

# IDENTYFIKACJA SYSTEMÓW NIELINIOWYCH PRZY POMOCY KERNELOWEGO ALGORYTMU LMS Z OGRANICZENIEM ZASOBÓW

**Dominik Rzepka, Piotr Otfinowski**

AGH Akademia Górniczo-Hutnicza, Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki

**Streszczenie.** W artykule zaprezentowano zastosowanie nowej, nieliniowej wersji algorytmu LMS wykorzystującej funkcje kernelowe do identyfikacji systemów nieliniowych. Aby ograniczyć ilość wektorów nośnych, będących niezbędnym elementem algorytmów opartych o metody kernelowe zastosowano kryterium selekcji. Nowy wektor wejściowy jest przyjmowany do słownika, a następnie w słowniku wyszukiwany i usuwany jest wektor, który ma najmniejszy wpływ na tworzony model nieliniowy. Przedstawiony przykład identyfikacji systemu nieliniowego potwierdza skuteczność porównywalną do algorytmów wykorzystujących większą liczbę wektorów nośnych.

**Słowa kluczowe:** metody kernelowe, uczenie maszynowe, regresja nieliniowa, algorytm LMS, dobór wektorów nośnych

## IDENTIFICATION OF NONLINEAR SYSTEMS USING FIXED BUDGET KERNEL LMS ALGORITHM

**Abstract.** In this paper a new version of kernel normalized least mean squares algorithm is applied to identification of nonlinear system. To maintain a fixed amount of support vectors, requisite for practical kernel-based algorithm, a pruning criterion is used. After admitting a new input vector to the dictionary, a least important entry is selected and discarded. A case of nonlinear system identification is presented, proving that algorithm performs well and it can maintain a performance comparable to state-of-the-art algorithms, using smaller number of support vectors.

**Keywords:** kernel methods, machine learning, nonlinear regression, least mean squares, pruning criterion

### Wstęp

Modelowanie i identyfikacja systemów, których odpowiedź jest nieliniową funkcją sygnału wejściowego jest wykorzystywana w szerokiej gamie aplikacji, od akustyki, przez automatykę, radiokomunikację, po inżynierię biomedyczną i wiele innych dziedzin techniki. Najbardziej uniwersalnym modelem stacjonarnego systemu nieliniowego jest nieskończony szereg Volterry [3]. Umożliwia on przedstawienie odpowiedzi systemu przy pomocy funkcji wielomianowej wielu zmiennych. Jego złożoność jest jednak wadą, która uniemożliwia jego praktyczne zastosowanie. Istnieją również modele Wienera i Hammersteina cechujące się mniejszą złożonością. Składają się one z systemu z pamięcią (filtr FIR) oraz funkcji nieliniowej modelowanej przez wielomian jednej zmiennej. Takie struktury nie są jednak w stanie przedstawić dowolnego systemu nieliniowego. Efektywnym rozwiązaniem tego zagadnienia jest zastosowanie metod kernelowych (nazywanych w polskiej literaturze również metodami jądrowymi; w artykule przyjęto jednak nomenklaturę stosowaną w [2]) [1, 2, 8], które charakteryzują się elegancką strukturą matematyczną, a możliwościami modelowania dorównują modelowi Volterry. Jeżeli modelowanie zostanie ograniczone do pewnego podzbioru sygnałów wejściowych, to model systemu nieliniowego może być opisany przy pomocy skończonej liczby punktów odniesienia (wektorów nośnych), dzięki czemu złożoność obliczeniowa zostanie ograniczona do poziomu umożliwiającego praktyczne zastosowanie. Identyfikacja systemu przy pomocy tego rodzaju modeli może być przeprowadzana z wykorzystaniem algorytmów adaptacyjnych, minimalizujących średniokwadratowy błąd aproksymacji. Popularnym algorytmem stosowanym powszechnie do tego celu jest LMS (ang. *least mean squares*), który cechuje się liniową złożonością obliczeniową [3]. W niniejszym artykule zostanie przedstawiona znormalizowana, kernelowa wersja LMS, wzbogacona o prosty algorytm ograniczania ilości wektorów nośnych.

### 1. Model systemu nieliniowego

Liniowy system  $U$ , z wejściem  $\mathbf{x} = (x_1, x_2, \dots, x_N) \in \mathbb{R}$  i wyjściem  $d \in \mathbb{R}$  można przedstawić jako iloczyn skalarny wektora wag  $\mathbf{w}$  i wektora wejściowego  $\mathbf{x}$ .

$$U(\mathbf{x}) = \mathbf{w}^T \mathbf{x} = w_1 x_1 + w_2 x_2 + \dots + w_N x_N \quad (1)$$

Model ten opisuje również systemy ze skalarnym wejściem  $x$  i pamięcią  $N$  próbek, w których wektor wejściowy w chwili  $n$  to  $\mathbf{x}(n) = (x_n, x_{n-1}, \dots, x_{n-N+1})$ . Jeżeli system  $U$  nie jest liniowy a funkcja  $U(\mathbf{x})$  jest ciągła to zależność wyjścia od wejścia można przedstawić przy pomocy rozwinięcia wielomianowego,

nazywanego dyskretnym szeregiem Volterry. Uogólniając twierdzenie Weierstrassa-Stone'a można udowodnić, że każda ciągła funkcja  $U(\mathbf{x})$  może być aproksymowana jednostajnie z dowolną dokładnością przez skończony szereg Volterry odpowiednio wysokiego rzędu [3]. Szereg rzędu  $m$  ma postać

$$U_m(\mathbf{x}) = \sum_{i=1}^{(m+1)^N} \mathbf{w}_{p_i}^T \mathbf{x}^{p_i} \quad (2)$$

gdzie:

$\mathbf{x} = (x_1^{p_1}, x_2^{p_2}, \dots, x_N^{p_N}) \in \mathbb{R}^N$ , a wektory  $\mathbf{p} = (p_1, p_2, \dots, p_N) \in \mathbb{R}^N$  należą do zbioru wszystkich możliwych wariacji z powtórzeniami liczb całkowitych od 0 do  $N$ . Przykładowo system z dwoma wejściami ( $N=2$ ) opisany szeregiem Volterry drugiego rzędu ma postać

$$\begin{aligned} U_2(x) &= w_{00} + w_{10}x_1 + w_{01}x_2 + w_{11}x_1x_2 + w_{20}x_1^2 + w_{00} \\ &\quad + w_{02}x_2^2 + w_{12}x_1x_2^2 + w_{21}x_1^2x_2 + w_{22}x_1^2x_2^2 \\ &= \widehat{\mathbf{w}}^T \widehat{\mathbf{x}} \end{aligned}$$

gdzie  $\widehat{\mathbf{w}}$ ,  $\widehat{\mathbf{x}}$  są wektorami  $\mathbf{w}$ ,  $\mathbf{x}$  rozszerzonymi o dodatkowe człony nieliniowe.

### 2. Metody kernelowe

Rozszerzenie wektora  $\mathbf{x} \in \mathbb{R}^N$  o dodatkowe człony nieliniowe można zapisać w postaci przekształcenia  $\varphi: \mathbb{R}^N \rightarrow \mathbf{F}$ , gdzie  $\mathbf{F}$  jest nazywane przestrzenią cech [2]. Przekształcenia  $\varphi: \mathbb{R}^N \rightarrow \mathbf{F}$  (gdzie  $\mathbf{F}$  jest przestrzenią  $K$ -wymiarową) mogą mieć różną postać, ponieważ rozszerzony wektor  $\widehat{\mathbf{x}} \in \mathbf{F}$  może sam w sobie zawierać współczynniki skalujące przy poszczególnych składowych. W przypadku wykorzystania modelu Volterry (2) wymiarowość przestrzeni cech wynosi  $(m+1)^N$ . Od wymiarowości  $\mathbf{F}$  zależy zwykle dokładność opisu systemu nieliniowego, jednak zwiększenie jej rozmiaru zwiększa ilość obliczeń potrzebnych do wykonania iloczynu skalarnego modelującego system  $U$ . Istnieje możliwość ominięcia tego problemu, poprzez zastosowanie metod kernelowych. Dla niektórych przekształceń  $\varphi: \mathbb{R}^N \rightarrow \mathbf{F}$  istnieje funkcja  $\kappa: \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}$  zwana kernelem (jądrem), która spełnia warunek Mercera [1]

$$\iint g(x) \kappa(x, y) g(x) dx dy \geq 0$$

gdzie  $g(x)$  jest dowolną funkcją całkowalną z kwadratem.

Kryterium Mercera jest więc analogiczne do kryterium definiującego formę półdefinito określoną. Kernel  $\kappa: \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}$  umożliwia obliczenie iloczynu skalarnego rozszerzonych wektorów bez bezpośredniego przejścia do wysokowymiarowej przestrzeni  $\mathbf{F}$ :

$$\varphi(\mathbf{x})^T \varphi(\mathbf{y}) = \kappa(\mathbf{x}, \mathbf{y}) \quad (3)$$

Wykorzystanie tej zależności nazywane jest trikiem kernelowym i umożliwia uzyskanie nieliniowych wersji algorytmów wykorzystujących iloczyn skalarny, poprzez zastąpienie go obliczeniem wartości funkcji kernelowej. Najczęściej wykorzystywaną funkcją kernelową jest kernel Gaussa [6, 2].

$$\kappa(\mathbf{x}, \mathbf{y}) = \exp\left(\frac{-\|\mathbf{x} - \mathbf{y}\|_2^2}{\sigma^2}\right) \quad (4)$$

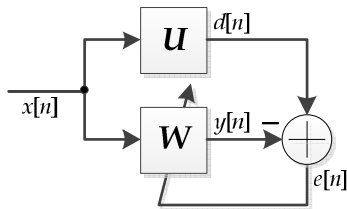
Parametr  $\sigma$  jest doborany do danych wejściowych na podstawie ich właściwości statystycznych [4].

Najbardziej znanym zastosowaniem metod kernelowych są algorytmy klasyfikatorów, zwane maszynami wektorów nośnych (ang. *support vector machines*, SVM). SVM może być również wykorzystany do regresji [7], jednak nie są one zazwyczaj stosowane jako algorytmy online. Nieco bardziej obszerny wstęp do zagadnień metod kernelowych można znaleźć w [1, 2] i [8].

### 3. Adaptacyjna identyfikacja systemu

Identyfikacja systemu polega ustaleniu zależności pomiędzy jego wejściem a wyjściem. W przypadku systemu liniowego wystarczające jest podanie na wejście  $N$  wektorów z bazy jednostkowej, aby na podstawie odpowiedzi w pełni określić wartości wektora  $\mathbf{w}$ . W systemie ze skalarnym wejściem i pamięcią odpowiada to identyfikacji poprzez podanie na wejście delty Kroneckera. Dla systemu nieliniowego identyfikacja za pomocą skończonej liczby wektorów testowych jest możliwa jedynie wtedy, gdy system ten jest opisywany przez szereg Voltery skończonego rzędu. W praktyce jednak liczba wektorów testowych musi być większa niż tego wymaga rząd szeregu Voltery, ponieważ pomiar wartości na wyjściu systemu identyfikowanego jest obciążony błędem wynikającym z szumów. Poza tym nie zawsze istnieje możliwość podania na wejście identyfikowanego systemu dowolnego wymuszenia i w takiej sytuacji identyfikacja musi ograniczyć się do analizowania dostępnego sygnału wejściowego i wyjściowego.

Efektywną metodą identyfikacji jest wykorzystanie algorytmu adaptacyjnego, którego schemat użycia prezentuje rys. 1.



Rys. 1. Identyfikacja systemu  $U$  za pomocą algorytmu adaptacyjnego

Jego celem jest zminimalizowanie wartości oczekiwanej z kwadratu błędu modelowania  $e(n)$  (7), określonego jako różnica pomiędzy wyjściem  $d(n)$  modelowanego systemu  $U$  oraz wyjściem  $y(n)$  modelu  $W$ , który podlega adaptacji

$$y(n) = \mathbf{w}^T \mathbf{x}(n) \quad (6)$$

$$\min E[\|d(n) - \mathbf{w}^T \mathbf{x}(n)\|_2^2] \quad (7)$$

Adaptacja jest realizowana poprzez modyfikację wartości wektora  $\mathbf{w}$ . Minimalizacja błędu jest przeprowadzana sukcesywnie na podstawie kolejnych wektorów  $\mathbf{x}(n)$  ze zbioru uczącego, gdzie  $n$  jest indeksem wektora wejściowego.

### 4. Kernelowy algorytm LMS

Minimalizacja zależności (7) prowadzi do rozwiązania Wienera [3]:

$$\begin{aligned} \nabla_{\mathbf{w}} E[e^2(n)] &= \nabla_{\mathbf{w}} \sum_{i=0}^n (d(i) - \mathbf{w}^T \mathbf{x}(i))^2 \\ &= -2 \sum_{i=0}^n \mathbf{x}(i) (d(i) - \mathbf{w}^T \mathbf{x}(i)) = 0 \end{aligned} \quad (8)$$

Rozwiązanie to można przedstawić w postaci układu równań liniowych, którego rozwiązanie wymaga odwrócenia macierzy, będącej operacją o złożoności obliczeniowej  $O(N^3)$ . Alternatywa w postaci rekursywnego uaktualniania macierzy odwrotnej dla każdego nowego wektora wejściowego wymaga  $O(N^2)$  operacji [3]. Dalsze zmniejszenie złożoności jest możliwe, jeżeli zamiast minimalizacji wartości oczekiwanej  $e(n)$  zastosuje się minimalizację błędu chwilowego  $e(n)$  przy pomocy metody gradientu prostego

$$\begin{aligned} \nabla_{\mathbf{w}} e^2(n) &= -2\mathbf{x}(n)(d(n) - \mathbf{w}^T \mathbf{x}(n)) = -2\mathbf{x}(n)e(n) \\ \mathbf{w}(n+1) &= \mathbf{w}(n) - \eta \nabla_{\mathbf{w}} e^2(n) \\ &= \mathbf{w}(n) + 2\eta \mathbf{x}(n)e(n) \end{aligned} \quad (9)$$

gdzie  $\eta$  jest krokiem algorytmu zapewniającym zbieżność. Algorytm ten jest nazywany filtrem adaptacyjnym LMS (ang. *least mean squares*). Często stosowaną modyfikacją jest normalizacja poprawiająca zbieżność

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\eta \mathbf{x}(n)e(n)}{\|\mathbf{x}(n)\|^2} \quad (10)$$

W przypadku systemu nieliniowego wyjście modelu  $W$  jest opisywane zależnością

$$y(n) = \widehat{\mathbf{w}}^T \varphi(\mathbf{x}(n)) \quad (11)$$

Aby możliwe było wykorzystanie triku kernelowego (4) konieczne jest przedstawienie wektora  $\widehat{\mathbf{w}}$  przy pomocy przekształcenia  $\varphi: \mathbb{R}^M \rightarrow \mathbf{F}$ . Rozwiązaniem jest zapisanie  $\widehat{\mathbf{w}}$  jako kombinacji liniowej przekształconych wektorów trenujących

$$\widehat{\mathbf{w}} = \sum_{m=1}^M \alpha_m \varphi(\mathbf{x}_m) \quad (12)$$

Wektory  $\mathbf{x}_m$  noszą nazwę wektorów nośnych, zawartych w  $M$ -elementowym słowniku. Podstawiając (12) do (11) otrzymujemy

$$\begin{aligned} y(n) &= \widehat{\mathbf{w}}^T \varphi(\mathbf{x}(n)) = \sum_{m=1}^M \alpha_m \varphi(\mathbf{x}_m)^T \varphi(\mathbf{x}(n)) \\ &= \sum_{m=1}^M \alpha_m \kappa(\mathbf{x}_m, \mathbf{x}(n)) = \boldsymbol{\alpha}^T \mathbf{k}(n) \end{aligned} \quad (13)$$

gdzie  $\mathbf{k}(n)$  jest wektorem złożonym z iloczynów skalarnych, wykonanych przy pomocy funkcji kernelowej  $\kappa(\cdot, \cdot)$ . Jego  $m$ -tym elementem jest  $\kappa(\mathbf{x}_m, \mathbf{x}(n))$ . Zależność ta stanowi odpowiednik (6) dla algorytmu liniowego. Postępując analogicznie jak w przypadku równania (7) otrzymuje się znormalizowany algorytm LMS wykorzystujący przekształcenie kernelowe (KNLMS – *kernel normalized LMS*)

$$\alpha(n+1) = \alpha(n) + \frac{\eta \mathbf{k}(n)e(n)}{\|\mathbf{k}(n)\|^2} \quad (14)$$

Należy w tym miejscu zaznaczyć, że zasada działania algorytmów z rodziny KLMS i KRLS jest powiązana z algorytmami SVM używanymi w trybie regresji. Oba rodzaje algorytmów mają na celu zamodelowanie funkcji przy pomocy dostępnych danych trenujących. Różnica pomiędzy tymi rodzinami algorytmów wynika z faktu, że algorytmy KLMS i KRLS są rozszerzeniem filtrów adaptacyjnych LMS i RLS, których nieodłączną cechą jest działanie w trybie online. Dlatego też tworzą one odrębną grupę algorytmów, opisywaną często bez odwołań do SVM i pochodnych [3, 4, 5, 6, 10, 11].

## 5. Dobór wektorów nośnych

Pełne wykorzystanie informacji zdobytych podczas adaptowania modelu  $W$  wymagałoby włączenia do zbioru wektorów nośnych  $\{x_m\}_{m=1}^M$  wszystkich wejściowych wektorów trenujących. Jednak złożoność obliczeniowa algorytmu KLMS jest wprost proporcjonalna do rozmiaru słownika  $M$ , więc korzystnie jest ją ograniczyć poprzez selekcję wektorów trenujących, przyjmowanych do słownika.

Ponieważ wektor  $\hat{\mathbf{w}}$  jest przedstawiany jako kombinacja liniowa przekształconych wektorów nośnych, to bezcelowe jest przyjmowanie do słownika wektorów, które po przekształceniu będą liniowo zależne od wektorów obecnych w słowniku. Jednak ponieważ wymiarowość przestrzeni  $F$  jest wysoka, to niskie jest prawdopodobieństwo, że dwa wejściowe wektory będą liniowo zależne. Dlatego też kryterium przyjmowania nowych wektorów testuje ich przybliżoną liniową zależność [4]. W tym celu obliczana jest taka reprezentacja  $\mathbf{a}$  nowego wektora w bazie złożonej z wektorów nośnych, która minimalizuje długość wektora błędu reprezentacji.

$$\min_{\mathbf{a}} \left\| \sum_{m=1}^M \mathbf{a}^T \varphi(\mathbf{x}_m) - \varphi(\mathbf{x}(n)) \right\| < \nu \quad (15)$$

Jeżeli jest on mniejszy od zadanego progu  $\nu$ , oznacza to, że wektor ten jest w przybliżeniu liniowo zależny od wektorów obecnych w słowniku. Ponieważ da się go dobrze przedstawić przy pomocy dotychczasowych wektorów, a więc nie jest konieczne przechowywanie go w słowniku. Do obliczania reprezentacji  $\mathbf{a}$  w bazie wektorów nośnych (15) potrzebne jest odwrócenie macierzy, które – korzystając z algorytmu rekursywnego [5] – można wykonać przy użyciu  $O(M^2)$  operacji. Ponieważ sam algorytm KLMS ma złożoność  $O(M)$ , to stosowanie tego kryterium podwyższyłoby całkowitą złożoność. Alternatywą dla niego jest kryterium koherencji [5] o złożoności liniowej. Zamiast kosztownej aproksymacji w bazie wektorów nośnych, nowy wektor jest oddzielnie aproksymowany każdym z wektorów nośnych, z użyciem niekosztownego iloczynu skalarnego.

$$\min_{\mathbf{a}, \nu} \left\| \mathbf{a}_m \varphi(\mathbf{x}_m) - \varphi(\mathbf{x}(n)) \right\| < \nu \quad (16)$$

Jeżeli któraś z aproksymacji daje błąd mniejszy od zadanego progu  $\nu$ , to wektor może być uważany za w przybliżeniu liniowo zależny od wektorów obecnych w słowniku.

Oba te kryteria ograniczają ilość nowych wektorów przyjmowanych do słownika, ale nie umożliwiają wprowadzenia górnego ograniczenia ilości wektorów nośnych. Aby było to możliwe, należy użyć kryterium selekcji, umożliwiającego usunięcie również wektorów obecnych w słowniku [11][11]. Najprostszym tego rodzaju kryterium jest określenie wkładu, jaki każdy z wektorów nośnych wnosi do modelu systemu  $U$  [9, 10]. Wagi przypisane każdemu z wektorów nośnych są zapisane w wektorze  $\alpha$ , a więc kryterium selekcji najmniej ważnego wektora nośnego można zapisać w formie

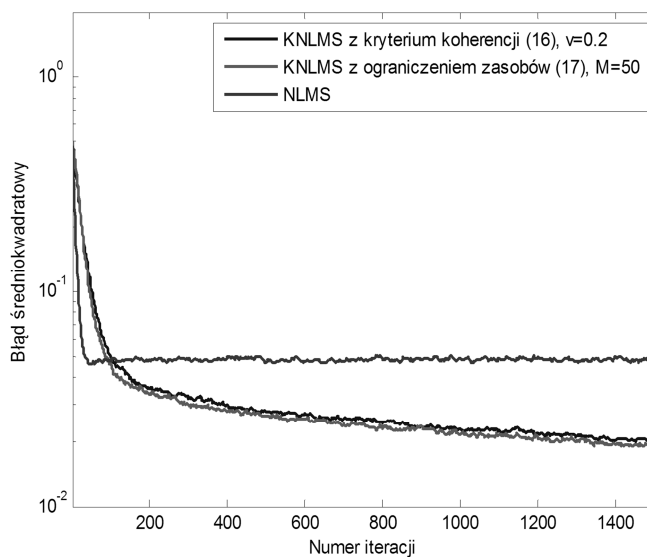
$$\min_m |\alpha_m| \quad (17)$$

Przed zastosowaniem kryterium (17) nowy wektor jest przyjmowany do słownika z wagą  $\alpha_{m+1}$  równą wartości  $e(n)$  i bierze udział w procesie selekcji. Dzięki temu wektory wejściowe, przy których błąd modelowania  $e(n)$  był wysoki są z dużym prawdopodobieństwem przyjmowane do słownika, ponieważ istniejąca baza wektorów nośnych nie umożliwiała dobrego przybliżenia wartości na wyjściu modelowanego systemu. Kryterium (17) umożliwia więc ustalenie maksymalnej ilości  $M$  wektorów nośnych w słowniku. Jednocześnie jedyną operacją wymaganą do jego zastosowania jest sprawdzenie maksymalnej wartości wektora  $\alpha$ , nie wymagające mnożenia, co jest cechą cenną z punktu widzenia implementacji algorytmu.

## 6. Symulacje

Aby sprawdzić jakość identyfikacji przy pomocy kernelowego algorytmu LMS przeprowadzone zostały testy symulacyjne. Identyfikowany system nieliniowy składał się z filtru FIR o transmitancji  $U(z) = 1 - 0.8326z^{-1} + 0.6656z^{-2} - 0.7153z^{-3}$ , którego wyjście było przekształcane funkcją nieliniową  $f(x) = \tanh(x)$ . Wejściowy sygnał  $\mathbf{x}(n)$  składał się z wektorów o wartościach losowanych jednorodnie ze zbioru  $\{-1, -1/3, 1/3, 1\}$ . Pełny słownik wektorów nośnych dla takiego sygnału liczy 256 elementów. Do sygnału z wyjścia systemu nieliniowego został dodany szum na poziomie  $SNR = 30\text{dB}$ .

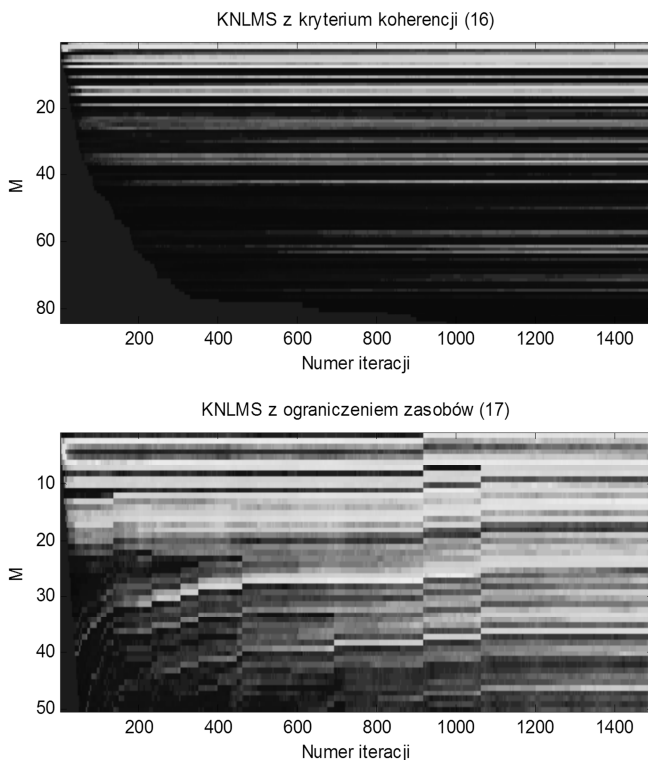
Symulacja została przeprowadzona z kernelem Gaussa (4), którego parametr  $\sigma = 1$  został dobrany eksperymentalnie. Wartość kroku algorytmu KNLMS wynosiła  $\eta = 0.2$ . Wyniki, uśrednione ze 100 symulacji Monte Carlo z losowymi danymi wejściowymi, zostały przedstawione na rys. 2.



Rys. 2. Porównanie błędów identyfikacji dla znormalizowanego kernelowego LMS (KNLMS) z różnymi kryteriami ograniczania zasobów oraz liniowego znormalizowanego LMS (NLMS).

Wartość błędów średniokwadratowych po każdej iteracji trenującej była sprawdzana na zbiorze testowym, złożonym z 200 losowych wektorów. Przetestowane zostały dwa kryteria selekcji – (16) i (17). W obu przypadkach końcowy błąd średniokwadratowy po 1500 iteracjach był zbliżony do wartości  $MSE = 0.0198$ . Istotna różnica dotyczy ilości wektorów nośnych wykorzystanych przy identyfikacji. W przypadku kryterium koherencji (16) w 100 próbach wyniosła ona średnio  $M \approx 83$ , natomiast dla kryterium (17) była zgodna z narzuconym ograniczeniem  $M = 50$ . Kryterium (17) umożliwia więc znacznie efektywniejszą identyfikację systemu nieliniowego. Dla porównania wykonano również identyfikację systemu nieliniowego przy pomocy algorytmu NLMS modelującego jedynie zależności liniowe. Wyniki pokazują wyraźną różnicę w możliwościach modelowania pomiędzy algorytmami KNLMS i NLMS. Szybka zbieżność algorytmu NLMS wynika z niskiej wymiarowości wektora wag

$w \in \mathbb{R}^4$ . W przypadku algorytmu KNLMS wymiarowość wektora  $\alpha \in \mathbb{R}^M$  zmieniała się w trakcie jego działania, jednak w obu przypadkach była znacznie większa niż 4, co spowolniło adaptację. Proces adaptacji i przyjmowania nowych wektorów można zaobserwować na rys. 3, obrazującym wartości bezwzględne elementów wektora  $\alpha$ . W przypadku kryterium koherencji (16), ograniczającego ilość przyjmowanych wektorów, wzrost wielkości słownika jest powolny, jednak ich ilość jest ostatecznie dość wysoka. Zmiany wartości wag  $\alpha$  są niewielkie. W przypadku użycia kryterium selekcji (17) algorytm przyjmuje pierwsze  $M$  wektorów bez względu na ich wartości, a następnie szybko odrzuca część z nich, zastępując je lepszymi. Zmiany wartości wag  $\alpha$  są szybsze a największa zmienność występuje wśród nowoprzyjętych wektorów (wysokie indeksy  $m$ ). Wynika to z faktu, że początkowa grupa wektorów ma dominującą rolę w opisie funkcji  $U(\mathbf{x})$ , a kolejne wektory spełniają rolę pomocniczą. W obu przypadkach stabilny skład słownika wektorów nośnych jest uformowany po ok. 1000 iteracji algorytmu.



Rys. 3. Wartości  $|\alpha_m|$  dla różnych kryteriów ograniczania zasobów

Porównując oba przebiegi wartości na rys. 3 można również zauważyć, że algorytm z kryterium koherencji daje wektor wag  $\alpha$  o szerokim zakresie wartości składowych – na rysunku są obecne zarówno obszary odpowiadające wysokim wartości  $|\alpha_m|$  jak i ciemne dla niskich wartości  $|\alpha_m|$ . Natomiast algorytm z ograniczeniem zasobów generuje wektor  $\alpha$  w którym dominują wartości wysokie. Wynika to z faktu, iż najniższe wartości  $\alpha_m$  są odrzucane. Mimo to oba algorytmy wykazują podobną efektywność w zakresie minimalizowania błędu aproksymacji. Tak więc algorytm (17) skutkuje zapisaniem podobnej ilości informacji przy użyciu mniejszej ilości liczb, przy takiej samej precyzji numerycznej, a więc dokonuje pewnego rodzaju kompresji.

## 7. Podsumowanie

W artykule została zaprezentowana metoda identyfikacji systemu nieliniowego przy pomocy znormalizowanego kernelowego algorytmu LMS (KNLMS). Wykorzystanie metod kernelowych jest alternatywą dla modelowania systemów nieliniowych przy pomocy szeregu Volterry, który narzuca wysoką wymiarowość poszukiwanego wektora wag. W przypadku metod kernelowych możliwe jest selektywne ograniczenie ilości wektorów nośnych używanych do reprezentacji identyfikowanego systemu. Zaprezentowana prosta metoda ograniczania rozmiaru słownika umożliwia efektywne wybieranie wektorów nośnych, osiągając przy tym dokładność modelowania porównywalną ze stosowanym dotychczas w algorytmie KNLMS kryterium koherencji.

## Literatura

- [1] Aronszajn N.: *Theory of reproducing kernels*. Transactions of the American Mathematical Society, 68(3)/1950, s.337–404.
- [2] Bartkowiak, A.: *Wykłady nt. Sieci Neuronowych: w11 Kernele, siecie SVM i sieci GDA*. <http://www.ii.uni.wroc.pl/~aba/>
- [3] Diniz P. S.: *Adaptive Filtering: Algorithms and Practical Implementation*. Kluwer Academic Publishers, Norwell, MA, USA, 2 edition, 2002.
- [4] Engel Y., Mannor S., Meir R.: *The kernel recursive least squares algorithm*. IEEE Transactions on Signal Processing, 52/2003, s.2275–2285.
- [5] Honeine P., Richard C., Bermudez J. C. M.: *On-line nonlinear sparse approximation of functions*. Information Theory, 2007. ISIT 2007. IEEE International Symposium on, 2007, s. 956–960.
- [6] Liu W., Principe J. C., Haykin S.: *Kernel Adaptive Filtering*. John Wiley & Sons, Inc., 2010.
- [7] Schölkopf B., Smola A.: *A tutorial on support vector regression*. Journal Statistics and Computing, vol. 14, iss. 3, August 2004, p. 199–222.
- [8] Schölkopf B., Smola A.: *Learning With Kernels*. Cambridge, MA: MIT Press, 2002.
- [9] Suykens J., Brabanter J. D., Lukas L., Vandewalle J.: *Weighted least squares support vector machines: robustness and sparse approximation*. Neurocomputing, 48(14)/2002, s.85–105.
- [10] Van Vaerenbergh S., Santamaria I., Liu W., Principe J.: *Fixed-budget kernel recursive least-squares*. Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on, 2010, s. 1882–1885.
- [11] Van Vaerenbergh S., Via J., Santamaria I.: *A sliding window kernel RLS algorithm and its application to nonlinear channel identification*, Proceedings of the International Conference on Accoustics, Speech and Signal Processing 5/2006, s.789–792.

**Mgr inż. Dominik Rzepka**  
e-mail: dominik.rzepka@agh.edu.pl

Ukończył studia na AGH na kierunku Elektronika (2009), a obecnie jest uczestnikiem studiów doktoranckich w Katedrze Elektroniki AGH, współpracując z Zespołem Widzenia Komputerowego i Systemów Wbudowanych. Jego zainteresowania obejmują radiowe techniki transmisji danych, przetwarzanie sygnałów cyfrowych, uczenie maszynowe i sprzętową implementację algorytmów.



**Mgr inż. Piotr Otfinowski**  
e-mail: piotr.otfinowski@agh.edu.pl

Ukończył Akademię Górniczo-Hutniczą w 2009, kierunek: Elektronika i Telekomunikacja, specjalność: Sensory i mikrosystemy. Obecnie jest na trzecim roku Studiów Doktoranckich na Wydziale EAIIE, AGH. Pracuje jako asystent w Katedrze Metrologii, AGH. Jego głównym tematem badań naukowych są układy scalonych przetworników analogowo-cyfrowych.

