CYCLIC-MODULAR APPROACH TO DEVELOPMENT OF ELECTRONICALLY-MECHANICAL CONTROL SYSTEMS OF THE PNEUMATIC DRIVE

Aleksandr GUBAREV*, Oleg JACHNO*, Oksana GANPANTSUROVA*

* National Technical University of Ukraine "KPI", 37, Peremogy av., Kiev, 03056, Ukraine

gubarev@i.ua

Abstract: The module approach for build-up of cyclic action systems is considered. The main property of approach is system decomposition on cyclic action subsystems (modules). All modules have same structure: actuating device (drive), control device (directional valve), status sensors. Methods of approach application are realized by means of graphs. Advantage of module approach is possibility of physically heterogeneous devices combination (devices with pneumatic or hydraulic control, electro-relay or PLC) not only in all system, but also inside of each modulus, that reduced a number of signal conditioning devices. Such synthesis is possible owing to a minimal structure of each modulus.

1. INTRODUCTION

Modern production lines of an industrial automation are bound with making of heterogeneous systems. Methods of design, representative for one field, for example, pneumatics, should be matched with other technical means, including control algorithms for PLC. Good results display methods, based on logic model of system, which one is invariant to kind of realization technique in execute and control means (Ciskowski at al., 2007; Ebel and von Terzi, 2000; Schmidt, 2002; Woźniak and Jędrzejkiewicz, 2007).

2. PREMISES OF THE APPROACH

The tendered approach originally was intended for the hydraulics and the pneumatics cyclic systems for which one application of traditional methods was limited by number of drives and steps of system operation, by cost, response and asynchronous actions of the hydraulic and pneumatic apparatuses. The approach allows: propagation of reengineering problems, necessity of fast junction to issue of other product modifications, shortening of the testing and adjusting time.

2.1. Problems

- Decomposition of large systems on a projecting phase.
- Rational combination of an algorithmic, electronic, electric, pneumatic and hydraulic tool.
- Diagnostics, navigation of errors, troubleshooting.
- Build-up of systems with the open structure.

2.2. Physical premises

- The automation object consists of the fixed number of elements.
- Each element fulfills the action entering in an operating procedure multiply.
- Process of object activity can be represented in the form of operation or of a work cycle completely consisting of actions of object elements.
- Deflections from a cycle can be represented as alternative actions of some elements.

2.3. Theoretical premises

Cyclical process can be considered, as the functional form of product description (result of system operation) (Губарев, 1997).

Premise 1. Correspondence of product (purpose), process (technology of production), object (manufacturing system):

$$O(*,*,*,T) \oplus W \Leftrightarrow \exists P : \forall p \in P \Rightarrow W \oplus O \xrightarrow{G} p \qquad (1)$$

$$\forall \tau \in T \exists \Phi : O(*,*,*,\tau) \xleftrightarrow{\Phi} O(*,*,*,\tau-1)\,, \qquad (2)$$

where: W - the medium of object operation; O (E, Φ, F, T) – object; $E = \{e_1, \ldots, e_n\}$ – an object state as a set of states *n* of its elements; $\Phi = \{\varphi_1, \ldots, \varphi_m\}$ – control functions by object; $F = \{f_1, \ldots, f_m\}$ – unctions of actions (working off of control); $T = \{\tau_1, \ldots, \tau_k\}$ – the discrete time (argument of cause-effect relationship between actions); $P = \{p_1, \ldots, p_r\}$ – a product (a result of object interacting with exploitation medium);

Premise 2. The technology for production line does not vary during operation:

$$\exists O(E,\Phi,F,T) \Rightarrow \forall(\tau_i \neq \tau_j):$$
$$\begin{cases} \exists n : E(\tau_i) \equiv (F(\Phi(E(\tau_j))))^{(n)} \\ \exists m : E(\tau_i) \equiv (F(\Phi(E(\tau_i))))^{(m)} \end{cases}. \quad (3)$$

Premise 3. There is an unequivocal correspondence between set of functional elements e_i and set of segments of process φ_i:

$$\begin{cases} \forall e_i \in E \exists \varphi_j \in \Phi : e_i \leftarrow \varphi_j \\ \forall \varphi_j \in \Phi \exists e_i \in E : \varphi_j \rightarrow e_i \end{cases}. \quad (4)$$

The model of elements system is reduced to a process model from actions. Then, system as process can be decomposed to actions and logical connections between them. Each modulus (an action plus connection), in consequence of (4), simultaneously is function and building block of object:

$$\{e_i\}_n = \{(f,\varphi)_i\}_n = \{(x,y)_i\}_n, \quad (5)$$

where $X = \{x_1, \ldots, x_n\} = \{X\}_n$ – process and result of fulfilment of the actions caused by logic functions in system:

$$\left(\begin{matrix} x_i=0 \\ e_i \\ y_i=1 \end{matrix} \Big|_{\tau} \equiv 0 \xrightarrow{f_i} \begin{matrix} x_i=1 \\ e_i \\ y_i=1 \end{matrix} \Big|_{\tau+1} \equiv 1 \right) \Leftrightarrow \xrightarrow{X_i}$$

$Y = \{y_1, \ldots, y_m\} = \{Y\}_n$ – the logic functions, which connect following actions with previous, where argument are results before the fulfilled actions $y_i = y_i(\{X\}_n)$.

$$\begin{cases} \forall(\tau_j, x_i)\exists! y_i : x_i(\tau_j) \Leftarrow y_i(\{X(\tau_j - 1)\}_n) \\ \forall(\tau_j, y_i)\exists! x_i : y_i(\{X(\tau_j - 1)\}_n) \Rightarrow x_i(\tau_j) \end{cases}. \quad (6)$$

Premise 4. Each action should iterate many times because of a process repetition:

$$\forall x_i \exists(\tau_{i1} < \tau_{i2} < \tau_{i3} < \tau_{i4}) \subset \{T\}: \\ : x_i(\tau_{i1}) \neq x_i(\tau_{i2}) \neq x_i(\tau_{i3}) \neq x_i(\tau_{i4}). \quad (7)$$

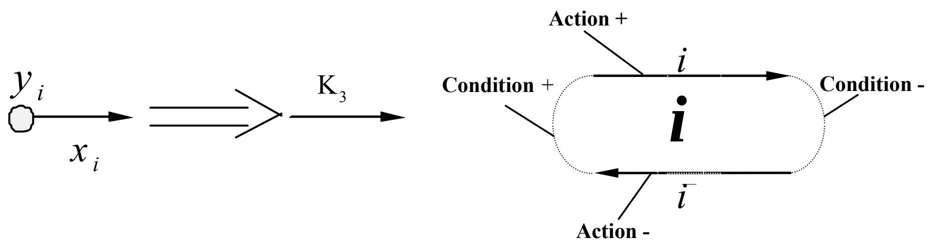
Analogously, and the reasons causing these actions, also should iterate. But, as to each action, before its repetition, the "return" action of the same modulus precedes, then should iterate and reasons of this action.

Therefore, and reasons of the modulus actions constitute couples, and mutually exclusive conditions:

$$\forall y_i \in \{Y\}_n \exists! y_j \in \{Y\}_n : \begin{cases} \overline{x_i} \Leftarrow y_j(\{X\}_n) \\ \overline{x_j} \Leftarrow y_i(\{X\}_n) \end{cases}. \quad (8)$$

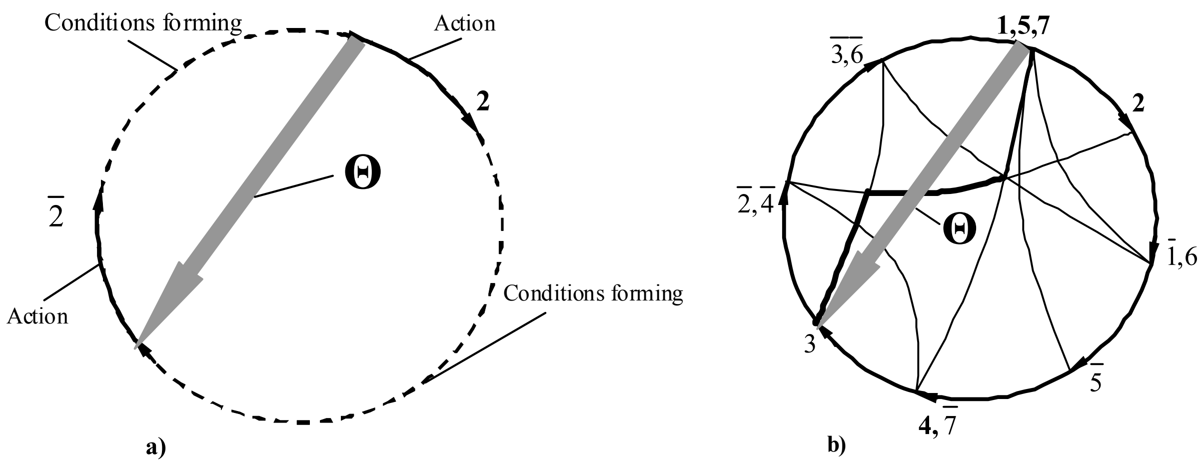
With allowance for (1) – (8):

1. The model of the cyclic system of drives is uniform (all modules have an equal constitution);
2. The model consists of cyclical modules which contain two inverse actions and two logic functions, setting a context of fulfilment of these actions.



a) b)

Fig.1. The scheme of one action (a) and of cyclical modulus for this action (b)



a) b)

Fig. 2. The cyclical sequence of operations represented in the form of a closed digraph: for one modulus (a) and for cyclic system (b)

Each action is marked out by number of the modulus and represented by an arc referred from a reason to result (Fig. 1a), where y_i – a logic function caused this action; x_i – action; i – a name of the modulus. The modulus consolidates two inverse actions and conditions of their running in to the form of a cycle (Fig. 1b).

Process of object activity consists of segments. Each segment is represented by a directional action of one modulus or simultaneous actions of several modules. The segment of process is represented by the directional arc with names of actions. Cyclical process receives form of a cycle or of the closed digraph, if there are parallel line-ups of actions. Arcs of the graph are segments $\{x_i\}$ of cyclical modules (9). Sections of conditions forming $\{y_i\}$ in cyclical modules (are marked out by a dot line) match to system operation between result of fulfilment back action and he beginning of the main action of the modulus (Fig. 2a).

Pictorially the model of system consists of a functional graph figuring an operational process:

$$G = \{P, L, \mathrm{M}\}, \tag{9}$$

where $P = \{p\}_n : \forall p_i \exists (l_{i-1}, l_i) \in L \Rightarrow l_{i-1} \to p_i \to l_i$ – set of the vertexes matching completion of the previous actions and forming of a condition of the beginning of following actions;

$L = \{l\}_m : \forall l_i \exists (p_{i-1}, p_i) \in P \Rightarrow p_{i-1} \to l_i \to p_i$ – set of the arcs matching fulfilment of actions;

$\mathrm{M} = \{\mu\}_m : \forall l_i \exists \mu_i = (k_{\mu 1}, k_{\mu 2}, \cdots) \neq \varnothing$ – et of marks for the arcs matching names of actions.

Inside of a functional graph there is a subgraph of connection which one adds actions of each modulus up to a cycle (Fig .2b).

Imitation of object activity realized by fulfilment of the graph under rules of an input, an output and traversal through arcs and vertexes. In this case, action can be considered as result of fulfilment of conditions (y_i=1), sufficient for its beginning. That is actions are an integral from conditions on a segment of one arc:

$$L(\tau) \Leftrightarrow \{x_1(\tau), x_2(\tau), \cdots, x_n(\tau)\}; \tag{10}$$

$$L'(\tau) \Leftrightarrow \{y_1(\tau), y_2(\tau), \cdots, y_n(\tau)\}; \tag{11}$$

$$L(\tau_k) = \begin{cases} * : ((L(\tau_k - 1) = 1) \wedge (L'(\tau_k) = \bar{1})) \vee \\ \vee (L(\tau_k - 1) = 0) \wedge (L'(\tau_k) = 1)) \\ L(\tau_k - 2) + \int_{\tau_k - 1}^{\tau_k} L'(\tau_k - 1) \partial\tau \end{cases}; \tag{12}$$

$$\frac{\partial L(x_i(\tau))}{\partial \tau}\Big|_{\tau_k} = \begin{cases} 1 : ((x_i(\tau_k + 1) = 1) \wedge (x_i(\tau_k) = 0)) \\ \bar{1} : ((x_i(\tau_k + 1) = 0) \wedge (x_i(\tau_k) = 1)) \\ (0 \vee y_i(\tau_k - 1)) : x_i(\tau_k) = x_i(\tau_k + 1) \end{cases}. \tag{13}$$

Analysis of systems model has displayed, that the information about finished actions is recorded in value $\{X\}_n$ in the alphabet (0, *, 1). With other words, $\{X\}_n$ is an expression of system memory, distributed on static component (value 0 and 1 – fulfilled actions) and dynamic component (value * – running actions):

$$L(\tau_i) = \int_{\tau_o}^{\tau_i} L'(\tau)\partial\tau = L(\tau)\Big|_{\tau_o}^{\tau_i - 1} + ,$$

$$+ \int_{\tau_i - 1}^{\tau_i} L'(\tau)\partial\tau = L_c(\tau_o, \tau_i) + L_v(\tau_i) \tag{14}$$

where – the static component of memory accumulated on the interval $L_C(\tau_o, \tau_i)$; $L_V(\tau_{i/o})$ – dynamic component of memory.

It is displayed, that memory distribution on $L_C(\tau_o, \tau_i)$ and $L_V(\tau_{i/o})$ depends on interval limits (τ_o, τ_i). It is established, that the availability of an interval (τ_o, τ_i) for which one the memory value is zero follows to separation of object into independent units (from τ_o up to τ_i and from τ_i up to τ_o). The informational completeness of object is mirrored with closed system measure:

$$\forall \tau_i \exists (L(\tau)\Big|_{\tau_i}^{\tau_i + 1} + L(\tau)\Big|_{\tau_i + 1}^{\tau_i}) \equiv 0 \tag{15}$$

and integrities measure:

$$\begin{cases} \forall(\tau_i, \tau_j) : \tau_i \neq \tau_j \pm 1 \Rightarrow S_{i/j} \neq 0 \\ \forall \tau_k \exists L : (L(\tau_k) = 1) \wedge (L(\tau_k + 1) = \\ = \bar{1}) \vee (L(\tau_k) = \bar{1}) \wedge (L(\tau_k + 1) = 1). \end{cases} \tag{16}$$

For the system, which satisfy these measures its development is reduced to build-up of a cyclical modules complete set. For build-up of one modulus it is necessary to realize two actions and two conditions. To the main actions there match arcs of the graph at which one transmission value x_i each time varies from 0 through * up to 1. To the back action - variation x_i^- from 0 through * up to 1. In the real system to it there matches fulfilment of productive function (for example, activity of a drive, connection of pressure).

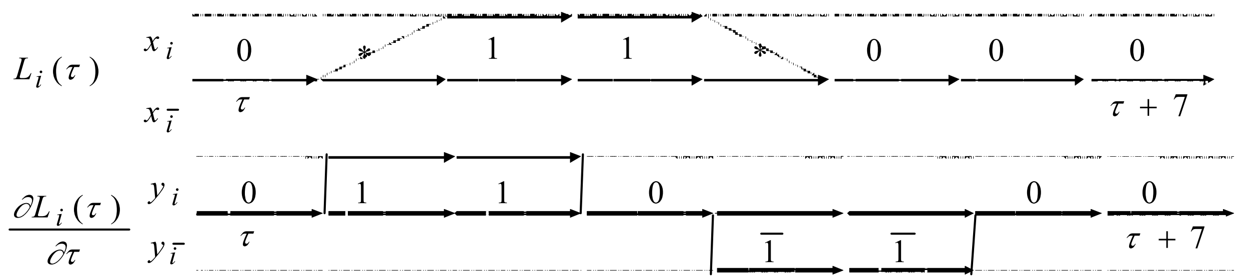


Fig. 3. The diagram of control and state for one cyclical modulus

Conditions are the vector which one closed up arcs and constituted from other actions segments (Fig. 2b). Conditions consist of several components: proximate causes, a context of action, a condition of a repetition and others. The procedure of deriving of minimum logic conditions is reduced to build-up of a closing vector:

$$y_{i(5)} \Leftarrow \bigcup_{v} \bigcup_{k=1}^{m} [\bigcap_{p}^{R1,R2} ((x_{k,1(i)} + \{x\}_{k,1c})_p * \cdots * (x_{k,n(i)} + \{x\}_{k,nc})_p * \{x\}_{k-\bar{k}})_p] * \{x\}_{k-\bar{m}}$$

$$y_{i(4)} \Leftarrow \bigcup_{k=1}^{m} (x_{k,1(i)} + \{x\}_{k,1c}) * \cdots * (x_{k,n(i)} + \{x\}_{k,nc}) * \{x\}_{k-\bar{k}} * \{x\}_{k-\bar{m}} \supset$$

$$\supset y_{i(3)} \Leftarrow \bigcup_{k=1}^{m} x_{k,1(i)} * \ldots * x_{k,n(i)} * \{x\}_{k-\bar{k}} * \{x\}_{k-\bar{m}} \supset \quad ; \quad (17)$$

$$\supset y_{i(2)} \Leftarrow \bigcup_{k=1}^{m} x_{k,(i)} * \{x\}_{k-\bar{k}} * \{x\}_{k-\bar{m}} \supset \quad ;$$

$$\supset y_{i(1)} \Leftarrow \bigcup_{k=1}^{1} x_{(i)} * \{x\}_{i-\bar{i}} \quad ,$$

where

$\bigcup_{v}$ - union of expressions of one modulus for alternate modes of system operation; $\bigcup_{k=1}^{m}$ - the total of expressions for operating modes of the modulus; $\bigcap_{p}^{R1,R2}$ - product of expressions of context reasons from alternate modes of system R1 and R2; $x_{k,\ j(i)}$ – state signal of proximate causes of k-th mode of i-th modulus; $\{x\}_{k,\ jc}$ – the logical expression enlarging a signal $x_{k,j(i)}$ before completion of action y_i; $\{x\}_{k-\bar{k}}$ – the logical expression ensuring a repetition of the modulus; $\{x\}_{k-\bar{m}}$ – the logical expression ensuring reasonableness of inverse actions of the modulus.

3. TECHNICAL PERFORMANCE OF SYSTEM

3.1. Actions of one modulus (X_i and $X_{\bar{i}}$)

In technical performance the drive is arranged with tools of controlling and the supervision of a given function fulfilment. For example, the pneumatic cylinder fulfilling a hold-down tool of a two details during its pasting together. (Fig. 4). The control device is the pneumatic directional valve. Monitoring devices - the pressure transducer (gain of a hold-down tool), position sensors (initial and working), an interval timer (an assembly time in gluing of details). The pasting together is finished if: the drive has transited the midposition sensor X'_{2S}, is not discontinued stroke of the cylinder, after achievement of gain of a pasting together drive has transited a technological time interval.

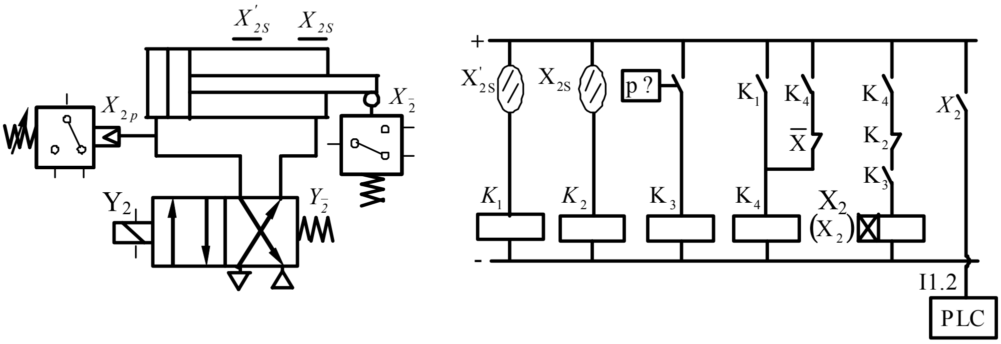


Fig. 4. Actuating part of modulus No 2 (the pneumatic cylinder fulfilling a hold-down tool of a two details during its pasting together)

2.4. Conditions of one modulus (Y_i and $Y_{\bar{i}}$)

In technical performance to conditions there match control diagrams (relay, pneumatic, hydraulic) or algorithms of PLC programs. Conditions are constituted for all modules in the minimum form (17). Procedures of drawing up of conditions allow for application of bistable and monostable valves. For example, for build-up of a logic control
command for the main action of the second modulus (Fig. 2b) the vector Θ, which one goes from the beginning of the main action ***2*** to the beginning back-action $\bar{2}$, is plotted. The vector Θ is made by segments of connection lines of modules: $7 + 2 \cdot \bar{3}$. The first condition is added by proximate causes $7 \rightarrow 7 \cdot 5 \cdot 1$, the last condition is invertible $\bar{3} \rightarrow 3$. Then we write down a logic function which one is a control command:

$$Y_2 = X_1 \cdot X_5 \cdot X_7 + X_2 \cdot \overline{X_3} .$$

This expression matches to build-up of command in case of application as the control device of a monostable valve.

Thus the drive control diagram can be realized depending on technical specifications and demands as by means of pneumatic, hydraulic tools, and by means of the program or to consolidate in it physically heterogeneous devices (Fig. 5).

3.2. The modules connecting up to a system

Process of modules connecting in to a system is fulfilled under the asynchronous circuit: each modulus provided by status signals, which dimensioned the logic conditions for its start and stop (Y_i and Y_i-), all system components provided with foreseen power supply. As well as by any alternative version of the circuit structure, engaging devices of system, a possibility of a stopping and alternative versions of control are provided. Further, each modulus operates under the logic rules set for it, and actions of modules, following these rules, constitute an operation period of system. Such method of modules connecting can be used as a tool for modernization and systems reengineering (Chrostowski et al., 2004; Ebel and von Terzi, 2000).]

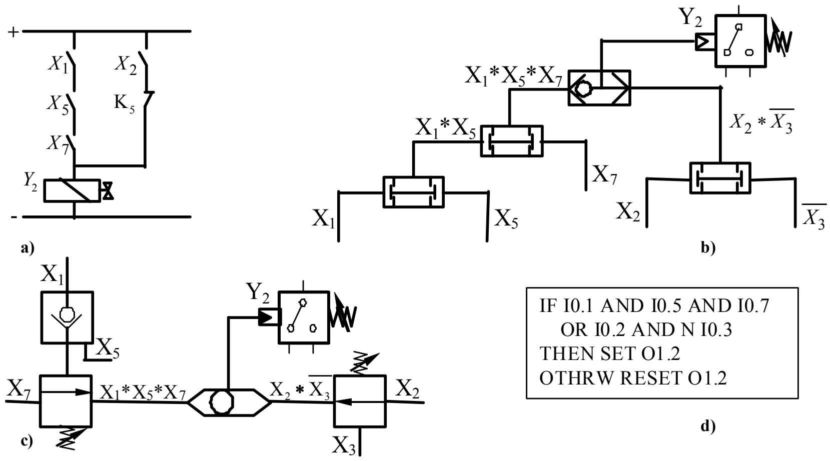


Fig. 5. Variants of realization of a conditional part of the modulus No 2: a) by means of the relay circuit; b) by a pneumatic tools; c) by hydraulic apparatuses; d) by making algorithm of the PLC program

The main segment of algorithm has view of a cyclic kernel. In this segment are displayed the complete set of modules description in the same form. For example, the STL algorithm for bistable control valve in modulus No 2 has such view:

```
IF          X1 AND  X5   AND    X7
   OR        X2  AND   N     X3
THEN  SET       Y2
         RESET  YN2
OTHRW       RESET   Y2
            SET     YN2
```

or monostable control valve:

```
IF        X1   AND   X5   AND   X7
    OR  X2   AND     N     X3
THEN        SET      Y2
OTHRW    RESET    Y2
```

The same forms for flag, timer, counter, multi-positional drive are designed. By making the algorithm this allows to have the same description method for all system components. Advantages of this method is: reduction of program list, simplicity of new modulus addition in to a system, possibility to consolidate alternative versions of system operation in one program, the program "watch" all modules all the time by a cyclic kernel.

4. SUMMARY

The cyclic-modular approach is effective at making systems with close to cyclical process of operation and quantity of actuating devices 5 and more. Efficiency of the approach is raised, if it uses at all phases - from development of the preliminary specifications, before drawing up of operating instructions, production of references on navigation and trouble-shooting. The method has formal measure for estimation of systems logic completeness, the algorithm of account of measure linearly depends on number a subsystem. The cyclic-modular constitution of systems simplifies the matching of various means in one object that get the possibility for addition the system by new modules and fulfilment of reengineering.

REFERENCES

1. **Chrostowski H., Domagala Z, Kedzia K**. (2004), Napędy i sterowania hydrauliczne w pierwszej dekadzie XXI wieku. *Hydraulika i Pneumatyka*, z. 2, 22-27.
2. **Ciskowski S., Dworzak L., Ganczarek M**. (2007), Modelowanie i programowanie metodą Grafpol pracy napędów pneumatycznych, I*nternational Scientific-Technical Conference: Hydraulics and Pneumatics*, 56-63.
3. **Ebel F., von Terzi M.** (2000), *Festo Didactic*, Mechatronics.: Festo Didactic GmbH&Co., D73770 Denkendorf.
4. **Guzowski A., Sobzyk A**. (2005), Modernizacja sterowania manipulatorów pneumatycznych, *Hydraulika i Pneumatika*, z. 1, 20-23.
5. **Schmid D.** (2002), *Mechatronika* – Polish edition REA, Warshawa.
6. **Wozniak M., Jedrzykiewicz Z.** (2007), Projektowanie układów hydraulicznych z wykorzystaniem algorytmów genetycznych, *Hydraulics and Pneumatics*, 126-132.
7. **Губарев А.П.** (1997), Дискретно-логическое управление в системах гидропневмоавтоматики: Учебное пособие.- К.: ИСМО.