A Review of Serbian Parametric Speech Synthesis Based on Deep Neural Networks

Tijana Delić, Milan Sečujski, and Siniša Suzić

Abstract — **In this paper the research related to the development of a deep neural network based speech synthesizer for the Serbian language, trained on recorded utterances of a single female voice talent, is described. Two separate networks are used for prediction of acoustic features and phonetic segment durations. Through a set of experiments the optimal values of the hyper-parameters of the neural networks are established, and then the influence of the amount of training data on the quality of synthesized speech is examined. The quality is evaluated through objective measures as well as appropriate listening tests. It has been confirmed that 4-layer deep neural networks with 512 units per hidden layer, trained on 3 hours of data, produce speech of very good quality. The results also suggest that a further increase in the amount of training data may contribute to further improvement in quality.**

Keywords — **DNN, HMM, speech synthesis.**

I. Introduction

The development of speech technologies represents a multidisciplinary problem which is highly language dependent. The language dependence of text-to-speech synthesis (TTS) is mainly related to its first task, which is to convert raw input text into a convenient linguistic specification (often referred to as "front end"). On the other hand, the generation of the actual speech signal on the basis of this specification (referred to as "back end") is largely language independent, and mostly done using either concatenative or parametric methods. Concatenative methods are based on concatenation of natural speech segments that exist in a pre-recorded database. Such a system in the Serbian language has been jointly developed by the Faculty of Technical Sciences and the AlfaNum [1] company from Novi Sad, and it is widely used. On the other hand, during the last decade, parametric methods have been rapidly gaining popularity. They are based on the idea to generate speech from parametric models trained on the pre-recorded speech database. Most of their popularity is due to their general flexibility, particularly in terms of changing speaker's characteristics. Such applications have become important with widespread use of speech technologies. The most popular parametric speech generation method was based on Hidden Markov Models (HMM). Such a system has been developed for Serbian as well [2], [3], but the speech quality was impaired by some typical HMM TTS drawbacks [4]. In the last few years this method is being replaced by deep neural network (DNN) approach. The first synthesizer based on DNN for the Serbian language [5] has been developed by using the open-source Merlin toolkit [6] and language-dependent resources and existing front end provided by the AlfaNum company.

The paper is organized in the following manner. The following section discusses different methods for speech synthesis, and in Section III the DNN synthesizer is presented in more details. Section IV presents results from different experiments concerned with the neural network architecture and the training database used for DNN synthesis. Section V presents an objective and subjective comparison of different synthesizers, and Section VI discusses the conclusions and outlines the directions of further research.

Paper received May 7, 2017; revised June 9, 2017; accepted June 10, 2017. Date of publication July 31, 2017. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Branimir Reljin.

This paper is a revised and expanded version of the paper presented at the 24th Telecommunications Forum TELFOR 2016 [5].

The research was conducted within the project "Development of Dialogue Systems for Serbian and Other South Slavic Languages" (TR32035), financed by Ministry of education, science and technological development of Republic of Serbia, EUREKA project DANSPLAT (E!9944), and project "Central audio library of the University of Novi Sad", financed by Provincial secretary for science and technological development of Autonomous province of Vojvodina.

Tijana Delić is with the Faculty of Technical Sciences, University of Novi Sad, Serbia (phone: 381-63-503494; e-mail: tijanadelic@uns.ac.rs).

Milan Sečujski., is with the Faculty of Technical Sciences, University of Novi Sad, Serbia (phone: 381-21-4852533, e-mail: secujski@uns.ac.rs).

Siniša Suzić., is with the Faculty of Technical Sciences, University of Novi Sad, Serbia (phone: 381-21-4852533, e-mail: sinisa.suzic@uns.ac.rs).

II. Parametric methods for speech synthesis

Concatenative methods synthesize speech by directly concatenating the most appropriate segments of speech existing in the database, attempting to smooth out possible discontinuities between them. The segments themselves can be modified in order to better fit the specification given by the front end, but this modification is limited to changes in duration or fundamental frequency, while it is impossible to modify the timbre of the voice [7]. This means not only that high speech quality can be achieved with databases of at least several hours of speech, but it is also impossible to produce new speech styles or new speaker voices without preparing a new training database of the same size. As the preparation of speech databases is known to be an extremely time consuming process, this drawback drastically limits the applicability of concatenative synthesis.

The main idea of parametric approaches is the production of speech from models whose parameter values are obtained by training on an existing speech database. In other words, the model *learns* from the data instead of just reproducing it. Parametric speech synthesis is thus carried out in two phases. In the training phase feature vectors representing natural speech are extracted from the database and used for training models and setting their parameters. In the synthesis phase trained models are used for generation of speech parameters corresponding to new text. Parameter extraction as well as speech generation from parameters generated in the synthesis phase are performed by an appropriate vocoder [8].

The most dominant method in statistical parametric speech synthesis is based on hidden Markov models (HMM) and Gaussian mixture models (GMM) [9]. This method models conditional probability density functions of output speech parameters given linguistic information. Speech units that are modelled are context dependent phonemes. Since the initial number of models can be very large, there is not enough data to successfully estimate the parameters for all of them. This problem is solved using decision-tree-based clustering [10], which merges models in similar contexts and estimates joint parameters for all of them. Such averaging inevitably leads to some over-smoothing of parameters, resulting in somewhat muffled synthesized speech.

Most of the problems introduced by the clustering procedure in HMM-based synthesis can be overcome by using neural networks for prediction of acoustic parameters of speech. A neural network is used to establish a non-linear mapping between input linguistic features and output acoustic features, with the ultimate aim of making generated features sound more similar to the natural ones. By being able to train all models on the entire database instead of using only parts of the database appropriate for a certain model, it avoids data fragmentation and is thus able to generalize better than HMM. Furthermore, DNNs are more robust to high dimensionality than decision tree based clustering in HMM, although in the case of DNN values of parameters (weights in the neural network) are much more difficult to interpret [4].

III. Speech synthesis based on DNN

A. Linguistic text processing

When a speech synthesis system gets text as input, it has to perform some basic linguistic processing of the text in order to get its phonetic and prosodic transcription. Unlike the problem of obtaining phonetic transcription, the problem of obtaining prosodic transcription is particularly challenging for a language such as Serbian, owing to its complex morphology and accentuation system, as well as the dependence between the two. The existing front end for Serbian relies on a morphological dictionary and an expert system which disambiguates between text tokens that can be accented in different ways [11]. The expert system observes each word in its syntactic and semantic context, predicts accent type and position, but also assigns prosodic markers related to phrasing and semantic focus. The last step is the conversion from symbolic representation of prosody to acoustic features and phone durations [12]. While concatenative synthesis requires this last step to be explicitly performed within the front end (in the case of Serbian classification and regression trees (CART) are used), in parametric synthesizers it is left to the speech generation module.

After linguistic text processing context phoneme aligned dependent labels are created. The label corresponding to a phoneme contains starting time stamp and information about its context (phoneme identity, identity of phonemes that surround it, type of accent, number of syllables, etc.). For the purpose of DNN synthesizer, labels have to be aligned on the HMM state level. Because of that, using HTK toolkit [12], HMM models consisting of 5 states per phoneme are trained (as recommended by the Merlin toolkit), and finally starting timestamps for each state of each phoneme are added to labels.

B. Training process and generation of speech signal

In case of speech synthesis, there are two models to be trained. One is the acoustic model for prediction of acoustic features, and the other is duration model for prediction of phone durations. As input for both, acoustic and duration models, features presenting the information about linguistic context provided by the module for linguistic text processing are given. Moreover, the acoustic model is given the information regarding state and phone durations. In the training stage that information is gathered from HMM models through the procedure of forced state alignment, and in the synthesis stage, the output of the duration model is used as supplementary input of the acoustic model.

In the preparation phase, linguistic features are converted into a binary format. Because of the existence of numerical features as well, input labels are normalized to the range from 0 to 1. As the activation function in the input layer, tangent hyperbolic is used. As was previously mentioned, the acoustic model gives acoustic features as an output, from which the vocoder can synthesize speech. In the training stage, acoustic features extracted from audio files from database are used as target features. In all experiments WORLD vocoder [14] is used for extraction procedure. WORLD produces three types of parameters – spectral envelope, fundamental frequency and aperiodicity parameters. Spectral envelope is represented by FFT samples. This representation is not convenient for use as input feature due to high dimensionality. For this reason the envelope is represented using mel-generalized cepstrum analysis. Thus, used acoustic features are: mel-generalized cepstral Coefficients (MGC), logarithm of fundamental frequency (lf0) and band aperiodicity parameters (BAP). Dynamic features (both delta and delta-delta coefficients of all three types of static acoustic features) are also calculated and used in target feature vector. In addition, information whether the currently observed frame is voiced or unvoiced, is added as one binary feature – V/UV. Target features are normalized to zero mean and unit variance. For the output layer, the used activation function is linear.

Random initialization of weights and biases in DNN is used (Gaussian distribution). Usually, it is impossible to put

all the data in RAM memory and because of that, training of DNN is done in mini-batches. In case of a recurrent NN one batch represents one sentence, but in case of a feedforward NN, it can be arbitrarily chosen. On one hand, a small mini-batch gives stochastic results. On the other hand, if it is large it needs more RAM memory and time is required for its processing. The size of the mini-batch represents the number of frames that will be fed to the network before one back propagation training step is conducted. Once all data is seen by the network, one training epoch is finished. For the acoustic model, input and output features are frame aligned, but for the duration model, they are phone aligned. Model training generally represents the optimization of a cost function. Usually the MSE (Mean square error) function is used, which is especially recommended in the case when a linear activation function is used in the output layer. MSE is calculated between the predicted and the target features and it is optimized by back propagation. To prevent overfitting different regularization methods can be used, and the most commonly used one is L2 regularization. The number of training epochs (set to 25 in the paper) has to be determined before starting the training procedure, but the training can also be stopped earlier in case the error calculated on the validation set is increasing for a pre-determined number of epochs (set to 5 in this paper). For learning rate, a starting value is arbitrarily set to 0.004, and it is decreased during the training. In every consecutive epoch it is decreased by 15%, and in case the error on validation set increases, learning rate is decreased by 50%. For the duration model the procedure is similar, but its output features are the durations of each state of a phoneme. The architecture and the choice of parameters for those two models are not dependent, but in this paper the same architecture and the same parameters for both models are used in all cases. Silent phonetic segments are also taken into consideration, with the exception that the first and the last silence from each recording are discarded, because these are not dependent on speaker, but on the person who performed the segmentation of recordings. The silence is considered as an additional phoneme, with its different contexts.

In the synthesis phase, the input is a full context label provided by the front end. The duration model predicts the duration of each state of each phoneme based on the information on the linguistic context provided by the front end. The output of the duration model will be used as a supplementary input for the acoustic model in addition to linguistic features in order to get state-aligned full context labels. Finally, acoustic features are predicted on a frame-by-frame basis, which corresponds to the way the acoustic model has been trained. After being predicted, the acoustic features of individual frames are denormalized by using the mean and the variance saved during the training. Static and dynamic features are connected through a corresponding derivation equation but this constraint is not used during the synthesis procedure which could result in not so smoothed parameter trajectories. This problem is overcome using the maximum likelihood parameter generation procedure [15]. A post filtering procedure, which should emphasize spectral peaks, is also conducted before features are fed to the vocoder, which produces the output speech signal.

IV. Experiments

In this paper a 3h database is used, containing recorded utterances of a single female voice talent. It is proffesionally recorded in studio and it is the property of AlfaNum [1]. All recordings are in a wave format and resampled to 16 kHz. For each experiment, available utterances are divided into three sets: training, validation and test set. The test set was always the same and consisted of 10 utterances. In each experiment, 10% of utterances were randomly chosen to be used for validation, while the rest was used for training. Three sets of experiments, with different choices of hyper-parameters of the network were conducted. Hyper-parameters such as learning rate, regularization, number of training epochs, batch size, etc. were kept constant within each of the three experiment sets. In each experiment set, the architectures for both duration and acoustic models were the same. The experiment investigated the usability of DNNs with a tangent hyperbolic activation function for input and for each hidden layer, while for the output layer, the activation function was linear.

To measure the impact of NN hyper parameters on the

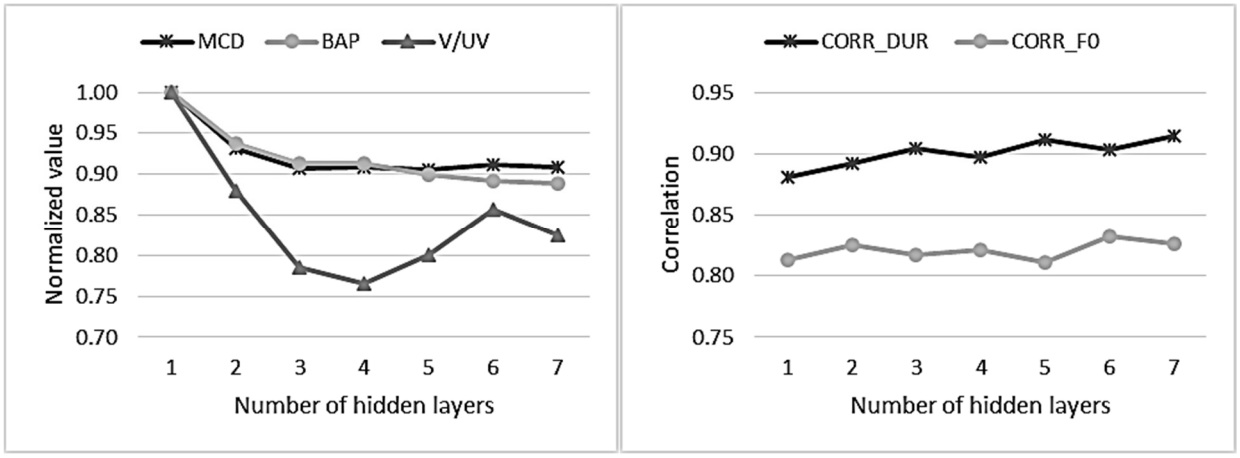

Fig. 1. Influence of the number of hidden layers on the objective measures. Left: Influence on MCD, BAP and V/UV (values of objective measures are normalized). Right: Influence on the correlation of F0 and phone durations.

quality of synthesized speech the following objective measures are used: mean squared error for MGCs (MCD – mel-cepstral distance) and band aperiodicities error (BAP), both given in decibels (dB), correlation between predicted and original fundamental frequency (F0) and durations of phonemes and the error of frame voicing prediction (V/UV). The root mean squared error (RMSE) for F0 and durations, that Merlin toolkit also calculates, were not used since it was considered not to be a good measure of the quality of synthesized speech. For instance, a synthesized utterance can have a slightly higher F0 than the original throughout its duration, and it will still sound almost the same, but the RMSE of F0 will be very large.

A. Influence of the number of hidden layers

For this set of experiments the training and validation set consisted of approximately 1h of speech, including silences. For each experiment the same division into training and validation data was used. The number of hidden layers varied from 1 to 7, but the number of neurons per layer was always 512. It can be seen in Fig. 1, that there is an improvement in objective measures for up to 4 layers, but thereafter the values start to stagnate or even deteriorate. The best MCD is achieved with 5 layers (4.67 dB), but the values for any number of layers between 3 and 7 remain within a relatively small range (4.69±0.02 dB). The best objective measure for BAP is achieved with 7 layers, but as in the case of MGC features, the variations are very small and lie within the range 0.258±0.004 dB when the number of layers varies from 3 to 7. For V/UV, the smallest error is obtained using 4 layers and it is 4.1%, while the error remains within the range 4.32±0.27% when the number of layers varies from 3 to 7. The best value of correlation is achieved with 5 and 6 layers for duration and F0, respectively. As for other measures, increasing the number of layers does not affect the results drastically. Since the complexity of the entire system, as well as the time required for its training, increases with the number of layers, 4 was chosen as the most appropriate number of layers.

B. Influence of the number of units per hidden layer

In this set of experiments the number of hidden layers was fixed to 4, while the number of units per hidden layer was varied from 64 to 1024 in steps doubling in size. The division of the available utterances into the training and validation set was the same as in the previous set of experiments. It has been shown that the objective measures remain almost the same for each experiment (with variations staying in a relatively narrow range, as was the case when changing the number of layers from 3 to 7). However, it has also been shown that, unlike more complex networks, which produce synthetic speech of relatively constant quality, a network with as few as 64 units per layer produces synthetic speech that is clearly inferior, with phone boundaries audibly oversmoothed.

C. Influence of the size of database

In this set of experiments the architecture of the models was fixed to 4 hidden layers with 512 units each. The total size of the training and validation set was varied from 5 minutes (including silences) to 3 hours in 6 steps, as shown in Fig. 2. It has been shown that with the increase of the amount of training data errors keep decreasing while the correlation measures increase. With 5 or 10 minutes of training data, the networks are able to generate hardly intelligible speech. With 15 minutes, the speech is almost fully intelligible but still with a lot of artefacts and muffled phones, and it sounds quite unnatural. With 30 minutes, there is an increase in both intelligibility and naturalness. Some speech artefacts are present, but the phone boundaries are clear and speech tempo seems quite natural. Further increase of the training set, even with a factor of 6, does not bring a difference in quality as significant as when increasing the training set from 5 to 30 minutes. However, although the differences are not significant, it can be noted that there is definitely improvement with the increase of the training set, confirmed with both objective measures and informal listening tests. The best objective measures achieved (with 3h of data for training) are: 4.31 dB for MCD, 0.24 dB for BAP, 3.21% for V/UV and correlation of 0.93 for phone durations and 0.87 for F0.

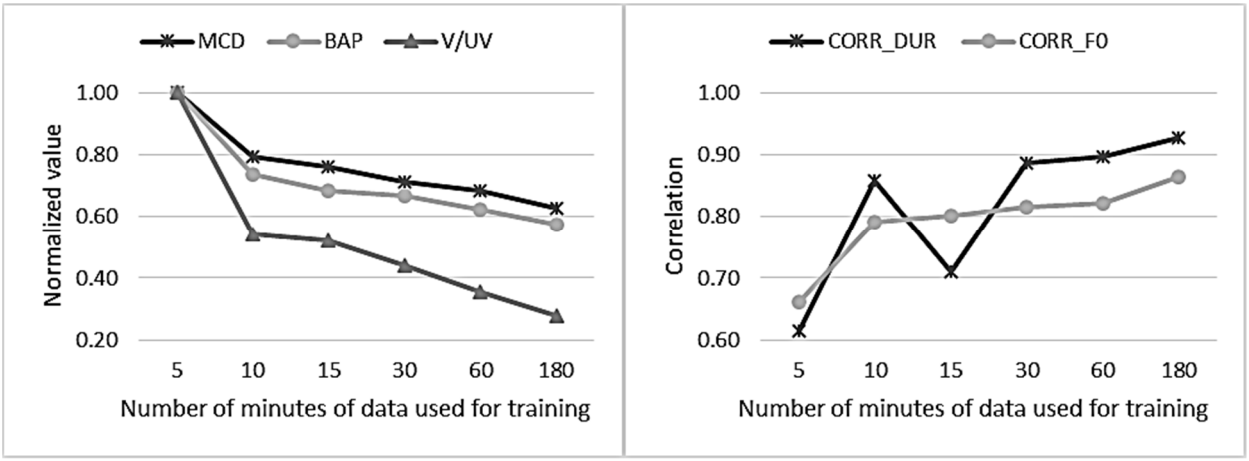

Fig. 2. Influence of the amount of the training data. Left: Influence on MCD, BAP and V/UV (values of objective measures are normalized by their respective maximum values). Right: Influence on correlation of F0 and phone durations.

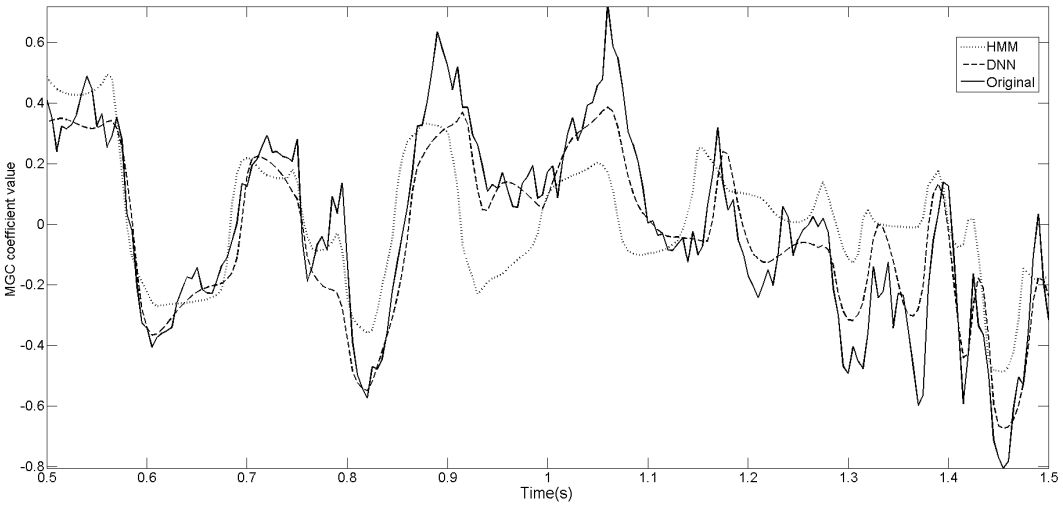

Fig. 3. Trajectory of 6th MGC coefficient (out of 40) generated by HMM, DNN and extracted from the original recording.

V. Comparison of Speech Synthesizers

As mentioned earlier HMM-based synthesis was for a long time state-of-the art among parametric speech synthesis methods. In order to compare parametric approaches for speech synthesis in terms of overall quality a set of experiments were performed.

HMM system was based on parameters extracted by WORLD vocoder - 40 MGCs representing spectral envelope, logarithm of fundamental frequency and 2 band aperiodicity parameters were used. Five-state, left-to-right, no-skip hidden semi-Markov models (HSMMs) were used. Logarithm of f0 and band aperiodicity parameters were modeled using multi-space probability distribution (MSD). The number of questions in context-clustering is 617 and default values of 1 for parameters controlling tree size were used. For DNN system, the same vocoder, acoustic parameters and database are used. For the NN architecture the one with best objective measures is chosen – 4 tangent hyperbolic hidden layers each with 512 units per layer. Both systems were trained on same database consisting of 3 hours of speech (the best NN objective measures are obtained with this architecture and database).

In order to objectively compare HMM and DNN synthesis, generated acoustic parameters, specifically their trajectories are compared with the trajectories of parameters extracted from the original recordings. The trajectories of several lowest MGC coefficients are almost the same for both DNN and HMM, and follow the original trajectory almost perfectly. Significant differences between HMM and DNN trajectories, as well as differences from the original utterance, start to occur on the 6th MGC coefficient (Fig. 3) and more drastically differ from the original one. Nonetheless, it can be seen that the DNN trajectory follows the original one much better than the HMM trajectory, and that there are no significant deviations. For higher coefficients, the differences between HMM and DNN are more emphasized and deviations from the original one increase. In other words, neither DNN nor HMM are able to accurately predict spectral details. All three trajectories (HMM, DNN and original) of fundamental frequency are more similar among themselves than the trajectories of MGC coefficients (Fig. 4). The DNN trajectory is, again, a better match to the original than the HMM trajectory.

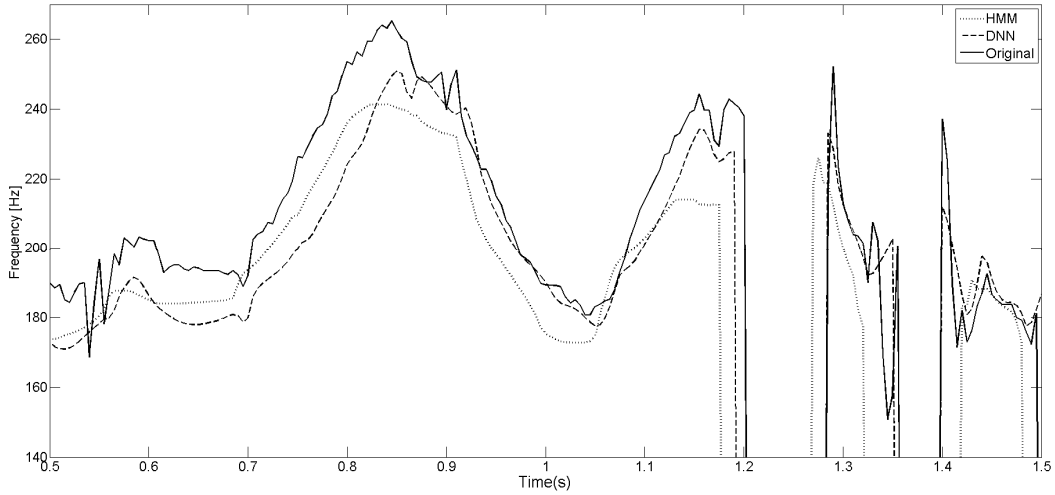

Fig. 4. Trajectory of the fundamental frequency generated by HMM, DNN and extracted from the original recording.

Furthermore, in Fig. 4, it can be clearly seen that DNN is a better predictor of whether a frame is voiced or not.

Subjective evaluation of the quality of synthesized speech for HMM and DNN approach was carried out by listening tests. Participants were 40 students, native speakers of Serbian, without expert knowledge of speech technologies. The test was distributed by Google forms and was organized in the following way. There were 3 audio files, each containing the same 4 unrelated utterances. The first file is generated by the HMM model, the second by the DNN model and the last one represents original recordings (natural speech). For each audio file, the participants were asked to answer 10 questions, taken from [15], by giving grades from 1 to 5. The first question is related to the overall impression, the next 5 are related to the intelligibility while the remaining 4 are related to the naturalness of the synthesized speech. Such a MOS scale approach, with 10 questions instead of conventional MOS evaluation, has been chosen in order to get a more detailed idea of the shortcomings of synthesized speech as perceived by humans. Figure 5 presents average grades for two main features of synthesized speech – intelligibility and naturalness. DNN performs better than HMM for both intelligibility and naturalness by almost half of a grade, while it lags behind original recordings for just 0.25 for intelligibility and 0.43 for naturalness.

By observing answers to individual questions, it can be noted that for each and every question, HMM got the lowest, and the original recording got the highest grade. For DNN, the grade was never below 3.7 for any of the questions, while for HMM it went below 3 once, for a question related to the naturalness of voice. The overall average grade, calculated by averaging grade for naturalness, intelligibility and the grade on the question about overall impression, for original recording is 4.7, for DNN is 4.3 and for HMM is 3.8.

VI. Conclusion

In this paper a review of parametric speech synthesis for Serbian based on neural networks is given. In a set of experiments we have shown that a network with 4 tangent hyperbolic hidden layers, each with 512 units per layer produces the best objective measures, and still is not too complex and does not require too many hours for training.

The DNN synthesizer performs better than HMM one both objectively and subjectively, and for the established values of hyper-parameters and amount of training data, produces speech which is very close to original, in terms of both intelligibility and naturalness. Further improvements of vocoder or integration with the newest methods for synthesis, such as WaveNet, are necessary.

Exploiting the parametric nature of DNN speech synthesis, our future work will focus on experiments related to voice conversion.

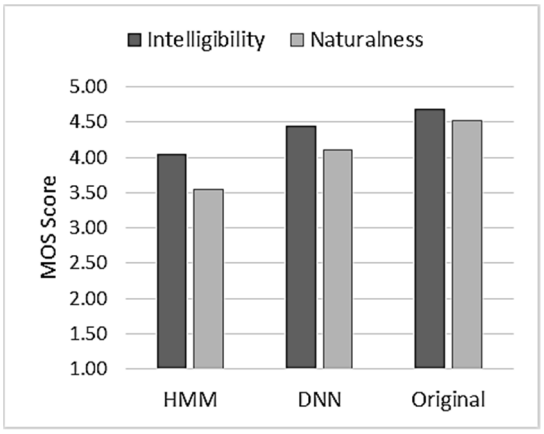

Fig. 5. Subjective evaluation of parametric synthesizers.

References

[1] Company AlfaNum, http://www.alfanum.co.rs/

[2] E. Pakoci, "Speech synthesis based on hidden Markov models for Serbian language," Bachelor thesis, FTS Novi Sad, 2012.

[3] R. Mak, "Use of global variance in order to improve quality of speech synthesis based on hidden Markov," Bachelor thesis, FTS Novi Sad, 2012.

[4] H. Zen, A. Senior, M. Schuster "Statistical parametric speech synthesis using deep neural networks," in *Proc. IEEE ICASSP 2013*, Vancouver, Canada, 2013, pp. 7962–7966

[5] T. Delić, M. Sečujski, "Speech synthesis for Serbian language based on artificial neural networks", *Telecommunication forum TELFOR 2016*, Belgrade, ISBN 978-1-5090-4085-8, pp. 403-406.

[6] The Centre for Speech Technology Research, http://www.cstr.ed.ac.uk/

[7] M. Schröder, "Expressive Speech Synthesis: Past, Present, and Possible Futures", *Affective Information Processing*, part II, London, 2009, pp. 111-126.

[8] H. Zen, K. Tokuda, A.W. Black, (2009). "Statistical parametric speech synthesis," *Speech Communication*, 51(11), 1039-1064.

[9] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, H. Yamagishi, and K. Oura, "Speech synthesis based on hidden Markov models," *Proc. IEEE*, vol. 101, no. 5, pp. 1234–1252, 2013.

[10] J. Odell, "The use of context in large vocabulary speech recognition," Ph.D. thesis, Cambridge Univ., 1995.

[11] M. Sečujski, "Obtaining Prosodic Information from Text in Serbian Language", in *Proc. IEEE EUROCON 2005*, Srbija, Belgrade, 2005, pp. 1654-1657

[12] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V Valtchev, "The HTK book (for HTK version 3.4)" Cambridge university engineering department. 2006

[13] M. Sečujski, V. Delić, "Automatic conversion of textual information into speech," Monographic series ISSN 1820-3418, of Scientific-technical information, ISBN 978-86-81123-25-6, Vol. XLVI, No. 4, VTI, Belgrade, 2011.

[14] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE transactions on information and systems*, 2016, vol. E99-D, no. 7, pp. 1877-1884

[15] K. Tokuda, T. Masuko, T. Yamada., T. Kobayashi and S. Imai, "Speech parameter generation algorithms for HMM-based speech synthesis", In *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, Vol. 3, pp. 1315-1318

[16] M. Viswanathan, M. Viswanathan, "Measuring speech quality for text-to-speech systems: development and assessment of a modified mean opinion score (MOS) scale", *Computer speech & language*, vol. 19, 2005, pp. 55–83.