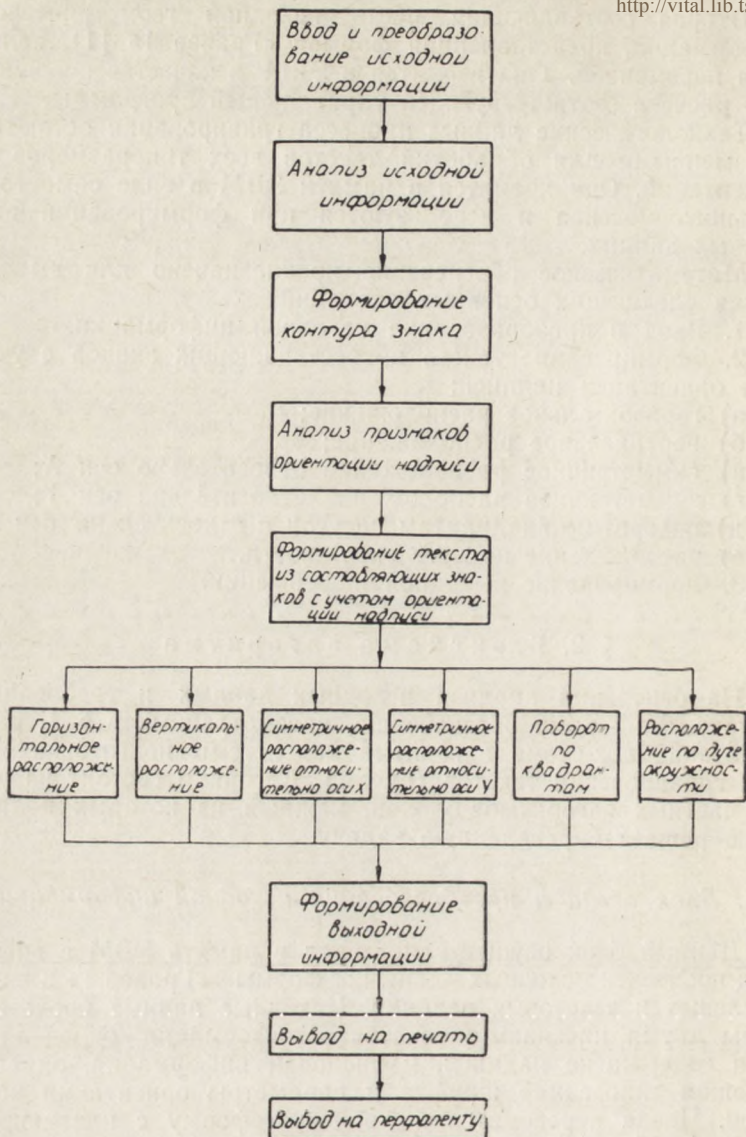


Рис. 1. Блок-схема алгоритма подготовки УП для гравирования текстов



## 2. Блок анализа исходной информации

Производится анализ исходных данных формы «Гравер 1». Координаты начальной точки  $X$ ,  $Y$  засылаются в ячейки текущих значений координат  $X_t$ ,  $Y_t$ . В соответствии с номером кегля в массиве постоянной геометрической информации отыскиваются все необходимые данные для формирования знаков заданного типоразмера. Анализируется значение масштабного коэффициента  $M$  и определяется вариант задания интервалов между буквами  $C$  и между строками  $f$ . Градусная мера измерения углов переводится в радианную.

## 3. Блок формирования контура знака

Осуществляется поиск описания очередного знака надписи в массиве формализованных описаний контуров букв и цифр. Структура каждого знака представлена в виде последовательности номеров опорных точек и ряда условных величин, определяющих начало и конец описания знака, наличие прерываний контура в знаке, обозначение радиусов округлений и т. д.

Координаты  $X$  и  $Y$  опорных точек контуров знаков представлены отдельным числовым массивом для максимального номера кегля. Координаты кегля  $i$  рассчитываются по следующим формулам:

$$X_i = x \cdot K_{x_i}; \quad (1)$$

$$Y_i = y \cdot K_{y_i}; \quad (2)$$

$$K_{x_i} = \frac{B - b}{B_i - b_i}; \quad (3)$$

$$K_{y_i} = \frac{H - b}{H_i - b_i}, \quad (4)$$

где  $B$  — ширина буквы максимального кегля;  
 $B_i$  — ширина соответствующей буквы для кегля  $i$ ;  
 $H$  — высота букв максимального кегля;  
 $H_i$  — высота букв кегля  $i$ ;

$b$  и  $b_i$  — толщина линий соответствующих кеглей [1].

При формировании дуги окружности радиус  $R_i$  рассчитывается по формуле:

$$R_i = K \cdot R; \quad (5)$$

$$K = \frac{1}{2} (K_{x_i} + K_{y_i}),$$

где  $R$  — значение радиуса скругления для букв максимального кегля.

Расчет координат центра  $X_{ц}$ ,  $Y_{ц}$ , начальной  $X_n$ ,  $Y_n$  и конечной  $X_k$ ,  $Y_k$  точек дуги производится, исходя из условия, что величина радиуса скругления любого знака для каждого кегля имеет одно и то же значение, а размер дуги составляет 90 градусов.

В случае масштабирования шрифта надписи в большую или меньшую стороны относительно заданного осуществляется расчет координат опорных точек знаков в соответствии со значением масштабного коэффициента  $M$ .

#### *4. Блок анализа признаков ориентации надписи*

Данный блок выявляет признаки и параметры ориентации надписи и осуществляет выбор и подготовку соответствующего блока ориентации.

#### *5. Блок формирования текста из составляющих знаков с учетом ориентации*

Алгоритм осуществляет формирование текста заданной ориентации с учетом расстояний между буквами  $S$ , между словами  $L$  и между строками  $f$ .

#### *6. Блок горизонтального расположения надписи*

При горизонтальном расположении надписи текущие значения координат  $X_n$ ,  $Y_n$  для  $n$ -й буквы вычисляются по следующим формулам:

$$X_n = X_n + \sum_0^{n-1} B_i + \sum_0^{n-1} C_i + L(\kappa - 1) + \sum_0^m x_j, \quad (7)$$

где  $X_n$  — начальная координата по оси  $X$ ;

$B_i$  — ширина  $i$ -й буквы;

$C_i$  — межбуквенное расстояние, определяемое конфигурацией двух соседних букв;

$K$  — количество слов в тексте;



$x_j$  — координата  $j$ -й опорной точки знака;  
 $m$  — количество опорных точек в  $n$ -й букве.

$$Y_n = Y_{\text{н}} + \sum_0^m y_j, \quad (8)$$

где  $Y_{\text{н}}$  — начальная координата по оси  $Y$ ;  
 $y_j$  — координата  $j$ -й опорной точки знака.

### 7. Блок вертикального расположения надписи

Данный блок определяет координаты  $X_n$ ,  $Y_n$  надписи, расположенной в вертикальном направлении.

$$X_n = X_{\text{н}} + \sum_0^m x_j; \quad (9)$$

$$Y_n = Y_{\text{н}} - H(n-1) - f(n-1) - L(\kappa-1) - \sum_0^m y_j, \quad (10)$$

где  $H$  — высота букв.

### 8. Блок симметричного расположения надписи относительно оси $X$

Координаты знаков при симметричном расположении надписи относительно оси  $X$  рассчитываются только для одной строки, расположенной в вертикальном направлении.

Координаты начальной точки надписи

$$X_{\text{н}} = X_{\text{ц}}; \quad (11)$$

$$Y_{\text{н}} = Y_{\text{ц}} + \frac{l}{2} - H,$$

где  $X_{\text{ц}}$ ,  $Y_{\text{ц}}$  — координаты центра симметрии;  
 $l$  — длина строки надписи.

$$l = H \cdot n + f(n-1) + L(\kappa-1), \quad (12)$$

где  $n$  — общее количество букв надписи.

### 9. Блок симметричного расположения надписи относительно оси $Y$

Координаты знаков рассчитываются для одной строки, расположенной в горизонтальном направлении.

Координаты начальной точки надписи

$$Y_n = Y_{ц}; \quad (13)$$

$$X_n = X_{ц} - \frac{l}{2}; \quad (14)$$

$$l = \sum_{i=1}^n B_i + \sum_{i=1}^{n-1} C_i + L(\kappa - 1). \quad (15)$$

#### 10. Блок поворота надписи по квадрантам

Данный блок формирует надпись под углом  $\alpha$  к оси  $X$  путем пересчета уже вычисленных значений координат опорных точек контуров знаков для случая горизонтального расположения.

Пересчет координат осуществляется по известным из аналитической геометрии формулам.

#### 11. Блок расположения надписи по дуге окружности

Исходными данными, определяющими ориентацию надписи по дуге окружности, являются: радиус дуги  $R_d$ , начальный угол дуги  $\alpha_{нач}$ , координаты центра дуги  $X_{ц}$ ,  $Y_{ц}$ .

Координаты начальной точки равны:

$$X_n = R_d \cdot \cos \alpha_{нач} + X_{ц}; \quad (16)$$

$$Y_n = R_d \cdot \sin \alpha_{нач} + Y_{ц}. \quad (17)$$

Угол поворота локальной системы координат первой буквы надписи относительно общей системы координат

$$\alpha = \alpha_{нач} - \frac{\pi}{2} - \frac{B}{2}, \quad (18)$$

где  $\beta$  — угол, определяемый шириной первой буквы надписи  $B$ .

$$\beta = \arcsin \frac{B}{R_d}. \quad (19)$$

Угол поворота локальной системы координат последующих букв

$$\alpha_i = \alpha_{i-1} - \frac{1}{2} \beta_{i-1} - \gamma_{i-1} - \frac{1}{2} \beta_i - \varphi(\kappa - 1), \quad (20)$$



где  $i$  — порядковый номер буквы;

$\alpha_{i-1}$  — угол поворота локальной системы координат  $i-1$  буквы;

$\beta_{i-1}$  — угол, определяемый длиной хорды, равной ширине  $i-1$  буквы;

$\gamma_{i-1}$  — угол, определяемый длиной хорды, равной межбуквенному расстоянию  $C$  между  $i$  и  $i-1$  буквами;

$\varphi$  — угол, определяемый расстоянием между словами  $L$ .

$$\gamma = \arcsin \frac{C}{R_d}; \quad (21)$$

$$\varphi = \arcsin \frac{L}{R_d}. \quad (22)$$

Расчет координат точек контуров знаков, расположенных по дуге окружности и повернутых на соответствующий угол, осуществляется блоком поворота.

## 12. Блок формирования выходной информации

Выходная информация формируется в виде двух массивов информации — геометрических и технологических данных соответственно. Геометрические данные включают порядковые номера опорных точек траектории движения инструмента, тип линии между соседними опорными точками и значения координат  $X$ ,  $Y$  этих точек. Технологические данные включают общие технологические параметры, перечисленные в [1], и номера тех опорных точек, в которых происходит изменение технологических данных, с указанием значения изменяемого параметра.

С целью обращения к постпроцессорам, разработанным для системы СППС, выходные данные системы «Гравер» преобразуются в форму входа в систему СППС. Формируются два массива входных данных — форма № 1, описывающая геометрию буквенно-цифровых знаков, и форма № 2, описывающая технологию процесса гравирования тех же знаков на станках с ЧПУ. Входные данные для системы СППС формируются в кодах ЭВМ «Минск-22». Преобразование информации, представленной в кодах ЭВМ «Минск-32», в коды «Минск-22» осуществляется с помощью специального программного блока согласно таблице соответствий кодов ЭВМ.

Сформированные массивы геометрической и технологической информации выводятся на печать и на перфоленду.

### § 3. Структурная схема программы

На основании логических и аналитических зависимостей, рассмотренных в § 2, составлена структурная схема программы. Ниже приводятся блок-схемы программ отдельных блоков, раскрывающие их содержание и непосредственную организацию программы.

а) Блок-схема программы формирования текста с учетом межзнаковых расстояний  $C, a, f$  (рис. 2).

Обозначения в программе:

\* (звездочка) — признак конца строки;

— (пробел) — признак интервала между словами;

РАСПН, КОНСБ, РДУГИ — стандартные ячейки, в которых соответственно хранятся: признак расположения надписи в горизонтальном или вертикальном направлении, значение межбуквенного расстояния, радиус дуги при задании ориентации по дуге окружности;

Л5 — 5, П9 + 2, Л6 — 5, ДУГА1, ВЫБКВ + 1 — адреса команд, которым передается управление программы.

б) Блок-схема программы расположения надписи по дуге окружности (рис. 3).

Обозначения в программе:

БЕТА, АЛЬФА, ГАММА, ХТЕК, УТЕК — адреса ячеек, в которых хранятся следующие данные: углы  $\alpha, \beta, \gamma$ , текущие значения координат  $X_T, Y_T$ , КОНСИ — адрес оператора программы.

## II. Алгоритм процесса подготовки управляющих программ гравирования символов

### § 1. Постановка задачи

Основная геометрическая информация, определяющая конфигурацию и размеры символов исходного набора, представлена в памяти ЭВМ в виде двух числовых массивов: массива формализованных описаний контуров символов и массива описаний типовых элементов (ТЭ) [1]. Эта информация является постоянной и используется при расчете управляющих программ обработки соответствующих символов на станках с ЧПУ.

Вторая составляющая часть исходной геометрической информации, представленная формой «Гравер 2» [1], включает



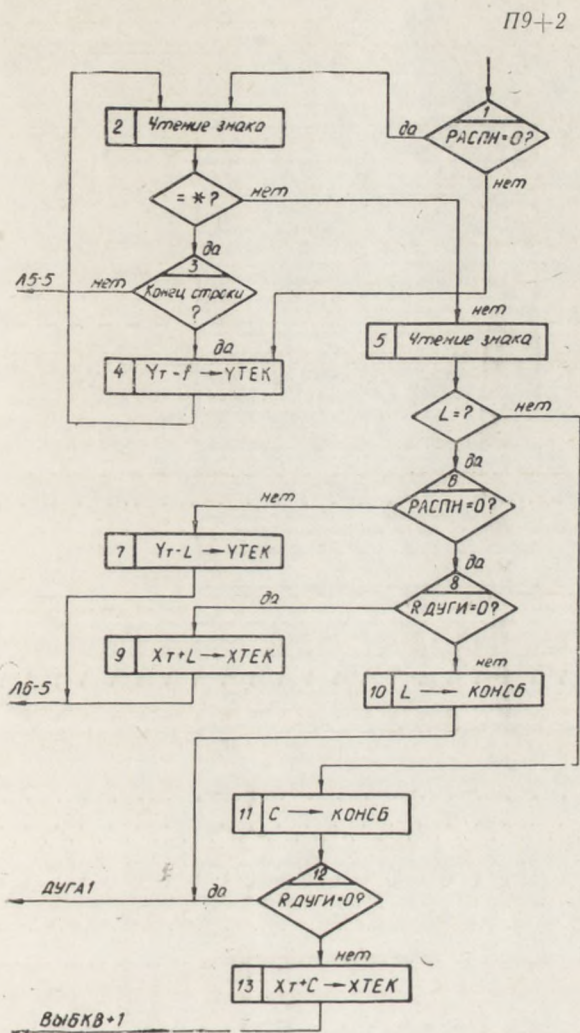


Рис. 2. Блок-схема программы формирования текста с учетом межзнаковых расстояний

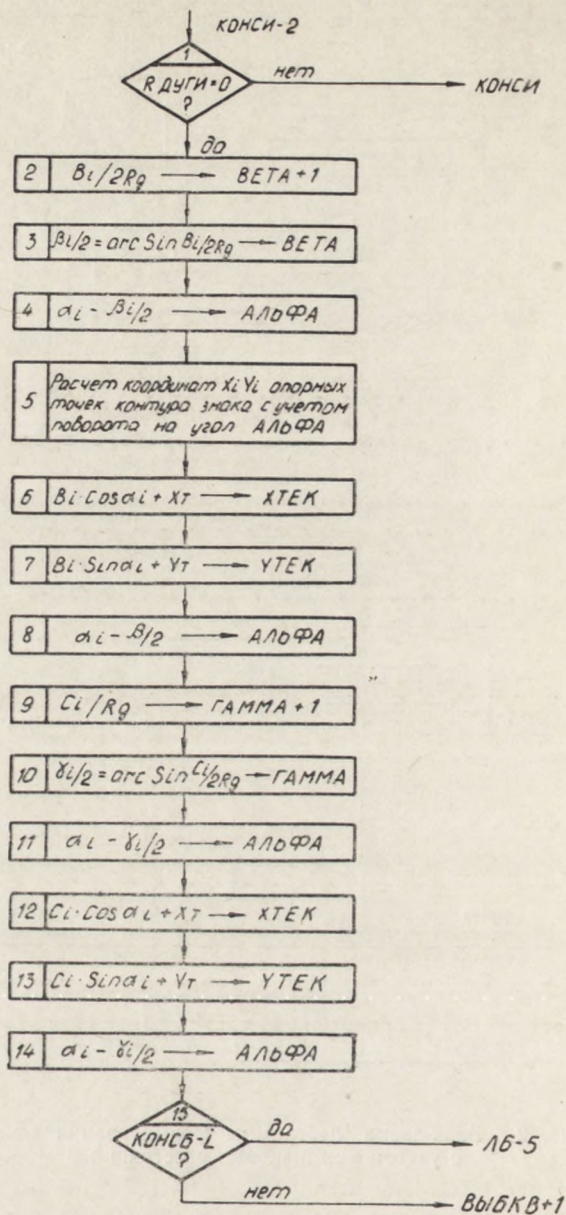


Рис. 3. Блок-схема программы расположения надписи по дуге окружности



коды обрабатываемых символов и координаты  $X$  и  $Y$  расположения символов в системе координат детали. Эта информация является переменной и вводится в ЭВМ каждый раз заново в качестве исходной для расчета соответствующих управляющих программ.

Технологические данные, необходимые для обеспечения процесса гравирования символов на станках с ЧПУ, аналогичны данным процесса гравирования текстов и хранятся в памяти ЭВМ самостоятельным массивом.

Математическое обеспечение данной подсистемы предназначено для выполнения следующих основных операций:

- а) ввод и преобразование исходных данных;
- б) формирование контура символа из набора типовых элементов;
- в) формирование выходной информации.

## § 2. Блок-схема алгоритма

На основании анализа исходных данных и требований к математическому обеспечению разработана блок-схема алгоритма подготовки УП для гравирования символов (рис. 4).

### *1. Блок ввода и преобразования исходной информации*

Данный блок предназначен для ввода в ЭВМ массивов описания символов и типовых элементов, а также для ввода исходных данных формы «Гравер 2» [1]. Ввод информации возможен как с перфокарт, так и с магнитной ленты. В остальном он выполняет операции, аналогичные блоку ввода и преобразования исходной информации алгоритма, рассмотренного в § 2 главы 1.

### *2. Блок поиска описания символа в массиве описаний символов*

Данный блок осуществляет поиск описания символа в соответствии с кодом, заданным в таблице исходных данных формы «Гравер 2». В том случае, если код задан неверно, выдается сообщение на пульт машины: «Неверно задан код символа».

### *3. Блок анализа описания символа*

Анализ описания символа осуществляется с целью выявления кодов типовых элементов, входящих в состав сим-

вола, подъемов и опусканий по оси  $Z$ , движений инструмента по дуге окружности. При наличии в описании символа участков скруглений по дуге окружности при координатах центра дуги формируются коды направления обхода дуги: 311 — по часовой стрелке и 322 — против часовой стрелки.

#### *4. Блок поиска описания типового элемента в массиве описаний типовых элементов*

Поиск описания ТЭ в массиве описания ТЭ осуществляется последовательно для всех ТЭ в соответствии с их кодами и порядком расположения в описании символа. В данном блоке предусматривается вывод сообщения: «Нет элемента ТЭ» на пульт оператора в случае отсутствия какого-либо из типовых элементов при поиске его в массиве описания ТЭ.

#### *5. Блок анализа описания типового элемента*

Анализ описания ТЭ производится с целью выявления признака поворота ТЭ, входящего в состав обрабатываемого символа, а также выявления признаков движения инструмента по дуге окружности.

Признак поворота ТЭ на угол  $\alpha$  относительно исходного положения его локальных координат определяется конфигурацией символа и указывается перед кодом ТЭ в описании символа. При наличии признака поворота ТЭ вычисляются значения  $\sin \alpha$  и  $\cos \alpha$ .

#### *6. Расчет координат опорных точек символа в системе координат детали*

Данный блок осуществляет расчет координат опорных точек типовых элементов  $X_{ттэ}$ ,  $Y_{ттэ}$  относительно системы координат символа  $X_6Y_6$  с последующим приведением координат опорных точек символа к системе координат детали. Расчеты производятся с учетом операций поворота и параллельного переноса ТЭ относительно систем координат символа и детали.

Координаты опорных точек символа  $X$ ,  $Y$  вычисляются по следующим формулам:

$$X' = X_{т.тэ} \cos \alpha + Y_{т.тэ} \sin \alpha; \quad (23)$$

$$Y' = X_{т.тэ} \sin \alpha + Y_{т.тэ} \cos \alpha; \quad (24)$$



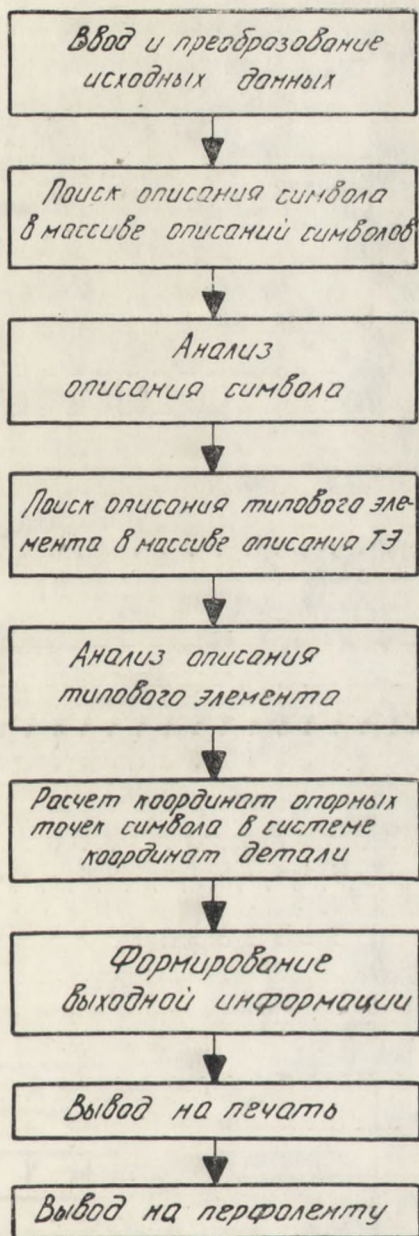


Рис. 4. Блок-схема алгоритма подготовки УП для гравирования символов





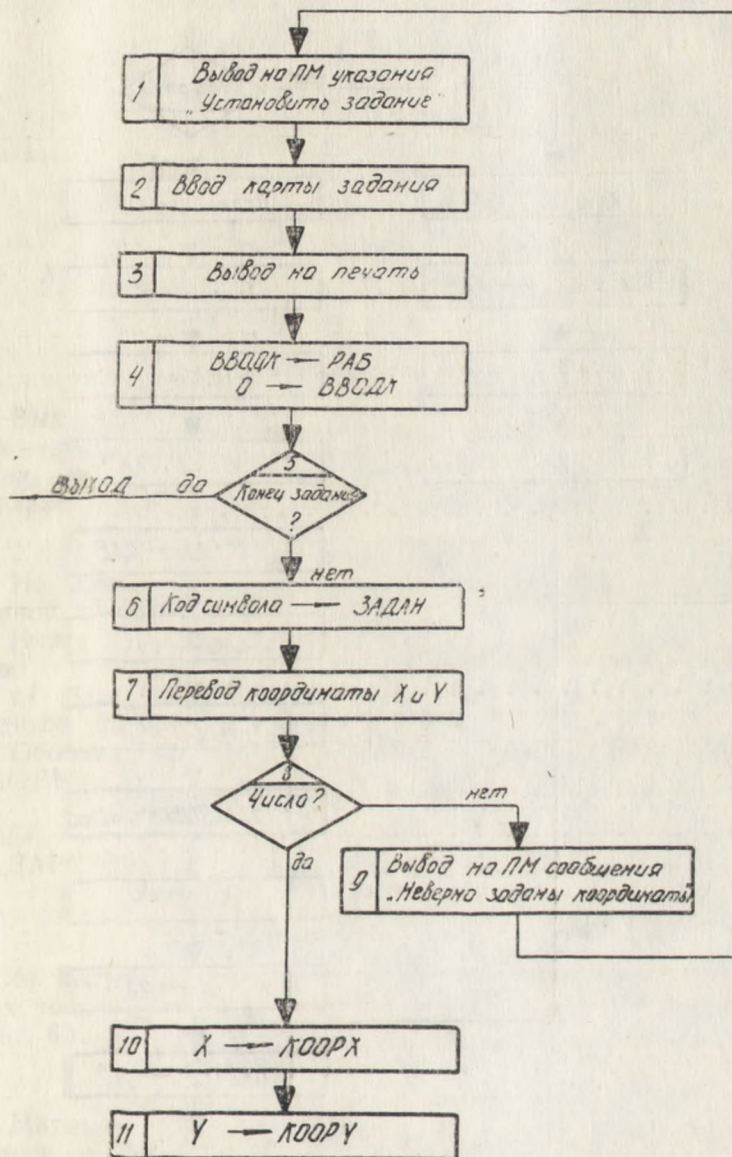


Рис. 15. Блок-схема программы ввода и преобразования исходных данных задания

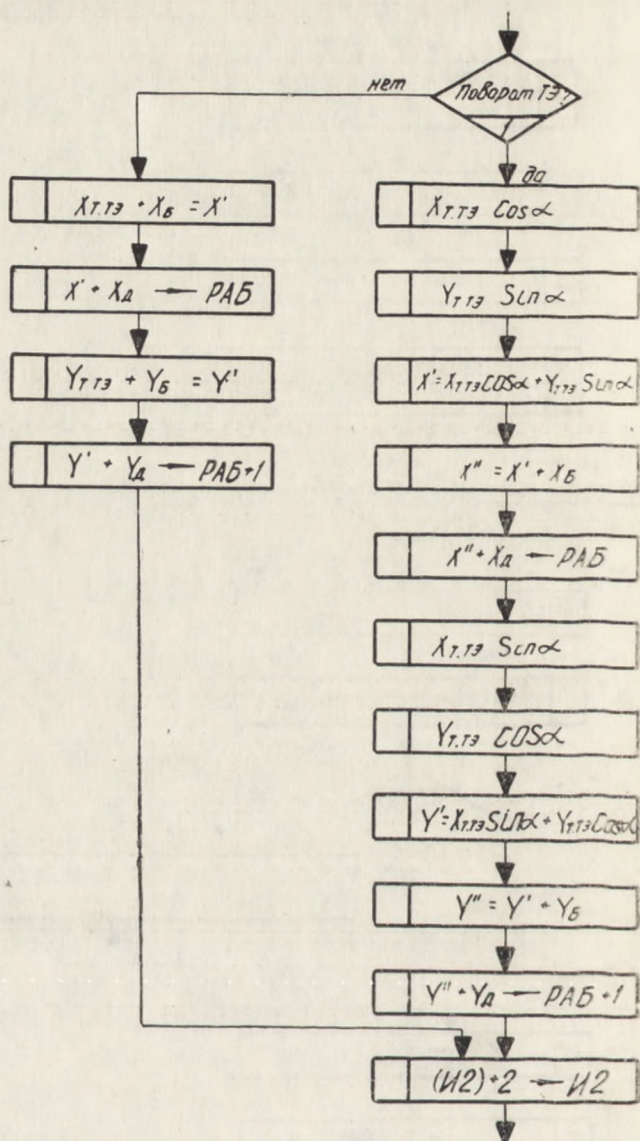


Рис. 6. Блок-схема программы приведения координат опорных точек типового элемента к системе координат детали



$$X'' = X' + X_6; \quad (25)$$

$$Y'' = Y' + Y_6; \quad (26)$$

$$X = X'' + X_d; \quad (27)$$

$$Y = Y'' + Y_d; \quad (28)$$

при  $\alpha = 0$

$$X = X_{т.тэ} + X_6 + X_d; \quad (29)$$

$$Y = Y_{т.тэ} + Y_6 + Y_d, \quad (30)$$

где  $X_d, Y_d$  — координаты начала локальной системы координат символа в системе координат детали.

### 7. Блок формирования выходной информации

Выходная информация формируется в виде двух массивов — геометрических и технологических данных — аналогично формированию выходной информации, рассмотренной в главе 1.

#### § 3. Структурная схема программы

На основе алгоритмических приемов, рассмотренных в § 2 данной главы, разработана структурная схема программы.

Ниже приводятся блок-схемы программ отдельных блоков.

а) Блок-схема программы ввода и преобразования исходных данных задания (рис. 5).

Обозначения в программе:

ВВОДК — адрес ячейки, в которой располагается признак отвода инструмента в исходное положение;

РАБ — рабочая ячейка памяти;

ЗАДАН, КООРХ, КООРУ — стандартные ячейки, в которых хранятся соответственно код символа, координата  $X$  и координата  $Y$ .

б) Блок-схема программы приведения координат опорных точек типового элемента к системе координат детали (рис. 6).

### III. Заключение

Математическое обеспечение системы «Гравер» разработано и отлажено на языке символического кодирования ЯСК ЭВМ «Минск-32» и занимает объем памяти около 16000 яче-

ек. Время подготовки и расчета управляющей информации для одного знака составляет в среднем 10—20 сек для простых и сложных знаков соответственно.

Разработанная система автоматизации подготовки управляющих программ для выполнения гравировальных работ на фрезерных станках с ЧПУ позволяет решать с высокой экономической эффективностью сложные задачи, возникающие при изготовлении панелей управления, приборных щитов и плат, фирменных табличек и других аналогичных деталей современных машин, приборов и аппаратов. До разработки системы «Гравер» перевод гравировальных работ на фрезерные станки с ЧПУ был фактически неосуществим ввиду чрезвычайной трудности безмашинной подготовки и расчета управляющих программ для этой операции. Опробование системы «Гравер» в производственных условиях показало ее высокое быстродействие и эффективность при подготовке весьма сложных управляющих программ.

#### ЛИТЕРАТУРА

1. Боголюбова М. Н., Лившиц В. И., Мартынов А. К. Автоматизация подготовки управляющих программ для выполнения гравировальных работ на фрезерных станках с ЧПУ. Настоящий сборник.

---



## ОБ ОДНОЙ РЕАЛИЗАЦИИ АЛГОРИТМА ОПРЕДЕЛЕНИЯ ВРЕМЕННЫХ ПАРАМЕТРОВ СЕТЕВЫХ ГРАФИКОВ

А. А. ТКАЧЕНКО

В настоящее время методы сетевого планирования находят широкое применение в составе автоматизированных систем управления предприятиями, а алгоритмы и их программы определения временных параметров сетевых графиков являются составной частью математического обеспечения соответствующих АСУП.

В ряде опубликованных работ [1, 2] излагаются алгоритмы и их программные реализации для расчетов временных параметров сетевых графиков и указывается, что они рассчитаны на обработку графиков, имеющих в своем составе до  $2000 \div 2500$  событий. Это объясняется тем, что предложенные реализации строятся на одно- или двухячеячных машинных документах и основаны на использовании лишь оперативной памяти ЭВМ.

Предлагается алгоритм определения временных параметров сетевых графиков, информация о которых значительно превосходит объем оперативной памяти ЭВМ, использующий возможность разбиения графа работ на слои [3].

Для наглядности порядок вычисления параметров иллюстрируется на примере, изображенном на рис. 1.

Пусть задан детерминированный многоисходный односторонний граф работ, моделирующий технологическую последовательность изготовления какого-либо объекта. Исходная информация о графе работ (рис. 1) в форме машинных документов упорядочена по слоям [3] и расположена на МЛ (магнитной ленте).

На рис. 1 введены обозначения:

$a$  — завершающая операция по изготовлению объекта;

$b, c, d, \dots$  — операция (изготовление узла, детали, проектирование технологии; изготовление приспособления и т. п.);

I, II, III, ... — номера слоев графа работ.

Далее используются обозначения:

$t$  — время выполнения операции;

$T_p^n$  — раннее начало выполнения операции;

$T_n^o$  — позднее окончание выполнения операции;

$R$  — полный резерв времени выполнения операции.

Исходный массив просматривается в прямом направлении от завершающей операции ( $a$ ) к начальным ( $p, r, s$ ) и по-

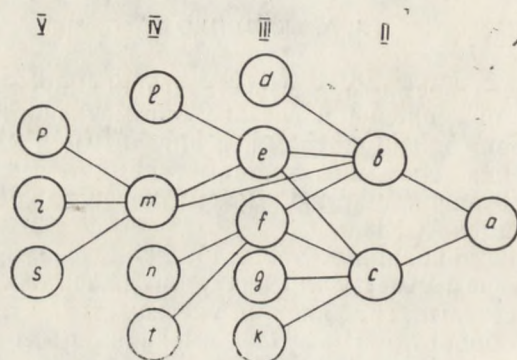


Рис. 1

следовательно разбивается на подмассивы (рис. 1). Каждый слой графа работ I, II, ... становится подмассивом. Составляются таблицы описания подмассивов. Дополнительно каждой операции присваивается  $T_p^n = 0$ ,  $T_n^o = \max$ .

Определение временных параметров начинается с нахождения  $T_p^n$ .

Так как для операций слоя V все  $T_p^n \equiv 0$  определены во время разбиения массива, то вычисления начинаются сразу со слоя IV.

Для определения  $T_p^n$  операций слоя IV (рис. 1) необходимо найти только  $T_p^n(m)$ , так как для операций  $l, n, t$   $T_p^n \equiv 0$ . Время выполнения операций  $p, r, s$  в общем случае различно, поэтому

$$T_p^n(m) = \max \{T_p^n(p) + t(p); T_p^n(r) + t(r); T_p^n(s) + t(s)\}.$$



Далее  $T_p^{\kappa}$  операций слоя III:

$$T_p^{\kappa}(d) = T_p^{\kappa}(g) = T_p^{\kappa}(\kappa) = 0;$$

$$T_p^{\kappa}(e) = \max \{T_p^{\kappa}(l) + t(l); T_p^{\kappa}(m) + t(m)\};$$

$$T_p^{\kappa}(f) = \max \{T_p^{\kappa}(t) + t(t); T_p^{\kappa}(m) + t(m); T_p^{\kappa}(n) + t(n)\}$$

так далее.

После определения  $T_p^{\kappa}(a)$  конечный срок окончания всех работ  $T$

$$T = T_p^{\kappa}(a) + t(a).$$

Следующий этап — определение  $T_n^o$  операций. Из рис. 1 видно, что для операций  $b$  и  $c$

$$T_n^o(b) = T_n^o(c) = T_p^{\kappa}(a) = T_n^{\kappa}(a).$$

Здесь  $T_n^{\kappa}(a)$  — позднее начало выполнения операции  $a$ .

Для операций, лежащих на слое III,  $T_n^o$  будет соответственно:

$$T_n^o(d) = T - [t(a) + t(b)] = T_n^{\kappa}(a) - t(b) = T_n^{\kappa}(b);$$

$$T_n^o(e) = T - \max \{[t(a) + t(b)]; [t(a) + t(c)]\} =$$

$$= T_n^{\kappa}(a) - \max \{t(b); t(c)\} = \min \{T_n^{\kappa}(b); T_n^{\kappa}(c)\}$$

и так далее.

Для определения  $T_n^o$  операций слоя III используются данные о  $T_n^o$  слоя II, а для слоя IV — данные предшествующего слоя III и т. д.

Одновременно с определением  $T_n^o$  вычисляются и полные резервы времени:

$$R = T_n^o - T_p^{\kappa} - t.$$

Таким образом, определение  $T_p^{\kappa}$ ,  $T_n^o$ ,  $R$  операций сетевого графика распадается на ряд последовательных шагов. При этом каждый следующий шаг использует информацию лишь о предшествующем. Это позволяет существенно уменьшить ограничения на размеры обрабатываемых сетевых графиков, накладываемые оперативной памятью ЭВМ. В самом деле, в рассмотренной процедуре ограничением является лишь размер слоя, который всегда заведомо меньше размеров всего графика.

Из анализа процедуры видно, что определение временных параметров сетевого графика осуществляется всего за пять просмотров исходного массива информации:

- разбиение на подмассивы, присвоение  $T_p^n = 0$ ,  
 $T_n^o = \max$ ;
- определение  $T_p^n$ ;
- определение  $T_n^o$ ;
- два дополнительных просмотра для формирования информации о предшествующих шагах при определении  $T_p^n$  и  $T_n^o$ .

Малое число обращений к внешним устройствам позволяет получить большое быстродействие программ, реализующих алгоритм. Поэтому, во-первых, сетевые графики могут содержать кроме обычных (временных) и другие важные при планировании параметры — исполнители, количество и т. д.; во-вторых, нет необходимости сетевые графики больших размеров обрабатывать по частям; в-третьих, кодирование вершин графа (сети) можно осуществлять шифрами, имеющими содержательный смысл, что очень удобно для составителей сетей.

Программа расчета временных параметров сетевых графиков по описанной процедуре была составлена для ЭВМ «Минск-22» и является составной частью математического обеспечения автоматизированной подсистемы планирования и управления ходом подготовки и освоения новых заказов. Программа работает в двух ключах: в первом — с небольшими сетевыми графиками — ограничение на слой 1023 операции; во втором — с большими сетями — ограничение на слой 2046 операций. В первом ключе быстродействие примерно в 1,5 раз выше, чем во втором.

За время эксплуатации получены следующие оценки:

- граф работ содержит  $\approx 1500$  операций — время счета  $\leq 20$  мин,
- граф работ содержит  $\approx 6000$  операций — время счета  $\leq 2$  час.

#### ЛИТЕРАТУРА

1. Библиотека программ анализа сетевых графиков. Новосибирск, 1970.
2. Козлов О. М., Лагутина З. П., Филиппович Е. И. Расчет временных характеристик сетевых графиков. — В сб.: Разработка и внедрение систем сетевого планирования и управления в промышленности УССР. 1969.
3. Кофман А., Дебазей Г. Сетевые методы планирования и их применение. 1968.



## МОДУЛЬНАЯ СТРУКТУРА ВЫЧИСЛИТЕЛЬНОГО АЛГОРИТМА В ЗАДАЧЕ ХИЛЛА — БРАУНА

Л. А. МОСКОВКИНА

Современный уровень развития вычислительной техники и средств математического обеспечения позволяет решать на вычислительных машинах большие задачи небесной механики со сложными алгоритмами. В частности, решаются аналитические задачи небесной механики с использованием для этих целей проблемно-ориентированных систем программирования или систем программирования аналитических выкладок общего назначения.

Разработка программ таких задач, как правило, является сложным и длительным процессом. Уже во время составления и отладки, а также в процессе эксплуатации подобные программы неизбежно подвергаются изменениям и переделкам. Для уменьшения этих трудностей при реализации больших программ целесообразно использовать модульный принцип, обоснованный, например, в [1].

Под модульным программированием понимается способ составления программ, при котором каждая программа рассматривается состоящей из нескольких частей (модулей) и допускается составление, трансляция и отладка каждого такого модуля независимо от других. Такой подход к построению сложных программ позволяет оперативно вносить поправки, связанные с усовершенствованием модулей, или полностью заменять соответствующие модули. Поскольку каждый модуль выполняет строго определенную функцию, достигается организационная стройность программы, облегчается ее отладка и эксплуатация.

Кроме того, модульный принцип обеспечивает существенную экономию МОЗУ за счет того, что в каждый момент

времени в оперативной памяти содержится незначительное число модулей, что немаловажно при реализации задач небесной механики, оперирующих обычно выражениями большого объема.

При построении отдельных модулей должна учитываться возможность их использования для решения других задач небесной механики. Вообще более целесообразным является подход, при котором строится система модулей, как базис для решения целого класса задач. Разработка базиса требует, как показано в [1], соответствующего анализа вычислительных алгоритмов, применяемых в данной области.

Проведем анализ для решения ограниченной задачи трех тел методом, известным в небесной механике под названием метода Хилла—Брауна. Постановка задачи подробно изложена в ряде работ, например [2] — [4]. Следуя указанным авторам, будем использовать уравнения движения возмущаемого тела в комплексных переменных Хилла  $u$ ,  $s$ ,  $z$  в виде:

$$\begin{aligned}(D + m)^2 u + \frac{1}{2} m^2 u + \frac{3}{2} m^2 s - \frac{\kappa u}{(us + z^2)^{3/2}} &= -\frac{\partial \Omega}{\partial s}, \\(D - m)^2 s + \frac{1}{2} m^2 s + \frac{3}{2} m^2 u - \frac{\kappa s}{(us + z^2)^{3/2}} &= -\frac{\partial \Omega}{\partial u}, \\(D^2 - m^2) z - \frac{\kappa z}{(us + z^2)^{3/2}} &= -\frac{1}{2} \frac{\partial \Omega}{\partial z}.\end{aligned}\quad (1)$$

Решение  $uT^{-1}$ ,  $sT$ ,  $z\sqrt{-1}$  системы (1) ищется в виде

$$a \sum_p \alpha^{p_1} e^{p_2} e'^{p_3} \kappa^{p_4} \sum_j N_{pj} T^{j_1+j_2+j_3g+j_4m}. \quad (2)$$

Здесь  $N_{pj}$  — неизвестные числовые коэффициенты, которые определяются в процессе решения;  $a$ ,  $\alpha$ ,  $e$ ,  $e'$ ,  $\kappa$ ,  $m$ ,  $c$ ,  $g$ ,  $\kappa$ ,  $n$ ,  $n'$  — известные величины, характеризующие орбиты возмущающего и возмущаемого тел;  $T = \exp(n - n')(t - t_0)\sqrt{-1}$  — независимая переменная Хилла;  $D = T \frac{d}{dT}$ ;  $p_i$ ,  $j_k$  — целые числа, границы изменения которых определяются желаемой точностью в координатах для конкретного объекта.

Известно выражение для  $\Omega$

$$\Omega = \sum_i \frac{m^2}{(a')^i} \sum_{jkl} f_{jkl} u^{j_1} s^{j_2} z^{j_3} e'^{l} T^{\kappa m}.$$



В задаче Хилла—Брауна используются следующие основные принципы. Последовательное определение неравенств нулевого, первого, второго порядков и т. д.; использование неопределенных коэффициентов и раздельное определение неравенств для каждой характеристики  $p$  порядка  $n$ . При этом под характеристикой мы понимаем выражение  $\alpha^{p1}, l^{p2}, l'^{p3}, \kappa^{p4}$  в формулах (2), а под порядком неравенств величину  $4 p_i$ .

Коэффициенты неравенств нулевого решения находятся особым образом. Для определения же неравенства порядка  $n$  (для  $n \geq 1$ ) с характеристикой  $p$  (при этом все коэффициенты неравенств до  $n-1$  порядка включительно уже известны) в уравнения движения (1) подставляются суммы комплексов неравенств с характеристиками, пробегающими все возможные для данной координаты значения. В правых и левых частях уравнений приравниваются члены, имеющие множитель  $p$ . Это приводит к одному из двух соотношений:

$$T^{-1}(D+m)^2 u_p + M u_p T^{-1} + M' s_p T = ap \sum_j A_{pj} T^{j_1+j_2c+j_3g+j_4m};$$

$$D^2 z_p \sqrt{-1} - 2Mz_p \sqrt{-1} = ap \sum_j A_{pj} T^{j_1+j_2c+j_3g+j_4m}. \quad (3)$$

Правые части этих уравнений состоят из известных величин, определенных на предыдущих этапах. Часть членов с характеристикой  $p$  порождена величинами  $\frac{\partial \Omega}{\partial s}$  и  $\frac{\partial \Omega}{\partial z}$ ; дру-

гая часть порождена разложениями в ряд выражений  $\frac{u}{r^3}$  и

$\frac{z}{r^3}$ , где  $r^2 = us + z^2$ ; остальные члены обусловлены наличием в (1) оператора  $D$ . Величины  $M$  и  $M'$  также являются известными.

Подстановка в соответствующее уравнение из (3) части искомого решения (с характеристикой  $p$  и фиксированным набором индексов  $j_k$ ;  $k=2, 3, 4$ ) в виде

$$ap \sum_{j_1} [N_{pj_k j_1} T^{j_1+j_2c+j_3g+j_4m} + N_{p-j_k j_1} T^{j_1-j_2c-j_3g-j_4m}]$$

с неопределенным коэффициентом  $N_{pj_k j_1}$  и приравнивание коэффициентов при одинаковых степенях  $T$  приводит к си-

стеме линейных уравнений. Решение таковой системы способом последовательных приближений дает искомые коэффициенты  $N_{p j_k j}$  неравенств порядка  $n$  с характеристикой  $p$  и фиксированными  $j_k$ . Аналогично находятся коэффициенты неравенств для всех возможных при данной характеристике наборов  $j_k$ .

Для решения ограниченной задачи трех тел методом Хилла-Брауна используется система программирования аналитических выкладок Авто-Аналитик [5]. Алгоритм решения данной задачи в этой системе описан в [6]. В нем сохранены все основные принципы метода, указанные выше. Но в целях сокращения объема промежуточных результатов, алгоритм предполагает особую форму записи рядов (2) в памяти машины. Вместо трех выражений (2) хранятся три массива выражений  $U_n, S_n, Z_n$  ( $n = 0, 1, 2, \dots$ ).

$$\begin{aligned} U_n &= \sum_p \alpha^{p_1} e^{p_2} e'^{p_3} \kappa^{p_4} F_m; \\ S_n &= \sum_p \alpha^{p_1} e^{p_2} e'^{p_3} \kappa^{p_4} V_m; \\ Z_n &= \sum_{p'} \alpha^{p'_1} e^{p'_2} e'^{p'_3} \kappa^{p'_4} W_m \end{aligned} \quad (4)$$

и совокупность выражений

$$\left. \begin{matrix} F_m \\ V_m \\ W_m \end{matrix} \right\} = \sum_{j_k} T^* (j_2; j_3; j_4 \div x; N_{p, j_k, -J+x}; N_{p, j_k, -J+x+2}; \dots). \quad (5)$$

Здесь  $\sum_i p_i = n$ ,  $\sum_i p'_i = n$ ,  $m$  — порядковый номер комбинации  $p_1, p_2, p_3, p_4$ , удовлетворяющей требованиям точности. Каждая формула вида (5) содержит всю необходимую информацию о неравенствах с характеристикой, имеющей номер  $m$ . Такое разделение записи решения позволяет на первых этапах алгоритма оперировать сокращенными решениями (4), а на конечных этапах операции производятся только над выражениями вида (5).

Исходя из изложенного выше и описанного в [6] алгоритма, приводим список модулей для решения данной задачи.

М1: Подстановка в исходные формулы выражений вида (4), в них раскрываются скобки и удерживаются члены порядка  $n$ .



Входные данные: величина  $n$ , исходные разложения, массивы решений вида (4).

Выходные данные: два выражения, содержащие только члены порядка  $n$ .

**M2:** Формирование выражений —  $T^{-1}(D^2 + 2mD)$  и  $-D^2z\sqrt{-1}$ , в них сохраняются только члены порядка  $n$ . Входные данные:  $n$ , массивы формул (4). Выходные данные: два результирующих выражения.

**M3:** Формирование формул вида  $p_1, p_2, p_3, p_4$  из четырех целых положительных чисел, для которых  $\sum_i p_i = n$  и выполняется условие точности.

Входные данные:  $n$ , условие точности. Выходные данные: совокупность наборов, число сформированных наборов  $k_n$ .

**M4:** Выборка из исходных формул слагаемых с характеристикой, соответствующей фиксированному набору  $p_1, p_2, p_3, p_4$ , и суммирование в одно выражение.

Входные данные: исходные формулы, выражение  $p_1, p_2, p_3, p_4$ . Выходные данные: выражение, содержащее слагаемые с фиксированной характеристикой.

**M5:** Каждая буква исходной формулы заменяется соответствующим выражением вида (5). Выполняются все необходимые операции, в результате которых исходная формула приводится к степенному ряду, записанному специальным образом.

Входные данные: исходное выражение, массив формул (5). Выходные данные: выражение вида

$$\sum_{j_k} T^* (j_2; j_3; j_4 \div x; A_{p, j_k, -j_k+x}; A_{p, j_k, -j_k+x+2}; \dots) \quad (6)$$

$k = 2, 3, 4$

**M6:** Для каждой пары слагаемых (с наборами  $j_k$  и  $-j_k$ ) исходной формулы вида (6) решением соответствующей системы линейных уравнений определяются неизвестные коэффициенты  $N_{p, j_k, j_l}$  и  $N_{p, -j_k, j_l}$ . Формируется часть решения  $\alpha^{p_1} e^{p_2} e^{p_3} k^{p_4} F_m$  или  $\alpha^{p_1} e^{p_2} e^{p_3} k^{p_4} W_m$ , соответствующая характеристике с номером  $m$ . Кроме того, формируются выражения  $F_m, V_m$  или  $W_m$ .

Входные данные: величины  $n, m$ ; набор  $p_1, p_2, p_3, p_4$ ; выражение вида (6). Выходные данные: выражения  $F_m, V_m$  или  $W_m$  вида (5); формулы

$$\alpha^{p_1} e^{p_2} e^{p_3} k^{p_4} F_m \text{ или } \alpha^{p_1} e^{p_2} e^{p_3} k^{p_4} W_m.$$

Как уже отмечалось, для решения задачи Хилла—Брауна используется система программирования Авто-Аналитик. Модульный принцип организации этой системы делает ее легко расширяемой и адаптируемой к требованиям пользователя. Основную практическую ценность системы составляет библиотека стандартных операторов. Описанные выше модули оформлены в виде операторов и включены в библиотеку Авто-Аналитика. Управляющая программа реализующая метод Хилла—Брауна, также организована в виде оператора, его стандартное описание состоит из последовательных обращений к указанным операторам.

При построении операторов для реализации алгоритма Хилла—Брауна активно используется накопленная совокупность модулей системы, а также группа вспомогательных операторов, специально построенных для решения задач небесной механики. Это, например, операторы Ограничение и Выборка, описанные в [7], а также операторы, выполняющие алгебраические действия над выражениями вида (6).

Изложенное представление вычислительного алгоритма позволяет просто реализовать различные модификации метода, а отдельные операторы использовать при решении других задач небесной механики.

## ЛИТЕРАТУРА

1. Коновалов А. Н., Яненко Н. Н. Модульный принцип построения программ как основа создания пакета прикладных программ решения задач механики сплошной среды.— В сб.: Комплексы программ математической физики. Новосибирск, 1972.
2. Brown E. W. Theory of the Motion of the Moon. Mem. Astr. Soc., 53, 1899; 54, 1904; 57, 1908; 59, 1908.
3. Токмалаева С. С. Аналитическая теория движения седьмого спутника Юпитера.— «Бюлл. ИТА», 4, 7, 1956.
4. Шор В. А. Применение быстродействующих вычислительных машин к решению ограниченной задачи трех тел методом Хилла—Брауна.— «Бюлл. ИТА», 7, 8, 1960.
5. Арайс Е. А., Сибиряков Г. В. Авто-Аналитик. Изд. Новосибирского университета, 1973.
6. Московкина Л. А. Алгоритм решения в системе Авто-Аналитик ограниченной задачи трех тел методом Хилла—Брауна.— «Астр. и геод. Труды ТГУ», вып. 5, 1975.
7. Московкина Л. А. О некоторых операторах для реализации метода Хилла—Брауна в системе Авто-Аналитик.— Материалы четвертой научн. конф. Томского ун-та по мат. и мех., Томск, 1974.



## РЕШЕНИЕ НЕКОТОРЫХ ЗАДАЧ КАЛЕНДАРНОГО ПЛАНИРОВАНИЯ НА ОСНОВЕ ОДНОЙ МОДЕЛИ МЕЖЦЕХОВЫХ СВЯЗЕЙ

В. А. КОРОБИЦЫН, Б. А. МАЛЬЦЕВ

Межцеховое планирование всегда играло значительную роль в структуре внутривзаводских связей. В условиях разработки АСУП мелкосерийного и единичного производства с большой номенклатурой товарной продукции решение задач календарного планирования ввиду значимости результатов планирования в управлении производством является первоочередным. Применение ЭВМ дает возможность детального календарного планирования цехов, ориентированного на производственную программу основного производства.

Основной задачей межцехового недельно-суточного планирования является формирование детальных календарных планов, увязывающих работу всех производственных подразделений предприятия с обеспечением необходимых пропорциональных заделов.

Принцип синхронности можно сформулировать следующим образом.

Запуск в производство любого продукта (детали, узла, изделия) должен производиться не позднее того момента времени, когда его запас становится меньше планового потребления за период, равный длительности цикла его производства.

Основываясь на конструктивной взаимосвязи всех продуктов предприятия, принцип синхронности можно представить в виде следующего рекуррентного соотношения [1] :

$$z_i(t) \geq d_i(t + T_i) + \sum_j z_j(t + T_i) r_{ij} - g_i - f_i, \quad (1)$$

где  $z_i(t)$  — план запуска  $i$ -го продукта;

- $d_i$  — директивный план выпуска  $i$ -го продукта;  
 $r_{ij}$  — норма расхода  $i$ -го продукта на изготовление единицы  $j$ -го продукта;  
 $T_i$  — длительность производственного цикла изготовления  $i$ -го продукта;  
 $g_i$  — количество готового  $i$ -го продукта;  
 $f_i$  — количество  $i$ -го продукта, находящегося в процессе производства.

Решение системы неравенств (1) в общем случае затруднено ввиду ее нелинейности и большой размерности.

Делаются следующие предположения о характере производства на предприятии:

1) при планировании выпуска товарной продукции решающим являются мощности основного производства предприятия;

2) структурная модель товарной продукции является простой (двух-, трехзвенной).

Будем решать систему (1) в два этапа. На первом этапе определим план производства товарной продукции при условии максимальной загрузки основного производства и далее, подставляя вычисленные значения  $z_i$  в уравнения системы (1), получим, в силу предположения 2, простую систему линейных неравенств. Заменяя неравенства на строгие равенства и решая полученную систему линейных уравнений простой структуры, получим план вспомогательного производства, увязанный с планом работы основного производства.

Таким образом, решение задачи межцехового планирования ищется в два этапа: этап недельно-суточного планирования основного производства (разработка детального календарного плана выпуска товарной продукции) и этап недельно-суточного планирования вспомогательного производства предприятия, увязанного с планом выпуска товарной продукции.

Считается, что часть заказов на товарную продукцию является директивным планом, а остальные заказы являются частью годовых обязательств. Необходимо на первом этапе решить задачу максимальной загрузки производственных мощностей основного производства.

Пусть между производственными подразделениями существует специализация. Сформулируем задачу в терминах линейного программирования [2]:



найти  $x_{il}$ , доставляющие максимум линейной формы

$$\max \sum_i \sum_{l \in M} \xi_i x_{il}, \quad M = \bigcup_q M_q, \quad (2)$$

и удовлетворяющие ограничениям:

$$\sum_{l \in M_q} x_{il} = y_{iq}, \quad q = 1, 2, \dots, q_n, \quad i \in I^*, \quad (3)$$

$$\sum_{l \in M_q} x_{il} \leq y_{iq}, \quad i \in I \setminus I^*, \quad (4)$$

$$\sum_l x_{il} r_{lk} \leq c_{kl}, \quad l \in \bigcup_q M_q, \quad (5)$$

$$x_{il} \geq 0, \quad (6)$$

где  $x_{il}$  — план запуска  $i$ -го продукта в  $l$ -й день периода  $M$ ;

$\xi_i$  — оценка  $i$ -го продукта;

$M_q$  —  $q$ -е подмножество дней периода  $M$ ;

$y_{iq}$  — потребность в  $i$ -м продукте за период  $M_q$ ;

$I$  — множество заказов;

$I^*$  — множество заказов директивной группы;

$r_{ik}$  — норма расхода  $k$ -го ресурса на изготовление  $i$ -го продукта;

$c_{kl}$  — наличие  $k$ -го ресурса в  $l$ -й день.

Условия (5) — (6) являются жесткими ограничениями, это вытекает из экономического смысла задачи и из ограниченности ресурсов. В случае если условия (3) — (6) являются противоречивыми, ищется компромиссное решение. Сущность компромисса в том, что условия (3) нарушаются не полностью и ставится цель получить решение при выполнении максимально возможного числа этих уравнений:

$$\max \left\{ \sum P_i \left| \sum_{l \in M_q} x_{il} = y_{iq} \right. \right\}, \quad i \in I^*, \quad q = 1, 2, \dots, q_r. \quad (7)$$

В [3] предложен алгоритм приближенного решения задачи (2), (4) — (7).

Используя полученное решение в дальнейшей процедуре моделирования межцеховых связей, получим решение задач недельно-суточного планирования производственных подразделений, увязанное с директивным планом поставки товарной продукции.

Разработанные модели положены в основу подсистемы оперативно-производственного планирования завода

**ЖБК-100** Томского территориального управления строительства. Первая очередь подсистемы (решение задачи недельно-суточного планирования формовочных цехов завода с учетом потребностей строительных объектов), основанная на алгоритме [3] решения модели (2), (4) — (7), функционирует на предприятии более года.

За это время при общем увеличении объема выпускаемой продукции, росте производительности труда удалось существенно улучшить комплектацию строящихся объектов. Так, если в 1973 году объем выпуска сборного железобетона за 11 месяцев составил 37394 м<sup>3</sup>, то за тот же период 1974 года он вырос до 45141 м<sup>3</sup>, рост 20,7%. Производительность труда на одного работающего за 11 месяцев 1974 года выросла против соответствующего периода прошлого года на 8,1%. Если в 1973 году уровень комплектации объектов конструкциями этого завода составил 65—70%, то в 1974 году он возрос до 88—90%, что особенно важно, так как для заводов сборного железобетона главной задачей является комплектация объектов.

При выпуске большого количества единичных и мелкосерийных изделий с широким диапазоном трудоемкости, различными сроками поставки в течение календарного месяца с низким нормативом готовой продукции, вопрос комплектации продукцией, выходящей непосредственно из формовочного цеха на строительную площадку, всегда чрезвычайно затруднен. Это обычно вызывается тем, что календарный план изготовления изделий просчитывается по одному какому-либо фактору — обычно по наличию оснастки или трудоемкости. Описываемая подсистема позволяет с учетом целого ряда факторов установить реальный календарный план формовочным цехам, позволяет вести ежедневный контроль за изготовлением и отгрузкой любого изделия.

#### ЛИТЕРАТУРА

1. Ефимов Ю., Кизев В., Невраев В., Седельников П. Разработка и исследование экономико-математической модели межцехового планирования и диспетчирования основного производства. — Материалы I итоговой конференции НИИАЭМ «Автоматизация производственных процессов и установок. Автоматизированные системы управления». Томск, 1973.
2. Юдин Д. Б., Гольштейн Е. Г. Линейное программирование (теория и конечные методы), М., Физматгиз, 1963.
3. Коробицын В. А. Решение некоторых задач недельно-суточного планирования ЗЖБК. — В сб.: Вопросы математического обеспечения АСУП, Томск, Изд-во Томского университета, 1974.



## СОДЕРЖАНИЕ

|                                                                                                                                                                                      |     |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Е. А. Арайс, В. М. Дмитриев. Символическое моделирование и анализ электронных схем на ЭВМ                                                                                            | 3   |
| Е. А. Арайс. Анализ нелинейных цепей в системе МАРС.                                                                                                                                 | 25  |
| В. М. Дмитриев. Графическая форма модельного метода                                                                                                                                  | 35  |
| В. М. Дмитриев. Связь модельного метода с другими методами анализа цепей                                                                                                             | 49  |
| Е. А. Арайс, В. Ф. Шадрин. Применение модельного метода к анализу цепей с распределенными параметрами                                                                                | 66  |
| Е. А. Арайс, Л. А. Шапиро. Исследование линейных цепей в системе МАРС                                                                                                                | 74  |
| С. А. Разин. Вычисление производных от выходных параметров линейных цепей                                                                                                            | 94  |
| Б. Ш. Гельфман. Реализация входного языка Авто-Аналитика на ЭВМ БЭСМ-6                                                                                                               | 110 |
| Е. А. Арайс, А. В. Шутенков. Язык МАРС                                                                                                                                               | 117 |
| А. В. Шутенков. Транслятор с входного языка системы МАРС                                                                                                                             | 136 |
| Е. А. Арайс, А. К. Мартынов, Л. А. Шапиро. О задаче оптимального выбора инструмента                                                                                                  | 148 |
| А. А. Никольский. Система интерпретации гравитационных и магнитных аномалий (СИГМА)                                                                                                  | 156 |
| А. С. Батулин, В. И. Лившиц, А. К. Мартынов, В. П. Сушко. Анализ экономической эффективности выполнения гравировальных работ на фрезерных станках с числовым программным управлением | 166 |
| М. Н. Боголюбова, В. И. Лившиц, А. К. Мартынов. Автоматизация подготовки управляющих программ для выполнения гравировальных работ на фрезерных станках с ЧПУ                         | 183 |
| М. Н. Боголюбова, В. И. Лившиц, А. К. Мартынов. Структура автоматизированной системы «Гравер» подготовки программ для станков с числовым программным управлением                     | 196 |
| А. А. Ткаченко. Об одной реализации алгоритма определения временных параметров сетевых графиков                                                                                      | 213 |
| Л. А. Московкина. Модульная структура вычислительного алгоритма в задаче Хилла — Брауна                                                                                              | 217 |
| В. А. Коробицын, Б. А. Мальцев. Решение некоторых задач календарного планирования на основе одной модели межцеховых связей                                                           | 223 |

ВОПРОСЫ ПРОГРАММИРОВАНИЯ  
И АВТОМАТИЗАЦИИ ПРОЕКТИРОВАНИЯ

Выпуск 3

Томск, Изд-во ТГУ, 1977 г., 228 с. + 2 вкл.

Редактор **А. В. Лисевич**  
Технический редактор **Н. А. Невиницына**  
Корректор **Н. И. Викторенко**

---

К302037. Сдано в набор 2/X-76 г. Подписано к печати 8/II-77 г.  
Бум. типографская № 3; формат 60×84<sup>1/16</sup>; п. л. 14,3 + 2 вкл.,  
усл. п. л. 13,6; уч-изд. л. 11,5  
Заказ 2356. Тираж 500. Цена 1 руб. 73 коп.

---

Издательство ТГУ. Томск-29, ул. Никитина, 17  
Типография издательства «Красное знамя», г. Томск, ул. Советская, 47





Цена 1 руб. 73 коп.

Томский госуниверситет 1878



Научная библиотека 00932482