BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 62, No. 3, 2014
DOI: 10.2478/bpasts-2014-0064

Biometric speech signal processing in a system with digital signal processor

T. MARCINIAK*, R. WEYCHAN, A. STANKIEWICZ, and A. DĄBROWSKI

Faculty of Computing, Chair of Control and Systems Engineering, Division of Signal Processing and Electronic Systems, Poznan University of Technology, 24 Jana Pawła II St., 60-965 Poznań, Poland

Abstract. This paper presents an analysis of issues related to the fixed-point implementation of a speech signal applied to biometric purposes. For preparing the system for automatic speaker identification and for experimental tests we have used the Matlab computing environment and the development software for Texas Instruments digital signal processors, namely the Code Composer Studio (CCS). The tested speech signals have been processed with the TMS320C5515 processor. The paper examines limitations associated with operation of the realized embedded system, demonstrates advantages and disadvantages of the technique of automatic software conversion from Matlab to the CCS and shows the impact of the fixed-point representation on the speech identification effectiveness.

Key words: biometry, speech processing, digital signal processor, Gaussian mixture models, vector quantization.

1. Introduction

Automatic speaker recognition is a scientific field with many years of fruitful research [1]. Various approaches have occurred to be useful for this purpose [2]. Among them are: neural networks, Gaussian mixture models (GMMs) and support vector machine (SVM) [3, 4] and even the generalized convolution [5].

However, almost no attention has yet been paid to the impact of practical, e.g. the digital signal processor (DSP) or even microcontroller-based realizations on the speech processing for biometric purposes [4, 6–10].

Modern microcontrollers have several features that are typical for digital signal processors. For example, microcontrollers currently manufactured by Atmel and Microchip have Harvard architecture and include a hardware multiplier that can perform operations on the fixed-point fractional numbers. Nevertheless, audio and speech signals should still be processed with a dedicated hardware, available only in DSPs. Among the relevant examples are: the maximum accuracy multiplication with special execution units, bit-reverse addressing, and the hardware acceleration for fast Fourier transformation (FFT) [1, 2]. The additional advantage of modern digital signal processors is also the optimized small power consumption, allowing these systems to work in mobile systems, powered by small batteries. Taking all these issues into account, in the presented design, the authors have chosen the TMS320C5515 DSP. Its most important features for the speech signal processing are discussed in Sec. 3.

Processing of speech signals in applications such as data compression, noise reduction, or speech/speaker recognition usually requires realization of several specific stages. Such a multistage speech processing for the speaker recognition is composed of: a parameterization of the input signal (preprocessing e.g. with the use of digital function filters [11], signal division into blocks, multiplication by a window function, determining the linear prediction coefficients or mel-cepstral coefficients with the use of the FFT), a modeling phase (such as the vector quantization and the creation of mixtures of Gaussians) and then comparing with the base pattern [2].

2. Speaker identification techniques in embedded systems

A simplified block diagram of the process realized in the tested speaker recognition system is presented in Fig. 1.

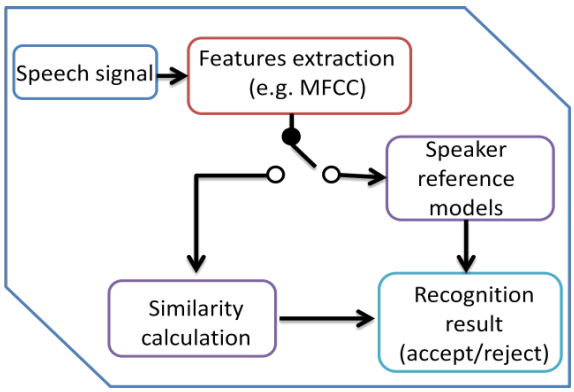

Fig. 1. Simplified block diagram for speaker recognition

Implementation of the speaker recognition system in a form of the embedded system requires consideration of the following key issues [4, 7, 8, 12]:

- quality (and denoising) of the input signal,
- influence of the input signal resolution,
- usage of the hardware and the software resources,

*e-mail: tomasz.marciniak@put.poznan.pl

- size of the system memory,
- facilities related to the implementation, e.g. libraries accessibility.

The quality of the speech recording should also be associated with encoding that occurs during transmission via telephone lines such as the GSM network or Internet network with the use of the VoIP standard. Identification of speaker in these cases is more difficult because of the speech compression provided with the phone encoders [13–15]. Therefore, the knowledge about the used voice encoder is necessary. The authors shown that the correct detection of the phone encoder and consequently, the application of the proper speaker model can improve the speaker identification accuracy even by ca. 10% [15].

3. Module with signal processor C5515

The C5000 family from Texas Instruments contains low-cost microprocessors, dedicated to industry applications, which are characterized by low power consumption, relatively large internal memory, and a large set of communication interfaces. Besides, due to the relatively high speed of data processing (internal clock speed up to 300 MHz), electronic modules equipped with these processors are very good solutions for biometric systems based on the voice processing. The TMS320C5515 eZDSP USB Stick [16], as a member of the C5000 family, has been chosen by the authors for the speech recognition experiments.

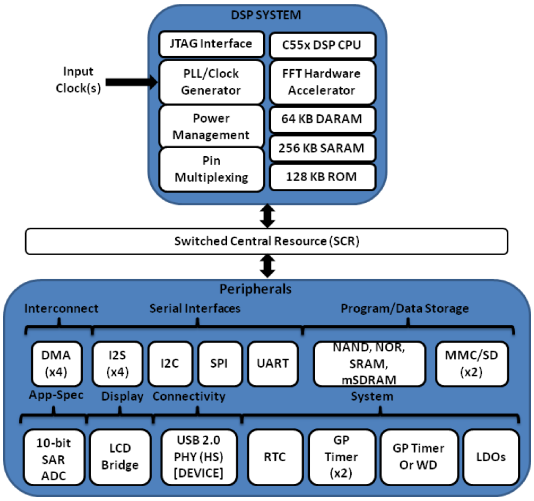

Fig. 2. Block diagram of TMS320C5515 processor (based on Ref. [9])

The TMS320C5515 signal processor has a 16-bit architecture adapted to the fixed-point arithmetic. The maximum clock rate is 120 MHz. This gives 8.33 ns for the minimum time of operation in case of one or two clock cycles per instruction. This chip is equipped with two ALUs (arithmetic logic units) and dual multipliers, which allow for 240 MMAC (million multiply-accumulate) operations per second. The FFT hardware accelerator supports 8 to 1024-point real and complex-valued FFTs and can significantly influence calculation effectiveness in the speech processing systems, in which the transformation to/from the frequency domain is the basic operation. The mentioned microprocessor is also equipped with the built-in LCD display driver and the SD card interface, connected by external memory interface (EMIF). It also allows for the serial data transmission using I2C or SPI standards. Figure 2 shows the block-design schema with this microprocessor.

The described microprocessor is equipped with 320 kB of RAM, 128 kB of ROM, and moreover with 8/16 bit EMIF controller and DMA (16 channels). The internal memory allows for acquisition of the maximum 20 seconds of speech with the assumption that the whole RAM is used for the 16-bit data resolution storage with the rate of 8000 S/s (samples per second). The memory map is presented in Fig. 3. In the case of the TMS320C5515 eZDSP USB Stick, the size of the additional accessible FLASH memory is 4 MB.

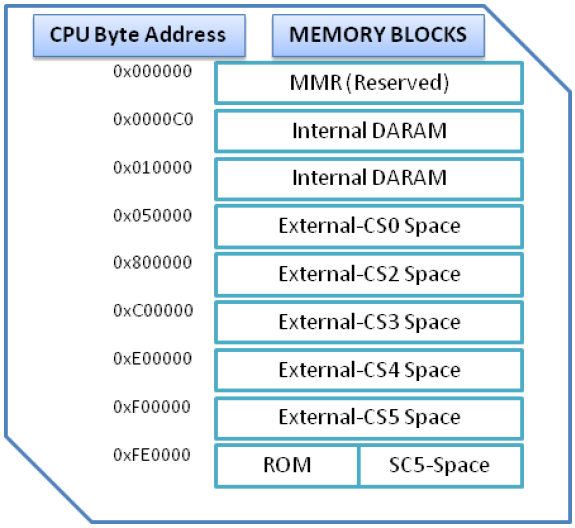

Fig. 3. Memory map of TMS320C5515 (based on Ref. 16)

4. Software conversion from Matlab to CCS

The Matlab environment (the MathWorks Company software) was used to test the developed algorithms. This environment has a set of tools allowing direct conversion of scripts written in the Matlab language to the C++ language adequate for the Code Composer Studio (the Texas Instruments Company software environment). In this way, we get an application that can be run on a particular microprocessor (of many available processor families from Texas Instruments) [17]. Most Matlab language features such as functions and matrix operations are supported. Furthermore, this mechanism makes it also possible to optimize the code structure in order to speed up the critical and the time-consuming calculations. The main features of the Matlab Coder are:

- C and C++ code generation compatible with ANSI/ISO,
- code generation for the fixed- and the floating-point calculations,
- ability to create libraries,
- tools for structures and data properties management,
- static and dynamic memory allocation for variables,

- support for Matlab language functions (i.e. matrix operations, subroutines, control instructions, structures, classes, complex numbers, and global variables) as well as specialized functions and objects included in the following libraries: Communications System Toolbox, DSP System Toolbox, and Computer Vision System Toolbox,
- ability to generate C code from the Matlab/Simulink models,
- possibility to validate the generated code even at the level of the Matlab language,
- possibility of the use of the generated code for the standalone execution, integration with other software, speeding up the algorithm through the use of executable MEX functions and implementation in digital signal processors.

For transformation of the Matlab code to the C language, determination of implementation requirements is needed. Matlab Coder is a tool that improves this stage by finding errors, checking syntax, and code compatibility with the selected device.

The Embedded Coder (as a part of the Matlab Coder) provides the code generation directly for the embedded processors, integrated fast prototyping modules, and mass production of microprocessors. Its main advantage is the easy configuration and additional optimization for features, files, and data. Particular detailed steps of the Matlab code conversion to the C language with the fixed-point representation are as follows:

1. algorithm implementation in the Matlab language
2. preparation of the test function using the implemented algorithm in order to test its behavior for various input data (fixed- and floating-point); this test should include the widest possible range of values, it is also useful to use a function that converts data from the floating-point to the fixed-point format,
3. building a MEX function for the algorithm developed in the Matlab environment,
4. running the program for the minimal and the maximal data values,
5. checking the report generated by the Matlab Coder,
6. separation data code from the function code,
7. rebuilding and validation of the MEX function until all errors are resolved and eliminated and the algorithm is optimized,
8. generation of the final C code.

During the experiments, the C code generation of the algorithm for the speaker identification has been tested. It turned out that using the Matlab Coder (MC) the main disadvantage is the generation of multiple C files. For example, for one of the main functions in the tested algorithm, namely *lmultigauss* (it calculates the log-likelihood of the multiple Gaussian mixtures) the MC creates 41 files with a total size of 98 KB after conversion to the C language. The code for this function in Matlab contains two files of 4 KB size only but it uses several built-in Matlab functions.

An additional drawback is the fact that some important functions (such as *dot* or *log*) are not supported for the fixed-point variables in Matlab. Therefore, the adaptation of the algorithm to the C language requires modification of these functions or the use of additional libraries.

5. Influence of fixed-point representation on speaker identification accuracy

In order to test effectiveness of the speaker identification, a specially prepared database was used. It contains recorded utterances of 40 individuals as described in paper [12]. Each individual spoke 6 short utterances in Polish (every utterance takes about 1 second). The utterances were repeated 10 times in three sessions, with intervals from 2 to 6 weeks between them. This gives a total number of 7200 recordings sampled at the rate of 22.5 kS/s with 16-bit resolution. For the experiment, the recordings were re-sampled down to 8 kS/s.

An analysis of the influence of the fixed-point representation on the speaker recognition accuracy was made for two types of the representation: 16-bit and 32-bit. The speaker recognition process was realized with the use of two methods: vector quantization (VQ) and Gaussian mixture models (GMMs). The detailed description of the algorithms based on these two methods was already presented in [18].

For the first experiment the algorithms for the speaker recognition were implemented in the Matlab environment. The speaker reference models were created with three randomly chosen recordings, while the remaining 27 were used during the test phase. This means that the model is based on the three-second recordings. As measures of the recognition quality the FAR/FRR (*false acceptance rate/false rejection rate*) and the EER (*equal error rate*) were used.

The results of the experiments are shown in Fig. 4. The algorithm based on 32-bit resolution floating-point data achieves accuracy of about 10% EER. By reducing the representation to the 16-bit fixed-point data, an increase of the EER up to 26% can be noticed.

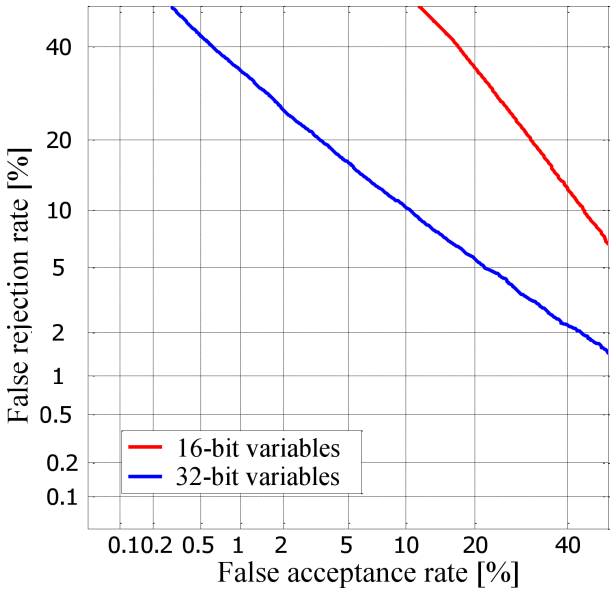

Fig. 4. FAR/FRR plots of speaker identification with the use of vector quantization (VQ)

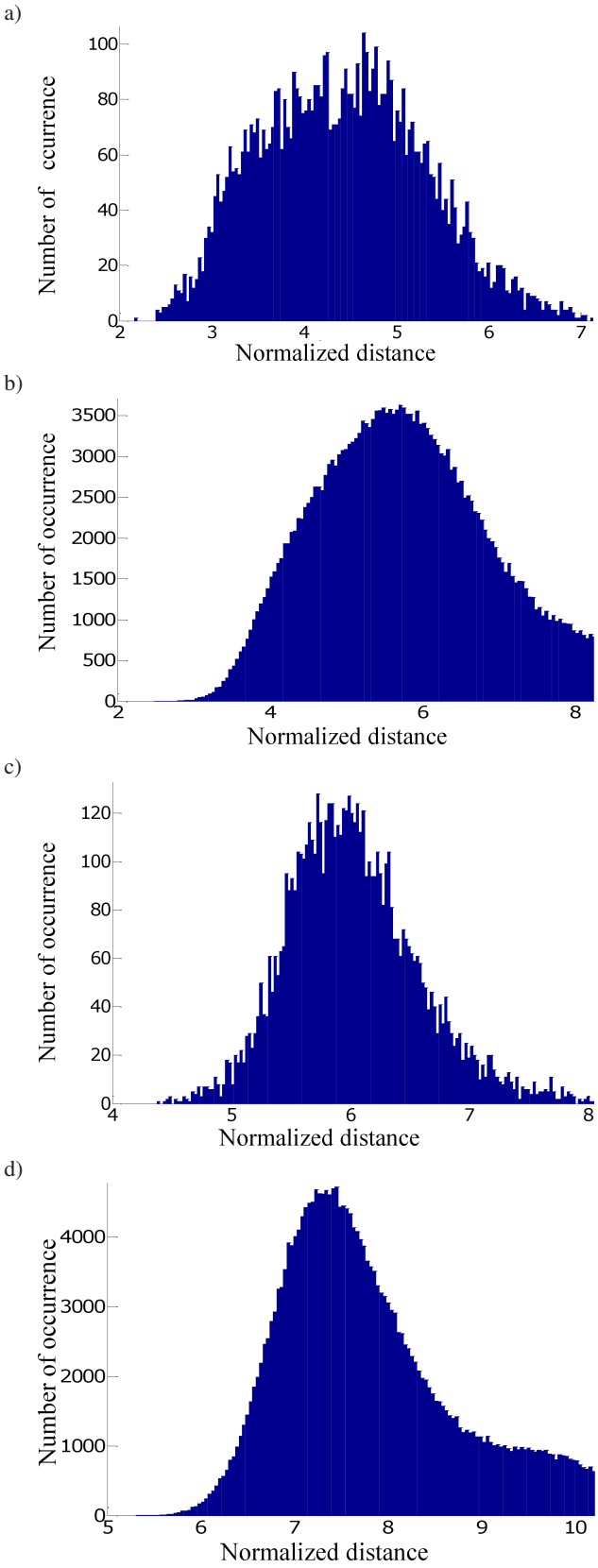

Fig. 5. Euclidean distances for VQ algorithm between the tested utterances and a) correct model for 16-bit representation, b) incorrect model for 16-bit representation, c) correct model for 32-bit representation, d) incorrect model for 32-bit representation

Figures 5a–d present histograms of the Euclidian distance between the tested utterances and the correct/incorrect models. A change in the data representation causes a shift of the mean value and an increase of the variance of the results in a comparison between the model and the test data.

An experiment using the GMM algorithm was conducted in the same way as the one described above. The results (FAR/FRR plots) are shown in Fig. 6. The EER parameter for the 32-bit floating-point data is about 31%. Reduction of resolution to 16 bits has no significant effect on this value – the EER parameter is about 32%. This is mainly due to high density of data in a small range for both 16- and 32-bit data as well as narrow range of the values of the comparison results between the test and the speaker models.

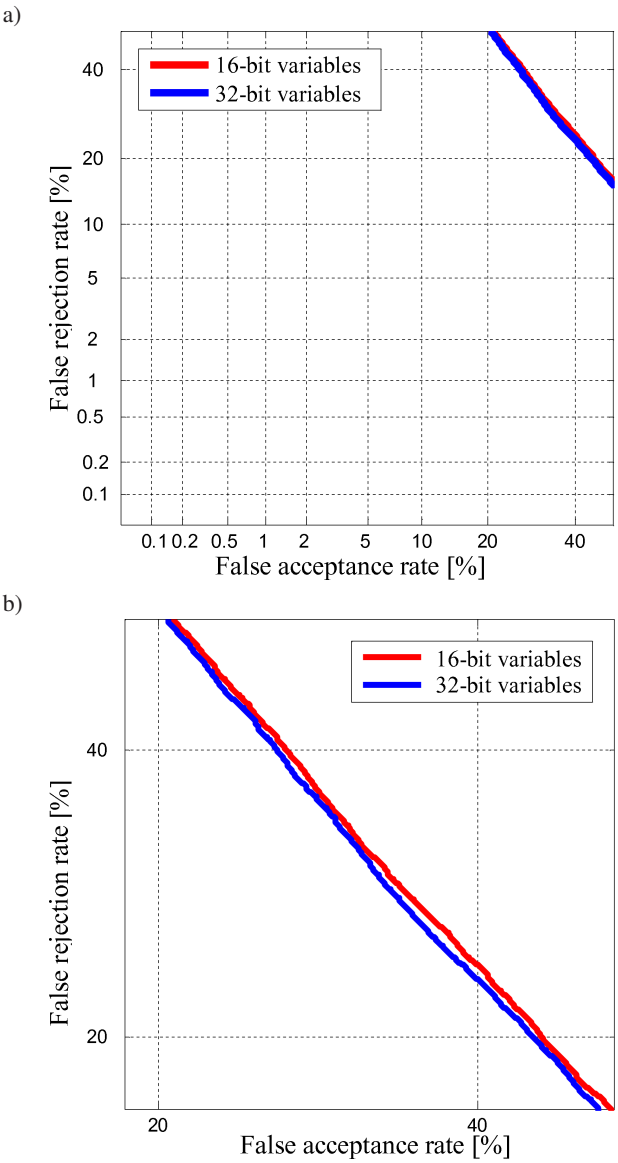

Fig. 6. FAR/FRR plots of speaker identification using Gaussian mixture models GMM (a) and its zoom (b)

Figures 7a–d illustrate the distribution of the results (comparisons of the characteristics between the correct and the in-

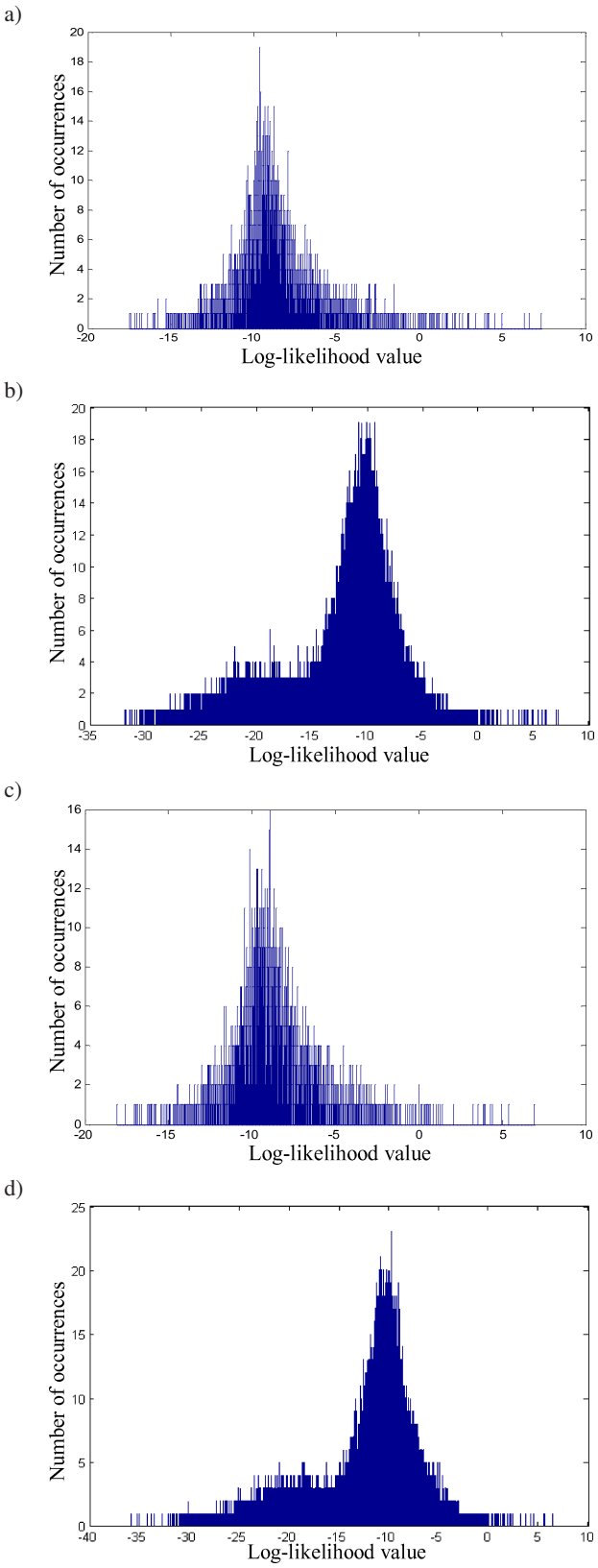

Fig. 7. Log-likelihood distances for GMM algorithm between the tested utterances and a) correct model for 16-bit representation, b) incorrect model for 16-bit representation, c) correct model for 32-bit representation, d) incorrect model for 32-bit representation

correct models for the speaker). In the horizontal axis there is the logarithmic value of the probability of the model and test coming from the same speaker. This value is computed on the basis of the Gaussian mixtures. A change of the representation causes a slight offset of the comparison values and their variance.

The TMS320C5515 eZDSP USB module has an integrated audio encoder (A/D and D/A converters) TLV320AIC3204 operating with the resolution of 16 bits. It should be noted that the TMS320C5515 has already a built-in 10-bit A/D converter. This is an interesting feature, which can make the end product cheaper. Besides, the voice processing with the use of the reduced quantization steps gives only slightly worse results [19], presented in Fig. 8.

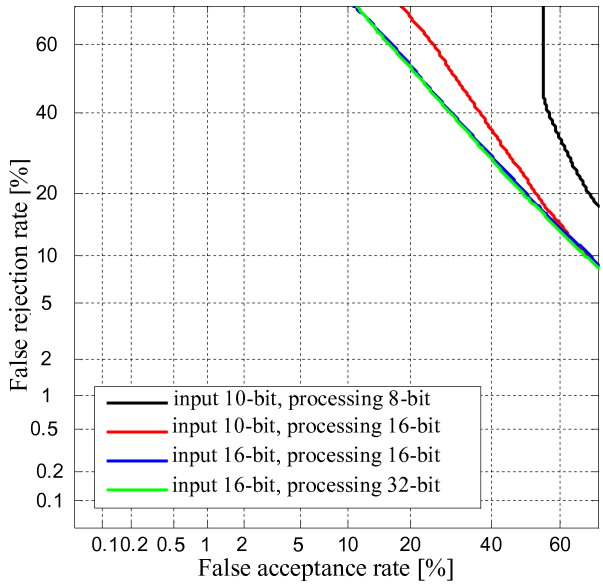

Fig. 8. Results of speaker recognition for selected input signal and processing representations

6. Conclusions

Manufacturers of digital signal processors are trying to facilitate the implementation of the speech signal processing algorithms. Built-in hardware, rapid prototyping methods [20] and most of the all ready-to-use software modules and libraries, give a possibility of shortening the process of designing the real-time signal processing embedded systems. It should be noted that C software generated from the Matlab environment requires a lot of manual intervention with programming and optimization.

Decreasing the data resolution used in the GMM algorithm has no significant impact on the quality of the speaker recognition. It can, therefore, be concluded that the automatic conversion of software from Matlab to the fixed-point 16-bit processor does not significantly affect the performance of the speaker recognition system. Moreover, basing on the results of the initial experiments [19], there exists also a possibility to use internal 10-bit A/D converter only, without any significant deterioration of the speaker recognition accuracy.

Acknowledgements. This work was prepared within the INDECT and DS projects.

REFERENCES

[1] S. Furui, "50 years of progress in speech and speaker recognition", *ECTI Trans. on Computer and Information Technology* 1 (2), 64–74 (2005).

[2] F. Bimbot, J. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-Garcia, D. Petrovska-Delacretaz, and D. Reynolds, "A tutorial on text-independent speaker verification", *EURASIP J. on Applied Signal Processing* 4, 430–451 (2004).

[3] S. Drgas and A. Dąbrowski, "Speaker recognition based on multilevel speech signal analysis on Polish corpus", *Multimedia Tools and Applications, Springer Verlang*, DOI: 10.1007/s11042-013-1502-0 (2013).

[4] Y.S. Moon, C.C. Leung, and K.H. Pun, "Fixed-point GMM-based speaker verification over mobile embedded system", *Proc. 2003 ACM SIGMM Workshop on Biometrics Methods and Applications (WBMA'2003)* 1, 53–57 (2003).

[5] P. Korohoda and A. Dąbrowski, "Generalized convolution as a tool for the multi-dimensional filtering tasks", *Multidimensional Systems and Signal Processing* 19 (3–4), 361–377 (2008).

[6] Jhing-Fa Wang, Jr-Shiang Peng, Jia-Ching Wang, Po-Chuan Lin, and Ta-Wen Kuan, "Hardware/software co-design for fast-trainable speaker identification system based on SMO", *Proc. 2011 IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC)* 2, 1621–1625 (2011).

[7] Zhenling Zhang, Yangli Jia, and Guang Xie, "Design and implementation of speaker recognition system", *Proc. 2011 IEEE 2nd Int. Conf. on Software Engineering and Service Science (ICSESS)* 1, 559–562 (2011).

[8] M. Lizondo, P.D. Agüero, A.J. Uriz, J.C. Tulli, and E.L. Gonzalez, "Embedded speaker verification in low cost microcontroller", *Congreso Argentino de Sistemas Embebidos* 1, 128–133 (2012).

[9] *TMS320C5515 Fixed-Point Digital Signal Processor*, SPRS645E VIII 2010, REV I, Texas Instruments (2012).

[10] *FFT Implementation on the TMS320VC5505, TMS320C5505, and TMS320C5515 DSPs* (Rev. B), Texas Instruments (2013).

[11] M. Siwczyński, A. Drwal, and S. Żaba, "The digital function filters – algorithms and applications", *Bull. Pol. Ac. Tech.* 61 (2), 371–377 (2013).

[12] T. Marciniak, R. Weychan, S. Drgas, A. Dąbrowski, and A. Krzykowska, "Speaker recognition based on short Polish sequences", *Proc. IEEE Signal Processing Conf. (SPA'2010)* 1, 95–98 (2010).

[13] A. Dąbrowski, S. Drgas, and T. Marciniak, "Detection of GSM speech coding for telephone call classification and automatic speaker recognition", *Proc. Int. Conf. on Signals and Electronic Systems (ICSES'2008)* 1, 415–418 (2008).

[14] R. Weychan and T. Marciniak, "Analysis of differences between MFCC after multiple GSM transcodings", *Przeglad Elektrotechniczny* 88 (6), 24–29 (2012).

[15] A. Krzykowska, T. Marciniak, R. Weychan, and A. Dąbrowski, "Influence of GSM coding on speaker recognition using Polish short sequences", *Proc. Joint Conf. New Trends in Audio and Video and IEEE Signal Processing Conf. (NTAV/SPA'2012)* 1, 197–202 (2012).

[16] *TMS320C5515 eZDSP USB Stick Technical Reference*, 512845-0001 Rev A II, Spectrum Digital (2010).

[17] *Matlab Coder Generate C and C++ Code from MATLAB Code*, MathWorks, Inc. (2012).

[18] T. Marciniak, A. Krzykowska, and R. Weychan, "Speaker recognition based on telephone quality short Polish sequences with removed silence", *Przegląd Elektrotechniczny* 88 (6), 42–46 (2012).

[19] R. Weychan, A. Stankiewicz, T. Marciniak, and A. Dąbrowski, "Analysis of the impact of data resolution on the speaker recognition effectiveness in embedded fixed-point systems", *Proc. IEEE Signal Processing Conference (SPA'2013)* 1, 327–331 (2013).

[20] R. Suszyński and K. Wawryn, "Rapid prototyping of algorithmic A/D convertets based on FPAA devices", *Bull. Pol. Ac. Tech.* 61 (3), 691–696 (2013).