# Automatic String Annotation of Classical Guitar Scores from Recordings

Patrick Gaft
SID: 520432082
GitHub: pgaft
https://github.com/pgaft/elec5305-project-520432082

## I. Project Overview

This project will develop a system to automatically identify which string is used for each note in a classical guitar recording and annotate a given MusicXML score with corresponding string numbers. The input is a recorded guitar performance and its known score. The system will perform audio analysis to detect note onset by its spectral flux, validate whether the detected note's frequency matches the score, classify the note's string by its inharmonicity, align the detected notes to the score and output an annotated score. The project will be implemented in Python to make use of existing libraries for the modification of the MusicXML score. The goal is to automate string number annotation, easing the friction of sight reading and avoiding the tedium of manual annotation. The project will be evaluated by its string identification accuracy.

## II. Background and Motivation

Guitars can play the same notated pitch on different strings, yielding differences in timbre and playability. A note like A3 can be played across four strings positions on a standard classical guitar. Each of these positions has distinct string length, thickness and tension, and material between bass and treble strings. Inharmonicity refers to deviations from the fundamental frequency in the overtones of the string vibration, each of these strings exhibits differing inharmonicity [1]. Many freely accessible classical guitar music scores do not specify string choices. This information is extremely valuable in sight-reading music, and provides easier inference of individual left-hand fingerings. An automated solution would assist guitarists by recovering a performer's string choices directly from audio, circumventing the time consuming process that is manual annotation. By leveraging inharmonicity as a unique characteristic of each string, we can combine other spectral analysis signal processing techniques for audio and score alignment to perform identification. Existing research attempts to capture a much larger set of information, such as plucking position, polyphonic pitch estimation and fret location among other markers for tabulature generation [1], [2], [3]. By limiting the scope to a small subset of automatic music transcription, this project provides a practical contribution without the complexity of full transcription.

## III. Proposed Methodology

- **Tools and Platforms:** Since the project aims to automate the annotation of scores, Python will be used to make use of existing libraries to modify MusicXML files [4]. MusicXML was chosen as the score's file format as it is an open-source universal file format for Western musical notation [5]. The librosa [6] Python package will be used for the audio analysis due to the strength of its onset detection algorithm [7].
- **Pre-processing:** The audio will be segmented into individual notes with the use of librosa's [6] onset detection algorithm. This determines the onset of the note by its spectral flux [6]. The fundamental frequency of the note will also be determined once it is segmented using the autocorrelation function, following the MPEG-7 guidelines [8].
- **Inharmonicity Analysis:** Each detected note will be analysed to predict which string it was played on. Each note played on a certain string and fret has a different inharmonicity depending on the string properties that offset the partials of the note from the fundamental [1]. By estimating the string's inharmonicity coefficient from the partials, we can classify the note with a string by analysing the frequency spectrum [1].
- **Score Annotation:** We first perform a validation check between the found fundamental frequency and what is described by the score. If the validation is passed we import the MusicXML file into Python and use the musicxml package [4] to annotate the notation. This file can then be exported to most notation software (MuseScore, Guitar Pro, Sibelius etc.) to be converted to PDF.
- **Evaluation:** The performance of the project will be quantitatively measured with string identification accuracy. This will either depend on a manual transcription of a video recording of the performance or a self-made recording to ensure known fingerings.
- **Data Sources:** These could include self-made recordings, manual transcriptions of video recordings, or taking sheet music with fingerings (or tabulature) of recordings and converting these into MusicXML.

## IV. Expected Outcomes

- Working prototype that takes a recording and its MusicXML score, performs note-by-note inharmonicity based string identification and annotates this information to the original score.
- String-identification accuracy on a labelled test set with error analysis.
- Short demo with audio and the score, illustrating the benefit of annotations.

## V. Timeline (Weeks 6-13)

| Week | Task |
|---|---|
| 6–8 | Literature review, select pieces and obtain their scores, implement the audio pre-processing code. |
| 9–10 | Implement the inharmonicity-based string identification. |
| 11–12 | Implement annotation to MusicXML and evaluate performance. |
| 13 | Finalise report, prepare a demo. |

## References

[1] I. Barbancho, L. Tardon, S. Sammartino, and A. Barbancho, "Inharmonicity-Based Method for the Automatic Generation of Guitar Tablature," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, pp. 1857–1868, Aug. 2012.

[2] J. Hjerrild and M. Christensen, *Estimation of Guitar String, Fret and Plucking Position Using Parametric Pitch Estimation*, Feb. 2019.

[3] Y. Zang, Y. Zhong, F. Cwitkowitz, and Z. Duan, "SynthTab: Leveraging Synthesized Data for Guitar Tablature Transcription," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2024, pp. 1286–1290, arXiv:2309.09085 [cs]. [Online]. Available: http://arxiv.org/abs/2309.09085

[4] "musicxml: generating musicxml." [Online]. Available: https://github.com/alexgorji/musicxml.git

[5] "Working with MusicXML files." [Online]. Available: https://musescore.org/en/handbook/4/working-musicxml-files

[6] B. McFee, M. McVicar, D. Faronbi, I. Roman, M. Gover, S. Balke, S. Seyfarth, A. Malek, C. Raffel, V. Lostanlen, B. v. Niekirk, D. Lee, F. Cwitkowitz, F. Zalkow, O. Nieto, D. Ellis, J. Mason, K. Lee, B. Steers, E. Halvachs, C. Thomé, F. Robert-Stöter, R. Bittner, Z. Wei, A. Weiss, E. Battenberg, K. Choi, R. Yamamoto, C. J. Carr, A. Metsai, S. Sullivan, P. Friesch, A. Krishnakumar, S. Hidaka, S. Kowalik, F. Keller, D. Mazur, A. Chabot-Leclerc, C. Hawthorne, C. Ramaprasad, M. Keum, J. Gomez, W. Monroe, V. A. Morozov, K. Eliasi, nullmightybofo, P. Biberstein, N. D. Sergin, R. Hennequin, R. Naktinis, beantowel, T. Kim, J. P. Åsen, J. Lim, A. Malins, D. Hereñú, S. v. d. Struijk, L. Nickel, J. Wu, Z. Wang, T. Gates, M. Vollrath, A. Sarroff, Xiao-Ming, A. Porter, S. Kranzler, Voodoohop, M. D. Gangi, H. Jinoz, C. Guerrero, A. Mazhar, toddrme2178, Z. Baratz, A. Kostin, X. Zhuang, C. T. Lo, P. Campr, E. Semeniuc, M. Biswal, S. Moura, P. Brossier, H. Lee, W. Pimenta, J. P. Åsen, S. Hyun, I. S, E. Rabinovich, G. Lei, J. Guo, P. S. M. Skelton, M. Pitkin, A. Mishra, S. Chaunin, BenedictSt, S. VanRavenswaay, and D. Südholt, "librosa/librosa: 0.11.0," Mar. 2025. [Online]. Available: https://doi.org/10.5281/zenodo.15006942

[7] J. Borg, "Onset Detection in Somax," 2022. [Online]. Available: http://repmus.ircam.fr/_media/merci/merci-ha2-onsetdetection-0.2.0.pdf

[8] Craig Jin, "ELEC5305 Lab - Audio Features."