AUTOMATYKA/AUTOMATICS • 2014 • Vol. 18 • No. 2

http://dx.doi.org/10.7494/automat.2014.18.2.51

Sergiusz Ciskowski*, Tadeusz Mikulczyński*,
Zdzisław Samsonowicz*, Rafał Więcławek*

Analytic Method Grafpol TM of Synthesing Sequential Control Systems

1. Introduction

Dynamic development of semiconductor microtechnology and communication (IT) systems caused that nowadays automation of manufacturing processes is based on microprocessor control systems and digital communication systems (e.g. industrial communication networks Profibus and Profinet) [4]. One of the basic automation tools of modern manufacturing processes is a programmable logic controller (PLC) [1, 3].

Transition from traditional synthesis of designing and realising traditional contactor relay-based sequential control systems to systemic synthesis consisting in programming microprocessor control systems enforced the necessity to search for new analytic methods of modelling and programming discrete manufacturing processes.

The universal Grafpol method of modelling and programming manufacturing processes, developed in the Institute of Machine Engineering and Automation of Wroclaw University of Technology, is a method of synthesising any schematic equation representing analytic mathematical model of a sequential control algorithm [2]. The schematic equation, being a function of all the output variables and the elementary memory cells, can be used for:

- designing and realising traditional, contactor relay-based control systems,
- writing the application PLC program by means of a programming language, e.g. LD, ST etc.

The only inconvenience of synthesising sequential control algorithms by the Grafpol method is the way of synthesising the memory. This is related to the necessity of presenting in graphic way, based on the Grafpol GS network representing the control algorithm, logical relationships representing external signals of the control system (INPUTS and OUTPUTS). It is only on the grounds of their analysis, according to the mandatory rules, that functions of all the output variables and the elementary memory cells can be determined.

* Politechnika Wrocławska, Wrocław, Poland

The above-mentioned inconvenience is completely eliminated by the new Grafpol TM method. With this method, the memory of sequential control algorithms is realised in analytic way, on the grounds of analysis of IN signals of the control algorithm, which are presented in the status table based on the Grafpol GS network. Such principles of the memory synthesis and, in consequence, of the schematic equation, optimise (minimise) the time of synthesising sequential control algorithms.

2. Grafpol TM method

The Grafpol TM method is a universal method of modelling and synthesising sequential control algorithms. It is based on the universal process algorithm (of the Grafpol GP network) that makes ground for building the control algorithm (of the Grafpol GS network) and for synthesising the schematic equation that represents the analytic form of the control algorithm on that ground it is possible to realise the traditional contactor relay-based control system or to write the application program of a PLC controller.

Modelling sequential processes and programming PLC controllers includes the following phases:

PHASE I

Developing a schematic functional diagram of the process, dividing the process to elementary stages and formulating a verbal description of its realisation (process algorithm).

PHASE II

Presenting the process algorithm in the form of a graphic-analytic mathematical model representing the Grafpol GP network.

PHASE III

Building the control algorithm. The control algorithm (Grafpol GS network) is obtained by transformation of the process algorithm. It consists in representing a set of elementary stages of the process by a set of output signals of the control system, which constitute reali-sation of individual elementary stages of the process.

PHASE IV

Synthesising the memory and a schematic equation of the sequential control algorithm. The memory is realised according to the regulations presented in this work, on the grounds of analysis of IN signals representing the control algorithm, i.e. the Grafpol GS network. Having at the disposal the Grafpol GS network and the memory determined on its ground it is possible to determine the schematic equation of the sequential control algorithm in the form:

$$F(Y,M) = \sum_{i=1}^{n} Y_i + \sum_{j=1}^{m} M_j \tag{1}$$

where:

Y_i – i-th output signal of the control system,

M_j – j-th elementary memory cell,

n – number of output signals of the control system,

m – number of elementary memory cells.

2.1. Process algorithm – Grafpol GP network

The algorithm of a discrete manufacturing process is represented by the Grafpol GP network. This network is composed of the triplet:

$$GP = \langle E, T, K \rangle \tag{2}$$

where:

E – finite, non-empty set of the locations representing elementary stages of the process, of the process,

T – finite, non-empty set of the transitions representing logic conditions of realising elementary stages of the process,

K – set of oriented segments.

$E \cap T = 0$, i.e. E and T are disjointed sets. K is a realisation determined on the set $E \cup T$, meeting the condition $K \in (E \times T) \cup (T \times E)$.

The Grafpol GP network is built using graphic symbols shown in Figure 1.

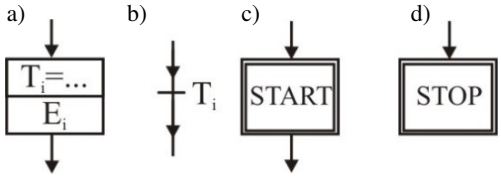


Fig. 1. Graphic symbols of elements of Grafpol GP network: a) location (of elementary stage), b) transition, c) phase START, d) phase STOP

Individual graphic symbols shown in Figure 1 have the following meanings:

LOCATION (Fig. 1a)

Location represents an elementary stage of the process. It is conventional that its graphic symbol includes transition that in analytic way represents the logic condition that determines beginning of execution of the elementary stage.

TRANSITION (Fig. 1b)

Transition represents a logic condition determining execution of the elementary stage of the process. Its graphic symbol is a horizontal segment on the vertical direction joining

two elementary stages. Analytic form of logic conditions representing individual transitions is conventionally placed in the windows of graphic symbols of the locations.

PHASE START (Fig. 1c)

The phase START represents status of the process at the moment when its execution begins.

PHASE STOP (Fig. 1d)

The phase STOP represents status of the process at the moment when it is completed.

The Grafpol GP network is presented in the form of an oriented graph having two kinds of nodes: location and transition. The network is written from top to bottom, according to the sequence of individual elementary stages of the modelled process.

Any elementary stage of a discrete manufacturing process can be in one of two states: active and inactive. An elementary stage is active from the moment of beginning its execution to the moment of beginning execution of the following stage. Individual elementary stages of the process are in the active state, when they meet logic relationships determining execution conditions of the stages. A stage E_i is activated when the relationship describing the transition T_i accepts logic value 1, and it is deactivated when the transition is $T_i + 1 = 1$. It results from this that individual states of the process are consequences of activity of its elementary stages. A change of the process status (change of activity of its stages) is represented by fulfilling the logic condition determining transition from execution of one stage to execution of the next stage. Building a Grafpol GP network is based on the following elements:

- functional diagram of the process,
- verbal description of the process algorithm.

The functional diagram must present the process in its initial (starting) state and include all actuating elements (assemblies) of individual elementary stages, as well as all the elements (outlet signals of the process) signalling complete execution of individual elementary stages.

Description of the process algorithm must include a description of all the elementary stages (according to the accepted standard). The sequence of writing individual stages should be in accordance with the presumed sequence of their execution (occurrence in the process algorithm). The accepted standard description of the elementary stage is as follows:

```
STAGE Ei: * stage name *
Execution: EWi (symbol of the actuator)
Signalling: Wi (symbol of the element signalling state
                of the actuator).
```

Application of the Grafpol TM method for writing the operation algorithm of two pneumatic drives is presented below. The functional diagram of two pneumatic drives S1 and S2

is shown in Figure 2. Description of operation of the actuators (pneumatic cylinders) S1 and S2, which includes four sequentially executed elementary stages E1 to E4, is as follows:

```
STAGE E1:* extension of piston rod S1*
Execution: S1+(EZ1+)
Signalling: WP2=1
STAGE E2:* extension of piston rod S2*
Execution: S2+(EZ3+)
Signalling: WP4=1
STAGE E3:* retraction of piston rod S2*
Execution: S2-(EZ4+)
Signalling: WP3=1
STAGE E4:* retraction of piston rod S1*
Execution: S1-(EZ2+)
Signalling: WP1=1
```

Operation of actuators S1 and S2 is cyclical.

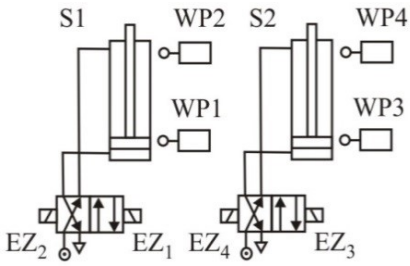


Fig. 2. Functional diagram of two pneumatic drives

The following symbolic designations were used in verbal description of the operation algorithm of the cylinders S1 and S2:

WP1–WP4 – represent signals of position indicators of piston rods,

S^+ – means extension of piston rod,

S^- – means retraction of piston rod,

EZ_2^+ – means excited state of the distribution valve coil,

EZ_2^+ – means idle state of the distribution valve coil.

Figure 3 shows the Grafpol GP network representing the operation algorithm of the cylinders S1 and S2.

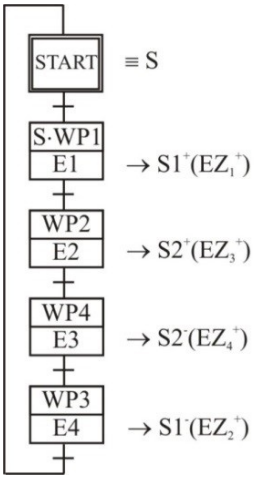


Fig. 3. Operation algorithm of two pneumatic drives

2.2. Control algorithm – Grafpol GS network

Synthesis of a control algorithm is based on the process algorithm. This fact is determined by the relation between the process and the control system:

$$\text{CONTROL} \Rightarrow \text{EXECUTION} \Rightarrow \text{PROCESS}$$

because execution of individual elementary stages of the process is controlled by output signals of the control system related to them. So, a change of the process status is a result of changed status of the control system. It results from this that state of elementary stages of the process must be equivalent to that of output signals from the control system, which control the elementary stages. Therefore, the control algorithm can be determined by transformation of the process algorithm.

Transformation of the process algorithm consists in representing a set of elementary stages of the process by a set of output variables of the control system, whose output signals control execution of the elementary stages.

Transformation of the process algorithm (Grafpol GP network) results in obtaining the control algorithm representing the Grafpol GS network. This network is composed of the triplet:

$$GS = \langle Y, T, K \rangle \tag{3}$$

where:

Y – finite, non-empty set of output signals of the control system,

T, K – sets equivalent to the sets T and K of the Grafpol GP network.

When representing a set of elementary stages of the process by a set of output signals of the control system, the required way of controlling the actuators must be considered – this concerns, first of all, pneumatic and hydraulic distribution valves. It should be emphasised here that the system states in that coils of specific distribution valves should be in the excited state, are unequivocally determined by the Grafpol GP network. So, it is only necessary to place in the Grafpol GS network a description of setting and cancelling outputs of the control system, which control current states of coils of determined distribution valves. This can be realised e.g. by using relay outputs PLC $Y(S)$ (for Set) – writing and $Y(R)$ (for Reset) – cancelling, or by using logic feedback loops (logic sums) and normally closed contacts in the case of designing contactor relay-based systems. Synthesis of the control algorithm by the Grafpol TM method is illustrated on an example of two pneumatic drives S1–S2, whose operation algorithm was presented under 2.1. Figure 4 shows the control algorithm determined on the ground of the process algorithm from Figure 3.

In this transformation, output variables of the control system were assigned to elementary stages of the process in the following way:

$$Y_1 \Rightarrow EZ_1^+ \Rightarrow E_1$$
$$Y_2 \Rightarrow EZ_3^+ \Rightarrow E_3$$
$$Y_3 \Rightarrow EZ_4^+ \Rightarrow E_3$$
$$Y_4 \Rightarrow EZ_2^+ \Rightarrow E_4$$

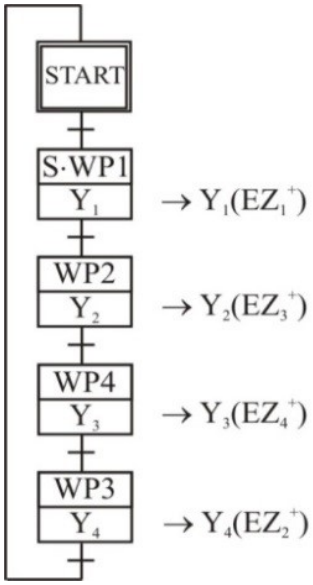


Fig. 4. Control algorithm for operation of two pneumatic drives S1–S2, determined by transformation of the process algorithm from Figure 3

3. Synthesis of memory and schematic equation of sequential control systems

Sequential control systems are such systems, whose present state of output systems depends not only on present state of output signals at the given moment t, but also on states of input signals occurring at previous moments $t-1, t-2, \ldots, t-n$. They are described by the following output function:

$$Y^t = f(X^t, X^{t-1}, \ldots, X^{t-n}) \tag{4}$$

It results from here unequivocally that a sequential control system must have a memory. Thus, synthesis of sequential algorithms of control systems is based on synthesising the memory of these systems.

In fact, the most important issue related to synthesis of sequential control algorithms is synthesising the memory of these control systems.

Extremely simple and unequivocal principles of synthesising sequential control algorithms have been specified in the Grafpol TM method. The developed method of synthesising the memory of sequential control algorithms is an analytic method. Thus, this is an outstanding achievement of its authors.

Synthesis of the memory of sequential control algorithms is determined by the following basic principles:

PRINCIPLE 1

A basis for synthesising a sequential control algorithm is established by the status table that represents states of input signals. The status table is based on the control algorithm represented by the Grafpol GS network.

PRINCIPLE 2

The status table should include only these input signals that signal initial state of actuators of elementary stages of the process. In the case of pneumatic drives, this is one of two signals signalising position of the cylinder piston.

PRINCIPLE 3

The memory (elementary memory cells) is written on the grounds of analysis of equivalent states (of input signals) present in the control algorithm. Equivalent states of input signals of the control algorithm are those having the same state of input signals when being in various states of the control algorithm. If it is, for example:

$$\begin{aligned} &X_1 = \{0,1,1,0,1\} \\ &\vdots \\ &X_4 = \{0,1,1,0,1\} \\ &\vdots \\ &X_9 = \{0,1,1,0,1\} \end{aligned} \tag{5}$$

so the states X_1, X_4 and X_9 are equivalent states.

Taking into account the presented principles, the following rules of synthesising the memory of sequential control algorithms, i.e. of writing and deleting elementary memory cells, can be formulated.

RULE 1

If the control algorithm includes such equivalent states between whom occurs one state only, so only one elementary memory cell $M_j(S)$ should be written in each of these states. This memory write concerns all pairs of the equivalent states.

RULE 2

Transition of the first state of a pair of equivalent states has the following form:

$$T_i^* = T_i \cdot \overline{M_j} \tag{6}$$

where:

T_i – transition of i-th equivalent state,

T_i^* – transition of i-th equivalent state in that the memory is considered (transition with memory),

$\overline{M_j}$ – negated output signal of the j-th elementary memory cell M_j.

Transition of the second state of a pair of equivalent states has the following form:

$$T_k^* = T_k \cdot M_j \tag{7}$$

where:

T_k – transition of k-th equivalent state,

T_k^* – transition of k-th equivalent state in that the memory is considered,

$\overline{M_j}$ – output signal of the j-th elementary memory cell M_j.

RULE 3

If the first states of equivalent state pairs belonging to different groups of equivalent states happen in the control algorithm one after the other, so writing one elementary memory cell should happen in the first following state that does not belong to any group of equivalent states (is not an equivalent state).

RULE 4

Transition of each first state of an equivalent state pair has the following form:

$$T_i^* = T_i \cdot \overline{M_j} \tag{8}$$

Transition of each second state of an equivalent state pair has the following form:

$$T_j^* = T_j \cdot \overline{M_j} \tag{9}$$

The following conclusion results from the rules 3 and 4: One elementary memory cell can comprise not one only, but several pairs of neighbouring expressions. This means that the Grafpol TM method permits determining the optimum (minimum) number of elementary

memory cells and thus determining the optimum (minimum) form of the schematic equation of a sequential control system.

RULE 5

Deleting elementary memory cells should take place in the following way:

1) In the last state of the control algorithm – deleting all the elementary memory cells, if transition of the last state has the form:

$$T_n^* = T_n \tag{10}$$

where n – number of the last state.

Then, the equation that describes deleting the memory is represented by the relationship

$$F[M(R)] = T_n^* \cdot \sum_{j=1}^{m} M_j(R) \tag{11}$$

where:
M_j – j-th elementary memory cell,
m – number of elementary memory cells.

2) In the first state of the control algorithm (equivalent to the phases START and STOP) – deleting such an elementary memory cell whose output signal is contained in the transition of the last state of the control system. If the transition is

$$T_n^* = T_n \cdot M_k \tag{12}$$

where M_k – k-th elementary memory cell, so deleting the memory M_k should be realised according to the following relationship:

$$M_k(R) = T_1 \tag{13}$$

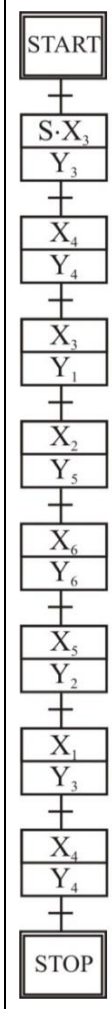
where T_1 – transition of the first state.

A practical application of the rules of synthesising the memory by the Grafpol TM method is illustrated by synthesis of an exemplary control algorithm shown in Figure 5.

Explanations concerning principles of writing and deleting elementary memory cells of the control algorithm shown in Figure 5 are given below.

The memory M1 concerning the equivalent states 1 and 3 should be written according to the rule 1, in the second state of the control algorithm. According to this rule, transitions of the states 1 and 3 should have the following forms:

$$T_1^* = T_1 \cdot \overline{M_1}, \quad T_3^* = T_3 \cdot M_1 \tag{14}$$



Process state	Input signals X_1	Input signals X_3	Input signals X_5	Writing memory $M_j(S)$	Deleting memory $M_j(R)$	Transition with memory T_i^*	Writing outputs Y(S)	Deleting outputs Y(R)
1	1	1	1			$T_1^* = S \cdot X_3 \cdot \overline{M_1}$	$Y_3(S)$	$Y_4(R)$
2	1	0	1	$M_1(S)$		$T_2^* = X_4 \cdot \overline{M_2}$	$Y_4(S)$	$Y_3(R)$
3	1	1	1			$T_3^* = X_3 \cdot M_1 \cdot \overline{M_2}$	$Y_1(S)$	$Y_4(R)$
4	0	1	1			$T_4^* = X_2 \cdot \overline{M_2}$	$Y_5(S)$	$Y_1(R)$
5	0	1	0	$M_2(S)$		$T_5^* = X_6$	$Y_6(S)$	$Y_5(R)$
6	0	1	1			$T_6^* = X_5 \cdot M_2$	$Y_2(S)$	$Y_6(R)$
7	1	1	1			$T_7^* = X_1 \cdot M_2$	$Y_3(S)$	$Y_2(R)$
8	1	0	1		$M_1(R)$	$T_8^* = X_4 \cdot M_2$	$Y_4(S)$	$Y_3(R)$
	1	1	1		$M_2(R)$	$T_{STOP}^* = X_3$		

Fig. 5. Illustration of memory realisation principles of an exemplary control algorithm, as well as of writing and deleting output signals of the designed control system

The memory M2 concerning the equivalent states 2 and 8, 3 and 7, 4 and 6 should be written according to the rule 3, in the fifth state of the control algorithm. According to this

rule, transitions of the states 2, 3 and 4 should have the following forms:

$$T_2^* = T_2 \cdot \overline{M_2}, \ldots, \; T_3^* = T_3 \cdot M_1 ? \overline{M_2}, \ldots, \; T_4^* = T_4 \cdot \overline{M_2}, \tag{15}$$

but transitions of the states 6, 7 and 8 are described by the relationships:

$$T_6^* = T_6 ? M_2, \; T_7^* = T_7 ? M_2, \; T_8^* = T_8 ? M_2 \tag{16}$$

Deleting the memory M_1 should take place in the state 8 of the control algorithm, so it is determined by the transition T_8^*, but deleting the memory M_2 must take place in the state STOP, so it is determined by the transition $T = X_3$.

3.1. Synthesis of schematic equation of a sequential control algorithm

As well-known, the Grafpol GP network is the mathematical model of a process algorithm. It represents elementary stages of the process and logic conditions of their realisation. These conditions represent the transition T_i.

In order that realisation of the process stages proceeds according to the assumed algorithm, the transitions should meet the following conditions:

- $T_i = 1$ in the state of the control system responsible for starting execution of the stage E_i,
- $T_{i+1} = 0$ in the state of the control system in that the variable Y_i is deleted.

The Grafpol GS network is a mathematical model of the control algorithm. It represents only external signals of the control system, since it is determined on the grounds of transformation of the Grafpol GP network. Because of the above, external signals of the control system can be described by the conditions equivalent to the conditions (1):

- $T_i = 1$ in the state of the control system in that the variable Y_i is written,
- $T_{i+1} = 0$ in the state of the control system in that the variable Y_i is deleted.

In order that the control system ensures realisation of the process in accordance with the assumed algorithm, it is necessary to consider the memory in the control algorithm. Then, it is possible to determine, on the grounds of the rules of the memory realisation described in the previous section, the transitions T_i^* and T_j^* describing functions of writing and deleting output variables Y_i and elementary memories M_j.

The functions of writing and deleting all output variables, as well as of writing and deleting elementary memory cells, make grounds for determining a schematic equation of the control algorithm in the following form:

$$F(Y,M) = \sum_{i=1}^{n} T_i^* Y(S) + \sum_{i=1}^{n} T_i^* Y(R) + \sum_{j=1}^{k} T_j^* M(S) + \sum_{j=1}^{k} T_j^* M(R) \tag{17}$$

4. Examples of modelling and programming sequential control algorithms

Application of the Grafpol TM method for synthesising application programs of PLC controllers is illustrated by the below-mentioned examples of modelling operation of pneumatic drives and programming PLC controllers.

Example 1

Figure 6 shows a functional diagram of three pneumatic drives S1–S3 controlled by double-sided solenoid distribution valves.

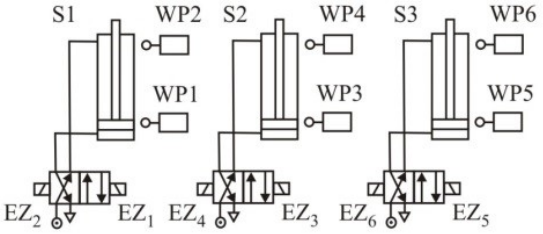


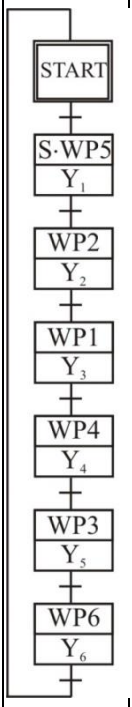
Fig. 6. Functional diagram of three pneumatic drives

The operation algorithm of the pneumatic drives shown in Figure 6 comprises of a sequence of the following elementary stages:

STAGE E1:* extension of piston rod of the cylinder S1*
Execution: $S1^{+}(EZ_1^{+})$
Signalling: WP2=1
STAGE E2:* retraction of piston rod of the cylinder S1*
Execution: $S1^{-}(EZ_2^{+})$
Signalling: WP1=1
STAGE E3:* extension of piston rod of the cylinder S2*
Execution: $S2^{+}(EZ_3^{+})$
Signalling: WP4=1
STAGE E4:* retraction of piston rod of the cylinder S2*
Execution: $S2^{-}(EZ_4^{+})$
Signalling: WP3=1
STAGE E5:* extension of piston rod of the cylinder S3*
Execution: $S3^{+}(EZ_5^{+})$
Signalling: WP6=1
STAGE E6:* retraction of piston rod of the cylinder S3*
Execution: $S3^{-}(EZ_6^{+})$
Signalling: WP5=1

Figure 7 shows the control algorithm and the status table. The control algorithm was determined by transformation of the process algorithm, assigning output variables of the control algorithm to elementary stages of the process in the following way:

$$Y_1 \Rightarrow EZ_1^+ \Rightarrow E_1$$
$$Y_2 \Rightarrow EZ_2^+ \Rightarrow E_3$$
$$Y_3 \Rightarrow EZ_3^+ \Rightarrow E_3$$
$$Y_4 \Rightarrow EZ_4^+ \Rightarrow E_4$$
$$Y_5 \Rightarrow EZ_5^+ \Rightarrow E_5$$
$$Y_6 \Rightarrow EZ_6^+ \Rightarrow E_6$$



Process state	Transition T_i	Input signals			Writing memory $M_j(S)$	Deleting memory $M_j(R)$	Transition with memory T_i^*	Writing outputs Y(S)	Deleting outputs Y(R)
		WP1	WP3	WP5					
1	T_1=S·WP5	1	1	1			$T_1^* = T_1 \cdot \overline{M_1}$	Y_1(S)	Y_6(R)
2	T_2=WP2	0	1	1	M_1(S)		$T_2^* \equiv T_2$	Y_2(S)	Y_1(R)
3	T_3=WP1	1	1	1			$T_3^* = T_3 \cdot M_1 \cdot \overline{M_2}$	Y_3(S)	Y_2(R)
4	T_4=WP4	1	0	1	M_2(S)		$T_4^* \equiv T_4$	Y_4(S)	Y_3(R)
5	T_5=WP3	1	1	1			$T_5^* = T_5 \cdot M_2$	Y_5(S)	Y_4(R)
6	T_6=WP6	1	1	0		M_1(R) + M_2(R)	$T_6^* \equiv T_6$	Y_6(S)	Y_5(R)

Fig. 7. Control algorithm of operation of pneumatic drives S1–S3 with the status table making ground for synthesis of schematic equation of the control system

When all functions of writing and deleting the output variables and the elementary memory cells represented by the status table are known, it is possible to formulate a schematic equation of the control algorithm for operation of the pneumatic drives S1–S3.

The schematic equation has the following form:

$$F(Y,M)=\sum\begin{cases} S\cdot WP5\cdot\overline{M_1}\cdot[Y_1(S)+Y_6(R)] \\ WP2\cdot[Y_2(S)+Y_1(R)+M_1(S)] \\ WP1\cdot M_1\cdot\overline{M_2}\cdot[Y_3(S)+Y_2(R)] \\ WP4\cdot[Y_4(S)+Y_3(R)+M_2(S)] \\ WP3\cdot M_2\cdot[Y_5(S)+Y_4(R)] \\ WP6\cdot[Y_6(S)+Y_5(R)+M_1(R)+M_2(R)] \end{cases} \tag{18}$$

Figure 8a shows a diagram of IN/OUT signals of the PLC controller and Figure 8b shows the application PLC program written in the LD language.

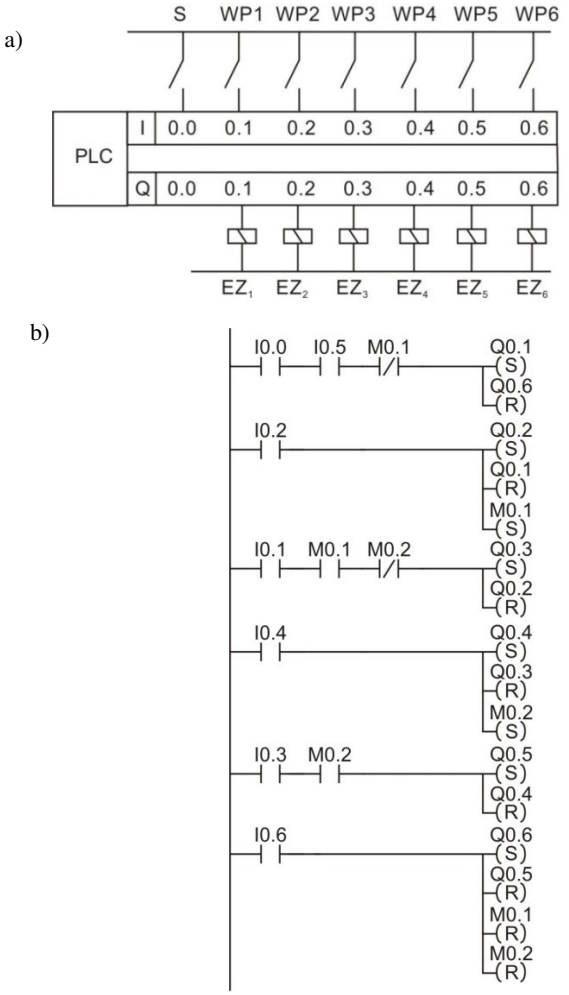


Fig. 8. Schematic diagram of IN/OUT signals of PLC controller a) and application PLC program in LD language b) controlling operation of three drives S1–S3

Example 2

Figure 9 shows a functional diagram of two pneumatic drives S1–S2 controlled by double-sided solenoid distribution valves.

The operation algorithm of the pneumatic drives shown in Figure 9 comprises of a sequence of the stages E1–E6.

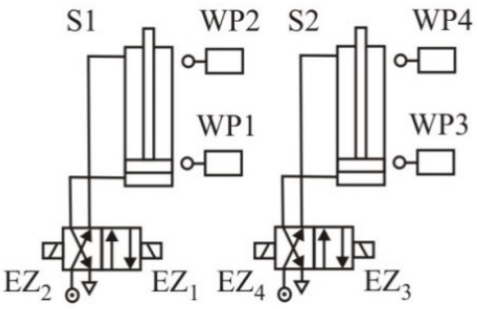
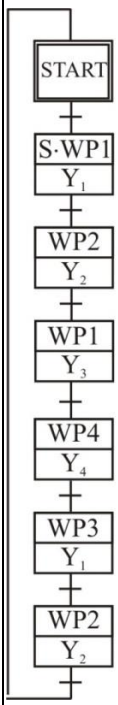


Fig. 9. Functional diagram of two pneumatic drives S1–S2

```
STAGE E1:* extension of piston rod of the cylinder S1*
Execution: S1+(EZ1+)
Signalling: WP2=1
STAGE E2:* retraction of piston rod of the cylinder S1*
Execution: S1-(EZ2+)
Signalling: WP1=1
STAGE E3:* extension of piston rod of the cylinder S2*
Execution: S2+(EZ3+)
Signalling: WP4=1
STAGE E4:* retraction of piston rod of the cylinder S2*
Execution: S2-(EZ4+)
Signalling: WP3=1
STAGE E5:* extension of piston rod of the cylinder S1*
Execution: S1+(EZ1+)
Signalling: WP2=1
STAGE E6:* retraction of piston rod of the cylinder S1*
Execution: S1-(EZ2+)
Signalling: WP1=1
```

A schematic equation of the designed sequential control system can be determined on the grounds of the status table shown in Figure 10. The equation has the following form:

$$(Y,M)=\sum\begin{cases} S\cdot WP1\cdot\overline{M_1}\cdot[Y_1(S)+Y_2(R)+M_3(R)] \\ WP2\cdot\overline{M_2}\cdot[Y_2(S)+Y_1(R)+M_1(S)] \\ WP1\cdot M_1\cdot\overline{M_2}\cdot[Y_3(S)+Y_2(R)] \\ WP4\cdot[Y_4(S)+Y_3(R)+M_2(S)] \\ WP3\cdot M_2\cdot\overline{M_3}\cdot[Y_1(S)+Y_4(R)] \\ WP2\cdot M_2\cdot[Y_2(S)+Y_1(R)+M_3(S)] \\ WP1\cdot M_3\cdot[M_1(R)+M_2(R)] \end{cases} \tag{19}$$



Process state	Transition T_i	Input signals WP1	Input signals WP3	Writing memory $M_j(S)$	Deleting memory $M_j(R)$	Transition with memory T_i^*	Writing outputs Y(S)	Deleting outputs Y(R)
1	T_1=S·WP1	1	1		$M_3(R)$	$T_1^* = T_1 \cdot \overline{M_1}$	$Y_1(S)$	$Y_2(R)$
2	T_2=WP2	0	1	$M_1(S)$		$T_2^* = T_2 \cdot \overline{M_2}$	$Y_2(S)$	$Y_1(R)$
3	T_3=WP1	1	1			$T_3^* = T_3 \cdot M_1 \cdot \overline{M_2}$	$Y_3(S)$	$Y_2(R)$
4	T_4=WP4	1	0	$M_2(S) \equiv M_2^*(S)$		$T_4^* \equiv T_4$	$Y_4(S)$	$Y_3(R)$
5	T_5=WP3	1	1			$T_5^* = T_5 \cdot M_2 \cdot \overline{M_3}$	$Y_1(S)$	$Y_4(R)$
6	T_6=WP2	0	1	$M_3(S)$		$T_6^* = T_6 \cdot M_2$	$Y_2(S)$	$Y_1(R)$
7	T_7=WP1	1	1		$M_1(R)$ $M_2(R)$	$T_7^* = T_7 \cdot M_3$		

Fig. 10. Control algorithm of operation of pneumatic drives S1-S2 with the status table

Figure 11 shows a diagram of IN/OUT signals of the PLC controller and Figure 12 shows the application PLC program written in the LD language.

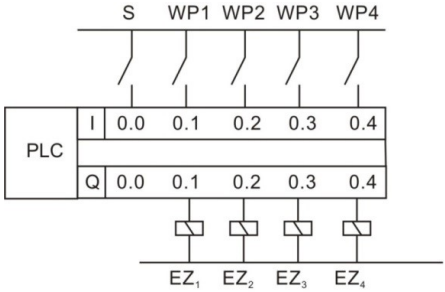


Fig. 11. Schematic diagram of IN/OUT signals of PLC controller

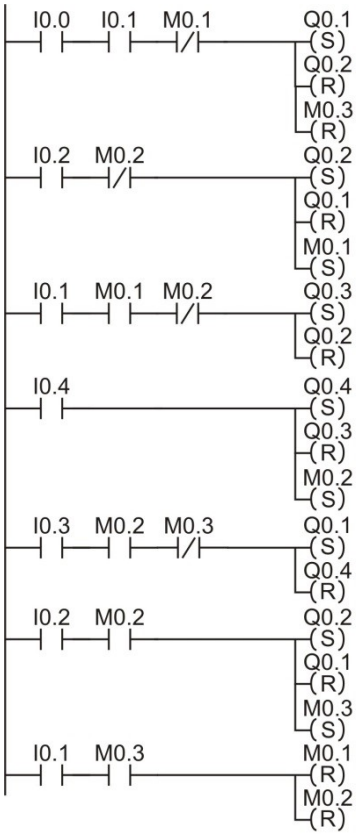


Fig. 12. Application PLC program in LD language controlling operation of two drives S1–S2

5. Conclusions

The paper presents the new Grafpol TM method of synthesising sequential control algorithms. This method is a universal method that permits determining the optimum (minimum) form of a schematic equation of any sequential control algorithm.

The extraordinary advantages of the Grafpol TM method have been obtained by basing synthesis of sequential control algorithms on the status table and on the formulated principles of realising memories and synthesising functions of output variables of sequential control algorithms. As a result, the table makes a basis for writing the sequential control algorithm in the form of a schematic equation.

The schematic equation of a sequential control algorithm can be applied for: designing a traditional contactor relay-based control system, writing an application program for the PLC controller.

Advantages of the Grafpol TM method and possibilities of its use for modelling and programming any sequential control algorithms will be presented in the next publications.

References

[1] Goszczyński T., *Sterowniki PLC – przegląd rynku.* Pomiary Automatyka Robotyka, 10:14–22, 2011.

[2] Mikulczyński T., *Automatyzacja procesów produkcyjnych*. Wydawnictwa Naukowo-Techniczne, Warszawa, 2006.

[3] Oprzędkiewicz K., *Porównanie języków programowania sterowników PLC pod kątem spełnienia wymagań czasu rzeczywistego.* Pomiary Automatyka Robotyka, 12:5–9, 2006.

[4] Rosół M., Knapik D., Marchewka D., Kołek K., Turnau A., *Analysis of Industrial PROFINET in the Task of Controlling a Dynamic System.* Automatics, 17(1):65–72, 2013.