Implementacja algorytmu regulacji predykcyjnej MPC w sterownikach programowalnych

Jarosław Tarnawski

Katedra Inżynierii Systemów Sterowania, Wydział Elektrotechniki i Automatyki, Politechnika Gdańska

Streszczenie: Sterowniki programowalne PLC (ang. *Programmable Logic Controllers*) są główną przemysłową platformą implementacji algorytmów sterowania bezpośredniego. Standardowo producenci PLC udostępniają programistom jedynie podstawowe algorytmy sterowania. W niniejszym artykule rozważana jest implementacja w PLC zaawansowanej metody sterowania – algorytmu MAC/MPC (ang. *Model Algorithmic Control/ Model Predictive Control*) ze względu na jego walory użytkowe oraz stosunkowo złożony proces implementacji. Regulacja MPC może być implementowana również w warstwie nadrzędnej hierarchicznego modelu systemu sterowania, dlatego prezentowany jest cały model. W artykule opisano proces weryfikacji regulatora MPC zaimplementowanego w PLC w warunkach pętli sprzętowej. Badane są czasy wykonania jednej iteracji regulatora oraz wymagania wobec wielkości pamięci PLC dla konkretnych parametrów regulatora, co może być wykorzystane do szacowania stosowalności na innych platformach i przydatności do sterowania konkretnymi obiektami. Właściwości regulatorów MPC są niedostępne dla klasycznych regulatorów (klasy PID wraz z modyfikacjami i rozwinięciami), zatem implementacja regulacji MPC w PLC istotnie rozwija możliwości sterowania na tej platformie.

Słowa kluczowe: hierarchiczny model systemu sterowania, zaawansowane algorytmy sterowania, regulacja predykcyjna MPC, sterowniki programowalne, norma IEC 61131-3, ograniczenia platformy PLC, proces implementacji algorytmu regulatora MPC, weryfikacja w pętli sprzętowej

Sterowniki programowalne opracowano w celu zastąpienia układów sterowania przełączającego, realizowanych za pomocą styczników i przekaźników. Zmiana algorytmu sterowania nie wymagała już fizycznych przełączeń elementów elektrycznych tylko zmiany programu sterującego dla PLC. Głównie do tych celów powstał język drabinkowy przypominający schematy stycznikowo-przekaźnikowe. Pierwotnie PLC obsługiwały wyłącznie sygnały dyskretne, następnie zostały wyposażone w przekaźniki czasowe i zegary umożliwiające włączanie i wyłączanie urządzeń nie tylko w zależności od stanu sygnałów pomiarowych, ale również z opóźnieniem, przesunięciem czasowym, czy o określonej godzinie. Zatem na początku PLC uwzględniały wyłącznie sterowanie binarne.

Inaczej były rozwijane systemy klasy DCS (ang. *Distributed Control System*), zorientowane od początku na realizację sterowania ciągłego: PID, MPC, sterowania rozmytego i wielu innych. Systemy klasy DCS zostały jednak ulokowane w tzw. dużym przemyśle: energetyce, petrochemii itd. PLC ze względu na elastyczność, dużą konkurencję na rynku i pozycjonowanie przez producentów można znaleźć w instalacjach przemysłowych różnej skali, od najmniejszych do bardzo dużych. Wraz z rozwojem techniki mikroprocesorowej i komputerowej PLC zostały wyposażone w podstawowy aparat matematyczny początkowo bazujący na liczbach całkowitych, a następnie zmiennoprzecinkowych. Pierwotnie pamięć PLC była widziana jako wektorowa, aktualnie może być również organizowana w postaci macierzy. Obecnie możliwości sprzętowe i programowe udostępnione przez producentów dla programistów PLC zbliżyły się do tych dostępnych w komputerach klasy PC. Różnice między systemami PLC wraz z systemem SCADA oraz DCS stały się dużo mniejsze, jednak wciąż ograniczona jest biblioteka algorytmów sterowania dostępnych standardowo w PLC. Historię rozwoju oraz szczegółową charakterystykę PLC czytelnik odnajdzie m.in. w [4, 5].

Regulacja predykcyjna ma zalety niedostępne dla prostych regulatorów realizowanych standardowo w PLC. Jej wykorzystanie w przemyśle nie jest powszechne. Główne bariery stosowania tego sposobu regulacji to: znacznie większe od metod klasycznych (np. PID), potrzeby pamięciowe oraz obliczeniowe, większa złożoność koncepcyjna, większa liczba parametrów strojenia regulatora i trudności w implementacji w docelowym urządzeniu sterującym.

W artykule zaprezentowano proces implementacji liniowego regulatora predykcyjnego MAC/MPC na platformie PLC. Większość podanych informacji może być z powodzeniem wykorzystana na innych platformach sprzętowych, takich jak mikrokontrolery oraz komputery przemysłowe.

1. Przegląd dostępnych algorytmów regulacji w PLC

Sterowniki jako urządzania programowalne oferują inżynierom predefiniowane bloki programowe: operacji logicznych, operacji i relacji matematycznych, operacji związanych z odmierzaniem czasu i zliczaniem zdarzeń, operacji tablicowych, funkcji konwersji pomiędzy typami danych, funkcji sterujących przebiegiem programu. Obsługiwane typy danych obejmują operacje na liczbach binarnych, sło-

wach bitowych, operacje na liczbach całkowitych i od kilku lat na liczbach rzeczywistych o różnej długości słowa.

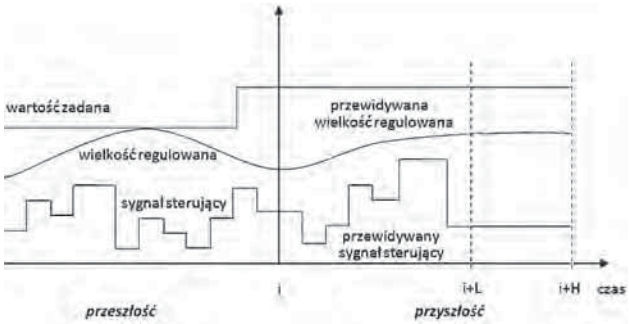
Sterowniki programowalne różnych producentów, oprócz podanych podstawowych bloków programowych, wyposażone są w podstawowe algorytmy sterowania: dwu- i trójstanowego oraz najczęściej stosowany w przemyśle regulator PID w wersji rozbudowanej, uwzględniającej np. martwą strefę, nasycenia, ograniczenia prędkości narastania sygnału sterującego i skalowanie sygnałów wartości zadanej, regulowanej i sygnału sterującego. Podane sposoby regulacji przeważnie wyczerpują ofertę standardowo udostępnioną przez producentów. W sterownikach uniwersalnych (a nie w specjalizowanych bądź rozbudowanych DCS) rzadko można napotkać zaawansowane algorytmy sterowania np. algorytmy adaptacyjne, predykcyjne, rozmyte, czy neuronowe. Najczęściej algorytmy te implementowane są na platformach komputerów osobistych ze wspomaganiem oprogramowania naukowo-badawczego i oprogramowania do szybkiego prototypowania.

Sterowniki programowalne umożliwiają wprowadzenie programu użytkownika i tym samym realizację niemal dowolnego algorytmu sterowania. Możliwa zatem jest implementacja również zaawansowanych algorytmów.

2. Charakterystyka sterowania predykcyjnego

Istnieje wiele odmian sterowania predykcyjnego jednak we wszystkich z nich można wskazać cechy wspólne:
- sterowanie z wykorzystaniem modelu dynamicznego obiektu,
- kryterium matematyczne (funkcję celu regulacji predykcyjnej),
- wydłużone horyzonty predykcji wyjść obiektu sterowania,
- strategię repetycyjną,
- wykorzystywanie trajektorii referencyjnej.

Idea sterowania predykcyjnego została przedstawiona na rys. 1. W każdej iteracji algorytmu (w każdym kroku próbkowania), bazując na modelu obiektu sterowania, bieżących wartościach na wyjściu obiektu (pochodzących z pomiarów) i przeszłych (zapamiętanych wartościach sygnałów sterujących) oraz trajektorii referencyjnej wyznaczany jest ciąg sterowań dla tzw. horyzontu sterowania – L. Sterowania wyznaczane są zgodnie z zasadą minimalizacji kryterium jakości (zależnego od rodzaju sterowania predykcyjnego), uwzględniającego m.in. różnicę między trajektorią referencyjną, a predykowanym wyjściem obiektu na tzw. horyzoncie predykcji – H. Spośród wyznaczonych sterowań uwzględniane jest tylko pierwsze, a następnie w kolejnych krokach procedura jest powtarzana.



Rys. 1. Idea sterowania predykcyjnego [1–3]
Fig. 1. Idea of model predictive control

Do zalet regulacji predykcyjnej należy zaliczyć możliwość sterowania:
- z uwzględnieniem ograniczeń zarówno na wielkości regulowane, jak i sterujące (co do amplitudy, jak i szybkości zmian),
- obiektami wielowymiarowymi,
- w obecności dużych opóźnień czasowych – obiektami niestabilnymi i nieminimalnofazowymi.

3. Rodzaje regulacji predykcyjnej

W [1–3] można znaleźć różne odmiany regulacji predykcyjnej o różnych właściwościach użytkowych oraz implementacyjnych. W [7] dokonano interesującej oceny różnych rodzajów sterowania predykcyjnego.

Dynamic Matrix Control – DMC

Bazuje na modelu odpowiedzi skokowej, jest odporny na zmiany parametrów, umożliwia wprowadzanie ograniczeń, jednak jakość sterowania jest słaba dla zakłóceń narastających liniowo oraz dla obiektów wielowymiarowych z silnymi interakcjami. Parametrami strojenia tego regulatora są: horyzont sterowania oraz współczynniki kar.

Model Algorithmic Control – MAC

Jest łatwy do zaimplementowania i znacznie lepszy niż DMC dla obiektów wielowymiarowych, ale wyłącznie stabilnych. Stosowany jest tu model odpowiedzi impulsowej. Jest odporny na błędy identyfikacji i zmiany parametrów obiektu. Jego zastosowanie jest ograniczone do stosunkowo wolnych trajektorii referencyjnych. Parametrami strojenia są: horyzont sterowania, współczynniki kar oraz współczynnik zmiany sygnału sterującego.

Predictive Functional Control – PFC

Wykorzystuje model w przestrzeni stanów, stosowany dla szybkich procesów, jest odporny na błędy modelowania, nad- i niedoparametryzowanie modelu. Nie wymaga dużych mocy obliczeniowych, ale gwarantuje dużą precyzję sterowania. Może uwzględniać ograniczenia. Zakres stosowania ograniczony jest wyłącznie do obiektów stabilnych. Strojenie odbywa się za pomocą trajektorii referencyjnej oraz doboru kroku czasowego.

Extended Predictive Self-Adaption Control EPSAC

Podstawą jest transmitancja dyskretna i proste prawo sterowania. Zawiera sprzężenie wprzód (*feedforward*) i uwzględnia mierzalne zakłócenia. Realizuje stałą wartość zadaną i radzi sobie z obiektami nieminimalnofazowymi. Strojenie *on-line* horyzontu predykcji wpływa na strukturę wielokrokowego predyktora i strukturę regulatora. Strojenie odbywa się za pomocą: horyzontu sterowania, współczynnika wagowego oraz wielomianu filtrującego.

MOST WIEDZY Pobrano z mostwiedzy.pl

Extended Horizon Adaptive Control

Bazuje na modelu transmitancyjnym. Implementacja jest łatwa, horyzont sterowania jest jedynym parametrem strojenia. Gwarantuje stabilność dla stacjonarnych obiektów, również nieminimalnofazowych.

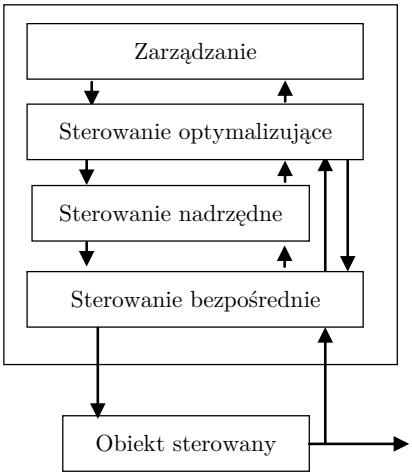
Generalized Predictive Control GPC

Umożliwia określanie stałych i danych w postaci trajektorii wartości zadanych. Niezbędne dla predykowania jest rekurencyjne rozwiązywanie równania diofantycznego. Parametry strojenia: horyzont sterowania, współczynniki kar i wag umożliwiają osiągnięcie dobrej jakości sterowania. GPC umożliwia sterowanie obiektami niestabilnymi, nieminimalnofazowymi z opóźnieniami.

4. Struktura warstwowa systemu sterowania i lokalizacja sterowania predykcyjnego

Scentralizowane sterowanie złożonymi obiektami nie jest możliwe. Ukształtowane zostało podejście hierarchiczne [2, 6], które jest realizowane w strukturze sterowania złożonej z warstw: bezpośredniej, nadrzędnej, optymalizacji i zarządzania (rys. 2). Zadania poszczególnych warstw wynikają z dekompozycji obiektu sterowania. Prostą, intuicyjną formą dekompozycji wynikającą z budowy, rozmiarów obiektu, ulokowania i podziału na sekcje jest dekompozycja przestrzenna. Bardziej złożona jest dekompozycja funkcjonalna, w której następuje wydzielenie celów cząstkowych (umożliwiających realizację celu podstawowego) możliwych do zrealizowania w różnych skalach czasu i przez różne urządzania, z różnym udziałem człowieka.

W warstwie sterowania bezpośredniego, znajdującej się najbliżej obiektu sterowania i pracującej z częstotliwością dostosowaną do dynamiki obiektu i potrzeb układów sterowania i zabezpieczeń, przeważnie realizowane są proste algorytmy. Dominują układy przełączające, logika IF-THEN, regulatory dwu- i trójstanowe oraz regulacja PID. Typowymi przedstawicielami urządzeń pracujących w tej warstwie są mikrokontrolery, sterowniki PLC i PAC, regulatory wielofunkcyjne.



Rys. 2. Hierachiczna komputerowa struktura sterowania
Fig. 2. Hierarchical computer control structure

W warstwie sterowania nadrzędnego realizowane są funkcje nadzoru wielkości jakościowo opisujących sterowany proces, często inne niż te regulowane w warstwie bezpośredniej. Rozwój tej warstwy nastąpił wraz z rozwojem sterowania predykcyjnego [2]. W wyniku rozwiązania zadania zdefiniowanego w postaci funkcji celu (definiującego jakościowe wymagania wobec systemu sterowania) wyznaczane są trajektorie, które następnie przekazywane są do warstwy sterowania bezpośredniego, jako wartości zadane do zrealizowania przez podstawowe algorytmy sterowania, np. PID. Warstwa sterowania nadrzędnego najczęściej realizowana jest w ramach systemów SCADA. Należy podkreślić [2], że warstwa sterowania nadrzędnego może być pominięta, a jej zadania rozdzielone między warstwę bezpośrednią i optymalizującą. Opisy zadań warstw optymalizującej i zarządzania oraz rozbudowane rozważania o systemach hierarchicznych omówiono w [2, 6].

Sterowanie MPC początkowo było implementowane głównie w warstwie nadrzędnej, m.in. ze względu na ograniczenia urządzeń sterujących. Obecnie, ze względu na znacznie większe możliwości przechowywania danych, prowadzenia obliczeń i swobody programowania w językach wysokiego poziomu w docelowych, przemysłowych urządzeniach sterujących, algorytmy MPC mogą być realizowane również w warstwie sterowania bezpośredniego.

W artykule rozpatrywana jest implementacja regulacji MPC w warstwie sterowania bezpośredniego jako alternatywa dla powszechnych, prostych i posiadających znane ograniczenia metod klasy PID.

5. Regulacja MAC/MPC z modelem odpowiedzi impulsowej

Do ilustracji implementacji metod zaawansowanego sterowania w PLC wybrany został regulator z modelem odpowiedzi impulsowej MAC. Ponieważ nazywany jest też algorytmem MPC [1], zatem dalej będzie stosowana właśnie ta nazwa (MPC będzie oznaczało ten konkretny rodzaj regulacji predykcyjnej). Jest to przykład regulatora predykcyjnego z modelem nietransmitancyjnym. Ma to określone zalety: nie trzeba znać transmitancji obiektu ani nawet struktury tej transmitancji, model może być wyznaczony w wyniku prostego eksperymentu identyfikacyjnego jak rejestracja odpowiedzi na impuls Diraca lub za pomocą metod estymacji najmniejszych kwadratów czy gradientowych, model nie zawiera wprost wyznaczanego opóźnienia.

Przyjmujemy model postaci:

$$y(i) = Vu(i-1)$$

$$V = v_0 + v_1 z^{-1} + v_2 z^{-2} + \dots v_{nV} z^{-nV}$$

Współczynniki wielomianu V są parametrami odpowiedzi impulsowej. Model jest typu średniej ruchomej MA (*Moving Average*). W modelu tym założono, że dyskretny czas opóźnienia jest równy 1. Zakłada się więc, że pierwsze współczynniki wielomianu V mogą być równe zeru. W syntezie regulatora czas opóźnienia nie jest uwzględniany wprost, więc w postaci adaptacyjnej regulator MPC jest mało wraż-

MOST WIEDZY Pobrano z mostwiedzy.pl

liwy na zmiany czasu opóźnienia. Ta cecha jest bardzo pożądana w zastosowaniu do obiektów, w których zmiana opóźnienia transportowego jest cechą naturalną.

Celem regulacji MPC jest minimalizacja różnicy między przewidywaną trajektorią sygnału wyjściowego i trajektorią odniesienia z uwzględnieniem wagi na odchylenie sterowania od wartości $u(i-1)$. Należy więc wyznaczyć taką wartość sygnału sterującego, aby osiągnąć minimum wskaźnika:

$$J = \sum_{j=1}^{H} \left\{ \left[\overline{y}(i+j) - \overline{w^0}(i+j) \right]^2 + \rho \delta u^2(i+j-1) \right\} \tag{1}$$

gdzie:

$\overline{y}(i+j)$ – wyznaczona w chwili i predykcja wartości sygnału wyjściowego na chwilę $(i+j)$,
$\overline{w^0}(i+j)$ – wartość trajektorii referencyjnej w chwili $(i+j)$,
ρ – parametr strojenia, określający wagę przyrostu sterowania we wskaźniku,
H – horyzont predykcji,
L – horyzont sterowania, w obrębie którego uwzględnia się zmiany przyszłego sygnału sterującego. Zakłada się, że począwszy od chwili $i+L$ przyrosty sygnału sterującego będą równe zeru.
$\delta u(i+j)$ – zmiana sygnału sterującego wg (2)

$$\delta u(i+j) = u(i+j) - u(i-1) \tag{2}$$

Algorytm regulacji [1] minimalizujący podany wskaźnik dany jest wyrażeniem

$$u(i) = u(i-1) + q^T(\overline{w}^0 - \overline{y}^0) \tag{3}$$

gdzie:

$$q^T = [q_1, q_2, \ldots, q_H] = \overline{1}^T [Q^T Q + \rho I]^{-1} Q^T \tag{4}$$

$$\overline{w}^0 = \left[w^0(i+1)\ w^0(i+2)\ \ldots, w^0(i+H) \right]^T, \tag{5}$$

$$\overline{y}^0 = \left[\overline{y}^0(i+1)\ \overline{y}^0(i+2)\ \ldots, \overline{y}^0(i+H) \right]^T, \tag{6}$$

$$Q = \begin{bmatrix} v_0 & 0 & \cdots & 0 \\ v_1 & v_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ v_{L-1} & v_{L-2} & \cdots & h_0 \\ \vdots & \vdots & & \vdots \\ v_{H-1} & v_{H-2} & \cdots & h_{H-L} \end{bmatrix} \tag{7}$$

$$h_j = \sum_{s=0}^{j} v_s \tag{8}$$

Do wyznaczenia (predykcji) przyszłych wartości sygnału wyjściowego y korzystamy z zależności

$$\overline{y}^0(i+j) = \overline{y}^0(i+j-1) + V_j^2 \Delta u(i-1) \tag{9}$$

z warunkiem początkowym

$$y^0(i) = y(i)$$

Wielomian V_j^2 dany jest przez

$$V_j^2 = v_j + v_{j+1} z^{-1} + \ldots + v_n z^{-n+j} \tag{10}$$

6. Wymagania na parametry regulatora

Ze względu na sposób działania regulatora na parametry tego regulatora nałożone są następujące wymagania:
- czas opóźnienia obiektu nie może być większy niż $nV+1$, ponieważ model obiektu nie będzie mógł być dobrym odwzorowaniem obiektu;
- do poprawnej realizacji operacji na macierzy Q konieczne jest spełnienie warunku: $2H - L <= 2nV - 1$;
- czas opóźnienia musi być mniejszy od H, aby horyzont sterowania uwzględniał opóźnienie obiektu;
- gdy $H = L$ i $\rho = 0$, obiekt nie może być nieminimalnofazowy, opóźnienie musi być równe 1.

7. Ograniczenia implementacyjne w PLC (pamięć, moc obliczeniowa, operacje macierzowe, języki)

Realizacja algorytmu sterowania wymaga operacji macierzowych: dodawania, odejmowania, mnożenia, odwracania. Wymiar macierzy zależy od przyjętych parametrów sterowania predykcyjnego: horyzontów predykcji i sterowania oraz rzędu modelu obiektu.

Sterowniki programowalne cechują pewne ograniczenia, które należy wziąć pod uwagę podejmując temat implementacji zaawansowanych algorytmów sterowania. Tymi ograniczeniami są: wielkość pamięci dostępnej dla programu i danych użytkownika, wydajność obliczeniowa, reprezentacja liczb, możliwość operowania na macierzach.

Przyjęte założenia:
- PLC realizuje operacje matematyczne na liczbach zmiennoprzecinkowych,
- sterownik umożliwia adresowanie pamięci w postaci tablic dwuwymiarowych (nie zawsze ma bloki operacji macierzowych),
- parametry regulatora H, L, nV, ρ nie są stałymi, są wybierane przez użytkownika (program zapewnia ich obsługę, ewentualnie ostrzega o nierealizowalności ze względu na zakres pamięci i czas obliczeń).

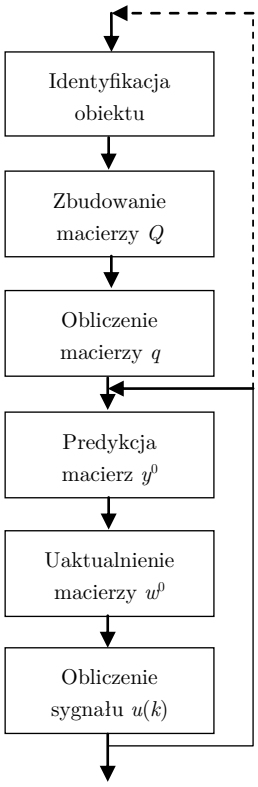
Ograniczenia platformy PLC/PAC i zasady implementacji zaawansowanych algorytmów sterowania przedstawiono w [8].

8. Implementacja w sterowniku z operacjami macierzowymi

Do implementacji opisanego regulatora MPC wybrano język ST. Realizacja algorytmu MPC w języku zdefiniowanym w normie, powinna umożliwiać jego przenoszenie na sterowniki różnych producentów, jednak w praktyce, ze względu na drobne odstępstwa od normy stosowane przez producentów, przenoszenie kodu wymaga dostosowania do specyfiki danego urządzenia.

Dodatkową zaletą języka ST jest możliwość korzystania z wysokopoziomowych konstrukcji pętli i operowania na danych w postaci macierzy. Realizację algorytmu sterowania predykcyjnego można podzielić na następujące etapy (rys. 3):
1. identyfikacja obiektu sterowania i wyznaczenie wielomianu V;

MOST WIEDZY Pobrano z mostwiedzy.pl



Rys. 3. Cykl realizacji algorytmu MPC
Fig. 3. Cycle of MPC algorithm implementation

2. zbudowanie macierzy Q wg (7) i (8);
3. obliczenie $q^T = 1^T[Q^TQ + \rho I]^{-1}Q^T$;
4. predykcja wyjść na podstawie (9) i zbudowanie (6);
5. wyznaczenie w^0 danego przez (5);
6. obliczenie sygnału sterującego (3).

Linią ciągłą (rys. 3) zaznaczone jest zapętlenie algorytmu dla prezentowanej w artykule wersji bez aktualizacji wiedzy o obiekcie, natomiast linią przerywaną wersję z aktualizacją wiedzy o obiekcie.

Identyfikację obiektu można przeprowadzić kilkoma sposobami. Można zarejestrować ciąg sygnałów sterujących oraz sygnałów wyjściowych z obiektu i korzystając z metod matematycznych dopasować parametry wielomianu tak, aby spełniały określone kryteria. Najpopularniejsze metody to metoda najmniejszych kwadratów oraz metody gradientowe. Metody te zastosowane on-line (najczęściej w postaci rekurencyjnej) mogą być wykorzystane do realizacji adaptacyjnej wersji regulacji predykcyjnej.

Alternatywą jest wyznaczenie wielomianu V za pomocą rejestracji

Tab. 1. Tablica zmiennych wykorzystanych w programie M – macierz 2-wymiarowa, W – wektor, S – skalar, R – rzeczywista, I – całkowita
Tab. 1. Program variables list M – 2 dimension matrix, W – vector, S – scalar, R – real, I – integer

Nazwa zmiennej	Org. w pamięci	Typ liczb	Wymiar	Opis
Q	M	R	H × L	Macierz Q z parametrami odpowiedzi skokowej wg (8)
QT	M	R	L × H	Transponowana Q
C	M	R	L × H	QT * Q + ro*I
A, B, X	M	R	3 × L × H	Macierze pomocnicze przy odwracaniu C; X = C^{-1}
Y	M	R	H × H	X * QT
Z	M	R	H	Pierwszy wiersz Y
roznica	W	R	H	$w^0 - y^0$MPC
V	W	R	nV	Parametry odpowiedzi skokowej
eu	W	R	nV	Opóźnione wejścia
y0MPC	W	R	H	Realizacja wektora y^0 (6) wg predyktora (9)
w0	W	R	H	Trajektoria zadana wg (5)
zadana	S	R	1	Wartość zadana w chwili i
regulowana	S	R	1	Wartość regulowana
nV	S	I	1	Wymiar V
H	S	I	1	Horyzont predykcji
L	S	I	1	Horyzont sterowania
i, j, k, m, n	S	I	5 × 1	Indeksy w pętlach
ro	S	R	1	Parametr regulatora MPC
hk	S	R	1	Parametr macierzy Q wg (8)
suma	S	R	1	Zmienna pomocnicza przy wyznaczaniu hk
wm, Max, Xmax, P, Sx, Tx, w	S	I	7 × 1	Zmienne pomocnicze przy odwracaniu macierzy

MOST WIEDZY Pobrano z mostwiedzy.pl

odpowiedzi obiektu na impuls podany na wejście obiektu. Rejestrując w chwilach próbkowania odpowiedź obiektu, można wyznaczyć parametry wielomianu V. To zaleta modelu nietransmitancyjnego. Macierz Q ma specyficzną postać, którą można otrzymać z V. Realizacja programistyczna sprowadza się do dwóch pętli adresujących położenie parametrów V oraz dodatkowej pętli, w której wyznacza się i lokuje w Q czynniki h_k.

Największym wyzwaniem programistycznym oraz czasochłonnym członem podczas wyznaczania sygnału sterującego jest obliczenie macierzy q. W wersji nieadaptacyjnej regulatora predykcyjnego wystarczy jednokrotne wyznaczenie tej macierzy i nie musi być ono realizowane w pętli programu. Może do tego posłużyć MATLAB, Scilab, a nawet Excel. Do wyznaczania wektora y^0, czyli predykcji sygnału wyjściowego korzystne jest zastosowanie wzoru rekurencyjnego. Potrzebne są również wartości przyrostów sterowań w chwilach poprzednich oraz parametry wielomianu V. Realizacja programistyczna sprowadza się do operacji mnożenia i dodawania realizowanych w pętlach.

Wartości zadane mogą być wprowadzane na kilka różnych sposobów. Można wymusić zmianę całego wektora w^0, czyli wprowadzenie nowej wartości zadanej dla całego horyzontu sterowania. Można wprowadzić wartość zadaną jako ostatni element wektora w^0, który zostanie osiągnięty po H chwilach czasowych; można również zastosować trajektorię referencyjną, tj. opis przejścia między poprzednią i nowo wprowadzoną wartością zadaną. Realizacja programistyczna tych scenariuszy jest stosunkowo prosta i sprowadza się odpowiednio do: zapisania całego wektora w^0 nową wartością zadaną, wpisania do w^0 na ostatniej pozycji i przesuwania wektora w każdej chwili czasowej, lub wpisanie do w^0 wartości wynikających z trajektorii (najczęściej inercja Igo rzędu) i przesunięcia wektora w każdej chwili działania algorytmu.

Wyznaczenie wartości $u(k)$ sprowadza się do odjęcia wektorów w^0 oraz y^0 i wymnożenie wyniku z wektorem q. Wynik tej operacji zsumowany z poprzednią wartością sygnału sterującego daje bieżąca wartość sterowania. W tym kroku można również nałożyć różnego rodzaju ograniczenia na sygnał sterujący np. co do wartości albo prędkości narastania.

9. Tablica zmiennych

Dokonując analizy danych (tab. 1) do realizacji regulatora MPC w zależności od nV, H i L można wyznaczyć rozmiar niezbędnej pamięci.

10. Weryfikacja regulatora MPC w PLC

Weryfikację implementacji regulatora zrealizowano w strukturze sprzętowo-programowej nazywanej sterowaniem w pętli sprzętowej (*hardware in the loop*). Schemat przedstawiono na rys. 4. Regulator MPC znajdował się w swoim docelowym miejscu, tj. w sterowniku programowalnym GE Fanuc RX3i z modułami we/wy analogowych ALG442, który był fizycznie połączony z komputerem PC wyposażonym w kartę akwizycji danych pomiarowych Advantech 1711 i oprogramowanie symulacyjne MATLAB/Simulink Real Time Windows Target. W tym środowisku były symulowane w czasie rzeczywistym obiekty testowe. Taka forma testowania implementacji algorytmu MPC pozwalała na sprawdzenie w docelowym urządzeniu i dla różnych symulowanych scenariuszy testowych przy zachowaniu doprowadzania sygnałów sterujących i sterowanych dla PLC w formie analogicznej jak dla obiektów rzeczywistych.

Rys. 4. Struktura sprzętowo-programowa weryfikacji regulatora MPC
Fig. 4. Hardware-software verification structure of MPC regulator

Dla celów realizacji interfejsu użytkownika i dokumentacyjnych zastosowano system SCADA Wonderware InTouch.

11. Obiekty sterowania

Weryfikację regulatora przeprowadzono dla szerokiej gamy obiektów testowych i parametrów strojenia regulatora. Po osiągnięciu zadowalających wyników regulacji dla obiektów inercyjnych, minimalnofazowych zaimplementowany regulator MPC poddano próbom regulacji zasadniczo trudniejszym do sterowania obiektom: nieminimalnofazowym, oscylacyjnym oraz z dużym opóźnieniem.

Obiekt 1. Obiekt nieminimalnofazowy, III rzędu oscylacyjny z opóźnieniem 500 ms (5 okresów próbkowania).

$$g_1(z) = z^{-5}\,\frac{-0{,}07464z^2 + 0{,}07589z + 0{,}07816}{z^3 - 2{,}075z^2 + 1{,}561z - 0{,}4066},\ T = 100\,\mathrm{ms}$$

Obiekt 2. Obiekt inercyjny I rzędu z dużym opóźnieniem

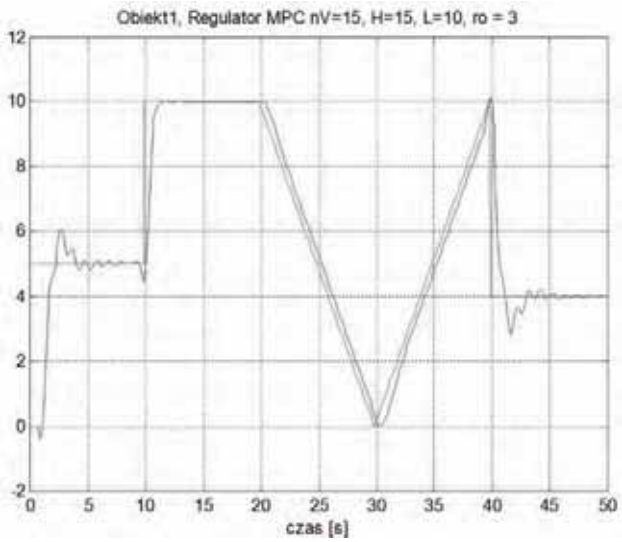
$$g_1(z) = z^{-20}\,\frac{0{,}6321}{z - 0{,}3679},\ T = 1\,\mathrm{s}$$

Realizacja regulacji obiektami 1 i 2 za pomocą regulatorów PID skutkowałaby dużymi przeregulowaniami i długimi czasami regulacji.

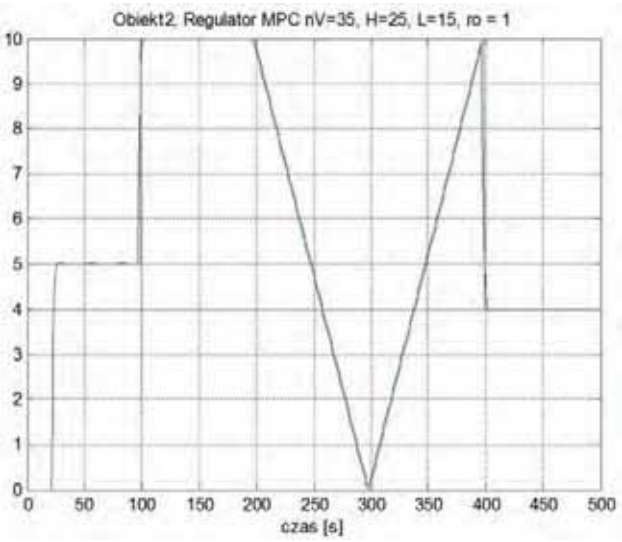
12. Wyniki regulacji

Wyniki sterowania obiektami 1 i 2 przedstawione są na rys. 5 i 6. Zastosowano skokowe i odcinkowo narastające i opadające trajektorie wartości zadanej.

MOST WIEDZY Pobrano z mostwiedzy.pl



Rys. 5. Wyniki regulacji MPC dla obiektu 1
Fig. 5. MPC regulation results for plant 1



Rys. 6. Wyniki regulacji MPC dla obiektu 2
Fig. 6. MPC regulation results for plant 2

Wyniki regulacji można uznać za satysfakcjonujące. Poprawność regulacji trudnymi z punktu widzenia sterowania obiektami wskazuje na prawidłową implementację i użyteczność regulatora predykcyjnego na platformie PLC.

13. Badania wydajnościowe i badania wykorzystania pamięci

Po implementacji i weryfikacji regulatora MPC na platformie PLC wykonano badania pod kątem niezbędnej pamięci do realizacji regulatora i czasu trwania jednej iteracji. Wyniki tych badań zaprezentowano w tab. 2.

Implementację zrealizowano w sterowniku PAC RX3i firmy GE Fanuc z jednostką centralną CPU310 (zegar CPU taktowany 300 MHz). Jednostka CPU310 umożliwia konfigurowanie dostępnej pamięci oraz przydzielanie pamięci rejestrowej reprezentowanej przez zmienne logiczne % R

Tab. 2. Zestawienie niezbędnych zasobów PLC do wykonania algorytmu regulatora MPC (CPU310)
Tab. 2. Summary of the necessary PLC resources to execute the algorithm MPC controller (CPU310)

parametry regulatora			zasoby PLC niezbędne do wykonania regulatora	
nV	H	L	pamięć [słowa 16-bit]	max czas wykonania kroku [ms]
10	5	5	450	9
20	10	10	1600	45
50	15	15	3570	137
50	20	20	6060	300
50	30	25	9790	679
50	40	25	10 870	814
50	40	30	14 420	1832
50	40	32	15 980	2400

w zakresie 0–32 640. Zmienne te są 16-bitowe i do zapisania liczby typu rzeczywistego niezbędne są dwie zmienne % R. Zatem do dyspozycji programisty jest 16 320 liczb rzeczywistych, które mogą być zorganizowane jako skalary, wektory lub dwuwymiarowe macierze.

Parametrami wpływającymi na stopień wykorzystania pamięci są nV, L, H, przy czym nV jest wektorem, a L i H określają wymiary macierzy i to głównie ich wartości wpływają na stopień wykorzystania pamięci. Ponieważ rozmiary macierzy muszą być w programie w języku ST zadane *a priori*, uwzględniając wszystkie operacje i dostępny limit 16 000 liczb, określono maksymalne wartości parametrów nV_max = 50, H_max = 50 i L_max = 50. Zastosowanie maksymalnych wartości tych zmiennych oznacza wykorzystanie 30 100 rejestrów % R. W toku prowadzonych badań okazało się, że wykorzystanie maksymalnych wartości tych parametrów nie będzie możliwe ze względu na czas wykonania jednego cyklu programu przy takich rozmiarach macierzy.

Sterownik PAC ma wbudowany układ kontrolujący czas trwania jednego cyklu (tzw. *watchdog*), przerywający działanie sterownika w przypadku przekroczenia tego czasu, np. powodowanego błędem logicznym w programie lub uszkodzeniem procesora czy pamięci PLC. Maksymalny czas trwania cyklu dla PAC RX3i wynosi 2550 ms. Oznacza to, że wszystkie operacje składające się na jeden cykl programu (inicjalizacja, wczytanie wejść, wykonanie części logicznej programu, ustawienie wyjść, komunikacja systemowa oraz czynności diagnostyczne) zostaną wykonane w czasie krótszym niż 2550 ms. W dokumentacji PLC można odnaleźć informacje o czasie trwania każdej instrukcji, np. mnożenia czy dodawania liczb rzeczywistych. Są to czasy rzędu mi-

MOST WIEDZY Pobrano z mostwiedzy.pl

krosekund, jednak wraz ze wzrostem wartości parametrów L i H oraz nV zasadniczo rośnie liczba wykonanych operacji (głównie mnożenia i dodawania liczb rzeczywistych) i czas trwania jednego cyklu. Ograniczeniem w wyborze wartości parametrów nie była więc pamięć, tylko „moc obliczeniowa" i związany z nią czas trwania obliczeń. Przy parametrach $nV = 50$, $H = 40$, $L = 33$ sterownik przechodził w tryb awaryjny z powodu przekroczenia maksymalnego dopuszczalnego czasu trwania cyklu.

Wyniki badań mogą być przydatne do oceny stosowalności regulatora w odniesieniu do analizowanego obiektu. Są one również pomocne w określeniu możliwości minimalnego kroku czasowego regulatora, z danych (tab. 2) wynika, że nie można użyć regulatora o parametrach nV = 50, H = 40, L = 30 z krokiem 100 ms, ponieważ obliczenia zajmują 814 ms. Dane te mogą być wykorzystane do oszacowania implementowalności regulatora MPC w innych PLC, tj. wielkości wymaganej pamięci oraz porównania mocy obliczeniowych.

Opracowany kod zawiera opisane procedury operacji na danych (mnożenia, dodawania, odwracania macierzy). Algorytm odwracania macierzy może być zrealizowany bardziej efektywnie, zatem istnieje możliwość istotnego poprawienia efektywności wykonywania programu zarówno w aspekcie niezbędnej pamięci jak i czasu wykonania.

14. Podsumowanie

W artykule przedstawiona jest implementacja zaawansowanej metody regulacji predykcyjnej MPC w PLC, czyli w docelowym przemysłowym urządzeniu sterującym.

Weryfikacja poprawności opracowanego w ten sposób algorytmu regulatora predykcyjnego została przeprowadzona w warunkach bliskich rzeczywistości – z wykorzystaniem pętli sprzętowej. Przeprowadzone eksperymenty regulacji trudnymi obiektami potwierdziły poprawność implementacji oraz użyteczność regulatora MPC.

Podano czas wykonywania obliczeń dla jednego kroku algorytmu wykonanego w sterowniku PAC dla różnych zestawów parametrów, co jest przydatne w szacowaniu stosowalności regulatora dla różnych obiektów i sterowników. Wskazano potencjalne etapy w implementacji regulatora, które można poddać strojeniu i optymalizacji kodu w celu redukcji konsumpcji pamięci i/lub uzyskania krótszego czasu wykonania kroku algorytmu.

Uzupełnienie przedstawionego regulatora o aktualizowanie on-line informacji o obiekcie pozwoliłoby na uzyskanie regulatora sterującego obiektami niestacjonarnymi o zmiennej strukturze i czasie opóźnienia. Regulator taki nie jest dostępny standardowo na żadnej platformie PLC.

Bibliografia

1. Niederliński A., Mościński J., Ogonowski Z., *Regulacja adaptacyjna*, WN PWN, Warszawa 1995.
2. Tatjewski P., *Sterowanie zaawansowane obiektów przemysłowych*, Akademicka Oficyna Wydawnicza EXIT, Warszawa 2002.
3. Maciejowski J.M., *Predictive Control with Constaints*, Prentice Hall, 2002.
4. Legierski T., Kasprzyk J., Wyrwał J., Hajda J., *Programowanie sterowników PLC*, Wydawnictwo Pracowni Komputerowej J. Skalmierskiego, Gliwice 1998.
5. Kasprzyk J., *Programowanie sterowników przemysłowych*, WNT, Warszawa 2007.
6. Korbicz J., Kościelny J.M., *Modelowanie, diagnostyka i sterowanie nadrzędne procesami. Implementacja w systemie Diaster*, WNT, Warszawa 2009.
7. Holkar K.S., Waghmare L.M., *An Overview of Model Predictive Control*, "International Journal of Control and Automation", vol. 3, no. 4, 2010, 47–63.
8. Tarnawski J., *Realizacja programowa algorytmów filtracji, estymacji i sterowania w PLC/PAC,* „Pomiary Automatyka Robotyka", 5/2013, 100–107.

Implementation of Predictive Control Algorithm in Programmable Logic Controllers

Abstract: PLCs (Programmable Logic Controllers) are the main industrial platform for the implementation of control algorithms. PLC manufacturers provide for programmers only basic control algorithms. With the development of PLC and their successors (Programmable Automation Controller – PAC) appeared increased CPU and memory capabilities of the equipment and fuller implementation of programming languages defined in the standard IEC-61131-3. PLCs and PACs now have the computing power and memory of the personal computer PC a few years ago, they can also be programmed in high level languages using matrix operations. Currently, PLC and PAC allow implementation of almost any discrete control algorithm. In this article it is considered the implementation of advanced control method – the algorithm MAC/MPC (Model Algorithmic Control/Model Predictive Control) due to its usability and for the presentation of the implementation process. MPC can be implemented also in the supervisory layer of hierarchical model of the control system, so the whole model is presented. Verification process of implemented MPC controller in the PLC with hardware-in-a-loop structure is presented. Results of cycle durations and memory requirements tests can be used to estimate the applicability of MPC controller on other platforms and for particular objects. Properties of MPC controllers are not available for the classic PLC controllers (PID class with modifications and expansions), therefore, the implementation of MPC in the PLC significantly expands the control capabilities of this platform.

Keywords: hierarchical control system, advanced control algorithms, model predictive control, programmable logic controllers, norm IEC-61131-3, PLC programming limits, process of MPC implementation, verification in hardware-in-the-loop

dr inż. Jarosław Tarnawski

Adiunkt w Katedrze Inżynierii Systemów Sterowania, Wydział Elektrotechniki i Automatyki Politechniki Gdańskiej. Obszary zainteresowań badawczych: synteza komputerowych systemów sterowania, przemysłowe sieci informatyczne, sterowanie obiektami z opóźnieniami, systemy środowiskowe.

e-mail: j.tarnawski@eia.pg.gda.pl

MOST WIEDZY Pobrano z mostwiedzy.pl