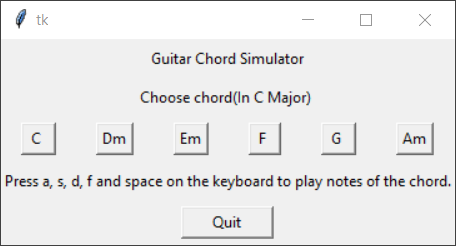
# DSP Lab Final Project Report - Guitar Simulator

Group member: Jingyuan Li(jl10915), Zongying Liu(zl3079)

## Introduction

Our guitar simulator project is designed to play guitar chords in real-time when users press the keyboard.



Users can choose different chords through the GUI. The choosing chord buttons are connected to a chord-changing function which can link 5 notes (frequencies) of the chosen chord to the keys on the keyboard(a, s, d, f, and space, from low-frequency note to high-frequency note). This function requires the text on the buttons as an input parameter.

After choosing a chord, users could press a, s, d, f, and space on the keyboard to play the notes of the chord. To simulate the tone of the guitar, the system will play the note by applying the Karplus-Strong algorithm to the original sound in real-time related to which key was pressed. There is a difference equation (filter) to implement each note as the original sound for the Karplus-Strong algorithm.

## Implementation

This project consists of two main parts. One part is note generating process using a second-order difference equation to filter the input. The other part is applying the Karplus-Strong algorithm to the note which is generated in the first step.

For the first part, we use 5 different second-order equation to filter each note separately, because for each chord there are 5 notes and we want them paly overlappingly. Each filter are using different variables which are calculated from the specific frequency. Also, all these 5 equations have its own input and output. In order to implement the overlap effect, each note is filtered with updating the status buffer which store the last two status whether the note are pressed or not. Then all these outputs of 5 notes are stored into the output buffer and become the input of the second part process. At this part, we use python library spicy and the function signal.lfilter().

For the second part, we use the output from the first part as input, and apply Karplus-Strong algorithm to the 5 signal respectively, and using 5 circular buffer to store the status which will be use at next iteration. Finally, all these 5 outputs are added and become the total output of this chord to be written to the output stream. Karplus-Strong algorithm implements in a feedback loop with three components: a lowpass filter, a delay and a gain. At this part, we choose to use a simple way to implement it, by the follow equations:

And we do not using any library, we implement it by calculating manually.