EIE3510 Digital Signal Processing – Project Report
Composition and Instrumental Music Generation from Bird Songs

Yang Jiao 118010121 Jifei Zhao 118010437

18 December 2021

1. Introduction

In the AI generation, researchers started exploring the boundary between machine intelligence and artworks by using the machine to compose and generate music automatically. Previous studies mainly focused on deep learning methods, such as Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and the transformer architecture borrowed from Natural Language Processing (NLP). However, instead of the unexplainable, massive data-driven approach, we are trying to design an explicit and lightweight model with only a tiny amount of unmusical audio input.

Inspired by nature, we investigated the similarity between the singing of birds and instrumental music. With the limited amount of input audio, the generated music may have lower quality than those adopting machine learning methods. Nevertheless, to hear sounds from nature is essentially an incentive method for a musician's composition, and thus our model may also provide some insights and inspirations for musicians.

In this project, a piece of recorded bird sound was given to (1) map the sound to a series of notes (composition), and then (2) play the melody by mimicking the timbre of piano (music generation).

2. System design

A system was designed with the objective of generating music from bird sounds. The integral process of this system is shown in Fig. 1, which can be divided into two stages.

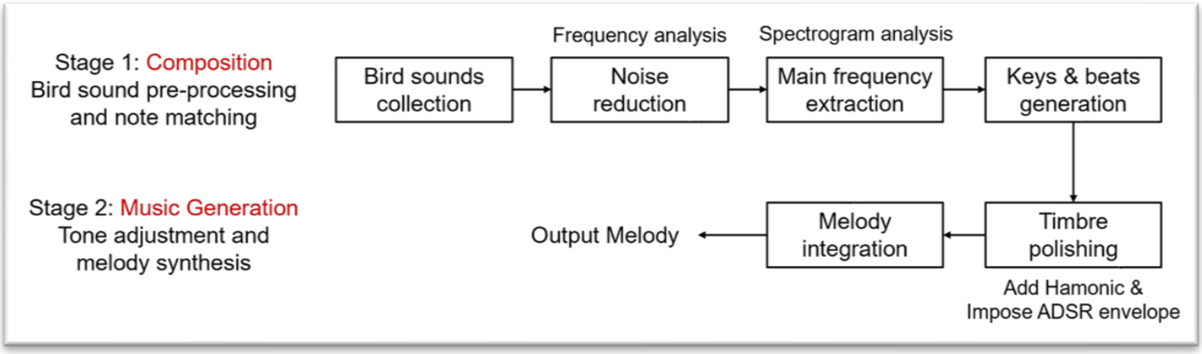


Figure 1: System of composition and instrumental music generation from bird songs

In Stage 1, the goal is transforming a piece of bird sound into two sequences of numbers representing proximate piano keys and corresponding beat of each key. The collected bird sounds were sent to two designed filters for noise reduction. After getting the filtered bird sounds, Short-time Fourier Transform was adopted to obtain the spectrogram for extracting the main frequency of each audio segment, which is generated from spectrogram. Based on the different main frequencies of audio segments, each segment was mapped to a standard piano key with the closest key frequency. To generate a music note that contains both the key value and beat, a mapping rule has been designed and utilized in this project. Finally, the notes were represented in the form of a sequence of piano keys and another sequence of beats.

The next stage focuses on the melody synthesis. According to the preliminary experiment in MATLAB, the timbre of the single-tone audio output is different from the timbre of piano. To better simulate the piano sounds, harmonics were added and the signal envelope will be properly modulated (ADSR envelope). The final step is integrating all the notes together to get a piece of melody.

3. Composition

3.1 Bird sounds collection

The dataset was obtained from an online database Xeno-canto (B. Planque & W. Vellinga, 2005), which provides access to a huge amount of sound recordings of wild birds from around the world. The website also gives a corresponding spectrogram graphic for each recording. In this project, a piece of Systellura longirostris's sounds with a sampling rate of 44100 Hz was applied in our system. It contains a complete period of a single bird's tweet and the relatively steady background noise. From the spectrogram graphic of this recording (Fig. 2), we can observe that the main component concentrates in a frequency interval of 3k Hz to 5k Hz, and the noise is distributed in a much broader range of frequencies. Based on the given spectrogram information, the strategies of noise reduction in following step can be derived.

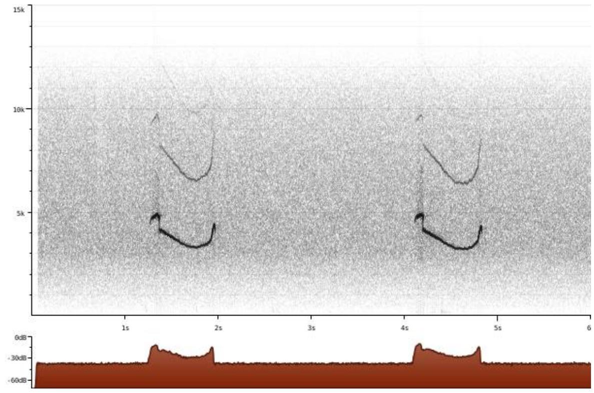
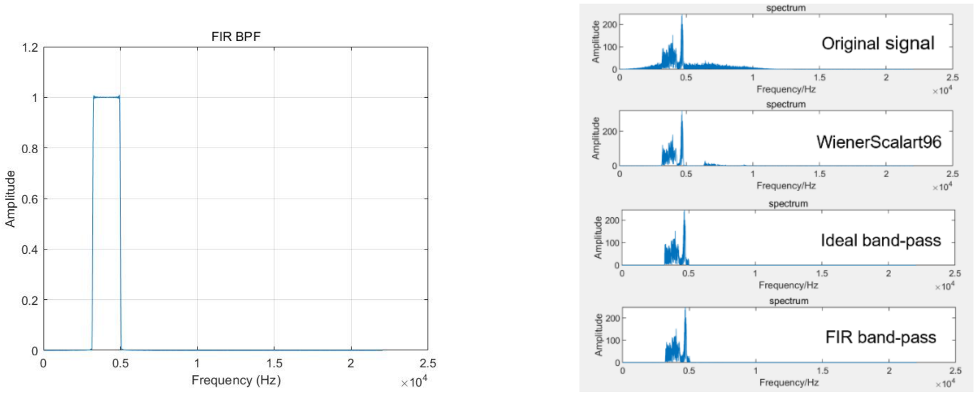


Figure 2: Spectrogram of Systellura longirostris's sounds recoding

3.2 Noise reduction

To remove noise from the signal, two kinds of filters have been designed in this project. The first one is an ideal bandpass filter, with two cutoff frequencies setting to 3.2k HZ and 5k Hz, according to the inspection and experiment. The second one is a FIR band-pass filter, with transition bands of 100 Hz width and ripples equal to 0.01. According to these specifications, a Kaiser window with M = 986 and β = 3.3953 can be obtained. The frequency response of this FIR band-pass filter is shown in Figure 3 (a). It can be seen that its shape is quite similar to the shape of previous ideal band-pass filter's frequency response. Thus, the filtering effects of both filters is maintaining the frequency components within the interval of 3.2k HZ to 5k Hz, while eliminating other frequency components outside of the interval.



(a) (b)

Figure 3: (a) Frequency response of designed FIR band-pass filter and (b) Filtering performances of three filters in frequency domain

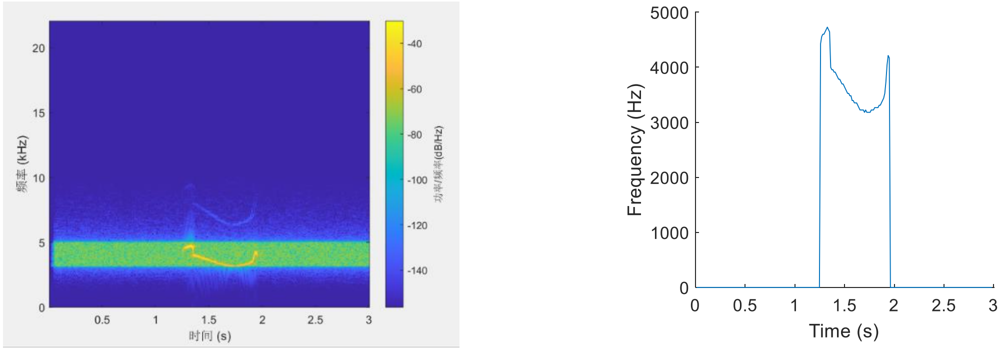
To figure out which one is better in this case, the above two filters were compared from three aspects—frequency domain performances, SNR and time complexity. A powerful filter designed by P. Scalart (1996) was also utilized as a reference. The detailed comparison can be found in Fig. 3 and Table 1. Both methods work fairly well and a slightly higher SNR of the filtering result based on Ideal band-pass filter can be seen. However, there is a nonnegligible difference of time complexity between the two filtering operations. That is because when the ideal band-pass filter is applied in MATLAB, the original signal should be transformed from time domain to frequency domain using "fft" function, and thus results in a $O(N\log_2 N)$ time complexity. On the other hand, when FIR band-pass filter is utilized, only the convolution operation in time domain should be done, which has a linear time complexity. Furthermore, during the real experiment, an evidently longer execution time of using the ideal band-pass filter can be observed. Consequently, to ensure a high speed of processing, the following implementation will entirely rely on the output signal from the FIR band-pass filter.

Table 1: SNR and time complexity of three filters

Filer name	SNR (dB)	Time complexity
Ideal band-pass filter	10.8853	$O(N\log_2 N)$
FIR band-pass filter	10.8275	$O(N)$
Scalart filter	13.2802	/

3.3 Main frequency extraction

After denoising, the next step is extracting the frequency information. Short-time Fourier Transform method is applied to get a more detailed spectrogram graphic (Fig. 4). In this spectrogram graphic, the widow length L is set to 1000 (23 ms Hamming window) with step R= 500 samples, and thus the signal is divided into small isometric segments in time domain. Significantly, too small window size results in bad frequency resolution. The chosen size is large enough for this project, because an approximation method will take place in later step of mapping frequency to key. In each segment, the frequency with largest intensity in frequency domain is taken as the main frequency. To get the digital representation, the final output in this step is a sequence of frequencies.



(a) (b)

Figure 4: (a) Spectrogram of denoised signal and (b) extracted main frequency over time

3.4 Keys and beats generation

The note that a piano played can be associated with a specific frequency (Weisstein, 2016). Based on the different main frequencies of audio segments, each segment can be mapped to a standard piano key with the closest key frequency using following equation. The frequencies are decreased by an octave in consideration of human perception.

$$Key(n) = \text{round}\left(12\log_2\left(\frac{f_{main}(n)}{440\ \text{Hz}}\right)\right) + 49 - 8 \tag{1}$$

Table 2: Mapping between number of repetitive frequencies and standard note duration

Number of repetitive frequency levels	Note value	Time duration (s)
1	Eighth note	0.125
2	Quarter note	0.25
3,4,5	Half note	0.5
6,7,8,9,10	Whole note	1

In music notation, a note value indicates the relative duration of a note, using the texture or shape of the notehead, the presence or absence of a stem, and the presence or absence of flags/beams/hooks/tails. Unmodified note values are fractional powers of two, for example one, one-half, one fourth, etc. In this project, four types of note are considered—Whole note, half note, quarter note, and eighth note, with the duration of a Whole note set to 1 second.

Table 3: Generated notes

Keys	89	90	87	86	85	84	83	83	84	85	86	87	88
Beats	0.25	1	1	1	1	1	1	0.5	1	0.25	0.125	0.125	0.25

A mapping rule between number of repetitive frequency levels and standard duration of notes is designed in this project (see Table 2). If repetitive frequency levels are more than 10, the first 8 levels should be firstly mapped to a Whole note and the rest should be mapped to a second note, and so on. According to the designed mapping rules for notes and beats, a digital music sheet can be obtained and is summarized in Table 3.

4. Music generation

Besides pitch, loudness, and duration, there is another fundamental aspect of sound referred to as timbre or tone color. Timbre allows a listener to distinguish the musical tone of a violin, an oboe, or a trumpet even if the tone is played at the same pitch and with the same loudness. In the second stage of this project, the piano sound of each note was mimicked and then integrated to a melody by considering the note and beat information (refer to the Stage 2 in Fig. 1). Intuitively, to implement the step of melody integration is straightforward since the only task required is to specify each note duration (according to the digital composition acquired previously) and connect the note signals (represented by vectors) in series. Thus, the focus of this section is to simulate the timbre of piano at certain note frequency. To perform timbre polishing, an appropriate envelop model and the superposition of some high order harmonics have been explored.

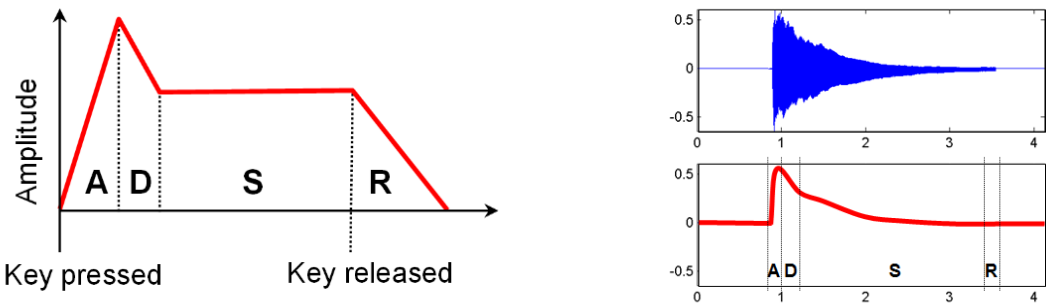
4.1 Envelope and ADSR model

Instead of directly apply the sinusoidal signal, the sound amplitude is variate in time and can be approximately described by the ADSR model. The model comprises the following four phases:

- **A**ttack phase: the sound builds up with a sudden increase of energy at the beginning of a music note;
- **D**ecay phase: the sound of a musical tone stabilizes and reaches a steady phase;
- **S**ustain phase: the sound energy remains more or less constant or slightly decreases;
- **R**elease phase: the musical tone faded away.

Fig. 5 shows a graphical representation of (a) the ADSR model (Muller, 2015) and (b) the ADSR approximation for piano sound. The piano sound usually establishes quickly with a very short attack time and then decays drastically. Hence, the ASDR envelope can be further approximated by an exponential function.

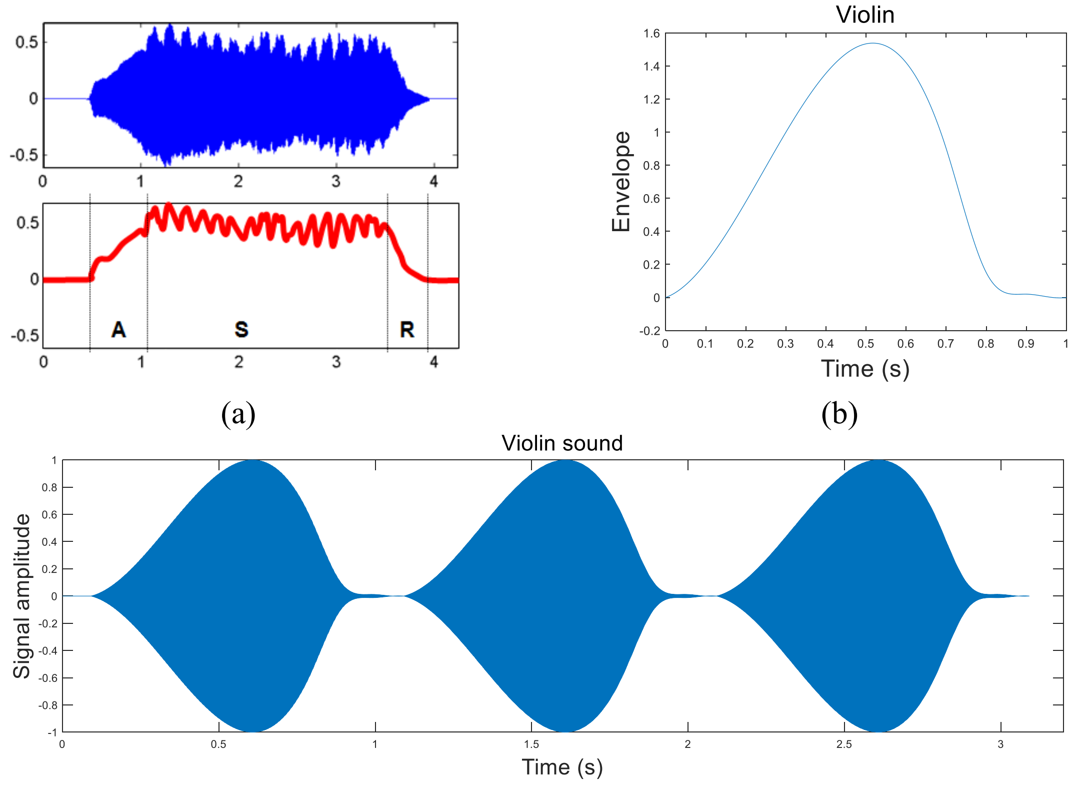
However, not the timbre of every instrument can be effectively simulated by the ASDR model. For example, the violin sound is difficult to mimic by an envelope of elementary function. Violin has effect of vibration on center frequency and tremolo on signal amplitude, and thus only simulate the sound by envelope modulation is not enough. Fig. 6 demonstrates the waveform for violin and our approximation by interpolation of some selected turning points.



(a)

(b)

Figure 5: (a) Schematic view of an ADSR envelope and (b) Waveform and amplitude envelope representation for piano playing the same note C4 (261.6 Hz)

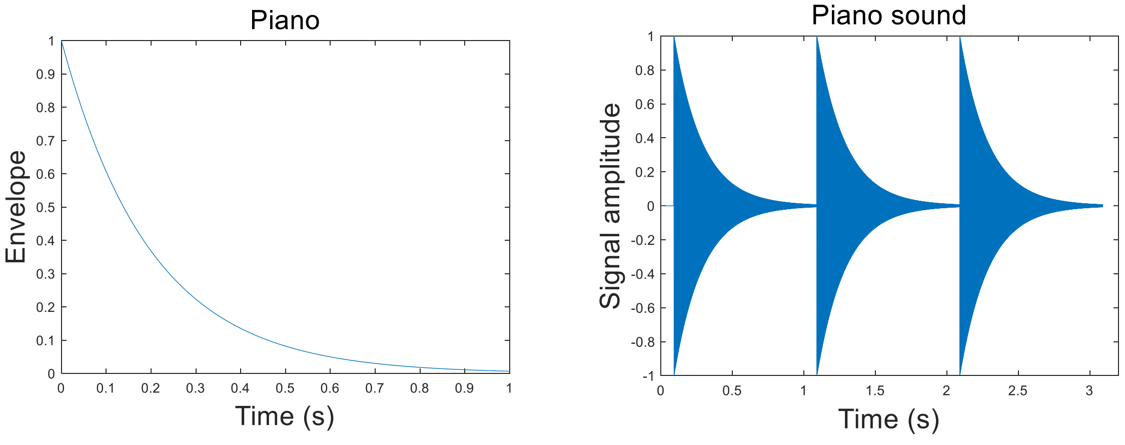


(a)

(b)

(c)

Figure 6: (a) Waveform and amplitude envelope representation for violin playing the same note C4 (261.6 Hz), (b) approximate interpolated envelope function, and (c) corresponding waveform for the notes C5, D5, E5 (do, re, mi)



(a)

(b)

Figure 7: (a) Graphical view of the envelope function $\mathrm{env}(t)$ and (b) corresponding waveform for the notes C5, D5, E5 (do, re, mi)

Consequently, it is reasonable and straightforward to simulate the piano sound and to polish the timbre by an exponential function as the envelope (Fig. 8). Here the envelope function is defined as:

$$\mathrm{env}(t - t_0) = e^{-a_0(t-t_0)}, \qquad \forall\, t \in [t_0, T + t_0] \tag{2}$$

where t_0 is the initial attack time of the piano key, a_0 is the decay coefficient, and T

represents the duration for a whole note. Here, $T = 1\text{s}$ and $a_0 = 5\text{s}^{-1}$. The signal amplitude has unit 1, and the value of a_0 is determined based on T, such that the modulated signal has proper amplitude decay in one whole note duration.

4.2 High order harmonics and final music output

In addition, instead of employing the single tone note, the notes were combined with some high order harmonics. Here we define the piano note frequency as the fundamental frequency f_c, and the harmonic signals usually has center frequencies at nf_c for some integer $n > 1$. These signals are called harmonics because the superposition of them on the fundamental signal component will not change the fundamental period of the signal. Thus, a general signal representation with harmonics can be described by Eq. 3.

$$s(t) = \cos(2\pi f_c(t - t_0)) + \sum_{n=2}^{N_0} a_n \cos(2\pi n f_c(t - t_0) + \phi_n) \tag{3}$$

where a_n is the amplitude of each harmonic, ϕ_n is the corresponding phase offset, $N_0 - 1$ is the total number of high order harmonics, and $s(t)$ is the resulted signal. The fundamental signal component is normalized with unit amplitude and zero phase offset without loss of generality.

With superposition of harmonics, the resulted sound would be more saturated and colorful. For professional music synthesis, one approach to determine the value for a_ns and ϕ_ns is by sampling and analyzing the real instrumental sound. Due the limitation of our accessibility to the instrument, a_ns and ϕ_ns were determined just by empirical tuning. Table 4 summarizes the adopted coefficient values.

Table 3: Value of coefficients N_0, a_n, and ϕ_n in Eq. 3

N_0	n	a_n	ϕ_n
3	2	0.4	$2\pi/3$
	3	0.15	$4\pi/3$

Besides, to apply the envelope function, it is similar as the procedure of amplitude modulation in communication system, and the final polished signal for one note at fundamental frequency f_c is:

$$y(t) = \text{env}(t) \times \frac{s(t)}{\max|s(t)|} \tag{4}$$

Here $s(t)$ is divided by its maximum value for amplitude normalization. The amplitude of $y(t)$ is thus limited in $[-1, 1]$ for audio output. Fig. 8 presents the output piano waveform.

After timbre polishing for a single note, the melody can be synthesized by associating each note with its assigned tempo and then connecting the notes in series. A graphical user interface (GUI) has been designed in MATLAB APP Designer for user to go through the entire procedure of this project interactively. The user manual can be found in the appendix.

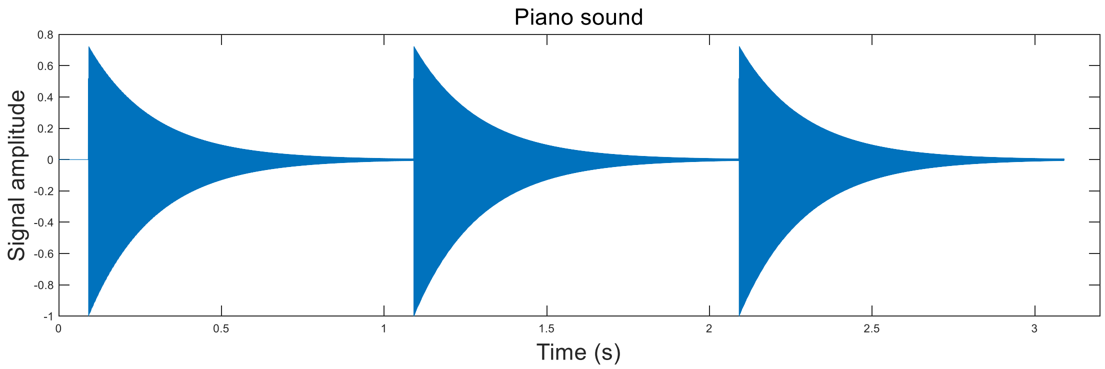


Figure 8: The piano sound waveform for the notes C5, D5, E5 (do, re, mi)

5. Conclusion and project evaluation

In this project, a piece of piano melody was created from a slice of bird songs. The bird song was pre-processed by noise reduction and the main frequency components were extracted by spectrogram analysis. With the time-frequency information, the bird song could be mapped to the digital notes and beats to complete the composition. Based on the digital music sheet, the piano melody was generated by timbre polishing, concerning the ASDR envelope and superposition of high order harmonics.

There is also space for improvement for this project. For example, the model can be generalized to adopt to different kinds of bird songs and may synthesize different kinds of music instrument sounds. To adaptively pre-process different kinds of bird songs, a prior step of frequency analysis is required to determine the cut-off frequencies of the bandpass filter, or the statistical information can be utilized for filtering (such as the Scalart filter). To synthesize different kinds of music instruments, a more accurate ASDR model is necessary, and it is better to consider the vibration and tremolo effects.

Nevertheless, the proposed schematic is complete and robust to background noises by comparing different filtering methods. The result of simulated piano sound is satisfying considering the low implementation complexity. This project effectively provides a firm basement model of composition and music generation for bird songs.

References

B. Planque and W. Vellinga, Xeno-canto, May 30, 2005. Accessed on: December 18, 2021. [Online]. Available: https://xeno-canto.org/

E. Weisstein, Eric Weisstein's Treasure Trove of the Music, February 1, 2016. Accessed on: December 18, 2021. [Online]. Available: http://www.ericweisstein.com/encyclopedias/music/about.html/

M. Meinard, “Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications.” Cham: Springer International Publishing AG, 2015. Print.

P. Scalart and J. V. Filho, “Speech enhancement based on a priori signal to noise estimation,” 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings, 1996, pp. 629-632 vol. 2, doi: 10.1109/ICASSP.1996.543199.

Appendix: Manual for the GUI mainAPP.mlapp

The GUI has a 3-tabbed control group at the top of the interface, which corresponds to sequential steps of this project.

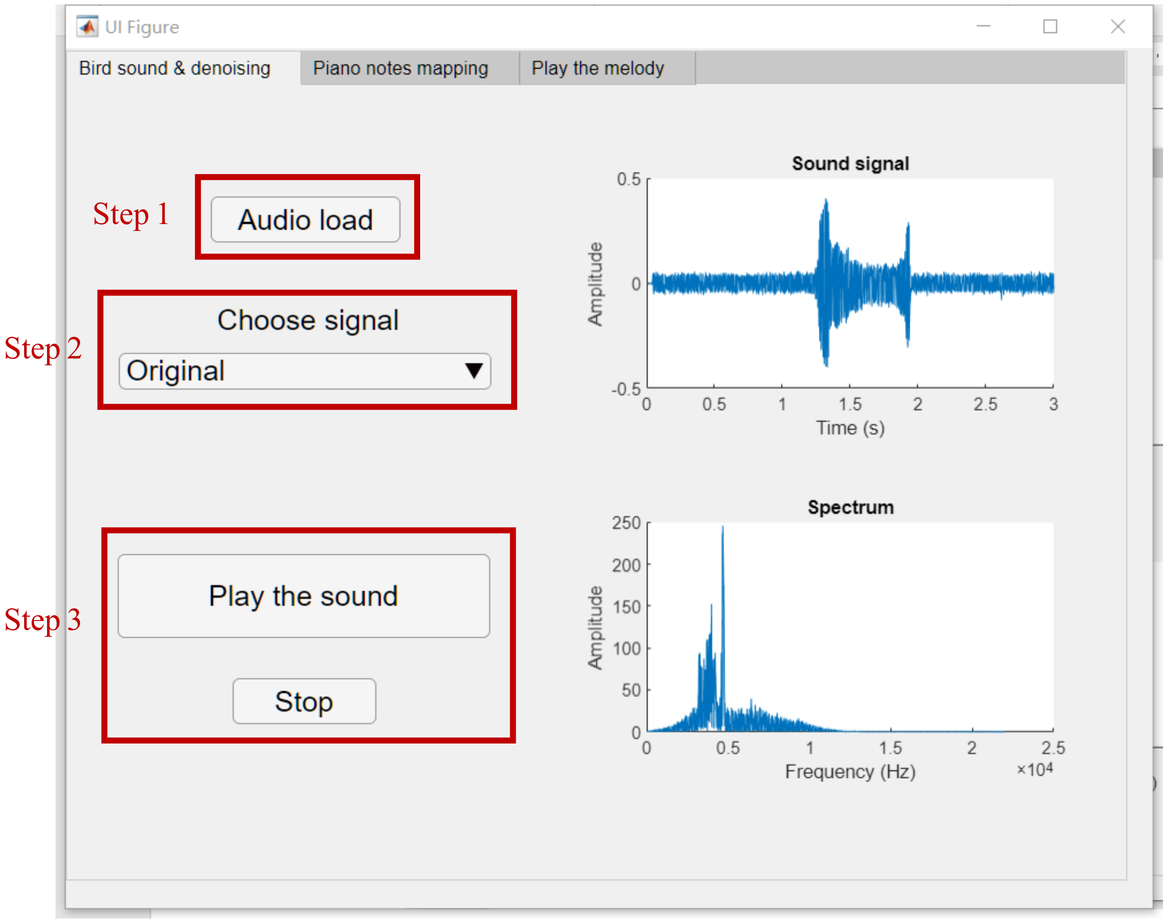


Figure 9: GUI page “Bird sound & denoising”

On the first page, click the button “Audio load” first to load the audio. To view the time and frequency domain representations of the orignal and filtered signals, click the drop-down menu “Choose signal” and select the desired signal as Fig. 10.

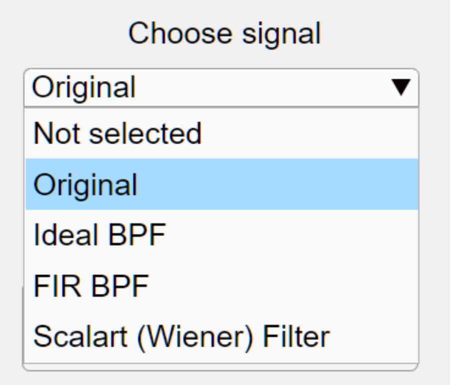
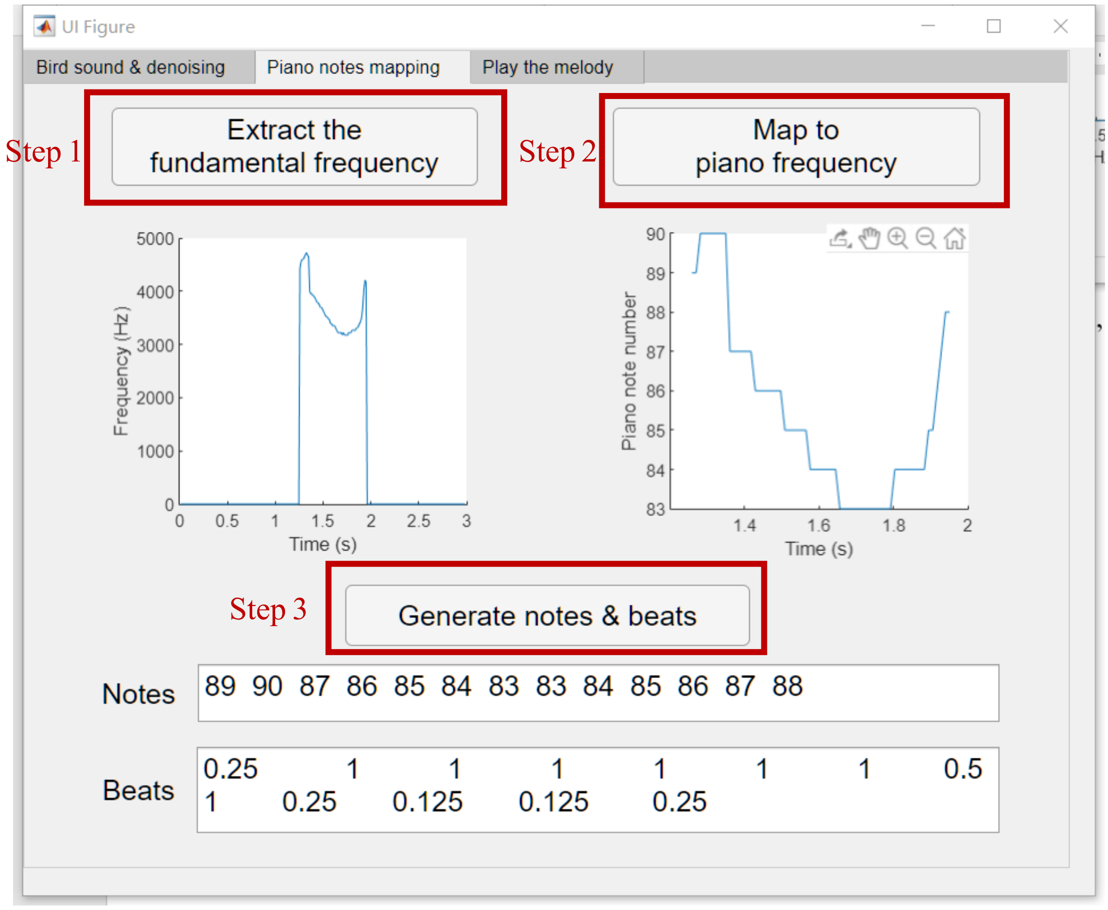


Figure 10: The signal options. From top to bottom: signal of original bird song, signal after ideal bandpass filter, signal after FIR bandpass filter, signal after Scalart (Wiener) filter (reference bird signal)



Notes 89 90 87 86 85 84 83 83 84 85 86 87 88

Beats 0.25 1 1 1 1 1 1 0.5
1 0.25 0.125 0.125 0.25

Figure 11: GUI page “Piano notes mapping”

After denoising, follow the steps in this page to first extract the fundamental frequencies, map the frequencies to piano frequencies, and then generate the notes and beats for representation of the digital music sheet. After clicking each button, corresponding figures or texts will show up below the buttons.

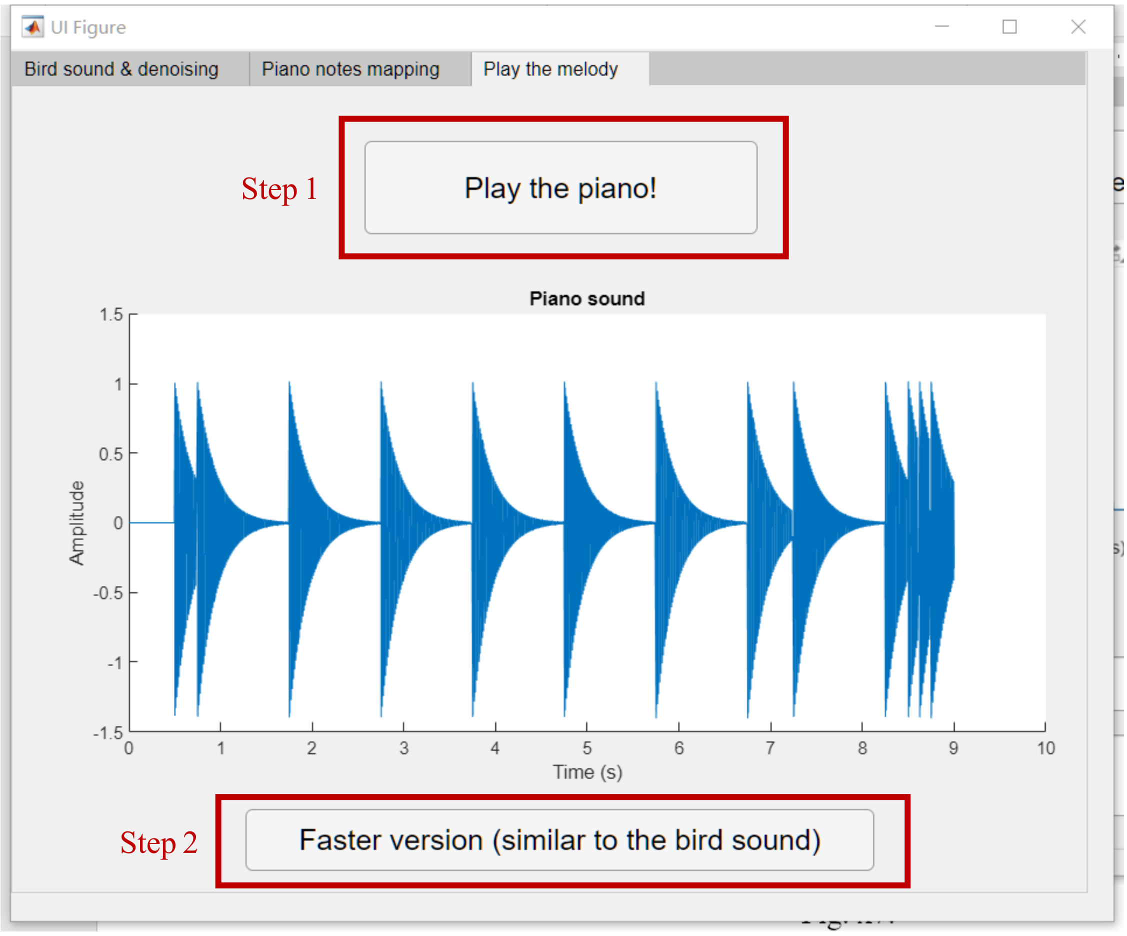


Figure 12: GUI page "Play the melody"

The final step is to play the melody. Click the "Play the piano!" button. The music will be broadcasted, and a graphic representation of the music signal will show up in the axes. If you think the melody sounds too different from the original bird song, you may click on the button "Faster version (similar to the bird sound)". This is because the piano version has an elongating effect on each fundamental frequency, resulting the original frequency components hard for recognition.