

# **Synthesis**

Fergus Duniho

**COLLABORATORS**

	<i>TITLE :</i> Synthesis		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Fergus Duniho	October 10, 2022	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>Synthesis</b>	<b>1</b>
1.1	TABLE OF CONTENTS . . . . .	1
1.2	Preface . . . . .	1
1.3	Introduction . . . . .	2
1.4	Creating Harmonics . . . . .	3
1.5	The Fundamental Method . . . . .	3
1.6	The Doubling Method . . . . .	3
1.7	The Pasting Method . . . . .	4
1.8	The Stretching Method . . . . .	5
1.9	Mixing Harmonics . . . . .	5
1.10	Choosing Which Harmonics to Mix . . . . .	5
1.11	How to Mix Harmonics . . . . .	6
1.12	Working Around Limitations . . . . .	6
1.13	Stringed Instruments . . . . .	7
1.14	Plucked Strings . . . . .	7
1.15	Hammered Strings . . . . .	8
1.16	Wind Instruments . . . . .	8
1.17	Realistic Sounds . . . . .	8
1.18	Postscript . . . . .	9
1.19	About the Author . . . . .	9
1.20	Me and Music Synthesis . . . . .	9
1.21	Amiga Software by Me . . . . .	10
1.22	My Addresses . . . . .	11

---

# Chapter 1

# Synthesis

## 1.1 TABLE OF CONTENTS

How to Create Realistic Synthetic Sounds in MED

by

Fergus Duniho

Preface

BASIC HARMONIC SYNTHESIS TECHNIQUES

Introduction

Creating Harmonics

Mixing Harmonics

ADVANCED HARMONIC SYNTHESIS TECHNIQUES

Stringed Instruments

Wind Instruments

Realistic Instruments

Postscript

About the Author

## 1.2 Preface

In this document, I share some of my knowledge on how to create realistic sounding instruments with the synthsound editor found in MED and OctaMED. My knowledge is nowhere near complete, and there are many gaps in

---

this article. If you know something on the subject that I didn't cover, please tell me though email. I would like to compile knowledge from other people and eventually release an updated version of this document.

I currently use OctaMED 4, and any specifics I say about the interface are for that version. The synthsound editor for MED 3.22 is essentially the same. I can't vouch that earlier versions of MED have all the features I refer to in this document. The interface for OctaMED 5 is very different, and it doesn't seem to have the "STRECH:" gadget that OctaMED 4 has.

MED is a freely distributable music composition program by Teijo Kinnunen. OctaMED is the more powerful commercial successor of MED. Only OctaMED is still in development. You should be able to find MED and demo versions of OctaMED on the Aminet and various BBSs.

Throughout this document I talk about harmonics. It will be useful to know more about sound and harmonics than I tell you in this document. But it isn't necessary to begin creating realistic synthsounds. For the sake of exposition, I speak as though there are only sixteen harmonics. But there are really a lot more than just sixteen. However, with a maximum length of 128, you can't generate anything beyond the sixty-fourth harmonic of the default square wave.

If you got this document from an archive called `harmonix.lha`, you will also find example synthsounds and sets of harmonics that will help in creating realistic synthsounds. If you're reading on usenet, you can find the examples and harmonics sets in `harmonix.lha` on the Aminet or from FileWorks in Rochester, NY.

### 1.3 Introduction

The nice thing about the synthsounds in MED is that they take up so little space. Unfortunately, they get little use, because they usually sound more like Commodore 64 sounds than they do like real sounds. The MED synthsound editor gives you the same standard wave forms that you get with the C64's SID chip: the triangle wave, the square wave, the sawtooth, and white noise. It also gives you a simple sine wave and a backwards sawtooth. But it doesn't give you any straightforward and easy way to generate more complex waveforms and so to create more realistic sounds. So it seems as though you can't really do much with MED synthsounds.

Nevertheless, however limited MED's synthsounds are, they are not nearly as limited as they first appear to be. The synthsound editor gives you the means to generate many waveforms besides its handful of default waveforms. But before I explain how, you should know about harmonics. A complex sound wave is made up of a combination of various frequencies, known as harmonics. Each harmonic is an integral multiple of a fundamental frequency, i.e. the fundamental harmonic. For example, the second harmonic is twice as fast as the fundamental harmonic, the third is thrice as fast, etc. Also, the second harmonic is an octave above the first, the fourth is two octaves above the first, the eighth is three octaves above, and so on. When we combine harmonics, we get even more complex sounds.

---

## 1.4 Creating Harmonics

There are basically three different ways to create a smooth sounding harmonic with MED's synthsound editor. One is fundamental to creating any harmonic, and the other two are recursive methods for creating harmonics even more complex than the fundamental method alone will allow. The two recursive methods are doubling and pasting. Through combinations of these three methods, you can create rich and realistic sounds. There is also a fourth method for creating harmonics, which I call stretching, but I don't recommend it.

The Fundamental Method

The Doubling Method

The Pasting Method

The Stretching Method

## 1.5 The Fundamental Method

Before you can use the recursive methods to generate complicated waveforms, you must start with a waveform. MED's synthsound editor makes this easy for you. It has defaults for such standard waveforms as the sine wave, the square wave, the triangle wave, the sawtooth (both inclining and declining), and white noise. MED also has gadgets that let you manipulate your waveform. For example, you can reverse the sine wave to get a cosine wave. You can also use the "MIX" gadget to blend together different waveforms, such as the sine and square, for instance. If you don't want to rely on the defaults, you can even draw a waveform with the mouse. But you are unlikely to get a smooth sounding wave if you draw it. [N.B.: You may have to draw the square wave for lengths other than 128. OctaMED 4, at least, gets the square wave wrong for lengths other than 128. You can check whether a square wave is accurate by clicking on the center gadget.]

In short, you have to begin by creating a waveform that will serve as your fundamental harmonic. Relying on the defaults is the simplest way to do this. But you don't have to restrict your fundamental harmonic to the defaults that MED gives you. Once you have your fundamental harmonic, you are ready to use the recursive methods to generate its overtones, i.e. the other harmonics.

## 1.6 The Doubling Method

This is the easier of the two recursive methods. Once you already have some harmonic, you can generate the harmonic with twice its frequency by doubling it. Just select the "RANGE ALL" gadget and then select the "DBL" gadget. Of course, be sure that you are working on a copy, or you will lose your initial harmonic.

---

This method is particularly useful for generating the second, fourth, eighth, and sixteenth harmonics of any fundamental harmonic. But it is also useful for generating the sixth and twelfth harmonics after you have used the pasting method to generate the third. In general, it is useful for generating any even numbered harmonic once you have generated a harmonic with half its frequency. So you can use it to get the fourteenth from the seventh, the tenth from the fifth, etc.

The main limitation of this method is that it will not generate odd harmonics, i.e. the third, fifth, etc. To generate odd harmonics, you should rely on the pasting method.

## 1.7 The Pasting Method

The pasting method will let you generate almost any harmonic. To use this method, you must first choose a length for your harmonic that is an even multiple of the harmonic you want to create. For example, its length must be divisible by six if you want to create the third harmonic. This is so for two reasons. First, MED will not let you choose an odd length. That is why the length must be even. Second, your harmonic will be too short or too long if the number of the harmonic doesn't divide into the length. This would cause your harmonic to sound scratchy.

Once you have settled on a length for your harmonic, divide the length by the number of the harmonic you want to create, and create a fundamental harmonic of that length. For example, if you want to create a third harmonic with a length of 120, first create a fundamental harmonic with a length of 40.

Depending upon what you're using as a fundamental harmonic, there are two ways to do this. If your fundamental harmonic is one of the default waveforms, set the length, in this case, to 40. Then click on the gadget for the default. This will create a copy of that waveform with a length of 40. This method will also work if your fundamental waveform is some variation of a default. Just make the same variations after you select the default. For example, if your default is the cosine wave, just select the sine wave and reverse it. Once you have this shortened version of your fundamental harmonic, select the "RANGE ALL" gadget followed by the "COPY" gadget. Then change the length back to the original length and select the "PASTE" gadget until you've completely filled your waveform with copies of your fundamental harmonic.

However, this method will not work if you originally drew your fundamental harmonic. And it will be too inconvenient to use if your fundamental is itself a complex harmonic. For example, suppose that you want to create a ninth harmonic with a length of 120. Because  $120/9 = 13.333333333333$ , you can't do this directly, i.e. you cannot generate the ninth harmonic if you choose the first harmonic as your fundamental harmonic. But you can create the ninth harmonic indirectly, as the third harmonic of the third harmonic. When you do this, you will have to use the third harmonic as your fundamental harmonic.

So, to create the ninth harmonic here, you want to generate a copy of the third harmonic with a length of 40. But how will you do that? You

---

can't just set the length to 40 and copy over the third harmonic. If you tried that, MED would just change the length back to 120. However, you can squish your third harmonic into a third of the space with the "STRETCH:" gadget. First of all, click on the left arrow gadget nearby this gadget. This will set both the start and the end of the range to the beginning of the waveform. Then enter into the "STRETCH:" gadget a number which is equal to the difference between the length you want your fundamental harmonic to be and the length that it currently is. In this case, enter 80, which is  $(120 - 40)$ . This will squish your fundamental harmonic into the remaining space.

Next, look to the right and you will see the "RANGE START" and the "RANGE END" gadgets. You will also see that both the start and the end of the range are at the start of that part of the waveform you want to make a copy of. For the sake of accuracy, use the "RANGE END" gadget to bring the end of the range to the end of the waveform. Copy the range by clicking on the "COPY" gadget. Then set the end of the range to the start of the waveform and click on the "PASTE" gadget enough times to fill up the waveform.

## 1.8 The Streching Method

The stretching method is an alternative to the pasting method. But I do not recommend it. The harmonics I have generated with this method have sounded scratchy. So I won't describe it in detail. In brief, it works like this. Using other methods, usually doubling, create a harmonic that is higher than the one you want to create through stretching. For example, first create the eighth harmonic if you want to create the seventh. In this case, position the cursor at the end of the seventh wave and stretch the first seven waves to the end of the waveform. Do similarly for other harmonics you want to create. Once again, I don't recommend this method, but you may want to try it for yourself.

## 1.9 Mixing Harmonics

Choosing Which Harmonics to Mix  
How to Mix Harmonics  
Working Around Limitations

## 1.10 Choosing Which Harmonics to Mix

Before you begin to mix harmonics, you should bear in mind that it is best to mix harmonics that have the same length. Otherwise, they will not blend together evenly, and you may not be happy with the result. Unfortunately, there is no way to generate all the harmonics between the first and the sixteenth for any particular wavelength without some of them sounding scratchy. So this puts a limit on which harmonics you can mix. For example, you won't be able to mix together the eleventh and thirteenth harmonics. The best you can do is create sets of all the harmonics you can generate for different lengths. Here are some lengths I've found most



useful, along with the harmonics you can generate for them:

78 - 1, 2, 3, 4, 6, 8, 12, 13, 16

90 - 1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, 16

120 - 1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, 16

126 - 1, 2, 3, 4, 6, 7, 8, 9, 12, 14, 16

## 1.11 How to Mix Harmonics

You should first notice that the synthsound editor has two waveform panels. Between them there is a "MIX" gadget with an arrow pointing from the right panel to the left. Whenever you click on this gadget, MED will mix the waveform on the right with the waveform on the left, and it will replace the waveform on the left with the combination of the two waveforms. So first decide which harmonics you want to mix. You could create each harmonic as you mix them, but it is easier to have them already made. What I have been doing is creating synthsounds with 17 waveforms. For each waveform from 1 to 16, I leave it blank or create the harmonic that corresponds to the number of the waveform. I then do all my mixing in waveform #0. As I move through the 17 different waveforms, each waveform appears in the left panel.

I start by copying the first waveform I want to mix to the right panel. I then go back to waveform #0 and copy the right waveform to the left. For each subsequent waveform that I want to mix, I copy it the right panel, go back to waveform #0 and mix the two waveforms by clicking the "MIX" gadget. I repeat this process until I've mixed all the waveforms I want to mix. If you want to mix together different sorts of harmonics, such as sines with squares, you'll have to follow a slightly different procedure. But the general principle will be the same. Do all your mixing on the left, and put each waveform you want to mix in on the the right.

## 1.12 Working Around Limitations

There are a couple limitations you should bear in mind. First, not every harmonic will mix smoothly with every other harmonic. This is mainly a problem with some of the higher harmonics, because synthsounds don't have enough space to represent them accurately. If you want to mix two harmonics that don't mix together smoothly, try creating the same harmonics for a different length and mixing those two together.

Second, the synthsound editor does not offer you genuine, full-fledged harmonic mixing. With full-fledged harmonic mixing, such as Synthia offers, you specify all the harmonics you want to mix together, and the computer then mixes them all together. But with the synthsound editor, you can mix only two waveforms at a time. So, if you want to mix together, say, the first, second, and fifth harmonics, you first have to mix together two of them and then mix in the other with the result. This will not give you quite the same mixing as you would get by mixing them all at once.

---

Moreover, the order in which you mix harmonics will make a difference.

Let me illustrate these points by showing you different ways to average numbers.

$$(1 + 2 + 5) / 2 = 4$$

$$(((1 + 2) / 2) + 5) / 2 = 3.25$$

$$(((1 + 5) / 2) + 2) / 2 = 2.5$$

$$(((2 + 5) / 2) + 1) / 2 = 2.25$$

Based upon these examples of averaging numbers, I would hypothesize that you can get the closest approximation to genuine mixing by starting with the lower harmonics and finishing with the higher harmonics.

## 1.13 Stringed Instruments

Plucked Strings

Hammered Strings

### 1.14 Plucked Strings

When a string is plucked, it generates many harmonics. Most of these harmonics, especially the higher ones, quickly begin to fade away, and the string settles into its natural vibration. The higher harmonics fade out almost completely, and the sound soon stops.

To simulate this, you have to decrease the volume and change the waveforms of your synthsound. To do this, you need to use the synthsound programming language. I'll just refer you to the documentation to familiarize yourself with it. After reviewing the documentation, you should know how to change the volume of your sound. What I'll tell you here is how to change the harmonic structure of your sound.

From reading the documentation, you should have a general idea about how to change waveforms. But you also need to know how to generate the right waveforms. Fortunately, MED makes it easy for you. To simulate a plucked string, you essentially need to generate only two different waveforms by hand. One is the initial waveform of the plucked string sound, and the other is the final waveform of the sound.

Generally, the initial waveform should be the final waveform with some more harmonics mixed into it, usually higher harmonics. So to create these two waveforms, start with the final waveform. Mix in the harmonics you want it to have and copy it over to the synthsound you are creating. Then to create the initial waveform, just continue to mix more harmonics into the waveform you have already started to create. When you are

finished, copy it over to ANOTHER waveform in the synthsound you are creating.

In order to have a smooth transition between the waveforms, add some waveforms between your final and initial waveforms, then use the "TRANSFORM:" gadgets to create the transitions. Enter the numbers for the initial and final waveforms and use the "G.TRN" gadget to create a gradual transition between them.

## 1.15 Hammered Strings

When a string is struck with a hammer, the upper harmonics linger longer and decay more slowly than they do when a string is plucked. As with plucked strings, the volume decays rapidly.

Creating a hammered string sound is similar to creating a plucked string sound. There are at least a couple different ways you may modify the procedure I gave for plucked strings to simulate a hammered string. One way is to include more upper harmonics in the final waveform of the sound. The other way is to create an intermediate waveform between the final and initial waveforms, which still has many of the upper harmonics. It would generally be best to create this waveform after you do the final waveform and before you do the initial waveform. It should generally be the same as the final waveform with some more upper harmonics, but not all the upper harmonics in the initial waveform.

## 1.16 Wind Instruments

The sound of some wind instruments is generally the opposite of a plucked string sound. The plucked string starts out loud and with a complex harmonic structure, and it fades away in both volume and harmonic structure. Some wind instruments start softly and build in both volume and harmonic structure.

Creating such a wind instrument is much like creating a plucked string, with just some minor variations. First, it should start off quietly and increase in volume. Second, your final and initial waveforms should be reversed from what they would be for a plucked string.

Woodwinds tend to have many odd harmonics. Unfortunately, MED doesn't make it easy to mix together many of the higher odd harmonics, because many of them are also prime numbers. However, there is another way to get odd harmonics. The square wave is full of odd harmonics. So you should be able to get more odd harmonics by mixing in variations on the square wave.

## 1.17 Realistic Sounds

---

The sounds from real instruments are not just combinations of pure sine tones. They also have minor imperfections and various non-harmonic overtones. You can generate non-harmonic overtones by mixing in waveforms besides sine wave harmonics. Furthermore, most instruments do not simply emit a single constant waveform. Real instruments change their waveforms over time, usually in some sort of natural progression. For example, certain harmonics may fade out or grow stronger.

## 1.18 Postscript

If you have found the information in this document useful, please send me some of the synthsounds you make. After I've collected a hundred or so, I'll put a collection of them on the Aminet. So send me the instruments you would be most proud to have other people know you made. You can send them to me via Internet email or Fidonet netmail. If you have anything you want to tell me about your instrument, I will include it in a text file that will accompany the collection of synthsounds. This file will also give a list of who synthesized which sounds.

## 1.19 About the Author

Me and Music Synthesis

Amiga Software by Me

My Addresses

## 1.20 Me and Music Synthesis

I have had an interest in music synthesis for most of my life. When I first got my Commodore Vic-20, which, for reasons I don't recall, I had sold my piano in order to buy, I wrote programs that let me use it as a musical instrument. It wasn't much compared to what I later got ahold of, but I did what I could with its four voices that each generated a single tone.

When I was in college, I finally replaced my Vic-20 with a Commodore-128. I wrote a BASIC program that turned it into a musical keyboard, but it was too slow. So I eventually wrote a BASIC synthesizer program which modified and called a machine language program that wrote directly to the SID chip whenever a key was pressed. The sounds I got from this program were far superior to what BASIC gave me. It was a multi-timbral synthesizer, which let me create complex sounds by using all three voices together. I also learned about scales, and I made my program use any scale I wanted: the just scale, the meantone scale, or any variation on the equally tempered scale. I could even have different voices on different scales. This allowed for interesting ring modulation

---

when I kept one of the voices silent. I never released this program, because I never hooked my C-128 to a modem.

After I had been to graduate school for a year, I replaced my C-128 with a used Amiga 500. I chose to upgrade to an Amiga because I wanted to upgrade to a computer with even better sound capability than the C-128. I had considered the Atari ST, but somebody told me it also used the SID chip. Of course, I never even considered the Mac or the PC as options.

One of the first pieces of software I bought for my Amiga was Synthia by the Other Guys. This is a set of five music synthesis programs, each using a different form of synthesis. What most impressed me was that the Amiga had realistic drum sounds. The C-128 had a default instrument called "drum," but it sounded nothing like a drum. A few months later, I ordered a Fred Fish disk with MED on it. After I got a modem and moved to Rochester, where I'm in another graduate program, I began collecting music modules. Every now and then I would dabble with Synthia or with MED's synthsound editor.

I've recently figured out how to easily use OctaMED to record music in real time. Maybe one day I'll begin writing music modules. I can spontaneously compose interesting monophonic music with OctaMED, but I can't yet do much more. I probably need some discipline and training. Maybe I should take some courses at Eastman. But I've always frustrated my music teachers. Maybe that's why I don't know as much as I would like to.

## 1.21 Amiga Software by Me

I once released a collection of some of the synthsounds I had created. This was before I discovered how to make realistic sounding synthsounds. The collection includes some replicas of Commodore 128 default instruments and some stuff with square waves. It is called SynthSounds.lha. I think I released it to GENie. I don't recall if I ever put it on the Aminet. I think I've also released a collection of some of the instruments I created with Synthia. But I don't remember what I called it.

I released a set of macros for Mill Dillon's text editor, DME, which I called DMD (Duniho's Macros for DME). DME has since been superceded by XDME, which I have yet to release anything for.

I have released a spelling checker called AlphaSpell and a pagination program called Paginate. AlphaSpell comes with DMD. I don't recall releasing it separately.

My best known and most popular program is the DDLI. This is a personality sorter like the Myers-Briggs Type Indicator. It is available in both Amiga and PC versions. The Amiga version is ddli341.lha, and it is available from the aminet in misc/misc/, from the FileWorks BBS at 1:2613/278, and (eventually) from Fred Fish. The PC version is ddli341.zip, and it is available from the SimTel Software Repository in msdos/educatin/ and from the Neutral Zone BBS at 1:2613/336.

## 1.22 My Addresses

Internet: fdnh@troi.cc.rochester.edu

Fidonet: Fergus Duniho @ 1:2613/278

Fergus Duniho @ 1:2613/336

US Mail: Fergus Duniho  
1095 Genesee St.  
Rochester, NY 14611-4148

---