



Rapping-Singing Voice Synthesis based on Phoneme-level Prosody Control

Konstantinos Markopoulos*, Nikolaos Ellinas*, Alexandra Vioni*, Myrsini Christidou*,
Panos Kakoulidis*, Georgios Vamvoukakis*, Georgia Maniati*, June Sig Sung†,
Hyoungmin Park†, Pirros Tsiakoulis*, Aimilios Chalamandaris*

* Innoetics, Samsung Electronics, Greece

† Mobile Communications Business, Samsung Electronics, Republic of Korea

{k.markop, n.ellinas}@samsung.com,
{a.vioni, m.christidou}@partner.samsung.com,
{p.kakoulidis, g.vamvouk, g.maniati, js6.sung, hm94.park, p.tsiakoulis,
aimilios.ch}@samsung.com

Abstract

In this paper, a text-to-rapping/singing system is introduced, which can be adapted to any speaker's voice. It utilizes a Tacotron-based multi-speaker acoustic model trained on read-only speech data and which provides prosody control at the phoneme level. Dataset augmentation and additional prosody manipulation based on traditional DSP algorithms are also investigated. The neural TTS model is fine-tuned to an unseen speaker's limited recordings, allowing rapping/singing synthesis with the target's speaker voice. The detailed pipeline of the system is described, which includes the extraction of the target pitch and duration values from an a capella song and their conversion into target speaker's valid range of notes before synthesis. An additional stage of prosodic manipulation of the output via WSOLA is also investigated for better matching the target duration values. The synthesized utterances can be mixed with an instrumental accompaniment track to produce a complete song. The proposed system is evaluated via subjective listening tests as well as in comparison to an available alternate system which also aims to produce synthetic singing voice from read-only training data. Results show that the proposed approach can produce high quality rapping/singing voice with increased naturalness.

Index Terms: text-to-speech, rapping voice synthesis, singing voice synthesis, text-to-rapping, text-to-singing, prosody control, prosody manipulation, neural models

1. Introduction

With the recent development of neural text-to-speech (TTS), the task of singing voice synthesis (SVS) is gaining popularity, since it has become feasible to produce natural and expressive speech more effectively. Before the development of deep neural network based synthesis, SVS systems were mainly based on unit selection technology [1, 2, 3, 4] or parametric TTS [5, 6]. During the last few years, with the establishment of neural TTS systems such as Tacotron [7], it has become possible to investigate approaches like neural rapping and singing.

SVS is a complicated task. As in regular TTS, a large and powerful neural model that accurately predicts acoustic features must be designed and tuned. Additionally, singing information such as musical notes and rhythm must be accurately followed, in order to produce high quality samples. For rapping, the procedure is the same, though the focus is less on musical notes, and more on pitch variation and rhythm, the latter translating into accurate phoneme durations.

1.1. Related work

Several approaches of rapping and singing voice synthesis have been proposed over the years. Early methods were based on unit concatenation [1, 2, 3, 4] and statistical parametric synthesis [5, 6]. There was also an attempt focused on speech-to-rap voice conversion, based on a phase vocoder and beat tracking [8]. Nevertheless, such approaches had significant limitations and did not achieve high quality synthesized speech. In recent years, many steps have been made towards high quality SVS, with the use of neural and deep learning methods. Hybrid, mixed and conditioned models have been introduced that advance further the SVS systems. Some notable approaches are the WaveNet variant architecture [9] used for parametric singing synthesis, WGANsing which is a pitch conditioned Generative Adversarial Network (GAN) [10] and an adversarially trained, pitch conditioned sequence-to-sequence Korean singing model [11]. Another GAN-based approach is unsupervised cross-domain singing voice conversion [12], which uses additional perceptual losses on its generator output. Mellotron [13], a multi-speaker expressive voice synthesis model based on GST-Tacotron 2 [14], also has SVS capabilities. Moreover, Jukebox [15] generates singing voice with accompaniments, and UTACO [16] consists of an attention-based sequence-to-sequence mechanism and a vocoder with dilated causal convolutions. Another approach was Durian-SC, a duration-informed, phoneme-to-acoustic features' alignment model with composite conditioning that allows SVS [17]. The transformer architecture has also been employed to address the problem, as in DeepSinger [18], which employs separate encoders for its features, and HiFiSinger [19], which includes a FastSpeech-based [20] model and multi-scale adversarial training approaches. Recently, ByteSing [21] was introduced, a Tacotron 2 model combined with a duration prediction model and uses linguistic along with musical embeddings.

1.2. Proposed method

In this paper, we propose a complete text-to-rapping and singing approach based on a system that, unlike other methods, relies solely on spoken data and can be adapted to an unseen target voice with very limited data. Our method is based on a fine-grained prosody manipulation multi-speaker TTS model presented in [22]. Combined with augmentation of training data, our method can achieve phoneme-level prosody manipulation (F0 and duration), which allows us to generate rapping and singing synthesized speech.

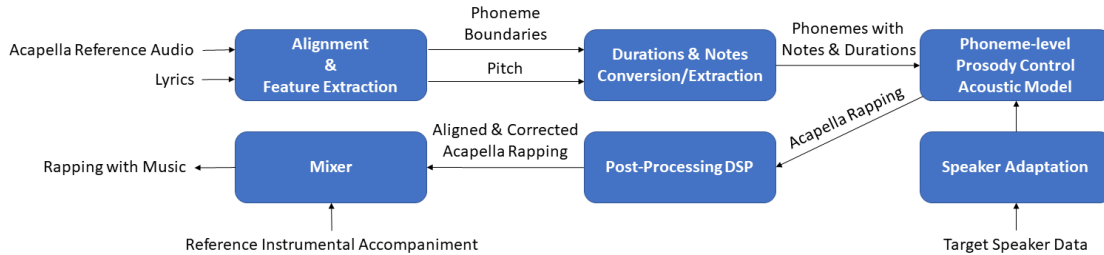


Figure 1: Proposed text-to-rapping/singing system

We train a multi-speaker multilingual TTS model on internal spoken data, in US English and Korean languages, and show that the prosody control capabilities at the phoneme-level musical note and duration are effective in all training speakers. For the case of speaker adaptation with very limited data, we re-sume the model training on 2 speakers of both languages with only about 11 minutes of spoken data, and show that the resulting models can also maintain the same prosody control capabilities. Extracting the desired prosody information from a given a capella song and presenting it as input to our model enables us to synthesize a capella utterances that closely follow the reference, allowing rapping/singing synthesis with the voice of every speaker included in the training set.

We present the proposed approach, from the stage of phoneme alignment and F0 extraction and discretization, up to the final post-processing steps required to produce the actual song with the voice of the selected speaker. This includes several processing steps and modules as well as a fine-tuning digital signal processing (DSP) stage based on synchronous overlap-add algorithms [23, 24], in order to achieve exact alignment of the synthetic speech to the target song music track. Finally, our system is evaluated via crowd-sourced listening tests against the ground truth samples, as well as against samples from Mellotron, a widely accepted state-of-the-art SVS system. Objective evaluation results are also presented, showing that our model follows the desired prosody patterns appropriately.

2. Method

2.1. Overview

Our proposed text-to-rapping/singing system aims at producing a song with the voice of a target speaker, based only on the lyrics and an a capella version of the song by its original singer. A block diagram of the system is shown in Figure 1 and consists of four main parts.

The first part involves the preprocessing of the lyrics by a front-end module in order to obtain the phonemes, as well as the extraction of the required prosodic features from the reference audio, i.e. phoneme durations and F0 (Section 2.2). Second, to avoid extreme F0 values which can hinder the output quality, F0 contours are converted so as to lie within the range of the target speaker (Section 2.3). Third, the converted F0 values along with the phoneme durations are discretized and used by the TTS model, which produces the synthesized song by controlling the prosody at the phoneme-level (Section 2.4). The fourth step is post-processing, including the introduction of a DSP module, which is necessary for accurate time alignment of the produced song with the original time specification, if the instrumental accompaniment track needs to be combined. This task can be performed by a mixer which combines the synthe-

sized a capella utterances with the music track (Section 2.5). A detailed analysis of each step is given in the following sections.

2.2. Alignment & feature extraction

A capella songs are used as reference since the acoustic model is trained only on neutral style spoken data which do not contain music tracks. Initially, phoneme alignments are automatically extracted from the song using an HMM monophone acoustic model trained using flat start initialization [25]. As the forced-alignment model is trained only on spoken utterances and may not produce precise phoneme boundaries, a manual correction of the alignments was performed. Most corrections were attributed to accommodate for long vowel durations in the singing data, which are not usually encountered in spoken speech.

F0 contours are calculated using the algorithm included in the Praat toolkit [26]. Interpolation is applied on the unvoiced regions in order to avoid zero values and the final contour is smoothed.

2.3. Note conversion/extraction

The acoustic model is trained to control the prosody within the range of each speaker. When a reference song is used as input to the proposed system, the F0 contour of the singing speaker may have different range than that of the target speaker. This is due to the fact that the song originates from a different speaker with diverse source characteristics or gender. Also, the singing speech itself may vary in extreme F0 values both in the lower and higher end between two speakers.

We use Eq (1) in order to transpose the F0 contour of the reference speaker to match the range of the target speaker:

$$f_{target} = \frac{\text{median}(f_{speaker})}{\text{median}(f_{ref})} \cdot f_{ref} \quad (1)$$

where $f_{speaker}$ and f_{ref} represents all the F0 values of the target and reference speaker’s utterances respectively.

Two approaches were examined on F0 transposition. The first approach takes one single (global) pitch value for the reference speaker, which is the median F0 value, while the second recalculates the median F0 value for every verse. Early results demonstrated that the former approach works better, producing more stable samples, and avoids the discontinuities in octaves that may occur due to the recalculation of the F0.

After the conversion, the average phoneme-level F0 is calculated by using the previously extracted alignments. The corresponding musical note and octave can be extracted by using formulas (2) and (3):

$$h = \left\lceil 12 \cdot \log_2 \frac{f_{target}}{440} \right\rceil + 57 \quad (2)$$

$$octave = \left\lfloor \frac{h}{12} \right\rfloor \quad \text{and} \quad note = (h \bmod 12) \quad (3)$$

where h represents the distance in semitones from the note C_0 .

2.4. Acoustic model

For controlling prosody, we use an acoustic model based on previous work [27]. This method uses unsupervised clustering on phoneme-level F0 and duration values in order to extract a sequence of discrete learnable prosodic labels for each utterance, which is then used to condition the decoder of a Tacotron-based acoustic model [7, 14] in parallel to the phoneme sequence. The model is also augmented with a secondary Mixture-of-Logistics (MoL) attention module [28] which operates on the prosodic sequence only, aiming to disentangle the phonetic and prosodic content. The final model is able to control the prosody at the phoneme level both for F0 and duration while maintaining high speech quality.

In [27], it is shown that this model is also effective at representing musical notes instead of F0 values that are derived from K-means unsupervised clustering. In the current work, we follow the procedure described in Section 2.3 to assign the notes and octaves to separate learnable embeddings, so that any possible combinations are modeled appropriately, even non-existing ones in the training set. This global representation of musical notes is also speaker-independent, being suitable for our multi-speaker setup described below.

For the duration labels, we follow an improved and more stable method than K-means, as in [22]. The values are sorted in ascending order and grouped into a desired number of intervals, so that an equal number of samples is contained in each interval. This alleviates the problem of voice quality deterioration in extreme values which are not common in the training set, while slightly decreasing the duration control range. In this work, in order to investigate the effect of the number of labels involved, we have experimented with two different setups, one with 15 and another with 30 duration labels.

The training setup follows a multi-speaker/multi-lingual scenario, similar to the parallel work done in [22]. A multitude of speakers of both genders and 2 languages is used instead of a single one, in order to capture as many prosody patterns as possible, which will help increase the range of the model in both F0 and duration. That way each speaker has the capability to rap/sing a wider variety of songs. Augmentation is also employed both in F0 and duration by applying pitch shifting and tempo alteration respectively, further increasing the quantity and range of the training data. This setup also allows us to perform speaker adaptation using limited data from unseen speakers and enable rapping/singing for the adaptation speaker in both languages. This process becomes easier as the model does not have unseen values due to the global musical note representation for the F0 and the duration intervals which are derived from all speakers and are common throughout the training procedure.

The different speakers are assigned a learnable speaker embedding, which also conditions the decoder at each step. We also use a linear adversarial speaker classifier on the phoneme encoder outputs, in order to make them speaker independent, as well as a residual variational encoder which captures other latent factors of the recordings [29]. This method is shown to improve naturalness and stability by simply using a zero vector during inference, which is essentially the prior mean. The multilingual setup does not include language embeddings, but simply considering every phone for each language with a different

Table 1: *Multilingual multi-speaker dataset for text-to-singing/rapping model training. ‘tr’ and ‘ad’ refer to training and adaptation speakers respectively, while ‘f’ and ‘m’ refer to gender.*

Speaker	Language	Rec. Hours	Utterances
us_tr_f1	en-us	41.21	36185
us_tr_f2	en-us	38.88	45841
us_tr_m1	en-us	36.82	40442
ko_tr_f1	ko	51.37	40503
ko_tr_f2	ko	54.29	29289
us_ad_m1	en-us	0.19	165
ko_ad_m1	ko	0.18	149

label, increasing the total number of phones to the sum of each language phoneset. For the speaker adaptation, we found that freezing the weights of the phoneme encoder, prosody encoder and attention modules yields better results. We can account this to the fact that the model has already learned rich representations which must not be forgotten in the adaptation stage by the target speaker’s limited data.

2.5. Post-processing

The multi-speaker/multi-lingual prosody control model produces a capella utterances which are derived between silence tokens from the original lyrics. These utterances must be concatenated in the time domain in order to obtain the full verse of the song. Additionally, the duration values are discrete in our model, resulting in some form of quantization, hence the final durations may not exactly match the original song. The accurate matching of the synthesized song with the original, requires lengthening or shortening of speech, which in our case is performed utilizing time-domain DSP methods.

This stage is based on two main algorithms: WSOLA [24] and PSOLA [30]. These algorithms are both able to modify the duration of each phoneme without affecting the pitch and resynthesize the original audio using the overlap-add technique. In early listening experiments, we found that WSOLA produced slightly better results in almost every comparison, so we included this method in the evaluations that follow. A Mixer element can also be included for producing the final song, mixing the time-aligned processed synthetic a capella utterance with the respective music track. Mixing more than one voices is also possible, providing a multi-speaker song.

3. Experiments & results

In this section, we describe our experimental setup, along with the method followed to objectively and subjectively evaluate the proposed system. The evaluation results are then discussed.

3.1. Experimental setup

The acoustic model is trained with an internal multilingual multi-speaker dataset containing 222 hours of speech in both US English (en-us) and Korean (ko) from 1 male and 4 female speakers. The recorded dataset is a general TTS corpus of neutral speaking style. For speaker adaptation, we use about 11 minutes of recordings from an unseen male speaker from each language. The adaptation utterances are selected with a corpus selection process described in [31] which ensures maximum phonetic coverage for the required amount of recordings. Details regarding the data used for training and adaptation are

Table 2: Average semitone pitch & duration (ms) score for objective evaluation

		Training Speakers	Adaptation Speakers
Pitch Error (semitones)	15-class	0.85 ± 0.22	0.86 ± 0.23
	30-class	0.93 ± 0.28	0.93 ± 0.21
	Mellotron	0.51 ± 0.15	
Duration Error (ms)	15-class	27 ± 9	29 ± 8
	30-class	29 ± 10	28 ± 7
	Mellotron	25 ± 9	

presented in Table 1. The augmentations applied to the original training data, similar to [22], include increasing and decreasing the F0 by 2, 4 or 6 semitones, and the speaking rate from 70% - 130%. The final training set size after the augmentations is 415 hours.

All audio data was resampled to 24 kHz. The acoustic features were extracted in order to match the modified LPCNet Vocoder [32] and consist of 20 Bark-scale cepstral coefficients, the pitch period and pitch correlation. The phoneme encoder maps the input phoneme sequence into 256 dimensional embeddings and further applies a CBHG module. In the prosody encoder, prosodic labels are mapped into 64 dimensional embeddings. These are processed by a single 128-dimensional feed-forward Pre-Net with ReLU activation and a bidirectional Gated Recurrent Unit (GRU) layer with 128 dimensions in each direction. The decoder contains 3 recurrent layers, a 256-dimensional attention GRU and two 512-dimensional residual LSTMs. The attention modules used have a mixture of 5 logistic distributions and 256-dimensional feed-forward layers. Dropout regularization [33] of rate 0.5 is applied on all Pre-Net and Post-Net layers and Zoneout [34] of rate 0.1 is applied on LSTM layers. We use the Adam optimizer [35] for training the network parameters with batch size 32. The learning rate is initially 10^{-3} and decays linearly to $3 \cdot 10^{-5}$ after 100,000 iterations. We also apply L2 regularization with factor 10^{-6} . Speaker adaptation involves further training of the prosody model for 5K iterations, with frozen weights in the phoneme encoder, prosody encoder and attention modules.

3.2. Objective evaluation

Objective evaluation of the proposed systems in terms of a capella song synthesis was attempted. Clips from 4 rap songs and 4 songs, all sung a capella in English, were used as ground truth for this evaluation¹. All the proposed models along with Mellotron [13] were evaluated against the ground truth a capella songs. The samples for Mellotron were produced by Mellotron’s latest GitHub repository along with a pretrained Mellotron model and respective WaveGlow model based on LibriTTS dataset. A random female voice from the model was selected for inference of the audio stimuli.

In Figure 2, the pitch contours of 2 a capella reference songs are presented, along with the pitch contours of the audio stimuli inferred by the systems under comparison. One can notice that the synthesized pitch contours closely follow the reference ones, while they are shifted appropriately in order to match the F0 mean value of the inference speaker of each model. In Figure 3, where the F0 contours of a reference and synthesized clip are illustrated in a MIDI-like graph, a MIDI value is produced

¹The reader is encouraged to listen to the audio samples at: <https://innoetics.github.io/publications/rappotron/index.html>

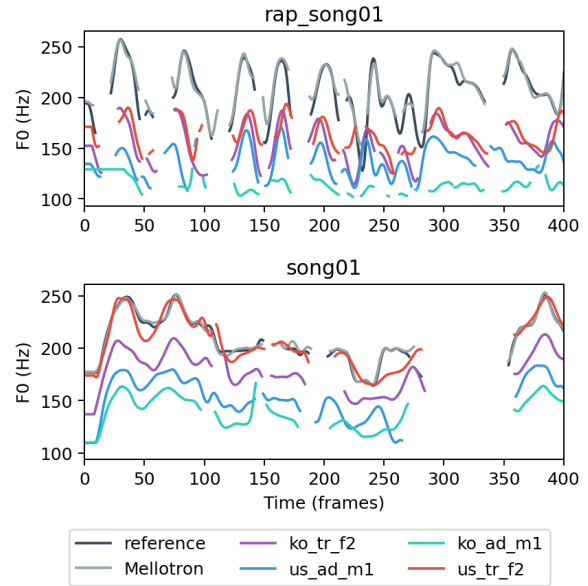


Figure 2: Pitch contours of the original song, Mellotron and proposed models.

for each phoneme, and the trend of the synthesized melody following in parallel the ground truth one is obvious.

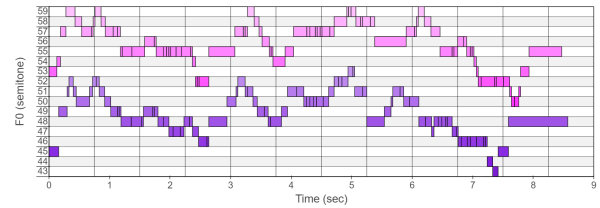


Figure 3: Pitch contours of the original and synthesized song in a MIDI representation (pink and purple values respectively). The MIDI values are calculated on a per phoneme basis.

As an objective metric for measuring the accuracy of our approach, we calculated the average distance of F0 and duration between the reference and the synthesized audio on a per phoneme basis. F0 values of the reference audio were transposed again, so as to target the speaker’s mean F0 value before calculation of the distance. As depicted in Table 2, there is no significant difference in these metrics between speakers seen in the training set and those used only for adaptation, neither on F0 nor on duration values. This observation is valid for both 15 and 30 duration label models. Mellotron seems to achieve an F0 contour closer to the reference, as was also illustrated in Figure 2.

Overall, our method manages to reproduce a capella singing with accurate pitch and duration phoneme values especially in cases where the respective a capella reference audio is straightforward, without notably long durations or extreme low/high notes. In the latter cases, we noticed that our approach did not accurately produce phonemes with the target F0 or duration values, leading either to attention failures or duration mismatches with the reference audio.

Table 3: Mean Opinion Score (MOS) evaluation results with 95% confidence interval

setup	30 duration labels				15 duration labels			
	rap_songs		songs		rap_songs		songs	
	plain	w/ post-proc	plain	w/ post-proc	plain	w/ post-proc	plain	w/ post-proc
us_tr_f1	3.60±0.36	3.65 ±0.30	3.36±0.33	3.71 ±0.33	3.65±0.32	3.72 ±0.35	3.69±0.33	3.69 ±0.29
us_tr_f2	3.60±0.30	3.86 ±0.29	3.38±0.38	3.50 ±0.38	3.53±0.31	3.74 ±0.35	3.81±0.29	3.55 ±0.37
us_tr_m1	3.60±0.33	4.00 ±0.29	3.64±0.32	3.60 ±0.36	3.51±0.27	3.56 ±0.32	3.64±0.34	3.62 ±0.29
ko_tr_f1	3.35±0.33	3.70 ±0.36	3.17±0.34	3.33 ±0.41	3.44±0.37	3.35 ±0.34	3.81±0.32	3.57 ±0.31
ko_tr_f2	3.37±0.34	3.42 ±0.35	3.45±0.39	3.33 ±0.37	3.40±0.32	3.40 ±0.34	3.69±0.35	3.81 ±0.32
us_ad_m1	3.67±0.35	3.74 ±0.29	3.74±0.35	3.33 ±0.37	3.79±0.31	3.51 ±0.35	3.57±0.34	3.33 ±0.39
ko_ad_m1	3.42±0.39	3.47 ±0.39	3.26±0.42	3.26 ±0.40	3.63±0.36	3.51 ±0.32	3.26±0.38	3.43 ±0.38
Total	3.52 ±0.34	3.69 ±0.33	3.43 ±0.36	3.44 ±0.37	3.56 ±0.32	3.54 ±0.33	3.64 ±0.33	3.57 ±0.33
Mellotron	3.33±0.38		3.17±0.49					

3.3. Subjective evaluation and discussion

A subjective evaluation was carried out via a formal listening test. Two clips from songs and two clips from rap songs, all in English a capella, were used as ground truth, and the respective audio stimuli produced by our system and Mellotron were rated. In the framework of this subjective evaluation, aside from the overall quality, we also opted to assess: a) the effect of the number of duration labels per phoneme, and b) the effect of the post-processing stage for duration matching to the reference audio via WSOLA (as described in Section 2.5). By combining the aforementioned parameters we produced all 4 possible models for each speaker described in Table 1. In total, 60 listeners (Amazon Mechanical Turkers) participated in the listening test, rating each synthetic stimulus on a 5-point Likert scale on both melody and intelligibility, with 1 indicating “Totally off-tune or wrong lyrics” and 5 indicating “Exactly same melody and lyrics”.

The average MOS and 95% confidence interval for each voice model versus song type (rap song and song) and post-processing are presented in Table 3. The results show that our approach provides satisfactory output, equally for both rapping and singing, even if the latter is considered as a more complex task. Overall, post-processing for matching the word boundaries between ground-truth and audio stimuli does not seem to provide any consistent and robust improvement. Although there is no statistical significance in the pairwise differences observed between models, an improvement tendency is observed when post-processing is used for rap songs and the 30-duration-labeled model. This post-processing stage remains necessary in order to align the generated songs with the music track at the final mixing stage, in case of music accompaniment of the synthesized a cappella voice. The 15-class duration labeling yields an improvement tendency for singing, which is most probably attributed to the fact that fewer but more populated classes in training entail better learning for our model.

A closer inspection of the results leads us to the conclusion that adapted speakers achieve similar quality results with the speakers who are seen during training. This is prominent especially for English, where the adapted voice performs equally well to the rest of the English training set voices, while, at the same time, the Korean adapted voice follows closely the performance of the Korean training voices. Such similar MOS ratings confirm our hypothesis that our approach is robust for limited speaker data scenarios. As far as the per speaker and language performance is concerned, Korean voice models have received lower MOS scores than the English ones. This is most probably due to the fact that the reference songs we evaluated are exclu-

sively in English. Our informal evaluation showed that these non-native voice models mainly suffered from lower intelligibility or lack of naturalness, as they did not bear native English accent and thus sounded more artificial when synthesizing English a capella songs.

Our informal listening evaluation of the samples showed that Mellotron output is melodic but often bears intelligibility issues or audio artifacts. This observation may justify why Mellotron scored lower compared to our system in the formal subjective evaluation (Table 3). Nevertheless, Mellotron outperformed our approach in a test song where vibrato singing was prominent, a voice characteristic that Mellotron can capture and transfer well, in contrast to our approach where only F0 and duration values per phoneme are provided.

4. Conclusions

In this paper, we presented an approach for producing high-quality singing and rapping synthesis from a Tacotron-based fine-grained prosody-control voice model trained solely on read data. Even though its results do not match the output of an SVS system trained on singing data, our approach achieves satisfactory results in both singing and rapping. Experiments showed that equally good singing synthesis can be achieved for limited-data target voices via adaptation. It is worth-noting that our system, similarly to other systems based on spoken-only data, suffers limitations in its ability to produce too long and extremely low or high-pitched sounds. In other words, although it can provide satisfactory results for rapping and simple songs, it may fail to produce adequate singing for more challenging songs with wide variations in both note values and duration. Another native limitation to our current approach is the lack of ability to transfer micro-prosodic characteristics into the synthetic output, such as vibrato or tremolo. Mellotron was shown to better imitate micro-prosodic singing voice qualities with the help of a more complex model and attention mechanism. Further research on controlling singing voice characteristics, such as loudness and vibrato, as well as on including singing data in the training process is required. Moreover, we plan to investigate ways for automating parts of the proposed process, such as the alignment optimization of the target singing data, so as to eliminate manual effort in our method.

5. References

- [1] M. Macon, L. Jensen-Link, E. B. George, J. Oliverio, and M. Clements, "Concatenation-based midi-to-singing voice synthesis," in *Audio Engineering Society Convention 103*. Audio Engineering Society, 1997.
- [2] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 1. IEEE, 1996, pp. 373–376.
- [3] H.-Y. Gu and J.-K. He, "Singing-voice synthesis using demisyllable unit selection," in *2016 International Conference on Machine Learning and Cybernetics (ICMLC)*, vol. 2. IEEE, 2016, pp. 654–659.
- [4] J. Bonada, M. Umberto Morist, and M. Blaauw, "Expressive singing synthesis based on unit selection for the singing synthesis challenge 2016," *Morgan N, editor. Interspeech 2016; 2016 Sep 8-12; San Francisco, CA. ISCA; 2016. p. 1230-4.*, 2016.
- [5] K. Nakamura, K. Oura, Y. Nankaku, and K. Tokuda, "HMM-based singing voice synthesis and its application to Japanese and English," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 265–269.
- [6] K. Saino, H. Zen, Y. Nankaku, A. Lee, and K. Tokuda, "An HMM-based singing voice synthesis system," in *Ninth International Conference on Spoken Language Processing*, 2006.
- [7] W. et al, "Tacotron: Towards end-to-end speech synthesis," in *Proc. Interspeech*, 2017.
- [8] M. Wu, C. Lu, and J. R. Jang, "Automatic conversion from speech to rap music," in *2014 International Conference on Electrical Engineering and Computer Science (ICEECS)*, 2014, pp. 245–250.
- [9] M. Blaauw and J. Bonada, "A neural parametric singing synthesizer modeling timbre and expression from natural songs," *Applied Sciences*, vol. 7, no. 12, p. 1313, 2017.
- [10] P. Chandna, M. Blaauw, J. Bonada, and E. Gomez, "Wgansing: A multi-voice singing voice synthesizer based on the WassersteinGAN." IEEE, 2019, pp. 1–5.
- [11] J. Lee, H.-S. Choi, C.-B. Jeon, J. Koo, and K. Lee, "Adversarially trained end-to-end Korean singing voice synthesis system," 2019.
- [12] A. Polyak, L. Wolf, Y. Adi, and Y. Taigman, "Unsupervised Cross-Domain Singing Voice Conversion," in *Proc. Interspeech 2020*, 2020, pp. 801–805.
- [13] R. Valle, J. Li, R. Prenger, and B. Catanzaro, "Mellotron: Multispeaker expressive voice synthesis by conditioning on rhythm, pitch and global style tokens," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6189–6193.
- [14] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan et al., "Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.
- [15] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," *arXiv preprint arXiv:2005.00341*, 2020.
- [16] O. Angelini, A. Moinet, K. Yanagisawa, and T. Drugman, "Singing synthesis: with a little help from my attention," 2020.
- [17] L. Zhang, C. Yu, H. Lu, C. Weng, C. Zhang, Y. Wu, X. Xie, Z. Li, and D. Yu, "DurlAN-SC: Duration Informed Attention Network Based Singing Voice Conversion System," in *Proc. Interspeech 2020*, 2020, pp. 1231–1235.
- [18] Y. Ren, X. Tan, T. Qin, J. Luan, Z. Zhao, and T.-Y. Liu, "Deepsinger: Singing voice synthesis with data mined from the web," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 1979–1989.
- [19] J. Chen, X. Tan, J. Luan, T. Qin, and T.-Y. Liu, "Hifisinger: Towards high-fidelity neural singing voice synthesis," 2020.
- [20] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," *arXiv preprint arXiv:1905.09263*, 2019.
- [21] Y. Gu, X. Yin, Y. Rao, Y. Wan, B. Tang, Y. Zhang, J. Chen, Y. Wang, and Z. Ma, "Bytesing: A Chinese singing voice synthesis system using duration allocated encoder-decoder acoustic models and wavernn vocoders," 2021.
- [22] M. Christidou, A. Vioni, N. Ellinas, G. Vamvoukakis, K. Markopoulos, P. Kakoulidis, J. S. Sung, H. Park, A. Chalamandaris, and P. Tsiakoulis, "Improved prosodic clustering for multispeaker and speaker-independent phoneme-level prosody control," in *SPECOM*, 2021 [SUBMITTED].
- [23] F. Charpentier and M. Stella, "Diphone synthesis using an overlap-add technique for speech waveforms concatenation," in *ICASSP'86. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 11. IEEE, 1986, pp. 2015–2018.
- [24] W. Verhelst and M. Roelands, "An overlap-add technique based on waveform similarity (wsola) for high quality time-scale modification of speech," in *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2. IEEE, 1993, pp. 554–557.
- [25] S. Raptis, P. Tsiakoulis, A. Chalamandaris, and S. Karabetsos, "Expressive speech synthesis for storytelling: the innoetics' entry to the blizzard challenge 2016," in *Proc. Blizzard Challenge*, 2016.
- [26] P. Boersma, "Praat: doing phonetics by computer," <http://www.praat.org>, 2006.
- [27] A. Vioni, M. Christidou, N. Ellinas, G. Vamvoukakis, P. Kakoulidis, T. Kim, J. S. Sung, H. Park, A. Chalamandaris, and P. Tsiakoulis, "Prosodic clustering for phoneme-level prosody control in end-to-end speech synthesis," in *Proc. ICASSP*, 2021.
- [28] N. Ellinas, G. Vamvoukakis, K. Markopoulos, A. Chalamandaris, G. Maniati, P. Kakoulidis, S. Raptis, J. S. Sung, H. Park, and P. Tsiakoulis, "High quality streaming speech synthesis with low, sentence-length-independent latency," in *Proc. Interspeech*, 2020.
- [29] Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Z. Chen, R. Skerry-Ryan, Y. Jia, A. Rosenberg, and B. Ramabhadran, "Learning to Speak Fluently in a Foreign Language: Multilingual Speech Synthesis and Cross-Language Voice Cloning," in *Proc. Interspeech 2019*, 2019, pp. 2080–2084. [Online]. Available: <http://dx.doi.org/10.21437/Interspeech.2019-2668>
- [30] Y. Laprie and V. Colotte, "Automatic pitch marking for speech transformations via td-psola," in *9th European Signal Processing Conference (EUSIPCO 1998)*. IEEE, 1998, pp. 1–4.
- [31] A. Chalamandaris, P. Tsiakoulis, S. Raptis, and S. Karabetsos, "Corpus design for a unit selection TTS system with application to Bulgarian," in *Proc. 4th Conference on Human Language Technology: challenges for computer science and linguistics*, 2009, pp. 35–46.
- [32] R. Vippera, S. Park, K. Choo, S. Ishtiaq, K. Min, S. Bhattacharya, A. Mehrotra, A. G. C. Ramos, and N. D. Lane, "Bunched lpcnet: Vocoder for low-cost neural text-to-speech systems," *arXiv preprint arXiv:2008.04574*, 2020.
- [33] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [34] D. Krueger, T. Maharaj, J. Kramár, M. Pezeshki, N. Ballas, N. R. Ke, A. Goyal, Y. Bengio, A. Courville, and C. Pal, "Zoneout: Regularizing RNNs by randomly preserving hidden activations," *arXiv preprint arXiv:1606.01305*, 2016.
- [35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.