Scientific Journal of Riga Technical University
Computer Science. Boundary Field Problems and Computer Simulation
2010
Volume 45

Principles of Creating Parallel Algorithms for Solving Identification Problems

Genady Burov, *Riga Technical University*

***Abstract* – In identification problems, it is necessary to process large amounts of information about transient processes in short time intervals. The increase in identification speed can be achieved due to application of parallel algorithms. Developed symbolical address models for allocation and rerouting of information allow representing parallel computing algorithms in a formalized mathematical form. They possess recursive and regular properties, which facilitates creation of software for solving identification problems. Address models are described in a symbolical form with the use of address lexicographic combinatory configurations formed in conditional addresses. Due to this fact, a connection between the structure of computing algorithms and the architecture of address models for information allocation in algorithms is established. Such approach allows to preserve the parameters of identified object and the parameters of dynamic test modes in address models in a symbolical form. It can have practical importance for imitation modelling of identification algorithms. It allows studying the regular properties of algorithms which in practice can be violated and are a reason of reception of unreliable results. The developed address models can be used for solving difference equation systems of identification and for derivation of their solutions in an analytical form. On the basis of such models, a number of important theoretical results, which are relevant for practical application of identification algorithms, have been obtained.**

***Keywords* – symbolical combinatory model, parallel algorithm, graphs, commutation algorithm, address graph models**

I. Problem Statement

The necessity to process large amounts of information in real-time mode of technical object functioning requires creation of parallel identification algorithms. Such problem arises, for example, at the flight test stage of space vehicles where onboard computers should process large amounts of results of measurements during the flight [2].

Lack of formalized mathematical methods for description of parallel algorithms complicates creation of software for their realization. As a result, the architecture of created parallel algorithms each time must be coordinated with the specific features of the problem being solved. For this reason, there are difficulties in optimization of software.

In this paper, it is offered to solve such problem on the basis of symbolical combinatory address models describing separate fragments of computing algorithms. In these models, it is expedient to apply special commutation algorithms that allow to carry out assembly of such fragments. They should provide allocation of information about the solved problem in the address models. Such models were developed by the author since 2003 and were further improved [5]–[9]. With their help, a number of proofs in the theory of identification of dynamic objects have been achieved. In particular, the analysis of methodical structural errors which lead to numerical results which have no reasonable physical interpretation and cannot be decoded has been made using these models.

Creation of software for solving problems of identification of technical object characteristics should be connected with the purpose of increasing the performance and noise stability of computing algorithms. The basic stages of their realization are based on operations with arrays of information, first of all, with operations with matrices of systems of difference equations formed from the results of measurements of transient processes. In this case, it is necessary to solve problems of allocation of such information in forms which would be coordinated with the character of matrix operations. In [3], [4] symbolic combinatory address models for operations with associative matrices of systems of difference equations for realization of such purposes have been developed. Symbolic combinatory address model is formed with the help of the generating operator that distributes local addresses of the set in which the information about the solved problem is placed.

II. Formation of Positional Address Models on the Basis of Numerical Sequences

On the basis of positional address models, software for realization of parallel computing algorithms can be created. In these models, arithmetic operators with application of operations of monitoring [1], [4] can be used for model transformation to reduce their complexity and to reroute the calculation results. In some cases, realization of arithmetic operations can be made directly in the space of address models. At this stage, reduction of computing complexity of algorithms for solving identification problems can be achieved in address space due to application of methods of unifying group specification of local addresses. On the basis of hierarchical address models, it is possible to realize methods for stage-by-stage reduction of complexity of algorithms, for tracking the course of computing process and for taking measures for preventing singular situations.

Solving the above listed problems is possible if architectures of address models and computing algorithms are coordinated. One of such measures is synchronization of positions of processed information fragments in algorithm and in address model of its allocation. The principle of positional allocation can be realized due to application of operations of lexicographic multiplication of subsets of local addresses used in the model [5], [7], [8]. The structure of address model can be analyzed on the basis of Kronecker lexicographic product of groups of local addresses:

$$D\left[A^{(n)}\right] \Rightarrow A_1^{(n_1)} \times\circ A_2^{(n_2)} \times\circ \cdots \times\circ A_m^{(n_m)} \qquad (1)$$

Here lexicographic factors are obtained by partitioning the initial set into parts:

$$A = \bigcup_{i=1}^{m} A_i^{(n_i)}; \quad \sum_{i=1}^{m} n_i = n \qquad (2)$$

The positional principle is realized because positions of elements of subsets are fixed and are coordinated with the positions of factors in the product (1). Let's consider the case when factors have identical dimensions and can be represented by vectors formed from elements of an interval of natural number sequence:

$$D(n,m) \Rightarrow (\overline{1.n}) \times\circ (\overline{1.n}) \times\circ ... \times\circ (\overline{1.n}) \qquad (3)$$

Vectors can be incorporated into a matrix, the structure of which is coordinated with the structure of two-dimensional array in which the real information is stored. Therefore, the structure of address model (1) can be linked with address model for allocation of elements of Cartesian power of sets:

$$D(n,m) \Rightarrow (1.n)^m; \quad N \Rightarrow \left|D(n,m)\right| = n^m \qquad (4)$$

The method of unifying specification, on the basis of which classification of components (4) can be realized, can be applied to this expression. It allows to express components (4) in a symbolical form as components of ordered numerical sequences $Sec(n,m)$ [5], [6], [7]:

$$\left[U^{(q)}\right]_j \Rightarrow \left(u_{j1}; u_{j2}; \cdots u_{jm}\right); \quad u_{jk} > u_{j,k-1} \qquad (5)$$

Let's introduce the notation of generating operator, the arguments of which should be determined by parameters of structure (4):

$$U^{(q)} \Rightarrow \psi\, NumSec(m) * (\overline{1.n})$$
$$q = \left[(n-m+1)\cdot(n-m+2)\cdots n\right] / (n-m)! \qquad (6)$$

This set is the carrier on the basis of which the numerical sequence is realized:

$$\psi\, NumSec(m) * (\overline{1.n}) \Rightarrow Sec(n,m) \qquad (7)$$

With the help of subtracting fixed components from (7)

$$\overline{\theta}^{(m)} = \overline{0.(m-1)} \qquad (8)$$

symbolical sequence (7) can be transformed into a numerical sequence and arithmetic operations can be made with its components:

$$\psi\, NumSec(m) * (\overline{1.n})[-]\overline{\theta}^{(m)} \Rightarrow Ich(n,m) \qquad (9)$$

Its components possess the following property:

$$\left[Ich(n,m)\right]_j \Rightarrow \left(c_{j1}; c_{j2}; \cdots c_{jm}\right)$$
$$\left(c_{jk} > c_{j,k-1}\right) \wedge \left(c_{jk} = c_{j,k-1}\right) \qquad (10)$$

The specification of its components can be expressed using notation of direct lexicographic product:

$$\left[Ich\ (n,m)\right]_j \Rightarrow \overline{s}_j \otimes\circ \overline{r}_j \qquad (11)$$

Here, the designation of operation of allocation of elements of one set relative to the elements of another set can be used. In this case, the elements of specification vector $\overline{r}_j$ specify the quantity of identical indices from the vector $\overline{s}_j$ that are placed into the chosen positions:

$$\left[Ich(n,m)\right]_j \Rightarrow \psi\ Accom(\overline{r}_j) \otimes\circ \overline{s}_j \qquad (12)$$

If vectors are incorporated into matrices, the following notation can be used:

$$\left[Ich(n,m)\right] \Rightarrow S \otimes\circ R$$
$$\left[Ich(n,m)\right] \Rightarrow \psi\, Accom(R) \otimes\circ S \qquad (13)$$

Classification depending on the parameter ν, which describes the number of nonzero elements in columns of matrix *S*, can be made relative to the components (13). Classification can be applied to numerical sequence (7) [4]–[6], so the address model for allocation of its components can be written down in the following form:

$$Ich(n,m) \Rightarrow$$
$$\Rightarrow \bigcup_{\nu=1}^{m} \left[\psi\ Accom(R_\nu) \otimes\circ \left[\psi\, NumSec(\nu) * (\overline{1.n})\right]\right] \qquad (14)$$

Here the positional principle is realized by allocation of elements of matrix *R* for various variants of positions numbers. The operator of such allocation we shall designate using the symbol $\varphi\, Perm$. Components of *R* are formed as partitions of parameter *m* into separate parts on ν positions with the help of the operator ($\varphi Part(\nu) * m$):

$$R_\nu \Rightarrow \varphi\ Perm * \left[\varphi Part(\nu) * m\right] \qquad (15)$$

Example: $Ich(4,3)$

$$S := \begin{pmatrix} 1 & 2 & 3 & 4 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 3 & 4 & 2 & 2 & 2 & 3 & 4 & 3 & 4 \\ 0 & 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 4 & 0 & 2 & 2 & 3 & 3 & 4 & 0 & 3 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 & 4 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$R := \begin{pmatrix} 3 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 3 & 1 & 1 & 1 & 1 & 1 & 3 & 1 & 1 & 3 \\ 0 & 2 & 2 & 2 & 2 & 1 & 1 & 2 & 1 & 2 & 0 & 2 & 2 & 2 & 1 & 2 & 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (16)$$

From proofs in [6]–[8] follows that the address model for (3) can be described by the expression:

$$\varphi\ Address\ \ *\ D(n,m) \Rightarrow\ Ked\ (\ n\ ;\ m) \quad (17)$$

$$Ked(\ n\ ; m) \Rightarrow$$
$$\Rightarrow \varphi\ Perm* \left\{ \bigcup_{\nu=1}^{m} \left[\psi\ \ Accom(R_{\nu}) \otimes\circ \left[\psi\ NumSec(\nu) * (\overline{1.n}) \right] \right] \right\} \quad (18)$$

Since the operator $\varphi\ Perm$ is linear:

$$Ked(\ n\ ; m) \Rightarrow$$
$$\Rightarrow \bigcup_{\nu=1}^{m} \left\{ \varphi\ Perm* \left[\psi\ \ Accom(R_{\nu}) \otimes\circ \left[\psi\ NumSec(\nu) * (\overline{1.n}) \right] \right] \right\} \quad (19)$$

Substituting here (15), we get:

$$Ked(\ n\ ; m) \Rightarrow$$
$$\Rightarrow \bigcup_{\nu=1}^{m} \left\{ \varphi\ Perm* \begin{bmatrix} \psi\ \ Accom\left[\varphi\ Perm* \left(\varphi\, Part(\nu) * m \right) \right] \otimes\circ \\ \otimes\circ \left[\psi\ NumSec(\nu) * (\overline{1.n}) \right] \end{bmatrix} \right\} \quad (20)$$

Example : $Ked(\ 3\ ; 3)$

$$\mathrm{Ked3} = \begin{pmatrix} 1 & 2 & 3 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 & 1 & 1 & 1 & 2 & 2 & 3 & 3 & 3 & 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 \\ 1 & 2 & 3 & 2 & 3 & 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 & 3 & 1 & 3 & 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 \\ 1 & 2 & 3 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \end{pmatrix}$$

III. Positional Address Models on the Basis of Graph Structures

Graph-based positional address models are formed using the algorithm for product (3):

$$\varphi\, Adres * D(n,m) \Rightarrow$$
$$\Rightarrow \bigcup_{\nu=1}^{m} \left\{ \begin{matrix} \varphi\ Perm* \\ * \begin{bmatrix} \psi\ \ Accom\left[\varphi\ Perm* \left(\varphi\, Part(\nu) * m \right) \otimes\circ \right] \\ \otimes\circ \left[\psi\ NumSec(\nu) * (\overline{1.n}) \right] \end{bmatrix} \end{matrix} \right\} \quad (21)$$

Here, $(\varphi\ Perm*)$ allocates elements of set into fixed positions of the model. In [5]–[7], it has been shown that for the address model (21) there is an equivalent description in the form of a graph. With its help, operations of allocation of vector elements in the columns of matrix H can be realized:

$$\varphi\, Perm(\bar{r}\ \ ) * \bar{q}_{\,j}^{\,(n_j)} \Rightarrow H_{\,j}\ ; \quad k_{\,j} = n_{\,j}\ ! / \left(\prod_{i=1}^{m} r_{\,i}! \right) \quad (22)$$

Graph is formed in view of the condition of absence of identical components:

$$\overline{h}_{\,i} - \overline{h}_{\,j} \neq \varnothing\ ; \quad i \neq j \quad (23)$$

To adhere to this condition, components of partition are formed from difference components of numerical sequences (7):

$$\varphi\ Perm(\bar{r}) * (\overline{1.\,m}) \Rightarrow \left\{ \left[\psi\ NumSec(r_{\,1}) * (\overline{1.m}) \right] \right\} \times\circ$$
$$\times\circ \left\{ \bigcup_{j2 \subset K2} \psi\ NumSec(r_{\,2}) * \left[(\overline{1.\,m}) \,/\, \overline{a2}_{\ j2} \right] \right\} \times\circ$$
$$\times\circ \left\{ \bigcup_{j3 \subset K3} \psi\ NumSec(r_{\,3}) * \left[(\overline{1.\,m}) \,/\, \overline{a3}_{\ j3} \right] \right\} \times\circ\cdots$$
$$\cdots\times\circ \left\{ \bigcup_{jn \subset Kn} \psi\ NumSec(r_{\,n}) * \left[(\overline{1.\,m}) \,/\, \overline{a\,n}_{\,j\,n} \right] \right\} \quad (24)$$

Parameters of numerical sequences are determined by the elements of vector of specifiers $\bar{r}$. Difference sets of type $\left[(\overline{1.\,m}) \,/\, \overline{a\,l}_{\,j\,l} \right]$ are defined in each branch of the graph. From (24) follows that the model (24) is suitable for parallel processing of information as the branches in it are independent from each other.

The parallel architecture of address models can be realized due to their decomposition into independent fragmentary address models. For this purpose, it is necessary to use corresponding methods of decomposition of numerical sequences (7) on the basis of which the partitions of graph are formed. For this purpose, the set $(\overline{1.n})$ is separated into k parts:

$$\psi\ NumSec(m) * (\overline{1.n}) \Rightarrow$$
$$\Rightarrow \psi\ NumSec(m) * \left\{ (\overline{1.n_{\,1}}) \circ (\overline{n_{\,1}{+}1.n_{\,2}}) \circ\cdots\circ (\overline{n_{\,k-1}{+}1.n_{\,k}}) \right\} \quad (25)$$

In general, such partitioning can be described as:

$$\psi\ NumSec(m) * \left\{ A_{\,1}^{\ (n_{1})} \circ A_{\,2}^{(n_{2})} \circ\cdots\circ A_{\,k}^{(n_{K})} \right\} \Rightarrow$$
$$\Rightarrow \psi\ NumSec(Q) * \left\{ \bigcup_{i=1}^{k} \circ\, A_{\,i}^{\ (n_{i})} \right\} \quad (26)$$

Here the matrix Q is generated from column vectors $\bar{q}_{\,j}$, which describe the components of decomposing the number m into parts. Numbers of rows in vectors are fixed and correspond to the numbers of positions of subsets in (26). The sum of elements in all columns is the same and equals m. Besides, the elements $\bar{q}_{\,j}$ should not exceed the dimensions of sets in (26) that are located in the same positions, that is, $q_{\,j\,i}\ (\langle \vee =) n_{\,i}$. In that case, the matrix Q should represent a set of solutions of system of conditional Diophantine equations:

$$\sum_{i=1}^{nk} q_{ji} \cdot \alpha_{ji} = m; \quad q_{ji} \le n_i \tag{27}$$

Their solution is based on filtration of subsets of integers with adherence to the following conditions:

$$\begin{aligned}(n_i \ge m) &\Rightarrow [n_i \Rightarrow (r_i = m)]\\ (n_i < m) &\Rightarrow [n_i \Rightarrow (r_i = n_i)]\end{aligned} \tag{28}$$

Components of the set

$$L \Rightarrow (\overline{1.r_1}) \times \circ (\overline{1.r_2}) \times \circ \cdots \times \circ (\overline{1.r_k}) \tag{29}$$

are grouped into row vectors of matrix *H*. Adherence to the conditions is checked by summation of elements of matrix rows:

$$\bar{h} \Rightarrow H \cdot [1 \quad 1 \quad \cdots \quad 1]^T \tag{30}$$

Filtration of rows of *H* is made on the basis of the following conditions:

$$[\bar{h}]_i = 0; \sum_{i=1}^{k} [\bar{h}]_i \ne m; H^{(N1\times K)} \Rightarrow Q^{(N2\times K)}; N2 < N1 \tag{31}$$

Elements $\bar{q}_i$ are used as parameters of operators (6) that form the address graph model [3], [4]:

$$\begin{aligned}&\psi\, NumSec(\bar{q}_i) * \{A_1^{(n_1)} \circ A_2^{(n_2)} \circ \cdots \circ A_k^{(n_K)}\} \Rightarrow\\ &\Rightarrow [\psi\, NumSec(q_{i1}) * A_1^{(n_1)}] \times \circ [\psi\, NumSec(q_{i2}) * A_1^{(n_1)}] \times \circ \cdots\\ &\cdots \times \circ [\psi\, NumSec(q_{ik}) * A_1^{(n_k)}]; \quad \sum_{j=1}^{k} q_{ij} = m\end{aligned} \tag{32}$$

For the whole set of rows of matrix *Q* we have:

$$\psi\, NumSec(m) * \left\{ \bigcup_{i=1}^{k} \circ A_i^{(n_i)} \right\} \Rightarrow \bigcup_{\bar{q}_i \in Q} G_i$$

$$G_i = \psi\, NumSec(\bar{q}_i) * \left\{ \bigcup_{i=1}^{k} \circ A_i^{(n_i)} \right\} \tag{33}$$

The number of component in each matrix G_i is equal to:

$$|G_j| = \Big(\coprod_{i\in\overline{1.k}} n_i!\Big) \left\{ \coprod_{i\in\overline{1.k}} q_{ji}! \coprod_{i\in\overline{1.k}} (n_i - q_{ji})! \right\}^{-1} \tag{34}$$

Applying the rule of sum and product to (32) and using the multivariate formula of Vandermonde's convolution [13], we find:

$$\left| \psi\, NumSec\,(m) * \left\{ \bigcup_{i=1}^{k} \circ A_i^{(n_i)} \right\} \right| =$$

$$= \sum_j \prod_{i=1}^{k} \binom{n_i}{q_{ij}} = \begin{pmatrix} \sum_{i\in\overline{1.k}} n_i \\ \sum_{i\in\overline{1.k}} q_{ji} \end{pmatrix} = \binom{n}{m} \tag{35}$$

Adherence to the conditions

$$A_i^{(n_i)} \bigcap A_r^{(n_r)} = \varnothing$$

$$\left| \psi\, NumSec(m) * \left\{ \bigcup_{i=1}^{k} \circ A_i^{(n_i)} \right\} \right| = \binom{n}{m} \tag{36}$$

proves the validity of the algorithm of partitioning (32).

Using (1) and (3) we shall find address model for allocation of information contained in the initial information matrix $I^{(n\times n)}$:

$$Adres\big(I^{(n\times n)}\big) \Rightarrow$$

$$\Rightarrow \varphi Perm * \left\{ \sum_{\nu=1}^{m} \varphi\, Accom\,(R_\nu) * [\psi\, NumSec(\nu) * (\overline{1.n})_\nu] \right\} \tag{37}$$

Elements of $I^{(n\times n)}$ in some problems can be expressed as function of positions numbers:

$$I_{rl} \Rightarrow \sum_{i=1}^{m} [\varphi\, Accom[c \Rightarrow f\,(r, l)] * q_i] \tag{38}$$

In some computing algorithms, it is advisable to use multi-level address models. In this case, coordinate functions have more complex form. They can be used to specify multiple positions which are reserved for accommodation of a block of decomposition. The architecture of decomposition is determined by argument of the generating operator that forms the address graph model.

As it shown in [3], [4], such decomposition can be used in algorithms of rerouting and commutation of the processed information. They are applied also as means of regularization of computing algorithms [2] with the purpose of increasing their noise tolerance. It can be improved, if instead of an address model with big dimension, decomposition from models of lower dimension is used.

For example, instead of 9th order address graph models (32), decomposition from 3rd order models can be used:

$$\varphi Perm\ (3,3,3) * (\overline{1.9}) \Rightarrow \left\{ \begin{array}{l} a(123) \times \left\{ \begin{array}{l} b(456) \times c\ (789) \\ b457) \times c(689) \\ \text{..........} \ \text{..........} \ \text{..........} . \\ b(789) \times c(456) \end{array} \right\} \\ \text{..........} \ \text{..........} \ \text{.........} \\ (789) \times \left\{ \begin{array}{l} b(124) \times c(356) \\ b(125) \times c(346) \\ \text{..........} \ \text{..........} \ \text{........} \\ b(456) \times c(123) \end{array} \right\} \end{array} \right\} \quad (39)$$

Here, letters designate partitions of the graph in which third order local address models are placed. The method of decomposition allows to solve ill-conditioned problems of identification with higher accuracy. The risks connected to occurrence of singular situations can be considerably reduced in this case. The number of independent graph fragments, which can be processed in parallel mode, is determined according to the formula:

$$|\varphi Gr(*)| = \prod_{j=1}^{q} \left[((n - \sum_{i=1}^{j} m_i)!m_j!)/((n - \sum_{i=1}^{j+1} m_i)!m_j!) \right] \quad (40)$$

In this case, a compressed information form can be used where the whole partition uses uniform model of switching from local addresses of components of difference sets which come out of previous partition. Its structure is determined by parameters of the matrix of specifiers *R*, (13). Simultaneously the model of switching is also a carrier for components that are lexicographically joined to the vertices of previous partition. It is also formed from components of numerical sequence.

Graph models can be used for realization of arithmetic operations which are made in its branches. For them, uniform positional models from local addresses based on expressions (3) can be used:

$$Dec^{(N_\nu \times m)} \Rightarrow \bigcup_{\nu=1}^{m} \{\psi\ Accom(R_\nu) \otimes \circ \varphi Kc(\nu) * (\overline{1.n})\} \quad (41)$$

IV. Recursive Positional Address Models

In operators of formation of address models, vectors consisting from components of decomposition of parameter *m* into a set of classes, designated by points of numerical interval, are used as arguments of generating operators. For each class, the carrier can be created as matrix $H^{(\nu \times k)}$ from *k* column vectors with dimension ν. On the basis of the principle of equal representation [6], [7], matrices of these vectors can be calculated for all classes of decomposition. Taking into account the linearity of the operator we have:

$$H^{(\nu \times k)} \Rightarrow R_\nu \Rightarrow \bigcup_k \varphi Perm(\overline{1.\nu}) * \bar{h}_k \quad (42)$$

$$R_\nu \Rightarrow \bigcup_k \varphi Perm(\overline{1.\nu}) * \underline{\bar{a}}_k \otimes \circ \bar{b}_k \Rightarrow$$
$$\Rightarrow \bigcup_k \varphi Perm(\arg = \bar{b}_k) * \underline{\bar{a}}_k \quad (43)$$

It leads to allocation of elements of repetitions above the elements of components of numerical sequence and to expansion of dimension of the set used in the ν-th class. All formed fragments are independent and the parallel principle of formation of positional address model is provided.

Recursive positional address models were examined in [4], [6], [7]. As they are used in graph address models (21), they are also advisable to be expressed as graphs.

The numerical sequence $\psi\ NumSec(m) * (\overline{1.n})$ possesses the property of recursivity that can be used for its representation in the form of a graph consisting of separate partitions. Vertices of the partition are generated according to the principle of structure (21) in the following partitions also difference sets. The algorithm of their formation is best represented in the form of a mechanism for generating sets from points of ordered numerical series. Parameters of generated set are determined by number of graph vertices and parameters of numerical sequence:

$$\psi\ NumSec(m) * (\overline{1.n}) \Rightarrow GrafKc(n,m) \Rightarrow$$
$$\Rightarrow \bigcup_{j=1}^{N} \{ \times \circ \bigcup_i \overline{k_{ji}.k_{ji+1}} \} \quad (44)$$

The set of graph partitions is characterized by vector of maximal values $d^{(m)}$ used at formation of sets in partition. The rule of formation of difference sets is defined by the expression:

$$k_j \Rightarrow \overline{(k_j + 1).d_{j+1}};\ \overline{d^{(m)}} \Rightarrow \overline{d_1.d_m};\ d_j = n - m + j \quad (45)$$

Model (44) possesses recursive properties and all conditions of its formation are observed.

Numerical sequences $Ich(n,m)$ are connected to (44) by the transformation (9). Therefore, for them, there also should be address models with recursive properties as they exist for (44).

The rule of formation of difference sets is defined by the expression:

$$Ich(n,m) \Rightarrow Graf\ Ich(n,m) \quad (46)$$

$$k_j \Rightarrow \left(\overline{k_j.d}\right)_{(j+1)};\quad d_j = d_i = d = m \quad (47)$$

Second form of this model with reverse order of elements in graph branches can be used:

$$Ich(n,m)^T \Rightarrow Graf\ Ich(n,m) \tag{48}$$

$$k_j \Rightarrow \left(\overline{k_j.1}\right)_{(j+1)};\quad d_j = d_i = d = m \tag{49}$$

Applying the operator $\varphi\Delta$ of calculation of inter-element differences in branches of graphs, the components of partitions used in model (20) can be obtained:

$$\varphi\Delta * GrafKc(n,m) \Rightarrow Graf\ Part\left\lfloor m, \left(\overline{m..n}\right)\right\rfloor \Rightarrow$$
$$\Rightarrow \varphi\ Perm * \left\lfloor \varphi Part(m) * (\overline{m.n})\right\rfloor \tag{50}$$

Here, generation is based on application of the operator of calculation of inter-element differences in components. Components of the resulting matrix can be decomposed into classes depending on values of numbers that have been decomposed into *m* parts.

However, every class can be found on the basis of a separate primary graph, in the initial vertex of which the decomposed number is placed:

$$Graf\ Part(n,m) \Rightarrow \varphi\Delta * Graf(n,m) \tag{51}$$

$$Graf\ (m,n) : k_{ji} \Rightarrow \overline{(k_{ji} - 1).(m - j)}_{(j+1)} \tag{52}$$

V. CONCLUSIONS

The developed address models for programming of computing algorithms possess high degree of formalism, and regular and recurrent properties. It allows to use them for optimization of programs, which can reduce the amount of necessary arithmetic operations. Such models allow to transform traditional computing algorithms into algorithms with parallel architecture. Depending on the character of solved problem, this architecture can be changed flexibly by using control parameters in generating operators. Parallelization of algorithms allows to increase their performance. This is an important advantage as in identification problems, it is necessary to process large amounts of information in modes of normal functioning of technical objects.

The developed address models allow to derive descriptions of computing algorithms in an analytical form. Therefore, they have been used for proving some important facts in the theory of identification of dynamic objects. Such descriptions can be used for studying the properties of identification algorithms and possibilities of their application in practice. On their basis, conditions of occurrence of singular situations in algorithms can be investigated and methods for their prevention can be developed to exclude application of unreliable estimations of object's parameters.

REFERENCES

[1] G. Burov, "Optimized programming of computing algorithms using symbolical address structures," *Scientific Journal of Riga Technical University. Series: Computer Science. Applied Computer Systems*, vol. 42. Riga: RTU Publishing House, 2010.

[2] G. Burov, "Principles of automation of identification processes of aerospace object characteristics at the flight test stage," *Scientific Proceedings of Riga Technical University. Series: Computer science. Boundary Field Problems and Computer Simulation*, vol. 41. Riga: RTU Publishing House, 2009.

[3] G. Burov, "Address computing models for tasks of identification," *Scientific Proceedings of Riga Technical University. Series: Computer Science. Technologies of Computer Control*, vol. 19. Riga: RTU Publishing House, 2004.

[4] G. Burov, "Minimization of complexity of address structures of information accommodation in identification models," *Scientific Proceedings of Riga Technical University. Series: Computer Science. Technologies of Computer Control*, vol. 19. Riga: RTU Publishing House, 2004.

[5] G. Burov, "Formation of computing algorithms on the basis of graph address structures," *Scientific Proceedings of Riga Technical University. Series: Computer Science. Applied Computer Systems*, vol. 22. Riga: RTU Publishing House, 2005.

[6] G. Burov, "Principles of formation of parallel algorithms of the information processing in dynamic objects," *Scientific Proceedings of Riga Technical University. Series: Computer Science. Technologies of Computer Control*, vol. 15. Riga: RTU Publishing House, 2003.

[7] G. Burov, "Parallel architecture of algorithms of dynamic objects identification," *Scientific Proceedings of Riga Technical University. Series: Computer Science. Technologies of Computer Control*, vol. 15. Riga: RTU Publishing House, 2003.

[8] G. Burov, "Combinatorial methods of formation of parallel algorithms of the signals processing," *Scientific Proceedings of Riga Technical University. Series: Computer Science. Boundary Field Problems and Computer Simulation*, vol. 45. Riga: RTU Publishing House, 2003.

[9] G. Burov, "Combinatorial structure of parallel algorithms of linear algebra," *Scientific Proceedings of Riga Technical University. Series: Computer Science. Boundary Field Problems and Computer Simulation*, vol. 45. Riga: RTU Publishing House, 2003.

[10] Г. Буров, „Символьные комбинаторные методы аналитических вычислений и их применение в задачах линейной алгебры", *Алгебра и дискретная математика. Прикладные вопросы информатики.* Рига: Латвийский университет, 1989.

[11] Г. Буров, „Символьное комбинаторное исчисление и его применения в задачах линейной алгебры и идентификации динамических систем," *Латвийский Математический Ежегодник*, Nr. 34. Рига: Зинатне, 1993.

[12] Р. Стенли, *Перечислительная комбинаторика*. Мир, 1990.

[13] Г. Эндрюс, *Теория разбиений*. Москва: Наука, 1982.

[14] Дж. Ортега, *Введение в параллельные и векторные методы решения линейных систем*. Москва: Мир,1991.

Genady Burov was born 1937 in St. Petersburg, Russia. He received the doctoral degree from Riga Supreme Engineering School in 1966. Currently he is a researcher at the Environment Modelling Centre of Riga Technical University.
Address: 1/4 Meza str., Riga, LV-1048, Latvia
Phone: +371 67089511
E-mail: emc@cs.rtu.lv

Genadijs Burovs. Identifikācijas uzdevumu risināšanai paredzētu paralēlo algoritmu izveides principi
Rakstā piedāvāts skaitļošanas algoritmu programmēšanu tehnisku objektu identifikācijas uzdevumos veikt simbolisko adrešu modeļu telpā. Tas ļauj optimizēt izstrādājamās programmas samazinot nepieciešamo aritmētisko operāciju skaitu un saīsinot identifikācijas laiku. Piedāvātie modeļi ir matemātiski formalizēti, tiem piemīt regulāras un rekursīvas īpašības un tiem ir paralēla arhitektūra. Tas ļauj apstrādāt lielus informācijas daudzumus reālā laikā tehnisko objektu identifikācijai to darbības normālajos režīmos. Paralēlos adrešu modeļus piedāvāts izmantot skaitļošanas algoritmu regularizācijai, kas ļauj novērst gandrīz deģenerētu situāciju, kas var rasties atsevišķos skaitļošanas posmos, negatīvo ietekmi. Paralēlās arhitektūras vadīšanai adrešu modeļos piedāvāts mainīt modeļus veidojošo ģenerējošo operatoru argumentu vektora parametrus. Uz šādu modeļu pamata iespējams iegūt analītiskas izteiksmes identifikācijas diferenču

vienādojumu sistēmu risināšanas algoritmiem, kā arī citus teorētiskus rezultātus. Tas ļauj automatizēt programmatūras izstrādes procesu ņemot vērā risināmā uzdevuma raksturu. Monitoringa metode ļauj veikt programmu imitācijas modelēšanu un pētīt skaitļošanas procesa dinamiku. Šajā gadījumā ir iespējams novērtēt algoritmu pielietojamību, noteikt radušos deģenerētu situāciju raksturu un veikt darbības to kompensēšanai.

Геннадий Буров. Принципы создания параллельных вычислительных алгоритмов для решения задач идентификации
Программирование вычислительных алгоритмов в задачах идентификации технических объектов предлагается производить в пространстве символьных адресных моделей. Это позволяет оптимизировать создаваемые программы в целях уменьшения объема арифметических операций и сокращения времени идентификации. Предложенные модели математически формализованы, обладают регулярными и рекурсивными свойствами и имеют параллельную архитектуру. Это позволяет обрабатывать большие объемы информации при идентификации характеристик технических объектов в режимах их нормальной работы в темпе реального времени Адресные параллельные модели предлагается использовать в качестве средства регуляризации вычислительных алгоритмов, что позволяет избежать негативного влияния почти вырожденных ситуаций , которые могут возникать на отдельных этапах вычислительного процесса. Для управления параллельной архитектурой адресных моделей предлагается изменять параметры вектора аргументов производящих операторов, формирующих модели. На основе таких моделей могут быть получены аналитические выражения для алгоритмов решения систем разностных уравнений идентификации и некоторые другие теоретические результаты. Это позволяет автоматизировать процесс создания программного обеспечения с учетом характера решаемой задачи. Метод мониторинга позволяет проводить имитационное моделирование программ и исследовать динамику вычислительного процесса. В этом случае имеется возможность оценить работоспособность алгоритмов, выявить характер возникающих вырожденных ситуаций и принять меры для их компенсации.