В данном случае применительно к разработке алгоритма оптимизации рассматривается задача поиска упорядоченного множества полных путей

$$\{\Omega_0, \Omega_1, \Omega_2, ..., \Omega_k\},$$
$$L(\Omega_i) \le L(\Omega_{i+1}), \quad i \in [0, k-1]. \quad (9)$$

Система отношений между путями строится относительно полного пути Ω_0 с минимальной длиной, который может быть легко найден одним из известных алгоритмов. Такая система описывается граф-деревом G= (R, V), V = RxR, имеющим многоуровневую иерархическую структуру $R_0 = \bigcup_z R_{0z}$, где Ro – полное множество производящих контуров пути Ω_0, а R_{0z} – подмножество z-го уровня (рис. 3). Через отношение V элементы любого z-го уровня определяют элементы последующего (z+1) уровня.

Множество R_{00} элементов структуры самого верхнего (0-го) уровня полностью состоит из элементарных контуров, базирующихся по вершинам пути Ω_0, за исключением x'.

Отношение V, связывающее производящие контуры (поскольку по вершине сечения *b* базируется множество производящих контуров r_{0z}^{bi} z-го уровня, то индекс *i* служит для идентификации этих элементов во множестве) r_{0z}^{bi} и $r_{0(z+1)}^{bj}$, характерно следующими преобразованиями (рис. 4):

$$\phi(\Omega_s) = \phi(\Omega_0) + r_{0z}^{bi},$$
$$r_{0(z+1)}^{be} = r_{0z}^{bi} + r_s^e, \quad e < b, \quad (10)$$

то есть новый производящий контур $r_{0(z+1)}^{be}$ получается из производящего контура r_{0z}^{bi} предыдущего уровня, определяющего новый путь Ω_s за счет добавления к нему элементарного контура r_s^e, базирующегося по этому пути Ω_s в вершине сечения, расположенной левее вершины сечения предыдущего уровня (это условие связано с ориентацией на вершину входа x' при построении элементарных контуров – см. определение 5). Заметим, что оба контура – это производящие контуры пути Ω_0, то есть $r_{0z}^{bi}, r_{0(z+1)}^{be} \in R_0$. С другой стороны – $r_{0z}^{bi} \in R_{0z}$, $r_{0(z+1)}^{be} \in R_{0(z+1)}$.

На уровне множеств исходя из (10) отношение V можно представить с учетом того, что множество $R_{0z} = \bigcup_b \bigcup_i r_{0z}^{bi}$ производящих контуров *z*-го уровня определяет множество производящих контуров (z+1) уровня $R_{0(z+1)} = \bigcup_b \bigcup_i \bigcup_e (r_{0z}^{bi} + r_s^e)$.

Описанное структурирование отношений между путями сетевого графа позволяет получить эффективный алгоритм типа AMACONT [5], реализующий поиск полных путей, отвечающих условию (9). Этот алгоритм предназначен реализовать новый метод структурной оптимизации.

ЛИТЕРАТУРА

1. Харари Ф. Теория графов. – М.: УРСС, 2003. – 296 с.
2. Акулич И. С. Математическое программирование в примерах и задачах. – М.: Высшая школа, 2003. – 320 с.
3. Новиков Д. А. Сетевые структуры и организационные системы. – М.: ИПУ РАН, 2003. – 102 с.
4. David Lee. Coping with Discontinuities in Computer Vision: Their Detection, Classification, and Measurement// IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.12, № 4, 1990. pp 321–344.
5. Вдовин В. М., Суркова Л. Е., Валентинов В. А. Теория систем и системный анализ. – М.: Дашков и Ко, 2011. – 640 с. ●

STRUCTURAL OPTIMIZATION OF INTEGRATED AUTOMATIC CONTROL SYSTEMS

Popov, Alexander P. *– Ph.D. (Tech.), associate professor of the department of automated machining systems and tools of Moscow State University of Mechanical Engineering (MAMI), Moscow, Russia.*

ABSTRACT

A distinctive feature of the structuring of the network mathematical models is the use of available spectrum of weighting values of anti-symmetric connected graphs' nodes. Reflected mathematical objects make it possible to build an ordered structure of relations between full paths of network graph, which is described with an oriented graph- tree. And it provides, in turn, the ability to effectively solve the numerical problems on the models. Represented structuring process helps to get efficient search algorithm for full paths. This kind of algorithm contributes to the realization of a new method of a structural optimization.

ENGLISH SUMMARY

Background. *A wide range of problems of structural optimization of technical solutions is based on the use of network digraphs G= (X, U) ($U \subset X \times X$) as mathematical models, to each vertex of which $x_i \in X$ is assigned one or more numbers $\{\varepsilon(x_i)\}$, representing the optimization criteria.*

Objective. *In order to create a system of software for system structural optimization of integrated automatic control systems, it is necessary to meet the challenge of structuring appropriate mathematical models.*

Methods. *There are different methods of building structures in the ordering of graphs' nodes [1–4]. A distinctive feature of this structuring is the fact that it is built using the available spectrum of weighting values of nodes. In this respect, much attention is paid to anti-symmetric connected graphs (Pic. 1) with an attitude of strict order on the basis of the feature «there is a path from x_i to x_j », divided into layers so that:*

- *all elements of this layer do not have ancestors in the next layer;*
- *in the first and last layers there is one element (nodes of input x' and output x'' of the network);*
- *element of the last layer has no descendants;*
- *nodes of one layer are not connected together by arcs.*

Results. *Basic concepts. A free group P(U) over a generating set U of all arcs of the graph G = (X, U) is constructed as follows. As elements of P (U) is considered the set of formal linear combinations of elements of U with integral coefficients of the form*

$$p_j = \sum_{i=1}^{n} (\gamma_i \cdot u_i), \quad p_j \in P(U); \quad u_i \in U; \quad \gamma_i = 0, \pm 1,$$

where n – the number of graph's arcs.

As the binary additive operation the set P (U) is defined with the formula

$$\sum_{i=1}^{n} (\gamma_i \cdot u_i) + \sum_{i=1}^{n} (\gamma_i' \cdot u_i) = \sum_{i=1}^{n} (\gamma_i + \gamma_i') \cdot u_i .$$

Similarly, we construct a group H (X) over the set X of all graph`s nodes by defining its elements as

$$h_j = \sum_{i=1}^{m} (\gamma_i \cdot x_i), \quad h_j \in H(X); \quad x_i \in X; \quad \gamma_i = 0, \pm 1,$$

where m – the number of nodes (hereinafter when writing the elements of groups P (U) and H (X) we will omit their constituents with zero coefficients).

Definition 1. Differential d of group P (U) is a homomorphism d: P (U) -> H (X), which is defined as follows:

1) if $u_t = (x_i, x_j)$, then $du_t = x_j - x_i$;

2) if $p_q = \sum \gamma_t \cdot u_t$, then $dp_q = \sum \gamma_t \cdot du_t$, with $p_q \in P(U)$; $u_t \in U$; $x_i, x_j \in X$.

Definition 2. Element $r \in P(U)$ is p-contour (hereinafter – just contour, in the class of graphs under review there are no contours in the usual sense [2,4], which eliminates the confusion of these two concepts) if dr = 0. R = Ker d $\subset$ P(U) – is a subgroup of p-contours.

Lemma 1. Sum of several contours is a contour.

<u>Proof.</u> Firstly, the sum $\sum r_i$ is an element of group P (U) as a result of additive composition of its elements. Secondly, distributivity of imagery d, according to Definition 1, allows writing $d\sum r_i = \sum dr_i = 0$. Lemma is proved.

Using the generally accepted concept of path on a graph as a connected finite sequence of related arcs, this path is defined with formula (1).

In this respect, if $u = (x_1, x_2)$, then $x_1 = u^-, x_2 = u^+$. Length $L(\Omega)$ of path (1) will be defined in terms of mentioned numerical weighting estimates of nodes $\varepsilon(x)$ with the expression (2).

If the path connects the node of input x' with the node of output x'' of network, such a path is called full.

The imagery is defined with formula (3) and its inverse as follows:

$$\phi\left(\sum_{i=1}^{n} \gamma_i u_i\right) = \{|\gamma_1| u_1, |\gamma_2| u_2, ..., |\gamma_n| u_n\} .$$

Definition 3. The image of path $\Omega \subset U$ on a graph G = (X, U) is imagery.

Lemma 2. Differential of image of any full path on the network graph is defined as.

<u>Proof.</u> Since the full path Ω on graphs under review is a simple path, without many-fold arcs, its image in the light of (1) can be written as $\phi(\Omega) = u_1 + u_2 + ... + u_w$.

In accordance with the definition 1, we obtain $d\phi(\Omega) = du_1 + du_2 + ... + du_w$, or $d\phi(\Omega) = (u_1^+ - u_1^-) + (u_2^+ - u_2^-) + ... + (u_w^+ - u_w^-)$, or formula (4).

Incidence of arcs assumes $u_i^+ = u_{i=1}^-, i \in [1, w-1]$. Consequently, the expression (4) will be written as $d\phi(\Omega) = u_w^+ - u_1^-$ or $d\phi(\Omega) = x'' - x'$.

Lemma 3. Difference of images of two full paths on the network graph is a contour.

<u>Proof.</u> The difference of images of full paths $\phi(\Omega_i)$ and $\phi(\Omega_j)$, as the result of an additive function of composing two elements of group P (U), also belongs to this group. On the other hand, the differential of the obtained difference will be defined as $d\left[\phi(\Omega_i) - \phi(\Omega_j)\right] = d\phi(\Omega_i) - d\phi(\Omega_j)$. However, due to the assertion of Lemma 2. $d\phi(\Omega_i) = d\phi(\Omega_j)$ Consequently, $d\left[\phi(\Omega_i) - \phi(\Omega_j)\right] = 0$ that proves the assertion of lemma.

Imagery is defined with expression (5).

Definition 4. Value of contour r is value $\alpha(r)$.

Lemma 4. Difference between the lengths of two full paths on the network graph is equal to the value of contour, formed by the difference in their images.

<u>Proof.</u> Let us assume that $\Omega_i = \{u_{i1}, u_{i2}, ..., u_{iw}\}$ and $\Omega_j = \{u_{j1}, u_{j2}, ..., u_{js}\}$ are any two full paths of the graph. Then their lengths in accordance with (2) can be written as $L(\Omega_i) = \varepsilon(x') + \sum_{t=1}^{w} \varepsilon(u_{it}^+)$, $L(\Omega_j) = \varepsilon(x') + \sum_{t=1}^{s} \varepsilon(u_{jt}^+)$ and the difference of the lengths – as expression (6).

On the other hand, according to Lemma 3, the contour defined by the images of these full paths, will be written with regard to the expression (3) in the form $r = \phi(\Omega_i) - \phi(\Omega_j) = u_{i1} + u_{i2} + ... + u_{iw} - u_{j1} - u_{j2} - ... - u_{js}$, and the value of this contour with regard to (5) as expression (7).

Comparing the expressions (6) and (7), we obtain the assertion of the lemma.

Definition 5. Elementary contour r^{ab} in relation to graph's arc $u=(a,b)\in U$ is its sum with the difference between the images of the shortest paths (the shortest path is a path, having a minimum value of the length (2)), connecting nodes x' and a, as well as the nodes x' and b (Pic. 2).

Definition 6. Arc u_i is an incident one of the path Ω_0 at the node x_s, if the following conditions are met:

1) $u_i \notin \Omega_0$;

2) $\exists u_j \in \Omega_0, \quad u_i^+ = u_j^+ = x_s$. Node x_s is called the node of the Ω_0 path`s section (Pic. 2).

Definition 7. Elementary contour in relation to arc, incident to the path Ω_t at the top of path`s section is called based on the path Ω_t and is denoted as r_t^{bj} (Pic. 2) (since at b as a top may base several elementary contours, the index j is used to identify them and, if necessary, can be omitted).

Basing on Lemmas 3 and 4 ratio (8) are written, which bind certain full path Ω_0 to any other.

Contour r_{0z}^{bj} is called a generating contour of the path Ω_0 (meaning of index z and i will be revealed below).

Obviously, the set of generating contours of any full path of the network graph is a subset of the set R of all paths, and the number of elements is equal to the total number of full paths, reduced by one.

Structuring method. Previously introduced mathematical objects allow building an ordered structure of relations between full paths of the network graph, described by oriented graph-tree. This helps to effectively solve numerical problems on these models. And the kind of relationship is determined by optimization problem to be solved.

In this case, with regard to the development of an optimization algorithm the problem of the search for an ordered set of complete paths is considered. See (9).

System of relations between paths is built in relation to the full path Ω_0, having a minimum length, which can be easily found by one of the known algorithms. Such a system is described by a graph-tree G = (R, V), V = RxR, having a multi-level hierarchical structure $R_0 = \bigcup_z R_{0z}$, where Ro – complete set of generating contours of the path Ω_0, and R_{0z} is a subset of z- level (Pic. 3). Through the ratio V elements of any z-level define the elements of further (z +1) level.

R_{00} set of the structural elements of the upper (0) level consists entirely of elementary contours, based at the nodes of the path Ω_0, except for x'.

Ratio V, connecting generating contours (because at the top of section b r_{0z}^{bi} set of generating contours of z- level is based, the index i is used to identify these elements in the set) r_{0z}^{bi} and $r_{0(z+1)}^{bj}$, and is characterized by the following transformations (10). See Pic. 4.

So, a new generating contour $r_{0(z+1)}^{be}$ is obtained from the generating contour r_{0z}^{bi} of preceding level, defining a new path Ω_s by adding to it an elementary contour r_s^e, based on this path at the top of the section, which is located to the left of the section's top of the preceding level (this condition is related to the orientation to the node of input x' in the construction of elementary contours – see definition 5). Note that both contours are generating contours of the path Ω_0, i. e. $r_{0z}^{bi}, r_{0(z+1)}^{be} \in R_0$. On the other hand – $r_{0z}^{bi} \in R_{0z}, \;\; r_{0(z+1)}^{be} \in R_{0(z+1)}$.

At the sets' level on the basis of (10), the ratio V can be represented as follows. $R_{0z} = \bigcup_b \bigcup_i r_{0z}^{bi}$ – set of generating contours of z-level defines $R_{0(z+1)} = \bigcup_b \bigcup_i \bigcup_e \left(r_{0z}^{bi} + r_s^e \right)$ – set of generating circuits of (z +1) level.

Conclusion. *Described structuring of relations between paths of the network graph allows obtaining an efficient algorithm of AMACONT type [5], implementing the search for full paths corresponding to the condition (9). This algorithm is designed to implement a new method for structural optimization.*

Key words: ACS, network model, structuring, graph, digraph, graph-tree, contour, group, optimization, mathematical model, algorithm, differential.

REFERENCES

1. Harari, F. Graph Theory [*Teoriya grafov*]. Moscow, URSS publ., 2003, 296 p.

2. Akulich, I. S. Mathematical programming in examples and tasks [*Matematicheskoe programmirovanie v primerah i zadachah*]. Moscow, Vysshaya shkola publ., 2003, 320 p.

3. Novikov, D. A. Network structures and organizational systems [*Setevye struktury i organizatsionnye sistemy*]. Moscow, IPU RAN publ., 2003, 102 p.

4. David Lee. Coping with Discontinuities in Computer Vision: Their Detection, Classification, and Measurement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.12, Iss.4, 1990. pp 321–344.

5. Vdovin, V.M., Surkova, L.E., Valentinov, V. A. Systems' theory and system analysis [*Teoriya sistem i sistemnyj analiz*]. Moscow, Dashkov i Ko publ., 2011, 640 p.

Координаты автора (contact information): Попов А. П. (Popov A. P.) – pap60@bk.ru.

Статья поступила в редакцию / article received 18.12.2013
Принята к публикации / article accepted 24.02.2014