International journal of imaging science and engineering

ISSN: 1934--9955 www.ijise.net

Vol-09 Issue-01 May 2014

Music Synthesis using Sinusoid Generator, ADSR Envelope Generator and Composer Code

Sirishanaadu ,Germany

Article Info

Received: 27-02-2014 *Revised: 12 -03-2014* *Accepted: 16-04-2014* *Published:11/05/2014*

Abstract: The capacity to digitally synthesise waves is a widely used approach. Modems, software radios, and DTMF (Touch Tone) generators are just a few of the many pl3aces you could find this technique in use. Its use in music synthesis is among its most well-known consumer-oriented applications. Here, the performer generally uses a single synthesizer to emulate a wide variety of instruments and effects. Some of the applications of sampling and reconstruction theory may be shown via waveform synthesis, which is often taught early in a Digital Signal Processing (DSP) course. Furthermore, computer music provides a fascinating setting for practical experience with waveform synthesis. In order to modulate the loudness of the tone, two instruments are utilized: a sine-wave generator and an ADSR envelope generator. A curve often referred to as the Attack-Decay-Sustain-Release (ADSR) envelope may include the amplitude of the tone. These two components are the backbone of the project that will allow us to explore musical synthesis and computer-based music utilizing the PC's sound card and MATLAB's built-in sound capabilities.

Keywords: technologies, ADSR, digital signal processing, and computing

Introduction

1. The capacity to digitally synthesise waves is a widely used approach. Modems, software radios, and DTMF (Touch Tone) generators are just a few of the many places you could find this technique in use. Its use in music synthesis is among its most well-known consumer-oriented applications. Here, the performer generally uses a single synthesizer to emulate a wide variety of instruments and effects. As an example of how sampling and reconstruction theory may be used, waveform synthesis can be taught early on in a standard undergraduate Digital Signal Processing (DSP) course. Furthermore, computer music provides a fascinating setting for practical experience with waveform synthesis. We provide two basic MATLAB tools that we developed for a waveform synthesis project in this article. In order to modulate the amplitude of a tone, these instruments include a sine wave generator and an envelope generator. These two components are the backbone of the project that will allow us to explore musical synthesis and computer-based music utilizing the PC's sound card and MATLAB's built-in sound capabilities.

2. Implementation

The implementation of a music synthesizer (AM-based) involves three codes: 1) tone synthesizer or sinusoid generator, 2) ADSR envelope generator, 3) composer/player code. We assume digital synthesis at a rate of *fs* = 16,000 samples per second. At this rate we are able to reproduce all piano frequencies according to Nyquist theory.

2.1 ADSR Envelope Generation

The sound output of musical instruments does not immediately build up to its full intensity nor does the sound fall to zero intensity instantaneously. It takes a certain amount of time for the sound to build up in intensity and a certain amount of time for the sound to die away. The period of time during which a musical tone is building up to some amplitude (volume) is called the “attack time” and the time required for the tone’s intensity to partially die away is called its “decay time.” The time for final attenuation is called the “release time.” Many instruments allow the user to hold the tone for a period of time which is known as

International journal of imaging science and engineering

ISSN: 1934--9955 www.ijise.net

Vol-09 Issue-01 May 2014

the "sustain time" so that various note durations can be achieved. The amplitude of the tone can "fit" inside a curve often called the Attack-Decay-Sustain-Release (ADSR) envelope.

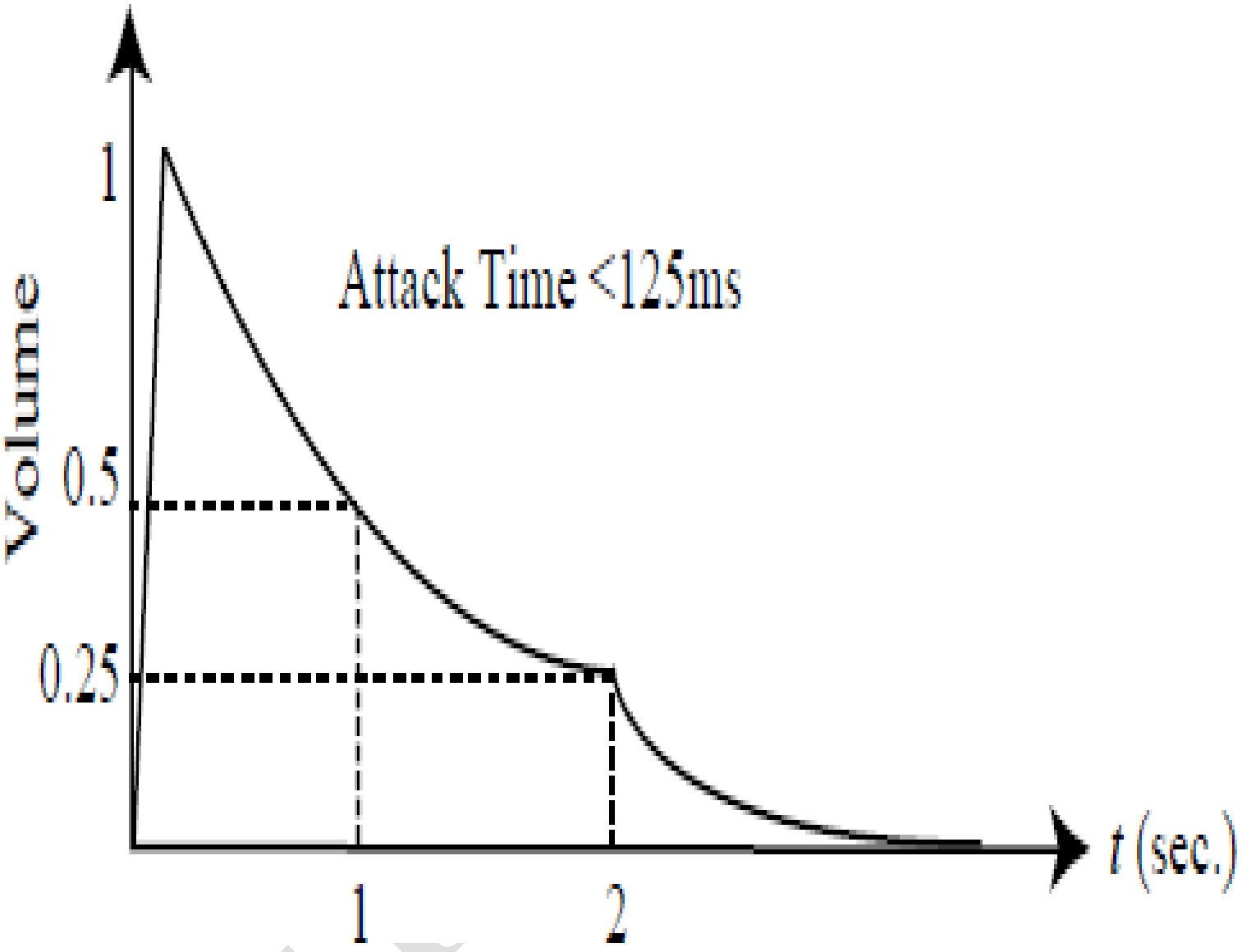


Figure 1: ADSR Envelope for Piano

A synthesizer duplicates the intensity (volume) variation of the tone by multiplying (modulating) the amplitude of the sinusoid with a scale factor dictated by the ADSR envelope, $a(t)$

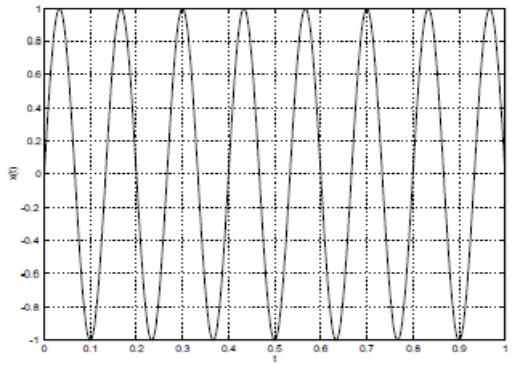
$$y(t) = a(t) * x(t) \qquad (1)$$

The resulting signal, $y(t)$ is referred to as the amplitude- modulated (AM) tone.

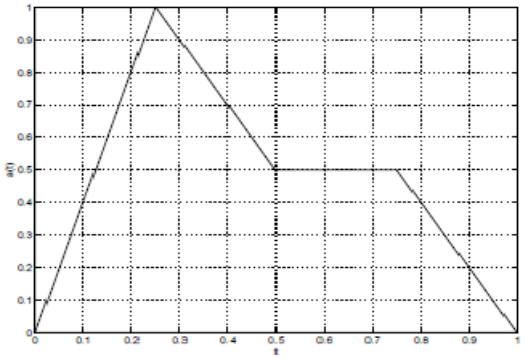
International journal of imaging science and engineering

ISSN: 1934--9955 www.ijise.net

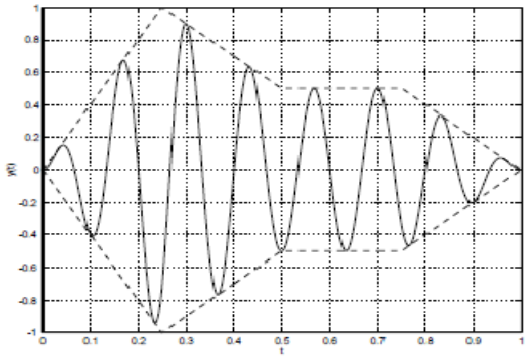
Vol-09 Issue-01 May 2014



(a)



(b)

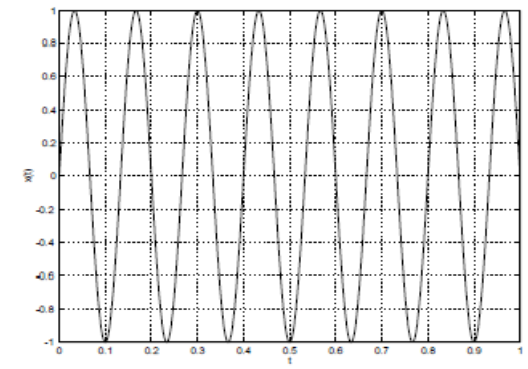


(c)

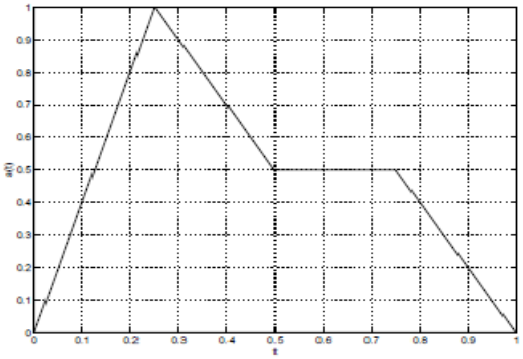
International journal of imaging science and engineering

ISSN: 1934--9955 www.ijise.net

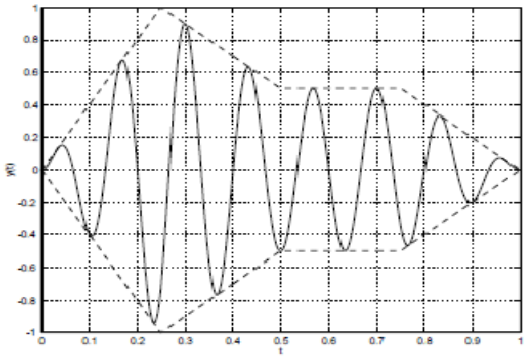
Vol-09 Issue-01 May 2014



(a)



(b)



(c)

Figure 1: (a) Sinusoid, (b) ADSR envelope, (c) Amplitude modulated (AM) Sinusoid

2.2 Envelope Generator

As described earlier, the envelope will give the sinusoid a volume characteristic which as a first approximation, imitates that of a real instrument. The envelope values are stored as a single vector so that a simple element-by-element product between the sinusoid vector and the envelope vector yields the amplitude modulated sinusoid. The envelope is constructed one segment (A, D, S, and R) at a time. We approximate each segment with a simple exponential which rises or decays asympotically to the target value. This approximation then leads to a simple digital filter implementation (difference equation) which we are familiar with, whose response yields samples of an exponential curve.

International journal of imaging science and engineering

ISSN: 1934--9955 www.ijise.net

Vol-09 Issue-01 May 2014

In addition, we allow for a gain parameter to control the speed at which the exponential reaches the target value. The difference equation is given by a single-pole filter,

a(n)=âg+(1-g)a(n-1)

where $a(n)$ are the envelope values, ˆ a is the target value, and g is the gain parameter.

C. Composer/Player Code

The final code segment generates sinusoids with the proper frequency and an ADSR envelope to amplitude modulate the sinusoid.

3. Result

3.1 MATLAB Code

Function to generate ADSR envelope

```
function[a]=adsr_gen(target,gain,duration) fs=16000;
a=zeros(fs,1); duration=round(duration./1000.*fs); start=2;
stop=duration(1);
%attack phase for n=(start:stop)
a(n)=target(1)*gain(1)+(1-gain(1))*a(n-1); end
%Sustain phase start=stop+1; stop=start+duration(2); for n=(start:stop)
a(n)=target(2)*gain(2)+(1-gain(2))*a(n-1); end
%Release phase start=stop + 1; stop=sum(duration); for n=(start:stop)
a(n)=target(3)*gain(3)+(1-gain(3))*a(n-1); end
```

B.Function to generate sinusoid function[x]=singen(f,fs,N) n=(0:N-1);
x=sin(2*pi*f/fs*n);

C. Composer/Player code function [] = sound_play(f) target=[0.99999;0.25;0];
gain=[0.005;0.0004;0.00075];
duration=[250;1250;500]; fs=16000;
tot_dur=floor(sum(duration)/fs); [adsr]=adsr_gen(target,gain,duration); figure(1)
plot(adsr); x=singen(f,fs,length(adsr)); figure(2)
plot(x);
b=adsr.';

y=b.*x; % Modulate wavplay(y,fs); figure(3)
plot(y);

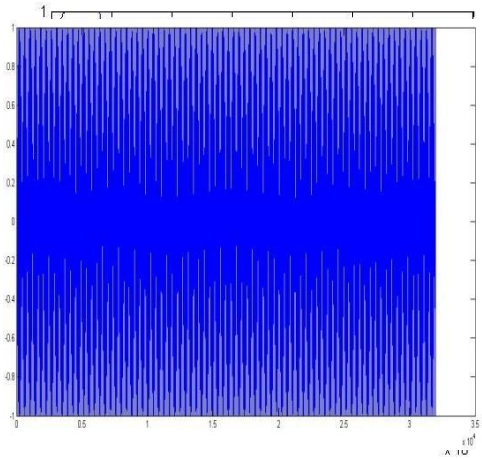
International journal of imaging science and engineering

ISSN: 1934--9955 www.ijise.net

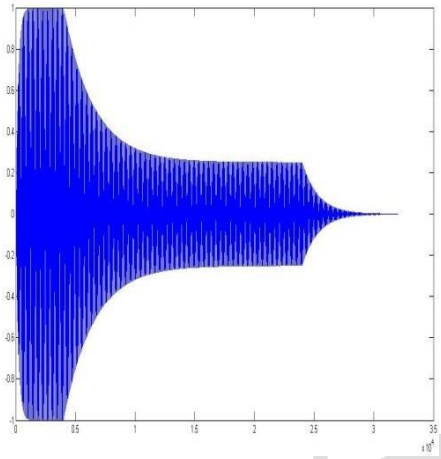
Vol-09 Issue-01 May 2014

D. Output waveforms (i.)ADSR envelope

(ii) Sinusoid wave



(iii) AM Modulated signal



4. Conclusion

We have created a computer music exercise for this article. This assignment uses three separate MATLAB programs to create a sine wave, an ADSR envelope to alter the tone's amplitude, and a melody out of the modulated tones.Prospects for the Future

With computer music as the eventual goal, the concepts may be expanded upon to include broader notions in waveform synthesis.

References

[1] JThe synthesis of complicated audio spectra by methods of frequency modulation was published in the Journal of the Audio Engineering Society by Chowning in September 1973, volume 21, issue 7, pages 526-534.

[2] Courses in Matlab available online.This training resource is available at: http://in.mathworks.com/academia/student_center/tutoria ls/.