

Univerza v Ljubljani



University of Ljubljana
Faculty of Computer and Information Science

SEMINAR 1

Synthesis of clarinet

Subject: Digital signal processing
Author: Timotej Tim Rus, 63180256

Ljubljana, december 2020

Table of Contents

1. Introduction	3
2. Method	4
3. Result.....	5
4. Discussion.....	7

1. Introduction

How does clarinet work?

The clarinet is a single reed instrument. Made from wood, the clarinet produces a fluid sound when air is blown between a single reed and the mouthpiece. By pressing metal keys with the fingers of both hands, the player has the ability to play many different notes very quickly. The clarinet can play in the low register, where the notes are rich and full. It can play in the middle register. It can also play in the high register.

Sound on a woodwind instrument comes from a vibrating column of air inside the instrument. The player makes this column of air vibrate in one of three ways: as air is blown across the top of an instrument (like the flute), across a single reed (like the clarinet), or across two reeds (like the oboe).

For notes i will use formulas from instructions:

$$y(n) = \cos(\omega * n) + 0.375 * \cos(3 * \omega * n) + 0.581 * \cos(5 * \omega * n) + 0.382 * \cos(7 * \omega * n) + \\ + 0.141 * \cos(9 * \omega * n) + 0.028 * \cos(11 * \omega * n) + 0.009 * \cos(13 * \omega * n), \quad (1)$$

$$\omega = 2 * \pi * \frac{F_0}{F_s}, F_0$$

Hypothesis

The signal which I will plot will be similar to one on the online classroom.

2. Method

Task1:

For first task I wrote a function that loops through list and checks if the length of note is equals to 0.25 or 0.5. This is how I determined the length on note for example: if the len in list is 0.25, I used the formula from the instructions ($w = 2 * \pi * F_0 / F_s$) and set $n = 0:1999$ (which is $\frac{1}{4}$ of F_s), after that I added result of function `note(w, n)` to my list `toPlay`.

```
playWithAdsr = [];  
  
for k=1:numel(song)  
    if len(k) == 0.25  
        w = 2 * pi * (song(k)/8000);  
        n = 0:1999; %1/4 of Fs  
  
    else  
        w = 2 * pi * (song(k)/8000);  
        n = 0:3999; %1/2 of Fs  
    end  
  
    t = n/8000; % time in seconds
```

After that I set values for Attack, Decay, Sustain and Release with `linspace` (generates linearly spaced vector) and save them to values list.

```
%Attack - The time it takes for the note to reach the maximum level  
A = linspace(0, 0.9, (length(note(w, n))*0.20));  
%Decay - The time it takes for the note to go from the maximum level to sustain level  
D = linspace(0.9, 0.9, (length(note(w, n))*0.0));  
%Sustain - The level while the note is held  
S = linspace(0.9, 0.9, (length(note(w, n))*0.70));  
%Release - The time it takes for the note to fall from the sustain level to zero when released  
R = linspace(0.9, 0, (length(note(w, n))*0.1));  
values = [A D S R];
```

Below that, I multiplied signal with envelope, so the sound will be more like clarinet-like.

```
a = zeros(size(note(w, n)));  
a(1:length(values)) = values;  
play = note(w, n) .* a; %tone
```

Task2:

First I made notes variables and add frequencies for each note, then add them to more tables named line1, line2, etc.. (in the order of my song - Ringaraja). Then added those lines of song to the list through which I am looping in task 1, and set length of each note in list on len.

```
%notes for my song
g = 392.00;
a = 440.00;
e = 329.63;
f = 349.23;
b = 493.88;

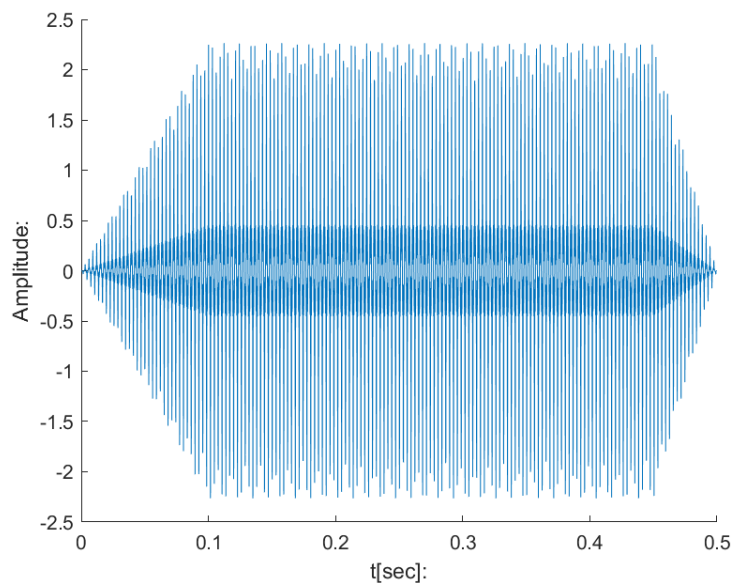
%tables of notes and lengths of notes
line1 = [g g a g];
line2 = [e e];
line3 = [g g a g];
line4 = [f f];
line5 = [g g a g];
line6 = [b b a];
line7 = [g g a f];
line8 = [e e];

%Ringaraja -> Slovenian children song
song = [line1 line2 line3 line4 line5 line6 line7 line8];
len = [0.25 0.25 0.25 0.25 0.5 0.5 0.25 0.25 0.25 0.25 0.5 0.5 0.25 0.25 0.25 0.25 0.25 0.5 0.25 0.25 0.25 0.25 0.25 0.5 0.5];
```

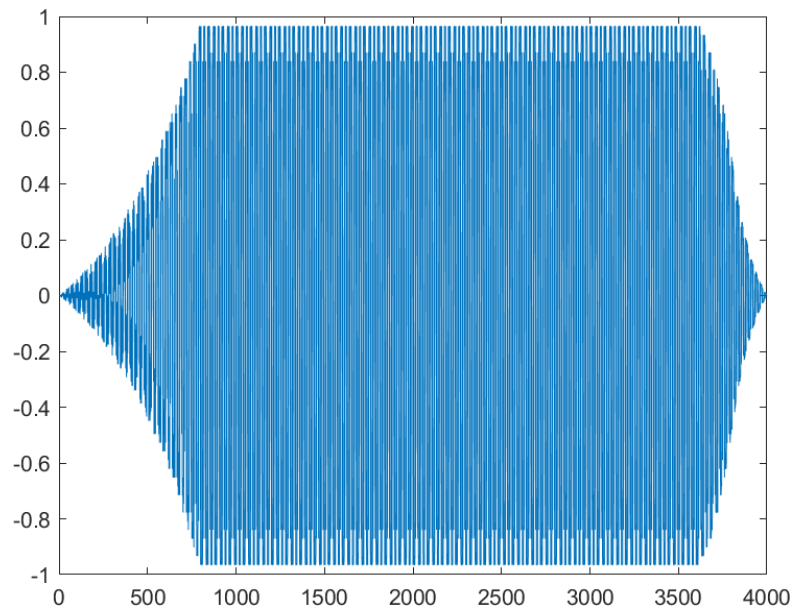
The song was played on the end of the file with function soundsc(playWithAdsr, Fs) from Matlab.

3. Result

Sound from my ADSR function is cleaner than sound which was just played with my notes in matlab. The plot of Ringaraja song with clarinet is similar to one on our online classroom.



Plot from my function (clarinet playing Ringaraja song).

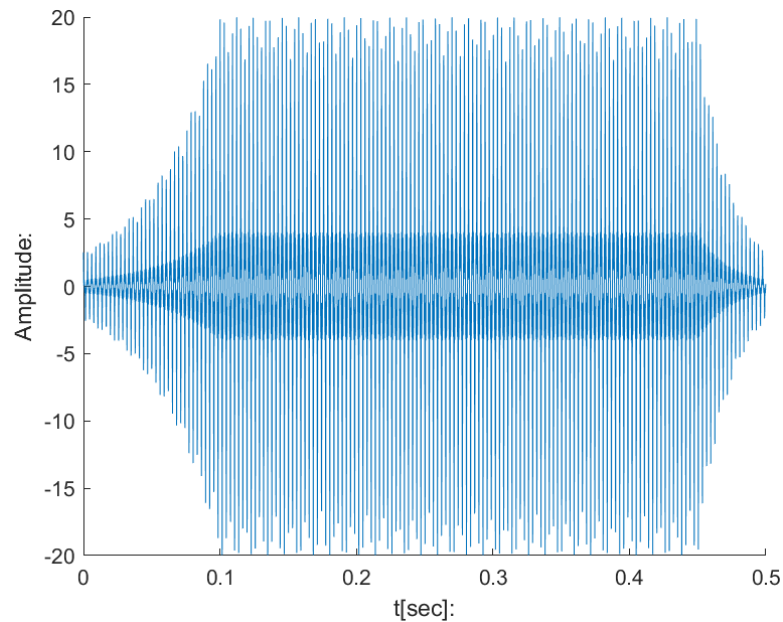


Plot from clarinet sound from our online classroom.

The most important components for ADSR envelope to generate more clarinete-like sound are A – Attack (The time it takes for the note to reach the maximum level) and R – Release (The time it takes for the note to fall from the sustain level to zero (silence) when released).

4. Discussion

I tried to get plot signal more like one from our classroom, with logspace function, but it was not as good as linspace (in sound and in signals). Picture below:



ADSR using logspace

I assumed that my result will be something similar than one I got from internet (clarinet.wav, from online classroom).