Journal of Multidisciplinary Engineering Science Studies (JMESS)
ISSN: 2458-925X
Vol. 11 Issue 5, May - 2025

Expressive MIDI-to-Singing Synthesis

Based On Melodic Accents And Harmony

Hung-Che Shen
Department of New Media Design
I-Shou University
Kaohsiung, Taiwan
e-mail: shungch@isu.edu.tw

***Abstract*—Expressive singing voice synthesis (SVS) seeks human-like control over timing, dynamics, and pitch. Current MIDI-to-singing systems (e.g., UTAU, Synthesizer V) often produce mechanical outputs due to limited expressiveness. We propose a rule-based framework that automates melodic accent modeling via pitch interval analysis and two-part harmony generation using triadic chords in UTAU. This lightweight framework significantly improves naturalness (mean score: 5.8 vs. 3.2 baseline) and emotional impact in both solo and duet synthesis, with potential for multi-voice extensions. Listening tests validate its effectiveness, offering a scalable, resource-efficient alternative to neural-based SVS for engineering applications.**

Keywords—Expressive SVS; MIDI-to-Singing; Melodic Accent; Two-Part Harmony

I. INTRODUCTION

Singing voice synthesis (SVS) produces realistic vocal performances from digital inputs, with precise control over pitch, timing, and dynamics. MIDI-to-singing synthesis, a subset of SVS, converts MIDI data (notes, velocity, duration) into vocals using tools like UTAU and Synthesizer V. However, these systems often prioritize pitch and timing accuracy over expressive phrasing, yielding mechanical outputs. Key limitations include the lack of perceptual melodic accent models and reliance on manual harmonization for ensemble synthesis.

We propose a rule-based framework for UTAU that automates melodic accent modeling and two-part harmony generation (melody and lower voice) using pitch intervals and triadic chords. This lightweight approach enhances naturalness and emotional impact in solo and duet synthesis. The method also shows potential for scalable extension to full multi-voice (e.g., SATB) arrangements. A demonstration is available online [1]

The paper is organized as follows: Section 2 reviews related work, Section 3 details the framework, Section 4 describes experiments, Section 5 discusses results, and Section 6 concludes with contributions and future directions.

II. RELATED WORK

Singing voice synthesis (SVS) has evolved significantly from early rule-based systems to contemporary neural-network-driven models. Early systems such as CHATR and MBROLA utilized unit selection or diphone concatenation methods, producing intelligible but monotonous singing voices [2]. These approaches laid the groundwork for later statistical parametric models such as HMM-based SVS (e.g., VOCALOID2), which introduced more flexible pitch and timing control but still lacked expressiveness comparable to human singing [3].

With the introduction of user-friendly platforms like UTAU, Synthesizer V, and NEUTRINO, the accessibility and quality of SVS systems improved considerably. These tools typically transform MIDI and phoneme inputs into synthesized vocal outputs using pre-trained voicebanks [4]. Although such systems can produce high-fidelity vocals, their expressiveness often relies on user-defined parameter curves (e.g., pitch bends, vibrato depth, and volume envelopes), making it labor-intensive to achieve natural phrasing and emotional nuance [5].

To address this, research in expressive SVS has expanded toward modeling human-like performance features. For example, Kim et al. [6] proposed a deep learning-based system to learn expressive timing and dynamics from real singing data, while Zhang et al. [7] applied variational autoencoders to capture different emotional singing styles. These systems represent substantial progress but generally focus on solo voice performance and lack comprehensive modeling of melodic accents, which are critical in shaping musical phrasing.

Melodic accents refer to the perceptual prominence of notes within a melodic line, influenced by rhythmic position, pitch contour, and dynamic emphasis [8]. Despite their importance in musical expression, few SVS models explicitly incorporate melodic accent features. Instead, most prioritize pitch and timing accuracy, often resulting in flat or mechanical-sounding outputs. Recent studies in related domains (e.g., expressive piano or violin synthesis) have shown that incorporating accent structures can greatly improve naturalness and musicality [9], suggesting a similar benefit for singing synthesis.

Harmonic synthesis in SVS, particularly through structured multi-voice modeling such as SATB arrangement, remains underexplored, with most existing tools limited to monophonic outputs or

requiring labor-intensive manual layering. While choral synthesis has a long history in speech synthesis—such as SATB voice modeling using rule-based systems [10]—modern SVS platforms are predominantly designed for monophonic vocal lines. Some efforts, such as Wu et al. [11], explore multi-singer neural vocoders or layered synthesis to simulate choir effects, but these methods often require extensive training data and high computational cost. Current MIDI-to-singing workflows for choral singing typically involve manually rendering each voice (Soprano, Alto, Tenor, Bass) and combining them in post-processing, often without precise attention to harmonic interaction.

Thus, integrating melodic accent modeling with harmonic blending in a MIDI-to-singing pipeline presents an opportunity to significantly enhance expressiveness. The proposed method in this study builds on previous SVS research by explicitly representing melodic accents as perceptual features and synthesizing individual SATB parts to simulate choral harmony. This dual emphasis aims to bridge the gap between mechanical MIDI reproduction and emotionally rich vocal performance.

III. Proposed Framework

To enhance expressiveness in MIDI-to-singing synthesis, our proposed framework integrates melodic accent modeling with harmony-based vocal part generation. The system operates in two stages: (1) computing melodic accent intensity for each note through sequential MIDI analysis, and (2) generating harmonized two-part voices (high and low) using triadic chord structures.

A. Melodic Accent Estimation and UST Flag Mapping

In UTAU, UST (UTAU Sequence Text) files encode lyrics, pitch, and expressive parameters via flags. We propose an automated method to model melodic accents—perceptual note emphases—using pitch intervals, informed by music perception research [8]. Let:

- P_i be the pitch (MIDI note number) of note i.
- P_{i-1} be the pitch of the previous note
- $\Delta_i = |P_i - P_{i-1}|$ be the pitch interval

Melodic accent intensity is quantified via pitch interval analysis, grounded in perceptual studies [8]. For a note sequence {Pi }, the accent level Ai is classified as:

- Low: (Ai=0): Δi < 3 semitones (subtle emphasis, e.g., stepwise motion)
- Medium: (Ai=1):3 ≤ Δi ≤ 5 semitones (moderate emphasis, e.g., minor thirds)
- High: (Ai=2): Δi > 5 semitones (strong emphasis, e.g., leaps)

Where Δi = | Pi− Pi-1 | This value is mapped to UTAU flags through a deterministic function f(Ai) in Table I.

TABLE I. Melodic Accents with Flags Mapping

Accent Level	Δ (Pitch Interval)	UTAU Flags |
Low	<3 semitones	g+0, t0, Y0, H0, B0
Medium	3–5 semitones	g+5, t20, Y10, H10, B10
High	> 5 semitones	g+10, t40, Y20, H20, B20
Flags adjust: g (gender shift), t (breathiness), Y (vocal edge), H (high-frequency boost), and B (brightness), enabling automated, musically coherent accentuation from MIDI input.		

```
[#VERSION]
UST Version1.2
[#SETTING]
Tempo=128.00
Tracks=1
ProjectName=C:\UTAU\Tiger.UST
VoiceDir=%VOICE%Xia_Voice_Bank_U301_Bundle
OutFile=C:\Sing\Tiger.wav
CacheDir=C:\Sing\Tiger.cache
Tool1=C:\UTAU\moresampler.exe
Tool2=C:\UTAU\moresampler.exe
Mode2=True
[#0000]
Length=480
Lyric=liang
NoteNum=65
Velocity=0
Intensity=85
Flags=Mt-10Mb2Mo0Mr0MC-10          <= Melodic Accent Flag
Envelope=44,45,20,186,100,100,40
PBW=96,96,96,96,96
PBS=1;1.3
PBY=-0.6,-3.0,-12.0,-2.0,0.0
PBM=,r,,,
[#0001]
Length=480
Lyric=zhi
NoteNum=67
Intensity=95
Flags=Mt-6Mb1Mo-10Mr-10MC-6        <= Melodic Accent Flag
Envelope=44,45,20,186,100,100,40
```

Fig. 1. *Example of UST file editing showing melodic accent flag mappings added.*

Fig.1. is the example of UST file editing showing melodic accent flag mappings and added harmony track. The UST file is a plain-text format representing time-aligned note sequences and voice parameters. By editing pitch, timing, and UTAU flag values directly, we implement both melodic accent emphasis and two-part harmony within the UTAU environment.

B. Melody Harmonization with Triadic Chords

Melody harmonization generates a lower voice to complement a melody, enhancing musical depth. Our framework produces two-part harmony (melody and lower voice) using triadic chords for UTAU synthesis.

1) Harmonization Strategy

The algorithm applies music-theoretic principles:

a) Chord Selection: For each melody note, choose a diatonic triad (I, ii, iii, IV, V, vi, vii°) where the note is the root, third, or fifth, using a scoring function to prioritize tonal continuity based on the previous chord.

b) Lower Voice: Choose a chord tone (third or fifth) 3–7 semitones below the melody, ensuring singability (within vocal range) and avoiding voice crossing.

c) Progression Logic: Harmonize locally per note, enforcing functional progressions (e.g., I → IV → V → I). For example, in C major, a melody sequence [C4, E4, G4] is harmonized with triads [C: C-E-G, C: C-E-G, G: G-B-D], yielding lower voice notes [G3, C4, B3].

2) Integration with UTAU

The melody and harmony are exported as separate UST tracks. Timbral contrast is achieved using distinct voicebanks or flags (e.g., g+20 for the lower voice). This rule-based method ensures automated, coherent two-part synthesis. The harmonization algorithm pseudocode is shown in Fig, 2.

```
Input: MIDI melody (notes: [n1, n2, ..., nN], key: K)
Output: Harmony voice (notes: [h1, h2, ..., hN])
for each note ni in melody:
    chords = find_diatonic_triads(K, ni)  // List triads containing ni
    chord = select_chord(chords, prev_chord)  // Score for continuity
    hi = select_lower_tone(chord, ni, 3-7 semitones)  // Choose third or fifth
    if hi is singable and below ni:
        append hi to harmony
return harmony
```

Fig. 2. *The Algorithm of two-part Harmonization.*

IV. Experiments and Evaluation

To evaluate the effectiveness of our expressive MIDI-to-singing framework, we conducted a comparative listening test using synthesized singing samples generated under three conditions: (1) Baseline (B1): MIDI-to-UTAU synthesis without accent flags or harmonization, (2) Accent Only (B2): MIDI-to-UTAU synthesis with accent-based flag mapping, and (3) Accent + Harmony: MIDI-to-UTAU synthesis with both accent-based flag mapping and two-part harmony

A. Dataset and Setup

We selected five well-known folk and classical melodies (e.g., “Scarborough Fair,” “Greensleeves,” “Ave Maria”) as the input MIDI data. Each MIDI file included a monophonic melody line with lyrics annotated. For harmony synthesis, we applied our triadic harmonization algorithm in the key of the original piece.

We used a standard Japanese VCV voicebank in UTAU for all experiments and applied identical rendering settings across conditions, aside from expressive flag differences. Synthesized audio was exported as WAV files for listening tests.

B. Subjective Listening Test

We recruited 15 participants with backgrounds in music or audio production. Each participant listened to a randomized set of 15 clips (5 songs × 3 versions) through studio headphones. They were asked to rate each clip on the following criteria using a 7-point Likert scale: (1) Naturalness (Does it sound like a human singer?), (2) Musicality (How expressive and musically convincing is the performance?), and (3) Emotional Impact (Does the performance convey feeling?).

C. Results

The listening test results demonstrated significant improvements across all evaluated metrics when melodic accent modeling and harmony generation were applied. As shown in Table II, the combined approach (Accent + Harmony) achieved the highest scores in naturalness (5.8), musicality (5.9), and emotional impact (5.6), outperforming both the baseline and accent-only conditions. The standard deviations and confidence intervals further confirmed the consistency of these preferences among participants. The listening test results (mean scores with standard deviations and 95% confidence intervals) are shown below:

TABLE II. Subjective Listening Test

Method	Naturalness (SD, 95% CI)	Musicality (SD, 95% CI)	Emotional Impact (SD, 95% CI)
Baseline (B1)	3.2(0.8,[2.9, 3.5])	3.0(0.7,[2.7, 3.3])	2.8 (0.9, [2.4, 3.2])
Accent Only (B2)	4.5(0.6,[4.3, 4.7])	4.8(0.5,[4.6, 5.0])	4.4 (0.6, [4.2, 4.6])
Accent + Harmony	5.8(0.5,[5.6, 6.0])	5.9(0.4,[5.7, 6.1])	5.6 (0.5, [5.4, 5.8])

Statistical analysis using one-way ANOVA confirmed that differences in ratings between conditions were statistically significant and not due to chance ($F(2, 126) = 45.3$, $p < 0.001$), with a large effect size ($\eta^2 = 0.42$), meaning the type of synthesis had a strong impact on how participants rated the samples. Further analysis using Tukey’s HSD test showed that: The Accent + Harmony version was significantly better than the Baseline ($p < 0.001$), with a very large effect size (Cohen’s $d = 1.8$). It was also better than Accent Only ($p < 0.01$), with a large effect size (Cohen’s $d = 0.9$). This means that listeners clearly noticed and preferred the expressive features added through both accent mapping and harmony generation. In summary, adding melodic accents alone helped improve performance quality, but adding both accents and harmonies made the result sound significantly more natural, musical, and emotionally engaging.

V. Discussion

The experimental findings confirm that the proposed rule-based framework effectively enhances MIDI-to-singing synthesis by automating melodic accent mapping and two-part harmony generation. The pitch interval-based accent system introduces perceptual emphasis without complex modeling, while triadic harmonization adds musical depth, improving naturalness and emotional impact. This lightweight approach requires minimal computational resources, making it suitable for real-time music production on standard hardware.

One limitation is the reliance on static diatonic key frameworks, which may limit applicability to modulating, chromatic, or non-Western scales unless expanded via adaptive key inference or corpus-driven chord selection. Additionally, while UTAU offers extensive timbral control via flags, the exact effect can vary across voicebanks, making results somewhat voicebank-dependent.
Future improvements could include:

- Incorporating dynamic or phrasing control (e.g., crescendo, legato) alongside accents.
- Extending harmonization to full SATB or more advanced AI-based harmonic modeling.
- Integrating emotion classifiers or sentiment tags for automatic mood adaptation.

VI. Conclusion

This paper presents a rule-based MIDI-to-singing synthesis framework for UTAU, automating melodic accent modeling and two-part harmony generation. By mapping pitch intervals to UTAU flags and generating triadic harmonies, our method significantly improves naturalness, musicality, and emotional impact, as validated by listening tests.

The framework's lightweight design enables real-time synthesis on resource-constrained systems, offering a practical alternative to neural-based SVS for music production and multimedia engineering. Future work will integrate neural prosody modeling for dynamic phrasing, adaptive key detection for complex melodies, and sentiment-driven synthesis for expressive app

References

[1] Shen, H.-C. (2025). Expressive MIDI-to-singing synthesis demonstration. MIDI-to-Singing Technical Blog. https://shungch.blogspot.com/p/mandarin.html

[2] Hunt, A. J., & Black, A. W. (1996). Unit selection in a concatenative speech synthesis system using a large speech database. In IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) (Vol. 1, pp. 373–376). IEEE.

[3] Tokuda, K., Yoshimura, T., Masuko, T., Kobayashi, T., & Kitamura, T. (2000). Speech parameter generation algorithms for HMM-based speech synthesis. In IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) (Vol. 3, pp. 1315–1318). IEEE.

[4] Amatrian, Y. (2020). Synthesizer V: Towards a practical neural singing synthesizer. Dreamtonics Technical Report.

[5] Kenmochi, H., & Ohshita, H. (2007). VOCALOID: Commercial singing synthesizer based on sample concatenation. In Interspeech 2007 (pp. 4011–4010).

[6] Kim, H., Lee, J., & Kim, K. (2018). Neural network-based expressive singing synthesis model trained on musical scores. In Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR) (pp. 595–602).

[7] Zhang, X., Blaauw, M., Bonada, J., & Serra, X. (2020). Variational inference for singing voice synthesis with expressive performance modeling. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 28, 2530–2543.

[8] Palmer, C. (1996). Anatomy of a performance: Sources of musical expression. Music Perception: An Interdisciplinary Journal, 13(3), 433–453.

[9] Sakar, M. S., Shalev-Shwartz, S., & Shashua, A. (2021). Learning expressive piano performance using note-level accents. arXiv preprint arXiv:2108.01630.

[10] Dutoit, T., Pagel, V., Pierret, N., Bataille, F., & van der Vreken, O. (1996). The MBROLA project: Towards a set of high-quality speech synthesizers free of use for non-commercial purposes. In Proceedings of the Fourth International Conference on Spoken Language Processing (ICSLP) (Vol. 3, pp. 1393–1396).

[11] Wu, Z., Valentini-Botinhao, C., Watts, O., & King, S. (2022). Polyphonic singing synthesis using multi-singer neural vocoders. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 30, 2572–2584.