String Synthesizer Keyboard

EE-UY 4163
Huihua Guan

Introduction to Project

- Focus on synthesis of two types of string instruments
  - Plucked (*pizzicato*)
    - Guitar
  - Bowed (*arco*)
    - Double Bass
    - Cello
    - Viola
    - Violin

Significance of Project

- Contemporary music composers incorporate synthesizers in modern tunes
- General purpose polyphonic synthesizers allows a wide range of instrument sounds to be played with a single equipment
- Need for inexpensive human generated string ensembles
- Non-professional musicians or hobbyist can also participate in creating music without paying a high price for musical instruments

Music Theory

Pitch range: range of a musical instrument from the lowest to the highest pitch it can play

STRINGS				
Violin	treble	G3-A7	G3-G6	as written
Viola	alto	C3-E6	C3-C6	as written
Cello	bass, tenor, treble	C2-C6	C2-G5	as written, but in treble clef may sound 8ve lower (Beethoven)
Double Bass	bass	C2-C5, also A2, B2	E2-G4	8ve lower
Banjo	treble	C3-A4		as written (tenor sounds 8ve lower)
Guitar	treble	E3-E6		8ve lower

Digital Signal Processing Theory

Karplus Strong Algorithm

- Physical modelling synthesis method
  - Simplest class of wavetable-modification algorithms (digital waveguide synthesis)
  - Loops a short waveform through a filtered delay line to simulate the sound of a plucked string
    - Can also be used to generate drum sound

Digital Signal Processing Theory

Karplus Strong Algorithm:

- Excitation of white noise is generated into a delay line

```
buf = [random() - 0.5 for i in range(N)]
```

- The delay line is pass through a filter (Read value from this buffer and average with the previous value)

```
lpfOut = 0.996*0.5*(buf[1]+buf[0])
```

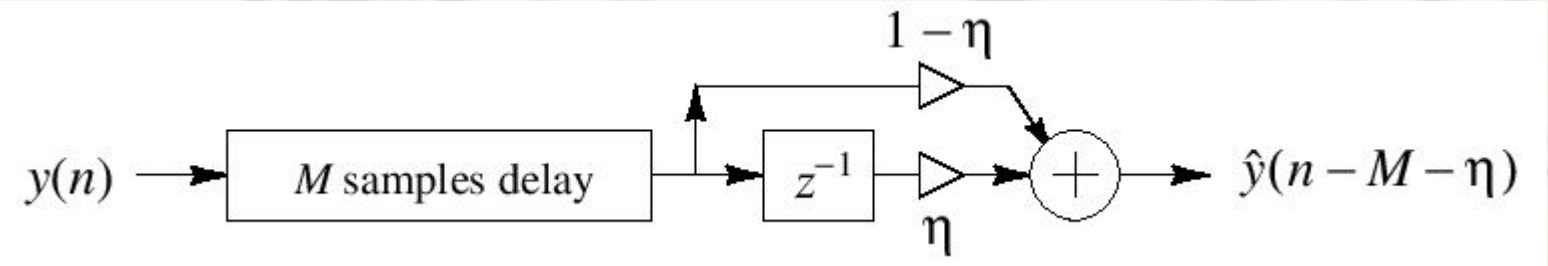
- Remove the first element of the delay line and append the new averaged value to the end

```
buf.append(lpfOut)
buf.pop(0)
```

Digital Signal Processing Theory

Extended Karplus Strong Algorithm (Digital Waveguide) for violin

- Generate low pass filter coefficients
- Set up parameters for (1) delay line length (2) wavetable length (3) fractional delay value (4) allpass factor for allpass filter
- Excitation of white noise is generated into a delay line
- Perform a linear interpolation on delay line
    - Values inserted into delay line will be allpass filtered and low pass filtered
    - More optimal than just a wave look up table because it jumps around rather than performs sequentially



How project works

- **`play_guitar()`** function will generate a plucked string sound for the duration of the key press
- **`play_bowed()`** function will generate a bowed string sound for the duration of the key press
- **`play_guitar()`** and **`play_bowed()`** will be called in real-time by the user pressing key(s) on the computer keyboard using Pygame
  - For monophonic sounds, there will be a small delay in the tone
  - For polyphonic sounds, there will be a larger delay in the chord
- Each function is assigned a frequency from the lookup dictionary **`note_tone_table`**

How project works

- **`play_guitar()`** function will generate a plucked string sound for the duration of the key press
- **`play_bowed()`** function will generate a bowed string sound for the duration of the key press
- **`play_guitar()`** and **`play_bowed()`** will be called in real-time by the user pressing key(s) on the computer keyboard using Pygame
  - For monophonic sounds, there will be a small delay in the tone
  - For polyphonic sounds, there will be a larger delay in the chord
- Each function is assigned a frequency from the lookup dictionary **`note_tone_table`**

How project works

- **`play_guitar()`** function will generate a plucked string sound for the duration of the key press
- **`play_bowed()`** function will generate a bowed string sound for the duration of the key press
- **`play_guitar()`** and **`play_bowed()`** will be called in real-time by the user pressing key(s) on the computer keyboard using Pygame
  - For monophonic sounds, there will be a small delay in the tone
  - For polyphonic sounds, there will be a larger delay in the chord
- Each function is assigned a frequency from the lookup dictionary **`note_tone_table`**

Demo

Improvements

- More processing for improved tone at lower frequencies such as below C2 and higher frequencies such as above C5
  - Higher frequencies fade very quickly
  - Lower frequencies drag for too long
- Consider the direction of bow strike and force used to play tone