A VITS-Based Low-Resource Speech Synthesis System with Pitch Control

Prithwiraj Bhattacharjee

Department of Computer Science & Engineering
Leading University, Sylhet-3112, Bangladesh
rizubhattacharjee@gmail.com

***Abstract*—In recent years, end-to-end text-to-speech (TTS) has made significant progress. These models enable single-stage training and parallel sampling. However, their audio quality still falls short compared to traditional TTS systems. Moreover, generating speech with fine-grained prosody control remains a challenge. This work presents a pitch-controllable end-to-end TTS system based on the VITS architecture. The proposed method produces more natural-sounding audio than existing two-stage models. For efficient and high-fidelity speech synthesis, a vocoder is employed. Since speech signals consist of sinusoidal waves with different periods, the vocoder enhances the overall audio quality. The model is trained on 13 hours of phonetically balanced single-speaker Bangla speech data. It achieves a PESQ score of 1.26 on a scale of -0.5 to 4.5 in objective evaluation. Compared to existing non-commercial Bangla TTS systems, the proposed VITS-based approach demonstrates superior naturalness.**

***Index Terms*— TTS, Low-resourced, VITS, PESQ, Objective.**

I. Introduction (Heading 1)

Science and technology have advanced rapidly. It is becoming deeply integrated into human life. Among them, speech technology plays a vital role since speech is the most natural form of communication for human beings. As a result, it is natural to expect computers to engage in spoken dialogue with people. This requires the integration of speech and language technologies. Speech synthesis (TTS) is the process of generating artificial speech from text. Over the past decades, researchers have developed various approaches to TTS. Traditional systems are mainly classified into Concatenative TTS and Statistical Parametric Speech Synthesis (SPSS). Concatenative methods include unit selection, diphone synthesis, and domain-specific synthesis. Each of those has its own limitations, such as large database requirements or restricted generalization. SPSS methods, such as Hidden Markov Model (HMM)-based and Deep Neural Network (DNN)-based systems, have improved flexibility. However, these still relied on complex front-end processing, duration modeling, acoustic modeling, and vocoders. More recently, end-to-end neural TTS has emerged as a powerful alternative. End-to-end systems do not require hand-crafted linguistic features or large pre-recorded databases. Instead, they learn the mapping from text to speech in a single framework. Among these, VITS (Variational Inference TTS) has gained attention for its ability to train the acoustic model and vocoder jointly. It enables parallel sampling. However, challenges persist in generating high-quality audio and achieving fine-grained control over prosody. Low-resourced language like Bangla still lacks robust TTS systems. Existing Bangla TTS systems are limited and often fail to capture naturalness. To address this gap, this work introduces a VITS-based end-to-end Bangla TTS system. Moreover, it has built-in pitch control. The system is designed to produce more natural and expressive speech while overcoming the limitations of traditional and existing Bangla TTS approaches.

II. Literature Review

TTS research has advanced greatly over the past few decades. Early systems were language-specific and later evolved toward more language-independent platforms. Research began with concatenative methods. This is followed by Statistical Parametric Speech Synthesis (SPSS) for improved efficiency. More recently, end-to-end speech synthesis has become the state of the art. It has been growing across different applications in different languages. Very few works are available on Bangla TTS. The first attempt was Katha [1], a concatenative unit-selection-based system developed by BRAC University in 2007. With the rise of Statistical Parametric Speech Synthesis (SPSS), Bangla TTS research moved forward. Hidden Markov Model (HMM)-based systems [2] became common. HTS [3] is one of the most popular frameworks. In 2014, an HMM-based Bangla TTS system [4] was developed. But it produced less natural speech due to the limitations of vocoders and over smoothing in acoustic modeling [5]. The introduction of Deep Neural Networks (DNNs) [6] brought major improvements to SPSS. Neural networks have been applied to acoustic modeling since the 1990s [8]. The availability of larger datasets and greater computational power enabled impressive results [9]. Models such as RNNs [5], LSTMs [10], and BLSTMs [11] further enhanced SPSS performance. However, no significant DNN-based Bangla SPSS system has yet been developed. In 2016, Google [12] released a Bangla TTS system. It is not open source. Some resources [13] were later released to support Bangla TTS research. Several open-source tools, such as Kaldi [15], Merlin [16], and Ossian [17], provided researchers with platforms for building and experimenting with TTS. WORLD vocoder [18] has been introduced. Another open-source system is Idlak Tangle [14]. It also further contributed to the research community. In recent years, significant progress has been made in end-to-end TTS. Notable systems include Deep Voice 1–3 [19–21] from Baidu, Char2Wav [22] from MILA, and FastSpeech [23] from Microsoft. Google has also released notable works. Some of them are WaveNet [24], Tacotron [25], and Tacotron 2 [26]. These commercial systems are rarely open source. But many independent researchers have reproduced these models and shared their implementations online. These make them valuable starting points for new TTS research. SPSS approaches provided better results Compared to concatenative methods. But they still required extensive manual annotation and language-specific preprocessing. To overcome these challenges, researchers are now moving toward end-to-end TTS systems. It directly generates speech from (text, speech) pairs without manual preprocessing. This shift has set the stage for advanced models such as VITS. This enables more natural and efficient speech synthesis.

III. Dataset Preparation

The first step in building a high-quality TTS system is collecting a sufficient amount of speech data. As Bangla is a low-resourced language, large-scale datasets are not widely available. Google has released about 3 hours of Bangla speech data [13]. BRAC University has released a 13-hour phonetically balanced single-speaker dataset [27]. It is now considered the standard for Bangla TTS research. For this work, the BRAC dataset was used. It is then further processed to make it compatible with the proposed model. To ensure phonetic balance, additional text data were collected from various domains. The final corpus covered all possible Bangla pronunciations and consisted of more than 9,000 utterances. All recordings were stored in .wav format with a 24 kHz sampling rate. To implement the VITS model, preprocessing was applied to make the dataset suitable for training. Sentences with ≤3 words or ≥12 words were removed as well as their speech. Silence at the beginning and end of audio files was trimmed. Stereo recordings were converted to mono format. Long sentences were split. Audio files were normalized to maintain consistent amplitude and volume. Then, Text normalization is used. It is the process of converting raw text into a pronounceable and consistent form. It started with converting non-standard words (NSWs) into their standard pronunciations. Then, expanding numerical words into full spoken forms. Then, handling abbreviations and other ambiguous cases. All training text was fully normalized to ensure accurate pronunciation during speech synthesis. These steps ensured cleaner data and better learning during model training.

IV. Methodology

This section describes the proposed methods and their architecture, shown in Figure 1. The work focuses on VITS-TTS. It is a fully end-to-end text-to-speech system. VITS-TTS relies on a conditional Variational Autoencoder (VAE) formulation. Its alignment estimation is derived from variational inference. Its adversarial training can improve synthesis quality. The methodology used for the proposed Bangla TTS system is explained in detail. The rationale has also been made for selecting VITS as the primary framework for this work.

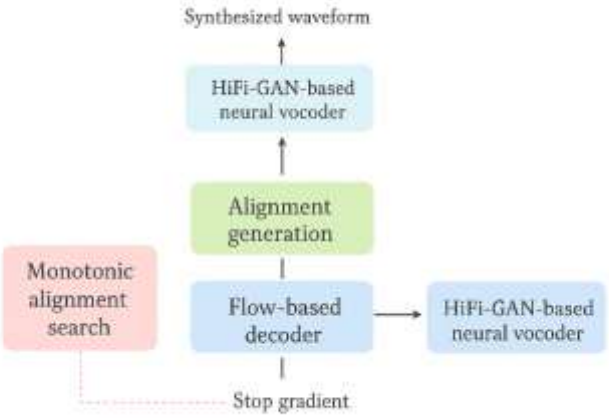

Fig 1: VITS Architecture for Low-resourced TTS

Variational Inference

VITS is a conditional Variational Autoencoder (VAE). Its goal is to maximize the evidence lower bound (ELBO) of the data. The training loss is the negative ELBO. It is the sum of the reconstruction loss and the KL divergence. In reconstruction loss, the model uses a mel-spectrogram as the target. Latent variables are up sampled to the waveform domain using a decoder. It is then converted to the mel-spectrogram domain. The difference between the predicted and target mel-spectrogram is used as the reconstruction loss. This approach improves perceptual quality. It is because the mel-scale mimics the human ear. The mel-spectrogram does not require trainable parameters. It uses STFT and linear projection. The mel-spectrogram is only used during training, not inference. The decoder does not process the full sequence at once for efficiency. It uses partial sequences in a method called windowed generator training. In KL divergence, the prior encoder uses phonemes from text as input. It also takes an alignment matrix. It shows how long each phoneme lasts in speech. The system estimates them during training since no true alignments exist. The posterior encoder gets more information by using the linear-scale spectrogram. Both the prior and posterior are modeled as factorized normal distributions. It makes the prior more expressive. This helps generate natural and realistic speech.

Alignment Estemation

The system uses Monotonic Alignment Search (MAS) to find the alignment between input text and target speech. The alignment is monotonic and non-skipping. It reflects how humans read text in order. MAS is adapted to maximize the ELBO. It reduces to finding an alignment that maximizes the log-likelihood of latent variables. From this alignment, phoneme durations are calculated. A stochastic duration predictor models these durations. It is to capture human-like variations in speech rhythm. Variational quantization and data augmentation are used to handle discrete durations. It also improves training. A stop-gradient operator prevents the duration predictor from affecting other parts of the model. Finally, adversarial training is applied with a discriminator. It distinguishes between generated speech and real speech. The system uses least-squares loss for the discriminator. Then it features a matching loss for the generator. This helps it produce natural and high-quality speech.

Model Architecture

The proposed model is composed of a prior encoder, posterior encoder, decoder, discriminator, and stochastic duration predictor. The posterior encoder and discriminator are used only during training.

- Prior Encoder: The prior encoder contains a transformer-based text encoder with relative positional representation. A linear layer projects the hidden representations to the mean and variance for constructing the prior distribution. A normalizing flow improves flexibility while remaining volume-preserving.
- Decoder: The decoder consists of transposed convolutions followed by multi-receptive field fusion (MRF) modules that combine residual blocks with different receptive fields.
- Discriminator: The discriminator follows the HiFi-GAN multi-period design. It uses sub-discriminators operating on different waveform periods, with the final setup covering [1, 2, 3, 5, 7, 11].
- Posteriori Encoder: The posterior encoder has 16 HiFi-GAN residual blocks. It produces latent variables of 192 channels from log-magnitude spectrograms. The decoder also operates on these 192-channel latent variables. To stabilize training, the bias term in the final convolution layer is removed.
- Stochastic Duration Predictor: The stochastic duration predictor estimates phoneme durations using residual blocks with dilated convolutions. Neural spline flows with rational-quadratic splines are applied for more expressive transformations. In the multi-speaker case, a speaker embedding is added through a linear layer.

Result Analysis

To assess the performance of the proposed system, objective evaluation metrics were used. The Perceptual Evaluation of Speech Quality (PESQ) measurement has been used for this. The Perceptual Evaluation of Speech Quality (PESQ) is a widely used objective measure. Among its variants, raw-PESQ and MOS-LQO are the most common. The value range for raw-PESQ is -0.5 to 4.5. The MOS-LQO ranges from 1.0 to 5.0. This study specifically chose the raw-PESQ score. To calculate PESQ, two waveforms are required: the original waveform and the synthesized waveform generated by the TTS system. In the experiment, 100 random sentences from the test dataset were selected along with their original recordings. Then the system generated speech for these sentences using our TTS system. For each pair of original and generated waveforms, the PESQ score has been calculated. Figure 5.3 shows the distribution of PESQ scores for all 100 synthesized waveforms.

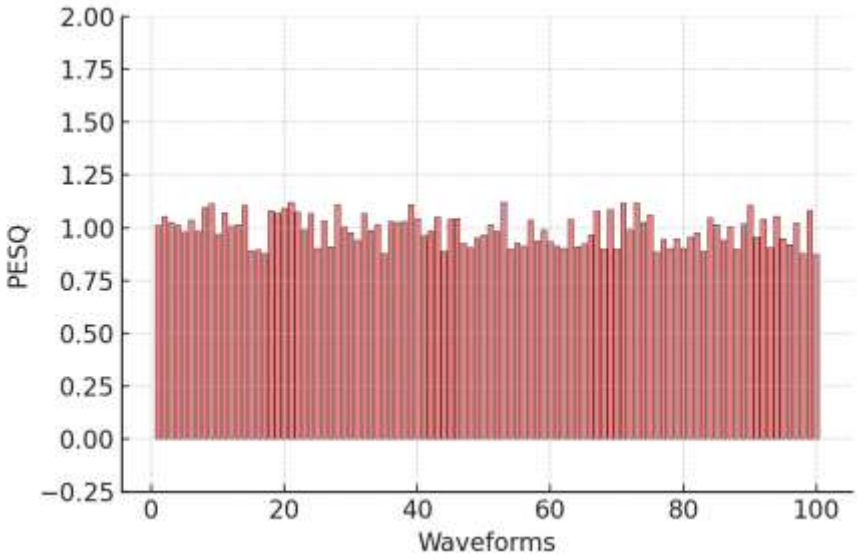

Fig 2: PESQ score of 100 synthesized wave in VITS-TTS

Conclusion

This work proposes a parallel TTS system based on VITS. It is capable of learning and generating speech in an end-to-end manner. The system synthesizes natural-sounding speech waveforms directly from text. It does not rely on predefined intermediate representations. Experimental results indicate that this method outperforms traditional two-stage TTS systems. It achieves speech quality close to human recordings. The approach can be applied to many speech synthesis tasks where two-stage systems are currently used. It can provide both performance improvement and a simplified training procedure. The pitch conditioning method is simpler than most approaches in the literature. It does not add computational overhead, such as interactive prosody adjustment. The model is fast, expressive, and shows strong potential for multi-speaker scenarios.

V. Acknowledgment

I would like to acknowledge Leading University CSE Department Thesis Committee, My colleagues, Co-supervisor and students for facilitating this research by providing resources.

References

[1] F. Alam, P. K. Nath, and M. Khan, “Text-to-speech for Bangla language using Festival,” 2007.

[2] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, “Simultaneous Modeling of Spectrum, Pitch and Duration in HMM-Based Speech Synthesis,” in *Sixth European Conference on Speech Communication and Technology*, 1999.

[3] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. W. Black, and K. Tokuda, “The HMM-Based Speech Synthesis System (HTS) Version 2.0,” in *SSW*, Citeseer, 2007, pp. 294–299.

[4] S. Mukherjee and S. K. D. Mandal, “A Bengali HMM Based Speech Synthesis System,” *arXiv preprint arXiv:1406.3915*, 2014.

[5] H. Ze, A. Senior, and M. Schuster, “Statistical Parametric Speech Synthesis using Deep Neural Networks,” in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2013, pp. 7962–7966.

[6] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury et al., “Deep Neural Networks for Acoustic Modeling in Speech Recognition,” *IEEE Signal Processing Magazine*, vol. 29, 2012.

[7] Z.-H. Ling, S.-Y. Kang, H. Zen, A. Senior, M. Schuster, X.-J. Qian, H. M. Meng, and L. Deng, “Deep Learning for Acoustic Modeling in Parametric Speech Generation: A Systematic Review of Existing Techniques and Future Trends,” *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 35–52, 2015.

[8] T. Weijters and J. Thole, “Speech Synthesis with Artificial Neural Networks,” in *IEEE International Conference on Neural Networks*, IEEE, 1993, pp. 1764–1769.

[9] O. Watts, G. E. Henter, T. Merritt, Z. Wu, and S. King, “From HMMs to DNNs: Where do the Improvements Come From?” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2016, pp. 5505–5509.

[10] Y. Qian, Y. Fan, W. Hu, and F. K. Soong, “On the Training Aspects of Deep Neural Network (DNN) for Parametric TTS Synthesis,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2014, pp. 3829–3833.

[11] Y. Fan, Y. Qian, F.-L. Xie, and F. K. Soong, “TTS Synthesis with Bidirectional LSTM Based Recurrent Neural Networks,” in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[12] A. Gutkin, L. Ha, M. Jansche, K. Pipatsrisawat, and R. Sproat, “TTS for Low Resource Languages: A Bangla Synthesizer,” in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, 2016, pp. 2005–2010.

[13] “Google International Language Resources,” accessed: July 28, 2019. \[Online]. Available: https://github.com/google/language-resources/blob/master/bn/festvox/phonology.json

[14] B. Potard, M. P. Aylett, D. A. Baude, and P. Motlicek, “Idlak Tangle: An Open Source Kaldi Based Parametric Speech Synthesiser Based on DNN,” in *INTERSPEECH*, 2016, pp. 2293–2297.

[15] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz et al., “The Kaldi Speech Recognition Toolkit,” in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, no. CONF, IEEE Signal Processing Society, 2011.

[16] Z. Wu, O. Watts, and S. King, “Merlin: An Open Source Neural Network Speech Synthesis System,” in *SSW*, 2016, pp. 202–207.

[17] “Ossian: A Simple Language Independent Text to Speech Front-end,” accessed: July 28, 2019. \[Online]. Available: https://github.com/CSTR-Edinburgh/Ossian

[18] 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*, vol. 99, no. 7, pp. 1877–1884, 2016.

[19] S. Ö. Arik, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, A. Ng, J. Raiman et al., “Deep Voice: Real-Time Neural Text-to-Speech,” in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, JMLR.org, 2017, pp. 195–204.

[20] A. Gibiansky, S. Arik, G. Diamos, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, “Deep Voice 2: Multi-Speaker Neural Text-to-Speech,” in *Advances in Neural Information Processing Systems*, 2017, pp. 2962–2970.

[21] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, “Deep Voice 3: Scaling Text-to-Speech with Convolutional Sequence Learning,” *arXiv preprint arXiv:1710.07654*, 2017.

[22] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. Courville, and Y. Bengio, “Char2Wav: End-to-End Speech Synthesis,” 2017.

[23] 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.

[24] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, “WaveNet: A Generative Model for Raw Audio,” *arXiv preprint arXiv:1609.03499*, 2016.

[25] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio et al., “Tacotron: Towards End-to-End Speech Synthesis,” *arXiv preprint arXiv:1703.10135*, 2017.

[26] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-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.

[27] F. Alam, S. M. Habib, D. A. Sultana, and M. Khan, “Development of Annotated Bangla Speech Corpora,” in *Spoken Languages Technologies for Under-Resourced Languages*, 2010.