A R C H I V E S O F M E T A L L U R G Y A N D M A T E R I A L S
Volume 55 2010 Issue 3

Ł. DWORZAK*, T. MIKULCZYŃSKI*

AUTOMATION OF FOUNDRY PROCESSES WITH GRAFPOL METHOD

AUTOMATYZACJA PROCESÓW ODLEWNICZYCH METODĄ GRAFPOL

Application of the Grafpol method for synthesising a complex procedure control algorithms for automation of foundry processes is presented. An example was based on the company's Technical MTP-3000 turbine mixer. The developed principles simplify the memory realisation. Thanks to this, time for synthesising the schematic equation can be significantly reduced in comparison to the network transformation method. The designed schematic equation makes a ground for writing an application program of a PLC using any language defined in defined in the standard PN-EN 61131-3.

Keywords: Grafpol method, programing PLC, schematic equation

W pracy zaprezentowano zastosowania metody Grafpol do syntezy złożonych algorytmów sterowania w automatyzacji procesów odlewniczych. Przedstawiony przykład bazuje na mieszarce turbinowej MTP-3000 firmy Technical. Rozwinięte zasady uproszczają sposób realizacji pamięci. Dzięki temu czas syntezy równania schematowego uległ istotnemu skróceniu w porównaniu z metodą transformacji sieci. Wysyntezowane równanie schematowe stanowi podstawę do zapisu algorytmu sterowania w sterowniku PLC za pomocą dowolnego języka zdefiniowanego w normie PN-EN 61131-3.

1. Introduction

Progress in the present-day industry, that refers to dynamic development of products, to meeting higher and higher qualitative requirements and minimising production costs, is possible, among others, thanks to complex automation of all the areas related to manufacturing – from company management through designing and production management till automation of manufacturing processes including casting. This is because automation of manufacturing processes makes a ground for modern manufacturing systems and attributes mostly to high quality of the manufactured products and to minimising the production costs.

Basic tools of automating modern manufacturing processes are: Programmable Logic Controllers PLC, Computer Numerical Control/Distributed Numerical Control CNC/DNC, Industrial Controllers IPC, Robot Controllers RC, Distributed Control Systems DCS and Hybrid Control Systems HCS, as well as microprocessor-based controllers.

Transition from traditional synthesis, which allowed designing and realising traditional contact/relay sequential control systems of manufacturing processes, to system synthesis consisting in programming microprocessor control systems (PLC) made it necessary to search new analytic methods which could be applied in modelling manufacturing processes and programming PLC controllers, that is in automation of modern manufacturing processes.

A new method of mathematical modelling the production processes and programming PLC controllers is the Grafpol method developed in the Laboratory of Basic Automation of the Institute of Machine Engineering and Automation of Wroclaw University of Technology. Modelling and programming the discrete production processes by means of pneumatic, hydraulic and electric drives with the Grafpol method includes the following phases:

Phase I

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

The functional diagram should present the process in its initial stage. It must include all the components or actuators of individual elementary stages and their verbal description.

* INSTITUTE OF PRODUCTION ENGINEERING AND AUTOMATION, WROCLAW UNIVERSITY OF TECHNOLOGY, 50-370 WROCŁAW, WYBRZEŻE WYSPIAŃSKIEGO 27, POLAND

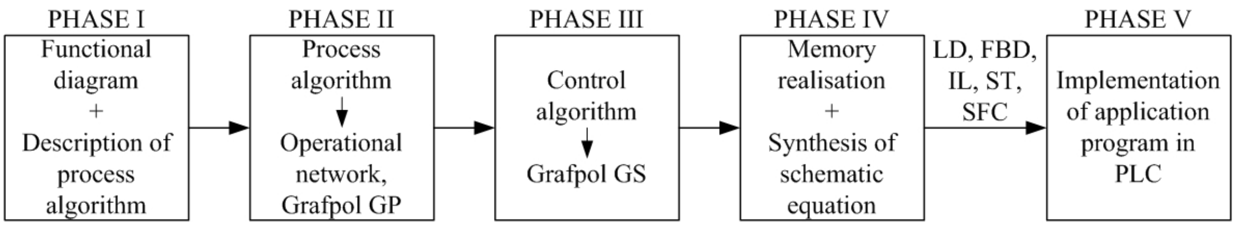

Fig. 1. Phases of modelling acc. to Grafpol method

Phase II

Building a graphical-analytical mathematical model of the process algorithm.

To present mathematical algorithms of discrete manufacturing processes with the Grafpol method, the operational network and the Grafpol GP network determined on its ground are applied.

Phase III

Synthesising a control algorithm – the Grafpol GS network.

The control algorithm is obtained by transforming the process algorithm. The transformation consists in representing a set of the process elementary stages by a set of the PLC output signals which control execution of individual elementary stages.

Phase IV

Realising a memory of the control algorithm and determining a schematic equation.

The memory is determined on the ground of the control algorithm. After realising the memory, it is possible to determine a schematic equation being a sum of all the output variables and elementary memory cells.

Phase V

Writing an application program for the PLC.

Notation of a PLC application program is based on the schematic equation. This program can be written using one of the PLC programming languages accepted in PN-EN 61131-3. These are the following languages: LD (Ladder Diagram), FBD (Function Block Diagram), IL (Instruction List), and ST (Structured Text). The above-mentioned phases are shown in Fig. 1.

Below, presented is the Grafpol method permitting synthesis of control algorithms which include sequential and concurrent procedures.

2. Grafpol method

Core of the Grafpol method [1] is a universal mathematical model of the manufacturing process algorithm represented by the operational network. An operational network is created by the three values:

$$SO = < E, W, O >$$

where:

E=$\{e_1, e_2, ..., e_e\}$ – finite, non-empty set of elementary stages of the process,

W=$\{w_1, w_2, ..., w_w\}$ – finite, non-empty set of logic conditions determining realisation of the elementary stages,

O=$\{o_1, o_2, ..., o_o\}$ – finite set of nodes of the alternative/conjunction operations.

Graphic symbols used for building an operational network are shown in Fig. 2.

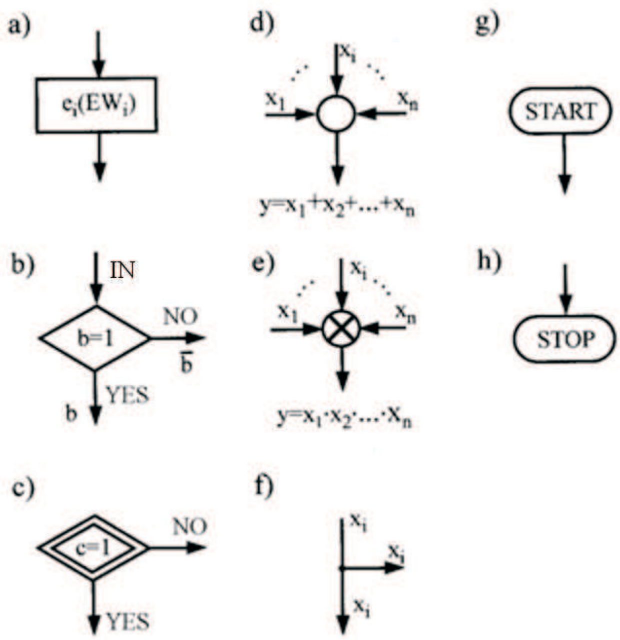

Fig. 2. Graphic symbols of operational network components: a) process symbol, b) decision symbol, c) independent decision symbol, d) alternative node, e) conjunction node, f) signal branching, g) START symbol, h) STOP symbol

An algorithm of a discrete manufacturing process can be presented by means of another, equivalent model of the operational network, namely by the Grafpol GP network.

A Grafpol GP network is created by the three values:

$$GP = < E, T, K >$$

where:

E={$e_1, e_2, ..., e_e$} – finite, non-empty set of elementary stages of the process,

T={$t_1, t_2, ..., t_t$} – finite, non-empty set of transitions representing logic conditions of executing elementary stages of the process,

K – set of oriented lengths which determine directions of signal flow in the Grafpol network.

Graphic symbols used for building a Grafpol GP network are shown in Fig. 3.

Figure 4 shows an exemplary diagram of two pneumatic drives 1A and 2A. Figure 5 shows model of an exemplary algorithm of operation of these pneumatic drives.

As a result of transformation of the process algorithm (Grafpol GP network), a control algorithm is obtained, represented by the Grafpol GS network.

A Grafpol GS network is created by the three values:

$$GS =< S, T, K >$$

where:

S={$s_1, s_2, ..., s_s$} – finite, non-empty set of elementary stages of the positions named “steps”. They represent output signals of the PLC controller, which control execution of individual elementary stages of the process.

The control algorithm makes a ground for realising the memory and, in consequence, determining a schematic equation on which ground the PLC application program is written (to control the process acc. to the assumed algorithm).

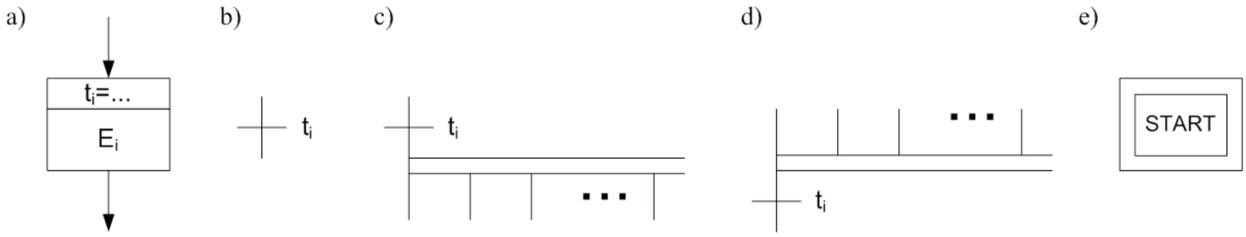

Fig. 3. Graphic symbols of Grafpol network: a) position, b) transition, c) transition representing start of the concurrent procedure, d) transition representing end of the concurrent procedure, e) START stage

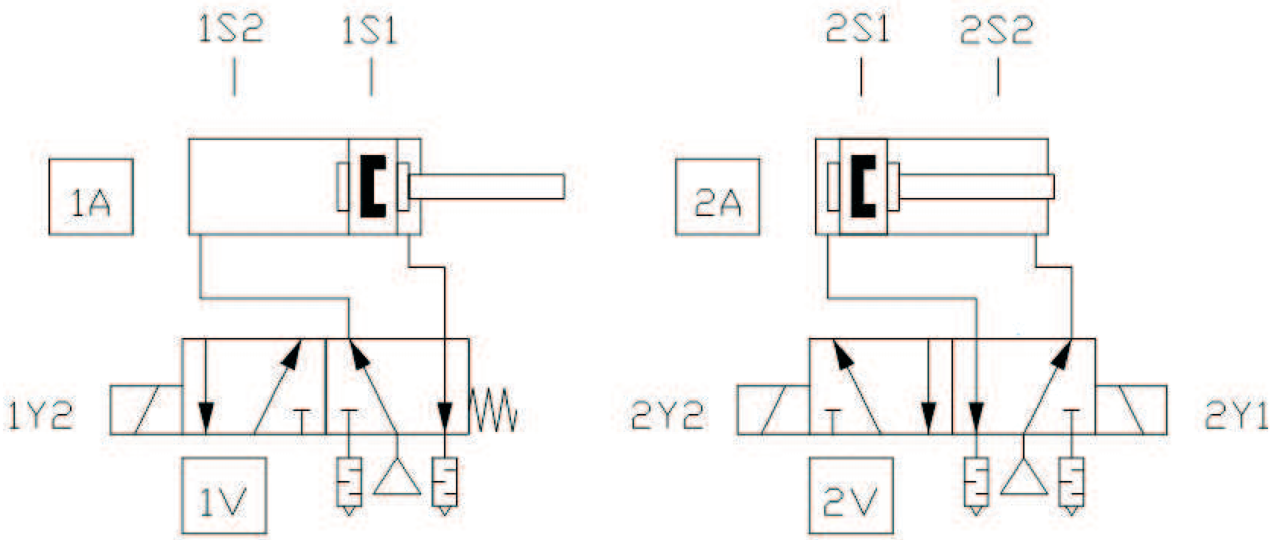

Fig. 4. Exemplary diagram of pneumatic drives 1A and 2A

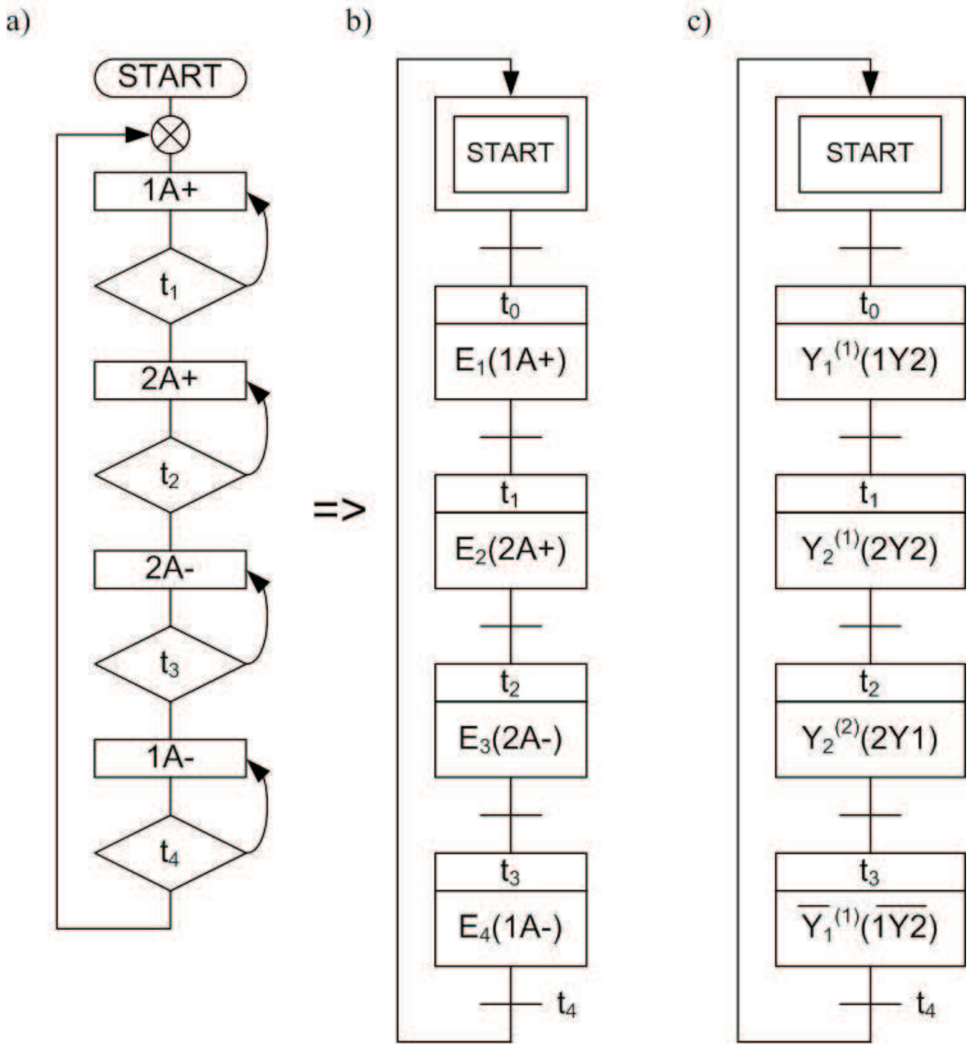

Fig. 5. Mathematical model of operation algorithm of two pneumatic drives 1A-2A: a) operational network, b) Grafpol GP network, where 1A+ means movement of the first pneumatic drive from its initial position and 1A- means movement of the first drive back to its initial position, c) Grafpol GS network, where $Y_1^{(1)}$ – output variable controlling movement of the first pneumatic drive from its initial position, $Y_1^{(2)}$ – output variable controlling movement of the first pneumatic drive back to its initial position, 1Y2 – application current to coil 1Y2,$\overline{1Y2}$ – remove current from coil 1Y2

2.1. Synthesis of a schematic equation of sequential control algorithms

Synthesis of a schematic equation (application program for PLC) is performed on the ground of the control algorithm represented by the Grafpol GS network. The following principles are obligatory during the synthesis:

Principle 1

Number of elementary memory cells used for determining a schematic equation is determined by the following relationship:

$$L = \frac{\sum_{i=1}^{n} S_i}{2} \tag{1}$$

where:

L – number of elementary memory cells,

n – number of control algorithm steps, equivalent to the number of elementary stages of the process algorithm,

S_i – i-th step of the control algorithm.

Principle 2

To determine the output variables Y of the control algorithm, it is necessary to know the transitions t_i^* in which the memory is considered.

Principle 3

The memory cells are written after the steps controlling the initial stages executed by individual pneumatic, hydraulic or electric drives are completed. These stages are described by the transitions t_i signalling completion of the steps S_i.

Individual elementary memory cells (M) which will be used for determining the output variables of the control system (Y) are written in the following way:

$$\begin{aligned} &\boldsymbol{M_1}(S) = t_1, \\ &\boldsymbol{M_j}(S) = t_i \cdot M_{j-1} \cdot t_{i-1}, \\ &\vdots \\ &\boldsymbol{M_L}(S) = t_k \cdot M_{L-1} \cdot t_{k-1} \end{aligned} \tag{2}$$

where:

$\boldsymbol{M}_j$ – j-th elementary memory cell,

M_j – output signal of the j-th elementary memory cell.

All the elementary memory cells are erased in the last state of the control algorithm, when the transition t_n^* is as follows:

$$t_n^* = t_n \cdot M_L, \tag{3}$$

Therefore, erasing all the elementary memory cells is described by the relationship:

$$\boldsymbol{M}_{1,2,\cdots,L}(R) = t_n \cdot M_L, \tag{4}$$

where:

t_n – transition describing the last state of the control algorithm.

Principle 4

Entry in any memory cell results in the following:

- All the transitions t_i^* preceding the one in that the memory cell was written till the transition in that the preceding memory cell was written (including its record state) have the form:

$$t_i^* = t_i \cdot \overline{M_L} \tag{5}$$

- All the transitions t_i^* following the one in that the memory cell was written till the transition in that the subsequent memory cell was written (including its record state) have the form:

$$t_i^* = t_i \cdot M_L \tag{6}$$

2.2. Synthesis of a schematic equation of concurrent control algorithms

Synthesis of a schematic equation of concurrent control algorithms is determined by the following principles:

Principle 1

Each sequential procedure is considered irrespective of the others.

Principle 2

According to the principles of synthesising a control algorithm using the Grafpol method for individual sequential procedures, conditions of writing and erasing individual elementary memory cells are determined and functions of output variables are defined.

Principle 3

Form of the transition t_0^* determining coordinates of starting the sequential procedures is described by the relationship:

$$t_0^* = t_{1,0}^* \cdot t_{2,0}^* \cdot \ldots \cdot t_{k,0}^* \tag{7}$$

where:

t_0^* – zero transition of concurrent procedure,

$t_{k,0}^*$ – zero transition of k-th concurrent procedure.

Principle 4

Erasing all the elementary memory cells used in the sequential procedures occurs always at the last state of the control algorithm and is determined by the following relationship:

$$\sum_{i=1}^{k} M_{i,1\ldots L}(R) = t_{1,n}^* \cdot t_{2,n}^* \cdot \ldots \cdot t_{k,n}^* \tag{8}$$

where:

L – number of elementary memory cells of j-th sequential procedure,

$M_{i,1\ldots L}(R)$ – relationship determining erasing all the elementary memory cells used in the i-th sequential procedure.

$t_{k,n}^*$ – last transition of k-th concurrent procedure.

3. Example

Practical use of the Grafpol method of modelling and programming PLC controllers was presented using a turbine mixer MTP-3000 made by Technical. The functional layout of the machine is shown in Fig. 6. Layout of the MTP-3000 turbine mixer actuators and sensors is shown in Fig. 7.

Because of continuous operation since starting-up the mixer, the components marked 1, 2 and 3 are not considered at the process modelling stage. Operation of the remaining sub-assemblies is performed and signalled in the following way:

- discharge flap (4) driven by a hydraulic cylinder controlled by a 4/2 monostable solenoid valve (coil marking 4Y2, signal “closed” 4S1, signal “open” 4S2),
- water feeder (5) controlled by a 2/2 monostable solenoid valve (coil marking 5Y2) and flowmeter with A/D converter (signal “set water volume metered” 5S3),
- strain-gauge weighing machine for mix (6) driven by a pneumatic cylinder controlled by

a 5/2 monostable solenoid valve (coil marking 6Y2, signal “closed” 6S1, signal “open” 6S2, signal “mix batch weighed” 6S3),

- mix feeder (7) to the weighing machine driven by two motors controlled by a monostable contactor (coil marking 7Y2, signal “motor stop” 7S1),
- strain-gauge weighing machine for mix (8) driven by a pneumatic cylinder controlled by a 5/2 monostable solenoid valve (coil marking 8Y2, signal “closed” 8S1, signal “open” 8S2, signal “mix batch weighed” 8S3),
- additives feeder (9) to the weighing machine driven by two motors controlled by a monostable contactor (coil marking 9Y2, signal “motor stop” 9S1).

Moreover, a strain gauge 4S3 is used for signalling emptying the pan and a thermocouple 0S1 for temperature measurement of the stirred mix.

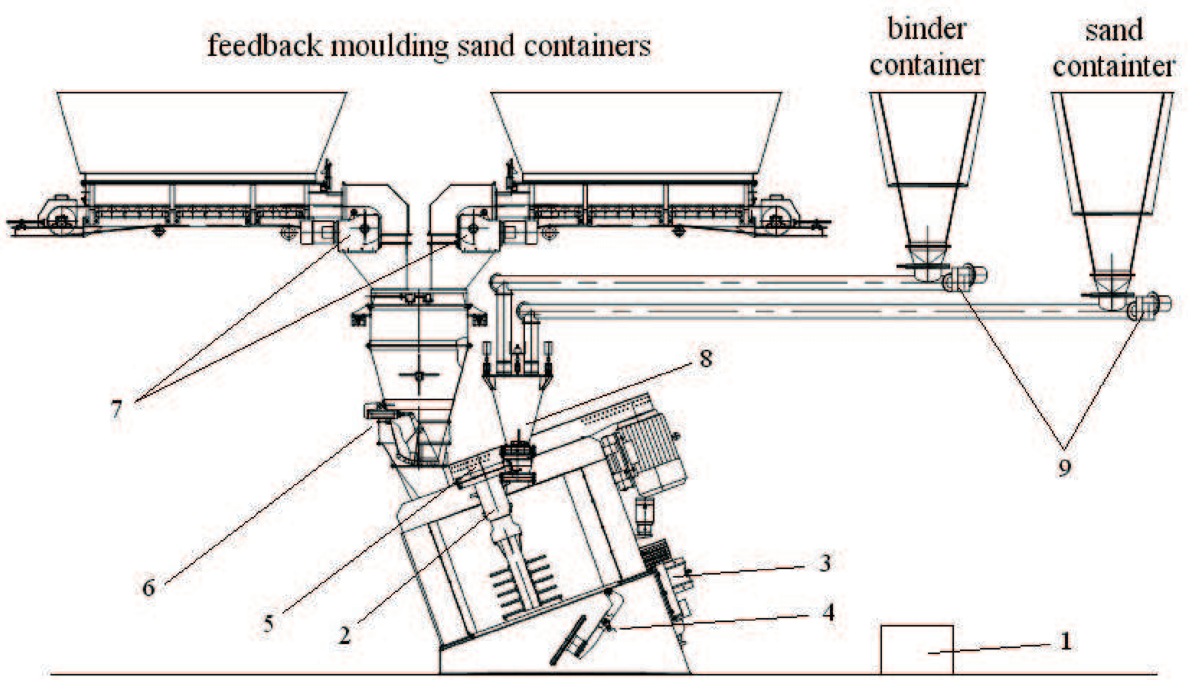

Fig. 6. Functional layout of the MTP-3000 turbine mixer: 1 – hydraulic power pack, 2 – turbine with drive, 3 – pan with drive, 4 – discharge flap, 5 – water feeder, 6 – strain-gauge weighing machine for mix, 7 – mix feeder to weighing machine, 8 – strain-gauge weighing machine for additives, 9 – additive feeder to weighing machine

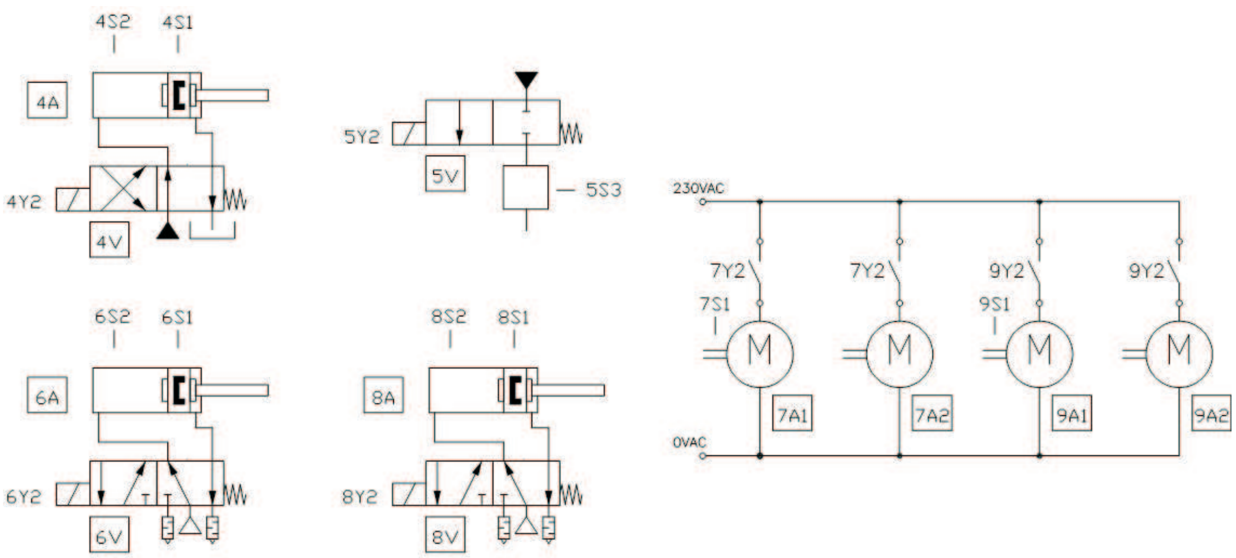

Fig. 7. Layout of the MTP-3000 turbine mixer actuators and sensors

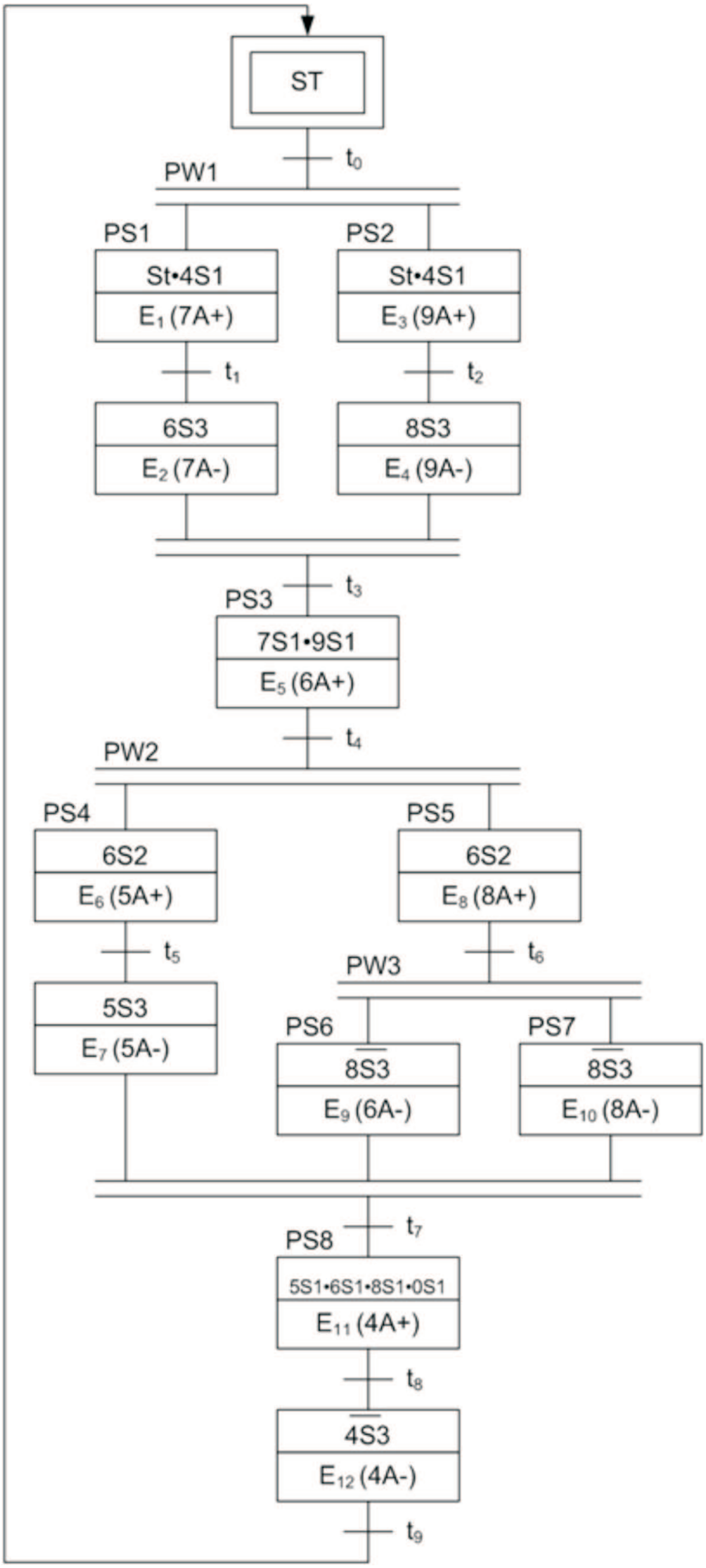

Fig. 8. Process algorithm written using Grafpol GP network, where signalling individual process stages is as follows: 4A+ – discharge flap open, 4A- – discharge flap closed, 5A+ – water feeder open, 5A- – water feeder closed, 6A+ and 8A+ – weighing machines open, 6A- and 8A- – weighing machines closed, 7A+ and 9A+ – dosing mix and additives to weighing machines open, 7A- and 9A- – dosing mix and additives to weighing machines closed

The mixer is to work according to the process algorithm shown in Fig. 8, which can be classified as a complex procedure. It consists of eight sequential procedures (PS1-PS8) creating three concurrent procedures (PW1-PW3).

In order to determine a schematic equation for the above example, the process algorithm was represented by a control algorithm in the following way presented equation 9.

Using the principles of synthesising a schematic equation of sequential and concurrent control algorithms, a control algorithm together with memories was realised, written by means of the Grafpol GS network, see Fig. 9.

$$
\begin{array}{l|l|l|l}
7Y2 \Leftarrow Y_7^{(1)} \Leftrightarrow E_1, & \overline{9Y2} \Leftarrow \overline{Y_9^{(1)}} \Leftrightarrow E_4, & \overline{5Y2} \Leftarrow \overline{Y_5^{(1)}} \Leftrightarrow E_7, & \overline{8Y2} \Leftarrow \overline{Y_8^{(1)}} \Leftrightarrow E_{10}, \\
\overline{7Y2} \Leftarrow \overline{Y_7^{(1)}} \Leftrightarrow E_2, & 6Y2 \Leftarrow Y_6^{(1)} \Leftrightarrow E_5, & 8Y2 \Leftarrow Y_8^{(1)} \Leftrightarrow E_8, & 4Y2 \Leftarrow Y_4^{(1)} \Leftrightarrow E_{11}, \\
9Y2 \Leftarrow Y_9^{(1)} \Leftrightarrow E_3, & 5Y2 \Leftarrow Y_5^{(1)} \Leftrightarrow E_6, & \overline{6Y2} \Leftarrow \overline{Y_6^{(1)}} \Leftrightarrow E_9, & \overline{4Y2} \Leftarrow \overline{Y_4^{(1)}} \Leftrightarrow E_{12},
\end{array} \tag{9}
$$

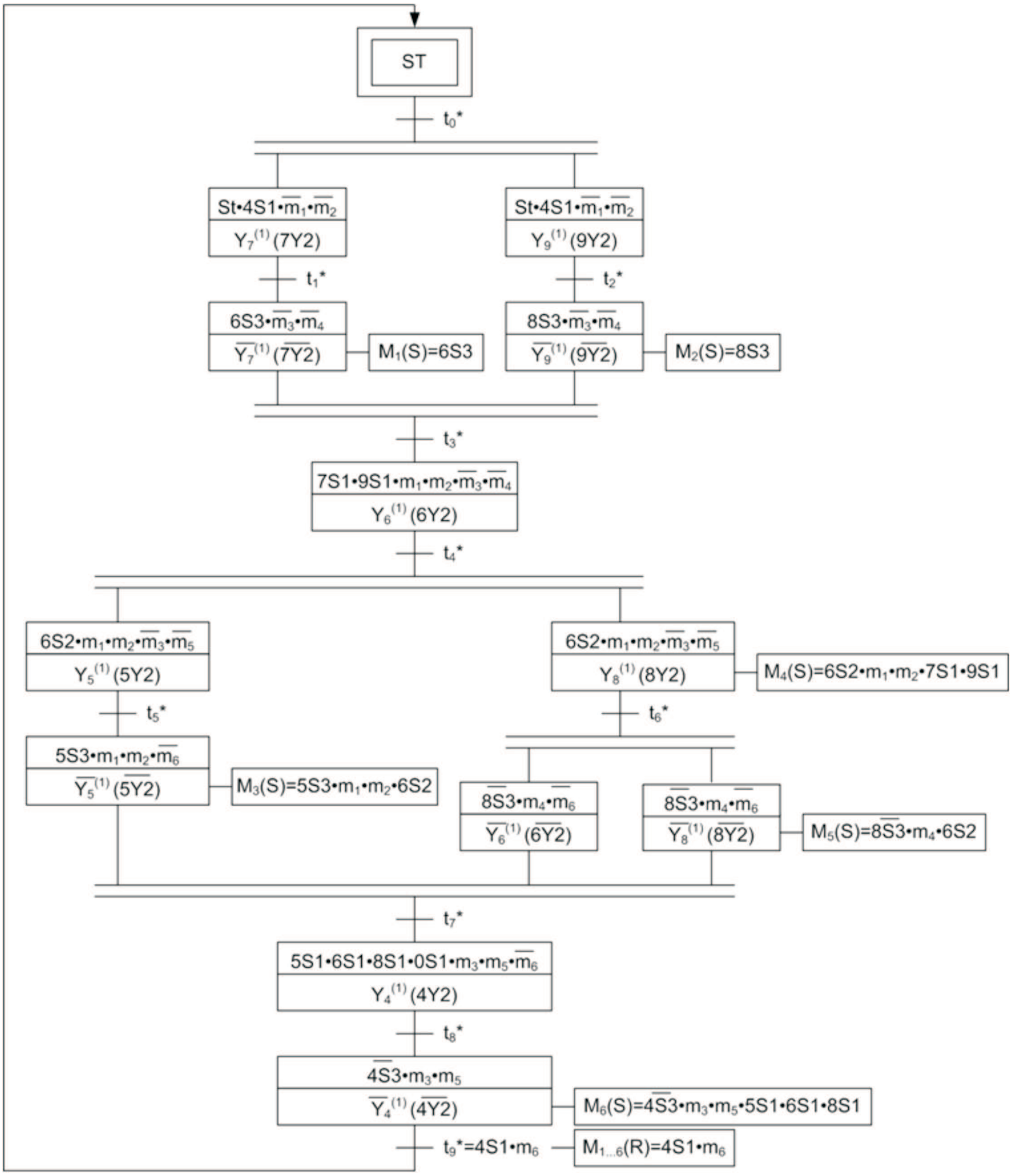

Fig. 9. Control algorithm with related memory written by means of Grafpol GS network; t_i^* – i-th transition with considered memory

$$F(Y,M) = \sum \begin{cases} ST \cdot 4S1 \cdot \overline{m_1} \cdot \overline{m_2} \cdot \left[Y_7^{(1)}(S) + Y_9^{(1)}(S)\right], \\ 6S3 \cdot \overline{m_3} \cdot \overline{m_4} \cdot Y_7^{(1)}(R), \\ 8S3 \cdot \overline{m_3} \cdot \overline{m_4} \cdot Y_9^{(1)}(R), \\ 7S1 \cdot 9S1 \cdot m_1 \cdot m_2 \cdot \overline{m_3} \cdot \overline{m_4} \cdot Y_6^{(1)}(S), \\ 6S2 \cdot m_1 \cdot m_2 \cdot \overline{m_3} \cdot \overline{m_5} \cdot \left[Y_5^{(1)}(S) + Y_8^{(1)}(S)\right], \\ 5S3 \cdot m_1 \cdot m_2 \cdot \overline{m_6} \cdot Y_5^{(1)}(R) \\ \overline{8S3} \cdot m_4 \cdot \overline{m_6} \cdot \left[Y_6^{(1)}(R) + Y_8^{(1)}(R)\right], \\ 5S1 \cdot 6S1 \cdot 8S1 \cdot 0S1 \cdot m_3 \cdot m_5 \cdot \overline{m_6} \cdot Y_4^{(1)}(S), \\ \overline{4S3} \cdot m_3 \cdot m_5 \cdot Y_4^{(1)}(R) \\ \\ 6S3 \cdot \mathrm{M}_1(S), \\ 8S3 \cdot \mathrm{M}_2(S), \\ 5S3 \cdot m_1 \cdot m_2 \cdot 6S2 \cdot \mathrm{M}_3(S), \\ 6S2 \cdot m_1 \cdot m_2 \cdot 7S1 \cdot 9S1 \cdot \mathrm{M}_4(S), \\ \overline{8S3} \cdot m_4 \cdot 6S2 \cdot \mathrm{M}_5(S), \\ \overline{4S3} \cdot m_3 \cdot m_5 \cdot 5S1 \cdot 6S1 \cdot 8S1 \cdot \mathrm{M}_6(S), \\ 4S1 \cdot m_6 \cdot [\mathrm{M}_1(R) + \mathrm{M}_2(R) + \mathrm{M}_3(R) + \mathrm{M}_4(R) + \mathrm{M}_5(R) + \mathrm{M}_6(R)] \end{cases} \tag{10}$$

On the ground of the Grafpol GS network and considering the fact that all the actuators are controlled in a monostable way [3], the schematic equation (10) was determined.

The developed schematic equation (10) makes a ground for writing a control algorithm together with the related memory as an application program of the PLC controller.

4. Summary

Presented is a procedure of fast and easy determining a schematic equation based on the Grafpol method, for an example of a turbine mixer. Thanks to the developed rules, the time for synthesising a schematic equation for the considered process algorithm was ca. 10 min. This means a significant reduction of the synthesis time in comparison to the time-transformation method (MTS) used till now for memory realisation by the Grafpol method [1], as well as by the other synthesis methods [5]. An important benefit of the developed method is also the necessity of using a smaller number of memory cells in comparison to the Grafcet method [6] and the possibility of writing the developed schematic equation as an application program of the PLC by means of any single language defined in PN-EN 61131-3.

Received: 10 July 2010.

REFERENCES

[1] T. Mikulczyński, Z. Samsonowicz, Automation of discrete manufacturing processes: Methods of modelling discrete processes and programming PLC, Warsaw, WNT 1997 (in Polish).

[2] Ł. Dworzak, S. Ciskowski, T. Mikulczyński, M. Bogdan, Synthesis of sequential control algorithms with Grafpol method, Measurements, Automation, Robotics 13, 2, 686-694 (2009)(in Polish).

[3] Ł. Dworzak, T. Mikulczyński, Synthesis of sequential control algorithms for pneumatic drives controlled by monostable valves, Archives of Foundry Engineering **9**, 3, 35-40 (2009).

[4] Ł. Dworzak, T. Mikulczyński, J. Barycki, Grafpol synthesis of concurrent control algorithms of pneumatic drives controlled with bistable pneumatic valves. Hydraulic and pneumatic drives 2009: The national sector in turbulent market conditions: International Scientific-Technical Conference, Wroclaw, 7-9 October 2009. Wroclaw, Personnel Training Centre SIMP [2009], pp. 66-73 (in Polish).

[5] J. Siwiński, Switching systems in automation, Warsaw, WNT 1980 (in Polish).

[6] Z. Banaszak, A. Drzazga, J. Kuś, Methods of interactive modelling and programming discrete processes, Wroclaw, Wroclaw University of Technology Editorial Office 1993 (in Polish).