UDC 62-507

M. CHERKASHENKO

SYNTHESIS OF DISCRETE DRIVES CONTROL SYSTEMS

The classical methods of creation of schemes using tables of transitions, states, Carnot's cards and other means which dimension depends on number of inputs and outputs of the scheme are not acceptable for systems big dimension (with big number of inputs, outputs and internal states). Control systems of hydropneumatic units are the determined systems of big dimension. Now at synthesis of systems of hydropneumatic units the standard position structure having the known advantages, but which is characterized by a large number of elements of the projected scheme is used. Partial minimization of standard position structure was offered in the works Yuditsky S., Goedecke W., Belforte G., Reydzo J., etc. The method of full minimization of the standard position structure offered by the author allowed to receive the minimum structure of schemes due to receiving of minimum the operation graph and the offered mathematical model – the matrix conformity (MC) which dimension does not depend on quantity of entrances and exits of system.

Keywords: the standard position structure, the matrix conformity, the operation graph, minimization, hydropneumatic units, inputs, outputs, states, synthesis of systems, schemes.

М. ЧЕРКАШЕНКО

СИНТЕЗ ДИСКРЕТНИХ СИСТЕМ УПРАВЛІННЯ ПРИВОДАМИ

Класичні методи побудови схем, які використовують таблиці переходів, станів, карти Карно і інші засоби, розмірність яких залежить від числа входів і виходів схеми, не прийнятні для систем великою розмірністю (з великою кількістю входів, виходів і внутрішніх станів). А системи управління гідропневмоагрегатів є детермінованими системами великої розмірності. В даний час при синтезі систем гідропневмоагрегатів використовується стандартна позиційна структура, яка володіє відомими перевагами, але яка характеризується великою кількістю елементів проектованої схеми. Часткова мінімізація стандартної позиційної структури була запропонована в працях Yuditsky S., Goedecke W., Belforte G., Reydzo J. та ін. Метод повної мінімізації стандартної позиційної структури, запропонований автором, дозволив отримати мінімальну структуру схем за рахунок отримання мінімального графа операцій і запропонованої математичної моделі – матриці відповідностей (МС), розмірність якої не залежить від кількості входів і виходів системи.

Ключові слова: стандартна позиційна структура, матриця відповідностей, граф операцій, мінімізація, гідропневматична одиниця, входи, виходи, стан, синтез систем, схеми.

М. ЧЕРКАШЕНКО

СИНТЕЗ ДИСКРЕТНЫХ СИСТЕМ УПРАВЛЕНИЯ ПРИВОДАМИ

Классические методы построения схем, использующие таблицы переходов, состояний, карты Карно и другие средства, размерность которых зависит от числа входов и выходов схемы, не приемлемы для систем большой размерности (с большим числом входов, выходов и внутренних состояний). А системы управления гидропневмоагрегатов являются детерминированными системами большой размерности. В настоящее время при синтезе систем гидропневмоагрегатов используется стандартная позиционная структура, обладающая известными преимуществами, но которая характеризуется большим количеством элементов проектируемой схемы. Частичная минимизация стандартной позиционной структуры была предложена в трудах Yuditsky S., Goedecke W., Belforte G., Reydzo J. и др. Метод полной минимизации стандартной позиционной структуры, предложенный автором, позволил получить минимальную структуру схем за счет получения минимального графа операций и предложенной математической модели – матрицы соответствий (МС), размерность которой не зависит от количества входов и выходов системы.

Ключевые слова: стандартная позиционная структура, матрица соответствий, граф операций, минимизация, гидропневматическая единица, входы, выходы, состояния, синтез систем, схемы.

1. Introduction. Practical experience with discrete control systems for industrial robots with hydraulic and pneumatic
controls shows that such systems are characterized by several operating cycles and a large number of inputs, outputs, and states. The effectors of an industrial robot are hydraulic or pneumatic pistons whose position is monitored by sensors, such as finite switches, sequence valves, time relays, etc. Each effector input Z_ξ and each sensor output X_ξ may take on two discrete values $\{0, 1\}$. A component U_ξ of the control system is represented by a unit corresponding to the motion of one of the effectors in space. The input Z_ξ is treated as the component input and output X_ξ as the component output.

To describe the functioning of a control system, we use the language of operation graphs [1], since it provides the most complete representation of control systems of this type. An operation graph is a directed graph whose vertices $\{A\}$ are in one-to-one correspondence with the operations of a technological process, and its arcs correspond to transitions between different operations. On the arcs of the graph the operations are marked by sequences of the form $Q_\gamma \mapsto z_v$ (i.e., by formulas of the type "if condition Q_γ is true, then condition z_v, is true"). Here the set Q_γ contains the component outputs $\{X\}$, the actions of manual control systems, etc. The set z_v moves the control system between different operations. The set z_v contains the effector inputs whose values change in the given transition. The operation graph that describes the functioning of the control system of an industrial robot consists of cycles, each corresponding to one of the operating modes.

However, in practice, we are often required to solve problems associated with the local operating mode of the control system. In this connection, we will consider a synthesis method which, while preserving the standard positional structure [1] (with unitary coding of the internal states), minimizes the control system network by minimizing the number of memory elements and in some cases also the number of logic elements. As we shall show in the sequel, this is accomplished by constructing the partition graph. The proposed synthesis method consists of the following main stages: the control system operating conditions are written in the sequence-description language, which is the machine interpretation of the language of operation graphs intended for computer-aided synthesis of control systems; the cycles in the operation graph are identified; the set of combinations $\{Q\}$ is partitioned for each cycle separately by the method described in [2-4] and a partition graph is constructed, whose vertices correspond to a minimal number of internal states of the control system; logical equations describing the control system are constructed: in this stage the signals $\{Q\}$ marked on the graph which cause transitions between operations are lengthened by adjuncting signals (from the complete output combinations $\{P\}$active in transitions) that leave the transitions unchanged [3]. This approach avoids deep minimization of the complete combinations $\{P\}$. The resulting switching functions of memory elements and effectors can be minimized by using simple absorption and join formulas for Boolean functions. The last stage is the realization of the control system by logical elements.

2. Formalized Description of the Control System. Fig. 1 is the operation graph of the control system of a simple industrial robot serving a lathe. The vertices in this graph correspond to the operations $\{A\}$, and the arcs represent transitions between different operations. The arcs are marked by the sequences $\{S\}$, with the left-hand side containing the combination Q_γ which causes transition between different operations and the right-hand side the combination z_v which consists of the signals Z_ξ and $\overline{Z}_\xi$ switching the effector on and off, respectively. Only those inputs whose values change in a given transition are marked on the arcs. The operation graph G is represented by the adjacency matrix R_G of its vertices: the rows and the columns of the adjacency matrix correspond to operations, and at the intersection of row A_i and column A_j we enter the corresponding sequence if vertices A_i and A_j are joined by an arc, and 0 otherwise.

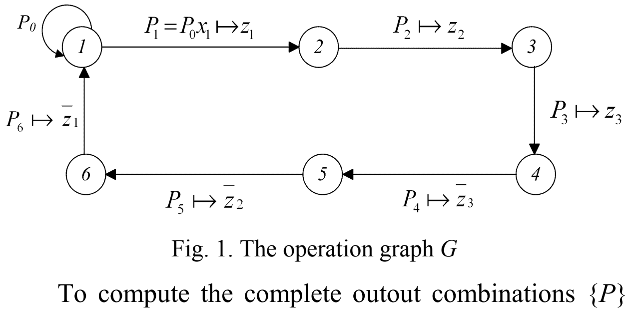

Fig. 1. The operation graph G

To compute the complete outout combinations $\{P\}$ active in transitions, we use the component matrix R_u. Each row U_ξ of the matrix R_u corresponds to one component, the first column corresponds to the initial position of the effectors of the components $\{U\}$, the remaining columns correspond to the subsequent monitored posi tions. At the intersection of row U_ξ and any column of matrix R_u we enter the corresponding output signals $\{X\}$ if the position is monitored, and 0 otherwise. If all the elements in row U_ξ to the right of element X_ξ are 0, or X_ξ is the last element of the row, the particular output monitors the final position of the effector.

The cycles in the operation graph are identified by an efficient algorithm described in [3]. This algorithm computes the determinant det (R_G) of the matrix R_G of the operation graph G, and selects from the set of terms a subset corresponding to the set of cycles of the graph G.

For the graph G in Fig. 1, we identify: $\{P_1 \mapsto z_1$, $P_2 \mapsto z_2$, $P_3 \mapsto z_3$, $P_4 \mapsto \overline{z_3}$, $P_5 \mapsto \overline{z_2}$, $P_6 \mapsto \overline{z_1}\}$.

3. Synthesis of Switching Functions for Memory Elements and Effectors. For every cycle of the operation graph, we construct a partition π of the set of combinations $\{P\}$ into blocks $\{B\}$, such that $\bigcup B_\alpha = \{P\}$ and $B_\alpha \bigcap B_\beta = \varnothing$. Identical combinations p_γ producing different input combinations z_v and z_μ should be assigned to different blocks in the partition and should not occur in adjacent blocks B_α and $B_{\alpha+1}$ (the first block is assumed to follow the last block); moreover, no combination p_γ of a successor block $B_{\alpha+1}$ should coincide with the last combination of the predecessor block B_α. A computer algorithm for the synthesis of partitions:

Synthesis Algorithm for the Partition π. The algorithm constructs a $2m_1$ matrix M (here m is the maximum number of combinations $\{P_m\}$ for one cycle), whose rows correspond to the combinations $\{P_m\}$ and the first column contains the decimal equivalents of the binary numbers corresponding to the output combinations (identical combinations p_γ producing different combinations z_v and z_μ are represented by equal numbers). The second column stores the decimal numbers corresponding to the blocks $\{B\}$ of the partition π. The algorithm consists of the following steps.

1. *i = a = 1 (i is the row index, a is the block index); go to 2.*
2. *M (i, 2) = a; go to 3.*
3. *i = i + 1, if M(i, 1)≠M(j, 1) (j takes on values such that M (j, 2) = a) go to 4; otherwise go to 5.*
4. *If M (i, 1)≠M(μ, 1) (μ is the last element of the block α -1), go to 2; otherwise go to 5.*
5. *a = a +1; go to 2.*

As noted, the last block is followed by the first block, and the procedure stops when two successive iterations produce essentially the same partition. In practice, no more than 3 iterations are needed. Stabilization of the process indicates that the particular partition is unique and minimal when the conditions formulated in the proposition are satisfied. Let us estimate the complexity of the algorithm. Assuming 3 iterations, we denote by $|B|$ the number of blocks and by m, the number of combinations $\{P_m\}$. On average, a block contains $m_1/|B|$ elements. The necessary number of comparisons within a block is $\frac{\frac{m_1}{|B|}\left(\frac{m_1}{|B|}-1\right)}{2} = \frac{m_1^2 - m_1|B|}{2|B|^2}$.

The number of comparisons with a predecessor block is $m_1/|B|$. The total complexity thus involves $3m_1(m_1+|B|)/2|B|^2$ comparisons of decimal integers. Since identical numbers never occupy adjacent positions, the complexity can be reduced by $3(m_1\text{-l})$ comparisons. We see from the formula that the complexity of the algorithm diminishes as the number of blocks increases. Storage space requirements are $2m_1$.

We establish correspondence between switching on of memory element α (in a standard positional structure, memory elements are represented by split-input flip-flops) and the last element p_σ of block $B_{\alpha-1}$ and construct a graph G_r of partitions $\{\pi\}$ for all the cycles. Every vertex of the graph G_r corresponds to a block B_α and is enclosed by a loop; the corresponding block contains more than one element. The arcs which are not loops form a single cycle of the graph G_r if the partitions $\{\pi\}$ contain the same number of blocks (with the exception of those partitions possibly containing a single block). Otherwise, the graph G_r contains several cycles.

Theorem. The partition graph G_r is realized by a standard positional structure if identical combinations p_γ producing different combinations z_v and z_μ are assigned to different nonadjacent arcs.

Proof of the Theorem. The only way to distinguish between transitions that involve identical combinations p_γ producing different combinations z_v and z_μ is by lengthening them by signals from the memory element outputs y_α and y_β ($\alpha \neq \beta; y_\alpha, y_\beta \in Y$; Y – is the set of memory element outputs). The partition π is constructed so that on the graph G_γ one internal state α (block B_α) corresponds to an arc which is not a loop and to an adjacent predecessor loop. Therefore, for two transitions A_i / A_{i+1} and A_j / A_{j+1} in the graph G, corresponding to a loop and to an adjacent successor arc in the graph G_γ, with identical combinations acting in the sequence $S_{i/i+1}(P_\gamma \mapsto z_v)$, corresponding to transition A_i / A_{i+1} and A_j / A_{j+1} in the sequence $S_{j/j+1}(P_\beta \mapsto z_\mu)$, corresponding to transition, the lengthening is $P_\gamma y_\alpha = P_\beta y_\alpha$, since $P_\gamma = P_\beta$, i.e., the input combination z_μ appears in transition A_i / A_{i+1} which precedes the transition A_j / A_{j+1} in the cycle. If identical combinations P_γ and P_β occur on adjacent arcs α and $\alpha = 1$ of the graph G_γ which are not loops, we consider analogous sequences $S_{i/i+1}$ and $S_{j/j+1}$ and obtain the lengthenings $P_\gamma y_\alpha$ and $P_\beta y_{\alpha+1}$. But since $P_\gamma y_\alpha$ corresponds to switching on the memory element $\alpha+1$, $P_\beta y_{\alpha+1}$ memory element $\alpha+2$, and $P_\gamma = P_\beta$ (recall that in standard positional structure, each predecessor memory element is switched off by the output of its successor), we obtain a "jump" from state α into state $\alpha+2$ via state $\alpha+1$, which contradicts the operating cycle of the robot (does not satisfy the stability condition [1]).

If the Identical combinations P_γ occur on the arc α and on the adjacent loop $\alpha+1$ of the graph G_γ, then in this case the lengthening $P_\gamma y_\alpha$ occurring in the sequence $S_{i/i+1}$ on the arc α which is not a loop corresponds to switching on the memory element $\alpha+1$. In the sequence $S_{j/j+1}$ the lengthening $P_\gamma y_{\alpha+1}$ belongs to the loop $\alpha+1$. But since $P_\gamma y_{\alpha+1}$ corresponds to switching on the memory element $\alpha+1$, the combination P_γ in the transition A_i / A_{i+1} of the graph G acts sequentially with signals y_α and $y_{\alpha+1}$. This implies that the input z_μ corresponding to transition A_j / A_{j+1} will prematurely appear in the transition A_j / A_{j+1}. Clearly, this allocation is possible only for $z_\mu = \overline{z_v}\left(z_\mu = z_v\right)$, i.e., when only effector(s) is (are) switched on and off in transitions A_i / A_{i+1} and A_j / A_{j+1}, respectively. Q. E. D.

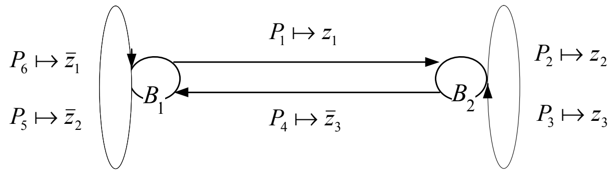

Fig. 2. Partition graph G_r

Thus, the arcs of the graph G_r which are not loops are marked with the last elements of the blocks, and the loops are marked with the ordered set of the remaining elements. The algorithm constructing the partition graph G_r may be stated in the following form:

1. Set the number of vertices of graph G_r equal to the number of subsets $|B|$ in the partition with the largest number of blocks.

2. Identify the partition blocks or equal elements of blocks, assigning them to one loop and to the adjacent successor arc of the graph G_r (if such exist).

3. Allocate the last elements of the subsets to arcs that are not loops. If one arc which is not a loop is marked with different elements (sequences), they are labeled by the number of the cycle to which they belong.

4. On the loops write (from top to bottom) the ordered subsets of the partitions (without the last elements). Different elements belonging to different cycles are labeled by the corresponding numbers.

Partitions for of the graph G:

$\pi = \{P_5 \mapsto \overline{z_2}\ **, P_6 \mapsto \overline{z_1}\ *, P_1 \mapsto z_1\}, \{P_2 \mapsto z_2\ *, P_3 \mapsto z_3\ **, P_4 \mapsto \overline{z_3}\}$.

By step 1 of the algorithm, we count the vertices of the graph Gr. By step 2, we identify identical elements of the blocks $P_6 \mapsto \overline{z_1}$, $P_2 \mapsto z_2$ and $P_3 \mapsto z_3$, $P_5 \mapsto \overline{z_2}$ assign them to one loop. By step 3, we allocate the last elements of the partition π to the arcs of the graph. By step 4, on loop 1 we write the ordered subset of identical elements of the blocks see step 2. Roman numerals label elements that do not belong to all the cycles. The resulting graph is shown in Fig. 2.

The graph G_r shows the complete combinations $P_1 - P_6$ active in the control system transitions which correspond to the subcombinations $Q_1 - Q_6$. Identical combinations are marked with asterisks.

The proof of the proposition shows that further minimization of the partition graph is possible only by breaking the standard positional structure, i.e., by using nonunitary coding for the internal states of the control system and switching one memory element more than once during an operating cycle.

Thus, graph G_r preserves the standard positional structure while minimizing the operation graph and produces a minimal partition graph satisfying the requirements of the proposition. Note that the allocation of the sequences in the graph G_r is not unique, but in any case the number of vertices remains minimal.

In order to minimize the number of conjunctions in sequences in the switching functions of memory elements and effectors, we have either to minimize the complete combinations $\{P\}$ or to lengthen the transition-activating subcombinations $\{Q\}$ by adjuncting signals from $\{P\}$ that leave the transitions unchanged. The first approach involves simultaneous minimization of a system of Boolean functions with very many variables and in practice does not yield optimal results. It is therefore better to try the second approach, lengthening the combinations $\{Q\}$. Since the obligatory lengthening of the combinations $\{Q\}$ by signals $\{Y\}$ from the memory element outputs is only necessary for identical combinations P_γ, we can estimate the complexity of the final network, remembering that the memory element and the two-input conjunction (disjunction) are realized by a single element. We distinguish between two cases: the number of vertices of the graph G_r, $|B| > 2$ and $|B| = 2$. For the first case, the complexity of the network for a graph with к cycles is

$$L_1 = |B| + k_1 + 2k_2 + 2\sum_{\zeta=1}^{m}\left(k_{3_\zeta} - 1\right) + \sum_{\alpha=1}^{|B|}\left(k_{4_\alpha} - 1\right) - \Delta s,$$

where $|B|$ – is the maximum number of blocks for one partition; k_1 – minimal number of lengthenings of the combinations $\{Q\}$. by signals from $\{P\}$ or $\{Y\}$, excluding the obligatory lengthening of the identical combinations from $\{P\}$, k_2 – number of identical output combinations: k_{3_ζ} – number of times the effector Z_ξ is switched on; m – number of effectors; k_{4_α} – number of times the memory element α is switched on for k graph G with к cycles; Δs_1 – gain from the decomposition of Boolean functions by finding the common parts; Δs_2 – gain from the factorization of the system of Boolean functions. In the second case,

$$L_2 = k_1 + 2k_2 + \sum_{\zeta=1}^{m}\left(k_{3_\zeta} - 1\right) + 2k_{4_\alpha} - 1 - \Delta s,$$

since in this case we select one memory element using both its inverse outputs, and $|B| = 1$. Since the different control system loops do not "function" simultaneously, many cycles in the operation graph can be realized using common memory and it is this consideration that determines $|B|$. The formulas for L_1 and L_2 do not include the elements realizing the subcombinations $\{Q\}$, since they are dictated by the technical specification of the robot design. The combinations $\{Q\}$ are lengthened by the procedure described in [3].

The proposed synthesis method yields one of the minimal systems of logical equations describing the functioning of an industrial robot. It has been tested for numerous control systems of industrial robots in operation.

4. The structural organization the hydraulic and of pneumatic systems of controls (Fig. 3) allows to reduce to a minimum number of elements of memory in the block of memory, and also number of the logic elements necessary for realization of the scheme (The patent № 1166064 (USSR).

Entrance block "AND" 1 intend for formation of set of conditions E. In the examined structure, unlike standard structure, signals from the entrance block 1 and from block "AND" 2 go directly in output block "OR" 6 whereas in standard structure in output block signals go from the block of memory. Block "OR" 3 is used at inclusion of one drive by different sets for different programs of work of a drive. Block "OR" 4 is used in case of several programs of work of a drive which lead to occurrence of various quantity of conditions for each program. The block of memory 5 includes consistently connected triggers with separate inputs, each previous trigger is switched off by a signal from an output of the following. Here one output of the trigger is used. Exception makes a case if the system contains two internal conditions, and the block of memory contains one trigger and two its inverse outputs are used. The output block "OR" 6 contains elements "OR". Inputs X consists of a subset of signals of inputs X_C automatic, and also from a of inputs X_B of manual management. On Fig. 1: Z – output signals, S – signals of inclusion of triggers, R – signals of cutting off of triggers.

Reduction of quantity of the equations is reached by two ways: reduction of quantity of the equations of inclusions of triggers (minimization of structure); reduction of quantity of the equations of inclusions of drives at use of valves with unilateral management.

We shall show method on an example of synthesis of the of pneumatic system of controls of the industrial robot. Operating condition of a control system is presented in the form of operation graph (Fig. 1) . The table of interaction of entrance and executive devices is given below:

Table 1 – Disposition of entrance and executive devices

Drive		Inputs signals	
Designation	Operation	Starting position	Final position
D_1	Holding mechanism	-	x_3
D_2	Telescopic arm	x_4	x_5
D_3	Hinged section	x_6	x_7
x_1 – Start button			

Formulas of a kind $Q_\gamma \mapsto Z_v$ (i.e. "the condition Z_v follow the condition Q_γ ") be on arches of graph. Here Q_γ represents entrance set which contains inputs of automatic control and of manual management. The set Z_v contains outputs which have changed the values on transition.

On graph: $P_0=\varnothing$, x_4x_6y; $P_1=x_1$, x_4x_6; $P_2=x_3$, x_4x_6; $P_3=x_5$, x_3x_6; $P_4=x_5$, x_3x_6; $P_5=x_7$, x_5; $P_6=x_6$, x_3x_5; $P_7=x_4$, x_3x_6.

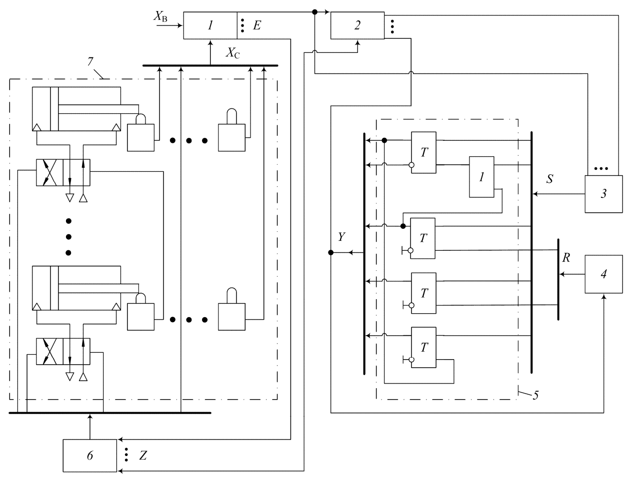

Fig. 3. The structural organization the hydraulic and of pneumatic systems of controls

The offered method reduces quantity of the equations synthesized owing to lengthening of sets Q . In this connection the quantity of columns of a matrix of conformity (MC) [3] is minimized. Rows of a matrix correspond full entrance sets, and columns correspond to the signals causing transitions. Algorithm of lengthenings of signals of inputs mentioned below.

1. *We go splitting sequence of entrance sets into blocks so that in each of them there were no identical entrance sets and last sets of the next blocks were not identical.*
2. *Construct matrix of conformity MC.*
3. *We extend signals of inputs of columns of matrix by signals of outputs of the trigger for identical sets.*
4. *We fill MC «1», if input signal of a column - a part of set of row of a matrix, and «0» differently.*
5. *We choose the equations for synthesis.*
6. *For the chosen equation we check a time interval of action of a output signal on column with the purpose of elimination «0», we spend lengthening of input signals. If elimination is not possible, we use the trigger in a power part.*
7. *Construct matrix of conformity MC when spend minimization.*

We write out the equations for functions of switching of triggers and functions of outputs. We form a disjunction of entrance sets for transitions of a time interval of action function $z_\zeta(\overline{z_\zeta})$. Let's consider a choice of the synthesized functions of outputs $\widetilde{Z} = f(X,Y)$. Where $\widetilde{Z}$ – functions of inclusion (disconnecting) of outputs; X – set of entrance signals, Y – outputs signals from triggers.

Functions of switching of triggers $S = f(X,Y)$, $R = f(Y)$. In case of two internal conditions $S, R = f(X)$,

Possible schemes of inclusion of channels of valve with unilateral management for drives are presented on Figure 3. For reduction of quantity of the equations it is enough to synthesize z_ζ or $\overline{z_\zeta}$. The basic criterion at a choice of function z_ζ or $\overline{z_\zeta}$ is the quantity of minimized columns MC (not used columns), and also quantity of lengthenings. The following criterion is quantity of transitions which make a time interval of action of outputs signals. If time intervals for both inverse functions identical it is expedient to write out both functions and to choose the optimum decision.

The of matrix conformity MC:

	x_1	x_3y	x_5y	x_7	$x_6\bar{y}$	$x_4\bar{y}$	
$\varnothing, x_4x_6 \mapsto \varnothing$	0	0	0	0	1	1	$\bar{y}$
$x_1, x_4x_6 \mapsto z_1 S$	1	0	0	0	1	0	
$^*x_3, x_4x_6 \mapsto z_2$	0	1	0	0	0	0	y
$^{**}x_5, x_3x_6 \mapsto z_3$	0	1	1	0	0	0	
$x_7, x_5 \mapsto \bar{z}_3 R$	0	0	0	1	0	0	
$^{**}x_6, x_3x_5 \mapsto \bar{z}_2$	0	0	0	0	1	0	$\bar{y}$
$^*x_4, x_3x_6 \mapsto \bar{z}_1$	0	0	0	0	1	1	

We choose functions of outputs of a control system scanning columns. For this purpose on a time interval of action z_ζ ($\overline{z_\zeta}$) is discover «0» which interrupt an interval. We spend lengthening Q of input signals. So for function z_1 (the first column) it is «0» on crossing with a column with an output $\overline{z_2}$ ($\bar{y}=1$). That does not allow to eliminate premature operation $\overline{z_1}=1$.

If to include the trigger in transition from the first top in the second the «0» in a time interval of action $y=1$ do not influence a time interval of action z_1 . Thus, for executive device D_1 we choose function $\overline{z_1}$. Here in a column of matrix $x_4\bar{y}$ on crossing with a row of a matrix for $z_1 = 1$, conflicts is eliminated at switching "memory". For executive device D_2 we choose "memory", namely valve with separate inputs, as elimination «0» is not possible. Besides, for executive device D_3 chosen function – z_3.

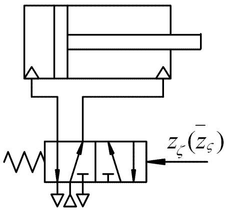

Fig. 4. Schemes of inclusion of channels of valve with unilateral management for drives

The matrix of conformity MC post facto minimization:

	x_3y	x_5y	$x_6\bar{y}$	$x_4\bar{y}$	
$\varnothing, x_4x_6 \mapsto \varnothing$	0	0	1	1	$\bar{y}$
$x_1, x_4x_6 \mapsto z_1 S$	0	0	1	0	
$*x_3, x_4x_6 \mapsto z_2$	1	0	0	0	y
$**x_5, x_3x_6 \mapsto z_3$	1	1	0	0	
$x_7, x_5 \mapsto \bar{z}_3 R$	0	0	0	0	
$**x_6, x_3x_5 \mapsto \bar{z}_2$	0	0	1	0	$\bar{y}$
$*x_4, x_3x_6 \mapsto \bar{z}_1$	0	0	1	1	

Synthesizing the equations, we receive the logic equations for this of case easier, than for an output block which contains triggers. Hence use of the offered algorithm leads to significant simplification of the logic equations. The pneumatic scheme of management of the industrial robot on Fig. 5 is present.

For synthesis of complex schemes it is expedient to take advantage of a method of decomposition of the equations.

Thus, the offered structure and the method of designing constructed on principles of discrete management, have doubtless advantages as the synthesized schemes contain the elementary discrete valves.

Thus, the synthesis method of schemes offered in article allows to reduce considerably number of valves, to thereby gain economic effect. So for the presented example, the scheme synthesized with use the structure standard position, contains 18 additional valves. Our scheme contains – 1.

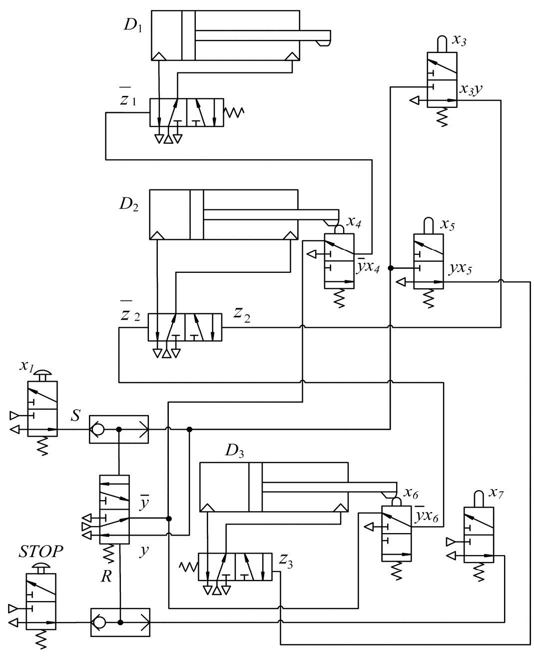

Fig. 5. The scheme of management of the industrial robot

References

1. Yuditsky S. To a question of the description and synthesis of discrete systems of industrial automatics. *Technical cybernetics*. 1976. № 1. pp. 131–141.
2. Cherkashenko M., [et. al.] Synthesis of discrete control systems of industrial robots. *Automation and Remote Control (USA).* 1981. № 5. pp. 148–153.
3. Cherkashenko M. Method of logic designing of discrete control systems of machines-automatic devices with pneumatic or hydraulic drive. *Pneumatic and hydraulic. Drives and control systems// Mechanical engineering*. 1981. № 8. pp. 181–189.
4. Sokol E., Cherkashenko M. *Synthesis of control schemes for hydroficated automation objects*. GmbH. Printed in Germany. 2018. 112 p.

References (transliterated)

1. Yuditsky S. To a question of the description and synthesis of discrete systems of industrial automatics. *Technical cybernetics*. 1976. no 1, pp. 131–141.
2. Cherkashenko M. [et. al.] Synthesis of discrete control systems of industrial robots. *Automation and Remote Control (USA).* 1981. no 5, pp. 148–153.
3. Cherkashenko M. Method of logic designing of discrete control systems of machines-automatic devices with pneumatic or hydraulic drive. *Pneumatic and hydraulic. Drives and control systems// Mechanical engineering*. 1981. no 8, pp. 181–189
4. Sokol E., Cherkashenko M. *Synthesis of control schemes for hydroficated automation objects*. GmbH. Printed in Germany. Publ. 2018. 112 p.

Received 09.11.2018

Відомості про авторів / Сведения об авторах / About the Authors

Черкашенко Михайло Володимирович (Черкашенко Михаил Владимирович, Cherkashenko Mikhaylo) – доктор технічних наук, професор, Національний технічний університет «Харківський політехнічний інститут», завідувач кафедри «Гідравлічні машини ім. Г. Ф. Проскури»; м. Харків, Україна; ORCID: https://orcid.org/0000-0003-3908-7935; e-mail: mchertom@gmail.com.