High Performance Processing for Speech Recognition

Milan Ramljak, Maja Stella, Matko Šarić

Abstract— The evolution of computer technology, including operating systems and applications, resulted in designing intelligent machines that can recognize the spoken word and find out its meaning. During the years, processing time required for speech recognition has been significantly improved, not only thanks to improvements in algorithms, but also with more processing power of nowadays computers. In this paper we analyze processing time and reconstructed speech quality of the three common front-end methods (Linear Predictive Coding - LPC, Mel-Frequency Cepstrum - MFC, Perceptual Linear Prediction - PLP) for calculating coefficients. Reconstructed speech quality is measured with Perceptual Evaluation of Speech Quality (PESQ) score. It is visible from our analysis that, if required, higher number of coefficients could be used without significant impact on processing time for MFC and PLP coefficients. Another very important aspect for processing time is a choice of back-end. In this paper we propose high performance neural network back-end implementation on distributed system based on Erlang programming language. Erlang processes can act as neural network neurons, and asynchronous message exchange is connection within processes transforming Erlang program in a normal neural network structure. With this kind of neural network implementation we have obtained significant increase in performance.

Keywords— speech recognition, coefficients, PESQ, processing time, neural network, Erlang.

I. Introduction

SPEECH recognition technology is constantly advancing and is becoming more present in everyday life, e.g. applications for mobile phones, cars, information systems, smart homes [1-5].

One of the most important components of the speech recognition system is the front-end. The general point for speech is that the sounds generated by a human are filtered by the shape of the vocal tract. This shape determines what sound is created. If we define the shape accurately, this should give us an accurate representation of the phoneme being produced. The shape of the vocal tract manifests itself in the envelope of the short time power spectrum, which is the basis for speech signal recognition. Besides the spectral envelope representation of speech signals, in practice there is a wide range of other possibilities like energy, level crossing rates, and zero crossing rates.

Many recognition systems use different models for front-end processing, like linear predictive coding (LPC) [6], Mel Frequency Cepstral Coefficients (MFCC) [7], Perceptual Linear Prediction (PLP) [8] and other. Besides speech recognition, predictive coding technologies are also an important factor in image processing systems. LPC provides a good approximation to the vocal tract spectral envelope creating simple relation between speech sound and spectral characteristics. The model is mathematically very precisely defined opening good perspective for implementation in hardware and software. This is especially important in case of real time signal processing, because of decrease of required processing and computational operations. The MFC model is successor of LPC and is the most widely used front-end in automated speech recognition. It employs auditory features like variable bandwidth filter bank and magnitude compression. Perceptual linear prediction, similar to LPC analysis, is based on the short-term spectrum of speech. In contrast to pure linear predictive analysis of speech, perceptual linear prediction (PLP) modifies the short-term spectrum of the speech by several psychoacoustical transformations in order to model a human auditory system more closely [9].

Another important aspect in speech recognition system design is a choice of back-end and machine learning algorithm. Besides hidden Markov Models (HMMs), neural networks have been widely used [10-13]. Artificial neural networks (ANNs) consist of simple processing units called neurons and aim to mimic the operation of human neural system. Typical topologies used are single/multilayer perceptron, Hopfield or recurrent networks and Kohonen or self-organizing networks. They are often trained by back-propagation learning, which is a slow methodical procedure for estimating the coefficients of a system. Since biological neural networks incorporate feed-back, (i.e., they are recurrent), it is natural that certain artificial networks will also incorporate that feature. The Hopfield neural networks do indeed employ both feed-forward and feedback. Once feedback is employed, stability cannot be guaranteed in the general case. Consequently, the Hopfield network design must be one that accounts for stability in its settings.

This work was supported by the Ministry of Science and Technology of the Republic of Croatia (project 023-0231924-1660 "Advanced Heterogeneous Network Technologies" and project 023-0231924-1661 "ICT Systems and Services Based on Integration of Information").

M. Ramljak is with the Ericsson Nikola Tesla, Poljička 39, HR-21000 Split, Croatia (e-mail: milan.ramljak@ericsson.com).

M. Stella is with the University of Split, FESB, R. Boškovića 32, HR-21000 Split, Croatia (e-mail: mstella@fesb.hr).

M. Šarić is with the University of Split, FESB, R. Boškovića 32, HR-21000 Split, Croatia (e-mail: msaric@fesb.hr).

Artificial neural networks have been widely researched in speech recognition. Two decades ago, researchers achieved some success using artificial neural networks with a single layer of nonlinear hidden units to predict Hidden Markov Model (HMM) states from windows of acoustic coefficients. At that time, however, neither the hardware nor the learning algorithms were adequate for training neural networks with many hidden layers on large amounts of data, and the performance benefits of using neural networks with a single hidden layer were not sufficiently large to seriously challenge Gaussian Markov Models (GMMs) [13]. Over the last few years, advances in both machine learning algorithms and computer hardware have led to more efficient methods for training Deep Neural Networks (DNNs) that contain many layers of nonlinear hidden units and a very large output layer. The large output layer is required to accommodate the large number of HMM states that arise when each phone is modeled by a number of different "triphone" HMMs that take into account the phones on either side. Even when many of the states of these triphone HMMs are tied together, there can be thousands of tied states. Using the new learning methods, several different research groups have shown that DNNs can outperform GMMs at acoustic modeling for speech recognition on a variety of data sets including large data sets with large vocabularies [13].

In this paper we analyze processing time and reconstructed speech quality of the three common front-end methods (LPC, MFC, PLP) for calculating coefficients. Reconstructed speech quality is measured with Perceptual Evaluation of Speech Quality (PESQ) score [14]. Although modern computers are continuously becoming faster, this computational time is still not negligible, as our results present, and the choice of the coefficients, besides in terms of recognition performance is also essential in terms of computational complexity, especially given the large number of users for server-oriented speech recognition (as used today by major companies in its products – Google Now, Apple Siri, Nuance Communications Dragon Naturally speaking server). We also propose high performance neural network back-end implementation on distributed system based on Erlang programming language [15]. Erlang processes can act as neural network neurons, and asynchronous message exchange is connection within processes transforming Erlang program in normal neural network structure.

The paper is organized as follows. In Section 2, common speech recognition front-ends are shortly described. In Section 3, we present their characteristics in terms of computational time and reconstructed speech quality. In Section 4, neural network implementation in Erlang is given. The paper is concluded with Section 5.

II. Common Front-Ends

Speech recognition front-ends are designed to transform speech signal into the appropriate characteristics that serve as input parameters for the recognition algorithm. These characteristics should emphasize the essential characteristics of speech signal, which are relatively independent of the speaker and channel conditions.

A. LPC

The basic idea behind the LPC model is that a speech sample at time *n*, *s(n)*, can be approximated as a linear combination of the past *p* speech samples:

$$s(n) \approx a_1 s(n-1) + a_2 s(n-2) + \ldots + a_p s(n-p) \tag{1}$$

where the coefficients $a_1, a_2, \ldots a_p$ are assumed constant in analysis frame. This assumption is good for most speech signals with low resonance. LPC analysis produces *N* complex poles, where *N* is the order of predictor. This would imply that specific order of predictor should be defined according to specific application of this signal analysis method.

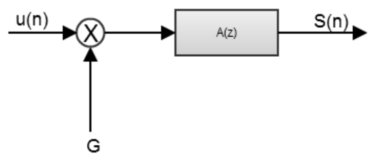

Fig. 1. Linear predictive coding diagram – adapted from [16]

$$H(z) = \frac{S(z)}{GU(z)} = \frac{1}{1 - \sum_{i=1}^{p} a_i z^{-i}} = \frac{1}{A(z)} \tag{2}$$

Fig. 1. represents the LPC model, defined with equation 2. Many formulations in definition of appropriate *p*-th order LPC model depend on mathematical convenience, statistical properties and stability of system.

Mathematical convenience is most important for LPC implementation in software and hardware automated recognition systems. This is narrowly connected with required mathematical operations for LPC calculations. Whole process should be parameterized and calibrated as much as possible to perform effective speech analysis. LPC with higher order can describe speech signal much better, but requires additional mathematical operations.

B. MFC

Mel Frequency Cepstral Coefficients (MFCCs) are a feature widely used in automatic speech and speaker recognition. The difference between the cepstrum and the mel-frequency cepstrum is that in the MFC, the frequency bands are equally spaced on the mel scale, which approximates the human auditory system's response more closely than the linearly-spaced frequency bands used in the normal cepstrum. MFCC values are not very robust in the presence of additive noise, and so it is common to normalize their values in speech recognition systems to lessen the influence of noise.

MFCCs are commonly derived in following steps: [17]

1. Take the Fourier transform of (a windowed excerpt of) a signal.
2. Map the powers of the spectrum obtained above onto the mel scale, using triangular overlapping windows.
3. Take the logs of the powers at each of the mel frequencies.
4. Take the discrete cosine transform of the list of mel log powers, as if it were a signal.

The MFCCs are the amplitudes of the resulting spectrum.

There can be variations on this process, for example, differences in the shape or spacing of the windows used to map the scale [18]. MFC coefficients are very popular and are the most widely used coefficients today, in the early 2000s ETSI defined a standardized MFCC algorithm to be used in mobile phones [19].

C. PLP

PLP coefficients are designed to model a human auditory system mode closely than previous methods. The calculation is similar to MFC coefficients, but they use critical bands, equal loudness curve and intensity-loudness power law. Calculation is performed in the following steps [20]:

- computation of short-term speech spectrum
- nonlinear frequency transformation and critical-band spectral resolution
- critical bands adjustments to the curves of equal loudness
- weighted spectral summation of power spectrum samples
- enforcing the intensity-loudness power-law
- all-pole spectrum approximation
- transformation of the PLP-coefficients to the PLP-cepstral representation.

III. Front-end Processing Characteristics and Results

The experiments were performed with the signal processing tool written in Matlab. The calculation of processing time required to obtain specific number of coefficients for these three methods is measured in Matlab with core i5 processor and 512 MB of RAM.

First, LPC spectrogram example is shown for the same input signal and different LPC orders. Fig. 2 shows three spectral diagrams of input speech signal. First part visualizes original spectrogram of the spoken test sequence. Graph can be easily used to identify spoken words phonetically but for automatic speech recognition purposes some user specific voice characteristics should be decreased. To accomplish this, the order of LPC can be reduced. We experimented with few different orders of LPC, measuring required computation time.

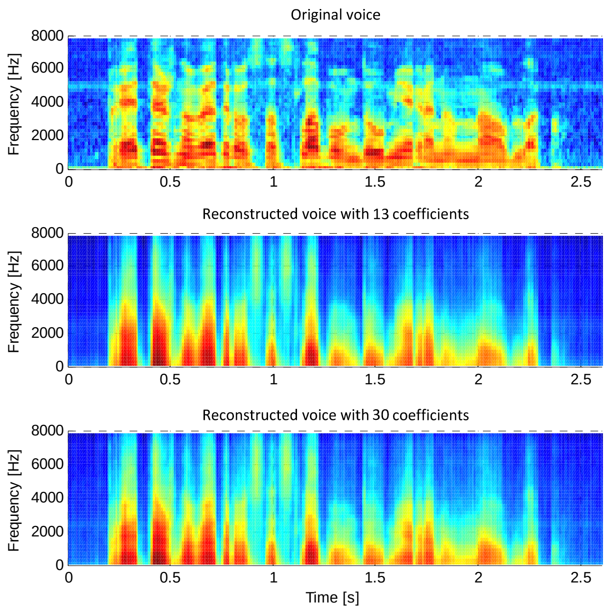

Fig. 2. Spectrograms for different LPC orders

By increasing the order of LPC, more information from spectral diagrams can be obtained, but on the other hand it requires more processing time. However, this does not necessarily mean better recognition performance and 13 coefficients are most commonly used in speech recognition.

Table 1. LPC processing time and reconstructed speech quality

LPC order	***Processing time***	***PESQ MOS***
5th	1.336002 s	1.212
13th	1.535249 s	2.347
40th	3.336561 s	2.451

Table 1. shows real processing times and PESQ scores measured for three different orders of LPC. For input speech signal with duration of 2.5 s, and using 13th order of LPC measured time is 1.535 s which implies that for each second of input signal, approximately 600ms of processing time is required. It should also be noted that this is only one part of speech recognition system, additional processing time is also required for back-end processing.

Fig. 3. shows similar plots for different number of MFC coefficients using the same test sentence.

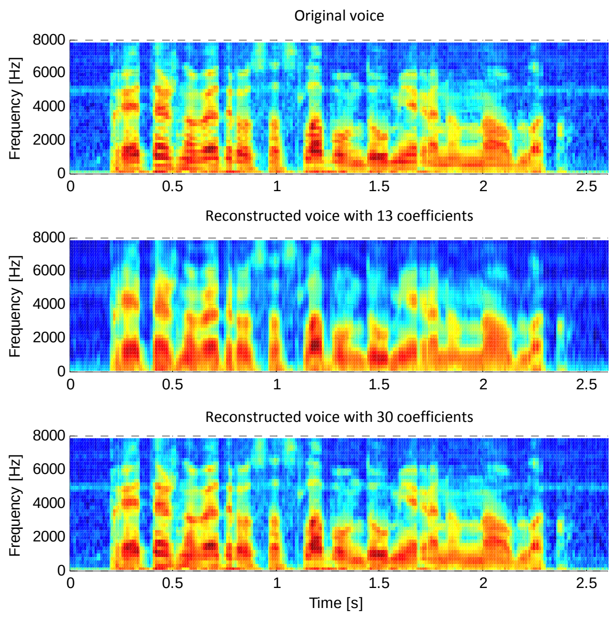

Fig. 3. Spectrograms for different MFC orders

Increasing the number of coefficients used in MFC spectrogram analysis, more precise representation of spoken words is obtained.

Table 2 – MFC processing time and reconstructed speech quality

MFC order	*Processing time*	*PESQ MOS*
5th	0.458499 s	2.342
13th	0.495099 s	2.344
30th	0.536869 s	2.412

According to Table 2, it is easy to see that the processing time required for MFC coefficients calculation is not so different in relation to its number. This is one of the desirable features of this speech analysis method.

For the case of PLP analysis, spectrograms of the same test sentence are given in Fig. 4. and processing time and PESQ scores are given in Table 3. Results are similar to MFC with slightly higher processing times and PESQ scores.

Table 3 – PLP processing time and reconstructed speech quality

PLP order	*Processing time*	*PESQ MOS*
5th	0.522713 s	2.102
13th	0.617283 s	2.434
20th	0.632076 s	2.503

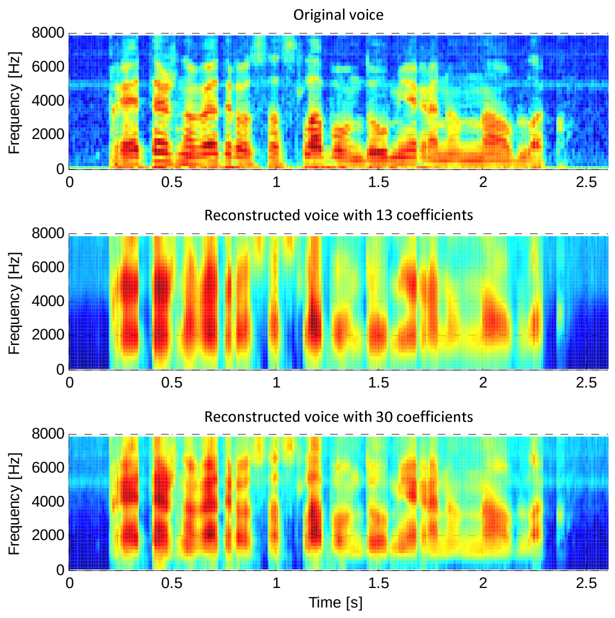

Fig. 4. Spectrograms for different orders of PLP

Comparing all the results it can be easily seen that processing time for calculating the coefficients is varying according to order of the model, but the influence of the order is much lower for MFC and PLP coefficients. The intelligibility of reconstructed speech, calculated with perceptual evaluation of speech signal (PESQ), is increasing with the order of the model, but the processing time is higher, especially for LPC coefficients. This ratio is always important, especially if the speech recognition system has limited processing capabilities. Another interesting thing that we can find out from measurement tables is that in more complex front-ends such as PLP where some knowledge from psychoacoustics is applied to make it closer to human speech recognition, we can obtain better reconstructed speech quality values.

IV. High Performance Neural Networks in Erlang

Neural networks take a different approach to problem solving than that of conventional computers. Conventional computers use an algorithmic approach i.e. the computer follows a set of instructions in order to solve a problem. Unless the specific steps that the computer needs to follow are known, the computer cannot solve the problem.

Neural networks process information in a similar way the human brain does. Neural networks learn by example. They cannot be programmed to perform a specific task. The examples must be selected carefully, otherwise useful time is wasted or even worse the network might be functioning incorrectly. The disadvantage is that because the network finds out how to solve the problem by itself, its operation can be unpredictable.

Neural networks and conventional algorithmic computers are not in competition but complement each other. There are tasks more suited to an algorithmic approach like arithmetic operations and tasks that are more suited to neural networks. Even more, a large number of tasks require systems that use a combination of the two approaches (normally a conventional computer is used to supervise the neural network) in order to perform at maximum efficiency.

Neural networks have found widespread application in various fields of engineering. The performance is crucial when neural network is applied to large set of inputs. Next part of this paper is focused on the performance analysis of neural network classifiers on distributed microprocessors. It is hard to find information of neural network performances on general purpose computers. Designers of neural networks use to ask "Is it possible to predict/estimate the potential processor performance of some neural network application using Standard Performance Evaluation Corporation characteristics". If the execution characteristics are similar, then one can conclude that there is no reason to distinguish neural network classifiers applications performance [21].

To start the analysis, we will use three different types of neural networks

Table 4 Three types of Neural Networks

Type	*Description*	*I/O number*
RNN	Recurrent Neural Network	4/3
SONN	Self-organizing Neural Network	4/3
GRN N	General Regression Neural Network	4/3

List of neural network types defined in Table 4 was selected due to their numerous applications. The general regression neural network (GRNN) [22] is a feed-forward architecture divided in four layers; the input layer, the pattern layer, the summation and division layer and the output layer. A GRNN is capable to approximate a continuous function with great degree of accuracy and that is the reason to use it for continuous function approximation. Kohonen or a self-organizing neural network (SONN) is providing a codebook of stable patterns in the input space that characterizes an arbitrary input vector, by a small number of representative clusters [23]. The last one, Hopfield or recurrent neural network (RNN) is the network in which the input to each computational element includes both inputs as well as outputs. The Hopfield network has a set of stable attractors and repellers. Every input vector is either attracted to one of the fixed points or repelled from another of the fixed points. The strength of this network is the ability to correctly classify "noisy" versions of the patterns [23].

These neural networks types were running on SUSE 11 Linux platform. The programs are all written in Erlang programming language [15]. Erlang is a declarative language for programming concurrent and distributed systems which was developed by the authors at the Ericsson and Ellemtel Computer Science Laboratories. Many of the Erlang primitives provide solutions to problems which are commonly encountered when programming large concurrent real-time systems.

Erlang has a process-based model of concurrency. Concurrency is explicit and the user can precisely control which computations are performed sequentially and which are performed in parallel. Message passing between processes is asynchronous, that is, the sending process continues as soon as a message has been sent.

The only method by which Erlang process can exchange data is message passing, resulting in applications which can easily be distributed. An application written for a uniprocessor can easily be changed to run on a multiprocessor or network of uniprocessors. Except multiprocessing power, continuous operation allows code to be replaced in a running system. This is of great use in "non-stop" system where the systems cannot be halted to make changes in the software.

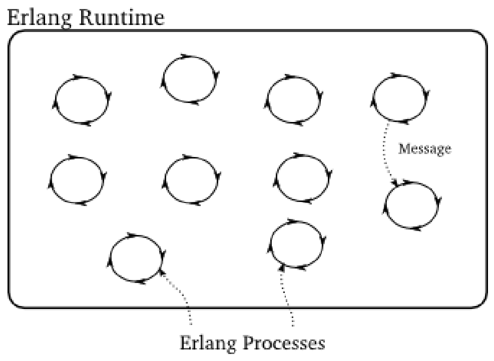

Fig. 2 Erlang program example [24]

Note that Fig. 2 of an Erlang program has each process running "under its own power", rather than being spun around by callbacks. And that is very much the case. With the job of the reactor subsumed into the fabric of the Erlang runtime, the callback no longer has a central role to play [24]. Each process is running in parallel, what is a great technology improvement in comparison with classical operations systems.

Advanced programming efforts were made to tune the software performance of programmed types of neural networks. Fig. 3 shows a model of a recurrent neural network with its structure in Erlang runtime. Erlang processes acts as neurons of neural network, and asynchronous message exchange is connection within processes transforming Erlang program in normal neural network structure.

All programs were trained on the IRIS data set [25] and processed on 500 data elements.

Table 5 shows for each type application two measureable values, its execution size and the number of instructions executed for the input data set of 500 elements. Summing all executed instructions we are closing to 90 thousands. Great number of instruction we have to execute while training neural network leads us to very big performance requirements.

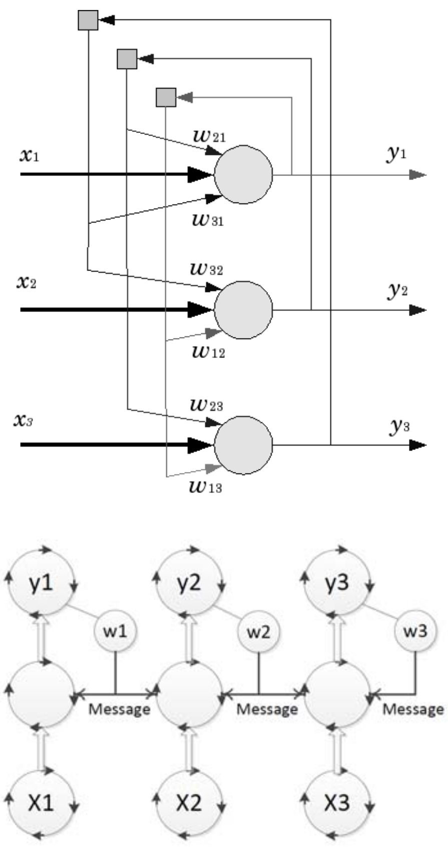

Fig. 3 RNN model (top) and its structure in Erlang runtime (bottom)

This problem could be solved using our proposal of RNN structure in Erlang runtime, as shown in Fig. 3. A new system load can be distributed over processes, since each process (or state in RNN) is parallel and executed in the same time. Not only that we could create new process, but can be used on totally new processor, without need to restructure RNN.

Table 5 Programs execution characteristics

Type	*Executed Instructions*	*Processed Size (KB)*
RNN	14,322,109	14,76
SNN	23,302,431	15,42
GRNN	15,401,396	14.25

System concurrency is good basis to design structure in which one process could also be a neural network. Thinking in this way, top structure of neural network can be relatively simple, but inside it could form a deep neural network (DNN) structure of processes, Fig. 4.

Good thing is that each process can communicate in real time, so network could be "live". Depending on outputs of a process, a new network can reconnect one of outputs to any other input. Interconnection is done by message exchange, in other hand resulting processing time increases as shown in Fig.5.

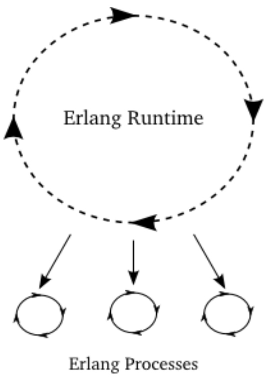

Fig. 4 Full neural network as process

Such neural network will be concurrent neural network (CNN), and will not have solid links among states. No matter what which way is chosen, performance of neural network will be the best possible.

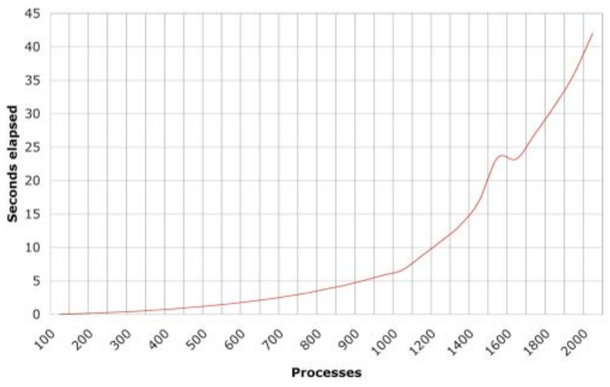

Fig. 5 Erlang message-passing benchmark [26]

This study was driven on distributed system with 6 separated processors. All programs were compiled using Erlang compiler R16B02 for Windows [27], and then deployed on Linux OS. In computer architecture, instructions per clock (IPC) are one aspect of a processor's performance: the average number of instructions executed for each clock cycle. Looking back, RNN structure in Erlang runtime hides very interesting value that can describe performance characteristic in comparison to object oriented ways of programming. In similar way, any neural network could be easily structured as a set of processes.

V. Conclusion

Speech recognition technology is constantly advancing and is becoming more present in everyday life. During the years, it has been significantly improved, not only thanks to improvements in algorithms, but also with more processing power of nowadays computers. In this paper we analyzed processing time and reconstructed speech quality of the three common front-end methods (LPC, MFC, PLP) for calculating the coefficients. As a measure of reconstructed speech quality, PESQ score was used. For PLP coefficients, where some knowledge from psychoacoustics is applied to make it closer to human speech recognition, our results showed that better reconstructed speech quality values can be obtained. Our analysis also showed that, if required, higher number of coefficients could be used without significant impact on processing time for MFC and PLP coefficients.

Although modern computers are continuously becoming faster, this computational time is still not negligible, and the choice of the coefficients, besides in terms of recognition performance is also essential in terms of computational complexity, especially given the large number of users for server-oriented speech recognition (as used today by major companies in its products – Google Now, Apple Siri, Nuance Communications Dragon Naturally speaking server).

This computational time could be improved if the computation is performed in distributed processing system, but this implementation is not simple and we now leave it for the future work. On the other hand, neural network structure by itself is very convenient for implementation in a distributed processing environment and in this paper we propose high performance neural network based speech recognition back-end implementation based on Erlang programming language. Erlang processes can act as neural network neurons, and asynchronous message exchange is connection within processes transforming Erlang program in normal neural network structure. With this kind of neural network implementation we have obtained significant increase in performance. Not only to neural network, Erlang characteristics could be expanded to any other part of a speech recognition system, and high processing performance could be achieved.

REFERENCES

[1] Z.-H. Tan, B. Lindberg, "Speech recognition on mobile devices", *Mobile Multimedia Processing*, 2010, pp. 221–237.

[2] W. Li, K. Takeda, F. Itakura, "Robust in-car speech recognition based on nonlinear multiple regressions", *EURASIP Journal on Advances in Signal Processing*, 2007.

[3] W. Ou, W. Gao, Z. Li, S. Zhang, Q. Wang, "Application of keywords speech recognition in agricultural voice information system", *Computational Intelligence and Natural Computing Proceedings (CINC), Second International Conference on, IEEE*, pp. 197–200, 2010.

[4] I. McLoughlin, H. R. Sharifzadeh, "Speech recognition for smart homes", *Speech Recognition, Technologies and Applications*, pp. 477–494, 2008.

[5] Yamamoto, Shota, Yasunari Yoshitomi, Masayoshi Tabuse, Kou Kushida, and Taro Asada, "Detection of baby voice and its application using speech recognition system and fundamental frequency analysis." *Selected Topics in Applied Computer Science*, pp. 341-345. World Scientific and Engineering Academy and Society (WSEAS), 2010.

[6] F. Itakura, "Minimum prediction residual applied to speech recognition", *IEEE Trans. Acoustics, Speech, Signal Processing*, 1975, vol. 23, no. 1, pp. 67-72.

[7] S. B. Davis and P. Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1980, vol. 28, no. 4, p. 357–366.

[8] H. Hermansky, Perceptual linear prediction (PLP) of speech, Journal of the Acoustic Society of America, 1990, vol. 87, no. 4, pp. 1738-1752.

[9] H. Hermansky, N. Morgan, A. Bayya and P. Kohn, Rasta-PLP Speech Analysis Technique, ICASSP-92, April 1992.

[10] P. Talukdar, M. Sarma, K. K. Sarma, "Recognition of Assamese SpokenWords using a Hybrid Neural Framework and Clustering Aided Apriori Knowledge", *WSEAS TRANSACTIONS ON SYSTEMS*, Issue 7, Volume 12, July 2013.

[11] K. Daqrouq, A. Morfeq, M. Ajour, A. Alkhateeb, "Wavelet LPC With Neural Network for Speaker Identification System", *WSEAS TRANSACTIONS on SIGNAL PROCESSING*, Issue 4, Volume 9, October 2013.

[12] J. Karam, "Radial basis functions with wavelet packets for recognizing Arabic speech." *Recent Researches in Circuits, Systems, Electronics, Control & Signal Processing*, pp. 34-39. World Scientific and Engineering Academy and Society (WSEAS), 2010.

[13] G. Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior et al. "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups." *Signal Processing Magazine, IEEE* 29, no. 6, 82-97., 2012.

[14] ITU-T, Rec. P.862. "Perceptual evaluation of speech quality (PESQ): an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs", 2001.

[15] The Erlang open source webpage. [Online] Available: http://www.erlang.org/

[16] L. Rabiner and B. H. Juang, "Fundamentals of Speech Recognition", Prentice Hall, 2, 42-65. 1993.

[17] Sahidullah, Md, and Goutam Saha, "Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition", *Speech Communication* 54.4, 543-565, 2012.

[18] Zheng, Fang, Guoliang Zhang, Zhanjiang Song, "Comparison of different implementations of MFCC", Journal of Computer Science and Technology, vol. 16, issue 6, pp. 582-589., 2001.

[19] ETSI, "Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms," European Telecommunications Standards Institute Technical standard ES 201 108, v1.1.3., 2003.

[20] J. Psutka, "Comparison of MFCC and PLP Parameterizations in the Speaker Independent Continuous Speech Recognition Task", *Eurospeech*, pp. 1813-1816., 2001.

[21] O. Hammami, "Neural network classifiers execution on superscalar microprocessors." *High Performance Computing*, pp. 41-54. Springer Berlin Heidelberg, 1999.

[22] D.F.Specht, "A General Regression Neural Network", *IEEE Trans. on Neural Networks*, Vol.2, No.6, pp.568-576, 1991.

[23] W.-H. Steeb, "Mathematical Tools in Signal Processing with C++ & Java Simulations", World Scientific Publishing, 2005.

[24] D. Peticolas, "An Introduction To Asynchronous Programming and Twisted," [Online]. Available: http://krondo.com/?p=2692

[25] Iris Dataset, [Online]. http://neural.cs.nthu.edu.tw/jang/books/dcpr/dataSetIris.asp?title=2-2%20Iris%20Dataset

[26] T. Bray, "Testing the T5120," [Online] Available: http://www.tbray.org/ongoing/When/200x/2007/10/09/Niagara-2-T2-T5120

[27] Erlang, [Online]. Available: http://www.erlang.org/download/otp_win64_R16B02.exe