

УДК 519.71

В. Н. ЕФАНОВ, М. И. АХМЕТОВ

### СОВМЕЩЕННЫЙ СИНТЕЗ АЛГОРИТМИЧЕСКОГО ОБЛИКА И АРХИТЕКТУРЫ БОРТОВЫХ ИНФОРМАЦИОННО-УПРАВЛЯЮЩИХ СИСТЕМ

В статье рассматривается проблема синтеза архитектуры бортовой информационно-управляющей системы с заданным набором свойств. Исследуются условия, при выполнении которых можно получить оптимальный облик синтезируемой системы. Предлагается аналитическая модель для описания вычислительных процессов в бортовой информационно-управляющей системе, являющаяся основой для решения задачи совмещенного синтеза алгоритма управления. *Фактор-множество, совмещенный синтез, монотонные преобразования, аналитическая модель, рекурсивные алгоритмы, макрокоманда, арифметико-логический граф, диспетчеризация*

#### ВВЕДЕНИЕ

Перспективные направления развития современного авиастроения связаны с реализацией ряда крупных научно-технических программ, выполненных как у нас, так и за рубежом (ИКБО-95, АРИА-200, DAIS, PAVE PILLAR и др.), которые регламентируют функциональный состав единой информационно-управляющей системы (ИУС) летательного аппарата, ее информационного интерфейса, устанавливающего архитектуру информационного обмена между их основными функциональными элементами, а также определяют порядок исследования динамики процессов, протекающих в подобных системах. В результате были сформированы принципы организации ИУС на основе территориально-распределенной неоднородной многомашинной вычислительной системы, обладающей определенными возможностями к реконфигурации на аппаратном уровне. Для объединения элементов многомашинной ИУС предлагается использовать стандартные цифровые соединения на основе специализированных технологий информационного обмена с централизованным или децентрализованным методом доступа (MIL-STD-1553B, STANAG3910, AS4074, AS4075). При этом предполагается, что разработка алгоритмического, программного и аппаратного обеспечения подобных многомашинных вычислительных комплексов может осуществляться независимо друг от друга, поскольку любой вычислительный алгоритм может быть реализован с той или иной долей успеха в рамках любой из известных топологий. Вместе с тем, независимое осуществление этапов разработки алгоритмического обеспечения и его технической реализации может привести к неэффективному, в целом, решению задачи проектирования ИУС. В самом деле, реализация одного и того же алгоритма управления с помощью различных вариантов архитектуры ИУС приводит к различным значениям производительности или пропускной способности, среднего запаздывания решения задач, среднего времени простоя компонентов ИУС, коэффициентов связности по общей памяти и общим данным, а также других показателей, определяющих эффективность работы многопроцессорной системы. Следовательно, для каждого вычис-

лительного алгоритма управления может быть выбрана наиболее выгодная архитектура ИУС. Очевидно, что оптимальное решение может не принадлежать к набору стандартных топологий, а требовать разработки специализированной архитектуры. С другой стороны, реализация различных вариантов алгоритмического обеспечения на ИУС с фиксированной топологией также приведет к различным значениям показателей вычислительной эффективности. Это означает, что не только для каждого вычислительного алгоритма существует своя оптимальная конфигурация вычислительной системы, но и для конкретного структурного варианта построения ИУС можно подобрать наиболее подходящий вычислительный алгоритм. Отсюда следует вывод о том, что синтез алгоритмов управления, планирование вычислительного процесса и выбор структурной организации аппаратной части необходимо осуществлять в рамках единой процедуры, обеспечивающей направленное формирование архитектуры ИУС с заданным набором свойств. Один из подходов к решению сформулированной задачи предлагается в данной работе.

#### 1. УСЛОВИЕ СОГЛАСОВАНИЯ ОСНОВНЫХ ЭТАПОВ ФОРМИРОВАНИЯ ОБЛИКА БОРТОВЫХ ИНФОРМАЦИОННО-УПРАВЛЯЮЩИХ СИСТЕМ

Рассмотрим условия, при выполнении которых последовательность операций выбора оптимальных алгоритмов управления и структурной организации аппаратной части позволяет получить оптимальный облик разрабатываемой многопроцессорной ИУС. Обозначим через  $X^{(1)}$  и  $X^{(2)}$ , соответственно, множество вариантов алгоритмов управления и топологических вариантов построения многопроцессорной ИУС. Для оценки эффективности рассмотренных вариантов введем две совокупности локальных критериев  $\varepsilon^{(1)}$  и  $\varepsilon^{(2)}$ , первая из которых определяет качество синтезированных алгоритмов управления, а вторая – оценивает вычислительную эффективность работы многопроцессорной системы. Полное множество возможных вариантов построения бортовой ИУС обозначим через  $X$ .

Разобьем теперь множество  $X^{(1)}$  на подмножества  $X^{(1,k)}$ ,  $k=1, 2, \dots, N_1$ , включающие элементы исходного множества, эквивалентные с точки зрения критерия  $\varepsilon^{(1)}$ . Каждому элементу  $X_i^{(1,k)} \in X^{(1,k)}$ ,  $i=1, 2, \dots, n_{1,k}$ , соответствует некоторое подмножество  $X[1,k,i] \subset X$ . Объединяя подмножества  $X[1,k,i]$  по всем  $i$ , получаем совокупность элементов

$$X[1,k] = \bigcup_i X[1,k,i] \subset X, \quad (1)$$

образующих класс эквивалентности с точки зрения критерия  $\varepsilon^{(1)}$ . Очевидно, что как первому, так и второму этапу разработки ИУС соответствует свое разбиение множества  $X$  в определенную совокупность непересекающихся подмножеств

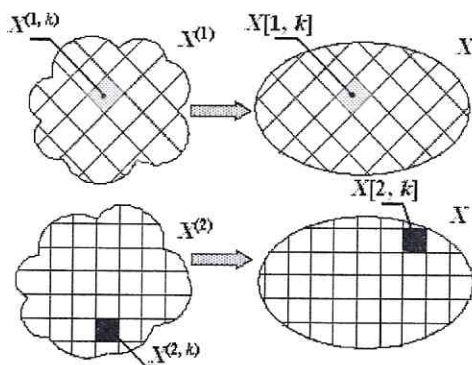


Рис. 1. Формирование фактор-множеств множества  $X$

Для того чтобы в результате двухэтапной процедуры был сформирован оптимальный вариант облика  $X_i^* \in X$ , необходимо на каждом этапе разработки ИУС выбирать в качестве оптимального такое подмножество  $X^{(l,k)*}$  ( $l=1, 2$ ), которому соответствует класс эквивалентности  $X^*[l,k]$ , включающий элемент  $X_i^*$ , т.е.

$$e^{(l)}(X^{(l,k)*}) = \text{extr}_{X^{(l,k)} \subset X^{(l)}} e^{(l)}(X^{(l,k)}), \quad (3)$$

где  $X_i^* \in X^*[l,k] \Leftrightarrow X^{(l,k)*}$ ,  $l=1, 2$ .

При выполнении сформулированного требования двухэтапной процедуре совмещенного синтеза будет соответствовать, как это показано на рис. 2, последовательность вложенных подмножеств вида

$$X^*[1,k_1] \subset X^*[2,k_2] \sim X_i^*. \quad (4)$$

Однако на любом из двух этапов описанной процедуры полное множество возможных вариантов  $X$  и, следовательно, вариант  $X_i^*$  не построены. Поэтому непосредственная проверка условия (4) оказывается невозможной. В то же время переход от первого ко второму этапу сопровождается дискретным

$$X = \bigcup_k X[l,k], \quad k=1, 2, \dots, N_l; \quad l=1, 2. \quad (2)$$

Графическая интерпретация подобного разбиения приведена на рис. 1.

Для того чтобы в результате двухэтапной процедуры был сформирован оптимальный вариант облика  $X_i^* \in X$ , необходимо на каждом этапе разработки ИУС выбирать в качестве оптимального такое подмножество  $X^{(l,k)*}$ , которому соответствует класс эквивалентности  $X^*[l,k]$ , включающий элемент  $X_i^*$ , т.е.

$$e^{(l)}(X^{(l,k)*}) = \text{extr}_{X^{(l,k)} \subset X^{(l)}} e^{(l)}(X^{(l,k)}), \quad (3)$$

где  $X_i^* \in X^*[l,k] \Leftrightarrow X^{(l,k)*}$ ,  $l=1, 2$ .

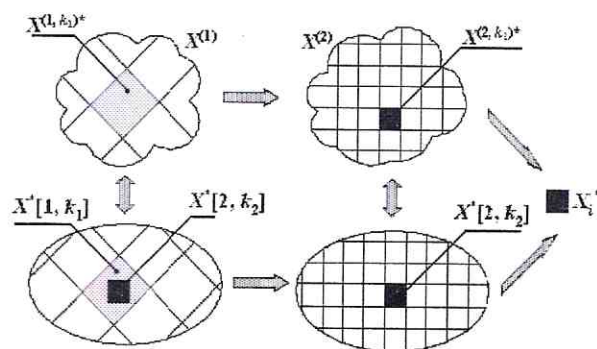


Рис. 2. Процедура совмещенного синтеза по критериям  $\varepsilon^{(1)}$  и  $\varepsilon^{(2)}$

изменением множества целевых функций  $\varepsilon^{(1)}$  и множества вариантов  $X[l,k]$  ( $l=1, 2$ ), первое из которых, как отмечалось выше, объединяет возможные алгоритмы управления, а второе – топологические варианты построения многопроцессорной ИУС. Это приводит к нарушению условий совместимости локальных решений (4).

Важным критерием соблюдения условий совместимости является существование монотонных преобразований, переводящих области значений функций  $\varepsilon^{(1)}$  в области значений  $\varepsilon^{(2)}$  [1, 2]. Сформулируем указанное требование применительно к рассматриваемой задаче формирования облика многопроцессорной ИУС в виде следующего утверждения.

**Утверждение 1.** Если существует множество монотонных преобразований  $W_i(\alpha)$ ,  $i=1, 2, \dots, m$ , таких, что  $i$ -е преобразование переводит область значений  $i$ -й компоненты критерия  $\varepsilon_i^{(2)}(X)$  в область значений критерия  $\varepsilon_i^{(1)}(X)$  и  $\varepsilon_i^{(1)}(X) = W_i\{\varepsilon_i^{(2)}(X)\}$  для всего множества  $X$ , то последовательная оптимизация локальных критериев  $\varepsilon_i^{(1)}(X)$ ,  $l=1, 2$ , приводит к формированию оптимального варианта облика системы.

**Доказательство.** Рассмотрим случай максимизации критериев  $\varepsilon_i^{(1)}$  и  $\varepsilon_i^{(2)}$  при монотонно-

возрастающем преобразовании  $W_i(\alpha)$ , остальные случаи доказываются аналогично.

Пусть  $X_1, X_2 \in X$  – такие, что  $\varepsilon_i^{(2)}(X_1) > \varepsilon_i^{(2)}(X_2)$ ,  $i = 1, 2, \dots, m$ . По условию утверждения  $W_i\{\varepsilon_i^{(2)}(X_1)\} > W_i\{\varepsilon_i^{(2)}(X_2)\}$ , следовательно,  $\varepsilon_i^{(1)}(X_1) > \varepsilon_i^{(1)}(X_2)$ . Данное неравенство выполняется для всех элементов соответствующих классов эквивалентности  $X_1 \in X[1,1], X_2 \in X[1,2]$ . В свою очередь, каждому классу эквивалентности соответствует определенное множество вариантов алгоритмического обеспечения МВС  $X^{(1,1)} \Leftrightarrow X[1,1], X^{(1,2)} \Leftrightarrow X[1,2]$ . Отсюда получаем  $\varepsilon_i^{(1)}(X^{(1,1)}) > \varepsilon_i^{(1)}(X^{(1,2)})$ . Таким образом, предпочтительный вариант в смысле критерия  $\varepsilon^{(2)}$  принадлежит, при выполнении условий утверждения, к множеству предпочтительных вариантов в смысле критерия  $\varepsilon^{(1)}$ , что соответствует сформулированному результату.

Доказанное утверждение служит методической основой для разработки процедуры совмещенного синтеза алгоритмов управления, планирования вычислительного процесса и выбора структурной организации аппаратной части, поскольку условия существования монотонных преобразований диктуют способ диспетчеризации вычислительного процесса, обеспечивающий совместимость локальных целей формирования алгоритмического обеспечения и топологии многопроцессорной ИУС.

Чтобы получить аналитическое выражение для множества монотонных преобразований  $W_i(\alpha)$ ,  $i = 1, 2, \dots, m$ , необходимо выразить характеристики возможных вариантов алгоритмов управления через соответствующие характеристики вариантов структурной организации аппаратной части. Для этого предлагается математическая модель, позволяющая описывать управляющую часть ИУС с учетом специальных структурных матриц, определяющих состав процессорных блоков и модулей памяти, а также структуру связей между ними. На базе этой модели реализуется процедура совмещенного синтеза алгоритмов управления и вариантов структурной организации бортовой многопроцессорной ИУС.

## 2. АНАЛИТИЧЕСКАЯ МОДЕЛЬ ДЛЯ ОПИСАНИЯ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ В БОРТОВЫХ ИНФОРМАЦИОННО-УПРАВЛЯЮЩИХ СИСТЕМАХ

Алгоритмы управления, реализующие заданную программу для фиксированного режима работы объекта управления, относятся к классу рекурсивных алгоритмов, т.е. в них повторяются операции одного и того же типа над последовательно поступающими данными. Такие алгоритмы в дискретной форме записи могут быть представлены в виде следующей совокупности разностных уравнений:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k); \\ y(k+1) &= Cx(k+1). \end{aligned} \quad (5)$$

Здесь векторы  $x(k)$  и  $u(k)$  определяют исходное состояние вычислительного процесса на  $(k+1)$ -м интервале управления и состояние внешних воздействий, влияющих на ход вычислительного процесса, вектор  $y(k)$  задает совокупность возможных результатов.

При этом следует учесть, что многорежимные управляющие алгоритмы бортовых систем относятся к рекурсивным арифметико-логическим алгоритмам, которые объединяют несколько возможных вариантов алгоритма (5) применительно к возможным контекстам общего алгоритма.

Основными характеристиками рекурсивных алгоритмов служат параллелизм и конвейерность алгоритма.

Алгоритм обладает параллелизмом уровня  $d$ , если максимально возможное количество операций, выполняемых за один отсчет времени (например, за такт исполнения макрокоманды) равно  $d$ . В свою очередь, алгоритм характеризуется конвейерностью длительности  $f$ , если количество отсчетов времени, за которое реализуется алгоритм, равно  $f$ .

Введенные понятия позволяют сопоставить свойства алгоритма управления с особенностями вычислительной системы, на которой он выполняется. Так, если алгоритм обладает параллелизмом единичного уровня (т.е. допускает только последовательное выполнение операций), то он может реализовываться на одном вычислительном блоке. При этом остальные вычислительные блоки многопроцессорной системы должны быть загружены другими задачами, либо будут простаивать. В то же время выполнение алгоритма, допускающего распараллеливание вычислений, на однопроцессорном вычислительном устройстве приводит к задержке вычислений. В общем случае длительность исполнения алгоритма, выраженная в тактах макрокоманд, не может быть меньше его конвейерности, даже при условии, что на каждом временном такте задействуется число процессоров, равное соответствующему уровню параллелизма алгоритма. Поскольку уровень параллелизма может меняться в зависимости от такта вычислительного алгоритма, то для обеспечения минимально возможной длительности исполнения алгоритма число процессоров в вычислительной системе должно быть не меньше, чем максимальный уровень параллелизма. Однако на тех тактах вычислений, для которых уровень параллелизма меньше максимального, отдельные процессоры будут незагруженными, что приводит к снижению производительности вычислительной системы. Таким образом, оптимальное количество процессоров в многопроцессорной ИУС должно выбираться на основе компромисса между длительностью исполнения алгоритма и производительностью ИУС.

Перейдем теперь к исследованию алгебраических свойств множества операций, образующих алгоритм управления. Обозначим результаты выполнения операций, декомпозированных до уровня макрокоманд микропроцессора, через  $z_j^i$ , где  $i$  ( $i = 1, 2, \dots, f$ ) – номер временного такта выполнения соответ-

вующих операций;  $j$  – номер операции в  $i$ -й совокупности операций, допускающих параллельное выполнение ( $j=1, 2, \dots, d_i$ ;  $d_i$  – уровень параллелизма на  $i$ -м такте). Отметим, что нумерация операций в пределах указанной совокупности является произвольной. Установим связь результатов операций  $z_j^i$  с математической моделью алгоритма управления.

Как правило, многопроцессная и многозадачная реализация рекурсивного арифметико-логического алгоритма характеризуется конечным числом дискретных состояний (контекстов)  $q_1, q_2, \dots, q_M$ , в каждом из которых выполняется своя ветвь вычислительного процесса. Переключение контекста осуществляется по текущей информации о состоянии вычислительного процесса  $x$ , внешней среды  $\mu$ , а также по командам на переключение  $q_c$ , поступающим из внешних систем. Аналитически процедуру переключения контекста можно выразить следующим образом:

$$q = c_K(x, \mu, q_c), \quad (6)$$

где  $q$  – символьная переменная, принимающая значения из области  $\Omega = \{q_1, q_2, \dots, q_M\}$ ,  $\rho_K$  – оператор квантования, который переменным величинам  $x, \mu$  и дискретным сигналам  $q_c$  ставит в соответствие значение символа  $q \in \Omega$ .

Опишем состояние каждого из характерных фрагментов алгоритма управления уравнениями следующего вида:

$$\begin{aligned} x_i(k+1) &= A_i x_i(k) + B_i u_i(k); \\ y_i(k) &= C_i x_i(k); \quad i = 1, 2, \dots, N. \end{aligned} \quad (7)$$

Здесь векторы  $x_i(k)$  описывают текущее состояние  $i$ -го фрагмента вычислительного процесса на  $k$ -м временном такте, векторы  $u_i(k)$  определяют состояние внешних воздействий, влияющих на ход вычислительного процесса, векторы  $y_i(k)$  содержат массивы выходных данных.

Взаимодействие отдельных фрагментов алгоритма управления применительно к каждому из возможных контекстов можно описать следующим уравнением:

$$u(k) = LE(q)y(k) + L_0 E_0(q)g(k), \quad (8)$$

где  $u(k), y(k)$  – прямые суммы векторов  $u_i(k)$  и  $y_i(k)$ ,  $g(k)$  – вектор задающих воздействий;

$L = \|L_{ij}\|_{N \times N}$ ,  $L_0 = \|L_{j0}\|_{N \times 1}$  – блочные матрицы;

$E(q) = \text{blockdiag} \{E_1(q), \dots, E_N(q)\}$  – блочно-диагональная матрица, элементы которой являются единичными или нулевыми матрицами в зависимости от того, принимает или не принимает участие в реализации  $q$ -го контекста соответствующий фрагмент алгоритма управления; аналогичный смысл имеет матрица  $E_0(q)$ , элементы которой определяют

участие в вычислительном процессе соответствующих внешних воздействий.

Объединяя (7) и (8), получим математическую модель рекурсивного ветвящегося алгоритма в следующем виде:

$$\begin{aligned} x(k+1) &= A_c(q)x(k) + B_c(q)u(k); \\ y(k) &= Cx(k). \end{aligned} \quad (9)$$

Здесь  $x(k)$  – прямая сумма векторов  $x_i(k)$ ,  $A_c(q) = [A + BLE(q)C]$ ,  $B_c(q) = BL_0 E_0(q)$ ,  $A = \text{blockdiag} \{A_1, A_2, \dots, A_N\}$ ,  $B = \text{blockdiag} \{B_1, B_2, \dots, B_N\}$ ,  $C = \text{blockdiag} \{C_1, C_2, \dots, C_N\}$ .

Используя введенные математические описания для исследуемых рекурсивных арифметико-логических алгоритмов, перейдем к построению аналитической модели для описания вычислительных процессов в бортовых ИУС.

Поставим в соответствие алгоритму (9) арифметико-логический граф, определяющий последовательность выполнения арифметических и логических операций, операций чтения-записи данных в буферные ОЗУ, вызова констант из ПЗУ и т. д. Этот ориентированный граф включает следующие типы вершин:

- вершины-истоки, соответствующие операциям чтения из буферных ОЗУ и ПЗУ или кэш-памяти процессора;
- арифметические вершины, соответствующие операциям сложения или умножения двух операндов – в том числе и умножения на логическую переменную;
- логические вершины, соответствующие определенным логическим операторам (например, условный оператор), которые определяют связи между операторами по управлению, задавая состав и порядок их выполнения.

Рассматриваемый граф является взвешенным, т.е. его ветвям присваиваются веса. С этой точки зрения ветви делятся на два класса:

- ветви с постоянными весами;
- ветви с логическими весами (соответствуют логическим переменным).

Рассмотрим способ построения подобного графа на следующем примере.

Пусть алгоритм управления описывается следующей совокупностью уравнений:

$$\begin{aligned} x_1(k+1) &= a_1 x_1(k) + b_1 u_1(k); \\ y_1(k+1) &= c_1 x_1(k+1). \end{aligned}$$

Для этого примера арифметико-логический граф будет иметь вид, представленный на рис. 3.

Из этого графа следует, что конвейерность данного алгоритма  $f=3$ , а уровень параллелизма составляет на первом такте  $d_1=2$ , на втором и третьем  $d_2=d_3=1$ .

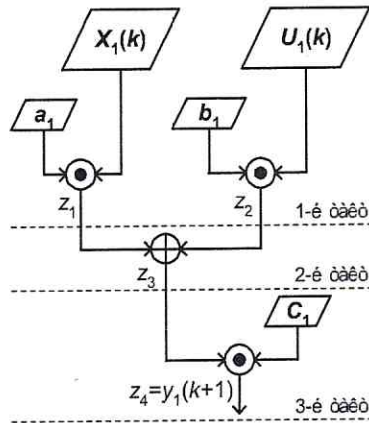


Рис. 3. Граф алгоритма без ветвлений

Перейдем теперь к исследованию алгебраических свойств множества операций, образующих алгоритм управления. Обозначим результаты выполнения операций, декомпозированных до уровня макрокоманд микропроцессора, через  $z_i$ , где  $i$  – номер соответствующей операции ( $i=1, 2, \dots, r$ ). Поставим в соответствие алгоритму управления, а, следовательно, и графу, следующую аналитическую модель:

$$z = \Pi_1 z + \Pi_2 v, \quad (10)$$

где  $z$  – обобщенный вектор результатов операций;  $v = [x(k); u(k)]^T$  – вектор исходных данных;  $\Pi_1, \Pi_2$  – матрицы, определяющие последовательность выполнения операций в соответствии с алгоритмом управления в зависимости от результатов выполнения предыдущих операций и исходных данных, соответственно.

Основная проблема, возникающая при разработке аналитических моделей вычислительных процессов, заключается в их высокой размерности. Это связано, в первую очередь, с наличием большого числа макрокоманд, таких как арифметико-логические операции, операции чтения-записи данных в буферное ОЗУ. Число подобных операций в зависимости от сложности алгоритма управления может достигать нескольких тысяч. Для сокращения размерностей разрабатываемых моделей предлагается следующее утверждение.

**Утверждение 2.** Пусть  $H_j = I_r - e_j \Pi_j^{(1)}$ , где  $\Pi_j^{(1)}$  –  $j$ -я строка матрицы  $\Pi_1$ ,  $e_j$  –  $j$ -й координатный вектор,  $r = \dim z$  – размерность вектора  $z$ , тогда решение системы (10) может быть представлено в следующем виде:

$$z = \prod_{i=f}^1 \left[ (d_i + 1)I_r - \sum_{j=p_i+1}^{p_i+d_i} H_j \right] \Pi_2 v, \quad (11)$$

где  $p_i = \sum_{k=1}^{i-1} d_k$ .

**Доказательство.** Тривиальное решение системы (10) имеет вид  $z = (I_r - \Pi_1)^{-1} \Pi_2 v$ . Однако вследствие чрезвычайно высокой размерности вектора  $z$  это решение оказывается непригодным. В то же время особенности построения вектора  $z$  приводят к тому, что матрица  $(I_r - \Pi_1)$  является нижней треугольной с единичной главной диагональю, поскольку результаты операций, выполняемых на некотором  $i$ -м временном такте, могут быть получены только на основе использования результатов ранее выполненных операций. Как известно, любая матрица вида  $(I_r - \Pi_1)$  может быть представлена как произведение матриц типа  $M$  [3], которые можно обозначить как  $H_j$ , т.е.  $(I_r - \Pi_1) = H_2 H_3 \dots H_r$ . Тогда для обратной матрицы справедливо следующее выражение:

$$(I_r - \Pi_1)^{-1} = H_r^{-1} H_{r-1}^{-1} \dots H_2^{-1} \quad (12)$$

Разобьем полученное произведение обратных матриц на группы сомножителей, относящихся к операциям, выполняемым на одном и том же временном такте:

$$H_r^{-1} H_{r-1}^{-1} \dots H_2^{-1} = \prod_{i=f}^1 \prod_{j=p_i+d_i}^{p_i+1} H_j^{-1}. \quad (13)$$

Поскольку нумерация операций в пределах каждой совокупности, объединяющей операции, допускающие параллельное выполнение, является произвольной, то матрицы  $H_j^{-1}$  из соответствующей совокупности являются перестановочными, т.е.

$$\prod_{j=p_i+d_i}^{p_i+1} H_j^{-1} = \prod_{j=p_i+1}^{p_i+d_i} H_j^{-1}. \quad (14)$$

В свою очередь  $H_j^{-1} = I_r + e_j \Pi_j^{(1)}$ , причем произведение этих матриц дает нижнюю треугольную матрицу с единичной диагональю и с ненулевыми элементами, расположенными в строках с номерами  $j = (p_i + 1, \dots, p_i + d_i)$ . Полученные таким образом нижние треугольные матрицы могут быть выражены в виде суммы единичной матрицы и элементарных матриц  $e_j \Pi_j^{(1)}$ , соответствующих ненулевым строкам:

$$\prod_{j=p_i+1}^{p_i+d_i} H_j^{-1} = I_r + \sum_{j=p_i+1}^{p_i+d_i} e_j \Pi_j^{(1)}. \quad (15)$$

Преобразуем последнее выражение следующим образом:

$$I_r + \sum_{j=p_i+1}^{p_i+d_i} e_j \Pi_j^{(1)} = (d_i + 1)I_r - \sum_{j=p_i+1}^{p_i+d_i} (I_r - e_j \Pi_j^{(1)}) = (d_i + 1)I_r - \sum_{j=p_i+1}^{p_i+d_i} H_j.$$

Объединяя полученное соотношение с (12)-(14), приходим к результату, сформулированному в утверждении:

$$z = (I_r - \Pi_1)^{-1} \Pi_2 v = \prod_{i=f}^1 \left[ (d_i + 1)I_r - \sum_{j=p_i+1}^{p_i+d_i} H_j \right] \Pi_2 v.$$

Проиллюстрируем результат, содержащийся в приведенном утверждении, на примере аналитического решения системы (10) для алгоритма, не содержащего условных переходов.

Для графа, представленного на рис. 3, векторы из системы уравнений (10) принимают вид:  $z = [z_1, z_2, z_3, z_4]^T$ ,  $v = [x_1, u_1]^T$ , а матрицы  $\Pi_1$  и  $\Pi_2$  будут следующими:

$$\Pi_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & c_1 & 0 \end{bmatrix}; \quad \Pi_2 = \begin{bmatrix} a_1 & 0 \\ 0 & b_1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Как отмечалось, данный алгоритм содержит число операций  $r=4$  и обладает конвейерностью  $f=3$ . Уровень параллелизма меняется в зависимости от вычислительного такта алгоритма следующим образом:  $d_i \in \{2; 1; 1\}$ , где  $i=1, \dots, 3$ . Следовательно, решение будет иметь вид:

$$z = \prod_{i=3}^1 \left[ (d_i + 1)I_4 - \sum_{j=p_i+1}^{p_i+d_i} H_j \right] \Pi_2 v.$$

Подставив численные значения величин в последнее выражение и выполнив необходимые вычисления, получим следующий результат:

$$z = \begin{bmatrix} a_1 & 0 \\ 0 & b_1 \\ a_1 & b_1 \\ a_1 c_1 & b_1 c_1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ u_1 \end{bmatrix}.$$

Используя полученное решение, мы можем найти аналитическую зависимость для финальной операции алгоритма —  $z_4$ , не выполняя промежуточные вычисления.

Перейдем теперь к построению модели распределения загрузки между процессорами в многопроцессорной вычислительной системе, т.е. к разработке алгоритма диспетчеризации. Обозначим через  $O = \{1, 2, \dots, r\}$  множество номеров операций, составляющих алгоритм управления. Разобьем это множество на совокупность подмножеств  $O_i = \{p_i + 1, \dots, p_i + d_i\}$  ( $i = 1, 2, \dots, f$ ) номеров операций, допускающих параллельное выполнение для  $i$ -го отсчета времени. Иначе говоря,  $O_i$  —  $i$ -я группа параллельных операций. Пусть в состав многопроцессорной ИУС входит  $s$  ( $1 \leq s \leq r$ ) процессоров. То-

гда каждая группа параллельных операций алгоритма может быть реализована данной совокупностью процессоров за  $(k_i+1)$  машинных тактов выполнения макрокоманд, если  $d_i$  не делится нацело на  $s$ , или за  $k_i$  тактов в противном случае. Здесь  $k_i = [d_i / s]$  — число полных тактов ( $[d_i / s]$  означает целую часть дроби, заключенной в квадратные скобки), на которых загруженными оказываются все  $s$  процессоров. Если  $d_i$  не делится нацело на  $s$ , то последний  $(k_i+1)$ -й такт этой последовательности временных тактов является неполным, т.е. на нем загруженными оказываются только  $(d_i - sk_i)$  процессоров.

Выделим теперь в составе подмножеств  $O_i$  совокупности номеров операций, реализуемых  $j$ -м ( $j = 1, 2, \dots, s$ ) процессором:  $O_i^j = \{l_{i,1}^j, l_{i,2}^j, \dots\}$ . В состав каждой такой совокупности входит  $(k_i+1)$  номеров, если  $j$ -й процессор оказывается загруженным на последнем неполном временном такте, и  $k_i$  номеров — в противном случае.

Допустим, что алгоритм управления, взятый в качестве примера, реализуется на 3-процессорном вычислительном комплексе ( $s=3$ ), а множества операций из  $i$ -той группы, выполняемых  $j$ -м процессором, имеют вид:

$$\begin{aligned} O_1^1 &= \{1\}, & O_1^2 &= \{2\}, & O_1^3 &= \emptyset; \\ O_2^1 &= \{3\}, & O_2^2 &= \{4, 6, 7\}, & O_2^3 &= \{5\}; \\ O_3^1 &= \{8, 12\}, & O_3^2 &= \{11\}, & O_3^3 &= \{9, 10\}; \\ O_4^1 &= \{13, 15\}, & O_4^2 &= \emptyset, & O_4^3 &= \{14\}; \\ O_5^1 &= \{17\}, & O_5^2 &= \emptyset, & O_5^3 &= \{16\}; \\ O_6^1 &= \emptyset, & O_6^2 &= \emptyset, & O_6^3 &= \{18\}; \\ O_7^1 &= \{19\}, & O_7^2 &= \emptyset, & O_7^3 &= \emptyset. \end{aligned}$$

Для оценки результатов проведенного распределения загрузки процессоров введем в рассмотрение индикаторную матрицу  $E = \text{diag} \{e_1, e_2, \dots, e_r\}$ , элементы которой  $e_i = \delta_{k,i}$  при  $i \in O_i^k$  показывают, что операция с номером  $i$ , соответствующая  $l$ -му отсчету времени, выполняется на процессоре с номером  $k$ . Умножая матрицы  $\Pi_1$  и  $\Pi_2$  на матрицу  $E$ , получаем матрицы  $G_1^* = E\Pi_1$ ,  $G_2^* = E\Pi_2$ , строки которых оказываются помеченными в соответствии с номерами процессоров, выполняющих данные операции.

Поскольку структура матрицы  $G_1^*$  аналогична структуре матрицы  $\Pi_1$ , то формула, определяющая аналитическое решение, сохраняется при замене матриц  $\Pi_1$  и  $\Pi_2$  на  $G_1^*$  и  $G_2^*$ . Вектор результатов операций  $z$ , вычисленный с использованием матриц  $G_1^*$  и  $G_2^*$ , будет зависеть от числа процессорных блоков ИУС и от структуры связей между ними, т.е. от архитектуры ИУС. Выделение в составе вектора  $z$  компонентов, определяющих текущее состояние вычислительного процесса, позволяет сформировать систему уравнений состояния управляющей части, от-

ражающую характер распределения вычислительных операций между отдельными процессорами

$$\begin{aligned} x(k+1) &= A(G_1^*; G_2^*)x(k) + B(G_1^*; G_2^*)u(k); \\ y(k+1) &= C(G_1^*; G_2^*)x(k+1). \end{aligned} \quad (16)$$

Поясним сказанное на примере. Пусть алгоритм управления описывается следующей совокупностью разностных уравнений

$$\begin{aligned} x_1(k+1) &= a_{11}x_1(k) + a_{12}x_2(k) + b_{11}u_1(k) + b_{12}u_2(k); \\ x_2(k+1) &= a_{21}x_1(k) + a_{22}x_2(k) + b_{21}u_1(k) + b_{22}u_2(k); \\ y_1(k+1) &= c_{11}x_1(k+1) + c_{12}x_2(k+1); \\ y_2(k+1) &= c_{21}x_1(k+1) + c_{22}x_2(k+1). \end{aligned}$$

Множество номеров операций соответствующего вычислительного процесса разбивается на следующие подмножества  $O_1 = \{1, 2, 3, 4\}$ ,  $O_2 = \{5, 6, 7, 8, 9, 10\}$ ,  $O_3 = \{11, 12\}$ ,  $O_4 = \{13, 14\}$ ,  $O_5 = \{15, 16\}$ ,  $O_6 = \{17, 18\}$ ,  $O_7 = \{19, 20\}$ . Пусть полученные подмножества операций необходимо распределить между двумя процессорами. Тогда первое подмножество операций может быть выполнено за два такта работы каждого из процессоров, второе — за три, а все остальные — за один такт. В соответствии с этим распределим операции между процессорами таким образом  $O_1^1 = \{1, 3\}$ ,  $O_1^2 = \{2, 4\}$ ,  $O_2^1 = \{5, 7, 9\}$ ,  $O_2^2 = \{6, 8, 10\}$ ,  $O_3^1 = \{11\}$ ,  $O_3^2 = \{12\}$ ,  $O_4^1 = \{13\}$ ,  $O_4^2 = \{14\}$ ,  $O_5^1 = \{15\}$ ,  $O_5^2 = \{16\}$ ,  $O_6^1 = \{17\}$ ,  $O_6^2 = \{18\}$ ,  $O_7^1 = \{19\}$ ,  $O_7^2 = \{20\}$ . Индикаторная матрица, построенная для данного распределения, будет иметь вид

$$E = \text{diag} \{ \delta_{11}, \delta_{21}, \delta_{11}, \delta_{21}, \delta_{12}, \delta_{22}, \delta_{12}, \delta_{22}, \delta_{12}, \delta_{22}, \delta_{13}, \delta_{23}, \delta_{14}, \delta_{24}, \delta_{15}, \delta_{25}, \delta_{16}, \delta_{26}, \delta_{17}, \delta_{27} \}.$$

Вычислим вектор  $z$  и выберем его 13-ю и 14-ю компоненты, соответствующие величинам  $x_1(k+1)$  и  $x_2(k+1)$ , а также 19-ю и 20-ю компоненты, соответствующие  $y_1(k+1)$  и  $y_2(k+1)$ :

$$\begin{aligned} x_1(k+1) &= \delta_{11}\delta_{12}\delta_{14}a_{11}x_1(k) + \delta_{22}\delta_{23}\delta_{14}a_{12}x_2(k) + \\ &+ \delta_{11}\delta_{12}\delta_{14}b_{11}u_1(k) + \delta_{22}\delta_{23}\delta_{14}b_{12}u_2(k); \\ x_2(k+1) &= \delta_{12}\delta_{13}\delta_{24}a_{21}x_1(k) + \delta_{21}\delta_{22}\delta_{24}a_{22}x_2(k) + \\ &+ \delta_{12}\delta_{13}\delta_{24}b_{21}u_1(k) + \delta_{21}\delta_{22}\delta_{24}b_{22}u_2(k); \\ y_1(k+1) &= \delta_{15}\delta_{17}c_{11}x_1(k+1) + \delta_{26}\delta_{17}c_{12}x_2(k+1); \\ y_2(k+1) &= \delta_{16}\delta_{27}c_{21}x_1(k+1) + \delta_{25}\delta_{27}c_{22}x_2(k+1). \end{aligned}$$

Подобное представление алгоритма с учетом распределения вычислительных операций между процессорами позволяет сформировать процедуру совмещенного синтеза алгоритма управления с планированием и организацией параллельных вычислительных процессов. Результатом такой процедуры станет не только обеспечение желаемого качества управления, но и задание таких показателей эффек-

тивности вычислительного процесса, как длительность  $T_n$  исполнения алгоритма, выраженная в тактах макрокоманд, и средняя загрузка процессоров  $\omega$

## ЗАКЛЮЧЕНИЕ

Предложенный в работе подход позволяет решать задачу комплексной проработки всей совокупности систем бортового оборудования на платформе массово-параллельных вычислительных систем. В результате появляется возможность для значительного увеличения производительности бортового вычислительного комплекса за счет объединения мощностей процессоров вычислительных систем самолетовождения, управления полетом и тягой, предупреждения критических режимов, устойчивости и управляемости, а также других подобных систем в рамках организации единой вычислительной структуры с параллельной архитектурой.

Параллельность архитектуры вычислительного комплекса предполагает, что принципы его функционирования базируются на параллелизме исполняемых алгоритмов, программ, структуры, данных, маршрутов передачи информации. В связи с этим разработка ИУС с параллельной архитектурой должна осуществляться в рамках единой процедуры, обеспечивающей направленное формирование облика ИУС с заданным набором свойств. Основу такой процедуры составляет модель распределения фрагментов алгоритма управления между отдельными вычислительными блоками многопроцессорного вычислительного комплекса, поскольку результаты распределения определяют важнейшие показатели эффективности вычислительного процесса, в том числе длительность исполнения алгоритма, выраженную в тактах макрокоманд, и среднюю загрузку процессоров. Тем самым, возникает возможность для выбора оптимального варианта распараллеливания всех элементов архитектуры вычислительного комплекса.

## СПИСОК ЛИТЕРАТУРЫ

1. Лазарев, И. А. Композиционное проектирование сложных агрегативных систем / И. А. Лазарев. М.: Радио и связь, 1986. 311 с.
2. Михалевич, В. С. Вычислительные методы исследования и проектирования сложных систем / В. С. Михалевич, В. Л. Волкович. М.: Наука, 1982. 286 с.
3. Воеводин, В. В. Матрицы и вычисления / В. В. Воеводин. М.: Наука, 1984. 318 с.