Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Application of network operator method for synthesis of optimal structure and parameters of automatic control system

A.I. Diveyev*, E.A. Sofronova**

* *Dorodnicyn Computing Centre of the Russian Academy of Sciences, 40, Vavilov st., Moscow, Russia, 119333*
(Tel: +7-905-711-44-27; e-mail: acxat@land.ru)
***Peoples' Friendship University of Russia, 6, Miklukho-Maklaya st., Moscow, Russia, 117198*
(Tel: +7-495-955-07-56; e-mail: sofronova_ea@mail.ru)

Abstract: The problem of optimal control for nonlinear systems is considered. Genetic programming is used to obtain functional dependence of control vector from problem space vector. Network operator is proposed as one of possible solutions for effective calculations. The problem of structure-parametric synthesis of satellite angular movement stabilization system is described.

1. INTRODUCTION

Method of dynamic programming introduced by Bellman is traditionally used for synthesis of optimal control system. The result of this method is differential equation in partial derivatives. The solution of the equation leads to functional dependence of control vector from problem space vector. In most cases the Bellman equation has no analytical solution. One of the famous solutions is obtained using quadratic functional, linear object and absence of restrictions on a control (DeRusso P.M. *et al.*, 1998). Then functional dependence of control from the state of the object is expressed by linear matrix transformation.

In this paper, the functional dependence search of control vector from problem space vector is realized with the help of genetic programming (GP) (Koza J.R., 1992). GP is used for the search of algorithms. Also GP is known to be successfully applied for synthesis of control systems (Koza J.R. *et al.*, 2002, Pohlheim H. and Marenbach P., 1996, Rodriguez-Vazquez K. *et al.*, 2004, Xiaofang Chen *et al.*, 2004, Keane M.A. *et al.*, 2002).

To present one of possible solutions GP uses Polish notation that is not effective in terms of calculation. Polish notation is a string of symbols that describes operations and arguments. Symbol of operation should be followed by a definite number of symbols of arguments. Some operations can be arguments for other ones as well. To calculate the equation lexical analysis is used. It helps to differ operations and arguments. In cases when the argument is an operation the recursion should be used that requires stacks. Both lexical analysis and stacks slow down calculations using Polish notations.

In this paper, an integer matrix is used as a basic item for genetic operations. The matrix defines the directed graph, which we call network operator. The matrix contains information about the structure of network operator and types of its edges and nodes. Directed graph, network operator, as well as the tree can represent any mathematical expression. If some argument is used in certain expression several times then we should add the same number of leaves to the tree. In network operator one node corresponds to one argument no matter how many times this argument is used in the expression. Moreover, integer matrix has some advantages on Polish notation. Evaluation of upper triangular matrix is a single-pass operation and neither analysis of lines nor stacks are needed.

2. PROBLEM STATEMENT

The following problem of optimal control is considered. The system of differential equations which describes the dynamics of the object is given

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \qquad (1)$$

where $\mathbf{x} = [x_1 \ldots x_n]^T$ - problem space vector, $\mathbf{u} = [u_1 \ldots u_m]^T$ - control vector, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \boldsymbol{U} \subseteq \mathbb{R}^m$, $m \le n$, $\boldsymbol{U}$ - limited set.

Given performance functional

$$J = \int_0^{t_f} f_0(\mathbf{x}(t), \mathbf{u}(t))dt \,. \qquad (2)$$

Given boundary conditions

$$\mathbf{x}(t_f) = \mathbf{x}^f = [x_1^f \ldots x_n^f]^T \,. \qquad (3)$$

Synthesize a control system in the following form

$$\mathbf{u} = \mathbf{h}(\mathbf{x}, \mathbf{c}), \qquad (4)$$

where $\mathbf{c} = [c_1 \ldots c_Q]^T$ - vector of parameters.

Equality (4) in combination with (1) should present a system of differential equations such that its solution for given initial value $\mathbf{x}(0) = \mathbf{x}^0 = [x_1^0 \ldots x_n^0]^T$ would reach the boundary conditions (3) in finite interval of time $t_f < \infty$, $\forall t \in [0, t_f]$, $\mathbf{u}(t) = \mathbf{h}(\mathbf{x}(t), \mathbf{c}) \in \boldsymbol{U}$, and minimize the performance functional (2). The desired function $\mathbf{h}(\mathbf{x},\mathbf{c})$ can be nondifferentiable and discontinuous, but it is a single-valued transformation, $\forall \mathbf{x} \in \mathbb{R}^n \ \exists \mathbf{h}(\mathbf{x},\mathbf{c}) \in \mathbb{R}^m$.

To solve the problem (1) – (4) we use algorithm that automatically gets equations in the form of $\mathbf{u} = \mathbf{h}(\mathbf{x},\mathbf{c})$.

Algorithm uses genetic selection and out of the range of equations finds (4) that satisfies (1) – (3) most of all.

We present the equations in PC memory as a special data structure that is based on description of equation in the form of directed graph or network operator.

2. NETWORK OPERATOR

Define some bounded ordered sets. Variable set is a set in which items can change their values in the computation process

$$\boldsymbol{V} = (v_1,\dots,v_P),\ v_i \in \mathbb{R}^1,\ i = \overline{1,P}. \quad (5)$$

Parameter set is a set in which items cannot change their values in the computation process

$$\boldsymbol{C} = (c_1,\dots,c_Q),\ c_i \in \mathbb{R}^1,\ i = \overline{1,Q}. \quad (6)$$

Unary operations set is a set of functions or single-valued transformations defined over a certain number set

$$\boldsymbol{O}_1 = (\rho_1(z), \rho_2(z), \dots, \rho_W(z)), \quad (7)$$

where $\rho_i(z)$: $\mathbb{R}^1 \to \mathbb{R}^1$, $\forall z \in \mathbb{R}^1$,

$\exists y \in \mathbb{R}^1 \Rightarrow y = \rho_i(z),\ i = \overline{1,W}$.

Commutative binary operations with unit element set is a set of single-valued transformations of two equal number sets in one the same number set

$$\boldsymbol{O}_2 = (\chi_1(z',z''), \chi_2(z',z''), \dots, \chi_V(z',z'')), \quad (8)$$

where $\chi_i(z',z'')$: $\mathbb{R}^1 \times \mathbb{R}^1 = \mathbb{R}^2 \to \mathbb{R}^1$,

$\forall z', z'' \in \mathbb{R}^1$, $\exists y \in \mathbb{R}^1 \Rightarrow y = \chi_i(z',z''), i = \overline{1,V}$.

$$\chi_i(z',z'') = \chi_i(z'',z'),\ i = \overline{1,V}, \quad (9)$$

$$\exists e_i \in \mathbb{R}^1 \Rightarrow \chi_i(e_i, z) = \chi_i(z, e_i) = z,\ i = \overline{1,V}. \quad (10)$$

Definition 1. Network operator is a directed graph with following properties:

0) graph should be circuit-free;
1) there should be at least one edge from the source node to any nonsource node;
2) there should be at least one edge from any nonsource node to sink node;
3) every source node corresponds to the item of variable set or parameter set;
4) every nonsource node corresponds to the item of binary operations set;
5) every edge corresponds to the item of unary operations set.

Definition 2. The correct notation of expression is notation with unary and binary operations. Binary operation is the external operation. The arguments of binary operation are unary operations or its unit element. The arguments of unary operation are binary operations, variables or parameters.

Theorem 1. The calculation of network operator for any correct notation of expression will get the same results as calculation of the expression itself.

Proof. Given a correct notation of expression. According to definition 2 only variables and parameters can be placed in the most internal parentheses. Each internal parenthesis corresponds to source node of network operator and appropriate variable or parameter.

Since internal parentheses of expression correspond to some unary operation then we can create outcoming edges from these nodes and set appropriate unary operations to these edges. At the ends of edges we place nodes that correspond to binary operations. If a binary operation has two unary operations as arguments then the node should be placed at the end of both edges. If the second argument is a unit element then we place a node with only one incoming edge that corresponds to unary operation. If a unary operation has a binary operations as its argument then the node will have an outcoming edge.

Perform the actions mentioned above for all elements in the expression and we get a directed graph. Graph is circuit free since every new binary operation corresponds to a new node

Since every following node is placed at the end of the edges there is at least one way from source node to nonsource node.

Since the edges come out of the nonsource nodes only if the binary operation that corresponds to that node is the argument for unary operation then we get a sink node for every external binary operation. This sink node has no outcoming edges.

Prove that according to steps of calculations the value in internal parentheses is calculated before the value in external ones. Assume a binary operation has two unary operations as arguments. Then according to creation of network operator it matches the node with two incoming edges. According to the second step the final result of calculation of binary operation cannot be obtained while unary operations are performed.

Suppose a unary operation has a binary operations as its argument. It corresponds to the node with one outcoming edge. The calculation of unary operation can be performed if we get the results of binary operation in the node. Removal of parentheses corresponds to deletion of edges and nodes of the graph.

Since calculation of network operator does not invert the order of removal of parentheses, the result obtained is equal to the calculation of expression. ■

Suppose mathematical expression is of the form

$$u = ax^2 + by^3 . \tag{11}$$

The sets are defined as

$$\boldsymbol{V} = (x, y), \tag{12}$$

$$\boldsymbol{C} = (a, b), \tag{13}$$

$$\boldsymbol{O}_1 = \left(\rho_1(z) = z, \rho_2(z) = z^2, \rho_3(z) = z^3\right), \tag{14}$$

$$\boldsymbol{O}_2 = \left(\chi_1(z', z'') = z' + z'', \chi_2(z', z'') = z'z''\right). \tag{15}$$

Then mathematical expression may be written as

$$u = \chi_1\Big(\chi_2\big(a, \rho_2(x)\big), \chi_2\big(b, \rho_3(y)\big)\Big). \tag{16}$$

Fig. 1 shows network operator for (16).

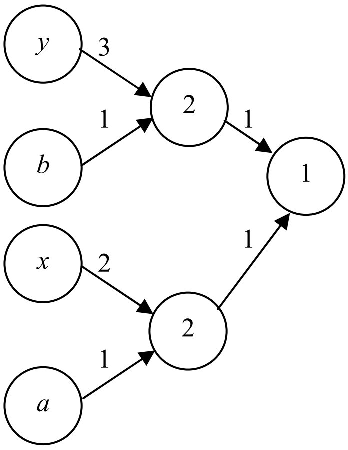

Fig. 1 Network operator for (16)

As we can see on the network operator graph variables and parameters are used in the source nodes, binary operations are used in the nonsource nodes and unary operations are presented on the edges.

Suppose mathematical expression contains noncommutative binary operation

$$u = x^y . \tag{17}$$

Then (17) may be written using additional unary operations

$$u = \exp(y \ln(x)) . \tag{18}$$

Let us add some new operations such as $\rho_4(z) = e^z$, $\rho_5(z) = \ln(z)$ to unary operations set (14). Then (18) may be written as

$$u = \rho_4\left(\chi_2\left(y, \rho_5(x)\right)\right) \tag{19}$$

If we try to create a network operator for (19) we will get an edge that corresponds to unary operation $\rho_4(z)$ and does not have an incoming node. So we will get incorrect network operator. To avoid this we have to add a binary operation $\chi_1(z', z'') = z' + z''$ and use 0 as a second argument. This action will not change the result but will guarantee the correctness of network operator. Expression (18) should be written as

$$u = \chi_1\left(\rho_4\left(\chi_2\left(\rho_1(y), \rho_5(x)\right)\right), 0\right). \tag{20}$$

It must be emphasized that the first operation in the expression must be a binary operation having unary operations as arguments. Again unary operations must have variables or/and parameters as their arguments.

Consider an example. Suppose mathematical expression contains some embedded unary operations

$$u = e^{x^2} . \tag{21}$$

Then the correct notation for (21) may be written as

$$u = \chi_1\left(\rho_4\left(\chi_1\left(\rho_2(x), 0\right)\right), 0\right) \tag{22}$$

and presented as a graph in Fig. 2. There is no need to depict unit elements for binary operations.

If a binary operation operates on more than two arguments then in mathematical expression these arguments can be grouped in pairs and in network operator this binary operation can be presented as a single node with an operation number in it.

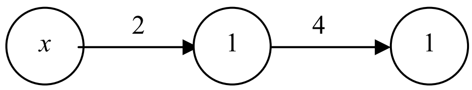

Fig. 2 Network operator for (22)

Consider an example. Suppose we have the following mathematical expression

$$u = x + y^2 + a^3 , \tag{23}$$

If we group in pairs the arguments of binary operation “sum”, we shall have the following notation

$$u = \chi_1\left(\rho_1(x), \rho_1\left(\chi_1\left(\rho_2(y), \rho_3(a)\right)\right)\right). \tag{24}$$

Fig. 3 shows two equivalent network operators for (24).

According to property (0) of network operator it is a circuit-free directed graph. This statement means that there can be several edges coming out of one node. Consider the following mathematical expression

$$u = ae^x(x + a) . \tag{25}$$

The correct notation for (25) may be written as

$$u = \chi_2\left(\rho_1(a), \rho_1\left(\chi_2\left(\rho_4(x), \rho_1\left(\chi_1\left(\rho_1(x), \rho_1(a)\right)\right)\right)\right)\right) \tag{26}$$

and presented as a graph in Fig. 4.

If we present notation (26) as a tree in terms of GP (Koza J.R., 1992), then we have to use two leaves for each of two arguments x and a .

Network operator can simultaneously include several mathematical expressions. For example we have

$$u_1 = (ax)^2 + y\,, \tag{27}$$

$$u_2 = ye^{(ax)^2}\,. \tag{28}$$

The correct notations for (27) and (28) are

$$u_1 = \chi_1\big(\rho_2\big(\chi_2(\rho_1(a),\rho_1(x))\big),\rho_1(y)\big), \tag{29}$$

$$u_2 = \chi_2\Big(\rho_1(y),\rho_4\Big(\chi_1\Big(\rho_2\big(\chi_2(\rho_1(a),\rho_1(x))\big),0\Big)\Big)\Big). \tag{30}$$

Fig. 5 shows network operator for (29) and (30).

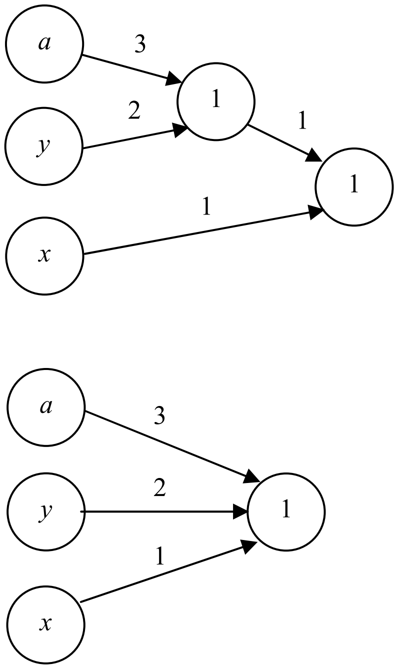

Fig. 3 Network operators for (24)

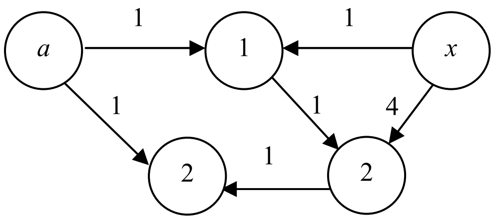

Fig. 4 Network operator for (26)

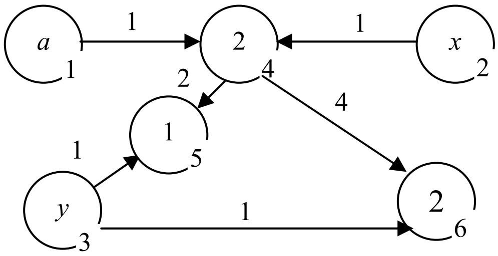

Fig. 5 Network operator for (29), (30)

The numbers of nodes are given in the bottom of each node.

3. NETWORK OPERATOR MATRIX

Since network operator is a circuit-free directed graph we can number all node so that the number of source node would be smaller then the number of incoming node. Then the incident matrix of such network operator is upper triangular.

The incident matrix consists of 0 and 1, where 1 indicates the edge between nodes, 0 indicates the absence of the edge. All unary operations are numbered. According to operation on certain argument these numbers are used instead of ones in superdiagonal elements. Binary operations are also numbered, but these numbers are used instead of zeros on the diagonal elements. So we get a network operator matrix (NOM).

Let us examine network operator shown in Fig.5. All nodes are numbered as described above. NOM for (27) and (28) is the following

$$\mathbf{\Psi} = \begin{bmatrix} 0&0&0&1&0&0\\ 0&0&0&1&0&0\\ 0&0&0&0&1&1\\ 0&0&0&2&2&4\\ 0&0&0&0&1&0\\ 0&0&0&0&0&2 \end{bmatrix}. \tag{31}$$

NOM $\mathbf{\Psi} = [\psi_{ij}]$, $i, j = \overline{1, L}$, of size $L \times L$, where L is the number of nodes in network operator.

Let us introduce the node vector

$$\mathbf{z}^{(k)} = \left[z_1^{(k)} \ldots z_L^{(k)}\right]^T, \tag{32}$$

where k is the number of iteration, $k = \overline{1, L-1}$.

If i -node is a source node then initial values of node vector elements $z_i^{(0)}$, $i = \overline{1, L}$, are set equal to appropriate elements of parameter or variable sets. For binary operations ψ_{ii} the value is set equal to unit element $e_{\psi_{ii}}$.

$$z_i^{(0)} = \begin{cases} \in \boldsymbol{V} \cup \boldsymbol{C}, \text{ if } i\text{ - node is a source node} \\ e_{\psi_{ii}}, \text{ if } i\text{ - node is a nonsource node} \end{cases}, \ i = \overline{1, L}. \tag{33}$$

To obtain an expression of matrix $\mathbf{\Psi}$ the following algorithm was worked out:

Algorithm 1 Algorithm for obtaining the expression from matrix $\mathbf{\Psi}$

Step 0. Given NOM $\mathbf{\Psi} = [\psi_{ij}]$, $i, j = \overline{1, L}$.

Step 1. Given initial values of node vector elements $z_i^{(0)}$, $i = \overline{1, L}$, according to (33). $i = 1$.

Step 2. $j = i + 1$.

Step 3. If $\psi_{ij} \neq 0$, then $z_j^{(i)} = \chi_{\psi_{jj}}\left(z_j^{(i-1)}, \rho_{\psi_{ij}}\left(z_i^{(i-1)}\right)\right)$.

Step 4. $j = j + 1$. If $j \leq L$, then go to step 4.

Step 5. $i = i+1$. If $i < L$, then go to step 2, else exit.

The elements of vector $\mathbf{z}^{(L-1)}$ that correspond to sink nodes of network operator are the results of computation.

Theorem 2. Assume given the expression in the form of correct notation. Assume network operator for the correct notation is described by NOM $\mathbf{\Psi} = [\psi_{ij}]$, $i, j = \overline{1, L}$. Then algorithm 1 guarantees the correct calculation of the expression.

Proof. To prove the theorem we should prove that algorithm calculates all operations in expression and keeps the order of parentheses.

Since matrix $\mathbf{\Psi}$ is upper triangular the numbers of unary and binary operations are in elements ψ_{ij}, $j \geq i$. Algorithm goes through rows from $i = 1$ to $i = L-1$ and columns from $j = i+1$ to $j = L$.

For nonzero element ψ_{ij} of NOM $\mathbf{\Psi}$ unary operation that corresponds to the edge (i, j) and binary operation that corresponds to node j are performed. Thus all operations will be performed but for node 1. Node 1 is a source node that is parameter or variable.

Let unary operations be arguments for some binary operation $\chi_k(\rho_m(z'), \rho_n(z''))$. According to topological sort number of the node j, that corresponds to binary operation $k = \psi_{jj}$, should be more than the numbers of nodes, whose outcoming edges go to node j. Let $j > i$, $j > l$, $m = \psi_{ij}$, and $n = \psi_{lj}$. Thus unary operations $\rho_m(z')$ and $\rho_n(z'')$ will be performed earlier than binary one.

Let binary operation be arguments for some unary operation $\rho_k(\chi_m(z', z''))$, $k = \psi_{ij}$, $m = \psi_{ii}$, $i < j$. Thus binary operation $\chi_m(z', z'')$ be performed earlier than unary one.

According to the algorithm binary operation $\chi_{\psi_{ii}}(z', z'')$ will be performed for all nonzero elements $\psi_{ki} \neq 0$ in column i and rows above i, $k < i$.

Since algorithm goes to next row only if all operations in rows above are performed. Operation $\rho_{\psi_{ij}}(\chi_m(z', z''))$ will be performed only if all unary operations $\rho_{\psi_{ki}}(\chi_{\psi_{kk}}(z', z''))$, $k < i$ are performed. Thus the algorithm 1 keeps the order of calculation for unary operations.

To sum, the algorithm 1 calculates all operations in expression and keeps the order of parentheses.□

4. SMALL VARIATIONS OF NETWORK OPERATOR

For network operator the following variations are defined:

a. replacement of unary operation on the edge;
b. replacement of binary operation in the node;
c. addition of an edge with a unary operation;
d. addition of a node with a binary operation;
e. deletion of the edge;
f. deletion of the node.

Variations (a)-(c) do not change the properties of network operator and thus do not influence on its correctness. If we perform variations (d)-(f) the properties should be taken into consideration. The deletion of an edge can occur only if there is at least one more edge that has the same source node and the sink node for deleted edge has at least one more incoming edge.

Addition of a node requires the addition of at least one incoming and one outgoing edges.

Incoming edge should outcome from the node that is before the new node and the outcoming edge should come into the next node on the way.

Deletion of the node should come together with deletion of in and outcoming edges.

All variations on the network operator can be presented as an integer variation vector that consists of four elements:

$$\mathbf{w} = [w_1 \ w_2 \ w_3 \ w_4]^T, \tag{34}$$

where w_1 - number of variation, w_2 - number of row in NOM, w_3 - number of column in NOM, w_4 - number of unary or binary operation. Element w_4 depends on w_1.

Let us consider network operator at Fig. 5, NOM (31) and two vectors $\mathbf{w}^1 = [0 \ 4 \ 6 \ 3]^T$, $\mathbf{w}^2 = [2 \ 2 \ 5 \ 3]^T$. Vector $\mathbf{w}^1$ replaces unary operation $\psi_{4,6} = 3$, $\mathbf{w}^2$ adds unary operation $\psi_{2,5} = 3$.

As a result of application of vectors $\mathbf{w}^1$ and $\mathbf{w}^2$ we get a new network operator matrix

$$\mathbf{w}^2 \circ \mathbf{w}^1 \circ \mathbf{\Psi} = \begin{bmatrix} 0&0&0&1&0&0 \\ 0&0&0&1&3&0 \\ 0&0&0&0&1&1 \\ 0&0&0&2&2&3 \\ 0&0&0&0&1&0 \\ 0&0&0&0&0&2 \end{bmatrix} \tag{35}$$

NOM (35) corresponds to new mathematical expressions

$$u_1 = (ax)^2 + y + x^3, \tag{36}$$

$$u_2 = y(ax)^3. \tag{37}$$

5. PRINCIPLE OF BASIS STRUCTURE

The search of optimal structure often faces the problem of checking its properties. The checking of structure properties complicates the algorithm and slows down the search. So we use integer matrix of special form to solve the problem.

NOM is upper triangular. It describes the graph of network operator and shows the relations of nodes and edges with elements of the sets. But not all upper triangular matrices are NOMs. For the matrix to be NOM it must has properties mentioned in definition 1.

The checking of structure properties for each matrix is time consuming. For time-saving in the problems of optimal structure search we use principle of basis structure.

Principle of basis structure. For the problem of optimal structure search we set a basis structure and define its permissible variations. To generate another structures we use permissible variations of basis structure. The search of optimal solution is done over the variation space. The ordered set of variations that transforms the basis structure to optimal structure is the solution.

In our case we set a basis structure of network operator on the assumption of common sense. Permissible variations are (*a*)-(*e*) that do not change the size of NOM.

To reduce the number of variations we can replace basis structure by the current best structure in the search process. Principle of basis structure can be effectively used in GP.

6. GENETIC ALGORITHM

To solve the problem (1) – (4) we use genetic algorithm. The set of possible solutions to a problem of the form (4) is defined by a set of network operator matrices. To generate the set of network operator matrices we use a principle of basis structure.

Basis structure is one of possible solutions of given problem (4). Ordered set of variation vectors is used as a chromosome

$$\boldsymbol{W} = \left(\mathbf{w}^1,\ldots,\mathbf{w}^M\right), \qquad (38)$$

where M is a given length of a chromosome.

The length of chromosome is one of adjustable parameters of algorithm.

Any chromosome influences basis structure and guarantees emergence of new structure

$$\Psi^l = \boldsymbol{W}^l \circ \Psi^0 = \mathbf{w}^{M,l} \circ \ldots \circ \mathbf{w}^{1,l} \circ \boldsymbol{\Psi}^0, \qquad (39)$$

where l is a number of a chromosome in population, $\boldsymbol{\Psi}^0$ is a NOM of basis structure.

First define the basis structure of possible solution $\boldsymbol{\Psi}^0$. Then generate the set of chromosomes $\boldsymbol{W}^l$, $l = \overline{1,H}$, out of variation vectors. Estimate the fitness function for each chromosome. The fitness function depends on (2) and error for boundary conditions (3)

$$F = \int_0^{t_f} f_0(\mathbf{x}(t),\mathbf{u}(t))dt + \alpha\sum_{i=1}^{n}\left(x_i^f - x(t_f)\right)^2,$$

where α - weight coefficient.

While estimating fitness function for each chromosome using nonlinear programming we find optimal parameters out of set (6).

All traditional GA operations are made on chromosomes in the process of evolution. These operations are selection, crossover, mutation and inversion. The probability of crossover depends on relation of fitness function values of chosen chromosomes to fitness of the best chromosome in population

$$\left(\xi < \frac{F^-}{F_{2i}}\right) \vee \left(\xi < \frac{F^-}{F_{2i+1}}\right),\ i = \overline{1,R},$$

where ξ - random value between 0 and 1, F^- - fitness function value for the best current chromosome in population, F_{2i}, F_{2i+1} - fitness function value for i couple of chromosomes, R - number of chosen couples.

After some number of generations called epoch we perform a digenesis with previous basis structure replaced by the best one found at last generation. To introduce a new basis structure it is necessary to save one chromosome identical to this basis structure. The number of epochs is also one of adjustable parameters.

7. PARAMETRIC OPTIMIZATION

While estimating the value of fitness function for each chromosome we perform the search of optimal parameters c_i, $i = \overline{1,Q}$ with a Hooke-Jeeves algorithm. For optimization we use fitness function as object function, $\min F(\mathbf{c})$.

We set random initial values of parameters in given restrictions $c_i^- \le c_i^0 \le c_i^+$, $i = \overline{1,Q}$. Estimate the object function $\tilde{F} = F\left(\mathbf{c}^0\right)$.

Perform the research near initial value

$$c_i^1 = \begin{cases} c_i^0 + \Delta c, \text{if } F\left(\mathbf{c}^1\right) < F\left(\mathbf{c}^0\right) \\ c_i^0 - \Delta c, \text{if } F\left(\mathbf{c}^1\right) < F\left(\mathbf{c}^0\right) \\ c_i^0, \text{otherwise} \end{cases},$$

where Δc - the step of research.

If after research the condition $F\left(\mathbf{c}^1\right) = F\left(\mathbf{c}^0\right)$ is not satisfied, then we increment the step of research. The algorithm is over, when $\Delta c < \varepsilon_c$, where ε_c is a given small value.

If $F\left(\mathbf{c}^1\right) < F\left(\mathbf{c}^0\right)$, then we perform one-dimensional search of slenderness with method of golden section in the found direction from $\mathbf{c}^0$ to $\mathbf{c}^1$ on criterion $\min_{\lambda} F\left(\mathbf{c}^2\right)$, where $\mathbf{c}^2 = (1-\lambda)\mathbf{c}^0 + \lambda\mathbf{c}^1$, λ - slenderness, $0 \le \lambda \le \lambda^+$. Then we substitute initial value $\mathbf{c}^0 = \mathbf{c}^2$ and repeat the research.

8. EXAMPLE

We have considered the problem of structure-parametric synthesis of satellite angular movement stabilization system

$$\frac{dx_1}{dt} = \frac{1}{3} x_2 x_3 + 100 u_1 , \tag{40}$$

$$\frac{dx_2}{dt} = -x_1 x_3 + 25 u_2 , \tag{41}$$

$$\frac{dx_3}{dt} = x_1 x_2 + 100 u_3 , \tag{42}$$

where x_1, x_2, x_3 - angular velocity of satellite, u_1, u_2, u_3 - control variables generated by jet engines.

The control should stabilize satellite location near the origin

$x_i(t_f) = x_i^f = 0$, $i = 1,2,3,$ and fuel consumption should be minimized

$$J = \int_0^{t_f} \sum_{i=1}^{3} |u_i(t)| dt \to \min .$$

Given initial values of variables

$x_1(0) = 200$, $x_2(0) = 30$, $x_3(0) = 40$.

Given restrictions for control

$-2 \le u_i \le 2$, $i = 1,2,3.$

Having used GA to solve the given problem we obtained the network operator. While estimating fitness function for each possible solution we found optimal parameters with a Hooke-Jeeves algorithm.

To solve the problem the following sets were used

$\boldsymbol{V} = (x_1, x_2, x_3)$, $\boldsymbol{C} = (c_1, c_2, c_3)$;

$\boldsymbol{O}_1 = (\rho_1(z), \ldots, \rho_8(z))$;

$\boldsymbol{O}_2 = (\chi_0(z', z''), \chi_1(z', z''))$,

where

$\rho_1(z) = z$,

$\rho_2(z) = \operatorname{sign}(z)\sqrt{|z|}$,

$\rho_3(z) = z^3$,

$\rho_4(z) = z^2$,

$\rho_5(z) = \begin{cases} 1, \text{if } z \ge 0 \\ 0, \text{otherwise} \end{cases}$,

$\rho_6(z) = e^z$,

$\rho_7(z) = -z$,

$\rho_8(z) = \dfrac{1 - e^{-z}}{1 + e^{-z}}$,

$\chi_0(z', z'') = z' + z''$,

$\chi_1(z', z'') = z' z''$.

One of possible solutions found is the following

$$u_i = \begin{cases} \operatorname{sign}(y_i) 2, \text{if } |y_i| \ge 2 \\ y_i, \text{otherwise} \end{cases}, \quad i = 1,2,3 ,$$

where

$$y_1 = -x_1 - x_2 + \operatorname{sign}(c_1 x_1)\sqrt{|c_1 x_1|} - c_3 x_3 + \\ + \left((-c_2 x_2)^3 - c_3 - c_1 x_1 - c_3 x_3\right)^3 + \operatorname{sign}(-c_3 x_3)\sqrt{|c_3 x_3|} ,$$

$y_2 = (-c_2 x_2)^3 + \dfrac{1 - e^{c_2 x_2}}{1 + e^{c_2 x_2}}$, $y_3 = -c_3 x_3$ with optimal parameters $c_1 = 1{,}0$, $c_2 = 0{,}0544$, $c_3 = 0{,}1539$. While parametric optimization the following restrictions were applied to parameters $0 \le c_i \le 1$, $i = 1,2,3$, step of research $\Delta c = 0.05$, maximal slenderness $\lambda^+ = 10$, and accuracy $\varepsilon_c = 0.001$.

Fitness function was of the form

$$F = \int_0^{t_f} \sum_{i=1}^{3} |u_i(t)| dt + \alpha \sum_{i=1}^{3} x_i^2(t_f) .$$

For equation obtained the value of fitness function F was equal to 3.242401. Fig. 6 shows NOM as the result performed by the program written on Delphi 7. Fig. 7 shows graphics of angular velocity of satellite with obtained control u_1, u_2, and u_3.

We used the following parameters of genetic algorithm: number of chromosomes in population – 50, number of generations – 50, number of couples – 20, number of generations in one epoch – 10, length of chromosome – 8, mutation probability – 0.5, inversion probability – 0.25, number of elitist chromosomes – 8.

$$\boldsymbol{\Psi} = \begin{bmatrix}
0&0&0&0&0&0&1&0&0&0&0&0&0&7&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&7&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&7&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&7&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&7&7&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&1&0&0&0&2&0&0\\
0&0&0&0&0&0&0&1&0&0&3&1&0&0&3&0\\
0&0&0&0&0&0&0&0&1&1&0&0&1&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&3&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&8&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&2&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{bmatrix}$$

Fig. 6 NOM for control of system (40) - (42)

To compare the results we simulated the system with linear dependence of control from state space $y_i = -c_i x_i$, $i = 1,2,3$.

The optimal parameters were $c_i = 1$, $i = 1,2,3$.

The value of fitness function F was equal to 4.05108.

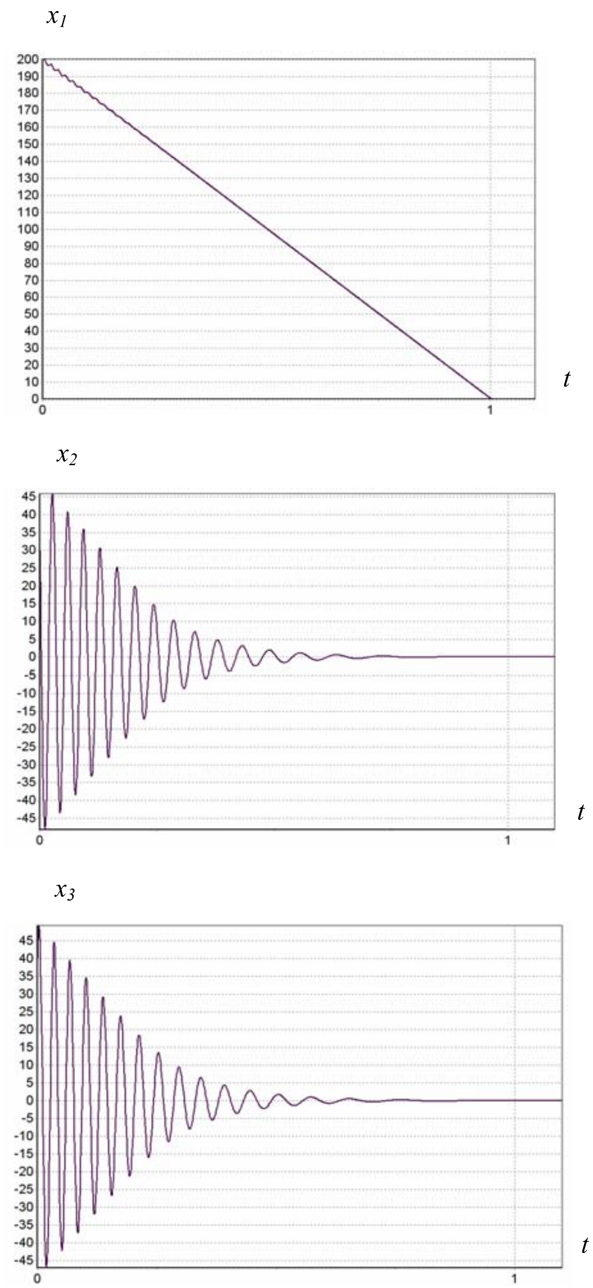

Fig. 7 Angular data of satellite with obtained control

On fig.7 we see that satellite stabilizes on coordinate x_1 in less than 1 second. Transient processes on coordinates x_2 and x_3 are oscillating and the system stabilizes in less than 0,5 seconds. The obtained control that depends on the problem space vector is robust to initial values.

CONCLUSION

A new data structure, network operator, is introduced. Network operator is an integer upper triangular matrix and is used to describe mathematical expressions. The algorithm for obtaining the expression from NOM is given and its correctness is proved. Genetic algorithm for synthesis is based on the principle of basis structure. This algorithm was applied to the problem of structure-parametric synthesis of satellite angular movement stabilization system. The system obtained is nonlinear and time independent and provides stabilization in about 1 sec.

REFERENCES

DeRusso P.M., Roy R.J., Close Ch.M. (1998) *State Variables for Engineers*. John Wiley & Sons Inc.

Koza J.R. (1992) *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, Massachusetts, London, MA: MIT Press, 819 p.

Koza J.R., Keane M.A., Streeter M.J., (2002) *Automatic synthesis using genetic programming of improved PID turning rules*, IFAC.

Pohlheim H., Marenbach P. (1996) *Generation of structured process models using genetic programming*, Workshop on evolutionary computing, Springer-Verlag.

Rodriguez-Vazquez K., Fonseca C.-M., Fleming P.-J. (2004) *Identifying the structure of nonlinear dynamic systems using multi objective genetic programming*, IEEE Transactions on Systems, Man & Cybernetics – Part A Systems & Humans, 2004, 34 (4), pp. 531

Xiaofang Chen, Weihua Gui, Yalin Wang, Lihui Cen (2004) *Multi-step optimal control of complex process: a genetic programming strategy and its application*, Engineering Applications of Artificial Intelligence, v. 17, pp. 491—500.

Keane M.A., Koza J.R., Streeter M.J., (2002), *Automatic Synthesis Using Genetic Programming of an Improved General-Purpose Controller for Industrially Representative Plants*, The 2002 {NASA/DoD} Conference on Evolvable Hardware, IEEE Computer Society, pp. 113-122.