

dr inż. Maciej Ławryńczuk
Instytut Automatyki i Informatyki Stosowanej
Politechnika Warszawska

SUBOPTYMALNY ALGORYTM REGULACJI PREDYKCYJNEJ Z MODELAMI NEURONOWYMI TYPU FIR

W pracy omówiono nieliniowy algorytm regulacji predykcyjnej wykorzystujący modele neuronowe typu FIR (ang. Finite Impulse Response). W algorytmie zastosowano modele bazujące na sieci perceptronowej. Przedstawiony algorytm jest efektywny obliczeniowo, ponieważ wymaga cyklicznego rozwiązywania zadania optymalizacji kwadratowej, co może być wykonane w czasie rzeczywistym. Algorytm cechuje się dużą dokładnością regulacji, porównywalną z algorytmami wykorzystującymi bieżącą nieliniową optymalizację.

SUBOPTIMAL PREDICTIVE CONTROL BASED ON NEURAL FIR MODELS

This paper describes a suboptimal nonlinear Model Predictive Control (MPC) algorithm based on FIR (Finite Impulse Response) neural models. Multilayer Perceptron (MLP) neural network is used. The algorithm is computationally efficient because it results in a quadratic programming problem, which can be easily solved on-line by means of a numerically reliable software subroutine. The algorithm gives good closed-loop control performance, comparable to that obtained in the nonlinear MPC technique, which hinges on nonlinear optimisation.

1. WSTĘP

Algorytmy regulacji predykcyjnej (ang. *Model Predictive Control*, w skrócie MPC) są jedyną zaawansowaną techniką regulacji, która nie tylko przyciągnęła uwagę teoretyków, ale również znalazła bardzo szerokie zastosowanie praktyczne [12, 20, 21, 22]. Algorytmy tej klasy są obecnie stosowane w regulacji wielu procesów technologicznych, przede wszystkim w przemyśle chemicznym, petrochemicznym, papierniczym, metalurgicznym oraz przetwórczym. Algorytmy regulacji predykcyjnej mają kilka istotnych zalet, wyróżniających je wśród innych technik regulacji. Po pierwsze, umożliwiają w naturalny sposób uwzględnienie ograniczeń sygnałów wejściowych i wyjściowych procesu, których spełnienie jest kluczowe z punktu widzenia jakości, efektywności ekonomicznej i bezpieczeństwa produkcji. Po drugie, algorytmy te można z powodzeniem zastosować do regulacji procesów wielowymiarowych, o wielu wejściach i wielu wyjściach. Algorytmy regulacji predykcyjnej można wykorzystać do regulacji procesów o trudnej dynamice, np. procesów z tzw. odwrotną odpowiedzią skokową. Ponieważ właściwości wielu procesów są nieliniowe, z coraz większym zainteresowaniem spotykają się nieliniowe algorytmy regulacji predykcyjnej [4, 15, 20, 22]. W ogólności, można wyróżnić dwie klasy algorytmów nieliniowych:

- a) algorytmy z nieliniową optymalizacją
- b) algorytmy suboptymalne z cykliczną linearyzacją nieliniowego modelu procesu.

W pierwszym przypadku do wyznaczenia aktualnej polityki sterowania wykorzystuje się nieliniowy model procesu bez żadnych uproszczeń. Wadą takiego podejścia jest konieczność rozwiązywania w czasie rzeczywistym nieliniowego zadania optymalizacji. Jest to trudne,

złożone obliczeniowe, czasochłonne. Przede wszystkim jednak, istnieje niebezpieczeństwo utknięcia procedury optymalizacji w płytkim minimum lokalnym.

W drugim przypadku, w aktualnym punkcie pracy wyznacza się liniowe przybliżenie modelu procesu, a następnie wyznacza aktualną politykę sterowania rozwiązując zadanie optymalizacji kwadratowej. Jego rozwiązanie może być uzyskane w skończonym czasie, możliwym od przewidzenia. Algorytmy suboptymalne zapewniają w praktyce dobrą jakość regulacji, niewiele gorszą od algorytmów z nieliniową optymalizacją [4, 22].

W pracy opisano suboptymalny algorytm regulacji predykcyjnej z modelami neuronowymi typu FIR (ang. Finite Impulse Response). Algorytm ten należy do klasy algorytmów z Nieliniową Predykcją i Linearyzacją (MPC-NPL) [8, 9, 11, 22, 23]. Wykorzystane w algorytmie dyskretne modele typu FIR opisują wyjście procesu wyłącznie w zależności od jego wejścia w poprzednich iteracjach. Ponieważ w modelu neuronowym brak jest zależności wyjścia od sygnału wyjściowego w poprzednich iteracjach, modele tego typu mogą być łatwo uczone przy wykorzystaniu klasycznego algorytmu propagacji wstecznej [3, 17], nie zachodzi potrzeba stosowania bardziej złożonych algorytmów rekurencyjnych [10, 17].

W ogólności, algorytm regulacji może wykorzystywać modele fizykochemiczne. Choć potencjalnie rozwiązanie takie wydaje się dobre, ma ono kilka wad. Opracowanie modeli wymaga dogłębnej wiedzy, jest zwykle długotrwałe i kosztowne, poważną trudnością może okazać się konieczność empirycznego wyznaczenia wartości parametrów, np. szybkości reakcji chemicznych. Modele fizykochemiczne są zwykle bardzo złożone (nieliniowe układy równań różniczkowych i algebraicznych), ich bezpośrednie zastosowanie w algorytmach regulacji nie jest zazwyczaj możliwe, mogą prowadzić do problemów numerycznych.

W przeciwieństwie do modeli fizykochemicznych, modele neuronowe mają kilka zalet, a mianowicie:

- a) Sieci neuronowe są uniwersalnymi aproksymatorami. Są one w stanie aproksymować dowolną funkcję nieliniową zadaną dokładnością [5], a więc mogą one również dokładnie modelować zachowanie procesów technologicznych [2, 6, 13, 16].
- b) Opracowano wiele algorytmów uczenia sieci i doboru ich architektury [3, 17].
- c) Sieci neuronowe, w przeciwieństwie do modeli wykorzystujących zasady logiki rozmytej, mają stosunkowo mało parametrów, nie występuje tzw. zjawisko „przekleństwa wymiarowości”, polegające na gwałtownym zwiększaniu ilości parametrów przy wzroście ilości argumentów.
- d) Sieci neuronowe mogą być stosunkowo łatwo wykorzystane w różnych odmianach algorytmów regulacji predykcyjnej [1, 6, 7, 8, 9, 11, 16, 18, 19, 23, 24].
- e) Algorytmy regulacji predykcyjnej z modelami neuronowymi nie prowadzą do postania problemów numerycznych.

2. REGULACJA PREDYKCYJNA

W algorytmie regulacji predykcyjnej, w każdej dyskretnej chwili k wyznacza się wektor przyszłych przyrostów sygnału sterującego

$$\Delta \mathbf{u}(k) = [\Delta u(k|k) \quad \dots \quad \Delta u(k + N_u - 1|k)]^T \quad (1)$$

Zakłada się, że $\Delta u(k + p|k) = 0$ dla $p \geq N_u$, przy czym N_u jest horyzontem sterowania. Celem algorytmu jest minimalizacja różnic między trajektorią zadaną $y^{zad}(k + p|k)$ a prognozowanymi wartościami sygnału wyjściowego $\hat{y}(k + p|k)$ na całym horyzoncie predykcji, dla

$p = 1, \dots, N$. Przewidywanie (predykcja) zmiennej wyjściowej procesu jest wykonywane na bieżąco w oparciu o istniejący model dynamiczny. Minimalizowana funkcja kosztu ma postać

$$J(k) = \sum_{p=1}^N \mu_p (y^{zad}(k+p|k) - \hat{y}(k+p|k))^2 + \sum_{p=0}^{N_u-1} \lambda_p (\Delta u(k+p|k))^2 \quad (2)$$

gdzie $\mu_p \geq 0$ i $\lambda_p > 0$ są współczynnikami wagowymi. W praktyce $N_u < N$, dzięki czemu ogranicza się wymiarowość problemu optymalizacji. Do sterowania wykorzystuje się jedynie pierwszy element wyznaczonej sekwencji (1), prawo regulacji ma postać

$$u(k|k) = \Delta u(k|k) + u(k-1) \quad (3)$$

W kolejnej dyskretniej chwili $k+1$ następuje aktualizacja pomiaru zmiennej wyjściowej, horyzont zostaje przesunięty o jeden krok do przodu i cała procedura jest powtórzona.

2.1. Zadanie optymalizacji algorytmu regulacji predykcijnej

Jak podkreślono we wstępie, możliwość uwzględnienia ograniczeń sygnałów wejściowych i wyjściowych procesu jest jedną z zalet algorytmów regulacji predykcijnej. W każdej iteracji algorytm rozwiązywane jest następujące zadanie optymalizacji

$$\min_{\Delta u(k)} \{J(k) = \sum_{p=1}^N \mu_p (y^{zad}(k+p|k) - \hat{y}(k+p|k))^2 + \sum_{p=0}^{N_u-1} \lambda_p (\Delta u(k+p|k))^2\} \quad (4)$$

przy ograniczeniach :

$$\begin{aligned} u_{\min} &\leq u(k+p|k) \leq u_{\max} & p &= 0, \dots, N_u - 1 \\ -\Delta u_{\max} &\leq \Delta u(k+p|k) \leq \Delta u_{\max} & p &= 0, \dots, N_u - 1 \\ y_{\min} &\leq \hat{y}(k+p|k) \leq y_{\max} & p &= 1, \dots, N \end{aligned}$$

Definiując wektory o długości N

$$\begin{aligned} \mathbf{y}^{zad}(k) &= [y^{zad}(k+1|k) \quad \dots \quad y^{zad}(k+N|k)]^T, & \mathbf{y}_{\min} &= [y_{\min} \quad \dots \quad y_{\min}]^T \\ \hat{\mathbf{y}}(k) &= [\hat{y}(k+1|k) \quad \dots \quad \hat{y}(k+N|k)]^T, & \mathbf{y}_{\max} &= [y_{\max} \quad \dots \quad y_{\max}]^T \end{aligned} \quad (5)$$

oraz wektory o długości N_u

$$\begin{aligned} \mathbf{u}_{\min} &= [u_{\min} \quad \dots \quad u_{\min}]^T, & \mathbf{u}^{k-1} &= [u(k-1) \quad \dots \quad u(k-1)]^T \\ \mathbf{u}_{\max} &= [u_{\max} \quad \dots \quad u_{\max}]^T, & \Delta \mathbf{u}_{\max} &= [\Delta u_{\max} \quad \dots \quad \Delta u_{\max}]^T \end{aligned} \quad (6)$$

problem optymalizacji (4) można przedstawić jako

$$\min_{\Delta u(k)} \{J(k) = \|\mathbf{y}^{zad}(k) - \hat{\mathbf{y}}(k)\|_{\mathbf{M}}^2 + \|\Delta \mathbf{u}(k)\|_{\mathbf{A}}^2\} \quad (7)$$

przy ograniczeniach :

$$\begin{aligned} \mathbf{u}_{\min} &\leq \mathbf{J} \Delta \mathbf{u}(k) + \mathbf{u}^{k-1} \leq \mathbf{u}_{\max} \\ -\Delta \mathbf{u}_{\max} &\leq \Delta \mathbf{u}(k) \leq \Delta \mathbf{u}_{\max} \\ \mathbf{y}_{\min} &\leq \hat{\mathbf{y}}(k) \leq \mathbf{y}_{\max} \end{aligned}$$

przy czym \mathbf{J} jest trójkątną dolną macierzą o wymiarowości $N_u \times N_u$, natomiast \mathbf{M} oraz \mathbf{A} są diagonalnymi macierzami o wymiarowości, odpowiednio, $N \times N$ oraz $N_u \times N_u$ zawierającymi współczynniki wagowe μ_p, λ_p .

3. SUBOPTYMALNY ALGORYTM REGULACJI PREDYKCYJNEJ Z MODELAMI NEURONOWYMI TYPU FIR

3.1. Struktura modelu neuronowego FIR

Zakłada się, że model dynamiczny procesu nieliniowego o jednym wejściu i jednym wyjściu (ang. Single-Input Single-Output, w skrócie SISO) dany jest w postaci równania różnicowego

$$y(k) = f(\mathbf{x}(k)) = f(u(k - \tau), \dots, u(k - n_B)) \quad (8)$$

Cechą szczególną zastosowanego modelu FIR (8) jest fakt, że wyjście procesu w aktualnej chwili k jest pewną nieliniową funkcją wyłącznie wejścia w poprzednich iteracjach, nie zależy ono natomiast od sygnału wyjściowego w poprzednich iteracjach jak w modelach typu ARX (ang. Auto Regressive with eXogenous input)

$$y(k) = f(\mathbf{x}(k)) = f(u(k - \tau), \dots, u(k - n_B), y(k - 1), \dots, y(k - n_A)) \quad (9)$$

Nieliniowa funkcja $f: \mathfrak{R}^{n_B - \tau + 1} \rightarrow \mathfrak{R} \in C^1$, gdzie $\tau \leq n_B$, jest realizowana przez sieć neuronową typu perceptronowego (ang. Multilayer Perceptron, w skrócie MLP) z jedną warstwą ukrytą i liniowym węzłem wyjściowym [3, 17]. Wyjście sieci obliczane jest ze wzoru

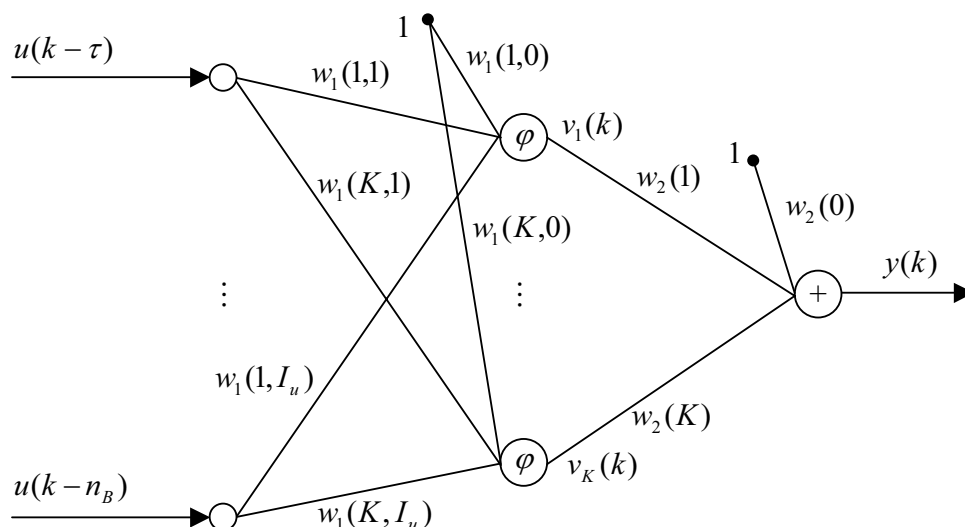
$$y(k) = w_0^2 + \sum_{i=1}^K w_i^2 v_i(k) = w_0^2 + \sum_{i=1}^K w_i^2 \varphi(z_i(k)) \quad (10)$$

gdzie wielkości $z_i(k)$ jest sumą sygnałów wejściowych i -tego neuronu ukrytego, natomiast $v_i(k)$ oznacza jego sygnał wyjściowy, $\varphi: \mathfrak{R} \rightarrow \mathfrak{R}$ jest nieliniową funkcją aktywacji neuronów warstwy ukrytej (np. $\varphi = \tanh$), K jest ilością neuronów ukrytych. Na podstawie argumentów ogólnego modelu nieliniowego FIR (8), dla sieci neuronowej otrzymuje się

$$z_i(k) = w_1(1,0) + \sum_{j=1}^{I_u} w_1(i,j)u(k - \tau + 1 - j) \quad (11)$$

Wagi pierwszej warstwy oznaczone są jako $w_1(i,j)$, przy czym $i = 1, \dots, K$, $j = 0, \dots, I_u$, natomiast wagi drugiej warstwy przez $w_2(i)$, gdzie $i = 0, \dots, K$. Całkowita ilość węzłów wejściowych sieci wynosi $I_u = n_B - \tau + 1$. Struktura sieci neuronowej pokazana jest na rys. 1.

Modele neuronowe FIR są uczone klasycznym algorytmem propagacji wstecznej, podczas gdy w przypadku modeli ARX najlepiej zastosować bardziej złożone obliczeniowo algorytmy rekurencyjne. Można wówczas mówić o uczeniu modelu OE (ang. Output Error).



Rys. 1. Struktura modelu neuronowego typu FIR

3.2. Zadanie optymalizacji suboptymalnego algorytmu regulacji predykcyjnej

Aby wyeliminować konieczność rozwiązywania w każdej iteracji algorytmu nieliniowego problemu optymalizacji (7), zostanie wykorzystane podejście suboptymalne, bazujące na algorytmie z Nieliniową Predykcją i Linearyzacją (NPL) [8, 9, 11, 22, 23]. W każdej iteracji k algorytmu NPL nieliniowy neuronowy model procesu jest wykorzystany dwa razy, a mianowicie do wyznaczenia lokalnego przybliżenia liniowego oraz do obliczenia nieliniowej trajektorii swobodnej. Zakłada się, że wektor predykcji sygnału wyjściowego $\hat{y}(k)$ na horyzoncie predykcji można przedstawić jako sumę trajektorii wymuszonej, która zależy wyłącznie od przyszłości (czyli od przyszłych przyrostów sygnału sterującego $\Delta u(k)$ na horyzoncie sterowania) oraz trajektorii swobodnej $y^0(k)$, która zależy wyłącznie od przeszłości

$$\hat{y}(k) = \mathbf{G}(k)\Delta u(k) + y^0(k) \quad (12)$$

Macierz dynamiczna o wymiarowości $N \times N_u$ zawiera współczynniki odpowiedzi skokowej aktualnie wyznaczonej liniowej aproksymacji modelu nieliniowego. Ma ona postać

$$\mathbf{G}(k) = \begin{bmatrix} s_1(k) & 0 & \dots & 0 \\ s_2(k) & s_1(k) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ s_N(k) & s_{N-1}(k) & \dots & s_{N-N_u+1}(k) \end{bmatrix} \quad (13)$$

natomiast wektor trajektorii swobodnej ma długość N

$$y^0(k) = [y^0(k+1|k) \quad \dots \quad y^0(k+N|k)]^T \quad (14)$$

Symbolem $s_j(k)$ oznaczono współczynniki odpowiedzi skokowej modelu zlinearyzowanego

$$s_j(k) = \sum_{i=1}^{\min(j, N_B)} b_i(k) \quad (15)$$

gdzie $b_i(k)$ są współczynnikami modelu zlinearyzowanego.

Dzięki zastosowaniu suboptymalnej predykcji (12), nieliniowy problem optymalizacji rozwiązywany w każdej iteracji algorytmu regulacji predykcyjnej (7) jest w istocie następującym zadaniem programowania kwadratowego

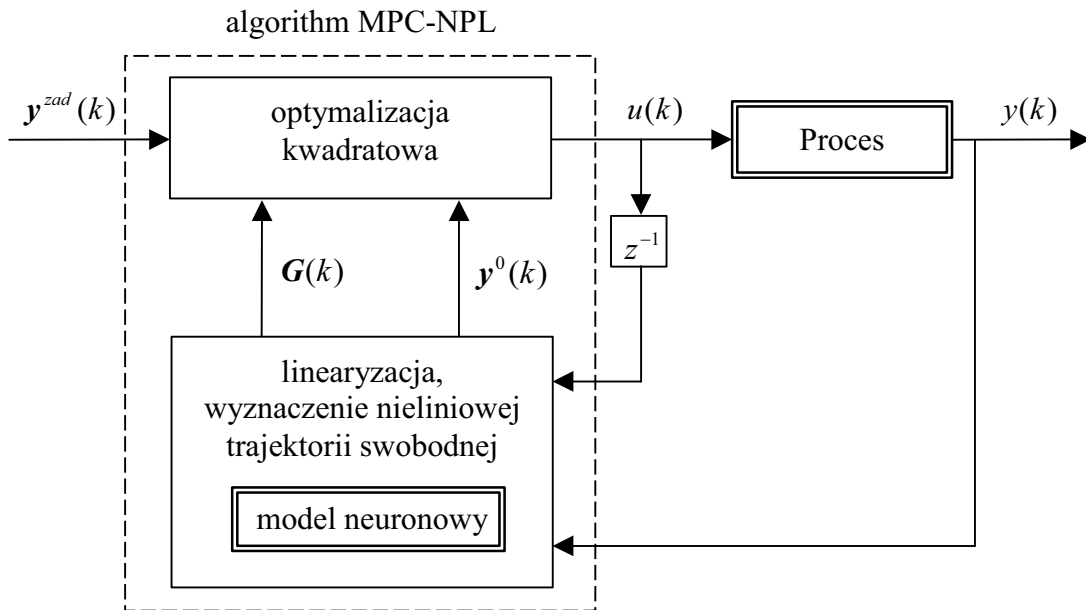
$$\min_{\Delta u(k)} \{J(k) = \|\mathbf{y}^{zad}(k) - \mathbf{G}\Delta u(k) - y^0(k)\|_M^2 + \|\Delta u(k)\|_A^2\} \quad (16)$$

przy ograniczeniach :

$$\begin{aligned} \mathbf{u}_{\min} &\leq \mathbf{J}\Delta u(k) + \mathbf{u}^{k-1} \leq \mathbf{u}_{\max} \\ -\Delta \mathbf{u}_{\max} &\leq \Delta u(k) \leq \Delta \mathbf{u}_{\max} \\ \mathbf{y}_{\min} &\leq \mathbf{G}(k)\Delta u(k) + y^0(k) \leq \mathbf{y}_{\max} \end{aligned}$$

Ogólna struktura algorytmu suboptymalnego z Nieliniową Predykcją i Linearyzacją (NPL) została przedstawiona na rys. 2. W każdej iteracji wykonane są następujące kroki:

1. Linearyzacja modelu neuronowego: wyznaczenie macierzy dynamicznej $\mathbf{G}(k)$.
2. Obliczenie nieliniowej trajektorii swobodnej $y^0(k)$ na podstawie modelu neuronowego.
3. Rozwiązanie zadania programowania kwadratowego (16) w celu wyznaczenia $\Delta u(k)$.
4. Zastosowanie do sterowania pierwszego elementu obliczonego wektora $\Delta u(k)$.
5. Podstawienie $k := k + 1$, przejście do kroku 1.



Rys. 2. Struktura algorytmu suboptymalnego z Nieliniową Predykcją i Linearyzacją (NPL)

3.3. Linearyzacja modelu neuronowego i obliczanie nieliniowej trajektorii swobodnej

Aktualna linearyzacja nieliniowego modelu neuronowego wyznaczona jest w chwili k na podstawie punktu pracy odpowiadającemu argumentom modelu (8)

$$\bar{\mathbf{x}}(k) = [u(k - \tau) \quad \dots \quad u(k - n_B)] \quad (17)$$

Korzystając z rozwinięcia funkcji f w szereg Taylora, model zlinearyzowany ma postać

$$y(k) = f(\bar{\mathbf{x}}(k)) + \sum_{l=1}^{n_B} b_l(\bar{\mathbf{x}}(k))(u(k-l) - \bar{u}(k-l)) \quad (18)$$

Uwzględniając strukturę modelu neuronowego opisanego równaniami (10) i (11), współczynniki modelu zlinearyzowanego są wyznaczone ze wzoru

$$b_l(k) = \begin{cases} 0 & l = 1, \dots, \tau - 1 \\ \sum_{i=1}^K w_2(i) \frac{d\varphi(z_i(\bar{\mathbf{x}}(k)))}{dz_i(\bar{\mathbf{x}}(k))} w_1(i, l - \tau + 1) & l = \tau, \dots, n_B \end{cases} \quad (19)$$

Jeżeli neurony warstwy ukrytej mają funkcję aktywacji φ typu tanges hiperpoliczny

$$\frac{d\varphi(z_i(\bar{\mathbf{x}}(k)))}{dz_i(\bar{\mathbf{x}}(k))} = 1 - \tanh^2(z_i(\bar{\mathbf{x}}(k))) \quad (20)$$

Trajektoria swobodna obliczana jest tylko przy uwzględnieniu przeszłości. Ze wzoru (10)

$$y^0(k+p|k) = w_2(0) + \sum_{i=1}^K w_2(i) \varphi(z_i^0(k+p|k)) + d(k) \quad (21)$$

Korzystając z (11), przy założeniu braku zmian sygnału sterującego od chwili k

$$z_i^0(k+p|k) = w_1(1,0) + \sum_{j=1}^{I_{wf}(p)} w_1(i,j) u(k-1) + \sum_{j=I_{wf}(p)+1}^{I_u} w_1(i,j) u(k-\tau+1-j+p) \quad (22)$$

gdzie $I_{uf}(p) = \max(\min(p - \tau + 1, I_u), 0)$. W przeciwieństwie do modelu ARX (9), trajektoria swobodna nie zależy rekurencyjnie od predykcji obliczonych dla poprzednich chwil ($\hat{y}(k+1|k), \hat{y}(k+2|k), \dots$) oraz zmierzonych uprzednio wartości sygnału wyjściowego ($y(k-1), y(k-2), \dots$) [8, 9, 11, 22, 23].

W opisywanym algorytmie zastosowano model zakłóceń typu DMC [12]. Zakłada się mianowicie, że niemierzalne zakłócenie $d(k)$ oddziałujące na proces w chwili k nie zmienia swojej wartości do końca horyzontu predykcji. Jego wartość obliczana jest ze wzoru

$$d(k) = y(k) - y(k|k-1) \quad (23)$$

przy czym $y(k)$ jest zmierzoną wartością sygnału wyjściowego, natomiast wielkość $y(k|k-1)$ jest obliczana na podstawie modelu. Ze wzoru (10)

$$d(k) = y(k) - \left(w_2(0) + \sum_{i=1}^K w_2(i) \varphi(z_i(k)) \right) \quad (24)$$

4. WYNIKI SYMULACJI

4.1. Reaktor polimeryzacji

Rozważanym procesem jest reaktor polimeryzacji [14]. Jego wyjściem jest średnia masa cząsteczkowa $NAMW$ (ang. *Number Average Molecular Weight*), natomiast wejściem jest natężenie dopływu inicjatora F_I . Omawiany reaktor jest często stosowany w celu porównania nieliniowych algorytmów regulacji [8, 9, 22, 23]. Algorytmy regulacji predykcyjnej wykorzystujące model liniowe są nieefektywne przy szerokich zmianach punktu pracy.

4.2. Modelowanie procesu polimeryzacji

Model fizykochemiczny [14] jest traktowany podczas symulacji jako rzeczywisty proces. Zostały wygenerowane dwa zbiory danych liczące 2000 próbek, a mianowicie zbiór danych uczących oraz danych testowych. Pierwszy z nich służy do identyfikacji (uczenia) modeli neuronowych, drugi – wyłącznie do oceny jakości otrzymanych modeli. Aby oddać warunki panujące w przemyśle do wyjścia procesu dodano niewielki szum. Przygotowano pięć modeli FIR różniących się rzędem n_B , we wszystkich przypadkach przyjęto $\tau=2$. W tabeli 1 porównano dokładność otrzymanych modeli pod względem sumy kwadratów błędów (ang. *Sum of Squared Errors*). Na rys. 3 przedstawiono dane testowe oraz wyjście modelu o małej ilości parametrów ($n_B = 5$), natomiast na rys. 4 przedstawiono dane testowe oraz wyjście modelu o dużej ilości parametrów ($n_B = 25$).

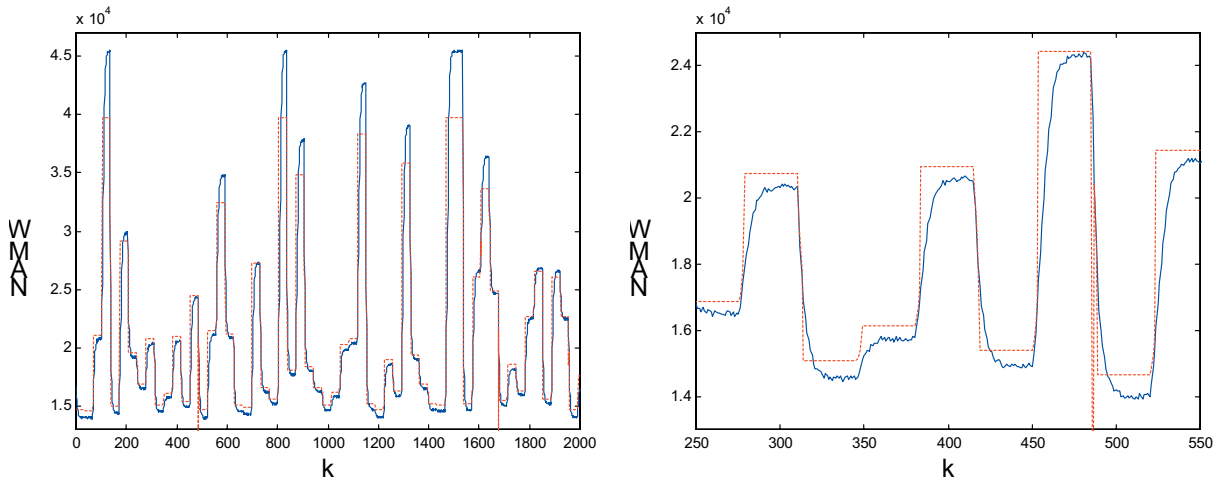
Tabela 1. Wpływ rzędu modelu n_B na dokładność (suma kwadratów błędów modelu)

n_B	dane uczące	dane testowe
5	$1,2930 \cdot 10^{10}$	$1,3557 \cdot 10^{10}$
10	$2,9811 \cdot 10^9$	$3,0655 \cdot 10^9$
15	$3,2952 \cdot 10^8$	$4,0955 \cdot 10^8$
20	$2,6386 \cdot 10^7$	$1,0361 \cdot 10^8$
25	$1,1070 \cdot 10^7$	$8,8185 \cdot 10^7$

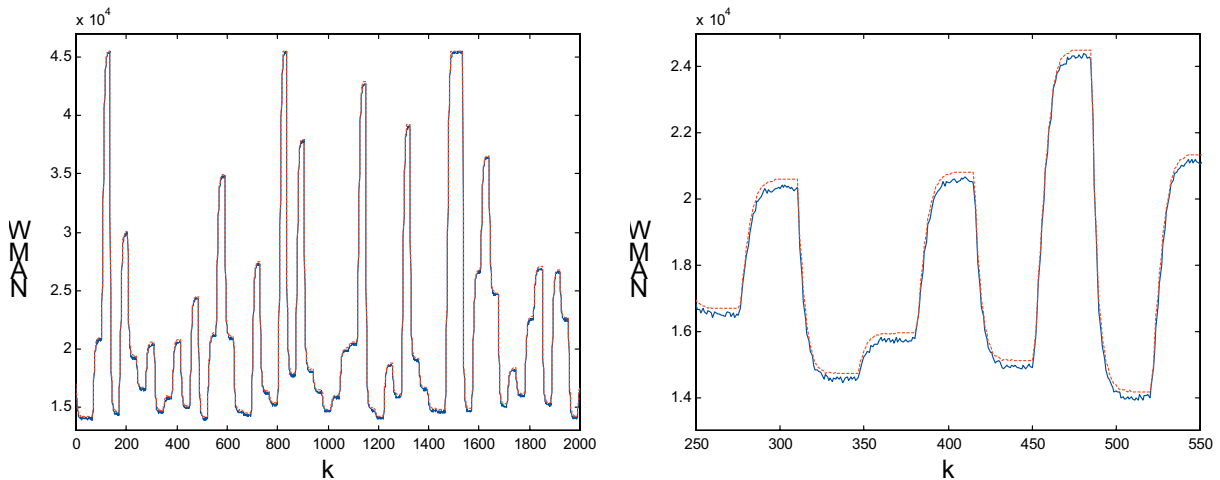
4.3. Regulacja predykcyjna procesu polimeryzacji

Porównano dwie wersje algorytmów regulacji predykcyjnej, a mianowicie algorytm suboptymalny z nieliniową predykcją i linearyzacją oraz algorytm z nieliniową optymalizacją. W

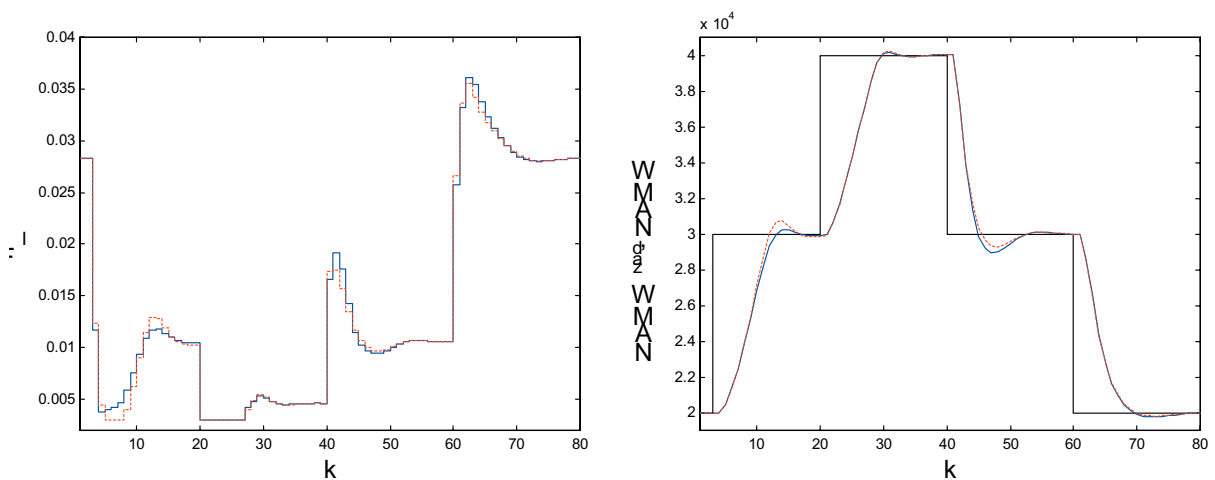
obu algorytmach zastosowano ten sam model FIR ($n_B=25$) oraz te same parametry: $N=10$, $N_u=3$, $\mu_p=1$, $\lambda_p=0,2$, $F_{lmin}=0,003$, $F_{lmax}=0,06$. Czas próbkowania wynosi 1,8 min.



Rys. 3. Dane testowe oraz wyjście modelu o małej ilości parametrów ($n_B=5$), cały zbiór oraz powiększony fragment



Rys. 4. Dane testowe oraz wyjście modelu o dużej ilości parametrów ($n_B=25$), cały zbiór oraz powiększony fragment



Rys. 5. Wyniki symulacji algorytmu suboptymalnego oraz z nieliniową optymalizacją

Wyniki symulacji obu porównywanych algorytmów regulacji przedstawiono na rys. 5. Są one bardzo podobne. Algorytm suboptymalny z cykliczną linearyzacją i optymalizacją kwadratową pozwolił w badanym przypadku na osiągnięcie jakości regulacji niewiele gorszej niż w algorytmie z nieliniową optymalizacją powtarzaną w każdej iteracji. W algorytmie suboptymalnym suma kwadratów uchybów wynosi $1,5868 \cdot 10^9$, w algorytmie z nieliniową optymalizacją $1,5827 \cdot 10^9$.

5. PODSUMOWANIE

Opisany w pracy suboptymalny algorytm regulacji predykcyjnej z modelami neuronowymi typu FIR cechuje się dużą jakością regulacji, efektywnością obliczeniową, natomiast jego praktyczna implementacja jest stosunkowo prosta. Dzięki cyklicznej linearyzacji modelu neuronowego i wyznaczaniu nieliniowej trajektorii swobodnej w każdej iteracji rozwiązuje się kwadratowe zadanie optymalizacji. Umożliwia on jednocześnie osiągnięcie wysokiej jakości regulacji, porównywalnej z możliwą do uzyskania w algorytmie z nieliniową optymalizacją, w którym w każdej iteracji rozwiązuje się nieliniowe zadanie optymalizacji.

W przedstawionym algorytmie wykorzystano dyskretne modele typu FIR. Opisują one wyjście procesu wyłącznie w zależności od jego wejścia w poprzednich iteracjach. Dzięki temu modele tego typu mogą być łatwo uczone przy wykorzystaniu klasycznego algorytmu propagacji wstecznej, nie zachodzi potrzeba stosowania bardziej złożonych obliczeniowo algorytmów rekurencyjnych. Należy jednak pamiętać, że modele FIR wymagają zwykle wielu parametrów, znacznie więcej niż odpowiadające im modele ARX o podobnej dokładności. Fakt ten należy mieć na uwadze podczas identyfikacji modeli i syntezy algorytmu regulacji.

LITERATURA

- [1] B. M. Åkesson, H. T. Toivonen (2006): A neural network model predictive controller. *Journal of Process Control*, tom 16, nr 3, s. 937-946.
- [2] N. Bhat, T. J. McAvoy (1990): Use of neural nets for dynamic modeling and control of chemical proceses. *Computers and Chemical Engineering*, tom 14, nr 4/5, s. 573-583.
- [3] S. Haykin (1999): *Neural networks – a comprehensive foundation*. Prentice Hall, Upper Saddle River.
- [4] M. A. Henson (1998): Nonlinear model predictive control: current status and future directions. *Computers and Chemical Engineering*, tom 23, nr 2, s. 187-202.
- [5] K. Hornik, M. Stinchcombe, H. White (1989): Multilayer feedforward networks are universal approximators. *Neural networks*, tom 2, nr 5, str. 359-366.
- [6] M. A. Hussain (1999): Review of the applications of neural networks in chemical process control – simulation and online implementation. *Artificial Intelligence in Engineering*, tom 13, nr 1, s. 55-68.
- [7] G. P. Liu, V. Kadiramanathan, S. A. Billings (1998): Predictive control of nonlinear systems using neural networks. *International Journal of Control*, tom 71, nr 6, s. 1119-1132.
- [8] M. Ławryńczuk (2007): A family of model predictive control algorithms with artificial neural networks. *International Journal of Applied Mathematics and Computer Science*, tom 17, nr 2, s. 217-232.

- [9] M. Ławryńczuk, P. Tatjewski (2007): A computationally efficient nonlinear predictive control algorithm with RBF neural models and its application. *Lecture Notes in Artificial Intelligence, tom 4585, Springer: The International Conference Rough Sets and Emerging Intelligent Systems Paradigms, RSEISP 2007, Warszawa*, s. 603-612.
- [10] M. Ławryńczuk (2007): Rekurencyjne sieci neuronowe w modelowaniu nieliniowych procesów dynamicznych. *Pomiary Automatyka Robotyka*, nr 2, CD-ROM: Konferencja Automation 2007.
- [11] M. Ławryńczuk, P. Tatjewski (2006): An efficient nonlinear predictive control algorithm with neural models and its application to a high-purity distillation process. *Lecture notes in Artificial Intelligence, tom 4029, Springer: The Eighth International Conference on Artificial Intelligence and Soft Computing ICAISC 2006, Zakopane*, s. 76-85.
- [12] J. M. Maciejowski (2002): *Predictive control with constraints*. Prentice Hall, Harlow.
- [13] J. C. MacMurray, D. M. Himmelblau (1995): Modeling and control of a packed distillation column using artificial neural networks. *Computers and Chemical Engineering*, tom 19, nr 10, s. 1077-1088.
- [14] B. R. Maner, F. J. Doyle, B. A. Ogunnaike, R. K. Pearson (1996): Nonlinear model predictive control of a simulated multivariable polymerization reactor using second-order Volterra models. *Automatica*, tom 32, nr 9, s. 1285-1301.
- [15] M. Morari, J. Lee (1999): Model predictive control: past, present and future. *Computers and Chemical Engineering*, tom 23, nr 4/5, s. 667-682.
- [16] M. Nørgaard, O. Ravn, N. K. Poulsen, L. K. Hansen (2000): *Neural networks for modelling and control of dynamic systems*. Springer, London.
- [17] S. Osowski (1996): *Sieci neuronowe w ujęciu algorytmicznym*. WNT, Warszawa.
- [18] S. Piche, B. Sayyar-Rodsari, D. Johnson, M. Gerules (2000): Nonlinear model predictive control using neural networks. *IEEE Control Systems Magazine*, tom 20, nr 3, s. 56-62.
- [19] M. Pottmann, D. E. Seborg (1997): A nonlinear predictive control strategy based on radial basis function networks. *Computers and Chemical Engineering*, tom 21, nr 9, s. 965-980.
- [20] S. J. Qin, T. Badgwell (2003): A survey of industrial model predictive control technology. *Control Engineering Practice*, tom 11, nr 7, s. 733-764.
- [21] J. A. Rossiter (2003): *Model-based predictive control*. CRC Press, Boca Raton.
- [22] P. Tatjewski (2007): *Advanced control of industrial processes, structures and algorithms*. Springer, London.
- [23] P. Tatjewski, M. Ławryńczuk (2006): Soft computing in model-based predictive control. *International Journal of Applied Mathematics and Computer Science*, tom 16, nr 1, s. 101-120.
- [24] D. L. Yu, J. B. Gomm (2003): Implementation of neural network predictive control to a multivariable chemical reactor. *Control Engineering Practice*, tom 11, nr 11, s. 1315-1323.