Tomasz Marciniak, Radosław Weychan,
Adam Dąbrowski
Politechnika Poznańska
Katedra Sterowania i Inżynierii Systemów
Pracownia Przetwarzania Sygnałów i Układów Elektronicznych
ul. Piotrowo 3a, 60-965 Poznań,
e-mail: Tomasz.Marciniak@put.poznan.pl

ANALIZA SZYBKIEJ IDENTYFIKACJI MÓWCY IMPLEMENTOWANYCH W ŚRODOWISKU MATLAB ORAZ CODE COMPOSER STUDIO

Streszczenie – Artykuł prezentuje wyniki badań eksperymentalnych analizy parametrów sygnału mowy w procesie identyfikacji mówcy na podstawie krótkich wypowiedzi. Eksperymenty przeprowadzono w środowisku MATLAB. Pokazano wydajność działania oprogramowania oraz skuteczność identyfikacji przy zastosowaniu kwantyzacji wektorowej. Implementacja systemu identyfikacji mówcy, działającego jako system wbudowany, wykorzystuje moduł ze zmiennoprzecinkowym procesorem sygnałowym TMS320C6713 zaprogramowanym z użyciem środowiska Code Composer Studio.

1 Identyfikacja biometryczna na podstawie sygnału mowy

Dobór parametrów akwizycji i reprezentacji sygnału biometrycznego jest istotnym elementem skuteczności oraz szybkości działania systemu identyfikacji. Współczesne metody identyfikacji biometrycznej działają głównie w oparciu o analizę obrazu np. linii papilarnych, twarzy, tęczówki, małżowiny usznej, dłoni [1, 2]. Techniki identyfikacji na podstawie sygnału akustycznego (głosu) są mniej popularne i na komercyjnym rynku biometrycznym posiadają ok. 3 %-wy udział [1]. Należy zauważyć jednak, że identyfikacja mówcy posiada szereg zalet i może służyć do kontroli dostępu do wielu usług i systemów takich jak: wybieranie głosowe opcji, bankowość telefoniczna, zakupy przez telefon, dostęp do baz danych, poczta głosowa, usługi informacyjne, dostęp do stref zamkniętych, dostęp do komputerów itp. Rozpoznawanie mówcy może być elementem multimodalnego systemu biometrycznego, badającego wiele

cech biometrycznych i tym samym pozwalającego uzyskać poprawę skuteczności identyfikacji.

Techniki rozpoznawania mówcy na podstawie indywidualnych cech sygnału mowy dzieli się na dwa typy: identyfikację i weryfikację.

Pierwszy z nich (identyfikacja) polega na określeniu, która osoba mówi spośród osób zarejestrowanych. Parametry sygnału wejściowego (sygnału mowy) są porównywane z bazą parametrów referencyjnych *N*-modeli. Następnie selektor maksimum wskazuje największe podobieństwo do modelu referencyjnego i na wyjście jest wysyłany identyfikator odpowiedniego mówcy.

Drugi typ rozpoznawania mówcy (weryfikacja) polega na akceptacji lub odrzuceniu osoby mówiącej. Mowa wejściowa jest porównywana z modelem referencyjnym, po czym zapada decyzja według określonego progu rozpoznania (ang. *treshold*), czy rezultat weryfikacji jest akceptowany, czy odrzucany. Weryfikacja jest zadaniem prostszym niż identyfikacja.

W przypadku technik rozpoznawania osób na podstawie parametrów ich głosu stosowane są m. in. techniki kwantyzacji wektorowej VQ (ang. *vector quantisation*) czy też techniki modelowania statystycznego z użyciem algorytmów GMM (ang. *Gaussian mixture modelling*) korzystających ze współczynników mel-cepstralnych MFCC (ang. *mel frequency cepstral coefficients*). Proces akwizycji danych identyfikacyjnych sygnału mowy nie powinien przekraczać 10 sekund (mowa netto).

Metody rozpoznawania mówcy można podzielić na trzy zasadnicze kategorie [3]:

- zależne od tekstu (ang. *text-dependent*) – identyfikacja jest przeprowadzana na podstawie wypowiedzenia określonego wyrazu / frazy, np. hasła, numeru PIN
- zależne od tekstu, który może się zmieniać (ang. *text-prompted*) – system wymaga od użytkownika wypowiedzenia określonego wyrazu / frazy, losowo wybranego z pewnej grupy wyrazów / fraz
- niezależne od tekstu (ang. *text-independent*) – identyfikacja jest przeprowadzana na bazie cech charakterystycznych mowy niezależnie od tego co jest wypowiadane.

System rozpoznawania mowy zawiera dwa etapy działania:

- etap nauki (ang. *training phase*) – każdy użytkownik przewidziany do korzystania z systemu musi dostarczyć do systemu próbkę swojej mowy, która będzie służyła do jego późniejszej identyfikacji. W systemach weryfikacyjnych dla każdego użytkownika określa się dodatkowo próg rozpoznania, określający granicę między jego zaakceptowaniem a odrzuceniem,
- etap testowania / działania (ang. *testing / operational phase*) – system dokonuje porównania mowy użytkownika z modelami referencyjnymi i decyduje o identyfikacji / weryfikacji.

Problemy z realizacją systemów rozpoznawania mówcy są związane ze zmiennością sygnału wejściowego (czyli mowy). Na zmianę głosu (tym samym zmianę pewnych cech sygnału mowy) wpływają różne czynniki, m.in.: czas (zmiana głosu w czasie), choroby – np. przeziębienie, szybkość mówienia, a także czynniki zewnętrze: warunki otoczenia i dźwięki tła.

2 PARAMETRYZACJA SYGNAŁU MOWY DLA IDENTYFIKACJI

Wydzielanie cech wypowiedzi określonej osoby do celów jej identyfikacji składa się z następujących etapów:

- podziału spróbkowanego sygnału na bloki o długości odpowiadającej czasowi 30 ms
- przemnożeniu bloków próbek przez funkcję okna (okno Hamminga)
- obliczeniu transformaty DFT z wykorzystaniem szybkiej transformacji Fouriera (FFT)
- przeskalowaniu sygnału do skali melowej
- wyznaczniu cepstrum (współczynników MFCC).

Następnie należy przystąpić do właściwej operacji identyfikacji. W tym celu są wykorzystywane m. in. następujące techniki:

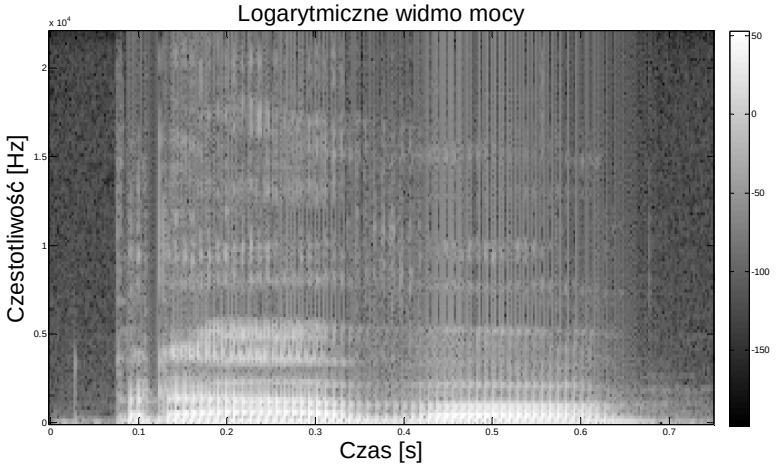
- dynamiczne dopasowanie czasowe (DTW – *dynamic time warping*) [3]
- ukryte modele Markowa (HMM – *hidden Markov models*) [4]
- kwantyzacja wektorowa (VQ – *vector quantization*), która jest także wykorzystywana w DTW i HMM [5].

Kwantyzacja wektorowa jest metodą, w której analizowane dane są modelowane za pomocą małego zbioru wektorów – centrów skupień w przestrzeni cech. W przeciwieństwie do metody GMM [6, 7], kwantyzacja wektorowa nie modeluje rozkładu prawdopodobieństwa danych. Kwantyzacja wektorowa zapewnia dużą skuteczność zarówno w przypadku rozpoznawania zależnego jak i niezależnego od tekstu dla stosunkowo niedługich wypowiedzi. Zaletą metody VQ są niewielkie wymagania pamięciowe. W artykule wykorzystano kwantyzację wektorową, której podstawy zaimplementowano w oprogramowaniu [8].

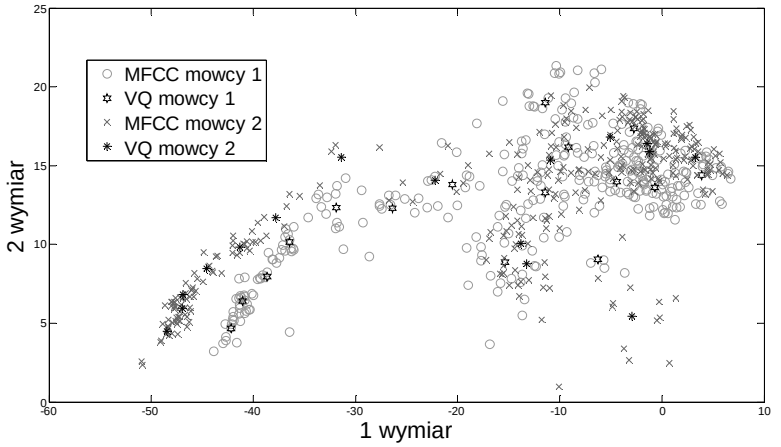
3 Badania w środowisku MATLAB/SIMULINK

Środowisko MATLAB / Simulink [9] jest oprogramowaniem, które umożliwia analizę i prawidłowy dobór parametrów reprezentujących sygnał mowy. Oprócz możliwości skorzystania z przygotowanych bibliotek z funkcjami akwizycji i przetwarzania sygnału audio, w stosunkowo prosty sposób dokonuje się wizualizacji wyników, co ułatwia ich interpreta-

cję. Przykładową wizualizację logarytmicznego widma mocy słowa „prawo” pokazano na rys. 1.



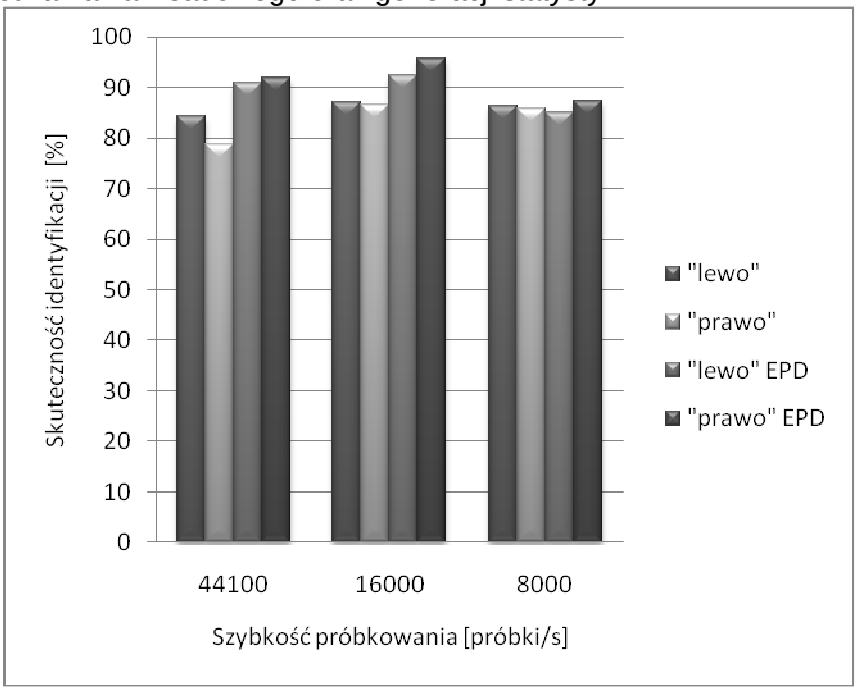
Rys. 1. Przykładowy spektrogram słowa „prawo”



Rys. 2. Ilustracja procesu kwantyzacji wektorowej (2 wymiary)

Implementacja algorytmu identyfikacji mówcy [8] w środowisku MATLAB dzieli się na dwa etapy, jak wspomniano w rozdziale 1. W etapie treningu każdy zarejestrowany mówca dostarcza próbki swojego głosu. W ten sposób system generuje model odniesienia. Do realizacji tych zadań wykorzystuje się funkcje: *wavread* – do odczytu plików WAVE, *mfcc* – do obliczania wektorów MFCC [10], *vqlbg* – zawierający algorytm

LBG (Linde, Buzo, Gray) [11] kwantyzacji wektorowej (rys. 2) do utworzenia książki kodowej oraz *disteu* – do obliczania odległości euklidesowej. Oryginalne składniki oprogramowania uzupełniono o możliwości przetwarzania wsadowego oraz generacji statystyk.



Rys. 3. Skuteczność identyfikacji

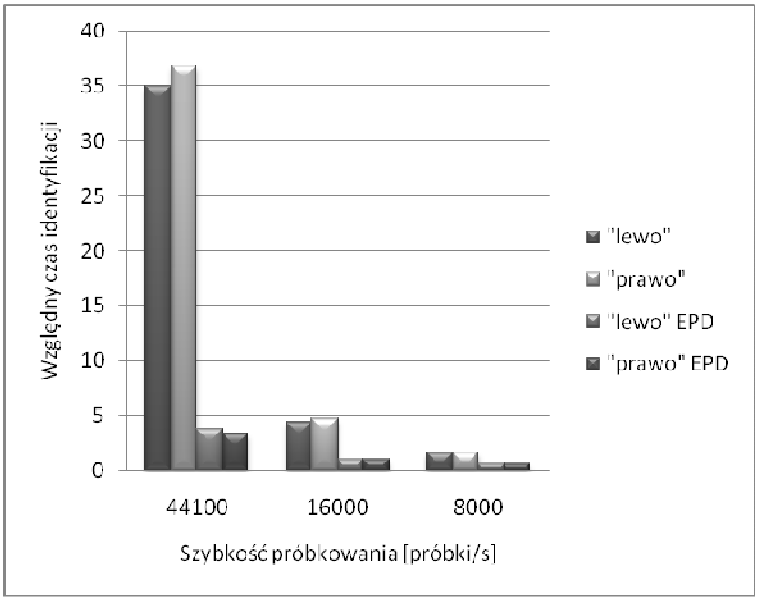
W celu przetestowania skuteczności identyfikacji, nagrano 15 mówców, którzy 30-krotnie wypowiadali słowo „prawo” oraz „lewo”. Badań dokonano dla następujących szybkości próbkowania i rozdzielczości kwantyzacji:

- 44 100 próbek/s, 16 bitów/próbkę
- 16 000 próbek/s, 16 bitów/próbkę
- 8 000 próbek/s, 8 bitów/próbkę.

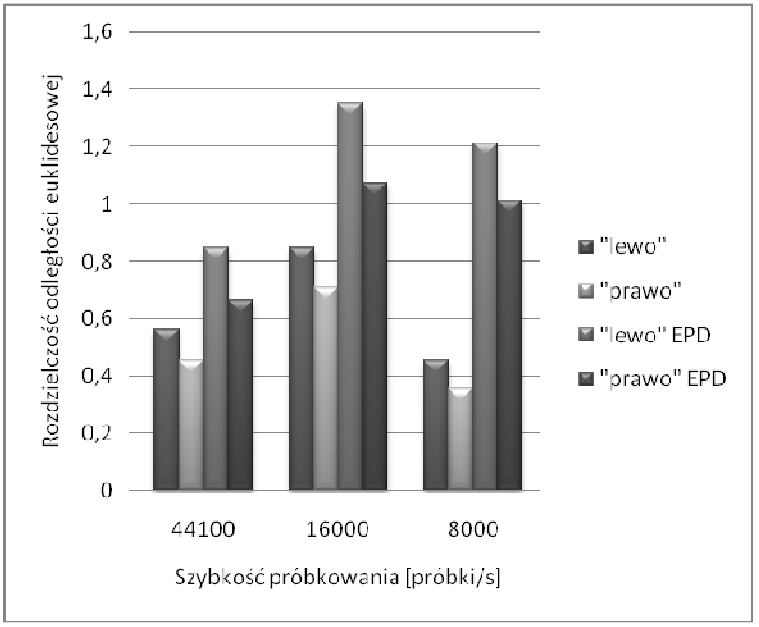
Łącznie uzyskano wykorzystano 1350 nagrań. Na rys. 3 pokazano skuteczność identyfikacji. Możemy zaobserwować, że jakość sygnału nie wpływa w dużym stopniu na poprawność rozpoznawania, ale jak zauważamy na rys. 4, wpływa na szybkość działania algorytmu.

Oryginalne oprogramowanie [8] nie zawiera etapu dokładnej detekcji początku i końca słowa (EPD – *end points detection* [12]). Dokonano więc uzupełniania algorytmu i ponownie zbadano skuteczność identyfikacji. Okazuje się, że dla badanej grupy mówców zauważamy kilkuprocentową poprawę jakości rozpoznawania. Wyniki na rys. 4 znormalizo-

wano względem słowa „prawo” z procedurą EPD przy szybkości próbkowania 16 000 próbek/s (najwyższa skuteczność identyfikacji).



Rys. 4. Porównanie względnych czasów identyfikacji



Rys. 5. Porównanie odległości euklidesowych dla VQ

Orientacyjny czas parametryzacji i obliczania odległości euklidesowej w tym przypadku trwa ok. 49 ms dla jednego słowa (MATLAB wer.7.8.0, komputer o wydajności MATLAB Bench Relative Speed=28). Należy też zauważyć, zastosowanie EPD, spowodowała, zwiększenie rozdzielczości odległości euklidesowych (tj. odległości pomiędzy najbliższym wzorcem i kolejnym wzorcem), jak pokazano na rys. 5.

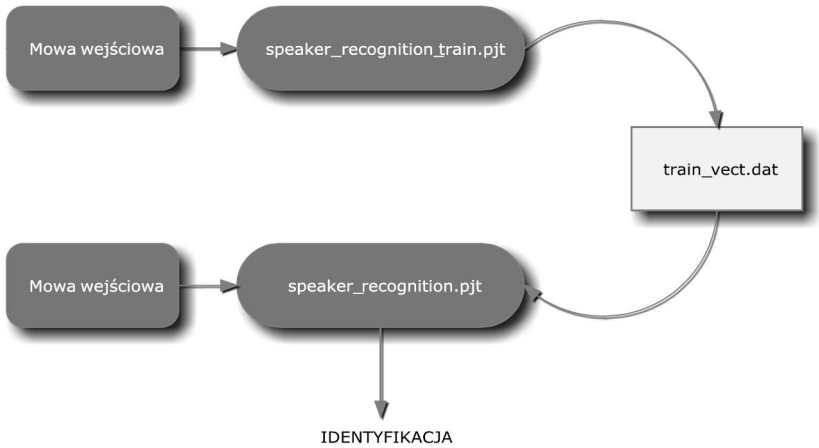
4 Implementacja identyfikacji mówcy w środowisku CODE COMPOSER STUDIO z użyciem modułu DSK C6713

Do realizacji systemu identyfikacji działającego jako system wbudowany wykorzystano elektroniczny moduł DSK ze zmiennoprzecinkowym procesorem sygnałowym C6713 firmy Texas Instruments [13]. Układ DSK jest przystosowany do pracy w aplikacjach audio z uwagi na zintegrowany stereofoniczny kodek TLV320AIC23 (AIC23) o szybkości próbkowania od 8 do 96 kpróbek/s. Moduł DSK jest programowany z użyciem zintegrowanego środowiska CSS (ang. *Code Composer Studio*) w języku C [14].

System rozpoznawania mówcy dla modułu DSK C6713, na podstawie propozycji zawartej w [15] składa się z dwóch oddzielnych projektów:

- *speaker_recogniton_train.pjt* – realizujący etap treningowy
- *speaker_recognition.pjt* – zwany etapem testowym.

Ogólną ideę działania oprogramowania ilustruje rys. 6.



Rys. 6. Schemat działania oprogramowania dla modułu DSK C6713

W pierwszym etapie są wyznaczane współczynników opisujących badaną próbkę głosu, a przy tym tworzący bazę danych mówcy. Rozpoznanie (etap testowy) również wyprowadza wektory MFCC, tym razem korzysta z próbki podlegającej identyfikacji. Obliczany jest kolejny wek-

tor opisujący daną próbkę. Wektory te są porównywane z wektorami uzyskanymi w etapie pierwszym. O zgodności dwóch próbek decyduje najmniejsza odległość między wektorami wzorcowymi, a wektorami uzyskanymi w etapie testowym. Oba projekty są do siebie podobne, zawierają te same etapy oraz korzystają z tych samych plików źródłowych. Pierwszy etap kończy się zapisaniem 20-to elementowych wektorów MFCC do pliku *train_vect.dat*, które są zapisane w formie zmiennoprzecinkowej - przykładowo {2118083.500000, 19609.228516, 23124.123047, 4914.815918, 8269.682617, 4336.663574, 1776.990234, 11475.291992, 3433.718506, 23589.646484, 5134.495117, 23331.150391, 11145.446289, 17316.996094, 7371.637207, 13281.588867, 837.974915, 4074.980713, 220.157837, 3000.194092 }. Wektory umieszcza się w pliku nagłówkowym *training.h* i dołącza do projektu *speaker_recognition.pjt*. Analizę uruchomienia oprogramowania i podstawowe badania skuteczności można znaleźć w pracy [16].

Należy zwrócić uwagę, że oryginalne oprogramowanie dla modułu C6713 nie zawiera operacji wielowymiarowej kwantyzacji wektorowej. Z tego powodu skuteczność identyfikacji jest stosunkowo niska (ok. 75% dla próby ośmioosobowej). W celu poprawy skuteczności identyfikacji mówcy zmodyfikowano oprogramowanie z [15], aby odwzorowywało implementację przygotowaną w tzw. m-plikach i zawierało operację kwantyzacji wektorowej. Tym samym pozwoliło to na uzyskanie wyników jak w przypadku badań w rozdziale 2.

5 Podsumowanie

Badania eksperymentalne przeprowadzone w środowisku MATLAB, umożliwiły przeprowadzenie analizy doboru parametrów sygnału mowy do celów identyfikacji mówcy. Biblioteki dla procesora sygnałowego C6713 pozwalają sprawnie przygotować projekt dla systemu wbudowanego pracującego w czasie rzeczywistym. W przypadku dobrej znajomości programowania w języku C oraz bibliotek dla procesora sygnałowego nie trzeba korzystać z etapu pośredniego (modelu w środowisku Simulink) pomiędzy opracowaną implementacją w środowisku MATLAB a końcową implementacją w CCS.

Uzyskana skuteczność identyfikacji na średnim poziomie około 90% pokazuje, że stosunkowo prosta metoda kwantyzacji wektorowej dobrze sprawdza się w przypadku bardzo krótkich wypowiedzi o czasie trwania poniżej 0,5 sekundy.

W kolejnym kroku badań eksperymentalnych, autorzy artykułu zbudowali bazęy wypowiedzi 25 mówców, nagrywanych w kilku sesjach. Baza zawiera krótkie wypowiedzi często pojawiające się w rozmowach telefonicznych np. „Dzień dobry”, „Dobry wieczór”, „Do widzenia”, „Moje

nazwisko". Dla takich wypowiedzi uzyskano skuteczność identyfikacji technikami GMM o kilka procent wyższą niż przy zastosowaniu kwantyzacji wektorowej VQ. Tym niemniej czas identyfikacji metodą VQ jest o połowę krótszy niż użycie GMM. Wyniki eksperymentów przestawiono w [17].

Literatura

[1] Biometrics Market and Industry Report 2009-2014, http://www.biometricgroup.com/reports/public/market_report.php

[2] Dąbrowski A., Marciniak T., Drgas Sz., Pawłowski P., Ekstrakcja informacji z obrazów, wideo i mowy w systemach ochrony i bezpieczeństwa, rozdział w monografii *Ergonomia - Technika i Technologia - Zarządzanie*, red. Marek Fertsch, ss.151-167, Wydawnictwo Politechniki Poznańskiej, Poznań 2009.

[3] Furui S., Speaker recognition, Scholarpedia (2008), http://www.scholarpedia.org/wiki/index.php?title=Speaker_recognition&printable=yes

[4] Rabiner L. R., A Tutorial on Hidden Markov Models and Selected Application in Speech Recognition, *Proc. IEEE*, vol. 72, no. 2, pp. 257-286, Feb. 1989.

[5] Wagner A., *Speaker Recognition Using a Coding-Length Based Segmentation Method for Vector Quantization* http://watt.csl.illinois.edu/~awagner/speaker_recognition_website/

[6] Reynolds D., Robust text-independent speaker identification using Gaussian Mixture Speaker Models, *IEEE Trans. Speech Audio Proc.*, Vol. 3, No. 1, 1995.

[7] Alexander A., Drygajlo A., Speaker identification: A demonstration using MATLAB, http://scgwww.epfl.ch/matlab/student_labs/2005/labs/

[8] MATLAB and Simulink for Technical Computing, http://www.mathworks.com/, 2010.

[9] DSP Mini-Project: An Automatic Speaker Recognition System http://www.ifp.uiuc.edu/~minhdo/teaching/speaker_recognition

[10] Oppenheim A.V., Shafer R.W., From Frequency to Quefrency: A History of the Cepstrum, *IEEE Signal Processing Mag.*, pp. 95-99, Sep. 2000.

[11] Linde Y., Buzo A., Gray R.M., An algorithm for vector quantizer design, *IEEE Trans. Commun.*, vol. COM-28, no. 1, pp. 84--95, Jan. 1980.

[12] Marciniak, T., Dąbrowski, A., *Influence of subband signal denoising for voice activity detection*, Elektronika – konstrukcje, technologie, zastosowania, nr 3/2009, ss. 67-70.

[13] DSKC6713 Support Page http://c6000.spectrumdigital.com/dsk6713/

[14] *Code Composer Studio IDE Subscription Service* http://focus.ti.com/docs/toolsw/folders/print/ccstudiosubscriptions.html.

[15] Chassaing R., *DSP Applications Using C and TMS320C6x DSK*, John Wiley & Sons, Inc., 2002.

[16] Nowak N., *Implementacja automatycznego rozpoznawania mówcy z zastosowaniem procesora sygnałowego TMS320C6713*, praca dyplomowa inżynierska (promotor dr inż. T. Marciniak), Politechnika Poznańska, 2009.

[17] Marciniak T., Weychan R., Drgas S., Dąbrowski A., Krzykowska A., Speaker recognition based on short Polish sequences, *Proc. of SIGNAL PROCESSING SPA'2010, Poland Section, Chapter Circuits and Systems IEEE*, pp. 95-98, Poznań, Poland, September, 23-25th 2010.

W pracy zaprezentowano wyniki osiągnięte w projektach PPBW, INDECT i DS.

ANALYSIS OF FAST SPEAKER IDENTIFICATION IMPLEMENTED IN MATLAB AND CODE COMPOSER STUDIO

Summary – This paper presents results of experimental analysis of speech signal parameters for speaker identification based on short utterances. The experiments were performed in the MATLAB environment, showing a performance of the software and an effectiveness of the identification based on the vector quantization. Implementation of the speaker identification system, working as an embedded system, uses an electronic module with the floating-point TMS320C6713 digital signal processor programmed in the Code Composer Studio environment.