Design of control algorithm for coal filling bolt in a power plant boiler

Łukasz Dworzak[1]

Abstract: *The article presents the steps followed for the development of the schematic equation by new rules for memory realization in Grafpol method. These steps are presented on the example of coal filling bolt in a power plant boiler.*

Keywords: *control algorithm, PLC programming, Grafpol method, sequential procedure*

1. Introduction

In times of global economic crisis and increasingly shorter product life cycles, the main feature that gives a competitive advantage is the ability to fast and flexible production setup. It also involves the need to reprogram the control system of production machines. This necessitates the need to search for a fast, reliable and simple for everyday use methods to develop control algorithms for the new processes. Classical methods for the synthesis of control algorithms [1,2] are impractical for everyday use, and the resulting solution is not to use them fully exploit the opportunities offered by today's control systems implemented by the PLC.

An alternative to these methods are Grafcet [3,4] and SFC [5] methods that are easy to use but have a few restrictions. The main disadvantage is the possibility of implementing only in some PLC. This is because the resulting code classified and unknown. In addition, the program wrote using Grafcet or SFC language requires a lot of computing power.

The solution to the above disadvantages has to be Grafpol method [6]. Due to complex and time-consuming way to implement memory it not found wider application. However, steps have been taken to develop new rules for the memory realization, the result of which would Grafpol method make easier and faster to use.

2. Grafpol method

Developed the early 90s of the 20th century Grafpol method [6] is used for modeling and programming sequential and concurrent procedures. Method strictly determined algorithm (Fig.1) from the definition of the actuator sequence to the final solution in the form of schematic equation that can be written as a program of the PLC.

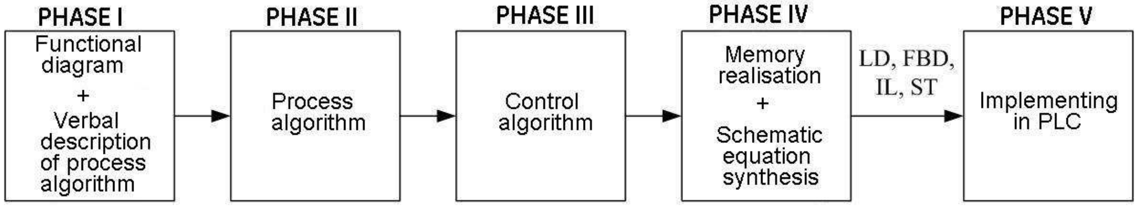
PHASE I
Functional diagram + Verbal description of process algorithm
PHASE II
Process algorithm
PHASE III
Control algorithm
PHASE IV
Memory realisation + Schematic equation synthesis
LD, FBD, IL, ST
PHASE V
Implementing in PLC

Fig. 1. Modelling and programming phases of sequential procedures by Grafpol method

[1] Wrocław University of Technology, Institute of Production Engineering and Automation (I-24),
Łukasiewicza 5, 50-371 Wrocław, lukasz.dworzak@pwr.wroc.pl

In previous Grafpol method form memory was realized in the fourth stage by the Transformation Network Method (MTS). This method requires analyzing the inputs and outputs signals of the control system, which involves a high labor-intensity.

MTS has a complicated realization of memory, monostable valves and the lack of implementation time-steps. Therefore, the work was undertaken and aimed at replacing the MTS by solution faster to implement and having a wider range of applications. As a result of the work developed new rules of memory realization in Grafpol method. The new rules are applicable to sequential and concurrent procedures with or without time steps. New rules using algorithm is shown in Fig. 2.

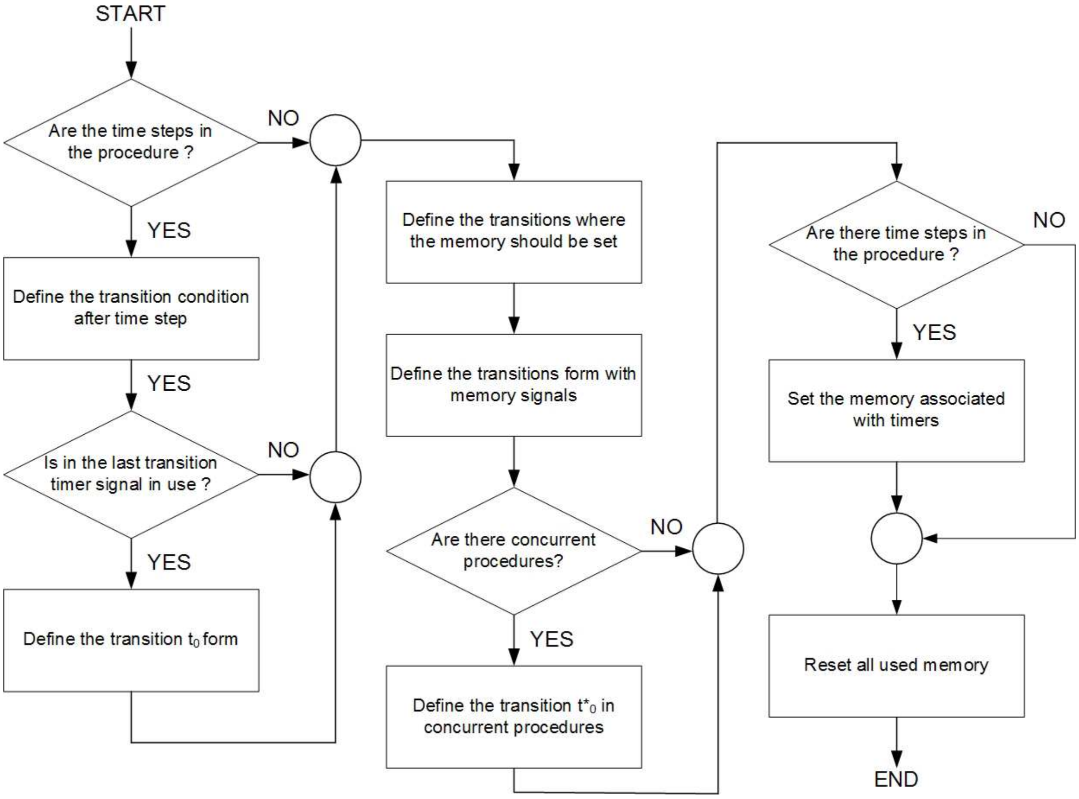
START
Are the time steps in the procedure ?
NO
YES
Define the transition condition after time step
YES
Is in the last transition timer signal in use ?
NO
YES
Define the transition t_0 form
Define the transitions where the memory should be set
Define the transitions form with memory signals
Are there concurrent procedures?
NO
YES
Define the transition t^*_0 in concurrent procedures
Are there time steps in the procedure ?
NO
YES
Set the memory associated with timers
Reset all used memory
END

Fig. 2. Algorithm of using new rules memory realization Grafpol method

2.1. Example of Grafpol method use

New rules of memory realization Grafpol method [7] will be presented as an example of the control system in the coal filling bolt in a power plant boiler. The bolt is used to shut off the filling of coal to the boiler and consists of two pneumatic drives. Drive 1A closes the bolt and drive 2A1 clamp the bolt after closing.

2.1.1. Functional diagram

Functional diagram outlines the process the device for which we will develop schematic equation in an initial state. Scheme shall contain all the signaling and control elements enabling clear definition of input and output devices. Functional diagram for coal filling bolt is shown in Fig. 3.

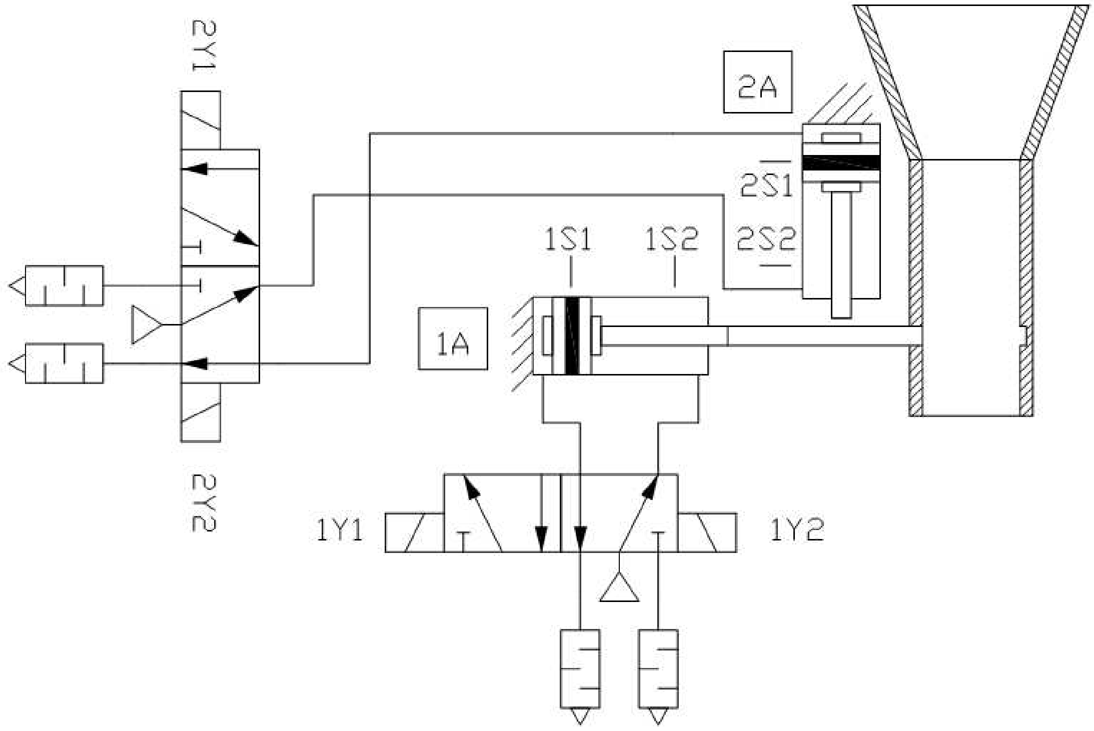

Fig. 3. Functional diagram for coal filling bolt

2.1.1. Process and control algorithm

Based on the functional diagram and description of the work we define the process algorithm, which is written by a network Grafpol GP (Fig. 4a). Already at this stage, we can take advantage of the new rules of memory realization define the transition form after time stage. The character of this transition is defined according to the principle: in transition after time stage initialization (t_{j+1}) of the k-timer we used k-timer output in the form of:

$$t_{j+1} = TON_k Q \tag{1}$$

On the basis of the process algorithm it is possible to develop a control algorithm by mapping the stages of the steps (Fig. 4b). The control algorithm is the basis for memory realization and synthesis schematic equation.

2.1.1. Memory realization and schematic equation synthesis

With the control algorithm (Fig. 4b) we begin to realize the schematic equation according to the new rules of memory realization (Fig.3).

First we specify the transitions in which the memory cells should be set according to the principle: If the actuator move from starting position is done the memory cell is set.

Conditions under which the memory cells should be set describe transition t_i. Equations describing the elementary memory cells have the following form:

$$\begin{aligned} & M_1(S) = t_1, \\ & M_2(S) = t_i \cdot m_1 \cdot t_{i-1}, \\ & \vdots \\ & M_L \cdot (S) = t \cdot \cdot m_{L-1} \cdot t_{k-1} \end{aligned} \tag{2}$$

In our example memory should be set in step 2 and 4.

Now we set the forms of transition with set memory signals according to the principle: If *j*-th memory cell (*Mj*) is set then:

- all transitions before the transitions in which there was a set of *j*-th memory cell until transition where previous memory cell was set or zero transition, has the form:

$$t_i{}^* = t_i \cdot \overline{m_j} \cdot \ldots \tag{3}$$

- all transitions following the transition in which there was a set of *j*-th memory cell until transition where next memory cell is set or last transition, has the form:

$$t_i{}^* = t_i \cdot m_j \cdot \ldots \tag{4}$$

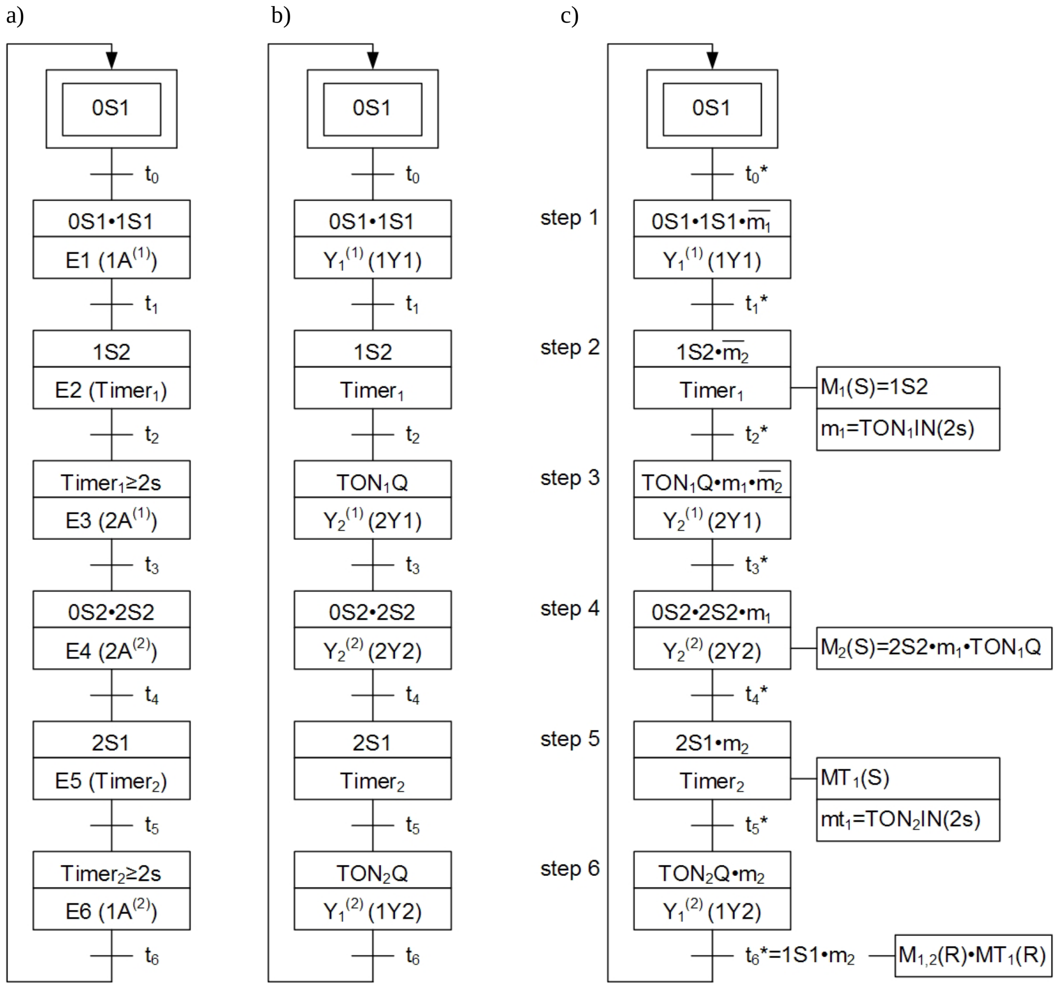

Fig. 4. a) Process algorithm, b) Control algorithm c) Control algorithm with realized memory and timers

Then we associate the memory cells with the timers according to the rule:
If the condition t_j (before the step where *k*-timer is initialized) is true then the associated with a timer memory ($\mathrm{MT_k}$) should be set.

If in this step some memory cell is already set then we use these memory as associated with timer. The associated with timers memory signals are used to initialize the timer. In this example, the associated memory should be saved in step 2 and 5. Because in step 2 is already set memory cell M_1 there is no need to set a new memory cell.

Finally, we define the form of the last transition, which is responsible for the reset of all used memory cell according to the rule:

If last transition condition with memory is true (t_n^*) then reset of all used memory cells.

The equation describing the memory cells reset is as follows:

$$t_n{*} = t_n \cdot m_L = M_1(R) \cdot M_2(R) \cdot \ldots \cdot M_L(R) \cdot MT_1(R) \cdot MT_2(R) \cdot \ldots \cdot MT_k(R) \qquad (5)$$

As a result of the action we get control algorithm with realized memory, time steps, and timer (Fig. 4c). On this base we can define schematic equation. For example it is shown as follows:

$$F(Y,M) = \sum \begin{cases} 0S1 \cdot 1S1 \cdot \overline{m_1} = Y_1^{(1)} \equiv 1Y1 \\ TON_1Q \cdot m_1 \cdot \overline{m_2} = Y_2^{(1)} \equiv 2Y1 \\ 0S2 \cdot 2S2 \cdot m_1 = Y_2^{(2)} \equiv 2Y2 \\ TON_2Q \cdot m_2 = Y_1^{(2)} \equiv 1Y2 \\ 1S2 = M_1(S) \\ 2S2 \cdot m_1 \cdot TON_1Q = M_2(S) \\ 2S1 \cdot m_2 = MT_1(S) \\ 1S1 \cdot m_2 = M_1(R) \cdot M_2(R) \cdot MT_1(R) \\ m_1 = TON_1IN(2s) \\ m_2 = TON_2IN(2s) \end{cases} \qquad (6)$$

3. Summary

The ability to quickly create control algorithms and the implementation of memory is essential for rapid set-up of production. The paper presents new rules of memory realization and synthesis of schematic equation in Grafpol method. Developed procedure and rules for the memory realization is shown for example the control system of coal filling bolt in power plant (sequential procedure with time-steps).

Compared with the original method of memory realization (Network Transformation Method – MTS) new rules of memory realization are characterized by a lack of analyze the in and out control signals, what simplifies implementation and significantly reduces the time to develop the schematic equation. The new rules allow the synthesized schematic equation for procedures involving time stages.

Grafpol method with the new memory realization rules has become very similar to the Grafcet and SFC methods. Although Grafpol is less advanced than the Grafcet and SFC, has several advantages over them. The main is transparency in the form of the schematic equation. This enables the implementation in almost any PLC and any language compatible with the PN-EN 61131-3.

Bibliography

[1] Misiurewicz P.: Układy automatyki cyfrowej. Wydawnictwa Szkolne i Pedagogiczne, Warszawa 1978

[2] Siwiński J.: Układy przełączające w automatyce. Wydawnictwa Naukowo-Techniczne, Warszawa 1980

[3] Banaszak Z., Drzazga A., Kuś J.: Metody interakcyjnego modelowania i programowania procesów dyskretnych. Wydawnictwo Politechniki Wrocławskiej, Wrocław, 1993

[4] Norma PN-EN 60848:2003 Język specyfikacyjny GRAFCET do schematów funkcji sekwencyjnych

[5] Norma PN-EN 61131-3 Sterowniki programowalne - Część 3: Języki programowania

[6] Mikulczyński T.: Automatyzacja procesów produkcyjnych – metoda modelowania procesów dyskretnych i programowania sterowników PLC. WNT, Warszawa 2006

[7] Dworzak Ł.: Nowe zasady syntezy metodą grafpol sekwencyjnych układów sterowania / Praca doktorska. Promotor: prof. dr hab inż. Tadeusz Mikulczyński. Politechnika Wrocławska, 2012 http://www.dbc.wroc.pl/publication/19355

PROJEKTOWANIE ALGORYTMU STEROWANIA ZASUWĄ PŁYTOWĄ ZASYPU WĘGLA W ELEKTROWNI

Jednym z czynników pozwalających uzyskać przewagę konkurencyjną jest zdolność do szybkiego przezbrajania maszyn produkcyjnych i programów nimi sterujących. Dlatego istotną kwestią jest dysponowanie metodami umożliwiającymi proste i szybkie syntezowanie niezawodnych równań schematowych, które opisują algorytmy sterowania.

Klasyczne metody syntezy równań schematowych opracowane zostały głównie do realizacji układów sterowania wykonanych jako układy stykowo-przekaźnikowe. Dlatego wraz z rozwojem sterowników PLC nie wykorzystują one ich możliwości w pełni, a ponadto są czasochłonne.

Alternatywę dla klasycznych metod stanowią metody Grafcet, SFC i Grafpol. Pomimo zalet metod Grafcet i SFC, takich jak szeroki zakres zastosowań oraz intuicyjność, ich zasadniczą wadą jest możliwość implementacji wyłącznie w wybranych sterownikach. Niedogodność ta nie występuje w przypadku metody Grafpol, która umożliwia implementację otrzymanego równania schematowego w dowolnym sterowniku. Dotychczas jednak pamięć realizowana była za pomocą Metody Transformacji Sieci, która jest pracochłonna. Dlatego też podjęto prace mające na celu opracowanie zasad umożliwiających realizację pamięci w sposób prostszy i szybszy niż w przypadku MTS. W wyniku prowadzonych prac wyznaczono nowe zasady realizacji pamięci metody Grafpol, które przedstawiono w artykule na przykładzie układu sterowania zasuwą zasypu węgla do kotła w elektrowni.

Opracowane rozwiązanie, w porównaniu z MTS, charakteryzuje się brakiem konieczności analizy przebiegu sygnałów wejściowych i wyjściowych oraz umożliwia syntezę równania schematowego także dla procedur zawierających etapy czasowe. W odniesieniu do metod Grafcet i SFC opracowane zasady umożliwiają opracowanie równania schematowego w jawnej postaci.