

Krzysztof Oprzędkiewicz*

Implementacja specjalnych algorytmów regulacji ciągłej na platformie „soft PLC” zrealizowanej na komputerze PC**

1. Uwagi wstępne

Systemy sterowania bazujące na koncepcji PLC (*Programmable Logic Controller*) są stosowane w praktyce przemysłowej od wielu lat. Powody ich szerokiego rozpowszechnienia są znane, znane są również ich wady, utrudniające lub wykluczające ich zastosowanie do sterowania niektórych klas procesów. Podstawową wadą większości sterowników PLC są zbyt małe możliwości obliczeniowe i zbyt mała szybkość działania jednostki centralnej sterownika – generalnie PLCs są zbyt „powolne” do sterowania procesów krytycznych pod względem czasowym (np. sterowanie numeryczne i robotyka), a próby zwiększenia możliwości obliczeniowych (ilość pamięci, szybkość działania) powodują szybki wzrost ceny urządzenia.

Z tego względu interesująca wydaje się rozwijana od kilku lat koncepcja tzw. „soft PLC”. System „soft PLC” jest systemem sterowania bazującym na środowisku MS WINDOWS i wykorzystującym interfejs procesowy z klasycznego sterownika PLC. Jako platformę sprzętową można wykorzystać każdy zarówno komputer PC, jak też inne urządzenia bazujące na WINODWS, np. panele wielofunkcyjne. W przypadku realizacji systemu „soft PLC” na komputerze klasy PC, niezbędnym elementem konfiguracji komputera jest karta komunikacyjna umożliwiająca komunikację pomiędzy „wirtualną” CPU (*Central Processor Unit*) i modułami sygnałowymi z wykorzystaniem sieci przemysłowej lub Ethernetu.

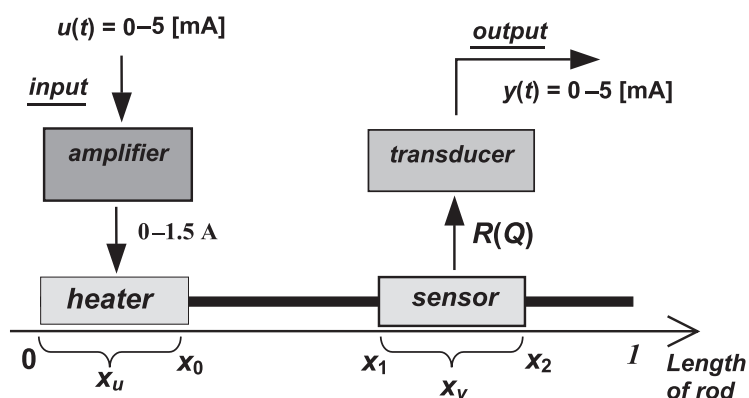
Jednocześnie należy dodać, że obecnie znaczną większość nowo budowanych złożonych systemów sterowania cyfrowego stanowią systemy bazujące na wykorzystaniu komputera klasy PC jako jednostki centralnej systemu sterowania (*PC based control*). Wśród tych systemów sterowania koncepcja „soft PLC” stanowi połączenie sprawdzonej od lat koncepcji PLC (interfejsy procesowe, metody i techniki programowania, przyzwyczajenia użytkowników) z najnowszymi trendami rozwojowymi w technice sterowania cyfrowego.

* Katedra Automatyki, Akademia Górniczo-Hutnicza w Krakowie

** Praca została zrealizowana w ramach projektu nr 3T11A00730

2. Obiekt regulacji i jego modele matematyczne

Do badań testowych jako obiekt regulacji wykorzystano laboratoryjny obiekt cieplny (rys. 1), omawiany w wcześniejszych pracach autora, np. [8, 10]. Obiektem regulacji jest pręt miedziany długości 260 mm i średnicy 2 mm ogrzewany na jednym końcu elektrycznym elementem grzejnym mającym postać spirali grzejnej nawiniętej (poprzez izolator ceramiczny) na pręt przy jednym z jego końców. Pręt nie jest izolowany cieplnie z boków i wymiana ciepła zachodzi wzdłuż całej jego długości. Element grzejny sterowany jest sygnałem prądowym $0 \div 1,5$ A 24 V DC z wzmacniacza mocy. Sygnałem wejściowym wzmacniacza mocy jest sygnał znormalizowany prądowy z zakresu $0 \div 5$ mA. Temperatura pręta jest odczytywana czujnikiem rezystancyjnym nawiniętym na pręcie w połowie jego długości. Zmiana rezystancji czujnika jest odczytywana w układzie mostkowym. Napięcie z przekątnej mostka o zakresie $0 \div 4$ mV jest przetwarzane na sygnał prądowy znormalizowany z zakresu $0 \div 5$ mA z wykorzystaniem przetwornika APU-11.



Rys. 1. Uproszczony schemat obiektu regulacji

W rozważanym przypadku jako obiekt regulacji należy traktować pręt wraz z czujnikami, przetwornikiem, wzmacniaczem mocy i elementem grzejnym, i w związku z tym sygnały: wejściowy i wyjściowy obiektu mają charakter elektryczny. Procesy fizyczne zachodzące w rozważanym eksperymentalnym obiekcie stanowią dobry przykład kilku klas rzeczywistych procesów przemysłowych, podlegających sterowaniu – pierwszą z nich są procesy cieplne, a inną – procesy z opóźnieniem.

Równanie przewodnictwa cieplnego

Po dokonaniu analizy zjawisk fizycznych zachodzących w rozważanym obiekcie, możemy stwierdzić, że jest on systemem o parametrach rozłożonych z rozłożoną obserwacją i sterowaniem. Obiekt ten może być opisany równaniem różniczkowym cząstkowym typu parabolicznego z jednorodnym warunkiem początkowym i jednorodnymi warunkami brzegowymi Neumana na końcach oraz boczną wymianą ciepła wzdłuż długości pręta. Dynami-

ka procesów termicznych zachodzących w obiekcie może być opisana układem równań (1). W układzie tym dla uproszczenia opisu przyjęto, że długość pręta jest równa 1

$$\begin{cases} \frac{\partial Q(x, t)}{\partial t} = a \cdot \frac{\partial^2 Q(x, t)}{\partial x^2} - R_a Q(x, t) + b(x)u(t), & 0 \leq x \leq 1, t \geq 0 \\ \frac{\partial Q(0, t)}{\partial x} = 0, & t \geq 0 \\ \frac{\partial Q(1, t)}{\partial x} = 0, & t \geq 0 \\ Q(x, 0) = 0, & 0 \leq x \leq 1 \\ y(t) = y_0 \int_0^1 Q(x, t)c(x)dx, & t \geq 0 \end{cases} \quad (1)$$

gdzie:

$Q(x, t)$ – temperatura pręta w chwili $t > 0$ i w punkcie $x \in [0, 1]$,
 $R_a, a > 0, b, c$ – funkcje elementu grzejnego oraz czujnika pomiarowego mające postać impulsu prostokątnego:

$$b(x) = \begin{cases} 1 & \text{dla } x \in (0, x_0) \\ 0 & \text{dla } x \notin (0, x_0) \end{cases} \quad \text{przy czym } x_0 = 1/13 \quad (2)$$

$$c(x) = \begin{cases} 1 & \text{dla } x \in (x_1, x_2) \\ 0 & \text{dla } x \notin (x_1, x_2) \end{cases} \quad (3)$$

gdzie: $y_0 = 25,7922$, $x_1 = 25/52$, $x_2 = 27/52$.

Wartości liczbowe współczynników x_0, x_1, x_2 opisują położenie i wymiary elementu grzejnego i czujnika w obiekcie.

Powyższe równanie przewodnictwa cieplnego może być zapisane w równoważnej postaci jako abstrakcyjne nieskończenie wymiarowe równanie stanu w przestrzeni Hilberta (zob. [10]). W sytuacjach praktycznych (np. dla celów symulacji komputerowej z wykorzystaniem pakietu MATLAB) stosuje się skończenie wymiarowe aproksymacje równania abstrakcyjnego. Wtedy operatory A, B i C mogą być interpretowane jako macierze, a skończenie wymiarowa aproksymacja równania abstrakcyjnego ma postać skończenie wymiarowego liniowego równania stanu o następującej postaci:

$$\begin{cases} \dot{Q}(t) = A Q(t) + B u(t) \\ y(t) = C Q(t) \\ Q(0) = 0 \end{cases} \quad (4)$$

gdzie: $Q(t) \in X_a \subset L^2(0, 1)$, $u(t) \in \mathbf{R}$, $y(t) \in \mathbf{R}$.

Bazą przestrzeni stanu X_a jest następujący układ wektorów

$$h_i = \begin{cases} 1, & i = 0 \\ \sqrt{2} \cos(i\pi x), & i = 1, 2, \dots, N \end{cases} \quad (5)$$

gdzie N – wymiar skończenie wymiarowej aproksymacji równania abstrakcyjnego.

Wartości własne macierzy stanu A są zdefiniowane następująco:

$$\lambda_i = -a\pi^2 i^2 - R_a, \quad i = 0, 1, \dots, N \quad (6)$$

Macierze A , B , oraz C układu są zdefiniowane następująco:

$$A = \text{diag}\{\lambda_1 \dots \lambda_N\} \quad (7)$$

$$B = [b_0 \dots b_N]^T \quad (8)$$

gdzie:

$$b_i = \int_0^1 b(x) h_i(x) dx \quad (9)$$

$$C = [c_0 \dots c_N] \quad (10)$$

gdzie

$$c_i = \int_0^1 c(x) h_i(x) dx \quad (11)$$

Wystarczającą dokładność rozwiązania układu (4) w porównaniu z rzeczywistym obiektem zapewnia przyjęcie $N = 25$ (wymiar aproksymacji) oraz parametrów modelu równych: $R_a = 2,71 \cdot 10^{-2}$, $a = 9,45 \cdot 10^{-4}$. Po przyjęciu tych wartości współczynników oraz uwzględnieniu relacji (5)–(11) macierze A , B oraz C przyjmą następujące wartości liczbowe (zob. [10]):

$$A = \text{diag} \{ -0,0271 \quad -0,0364 \quad -0,0644 \quad -0,1110 \quad -0,1763 \quad -0,2603 \quad -0,3629 \quad -0,4841 \\ -0,6240 \quad -0,7826 \quad -0,9598 \quad -1,1556 \quad -1,3702 \quad -1,6033 \quad -1,8551 \quad -2,1256 \quad -2,4148 \\ -2,7225 \quad -3,0490 \quad -3,3941 \quad -3,7578 \quad -4,1402 \quad -4,5413 \quad -4,9610 \quad -5,3993 \},$$

$$B = [0,0769 \quad 0,1077 \quad 0,1046 \quad 0,0995 \quad 0,0926 \quad 0,0842 \quad 0,0745 \quad 0,0638 \quad 0,0526 \quad 0,0412 \\ 0,0299 \quad 0,0190 \quad 0,0090 \quad 0,0000 \quad -0,0077 \quad -0,0139 \quad -0,0187 \quad -0,0218 \quad -0,0234 \\ -0,0235 \quad -0,0223 \quad -0,0200 \quad -0,0168 \quad -0,0130 \quad -0,0087]^T,$$

$$C = [0,9920 \quad 0,0000 \quad -1,3995 \quad 0,0000 \quad 1,3893 \quad 0,0000 \quad -1,3724 \quad 0,0000 \quad 1,3489 \quad 0,0000 \\ -1,3191 \quad 0,0000 \quad 1,2832 \quad 0,0000 \quad -1,2415 \quad 0,0000 \quad 1,1944 \quad 0,0000 \quad -1,1423 \quad 0,0000 \\ 1,0856 \quad 0,0000 \quad -1,0248 \quad 0,0000 \quad 0,9605].$$

Po dyskretyzacji modelu (4)–(11) otrzymuje się równoważny model dyskretny o następującej postaci:

$$Q^+(i+1) = A^+Q^+(i) + B^+u^+(i), \quad y^+(i) = C^+Q^+(i) \quad (12)$$

gdzie:

$$i = 0, 1, 2, \dots, A^+ = T(h), \quad B^+ = \int_0^h T(t)Bdt, \quad C^+ = C \quad (13)$$

przy czym $T(t) = e^{At}$, jest macierzą tranzycyjną układu (4) $h > 0$ jest okresem pracy ekstrapolatora i impulsatora.

W rozważanym przypadku, przy założeniu okresu próbkowania 5 s, dyskretne operatory stanu i sterowania będą mieć postać:

$$A^+ = \text{diag}\{0,8733 \ 0,8335 \ 0,7247 \ 0,5740 \ 0,4141 \ 0,2722 \ 0,1629 \ 0,0889 \ 0,0442 \ 0,0200 \\ 0,0082 \ 0,0031 \ 0,0011 \ 0,0003 \ 0,0001 \ 0,0000 \ 0,0000 \ 0,0000 \ 0,0000 \ 0,0000 \ 0,0000 \\ 0,0000 \ 0,0000 \ 0,0000 \ 0,0000\},$$

$$B^+ = [0,3597 \ 0,4924 \ 0,4471 \ 0,3818 \ 0,3077 \ 0,2354 \ 0,1718 \ 0,1201 \ 0,0806 \ 0,0515 \\ 0,0308 \ 0,0164 \ 0,0065 \ 0,0000 \ -0,0041 \ -0,0066 \ -0,0077 \ -0,0080 \ -0,0077 \\ -0,0069 \ -0,0059 \ -0,0048 \ -0,0037 \ -0,0026 \ -0,0016]^T.$$

Modele transmitancyjne

Równoważnym pod względem funkcjonalnym, a znacznie prostszym modelem rozważanego obiektu jest model w postaci transmitancji zastępczej, której parametry mogą być wyznaczone na podstawie odpowiedzi skokowej obiektu. Klasyczne przykłady modeli transmitancyjnych to model Kupfmuellera z opóźnieniem i model Strejca bez opóźnienia. Model zastępczy Kupfmuellera w postaci transmitancji obiektu inercyjnego I i II rzędu z opóźnieniem jest opisany następująco:

$$G_1(s) = \frac{ke^{-s\tau_1}}{Ts+1} \quad (14)$$

$$G_2(s) = \frac{ke^{-s\tau_2}}{(T_1s+1)(T_2s+1)} \quad (15)$$

Model Strejca w postaci transmitancji n -tego rzędu bez opóźnienia jest opisany następująco

$$G_3(s) = \frac{k}{(T_n s + 1)^n} \quad (16)$$

Wartości liczbowe współczynników transmitancji (14)–(16) zależą w rozważanym wypadku od lokalizacji i wymiarów elementu grzejnego oraz czujnika pomiarowego. Mogą

one być wyznaczone na podstawie odpowiedzi skokowej rzeczywistego obiektu z wykorzystaniem metody najmniejszych kwadratów. Dla rozważanego przypadku (lokalizacja i wymiary czujnika i elementu grzejnego podane w wzorach (2) i (3)) są one podane w tabeli 1.

Tabela 1
Wartości parametrów transmitancji zastępczych obiektu

Parametr	Wartość
k	1,1841
τ_1 [s]	21,9778
τ_2 [s]	12,19
T [s]	41,6128
T_1 [s]	14,07
T_2 [s]	44,17
T_n [s]	22,08
n	3

Powyższe modele, zarówno równanie stanu, jak i model transmitancyjny, po dyskretyzacji są stosowane do budowy rozważanych specjalnych algorytmów sterowania.

3. Testowane algorytmy specjalne

W ramach badań testowych przeprowadzono badania następujących algorytmów specjalnych:

Kompensator dyskretny

Zasadniczą ideą kompensatora dyskretnego jest kompensacja najsłabiej tłumionej (leżącej najbliżej brzegu okręgu jednostkowego) wartości własnej systemu dyskretnego. Dokładna analiza i opis działania kompensatora jest zamieszczony m.in. w pracach: [7–9]. Algorytm regulacyjny jest opisany następującym dyskretnym równaniem stanu:

$$\begin{bmatrix} w_1^+(i+1) \\ w_2^+(i+1) \end{bmatrix} = A_k^+ \cdot \begin{bmatrix} w_1^+(i) \\ w_2^+(i) \end{bmatrix} + B_k^+ [y^+(i) + M^+ r] \quad (17)$$

$$u^+(i) = K_1^+ w_1^+(i) + N^+ r$$

gdzie:

$$A_k^+ = \begin{bmatrix} A_1^+ - G_1^+ C_1^+ + B_1^+ K_1^+ & -G_1^+ C_2^+ \\ B_2^+ K_1^+ & A_2^+ \end{bmatrix},$$

$$B_k^+ = \begin{bmatrix} G_1^+ \\ 0 \end{bmatrix},$$

$y(i)$ – wyjście obiektu,

r – wartość zadana,

$w(i)$ – dyskretny wektor stanu kompensatora,

$u(t)$ – sterowanie wyznaczone przez kompensator i podawane na obiekt.

Parametry równania stanu kompensatora (17) są wyznaczone na podstawie dyskretnego równania stanu obiektu. Równanie (17) może być użyte do realizacji cyfrowej kompensatora na platformie PLC. Realizacja tego algorytmu na różnych platformach cyfrowych omówiona jest w pracach: [11, 12].

Predyktor Smitha

Algorytm ten jest jednym z najczęściej stosowanych w praktyce przemysłowej algorytmów specjalnych. Jest on omówiony np. w pracach: [4, 19]. Ma on postać dodatkowego, wewnętrznego sprzężenia zwrotnego, którym objęty jest regulator konwencjonalny (np. PID).

Do budowy predyktora Smitha wykorzystuje się model zastępczy w postaci transmitancji z opóźnieniem (14) lub (15), zapisanej w postaci dyskretniej. Sprzężenie zwrotne jest tak skonstruowane, że regulator jest w stanie „przewidzieć” zachowanie się układu. Jednocześnie należy dodać, że w przypadku realizacji tego układu na platformie PLC największe trudności napotyka realizacja sprzężenia zwrotnego. Problemy realizacji predyktora Smitha na PLC są przedstawione w pracach [15, 17].

Regulator redukcyjny

Zasadniczą ideą regulatora redukcyjnego jest kompensacja („redukcja”) transmitancji obiektu regulacji lub też najslabiej tłumionej jego części. Główną trudność podczas praktycznej realizacji regulatora redukcyjnego stanowi realizowalność fizyczna transmitancji regulatora. Trudność ta została ominięta w algorytmie zaproponowanym przez R. Góreckiego [2] i zmodyfikowanym przez autora pod kątem jego realizacji na PLC [13]. Do budowy tego algorytmu wykorzystuje się model zastępczy Strejca bez opóźnienia opisany przez (16).

W wersji zaproponowanej przez autora algorytm ten jest realizowany jako szeregowe połączenie 2 bloków, regulatora PI oraz korektora, które są opisane następującymi równaniami dyskretnymi:

$$x(n) = Ke(n) + I \sum_{i=0}^n e(i) \quad (18)$$

$$u(n) = \frac{a_{n-1}}{a_n} u(n-1) - \frac{a_{n-2}}{a_n} u(n-2) + \frac{b_n}{a_n} x(n) - \frac{b_{n-1}}{a_n} x(n-1) + \frac{b_{n-2}}{a_n} x(n-2) \quad (19)$$

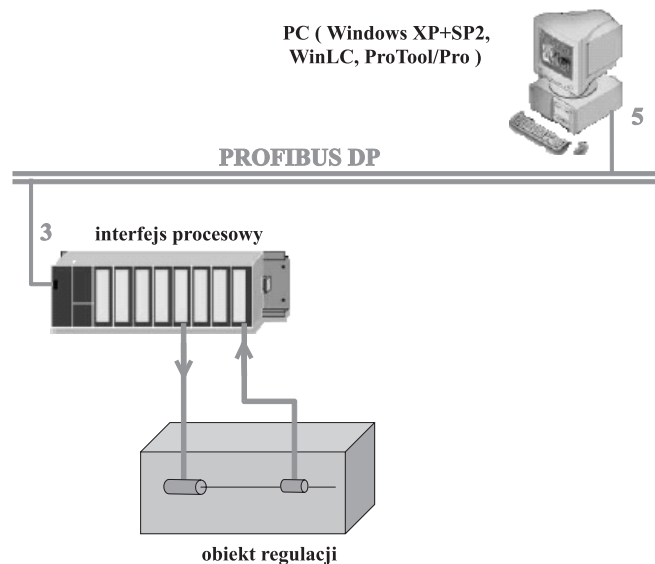
W wzorach (18) i (19) $e(n)$ oznacza dyskretny uchyb regulacji w chwili n , $u(n)$ oznacza sterowanie w chwili n , a oraz b są współczynnikami określonymi następująco:

$$\begin{aligned} a_n &= \frac{(\eta T_n)^2}{T_p^2} + \frac{3\eta T_n}{T_p} + 3, & b_n &= \frac{T_n^2}{T_p^2} + \frac{2T_n}{T_p} + 1 \\ a_{n-1} &= \frac{2(\eta T_n)^2}{T_p^2} + \frac{3\eta T_n}{T_p}, & b_{n-1} &= \frac{2T_n^2}{T_p^2} + \frac{2T_n}{T_p} \\ a_{n-2} &= \frac{(\eta T_n)^2}{T_p^2}, & b_{n-2} &= \frac{T_n^2}{T_p^2} \end{aligned} \quad (20)$$

W relacjach (20) η oznacza współczynnik redukcji dynamiki obiektu. Przy założeniu dokładnej znajomości modelu obiektu w postaci transmitancji zastępczej (16), stała czasowa transmitancji wymuszeniowej całego układu regulacji, zawierającego obiekt i regulator jest równa ηT_n .

4. Platforma sprzętowo-programowa do realizacji sterowania

Architektura systemu sterowania „soft PLC” zrealizowanego na komputerze PC pokazana jest na rysunku 2.



Rys. 2. Schemat sprzętowy układu sterowania „soft PLC” zrealizowanego na komputerze PC; 3, 5 – adresy elementów w sieci PROFIBUS

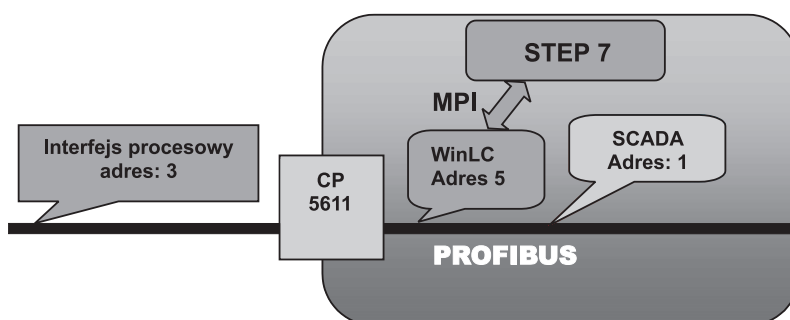
Zasadnicza różnica w stosunku do systemów „hard PLC” jest taka, że CPU sterownika jest zrealizowana **programowo** na komputerze PC, natomiast moduły sygnałowe są te same, co w przypadku systemu „hard PLC”. W tej konfiguracji komputer musi być wyposażony w kartę komunikacyjną umożliwiającą komunikację z modułami sygnałowymi. Takie rozwiązanie pozwala na radykalne zwiększenie możliwości obliczeniowych sterownika w stosunku do systemu „hard PLC”, gdyż „wirtualna CPU” ma pełny dostęp do wszystkich zasobów systemu. Łatwiejsza jest także budowa systemu SCADA, gdyż jest on zintegrowany z CPU na jednej platformie sprzętowej. W tym miejscu należy dodać, że sterownik „soft PLC” może być zrealizowany nie tylko w środowisku komputera PC, lecz także np. w środowisku wielofunkcyjnego panelu operatorskiego.

Najbardziej istotną cechą wszystkich omówionych powyżej rozwiązań jest to, że do budowy aplikacji we wszystkich sytuacjach wykorzystuje się to samo środowisko programowe, co w przypadku systemu „hard PLC” i przeniesienie aplikacji z systemu „hard PLC” do systemu „soft PLC” jest niezwykle proste i wymaga zwykle tylko przededefiniowania konfiguracji sprzętu i (ewentualnie) zmiany adresów wejść i wyjść sterownika.

Do realizacji układu został wykorzystany komputer PC o następujący podstawowych parametrach:

Procesor	Intel Celeron 2.8 GHz
Pamięć RAM	512 MB
Karta komunikacyjna PROFIBUS	CP 5611.

Schemat logiczny komunikacji w obrębie rozważanego systemu „soft PLC” podany jest na rysunku 3. Najbardziej istotną cechą rozważanego układu jest mechanizm wymiany danych SOFTBUS. Zasadnicza jego idea polega na tym, że poszczególne komponenty układu sterowania są połączone z sobą z wykorzystaniem sieci PROFIBUS, która jest częściowo „fizyczna” a częściowo wirtualna. „Wirtualna” sieć łączy komponenty systemu sterowania zlokalizowane w obrębie platformy sprzętowej komputera PC i z punktu widzenia komunikacji nie ma rozróżnienia pomiędzy „rzeczywistą” i „wirtualną” częścią sieci.



Rys. 3. Schemat logiczny oprogramowania i komunikacji w obrębie systemu „soft PLC”

Do realizacji omawianego systemu „soft PLC” zastosowano oprogramowanie opisane w tabeli 2.

Tabela 2
Oprogramowanie użyte do realizacji systemu

Oprogramowanie	Funkcja
Step 7 ver 5.4 Professional	Oprogramowanie konfiguracyjne i diagnostyczne
ProTool/Pro ver 6.0	Oprogramowanie SCADA
WinLC ver 4.0	Wirtualna CPU sterownika „soft PLC” (wersja podstawowa <u>bez jądra czasu rzeczywistego</u>)
Simatic Net	Oprogramowanie konfiguracyjne dla sieci

Powyższe oprogramowanie pracuje pod nadzorem systemu WINDOWS XP Prof.+SP2. Komunikacja pomiędzy SIMATIC MANAGEREM i WinLC odbywa się za pomocą programowego (wewnętrznego) interfejsu MPI, komunikacja pomiędzy pozostałymi komponentami systemu odbywa się z wykorzystaniem sieci PROFIBUS, przy czym w obrębie komputera odbywa się z użyciem mechanizmu SOFTBUS. Podczas budowy systemu program dla sterownika jest zintegrowany w obrębie jednego projektu z systemem SCADA.

5. Uwagi o metodach realizacji rozważanych algorytmów sterowania w środowisku PLC

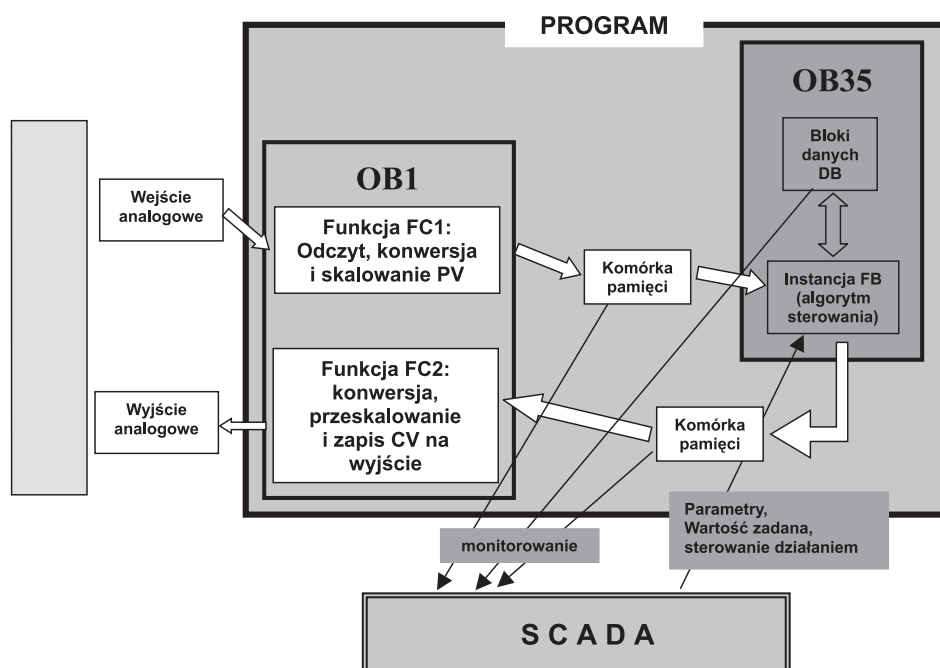
Szczegółowe przedstawienie zasad realizacji omawianych wcześniej algorytmów na platformie sprzętowo-programowej „soft PLC” jest zamieszczone m.in. w pracach: [16, 18]. Poniżej podane zostaną ogólne zasady dotyczące realizacji specjalnych algorytmów regulacyjnych na PLC, sformułowane na podstawie doświadczeń autora.

Ogólny schemat realizacji specjalnego algorytmu regulacji ciągłej w środowisku PLC podany jest na rysunku 4.

Wyznaczanie sterowania zgodnie z podanym schematem obejmuje zadania opisane w tabeli 3.

System SCADA pełni następujące funkcje:

- monitorowanie i nadzór działania (wyświetlanie wartości i trendów wielkości regulowanej, uchybu regulacji, sterowania i wartości zadanej, alarmowanie w przypadku przekroczenia dopuszczalnych wartości przez zmienne procesowe);
- ustawianie wartości zadanej i parametrów liczbowych algorytmów;
- sterowanie pracą algorytmów (przełączanie trybu pracy, zatrzymywanie i uruchamianie aplikacji sterującej, uruchamianie i zatrzymywanie archiwizacji itp.).



Rys. 4. Ogólny schemat realizacji rozważanych specjalnych algorytmów sterowania na platformie „soft PLC”

Tabela 3

Zadania realizowane podczas wyznaczenia sygnału sterującego

Zadanie	Elementy oprogramowania	OB nadzorujący
1. Odczyt wejścia		OB1
2. Konwersja i skalowanie wielkości regulowanej PV do zakresu 0,0–100,0	Funkcja FC1: wejście to sygnał typu WORD z zakresu 0–6912, wyjście to sygnał typu REAL z zakresu 0,0–100,0%	OB1
3. Wyznaczenie sygnału sterującego zgodnie z zadanym algorytmem	Bloki funkcyjne: FB1, FB2, SFB41 i stowarzyszone z nimi bloki danych: DB1, DB2, DB3; Blok danych DB4 zawierający parametry regulatora	OB.35
4. Konwersja sygnału sterującego typu REAL z zakresu 0,0– 100,0% do postaci typu WORD z zakresu 0–6912	Funkcja FC2: wejście to sygnał typu REAL z zakresu 0,0–100,0%, wyjście to sygnał typu WORD z zakresu 0–6912	OB1
5. Zapis wyjścia		OB1

Do wymiany danych pomiędzy elementami programu wykorzystano zmienne bezpośrednio reprezentowane (*Directly Represented Variables*). Są to zmienne, których nazwy są adresami komórek pamięci wewnętrznej sterownika. Z ich użyciem jest realizowany jeden z opisanych w normie IEC 1131-3 mechanizmów wymiany danych w obrębie programu (zob. np. [6]).

Definicje funkcji i bloków funkcyjnych zostały napisane z użyciem języka strukturalnego wysokiego poziomu STEP7 SCL, bloki organizacyjne OB1 i OB35 zostały oprogramowane z użyciem języka drabinkowego LD (*Ladder Diagram*).

Użycie obu ww. języków jest uzasadnione następująco:

- Język wysokiego poziomu STEP7 SCL jest „przyjazny” dla użytkownika, pozwala na łatwą realizację złożonych operacji obliczeniowych i zapewnia dobre własności oprogramowania w zakresie spełnienia wymagań czasu rzeczywistego (zob. [19]). Wynika to z faktu, że elementy napisane z użyciem tego języka zawsze są kompilowane do postaci Listy instrukcji.
- Język drabinkowy LD z kolei posiada jednoznaczną i przejrzystą interpretację oraz dobrze zdefiniowaną obsługę błędów, co pozwala na łatwą realizację sekwencyjnego wykonania programu zgodnie ze schematem pokazanym na rysunku 4.

6. Spełnienie wymagań czasu rzeczywistego podczas realizacji algorytmów

Jako miarę spełnienia wymagań czasu rzeczywistego podczas realizacji rozważanych specjalnych algorytmów sterowania przyjęto wartość czasu cyklu „wirtualnej CPU” podczas realizacji algorytmu (należy zwrócić uwagę, że nie jest to czas **każdego** cyklu programowego, gdyż algorytm wykonywany z założonym stałym okresem próbkowania nie jest realizowany w każdym cyklu). Pomiary czasu cyklu wykonano metodą programową opisaną w pracy [14].

Testy czasu cyklu rozważanego systemu „soft PLC” przeprowadzono przy następujących założeniach:

- w systemie działały tylko dwie aplikacje: „soft PLC” oraz system SCADA do zbierania wyników;
- jako obiekt regulacji zastosowano obiekt cieplny z opóźnieniem, omówiony powyżej;
- okres próbkowania podczas realizacji każdego algorytmu przyjęto równy 1 s;
- wydajność systemu optymalizowana na aplikacje;
- wartości czasu cyklu i czasu spoczynku sterownika (*sleep time*) sterownika przyjęto domyślne, zalecane przez producenta: minimalny czas cyklu: 0 ms, maksymalny czas cyklu: 6000 ms, czas spoczynku: 10 ms;
- pomiary przeprowadzono podczas normalnej pracy aplikacji w stanie ustalonym wielkość regulowanej (sterownik w trybie RUN-P);
- priorytety przerwań zegarowych sterownika związanych z uaktywnieniem bloku OB 35 i obsługą wejść i wyjść ustawiono na wartości domyślne;

- do pomiarów wykorzystano wewnętrzny zegar czasu rzeczywistego sterownika;
- liczbę pomiarów przyjęto równą 1000;
- badania wykonano dla trzech wartości priorytetów zadania sterownika „soft PLC” pod nadzorem WINDOWS:
 - niskiego równego 1 (najniższy priorytet),
 - difoltowego równego 8,
 - wysokiego równego 25.

Zmierzone wartości czasu cyklu dla rozważanych algorytmów specjalnych są podane w tabelach 4–6.

Tabela 4

Zmierzone wartości czasu cyklu dla układu z kompensatorem dynamicznym

Parametr	Priorytet	1	8	25
Min T_s [ms]	wartość	11	11	11
	liczba wystąpień	375	213	217
Najczęstszy T_s [ms]	wartość	12	12	12
	liczba wystąpień	543	674	783
Max T_s [ms]	wartość	129	70	12
	liczba wystąpień	2	3	783
Rozrzut między min i max wartością T_s [ms]		118	59	1

Tabela 5

Zmierzone wartości czasu cyklu dla układu z predykatorem Smitha

Parametr	Priorytet	1	8	25
Min T_s [ms]	wartość	10	10	10
	liczba wystąpień	215	154	347
Najczęstszy T_s [ms]	wartość	11	11	11
	liczba wystąpień	375	844	653
Max T_s [ms]	wartość	76	32	11
	liczba wystąpień	5	2	653
Rozrzut między min i max wartością T_s [ms]		65	21	1

Tabela 6
Zmierzone wartości czasu cyklu dla układu z regulatorem redukcyjnym

Parametr	Priorytet	1	8	25
Min T_s [ms]	wartość	11	11	11
	liczba wystąpień	875	993	1000
Najczęstszy T_s [ms]	wartość	11	11	11
	liczba wystąpień	875	993	1000
Max T_s [ms]	wartość	96	35	11
	liczba wystąpień	2	3	1000
Rozrzut między min i max wartością T_s [ms]		85	24	0

7. Uwagi końcowe

Uwagi końcowe do pracy mogą być sformułowane następująco:

- W żadnym z testowanych przypadków maksymalna zmierzona wartość czasu cyklu (pojedyncze wystąpienia i najniższy priorytet) była kilkakrotnie mniejsza niż okres próbkowania algorytmu, najczęściej była ok. 100 razy mniejsza.
- Podniesienie priorytetu zadania „soft PLC” w systemie WINDOWS znacznie skraca wartość czasu cyklu i zmniejsza rozrzut tej wartości.
- Reasumując, należy stwierdzić, że rozważany system „soft PLC” zrealizowany z użyciem podstawowej wersji oprogramowania i komputera PC klasy biurowej stanowi stabilną platformę sprzętowo-programową do realizacji specjalnych algorytmów sterowania rozważanej klasy.

Literatura

- [1] Astrom K., Wittenmark B.: *Computer controlled systems. Theory and design*. Prentice Hall Inc. 1997
- [2] Górecki R.: *Regulator redukcyjny*. Zeszyty Naukowe AGH Elektrotechnika, t. 15, z. 3, 1996, 267–279
- [3] Grega W.: *Sterowanie cyfrowe w czasie rzeczywistym*. Wyd. Wydziału EAIiE AGH, 1999
- [4] Grega W.: *Metody i algorytmy sterowania cyfrowego w układach scentralizowanych i rozproszonych*. Wyd. AGH, 2004
- [5] Isermann R.: *Digital Control Systems*. Vol. 1, Springer, 1989
- [6] Kasprzyk J.: *Programowanie sterowników przemysłowych*. Warszawa, WNT 2006
- [7] Mitkowski W.: *Stabilizacja systemów dynamicznych*. Warszawa, WNT 1991
- [8] Mitkowski W. Oprzędkiewicz K.: *Sterowanie procesem nagrzewu*. Pótrocznik AGH, Automatyka, t. 1, z. 1, 1997, 301–310

- [9] Mitkowski W. Oprzędkiewicz K.: *Skończenie wymiarowe dynamiczne sprzężenie zwrotne z ograniczeniami na sterowanie obiektu*. Kraków, Półrocznik AGH Automatyka, t. 3, z. 1, 1999, 236–242
- [10] Oprzędkiewicz K.: *Przykład identyfikacji obiektu parabolicznego*. Zesz. Nauk. AGH Elektrotechnika, t. 16, z. 2, 1997, 100–106
- [11] Oprzędkiewicz K.: *Praktyczna realizacja skończenie wymiarowego dynamicznego sprzężenia zwrotnego*. Półrocznik AGH Automatyka, t. 3, z. 1, 1999, 267–274
- [12] Oprzędkiewicz K.: *Realizacja kompensatora dynamicznego na sterowniku PLC*. PAR. Pomiary, Automatyka, Robotyka, R. 4, nr 12, 2000, 5–10
- [13] Oprzędkiewicz K.: *Dyskretny regulator redukcyjny i jego praktyczna implementacja*. Pomiary, Automatyka, Robotyka (PAR), nr 11, 2001, 8–15
- [14] Oprzędkiewicz K.: *Monitorowanie jakości regulacji w specjalnych układach sterowania*. Pomiary, Automatyka, Robotyka (PAR), nr 4, 2002, 5–7
- [15] Oprzędkiewicz K.: *Problemy realizacji cyfrowej predyktora Smitha w środowisku sterownika PLC*. Półrocznik AGH Automatyka, t. 6, z. 2, 2002, 139–150
- [16] Oprzędkiewicz K.: *Uwarunkowania czasowe realizacji specjalnych algorytmów sterowania w systemach PLC*. Pomiary, Automatyka, Robotyka (PAR), nr 4, 2003, 48–52
- [17] Oprzędkiewicz K.: *Spełnienie wymagań czasu rzeczywistego w środowisku sprzętowo-programowym „soft PLC” podczas realizacji predyktora Smitha*. Informatyka Teoretyczna i Stosowana, R. 4, nr 7, 2004, 49–58
- [18] Oprzędkiewicz K.: *Spełnienie wymagań czasu rzeczywistego na platformie sprzętowo-programowej „soft PLC” zrealizowanej na komputerze klasy PC*. AUTOMATION 2007 PIAP Warszawa, marzec 2007
- [19] Oprzędkiewicz K.: *Porównanie języków programowania sterowników PLC pod kątem spełnienia wymagań czasu rzeczywistego*. PAR, Pomiary, Automatyka, Robotyka, R. 10, nr 12, 2006, 5–9
- [20] Trybus L.: *Regulatory wielofunkcyjne*. Warszawa, WNT 1992

