

Estymacja i generacja sygnałów szumowych z zastosowaniem procesorów masowo-równoległych

Streszczenie: W ostatnich latach wiele badań i publikacji poświęcono analizie, występujących w różnych środowiskach, sygnałów szumowych. Efektywne przetwarzanie tych szerokopasmowych zakłóceń wymaga jednak znacznej mocy obliczeniowej, którą dysponują współczesne procesory o architekturze masowo-równoległej. W artykule przedstawiono implementację algorytmu EM, wykorzystywanego do estymacji i generacji sygnałów szumowych, zrealizowaną na procesorze graficznym (GPU - graphics processing unit).

Abstract: A large number of researchers have been studied the characteristics of noise processes encountered in different environments. Efficient processing of the wideband disturbances demands the computational performance, which can be delivered by the massively parallel processors. The paper presents an implementation of the EM algorithm for estimating and generating signals with an arbitrary probability density function on a graphics processing unit (GPU). **The use of massively parallel processors for estimating and generating noise signals.**

Słowa kluczowe: procesory masowo-równoległe, sygnały szumowe, algorytm EM.

Keywords: massively parallel processors, noise signals, EM algorithm.

Wprowadzenie

We współczesnych systemach i sieciach telekomunikacyjnych występuje wiele niepożądanych sygnałów szumowych, które zakłócają transmitowane w nich sygnały informacyjne. Zakłócenia te wytwarzane są zarówno w sposób naturalny jak i przez układy i systemy techniczne. Wiele zakłóceń o charakterze naturalnym jest modelowanych jako szum AWG (ang. *additive white Gaussian*), czyli addytywny, biały i gaussowski. Analiza i przetwarzanie sygnałów w systemach telekomunikacyjnych w obecności tego rodzaju zakłóceń nie sprawia wielu trudności pod warunkiem, że stosunek mocy sygnału do mocy szumu (SNR) jest poniżej określonego poziomu. Jednak w rzeczywistości systemy telekomunikacyjne są rzadko zakłócanie jedynie przez biały szum gaussowski [1]. Szumy generowane przez urządzenia i systemy techniczne jak i znaczna część zakłóceń o charakterze naturalnym są sygnałami impulsowymi [1, 2]. W celu analizy wpływu tego rodzaju zakłóceń na transmitowane sygnały niezbędne jest stosowanie odpowiednich algorytmów estymacji i generacji tych niegaussowskich sygnałów.

Estymacja gęstości prawdopodobieństwa

Metody estymacji gęstości prawdopodobieństwa można podzielić na trzy podstawowe grupy: parametryczne, nieparametryczne [3, 4, 5] oraz semiparametryczne [6]. Podstawowym zadaniem w metodach parametrycznych jest wyznaczenie wartości parametrów charakteryzujących daną gęstość na podstawie pozyskanej próby losowej. Najczęściej wartości parametrów wyznaczane są z wykorzystaniem estymatorów bazujących na teorii decyzji lub z użyciem klasycznych metod estymacji parametrycznej takich jak: metoda największej wiarygodności, metoda najmniejszych kwadratów czy metoda momentów. W przypadku braku dostatecznej wiedzy o estymowanej gęstości stosowane są metody nieparametryczne. Do tej grupy metod zaliczyć można histogram, estymator jądrowy oraz estymator k-najbliższych sąsiadów [3, 4, 5].

W przypadku estymacji gęstości histogram polega na klasyfikacji wartości zmiennej losowej do rozdzielnych klas. Zasadniczym znaczeniem dla zbieżności estymatora jest odpowiedni dobór szerokości klasy h . Przyjęcie zbyt małych wartości parametrów h prowadzi do lokalnych dopasowań i histogramu w postaci szpilek. Zaś zbyt duże wartości prowadzą do nadmiernego wygładzenia, maskującego specyficzne cechy estymowanego rozkładu. Przyjmując

średniokwadratową miarę dopasowania estymatora wartości parametru h można wyznaczyć z następującej zależności:

$$(1) \quad h_j = \left[\frac{6}{n \int f'(X)^2 dX} \right]^{1/3},$$

gdzie: n jest liczbą obserwacji zmiennej losowej X , a h_j j -tą składową szerokości klasy h . Wartości poszczególnych składowych szerokości klasy h zależą od nieznannej wartości estymowanej gęstości f . Jedną z podstawowych wad histogramu jest jego nieciągłość. Wady tej pozbawiony jest estymator jądrowy, który opisany jest następującym równaniem:

$$(2) \quad \hat{f}(X) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{X - X_i}{h}\right),$$

gdzie n jest liczbą obserwacji zmiennej losowej X , d jej wymiarem, a h tzw. parametrem wygładzania.

Konstrukcja estymatora sprowadza się do wyboru funkcji $K(X)$ i określenia wartości parametru h . Wartość parametrów h można wyznaczyć dla danej funkcji $K(X)$ z następującej zależności:

$$(3) \quad h = \left(\frac{d \int_{R^d} K(X)^2 dX}{n \left(\int_{R^d} X_j^2 K(X) dX \right)^2 \int_{R^d} (\nabla^2 f(X))^2 dX} \right)^{\frac{1}{d+4}}.$$

Również w przypadku estymatora jądrowego wartość parametru h zależy od nieznannej wartości estymowanej gęstości f . W przypadku metody estymacji gęstości z zastosowaniem algorytmu k-najbliższych sąsiadów (k-NN) estymowana wartość gęstości opisana jest następującą zależnością:

$$(4) \quad f(X) = \frac{k}{nv(X)},$$

gdzie k jest częścią n elementowej próby losowej, która przyjmuje wartości z obszaru v . Obszar v można określić za pomocą następującej zależności:

$$(5) \quad v(X) = k_d R^d(X) = \frac{\pi^{d/2}}{\Gamma\left(\frac{d+2}{2}\right)} R^d(X),$$

gdzie k_d jest objętością d -wymiarowej kuli jednostkowej, R^d jest odległością pomiędzy wartością X , a k -tym najbliższym sąsiadem, zaś Γ , oznacza funkcję gamma.

Podstawową wadą metod nieparametrycznych jest konieczność pamiętania całej próby losowej co wiąże się ze zwiększeniem wymaganej ilości pamięci i mocy obliczeniowej niezbędnej do estymacji rozkładu gęstości. Wady tej w znacznym stopniu pozbawione są estymatory semiparametryczne [6], łączące uniwersalność metod nieparametrycznych z szybkością metod parametrycznych. W metodach tych estymowany rozkład gęstości stanowi kombinację rozkładów o różnych parametrach. Przykładem metod semiparametrycznych są estymatory, w których wykorzystano sieci neuronowe o radialnych i hiperradialnych funkcjach bazowych (RBF, HRBF) [6].

Sieć neuronowa RBF działa na zasadzie wielowymiarowej aproksymacji, odwzorowując n różnych wektorów wejściowych X_i d -wymiarowej przestrzeni wejściowej w zbiór liczb rzeczywistych. Odwzorowanie to określa strukturę sieci, składającej się z trzech warstw. Warstwa wejściowej odpowiadającej za przechowywanie wartości sygnału wejściowego i przekazywanie go do warstwy ukrytej. Warstwy ukrytej, w której neurony opisane są nieliniową funkcją aktywacji oraz warstwy wyjściowej realizującej ważoną kombinację wyjść neuronów z warstwy ukrytej. Zgodnie z tym funkcja opisująca sieć RBF wyrażona jest następującą zależnością:

$$(6) \quad f(X) = \sum_{i=1}^n w_i \cdot G_i(X),$$

gdzie n jest liczbą neuronów w warstwie ukrytej, a $G_i(X)$ funkcją aktywacji i -tego neuronu w warstwie ukrytej. Najczęściej stosowaną funkcją $G_i(X)$ jest funkcja Gaussa, która dla sieci typu RBF przyjmuje następującą postać:

$$(7) \quad G_i(X) = \frac{1}{(2\pi)^{n/2} \det(\Sigma_i)} \exp\left(-\frac{(X - c_i)^T \Sigma_i^{-1} (X - c_i)}{2}\right),$$

gdzie Σ_i jest macierzą kowariancji.

Celem uczenia sieci RBF jest takie uporządkowanie neuronów (tj. określenie położenia centrów, wartości dyspersji i wag sieci), które zminimalizuje błąd popełniany przy estymacji gęstości prawdopodobieństwa parametrami poszczególnych neuronów. Zadanie to może być realizowane np. z zastosowaniem algorytmów opartych o proces samoorganizacji czy z zastosowaniem algorytmu EM (ang. *Expectation – Maximization*) [7].

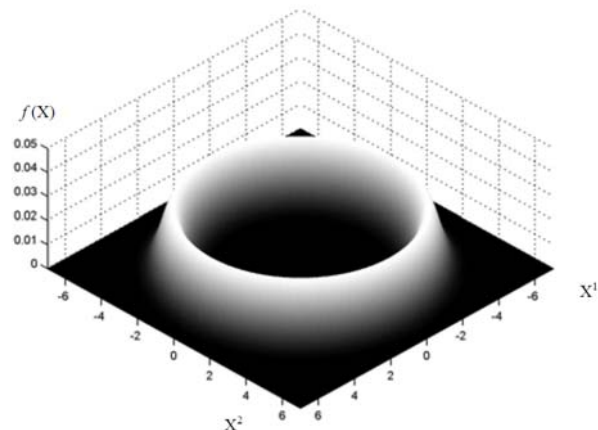
Zastosowanie estymatora neuronowego opisanego równaniem 6 umożliwia również prostą metodę generacji przebiegów losowych z wykorzystaniem metody kompozycji. W metodzie tej wynikowa gęstość prawdopodobieństwa stanowi mieszkankę prostszych rozkładów prawdopodobieństwa zgodnie z następującym równaniem:

$$(8) \quad f(X) = \sum_{i=1}^k p_i \cdot f_i(X) \quad \left(p_i \geq 0, \sum_{i=1}^k p_i = 1 \right).$$

Generacja zmiennej losowej realizowana jest w tej metodzie w dwóch krokach. W kroku pierwszym losowana jest liczba naturalna l taka, że $P(I=l) = p_l$, a w kroku drugim losowana jest liczba ze znanego rozkładu $f_l(X)$.

Wykorzystanie konkretnej metody estymacji gęstości prawdopodobieństwa uzależnione jest od kilku czynników. Do najważniejszych zaliczyć można wiedzę apriory o estymowanym rozkładzie (np. typ rozkładu, wymiar zmiennej losowej itp.), nieobciążalność, zgodność i efektywność estymatora oraz możliwość estymacji rozkładu zarówno na podstawie całej próby losowej jak i w czasie rzeczywistym na podstawie bieżących wartości zmiennej losowej.

Porównanie jakości estymatorów, estymujących gęstość prawdopodobieństwa na podstawie 1500 elementowej próby losowej pobranej z gęstości prawdopodobieństwa dwuwymiarowej zmiennej losowej $f(X^1, X^2)$ przedstawionej na rysunku 1, znajduje się w tabeli 1. Wybór estymowanej gęstości został dokonany tak, aby w jak największym stopniu zobrazować wady i zalety poszczególnych metod estymacji, a wielkość próby stanowiła kompromis pomiędzy jakością estymatorów, a nakładami obliczeniowymi niezbędnymi do ich wyznaczenia.



Rys. 1. Testowa dwuwymiarowa gęstość prawdopodobieństwa

Tabela 1. Względny średniokwadratowy błąd estymacji dla wybranych rodzajów estymatorów wyznaczony na podstawie estymacji gęstości prawdopodobieństwa przedstawionego na rysunku 1

Rodzaj estymatora	Błąd estymatora [%]
Parametryczny (przy założeniu rozkładu normalnego)	98 %
Histogram (estymowana wartość szerokości pasm)	18,5 %
Histogram (optymalna wartość szerokości pasm)	10 %
Estymator jądrowy (estymowana wartość parametru wygładzania)	16,7 %
Estymator jądrowy (optymalna wartość parametru wygładzania)	3,3 %
Estymator kNN (estymowana wartość parametru k)	8,5 %
Estymator kNN (optymalna wartość parametru k)	5 %
Estymator neuronowy HRBF uczony algorytmem EM (25 neuronów w warstwie ukrytej, 4 iteracje)	2,7 %

Implementacja algorytmu EM na procesorze graficznym GPU

Jak widać z tabeli 1 najmniejszy błąd estymacji uzyskano dla semiparametrycznej metody stosującej sieci neuronowe RBF uczone z zastosowaniem algorytmu EM. Algorytm EM maksymalizuje wartość następującej funkcji wiarygodności:

$$(9) \quad L(\theta) = \log \prod_{n=1}^N f(X_n),$$

gdzie: θ określa parametry sieci $f_j(X)$ (6). Wyznaczenie maksimum funkcji wiarygodności jest zatem równoważne z określeniem parametrów sieci estymującej gęstość prawdopodobieństwa. Przebiega to iteracyjnie w dwóch krokach. W kroku E wyznaczone są warunkowe prawdopodobieństwa $P_{i|X_n}$ dla wszystkich N obserwacji zmiennej losowej X_n zgodnie z następującą zależnością:

$$(10) \quad P_{i|X_n}^{(t)} = \frac{G_i(X_n | C_i, \Sigma_i)^{(t)} w_i^{(t)}}{f(X_n)^{(t)}},$$

gdzie t określa krok iteracji. W kroku M wyznaczone są wartości parametrów funkcji $G_i(X)$ oraz wartości wag neuronów w warstwie wyjściowej. W przypadku zastosowania jako funkcji aktywacji neuronów w warstwie ukrytej funkcji Gaussa odpowiednie wartości estymowane są zgodnie z następującymi zależnościami:

$$(11) \quad C_i^{(t+1)} = \frac{\sum_{n=1}^N P_{i|X_n}^{(t)} X_n}{\sum_{n=1}^N P_{i|X_n}^{(t)}},$$

$$(12) \quad \Sigma_i^{(t+1)} = \frac{\sum_{n=1}^N P_{i|X_n}^{(t)} (X_n - C_i^{(t+1)})^T (X_n - C_i^{(t+1)})}{\sum_{n=1}^N P_{i|X_n}^{(t)}},$$

$$(13) \quad w_i^{(t+1)} = \frac{1}{N} \sum_{n=1}^N P_{i|X_n}^{(t)}.$$

Pomimo bardzo dobrej zbieżności algorytm EM jest rzadko stosowany w zagadnieniach związanych z analizą gęstości prawdopodobieństwa sygnałów szumowych ze względu na znaczne wymagania obliczeniowe. Zapewnienie odpowiedniej do tego celu mocy obliczeniowej może być zrealizowane z zastosowaniem współczesnych procesorów graficznych GPU [8, 9, 10].

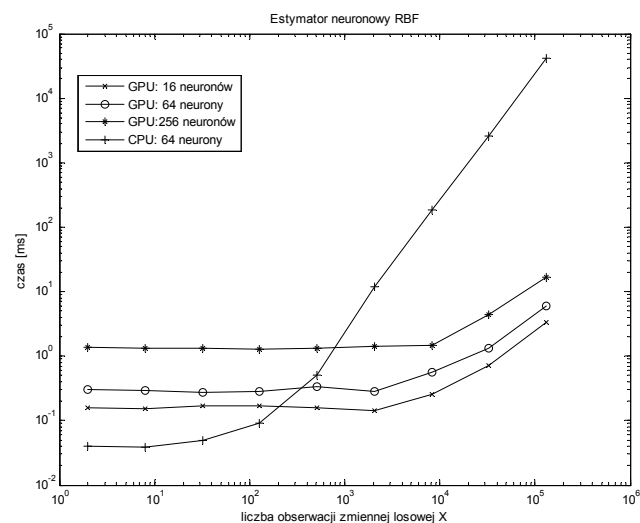
Początkowo układy graficzne funkcjonowały jako elementy systemów komputerowych o sztywno określonych funkcjach. Jednak wzrost wymagań związanych z przetwarzaniem grafiki doprowadził do tego, że współczesne procesory graficzne wykorzystują tysiące tzw. procesorów strumieniowych SP (ang. *stream processors*) (np. Radeon R9 290X posiada 2816, a GeForce GTX 780Ti - 2880 procesorów strumieniowych) umożliwiając realizację obliczeń zmiennoprzecinkowych z wydajnością przekraczającą 5000 GFLOPS. Drugą bardzo ważną cechą współczesnych kart graficznych, wpływającą na wzrost ich wykorzystywania, jest duża przepustowość pamięci wynosząca np. 320 GB/s dla procesora Radeon R9 290X, oraz 336 GB/s dla procesora GeForce GTX 780Ti.

Oczywiście, aby możliwe było wykorzystanie mocy obliczeniowej dostarczanej przez GPU niezbędne są odpowiednie narzędzia programistyczne. Problem ten dostrzegła bardzo szybko NVIDIA, która w 2007 roku wydała pierwszą wersję narzędzia CUDA (ang. *Compute Unified Device Architecture*), wspierającego proces pisanie programów z zastosowaniem produkowanych przez siebie procesorów graficznych. Jego odpowiednikiem było ATIStream dostępne dla procesorów produkowanych przez firmę ATI (obecnie AMD). Wadą obu tych środowisk jest konieczność korzystania z procesorów udostępnianych

przez ich producentów. Wady tej pozbawiony jest otwarty standard OpenCL [11] (ang. *Open Computing Language*) wspomagający tworzenie aplikacji działających na heterogenicznych platformach składających się z różnego rodzaju jednostek obliczeniowych (między innymi GPU). Standard ten jest obecnie rozwijany przez konsorcjum Khronos Group.

Architektura układów GPU zakładająca, że urządzenie wykonuje równoległe relatywnie proste obliczenia na bardzo dużej liczbie danych umożliwia efektywną realizację prezentowanego w artykule algorytmu.

W przypadku wyznaczania wartości parametrów sieci neuronowej zrównoleglenie jest realizowane poprzez jednoczesne obliczenie wartości gęstości prawdopodobieństwa dla wielu obserwacji zmiennej losowej X_n , na podstawie której estymowana jest gęstość prawdopodobieństwa. Wyniki pomiarów czasu realizacji wyznaczania wartości $f(X_n)$, w zależności od liczby obserwacji zmiennej losowej X_n i liczby neuronów w warstwie ukrytej estymatora neuronowego opisanego równaniem 6, zostały przedstawione na rysunku 2.

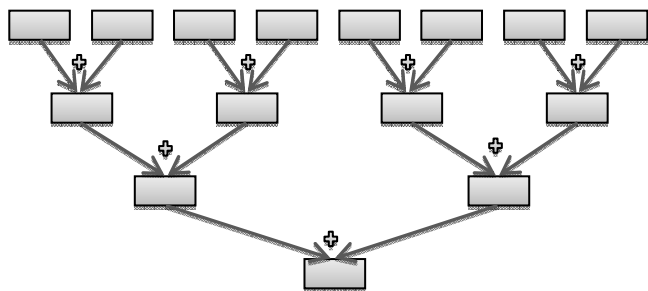


Rys. 2. Średni czas wyznaczania wartości gęstości prawdopodobieństwa w zależności od liczby obserwacji zmiennej losowej X_n i liczby neuronów w warstwie ukrytej obliczony na podstawie 10.000 realizacji przy wykorzystaniu GPU AMD Radeon HD5830 i procesora AMD4055e w środowisku OpenCL

Jak widać z rysunku 2 stosowanie procesora GPU jest efektywne obliczeniowo dopiero dla liczby obserwacji zmiennej losowej X_n przekraczającej 1000. Przy czym efektywność obliczeniowa wyraźnie wzrasta dla dużej liczby jednocześnie realizowanych obliczeń i umożliwia nawet 1000-krotną redukcję czasu obliczeń dla liczby obserwacji zmiennej losowej X_n przekraczającej 10⁵. Z wykresu widać także praktycznie liniowy wzrost czasu wykonywania obliczeń w zależności od liczby neuronów w warstwie ukrytej.

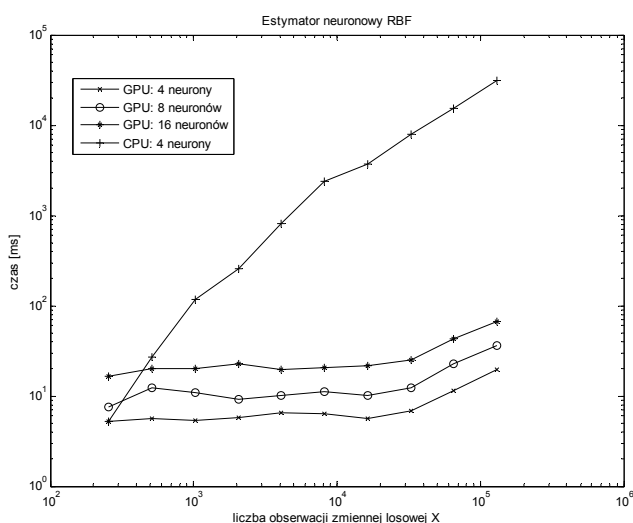
Zrównoleglenie obliczeń realizowane jest również w kroku E algorytmu EM do jednoczesnego wyznaczenia warunkowych prawdopodobieństw $P_{i|X_n}$ dla wszystkich N obserwacji zmiennej losowej X_n oraz w kroku M poprzez jednoczesne wyznaczenie wartości iloczynów $P_{i|X_n} X_n$ i $P_{i|X_n} (X_n - C_i^{(t+1)})^T (X_n - C_i^{(t+1)})$ wymaganych do określenia wartości parametrów sieci neuronowej danych wzorami (11) i (12). Również sumowanie wyznaczonych wartości iloczynów oraz warunkowych prawdopodobieństw $P_{i|X_n}$

podlega zrównolegleniu z zastosowaniem pokazanej na rysunku 3 prostej procedury redukcji.



Rys. 3. Algorytm redukcji umożliwiający wykorzystanie przetwarzania równoległego w procedurze sumowania danych

Wyniki pomiaru czasu wyznaczenia wartości parametrów sieci neuronowej estymującej gęstość prawdopodobieństwa realizowanej zgodnie z algorytmem EM (10-13) zostały przedstawione na rysunku 4.



Rys. 4. Średni czas wyznaczenia wartości parametrów sieci neuronowej estymującej gęstość prawdopodobieństwa w zależności od liczby obserwacji zmiennej losowej X_n i liczby neuronów w warstwie ukrytej obliczony na podstawie 10.000 realizacji przy wykorzystaniu GPU AMD Radeon HD5830 i procesora AMD4055e w środowisku OpenCL

Jak widać z rysunku 4 stosowanie procesora GPU jest efektywne obliczeniowo już gdy liczba obserwacji zmiennej losowej X_n przekracza 200. Przy czym i w tym przypadku efektywność obliczeniowa wyraźnie wzrasta dla dużej liczby jednocześnie realizowanych obliczeń i umożliwia osiągnięcie ponad 1000-krotnej redukcji czasu obliczeń dla liczby obserwacji zmiennej losowej X_n przekraczającej 10^5 . Podobnie jak w przypadku wyznaczenia wartości gęstości prawdopodobieństwa widoczny jest praktycznie liniowy

wzrost czasu wykonywania obliczeń w zależności od liczby neuronów w warstwie ukrytej.

Średnie czasy realizacji algorytmu wyznaczenia wartości gęstości prawdopodobieństwa (zgodnie z równaniem 6) oraz algorytmu EM (wykorzystanego do wyznaczenia wartości parametrów sieci neuronowej estymującej gęstość prawdopodobieństwa) zostały obliczone na podstawie 10.000 realizacji przy wykorzystaniu GPU AMD Radeon HD5830 i procesora AMD4055e w środowisku OpenCL [11].

Podsumowanie

Ogromne możliwości przyspieszenia obliczeń dzięki wykorzystaniu architektury procesorów masowo-równoległych powodują coraz częstsze stosowanie GPU zarówno w aplikacjach inżynierskich jak i badaniach naukowych. Przykładem takiej aplikacji jest estymacja gęstości prawdopodobieństwa sygnałów szumowych. Jak pokazano w artykule zastosowanie GPU znacznie redukuje czas wykonywania algorytmu EM. Umożliwia to na przykład stosowanie prezentowanej metody w zagadnieniach związanych z analizą i modelowaniem wpływu zakłóceń impulsowych na sygnały transmitowane we współczesnych systemach telekomunikacyjnych.

Autor: dr inż. Krzysztof Konopko, Politechnika Białostocka, Wydział Elektryczny, ul. Wiejska 45D, 15-351 Białystok, e-mail: k.konopko@pb.edu.pl;

LITERATURA

- [1] Middleton D., Non-Gaussian Noise Models in Signal Processing for Telecommunications: New Methods and Results for Class A and Class B Noise Models, *IEEE Transaction on information theory*, 45(1999), n.4, 1129 - 1149
- [2] Konopko K., The Comparison of the Probability Density Function Estimators for Signal Detection in Impulsive Noise, *Przegląd Elektrotechniczny*, (2010), n.03, 36-38
- [3] Devroye L., Non-Uniform Random Variate Generation, *Springer*, (1986)
- [4] Hörmann W., Leydold J., G. Derflinger G., Automatic Nonuniform Random Variate Generation, *Springer*, (2004)
- [5] Scott. D.W., Multivariate Density Estimation. Theory, Practice and Visualization, *Wiley*, (1992)
- [6] Konopko K., Metody wykrywania zmian struktury i parametrów systemów dynamicznych i sygnałów z zastosowaniem estymacji rozkładów prawdopodobieństwa, *Politechnika Białostocka*, (2005)
- [7] Jordan M.I., Jacobs R., Hierarchical Mixtures of Experts and the EM Algorithm, *Neural Computation* (1994), n.6, 181-214
- [8] Dąbrowski A., Pawłowski P., Stankiewicz M., Misiorek F., Fast and accurate digital signal processing realized with GPU technology, *Przegląd Elektrotechniczny*, (2012), n.06, 47-50
- [9] Konopko K., Zastosowanie procesorów masowo-równoległych w addytywnej syntezie sygnałów, *Przegląd Elektrotechniczny*, (2014), n.11, 231-234
- [10] Kirk D. B., Hwu W. W., Programming Massively Parallel Processors: A Hands-on Approach, *Morgan Kaufmann*, (2010)
- [11] Khronos Group, The OpenCL Specification Version 1.1, *Khronos Group*, (2011)