.

COMPARISON BETWEEN A HARDWARE AND A SOFTWARE SYNTHESIZER

Ubaldo Gaume Echeverría
e-mail: ubaldo.gaumeea@udlap.mx
Mtro. Félix Eduardo Guerrero Castro
e-mail: felix.guerrero@udlap.mx
Dr. José Miguel David Báez López

ABSTRACT: *The purpose of this project was to construct a modular analog Hardware synthesizer and compare it to a Software synthesizer of the same characteristics. The audio synthesis technique used for the hardware synthesizer was subtractive audio synthesis, therefore a voltage controlled oscillator was built to obtain the initial waveform and subsequent modules were constructed to get a richer sound, these modules include a voltage controlled filter, an ADSR envelope generator, a voltage controlled amplifier and a MIDI to CV as the last module used to control the other modules. In order to compare between Hardware and Software synthesizers a synthesizer was created using Synthedit which is a free synthesizer creator that can generate a synthesizer with all of the same modules used in Hardware, therefore a comprehensive comparison was achieved by inspection of the sound waveforms produced by both synthesizers.*

KEYWORDS: Synthesizer, Arduino, VCA, VCO, ADSR

1 INTRODUCTION

A synthesizer is an electronic instrument capable of producing and generating a great variety of sounds done by combining different frequencies [1]. An early synthesizer was built in 1876, when Elisha Gray found out by accident that he could control the sound of an electromagnetic circuit, which happened when he tried to invent a “musical telegraph”, this turned out to be the first one note oscillator [2]. After this, the first real synthesizer title goes to Leon Theremin’s “Theremin”. The Theremin was a musical instrument which produced sound using oscillators whose frequency varied by placing your hand closer or farther away from an antenna. In the following decades, further advances in synthesizer technology continued and new techniques for obtaining more complex synthesizers were explored. As mentioned earlier, different audio synthesis techniques were developed and applied to analog and digital synthesizers alike, due to the great number of techniques that currently exist only the most common of them will be explained next.

The additive synthesis technique produces sound by adding different sine waveforms. This procedure is based on Fourier theory, it can be described by considering these sine waveforms as blocks that, when put together, they add up to a different waveform and therefore a more complex sound can be obtained. This technique can potentially generate sounds that closely approximate the ones produced by acoustic musical instruments, but this can be a complicated task due to the analysis that must be done to achieve this.

Subtractive audio synthesis is another technique, which produces sounds by generating a waveform that contains more harmonic content than a sine waveform like a pulse or a sawtooth, then this waveform passes through a filter which subtracts harmonics to obtain the desired sound, this is the technique that is used in this project and it is normally associated with voltage controlled synthesizers due to its simple implementation when compared with other techniques. Granular audio synthesis generates sounds by putting together small samples of sound which have different characteristics; this allows more control over sounds parameters like pitch, duration and volume envelope. Other audio synthesis techniques include frequency modulation (FM), amplitude modulation (AM), wavetable synthesis, etc. These techniques could be further explored and another set of comparison results could be obtained [3].

This project has the goal of constructing a modular hardware and software synthesizer. This synthesizer will be used for obtaining a comparison between these two implementations. These synthesizers will use several modules which will be mentioned and explained next.

The scheme used for both synthesizers is shown in a block diagram in figure 1.

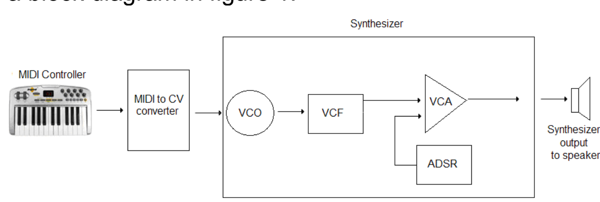

Figure 1. Block diagram of the synthesizer, its interface and the MIDI controller.

A voltage controlled oscillator will be used (VCO-Voltage Controlled Oscillator) to obtain a monophonic sound and generate 4 types of waveforms: square, sawtooth, sine and triangle. A voltage controlled filter will be used (VCF) which will filter the signals coming from the oscillator. The chosen circuit provides 3 types of filtering: a low pass filter, a high pass filter and a band pass filter. The use of a filter gives the performer another way to manipulate the electric signals thus obtaining a broader palette of sounds.

An ADSR envelope generator will be used (ADSR stands for Attack, Decay, Sustain and Release) as a means to

978-1-4244-5353-5/10/$26.00 ©2010 IEEE 311

produce an appropriate signal that can control the VCA (voltage controlled amplifier) stage. The individual control of each parameter is explained in figure 2. The attack parameter determines how fast the sound is going to arrive at its peak value after it has been produced. The decay time is how fast the sound will lower to the sustain level. The sustain parameter is the time that the sound will stay constant after the decay. Finally the release time indicates how fast the sound will drop to zero [4].

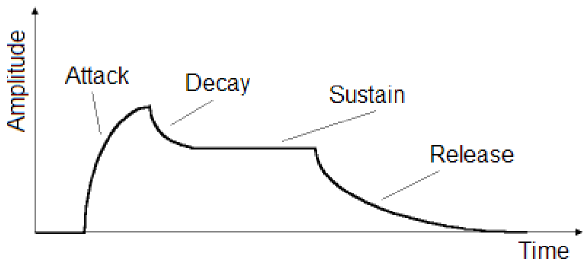

Figure 2. Graphical representation of each of the 4 parameters controlled with the ADSR generator with respect to time.

A VCA was also used in this project. It is used to handle the amplitude of the signal coming from the synthesizer. This signal will be amplified or attenuated according to the settings from the ADSR module. The ability to modify the amplitude envelope of a sound will give the performer the freedom to simulate the envelope occurring in real acoustic instruments.

These modules are controlled with a MIDI controller. This piece of hardware is a keyboard that sends only digital data in the form of MIDI messages. These messages are sent to an Arduino microcontroller and they are converted to a 6-bit word, which in turn is fed to an R-2R digital to analog converter (DAC) and this results in a control voltage that can be used as the input signal for the VCO (and it can be used as the control voltage for other modules as well). The microcontroller also produces two control signals (gate and trigger) that trigger the envelope generator when a key is pressed.

The software synthesizer uses the same modules as the hardware synthesizer.

2 Software synthesizer using Synthedit

Synthedit is a software program that gives users the tools to create audio applications by using a modular approach, the modules included in the program will generate a great number of processing options. The software is used to create audio effects and modular synthesizers [5].

All of the modules used in the hardware synthesizer are included in Synthedit. Figure 3 shows the connections between modules used to create the software version of the hardware synthesizer and in figure 4 the graphical interface and related controls are shown.

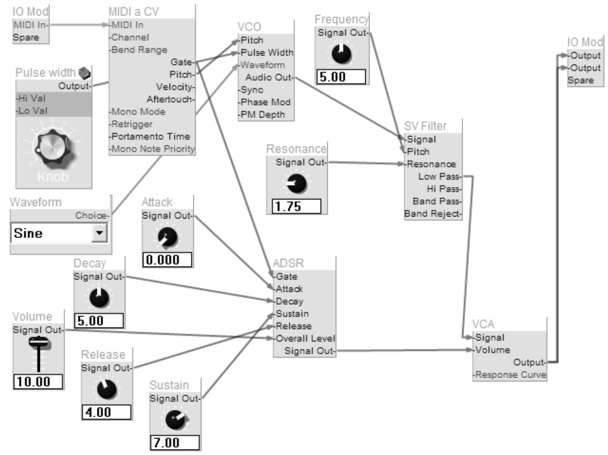

Figure 3. Modules used in the software synthesizer and their connections.

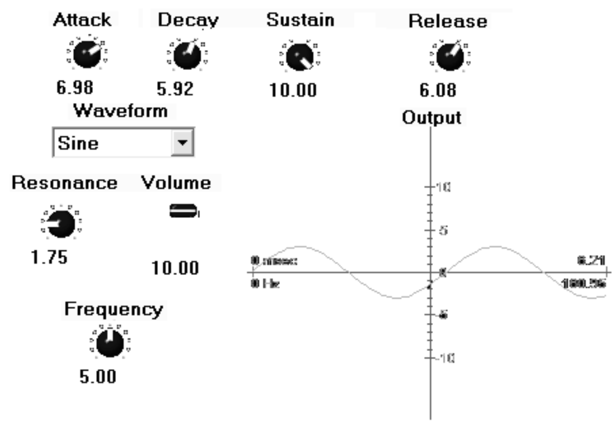

Figure 4. Graphical interface for the software synthesizer.

3 Hardware synthesizer

The hardware synthesizer consists of the same modules used in the software synthesizer, the implementation of each of these hardware modules required a detailed analysis, to do so, each circuit was divided in different blocks so that its use could be easily understood.

It is important to mention that the VCO uses a 1 Volt per octave scale, which is a common characteristic of hardware synthesizers since it is more convenient to represent a large number of notes with a reduced voltage range. This is a logarithmic response where, for example, a middle C note (C3) is represented by a voltage V, then the C note one octave above (C4) is represented by a voltage V+1. This scale is primarily used in the VCO, this is the module that receives the control voltage from the microcontroller and passes it through a voltage to current converter with a logarithmic relation. This current determines the frequency of oscillation for the sawtooth signal, which is the main waveform used as the basis for generating the other waveforms.

The control voltage that arrives to the oscillator is known as height or pitch and, as was previously stated,

978-1-4244-5353-5/10/$26.00 ©2010 IEEE

.

determines the frequency of the waveform and, in turn, the note being produced.
The microcontroller interprets the MIDI notes using a MIDI library that allows the Arduino to process the incoming data and turn them into a usable voltage for the oscillator [6].

The envelope generator uses operational amplifiers to receive the gate and trigger signals sent by the microcontroller. Both signals will be set whenever a note is played. The difference is that the trigger signal will always have a length of 100 ms and the gate signal will be held high as long as a key is pressed.

The envelope is generated using D-type flip flops, Schmidt triggers and AND gates. They are used in order to model each one of the ADSR parameters and activate them in the corresponding order. Using potentiometers the ADSR values can be independently adjusted.

The VCF is a state variable filter which uses OTAs (operational transconductance amplifiers) as voltage-controlled elements. The incoming voltage controls the cut-off frequency for the lowpass and highpass filters, and the center frequency for the bandpass filter. There is also a set of potentiometers for manual adjustment of the frequency and resonance of the filter.

The VCA is a relatively simple circuit that uses op-amps and OTAs to control the amplification of the incoming signals effectively modulating the amplitude of the VCO signal with the ADSR envelope.

4 Tests and results

Since the software synthesizer employs the same modules that the hardware synthesizer uses, a precise comparison between the two can be obtained.
In this project a step by step comparative was made for each one of the modules, for example, the signals produced by each of the voltage controlled oscillators in both synthesizers can be viewed using an oscilloscope. These synthesizers produce the four proposed waveforms.

In figure 5 a sawtooth waveform is produced by both oscillators at different frequencies, the signal to the left shows the hardware synthesizer waveform and the image to the right is the one obtained by the software synthesizer using the visualizing tool that Synthedit provides.

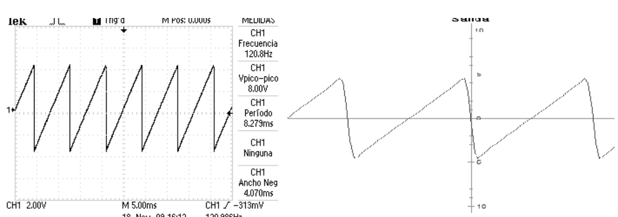

Figure 5. Sawtooth waveform obtained from a hardware (left) and software (right) synthesizer.

The voltage controlled filters used by both synthesizers produce the same results; both have the same control signals and the same set of output ports. In figure 6 the resulting signal of both filters with different levels of resonance is shown.

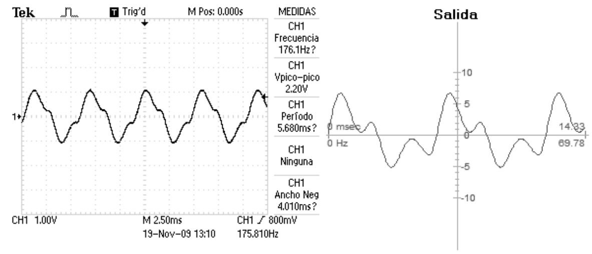

Figure 6. Output signal of both low pass filters with different resonance values, hardware VCF is shown on the left and software VCF is on the right.

The voltage controlled amplifier and the ADSR envelope generator in both synthesizers can be tested together since the amplifier takes the produced signal from the ADSR to give an amplitude control of the oscillator or filter waveform, figure 7 shows the results obtained in both synthesizers using similar ADSR values.

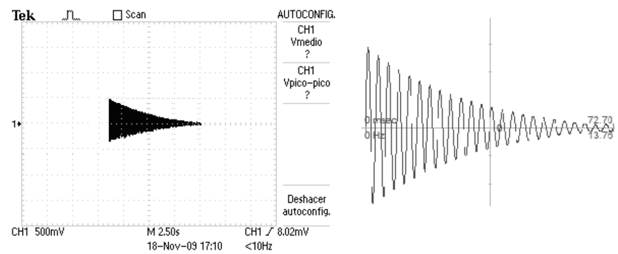

Figure 7. Output signal of the voltage controlled amplifier using similar envelope values.

5 Conclusions

The pictures presented here show the big similarities between both implementations of a synthesizer and this work forms the basis for a more in-depth research.

Future work could include more advanced modules, like an LFO (low frequency oscillator), a step sequencer, and a few analog effects (like chorus or saturation) and a polyphony scheme involving several VCOs, this can result in a more complex synthesizer and would allow further comparison between hardware and software implementations.

6 References

- [1] Devarahi, “The complete guide to Synthesizers”, Editorial Prentice-Hall, 1982.
- [2] Crawford A. James, “Frequency synthesizer design handbook”, Editorial Artech House, 1994.
- [3] The Sonic Spoth, “types of synthesis”. Found November 2008. http://www.sonicspot.com/guide/synthesistypes.html#additive
- [4] Réynes Gaspar Vidal, “El sintetizador”, found November 2008, http://www.aulaactual.com/especiales/sintetizador/sintetizador.htm.
- [5] H.G. Fortune, Peter Schoffhauzer y David Haupt, “Visual VST/I Programming”, Capitulo 1, Wizoo Publishing GmbH, 2007.

978-1-4244-5353-5/10/$26.00 ©2010 IEEE 313

.

- [6] Willman T. Timothy, “Arduino library”, found August 2009. http://timothytwillman.com/itp_blog/?page_id=240.

978-1-4244-5353-5/10/$26.00 ©2010 IEEE