# CS3642 Project Proposal

From Audio to Frets: AI-Driven Guitar Transcription and Robotic Replay

James Widner 001121770

September 14, 2025

# Summary (Executive Overview)

This project aims to create an AI system that listens to guitar audio and produces playable notes, chords, and tablature. The core phase focuses on automatic transcription from clean recordings, mapping sounds to musical notes and chords. A stretch goal is to handle "messy" audio—full songs with multiple instruments—by first separating the guitar track. A long-term extension is to connect the AI transcription output to a robotic guitar system with servo motors that can physically press frets and strum strings, enabling the machine to hear and replay performances in the real world.

## Introduction / Motivation

Learning songs by ear is rewarding but time-consuming, especially with complex voicings, distortion, and expressive guitar techniques. Automatic Music Transcription (AMT) seeks to automate this process, but guitar presents unique challenges due to polyphony, overlapping harmonics, and non-linear playing styles.

The goal of this project is to build a transcription assistant that can reliably detect chords and notes from guitar recordings and map them to tablature. Unlike preprogrammed robotic guitars that simply follow preloaded scores, this system is designed to **listen to arbitrary audio**, **transcribe it**, **and then replay it**—first virtually, and ultimately on a physical guitar controlled by servo motors. This vision combines machine learning, signal processing, and robotics, with applications in education, accessibility, and performance automation.

#### Literature Review (Preliminary Reading)

- Pitch Estimation: CREPE (CNN pitch tracker), YIN/pyin algorithms.
- Onset Detection: Spectral flux; Onsets and Frames (Hawthorne et al.).
- Chord Recognition: Chroma-based methods; deep learning on CQT spectrograms; HMM/Viterbi smoothing.
- Tablature Mapping: GuitarSet dataset with string/fret annotations.
- Source Separation: Demucs and Spleeter for isolating guitar stems from full mixes.
- Robotic Guitar Prototypes: Servo-based fret pressing and strumming systems demonstrate feasibility of physical replay.

# Proposed Methodology

#### System Pipeline:

- 1. **Preprocessing:** Normalize audio; compute spectrograms (CQT, Mel); extract chroma/CRP features. Apply harmonic–percussive source separation (HPSS).
- 2. **Guitar Isolation (stretch):** Apply Demucs or Spleeter to extract the guitar stem from multi-instrument songs.
- 3. Onset & Pitch Tracking: Detect note onsets; estimate pitch with CREPE or pyin. Smooth results for monophonic solos.

## 4. Chord Recognition:

- Baseline: Template matching on chroma features.
- AI: CNN/GRU classifier trained on spectrogram patches for chord classes (maj/min/7th).
- Postprocessing: Temporal smoothing (HMM/Viterbi).
- 5. **Tab Mapping:** Convert pitches to string/fret positions using GuitarSet constraints; optimize for playability (minimize shifts).
- 6. Replay: Export transcription as MIDI or tab.
  - Short-term: Virtual playback using guitar soundfonts.
  - Long-term: Servo-driven robot guitar that executes fret presses and strums.

(Figure placeholder: Pipeline diagram showing Audio  $\rightarrow$  Features  $\rightarrow$  Transcription  $\rightarrow$  Tabs/MIDI  $\rightarrow$  Robotic Guitar).

#### **Data Sources**

- GuitarSet: Annotated frets, strings, and F0s for supervised learning.
- IDMT-SMT-Guitar: Isolated notes and techniques.
- NSynth (Guitar subset): For pitch modeling.
- MedleyDB, MUSDB18: Multitrack recordings for separation experiments.
- **Personal recordings:** For demonstration and testing.

#### **Evaluation Plan**

- Chord Accuracy: Chord Symbol Recall (CSR), frame-level accuracy.
- Note Accuracy: Precision, recall, F1; onset accuracy; pitch error (in cents).
- Tab Mapping: String/fret correctness; playability analysis.
- Replay Quality: Informal MOS (listening test); alignment with original.

#### **Expected Outcomes**

- MVP: Chord recognition and MIDI/tab replay from clean guitar audio.
- Stretch: Robust note/chord transcription in messy audio via separation.
- Long-term: Physical replay on a robotic guitar.

## References (starter set; will expand for final report)

- 1. McFee et al., "librosa: Audio and Music Signal Analysis in Python."
- 2. Kim et al., "CREPE: A Convolutional Representation for Pitch Estimation."
- 3. Hawthorne et al., "Onsets and Frames: Dual-Objective Piano Transcription."
- 4. Schreiber & Müller, "A Single-Step Approach to Chord Transcription."
- 5. Xi et al., "GuitarSet: A Dataset for Guitar Transcription."
- 6. Défossez et al., "Demucs: Music Source Separation with CNNs."
- 7. Stoller et al., "Spleeter: A Fast and State-of-the-Art Music Source Separation Tool."
- 8. Humphrey & Bello, "From Chroma to Chords: Robust Chord Recognition."
- 9. van den Oord et al., "WaveNet: A Generative Model for Raw Audio."
- 10. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library."