Journal of Multidisciplinary Engineering Science Studies (JMESS)
ISSN: 2458-925X
Vol. 12 Issue 1, January - 2026

A Text Based MIDI Disassembly Approach to AI Melody Regeneration in Singing Synthesis Education

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

***Abstract*—In singing synthesis education, generating diverse melodic variants for instructional purposes remains a labor-intensive and time-consuming manual process. Traditional workflows often necessitate the complete recompositing of melodic lines to preserve harmony and rhythm, significantly limiting the scalable reuse of high-quality accompaniment materials. To address this challenge, this study proposes a lightweight, text-based MIDI disassembly framework for controlled AI-assisted melody regeneration. By converting standard multi-track MIDI files into human-readable text, the approach enables Large Language Models (LLMs)—including ChatGPT, Grok, and Gemini—to regenerate specific melody channels under explicit symbolic constraints while leaving the original accompaniment untouched. Evaluation results demonstrate that the proposed method ensures 100% structural integrity and timing consistency across models, while providing distinct melodic variations suitable for different pedagogical levels. To support reproducibility, a comprehensive Standard Operating Procedure (SOP) and a toolset are provided via a public GitHub repository. This research offers a practical and scalable framework for integrating generative AI into music education, shifting the instructor's role from manual editor to high-level pedagogical reviewer.**

Keywords—Text-Based MIDI; MIDI Disassembly; AI Melody; Singing Synthesis; Education

I. INTRODUCTION

Singing synthesis has become an important support tool in music education, enabling repeatable demonstrations of melody, vocal range, and stylistic variation [1][2]. In practical teaching environments, instructors frequently rely on commercially available accompaniment generation systems, such as Band-in-a-Box [3], to produce musically coherent songs suitable for singing practice. While these tools efficiently generate high-quality accompaniments, creating multiple melodic variants for instructional purposes remains largely manual and time-consuming.

In typical classroom workflows, modifying the melody of an existing song often requires recomposing the entire melodic line while preserving harmony, rhythm, and tempo. This process limits the scalable reuse of well-designed accompaniment materials and restricts instructors to a small number of melodic variants per song [4]. As a result, the pedagogical potential of high-quality MIDI-based resources is underutilized.

Recent advances in large language models (LLMs) have shown promising capabilities in symbolic music generation. However, most AI-based music systems emphasize end-to-end composition, generating complete songs from scratch. Such approaches offer limited structural control and frequently alter accompaniment, timing, or instrumentation—properties that must remain stable in educational settings [5].

This study addresses this gap by proposing a text-based MIDI disassembly workflow for controlled AI-assisted melody regeneration. Instead of generating full compositions, the proposed method operates directly on existing MIDI songs. A standard MIDI file is disassembled into a human-readable text format, the melody is predefined as a fixed MIDI channel, and an LLM is instructed to regenerate only that channel under strict symbolic constraints. The modified text is then reassembled into a valid MIDI file, guaranteeing that all non-melodic tracks remain unchanged.

By constraining AI intervention to a single melody channel, the proposed approach balances creative variation with deterministic structural preservation. The method is designed for practical classroom use, where instructors act as listeners and reviewers rather than full-time editors. To facilitate transparency and reproducibility, the complete workflow, tools, and demonstration cases are released as an open research repository. This paper evaluates the effectiveness of the approach using multiple LLMs and demonstrates its applicability through publicly accessible MIDI melody variation examples.

The remainder of this paper is organized as follows. Section II reviews related work on text-based MIDI disassembly, symbolic music generation, and AI-assisted melody variation. Section III describes the proposed MIDI-TEXT-based workflow in detail, including the toolchain, step-by-step operating procedure, and the parameterized prompt design for controlled melody regeneration. Section IV presents the evaluation methodology, combining objective MIDI-level analysis with instructor-guided auditory assessment, supported by publicly released materials in a GitHub repository. Section V discusses experimental results and pedagogical implications, with particular emphasis on structural reliability and singability. Finally, Section VI concludes the paper and outlines directions for future research.

II. Related Work

A. MIDI Disassembly for Symbolic Control

Text-based MIDI disassembly has long been used for precise symbolic inspection and modification of MIDI files. A representative example is the MIDI File Disassembler/Assembler developed by Jeff Glatt [7], which converts binary MIDI files into an assembly-style text format and enables lossless reconstruction after editing. Although originally intended for low-level MIDI debugging and batch processing, this approach provides exact event-level control over pitch, timing, and channel data.

Archived versions of Jeff Glatt's MIDI utilities and documentation remain accessible via the Internet Archive, including required dependencies and usage examples. Compared with higher-level MIDI libraries, disassembly-based workflows preserve complete symbolic fidelity and are well suited for deterministic transformations, making them appropriate for controlled educational applications.

B. AI-Based Symbolic Melody Generation

Symbolic music generation has progressed from rule-based systems [8] to deep learning and transformer-based models [9]. Recent LLMs have demonstrated the ability to generate symbolic music representations under textual prompts [10]. However, most systems focus on end-to-end generation and offer limited guarantees regarding structural preservation, particularly when modifying existing compositions.

For educational applications, uncontrolled generation is undesirable. This study differs from prior work by employing text-based MIDI disassembly as an explicit interface layer, allowing LLMs to operate under strict channel-level constraints. Rather than replacing traditional MIDI tools, the proposed method integrates LLM-based variation into a deterministic symbolic workflow supported by publicly available artifacts.

III. Method

A. Toolchain and Prerequisites

The proposed method employs a text-based MIDI disassembly and reassembly workflow using the MIDI File Disassembler/Assembler originally developed by Jeff Glatt. This Windows-based utility provides a deterministic and lossless mechanism for converting standard MIDI files (.mid) into a human-readable text representation (.txt), and subsequently reconstructing a valid MIDI file from the edited text.

Unlike high-level MIDI processing libraries, this tool exposes event-level symbolic data, including note-on/note-off events, delta-time values, tempo changes, and channel assignments. Such transparency is essential for educational use cases in singing synthesis, where fine-grained inspection and controlled modification of melodic structure are required. In this study, the tool is used exclusively for channel-constrained symbolic editing, ensuring that only the melody track is subject to AI-assisted regeneration while all accompaniment data remain intact.

The original distribution site of the MIDI File Disassembler/Assembler is no longer active; however, archived versions of the software and official documentation remain accessible via the Internet Archive [7]. To execute the tool correctly, two supporting dynamic link libraries (MidiFile DLL and GenMidi DLL) must be installed.

To support reproducibility, the complete toolchain, including archived executables, documentation, and usage instructions, is organized and redistributed for educational use in a public GitHub repository via the Internet Archive:

https://web.archive.org/web/20080129082157/http://www.borg.com/~jglatt/progs/software.htm [11]

B. Overview of the Workflow

The proposed workflow enables AI-assisted melody regeneration without disturbing the original accompaniment. Instead of generating music end-to-end, the method operates directly on an existing multi-track MIDI file and restricts symbolic modification to a predefined melody channel.

The process consists of three explicit operations: MIDI disassembly, text-based editing, and MIDI reassembly. This separation between conversion, editing, and reconstruction ensures pedagogical clarity, deterministic behavior, and reproducibility across different AI models and instructional contexts.

The three explicit user operations, each corresponding directly to a button in the MIDI File Disassembler/Assembler interface (Fig. 1). This explicit separation between conversion, editing, and reconstruction ensures pedagogical clarity and reproducibility.

From an engineering perspective, this work emphasizes process transparency and operational reproducibility rather than aesthetic optimization. By explicitly formalizing melody regeneration as a constrained symbolic transformation, the proposed approach aligns with instructional workflows where predictability and repeatability are essential. This design choice ensures that regenerated materials can be reliably reused across different teaching scenarios and AI models.

Furthermore, the proposed framework allows instructors to focus on pedagogical decision-making instead of low-level MIDI editing. The combination of deterministic symbolic constraints and AI-assisted variation provides a practical balance between creative flexibility and instructional control, which is particularly valuable in classroom-oriented singing synthesis applications.

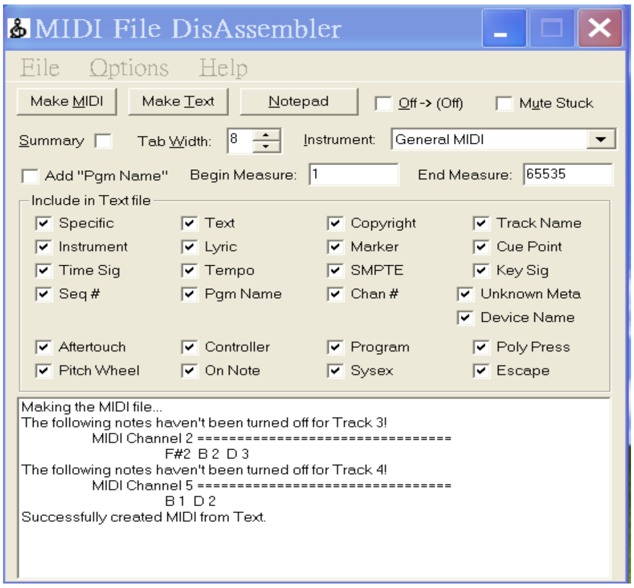

Fig. 1. Interface of MIDI File Disassembler/Assembler tool

C. Step-by-Step Operation Procedure

Step 1: MIDI Disassembly via [Make Text]

The process begins by converting a standard multi-track MIDI file into its textual representation.

- The user clicks the [Make Text] button in the interface.
- A file selection dialog appears, allowing the user to choose a .mid file.
- Upon confirmation, the tool disassembles the selected MIDI file and generates a corresponding text file (.txt) with the same base filename.

This disassembly process is lossless, preserving all event-level information such as note pitches, durations, channel numbers, tempo events, and system messages. The resulting MIDI-TEXT file serves as a complete symbolic description of the original MIDI content.

Step 2: Text Inspection and Editing via [Notepad]

After the text file has been generated, the user clicks the [Notepad] button.

- This action opens the previously generated MIDI-TEXT (.txt) file in a standard text editor (e.g., Windows Notepad).
- At this stage, users may manually inspect or edit the symbolic data directly.

In the proposed method, only the predefined melody channel is subject to modification. AI-generated melodic content can be inserted or substituted at this stage, while all other channels (e.g., accompaniment, bass, percussion) remain unchanged. This constraint ensures that structural, harmonic, and rhythmic elements of the original song are fully preserved. Fig. 2 illustrates this disassembly process, showing the transformation from binary MIDI to editable textual representation..

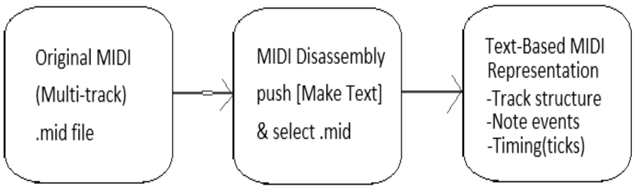

Fig. 2. MIDI Disassembly and Text Representation Workflow

Step 3: MIDI Reassembly via [Make MIDI]

After completing all edits, the modified text file is converted back into a standard MIDI file.

- The user clicks the [Make MIDI] button.
- A file selection dialog prompts the user to select the edited .txt file.
- The assembler reconstructs a new .mid file based on the symbolic instructions in the text.

During this reassembly process, all unmodified events are preserved exactly. Since AI intervention is limited to the melody channel, the resulting MIDI file retains identical accompaniment tracks, timing, and system-level events compared to the original input.

D. LLM-Based Melody Regeneration

Large language models (LLMs) are employed to perform controlled melody regeneration directly on the text-based MIDI representation (MIDI-TEXT). Rather than generating music freely, the model is instructed to operate under strict symbolic constraints, modifying only the melodic content assigned to a predefined MIDI channel while preserving all other musical information. In this study, the lead melody is fixed to MIDI Channel 1, and the complete MIDI-TEXT file is provided as input. The LLM is required to output a fully formatted MIDI-TEXT file that can be directly reassembled without post-processing.

Prompt Specification: The following prompt template is used consistently across all evaluated models.

Prompt:
You are given a complete MIDI file in text form (MIDI-TEXT).
Regenerate only the melody assigned to Channel 1.
Do not modify note onsets, durations, tempo, or any events on other channels.
Restrict pitches to MIDI notes 60–72 and avoid leaps larger than a perfect fifth.
Output the complete modified MIDI-TEXT with identical formatting.

This prompt explicitly constrains the scope of AI intervention at the symbolic level. By prohibiting changes to timing, tempo, and non-melodic channels, the prompt ensures deterministic preservation of accompaniment and rhythmic structure. Formatting consistency is enforced to guarantee successful MIDI reassembly.

Parameterized Prompt Design: To support reproducibility, extensibility, and strict symbolic control, the melody regeneration prompt is formulated in a parameterized form, allowing musical constraints to be explicitly defined and systematically adjusted without modifying the underlying workflow. Rather than encoding musical intent as implicit stylistic instructions, the proposed design translates pedagogical and musical requirements into explicit symbolic parameters.

TABLE I. PARAMETERIZED PROMPT STRUCTURE FOR MELODY REGENERATION

Parameter	**Description**	**Example Value**
TargetChannel	MIDI channel designated for melody regeneration	Channel 1
PitchMin	Minimum allowed MIDI pitch	60 (C4)
PitchMax	Maximum allowed MIDI pitch	72 (C5)
MaxInterval	Maximum allowed melodic leap	Perfect fifth
TimingEditable	Whether note timing may be modified	No (Fixed)
DurationEditable	Whether note durations may be modified	No (Fixed)
OtherChannelsEditable	Whether non-melody channels may be modified	No (Fixed)
OutputFormat	Required output format	Complete MIDI-TEXT

Table I summarizes the generalized parameter set used in this study. These parameters define clear operational boundaries for LLM-based melody regeneration, transforming abstract musical concepts—such as vocal range, melodic smoothness, and structural stability—into enforceable logical constraints.

IV. EVALUATION

A. Evaluation Design

The evaluation focuses on verifying whether the proposed text-based MIDI disassembly workflow enables structurally valid, musically coherent, and pedagogically usable melody regeneration. Rather than assessing creative originality, the primary goal is to confirm that AI-generated melodies satisfy deterministic constraints required for singing synthesis education. Evaluation materials, regenerated MIDI files, and corresponding MIDI-TEXT representations are publicly available in the accompanying GitHub repository [11], enabling independent verification.

B. Objective MIDI-Level Metrics

Objective evaluation is performed directly on regenerated MIDI files after reassembly. The following symbolic criteria are examined to ensure technical compliance:

- Pitch range compliance: Verification that all notes fall within the specified PitchMin and PitchMax parameters.
- Interval distribution: Assessment of melodic contours, focusing on the ratio of stepwise motion versus large leaps defined by MaxInterval.
- Rhythmic invariance: Confirmation of identical note onsets and durations compared to the original file, ensuring accompaniment alignment.
- Structural validity: Successful reconstruction into standard MIDI format and preservation of all non-melodic accompaniment tracks.

Across all test cases and models, regenerated MIDI files satisfied reassembly requirements and preserved tick-level alignment with the original accompaniment. No rhythmic drift or channel corruption was observed, confirming that the parameterized prompt constraints effectively enforce structural invariance.

C. Instructor-Guided Auditory Assessment

Beyond symbolic metrics, an instructor-guided auditory assessment evaluates the outputs of the regenerated files. The focus is on singability—the ease of vocal reproduction in coordination with the unchanged accompaniment. While minor pitch refinements are permitted to align with realistic pedagogical workflows, the core evaluation determines if the melody is accessible to students without excessive vocal strain.

Rather than scoring musical aesthetics, the evaluation assesses whether the melody can be comfortably sung by students without excessive technical difficulty. Minor edits are permitted only when necessary and are limited to pitch refinement, reflecting realistic pedagogical workflows.

V. RESULTS AND DISCUSSION

A. Structural Reliability of AI-Regenerated Melodies

All regenerated melodies across the tested LLMs were structurally valid and successfully reconstructed into standard MIDI files. Because AI intervention was strictly limited to the predefined melody channel, elements such as harmony, tempo, accompaniment texture, and overall musical form remained 100% identical to the original songs.

This confirms that text-based MIDI disassembly functions as an effective "symbolic boundary," allowing AI creativity to operate without compromising deterministic musical structure. From an engineering perspective, the workflow consistently produced deployable MIDI outputs suitable for downstream singing synthesis systems.

B. Singability and Pedagogical Usability

In this study, singability is operationally defined as melodies that:

- Remain within a moderate vocal range,
- Favor stepwise or small-interval motion,
- Avoid abrupt or extreme melodic leaps,
- Align naturally with the existing harmonic context.

The tested LLMs exhibited distinct characteristics in their melodic output within the constrained framework:

- ChatGPT (GPT-4o): Favored conservative, stepwise motion, making it highly suitable for introductory singing demonstrations.
- Gemini (1.5 Pro): Exhibited greater expressive variation while remaining largely singable, appropriate for intermediate-level instruction.
- Grok: Produced the widest melodic diversity; however, some outputs required minor instructor refinement to meet strict classroom singability expectations.

These findings confirm that explicit symbolic constraints, particularly pitch range and maximum interval limits, play a critical role in transforming AI-generated melodies from abstract musical ideas into vocally practical teaching materials.

VI. Conclusion

This study successfully establishes a deterministic, engineering-oriented framework for AI-assisted melody regeneration through text-based MIDI disassembly. By isolating AI intervention to a single, predefined melody channel, the proposed method effectively bridges the gap between AI creativity and the structural predictability required in educational settings. Our findings confirm that this lightweight workflow preserves accompaniment, timing, and harmonic integrity while enabling scalable generation of pedagogical materials with minimal manual effort.

Furthermore, the release of a complete, publicly accessible SOP—including open-source tools and demonstration cases—positions this work as a transparent and reproducible alternative to opaque end-to-end AI music generation systems. By grounding generative AI in deterministic symbolic representations, this framework provides a sustainable and responsible path for the integration of emerging AI technologies into singing synthesis training and music education at large.

References

[1] Kenmochi, H., & Ohshita, H. (2007). "VOCALOID - Commercial singing synthesizer based on sample concatenation." Interspeech.

[2] Dai, J., et al. (2024). "AI-based Singing Synthesis in Music Education: A Review of Pedagogical Applications." Journal of Music, Technology & Education. (Context for singing synthesis in vocal training).

[3] PG Music Inc. (2024). Band-in-a-Box®: Intelligent Automatic Accompaniment Software. Victoria, BC: PG Music Inc.

[4] Bamberger, J. (2013). The Mind behind the Musical Ear: How Children Develop Musical Intelligence. Harvard University Press. (Discusses the need for varied pedagogical materials in music education).

[5] Agostinelli, A., et al. (2023). "MusicLM: Generating Music From Text." arXiv preprint arXiv:2301.11325. (Context for LLM-based music generation and its limitations).

[6] Ji, S., et al. (2023). "A Survey on Deep Learning for Symbolic Music Generation: Representations, Algorithms, and Evaluation." ACM Computing Surveys.

[7] Glatt, J. (2008). MIDI Utilities: MidiDsm and Related Tools. Retrieved from Internet Archive: https://web.archive.org/web/20080129082157/http://www.borg.com/~jglatt/progs/software.htm

[8] Roads, C. (1996). The Computer Music Tutorial. MIT Press.

[9] Briot, J. P., Hadjeres, G., & Pachet, F. D. (2020). Deep Learning for Music Generation. Springer.

[10] Huang, C. Z. A., Vaswani, A., Uszkoreit, J., et al. (2018). "Music Transformer: Generating Music with Long-Term Structure." Proceedings of the 7th International Conference on Learning Representations (ICLR).

[11] Shen, H.-C. (2026). MIDI-TXT-Melody-Regeneration-SOP. GitHub repository: https://github.com/shungch-code/MIDI-TXT-Melody-Regeneration-SOP.