Low-cost Command-recognition Device

Martin HYBEN, Michal HODOŇ

Department of Technical Cybernetics, University of Žilina,
Faculty of Management Science and Informatics
Žilina, Slovak Republic
martin.hyben@fri.uniza.sk; michal.hodon@fri.uniza.sk

ABSTRACT: Speech recognition of nowadays has become a domain of computers and mobile devices with high computational performance, where the variety of applications is presented. However the problematic of speech recognition is still quite hard computational task for low cost and low performance microcontrollers. In this paper we propose an algorithm of command recognition system designed on FPGA that is ready for mass production together with low-cost microcontroller-based devices. These devices could cover a wide variety of applications, such as voice management of intelligent houses, voice-command applications, or simply just voice-controlled toys. The algorithm itself can be used as effective speech compression suitable for transmission to remote access point. Reduction of the amount of communication through communication channel also results in better power management.

KEYWORDS: command recognition, low complexity, FPGA, device development.

1. Introduction

Home automation has been around for years, however recently a new trend has become very modern. Smart homes are drawing attention of many people who want to implement hi-tech technology into their stylish lives. An important aspect of this idea together with the comfort brought along is energetic effectiveness. Our goal here was to ensure a comfort of the owner by allowing him entering commands to smart home system by the most natural way, through his voice. In addition, power effectiveness is ensured by the restriction of power demands to the minimal rate.

In this paper we will introduce a low cost device equipped by microphone for recording of the speech, which is capable of detecting and recognizing of commands on site. By using the compression of human speech into text or index of command known by the system, as well as by making use of a very little computational performance thanks to optimizations in recognition process, we achieved power efficiency and very low requirements on communication

channel. In the following chapter we will describe how we have achieved an optimization in speech recognition process. In the final chapter we will evaluate algorithm performance and determine a power effectiveness of the device compared to the standard recognition methods.

2. Algorithm design

The aim here is to design a device capable to recognize a limited number of commands, which are afterwards sent through transmitter to the remote access point. The main problem here is connected with computational cost and power requirements of such a device. A common device capable of mentioned task would have to first record a spoken command, then perform a Fourier analysis to transform signal into frequency domain, determine the parameters of speech e.g. by extracting the Mel-frequency cepstrum coefficients, [1], [2], and recognize the features using some classification algorithm, such Hidden Markov models, [3], [4], Dynamic time warping, [5], [6], or Neural networks [7]. Our aim was to focus on most computational-demanding parts of recognition process and provide optimization by reducing computational demands of this process. Therefore we started with frequency analysis, since this part of the process is most power-demanding. Then we focused on reducing of the number of speech parameters extracted from obtained frequency spectrum. Finally we have chosen the most suitable classification algorithm to provide recognition and then we measured algorithm's performance.

2.1. Frequency analysis on FPGA

Frequency analysis is without doubt the most computational-demanding part of the speech recognition process. By this part of the process optimizing, the whole recognition can be simplified and its performance can be increased.

Our approach, in contrast to standard methods, is to use for the recognition purposes an FPGA module instead of microcontroller itself. An analysis is then based on two steps. First the discrete Fourier transform (DFT) (1) is performed on initial microsegment with length of 128 samples, which is 16ms for sample rate of 8 kHz, using 16 bit sampling.

$$\boldsymbol{X(k)} = \sum_{\boldsymbol{n=0}}^{\boldsymbol{N-1}} \boldsymbol{x[n]} e^{-j\frac{2\pi k}{N}n} \tag{1}$$

The block schema for implementation of DFT on FPGA is depicted on Fig. 1 [8]. As we can see, a sample $x(n)$ from the input is multiplied by sine and cosines value from look up table (*LUT*) and after a summation, a real and imaginary parts of coefficient are stored in FIFO registers.

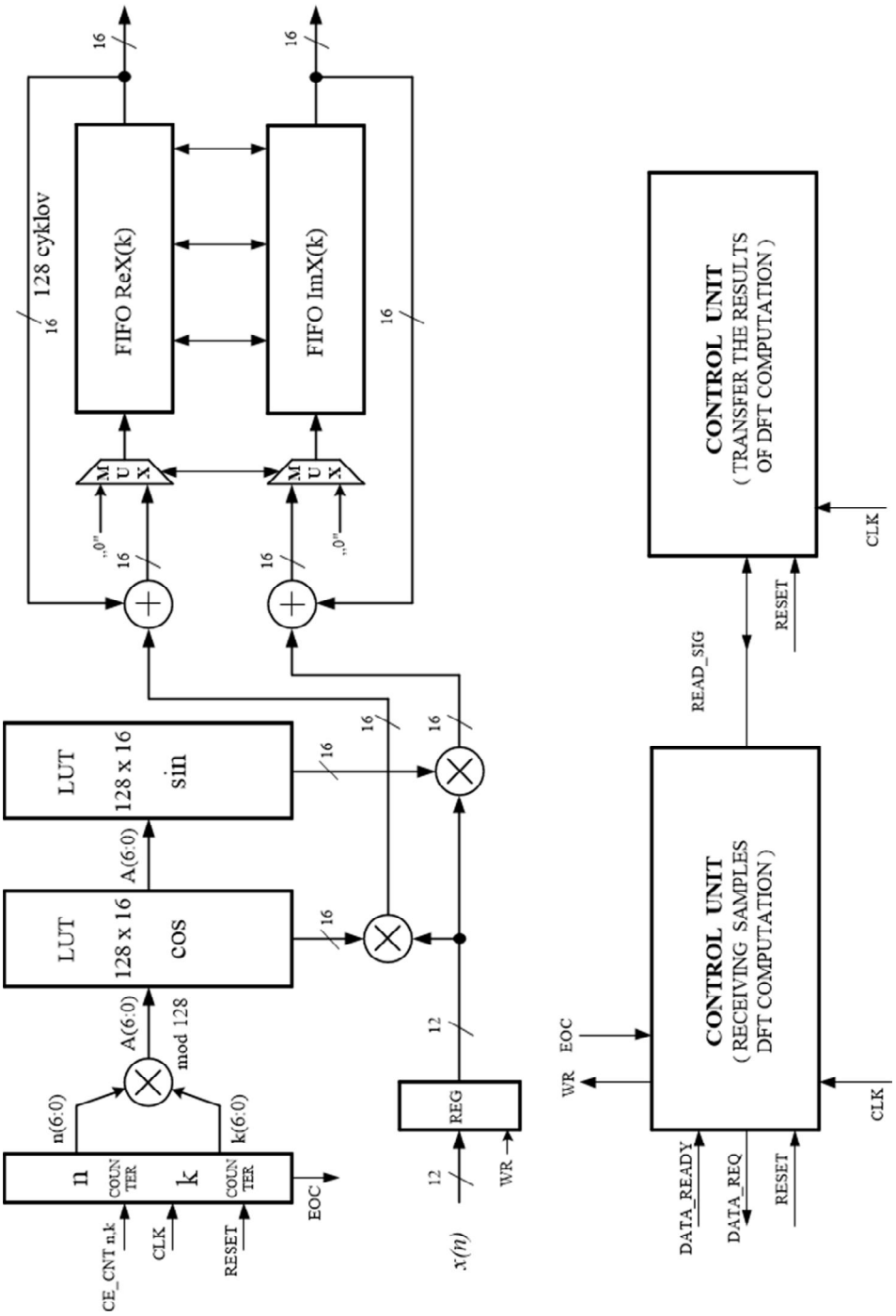

Fig. 1. Discrete Fourier transform on FPGA

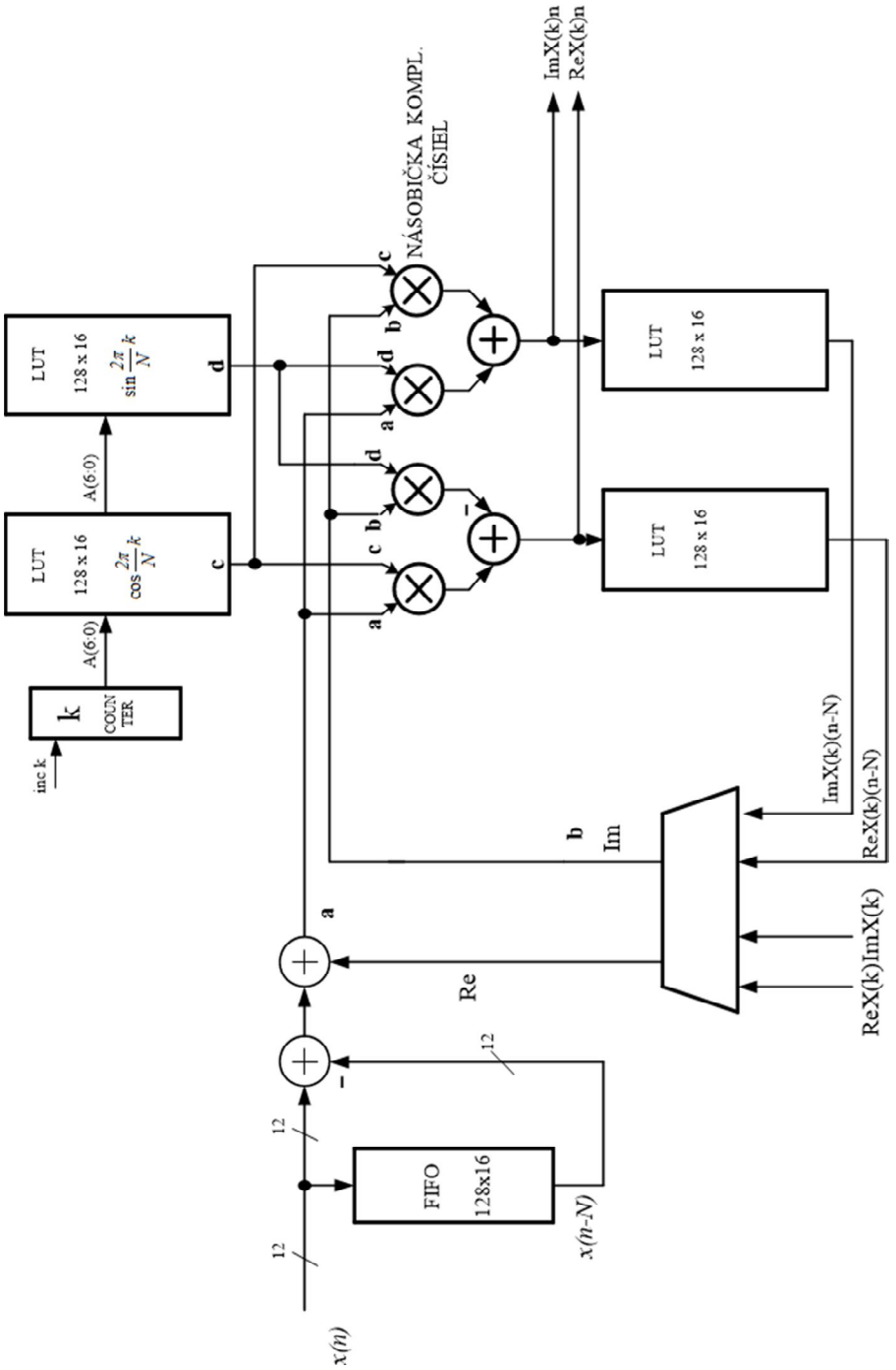

Fig. 2. Sliding discrete Fourier transform on FPGA

After initial analysis, the second step is to perform a sliding Fourier transformation (2) on the rest of the signal, calculating microsegments with length of 128 samples similar to previous step.

$$\boldsymbol{X_k(n) = X_k(n) - [x(n-N) + x(n)]e^{\frac{j2\pi k}{N}}} \tag{2}$$

In this step, as Fig. 2 shows, analysis is performed after every new sample $x(n)$ arrives. Starting with FIFO register filled with a coefficients from DFT part, a new sample $x(n)$ is subtracted by last coefficient from register. Then it is multiplied by values of *LUT* and after summation a new frequency coefficient is calculated, separated into real and imaginary part.

Frequency analysis calculated by FPGA can be used by microcontroller for further processing, when it is optimizing a calculation time to few ms, as the calculation is performed in parallel.

2.2. Optimization using spectral segmentation

An output of frequency analysis performed in previous step is a vector of particular frequency energy values derived from the given spectrum for every time step of Fourier transformation (3).

$$[\mathbf{1} \quad \mathbf{2} \quad \ldots \quad \boldsymbol{N}] \tag{3}$$

Segmentation is performed using following criterions: first criterion is an energy threshold criterion, which provides silence detection; second we need to determine contrast in frequency spectrum which can be calculated by using of Euclidian distance of adjacent vectors.

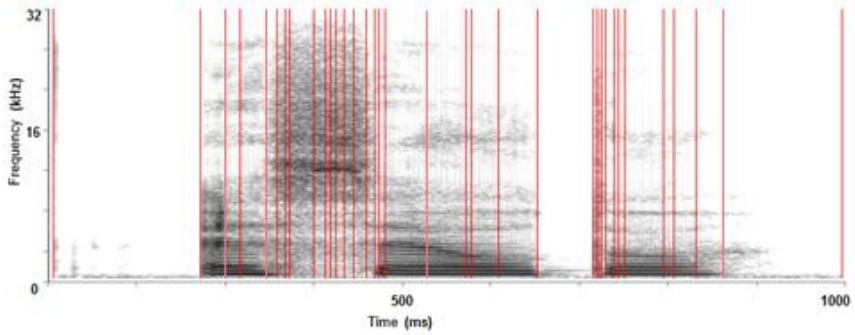

Fig. 3. Spectral segmentation

If contrast value is lower than threshold, vectors are grouped together into cluster. If energy value is higher than silence threshold and contrast crosses

certain threshold, average vector is computed from all vectors of smaller contrast included in cluster. Average vector is then extended by element $N + 1$, which contains information of number of vectors in current cluster (4). The cluster is cleared after every successful fulfill of criterions.

$$[\mathbf{1} \quad \mathbf{2} \quad \ldots \quad \boldsymbol{N} \quad \boldsymbol{N+1}] \tag{4}$$

By adjusting thresholds of contrast and silence detection, we can achieve optimal ratio between segmentation accuracy and number of selected vectors. This results in a reduction of a number of speech parameters in average by 86%.

2.3. Selection of classification algorithm

In this section we will describe the selection of suitable classification method for recognizer to be able to train considerable number of commands and recognize them with a maximum precision. We have considered three possible candidates:

- **Dynamic Time Warping** – Dynamic time warping (DTW) is the oldest classification method based on the time series analysis. It is based on the measuring a similarity between two sequences which may vary in time or speed and estimating which is the most suitable command from the set of prototype patterns. Its computational complexity is $\boldsymbol{O}(N^2)$, where N is a dimension of compared patterns, however the complexity is rising depending on the dimension of pattern set. By minimization of number or dimensions of patterns in set, a very good ratio performance/precision can be achieved.
- **Hidden Markov Models** – Another option was to use Hidden Markov Models (HMM) in order to use statistically based recognizer. In this case the performance of the algorithm also depends on the number of known commands, which correlates with number of models to be used with complexity of $\boldsymbol{O}$ $(N^2 T)$, where N is a number of state for each model and T is the length of sequence.
- **Deep Neural Networks** – Last option was to use deep neural networks (DNN) as this is a new approach in the field of speech recognition. It is well known that DNN increase the precision of recognition, what makes it very attractive method to be used. However the number of neurons in deep neural net is considerable and thus very complex to compute. The complexity of DNN is given as

$$\boldsymbol{O}\left(\boldsymbol{N^2 max_{i,j}}|\boldsymbol{w_{ij}}|\right), \tag{5}$$

where w_{ij} is a weight of i-th neuron in j-th layer.

Therefore, we considered given complexity as too high compared to the rest of methods. Also the precision is not as crucial in our case, since there is always an option to repeat spoken command if previous attempt was not correctly recognized. Therefore we decided not to test DNN as classification method.

3. Summary

In a matter of fact that only DTW and HMM could be selected as the classification methods in case of an computational-efficient speech recognition, where both algorithms have comparable computational complexity, we have decided to perform a test of recognition precision to determine the most suitable method. We have created a database of 20 Slovak spoken commands of numbers 0-9 as well as the common colors. Every command has been recorded in 35 iterations. We have chosen a training set containing 10 iterations of every command and trained the recognition system. The process itself consisted of calculation of 12 Mel-frequency cepstral coefficients, quantized by K-Means clustering algorithm that produced a sequence of centroid indexes. These sequences were then classified by DTW or HMM. The recognition has been performed on testing set containing the rest of iterations with and without usage of proposed spectral segmentation. The results of the tests are summarized in the following table.

Tab. 1. Spectral segmentation

Classification method	Without segmentation	With segmentation
DTW	91.50%	94.00%
HMM	81.76%	88.24%

As we can see from the summary in the table, DTW has proven to be more successful in both precision and computation complexity. Therefore we have selected DTW to be used as classification method.

Algorithm performance

Total algorithm performance is given by complexity of all its steps. A complexity of DFT is $O(N^2)$ and complexity of sliding DFT is $O(\mathrm{N}/2\ \log_2 \mathrm{N})$. By reducing its complexity to $O(N)$ on FPGA, we have achieved better computational complexity than using FFT, which is $O(N \log N)$.

Another optimization is to use spectral segmentation after the frequency analysis is performed. The computational complexity with use of our algorithm is $O(M^2)$, where M is number of input vectors after the segmentation computed with complexity of $O(N)$. Therefore an overall complexity of algorithm would be $O(N)+ O(M^2)$.

Final optimization is command transmission power efficiency. After recognition, a commands compressed into very small amount of information which is to be sent through communication channel to control unit. By using an algorithm proposed in [9] or [10], we could implement an efficient transmission channel together with conventional antennas as mentioned there.

4. Conclusion

In this paper we have introduced a necessary theoretical background required when designing a device capable of recording and recognizing commands optimized for the low-computational complexity. After recognition of the command, compressed information is sent through communication channel to the control unit which is characterized by minimal power requirements. Since the system is less demanding in both transmission and computational domain, power effectiveness has been achieved. In the future, we are planning to test a device in a real world environment and measure the real power effectiveness together with the precision of single commands recognition.

5. Acknowledgement

This contribution/publication is the result of the project implementation: *Centre of excellence for systems and services of intelligent transport II.,* ITMS26220120050 supported by the Research & Development Operational Program funded by the ERDF.

References

[1] ITTICHAICHAREON CH., SUKRSI S. AND YINGTHAWORNSUK T., *Speech Recognition using MFCC*, International Conference on Computer Graphics, Simulation and Modeling (ICGSM'2012), July 28-29, 2012, Pattaya, Thailand.

[2] XU H., ZHANG X. AND JIA, L., *The extraction and simulation of Mel frequency cepstrum speech parameters*, International Conference on Systems and Informatics (ICSAI) 2012, 19-20 May 2012,Yantai, China, page(s): 1765-1768, print ISBN: 978-1-4673-0198-5.

[3] WOMACK B.D. AND HANSEN J.H.L., *N-channel hidden Markov models for combined stressed speech classification and recognition*, IEEE Transactions on Speech and Audio Processing, volume:7, issue: 6, 06 August 2002, page(s): 668-677, ISSN:1063-6676.

[4] BORZESHI E.Z., PEREZ CONCHA O., XU R.Y.D. AND PICCARDI M., *Joint Action Segmentation and Classification by an Extended Hidden Markov Model*, IEEE Signal Processing Letters, volume: 20, issue: 12, October 2013, page(s): 1207-1210, ISSN: 1070-9908.

[5] CHE YONG Y., AL-HADDAD S.A.R. ANG CHEE KYUN N., *Dog voice identification (ID) for detection system*, Second International Conference on Digital Information Processing and Communications (ICDIPC) 2012, 10-12 July 2012, Klaipeda City, Lithuania, page(s): 120-123, print ISBN:978-1-4673-1106-9.

[6] PRIYADARSHANI P.G.N., DIAS N.G.J. AND PUNCHIHEWA A., *Dynamic Time Warping based speech recognition for isolated Sinhala words*, IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS), 2012, 5-8 Aug. 2012, Boise, Idaho, USA, page(s): 892-895, ISSN: 1548-3746.

[7] LU B., JING-JING W., YU W. AND JIN-PING L., *A speech recognition system based on multiple neural networks*, Sixth International Conference on Natural Computation (ICNC) 2010, 10-12 Aug. 2010,Yantai, Shandong, page(s): 48-51, print ISBN: 978-1-4244-5958-2.

[8] CHU P.P., *FPGA Prototyping by VHDL Examples: Xilinx Spartan-3 Version*, Wiley-interscience, ISBN 978-0-470-18531-5, March 2008.

[9] KOCHLÁŇ M., MIČEK J., *Energy-Efficient Communication Systems of Wireless Sensor Networks*, „Studia Informatica Universalis Journal" (in press). Expected publication date is Dec. 2013.

[10] CHOVANEC M., HODOŇ M., *Low power WSN network*, „Teleinformatics Review" (in press). Expected publication date is Dec. 2013.

Urządzenie niskonakładowego rozpoznawania komend

STRESZCZENIE: W dzisiejszych czasach rozpoznawanie mowy stało się domeną szeroko stosowanych komputerów i urządzeń mobilnych o dużej wydajności obliczeniowej. Problemem z rozpoznawaniem mowy jest jednak nadal dość trudne zadanie obliczeniowe dla tanich mikrokontrolerów o niskiej wydajności. W niniejszym artykule proponujemy algorytm systemu rozpoznawania komend zaprojektowany dla FPGA, który jest gotowy do masowej produkcji wraz z urządzeniami na bazie taniego mikrokontrolera. Urządzenia te mogą obejmować szeroki zakres zastosowań takich jak zarządzanie głosem aplikacjami w inteligentnym domu lub po prostu zabawki sterowane głosem. Sam algorytm może być stosowany jako skuteczny do kompresji mowy przy transmisji informacji do punktu dostępowego. Zmniejszenie liczby transmisji poprzez kanał komunikacyjny skutkuje również lepszym zarządzaniem energią.

SŁOWA KLUCZOWE: rozpoznawanie komend, mała złożoność, FPGA, wytwarzanie urządzeń.

Praca wpłynęła do redakcji: 4.12.2013 r.