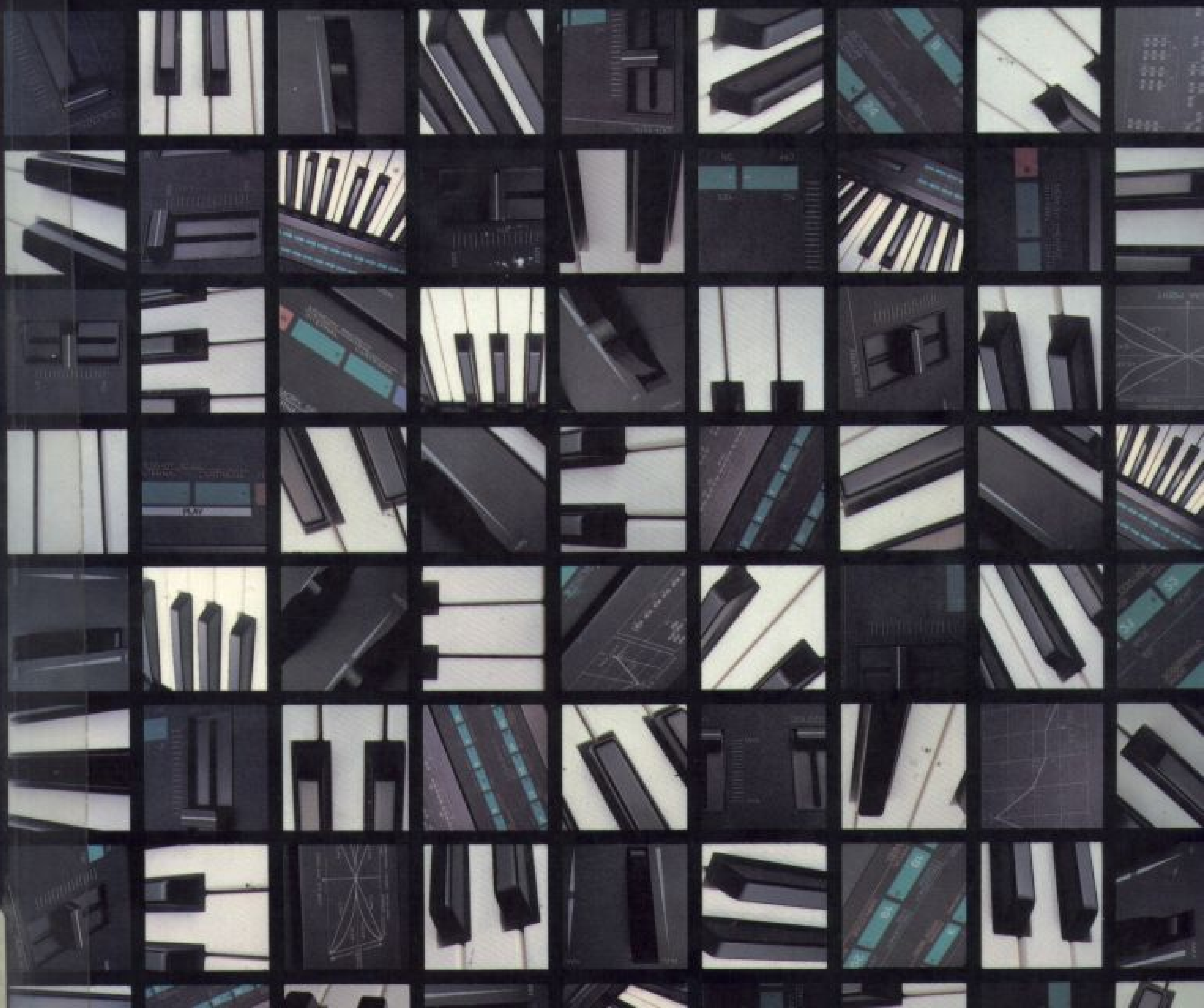
Synthesizer Basics.

The definitive beginner's guide by Dean Friedman. Tells you everything you need to know to really understand, progra
and start using your synthesizer. Also contains special sections on the Yamaha DX7, Sampling Devices and MIDI.



Synthesizer Basics.

by Dean Friedman.

CONTENTS

PREFACE

Before sex education in schools became commonplace, most of us learned the facts of life on the playground, on the streets, or in the back seat of a Chevy. At the time there was simply no other place to acquire this crucial knowledge. Our parents were reluctant to explain the more prurient details of the subject and, the fact is, even they weren't sure what was what. This left us with a seriou lack of information.

The same sorry state of affairs exists today in the world of synthesis. Synthesizer manufacturers have enough trouble translating their instruction manuals into some semblance of English, let alone attepting to explain the theory behind the instruments. The truth of the matter is that the entire field of sound synthesis is still so young there is still only a handful of books on the subject and the few of those that are any good are quickly becoming obsolete.

In any case, I wouldn't want an entire generation of synthesizer players to have to spend their formative years hopelessly misinformed just because the only place they could learn about oscillators and filters was huddled over some outdated electronics manual in the school bathroom during recess.

In the following pages, you will find out all you need to know about the birds and bees (or in this case the buzzes and tweets) of synthesis. One thing to remember, though, is that—as helpful as this book may be—in synthesis, as in sex, it is very important that you get as much "hands on" experience as possible. Got that? Good. Keep reading.

ACKNOWLEDGMENTS

I would like to acknowledge the writings of David Crombe, whose book *The Complete Synthesizer* greatly aided my own understanding of synthesis; also the writing staff of *Keyboard Magazine,* a periodical that continued to be an invaluable resource for any synthesist; the folks at Sam Ash for providing access to many of the synthesizers in this book; Jonny Mann for his advice and feedback and for letting me play with his neat toys; all the folks at New England Digital and also the folks at Music Net; Robert Sheldon of the National Museum of American History for his help in locating information about the Telharmonium; my friend Brett, who still actually believes I'm going to give him back his IBM PC when I've finished this book; and finally Clarabelle, and albino cockatiel, who collaborated with me on my last book but is currently busy eating the molding on the kitchen cabinets. Thanks everyone!

INTRODUCTION

The biggest problem with trying to write a book about synthesizers is this: They are all different. No single approach is going to be successful with every model. Fortunately, there are enough basic similarities among synthesizers so that it is possible to organize this material both to provide you with an understanding of the basic principles of synthesis and to prepare you for negotiating the different makes of synthesizers you are liable to come across.

I point this out because, despite their similarities, the disparity among features of the different makes of synthesizers is still so great that it will remain an important factor throughout this book. You'll constantly see terms such as *usually* or *on most systems* or *in many cases* or *some* when I'm describing a particular feature. Don't let this dismay you. In an effort to overcome this problem of the diversity of features, I've tried to come up with a typical "composite" synthesizer. This typical synthesizer will be used in diagrams to give you an example of the kinds of features you are likely to find on most (but not all) synthesizers.

Another big problem you should be aware of is the one involving terminology. Different manufacturers use different terms to describe the same component. This can get needlessly frustrating and confusing, but wherever I can I try to give the alternate terms for a particular feature.

I've tried to present the material in such a way that whether you own a synthesizer, or regardless of its make, you will still be able to get something out of this book.

Exercises are provided throughout the text for those of you with access to a synthesizer; but, even if you don't have one, I strongly recommend you read through the exercises because they do include occasional hints and suggestions that are supplementary to the main text and can help to give you another perspective on the theory being covered. (Note: These exercises are designed for analog synthesizers only—that is, all synthesizers except the Yamaha DX7 and the Casio CZ101.)

In order to execute some of the exercises, it will be necessary for you first to put your synthesizer into what I'll call *neutral*. Neutral is a state in which your synthesizer is set up to generate its most basic sound—a simple waveform. This will enable you to experiment with basic sound principles without having advanced modulations and effects getting in your way. Again, devising a formula for arriving at neutral won't be easy—given the vast differences among synthesizers—but we'll give it a try. (Note: All values for control settings will be given in the range of 0–10, where 0 = off or minimum, and 10 = on or maximum.)

GETTING NEUTRAL

1. Choose a preset sound that most closely resembles an organ. In other words, find the plainest, most boring sound to start with.
2. If you have an LFO (also called *sweep* or *mg*), find it and turn it off (to zero).
3. If you have sync, cross modulation, chorus, ensemble, or portamento, turn 'em off. Turn them all off.
4. If you have two oscillators per voice, find the **mix** or **balance** control and turn it so that you can hear only one of the oscillators.
5. Set the ADSR envelope(s) so that attack is at zero, decay is at zero, sustain is at 10 (maximum), and release is at zero.

6. Set the **filter cutoff** control at 10, the resonance at 0, and the envelope amount at 10.
7. Choose a waveform in the oscillator section.

If you've followed this recipe, then with any luck what you should be left with is a simple unadorned waveform without any effects or modulation in it. From this point you should be able to proceed with Exercise 1—examining the different types of waveforms (see p. 17).

If for some reason this recipe has not achieved the desired result, don't worry. It's not a problem. Just do the next best thing. Go back to the first step in the recipe and choose, from among the presets, the plainest and simplest sound you can find. This should enable you to successfully complete all of the exercises. (Warning: Until the controls on your synthesizer are activated—moved slightly—they may not necessarily reflect their true settings. See "Activating the Controls" on p. 97.)

One more point before I let you go. More and more synthesizers today are *step programmable.* That is, instead of having a separate switch or knob for every function, you have to access each function via an alpha-numeric keypad. For those of you who don't have a synthesizer, or who own an analog synthesizer that is not step programmable, the subject will be discussed in due course. If, however, you own a step-programmable synthesizer (each function will have a number assigned to it as opposed to its own knob), you may want to read Chapter 5, especially the section titled "Step Programming," before you begin the exercises.

I'm sorry if these preliminary explanations seem a little complicated. Don't let it scare you off. The rest of this book will be easy as pie. (sure.) So, enough jabbering. You wanted to learn something about synthesizers, right? Well, here we go.

1

SOUND, WAVES, SYNTHESIZERS

Synthesizers, Video Games, and Computers

Synthesizers are a lot like video games. You plug them in and they light up and make all kinds of strange and peculiar noises. The only difference is that instead of using switches and knobs to subtly maneuver your spaceship through hordes of alien invaders you use similar knobs and switches (and keys) to manipulate *sound*.

Synthesizers are a lot like video games.

I use the video game analogy because they represent the first instance in which computers were used on intimate terms by normal everyday humans. Video games are interactive, fun, sophisticated computers—but which any 5-year-old kid can play. In fact, kids have a distinct advantage when learning video games because they don't "know" that they are using computers and are therefore not as intimidated by them.

Here's my point. Today's synthesizers are simply keyboards attached to a computer. Don't let the computer part scare you. It just shuffles information around and will do whatever you tell it to.

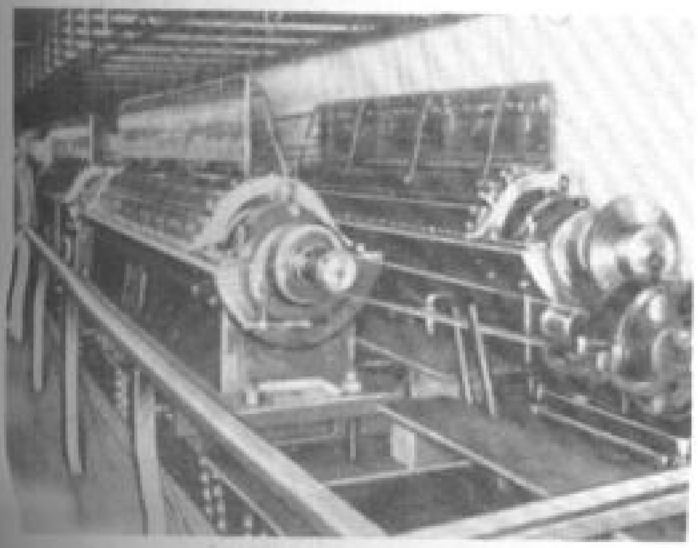
The First Synthesizer

In 1897 an inventor by the name of Thaddeus Cahill took out a patent on what is generally considered to be the very first electronic keyboard. It was called the Telharmonium. The Telharmonium was a polyphonic instrument with a touch-sensitive keyboard. Its sound was generated by a series of rapidly spinning alternators driven by banks of electronic motors that were so noisy they had to be housed in a separate room. It wasn't what you'd call portable either; it weighed in at about 200 tons and had to be hauled across the United States in six railway cars. The sounds generated by the Telharmonium were sent down regular telephone lines and then amplified at the receiving end by large paper horns.

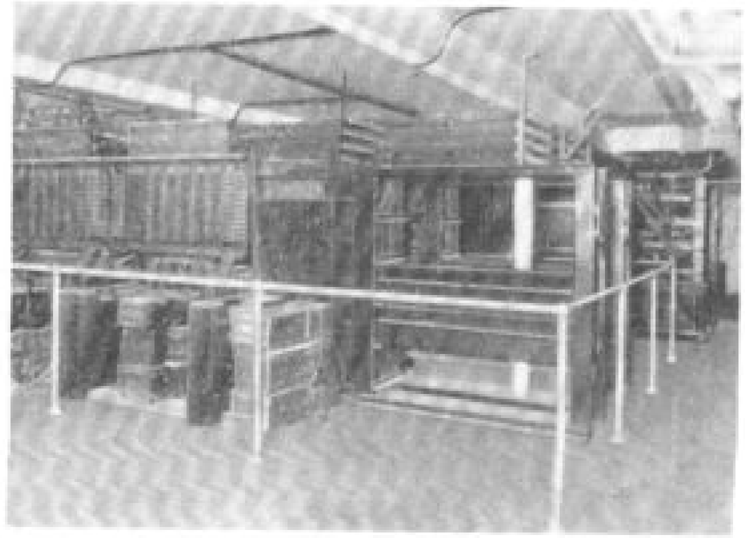
Cahill's dream was to transmit Telharmonium performances simultaneously over thousands of telephone lines into restaurants, hotel lobbies, and living rooms across the country—the original Muzak! Unfortunately Cahill's ambitious scheme went awry when two promoters, who had assisted in raising thousands of dollars from investors, suddenly disappeared . . . with the money. Due to these financial difficulties and the fact that the development of superior amplification systems and the "wireless" were gradually rendering the Telharmonium obsolete, the company eventually went bankrupt—but not before Cahill had demonstrated that electricity could not only generate light and run a motor but could create wonderful and exciting new music as well.

A little more than forty years later, in 1939, a gentleman by the name of Laurens Hammond introduced the first electric organ—an electronic keyboard based in part on the rotating disk system of the Telharmonium. The Hammond Organ was enormously popular and could soon be found in churches, recording studios, and homes around the world. It represented the first mass-produced, consumer-oriented electronic instrument.

The electric organ can be viewed as the predecessor o[f] today's synthesizer. Like the synthesizer it generates its sound using electricity. What distinguishes the electric organ from what we now refer to as a synthesizer is the fact that it is not programmable. *Programming* or *editing* refers to the process o[f] shaping and manipulating sounds electronically. While organs



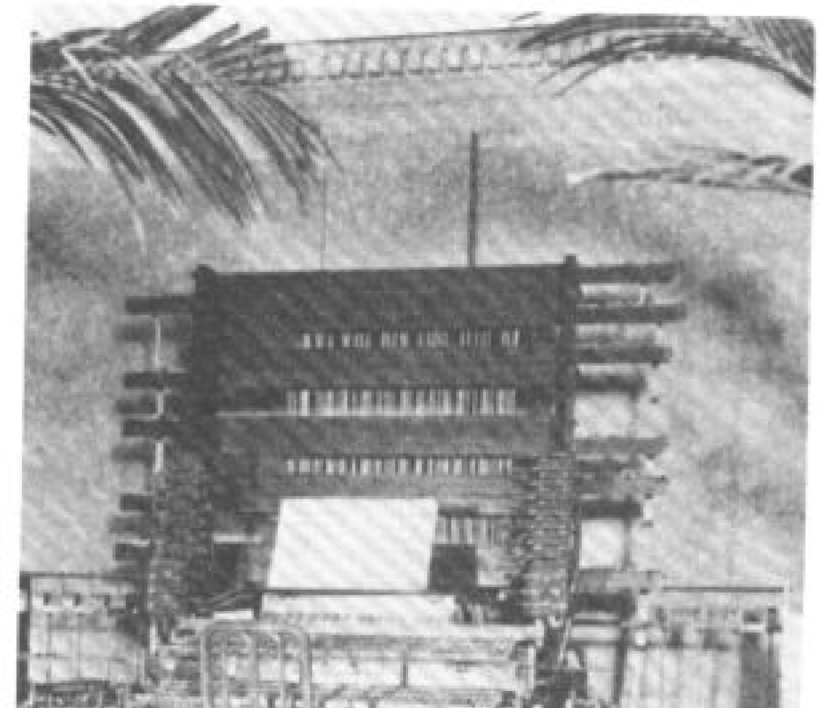
The Electric Current Generators



The Switchboard Room

These photographs show only a portion of the enormous Telharmonium—the world's first, and still the largest, synthesizer ever.

A Tone Generator



The Keyboard Console

do generate a host of different sounds electronically, they generally offer very few access points for actually going in and altering those sounds. Synthesizers, on the other hand, are designed specifically with programming and sound editing in mind. This ability to precisely mold and define the complex parameters of a sound is what synthesis is all about.

The first real synthesizer, as we know it today, was developed by a fellow named Bob Moog. What he did was take large and expensive electronic sound-generating components and replace them with small and inexpensive sound-generating components. He put them all into an elegantly simple and logical package and called it the Moog Modular System. This was the first commercially available synthesizer and was introduced in 1965. Six years later Moog had further refined and perfected his creation, making it smaller and more portable as well as less expensive, until finally he introduced what is still today one of the most popular synthesizers of all time—the Mini-Moog.

The original design of the Mini-Moog was so successful that to this day almost all synthesizers are based, at least in part, on that first system; and even though the Mini-Moog is no longer being manufactured, it is still used regularly in studios around the world.

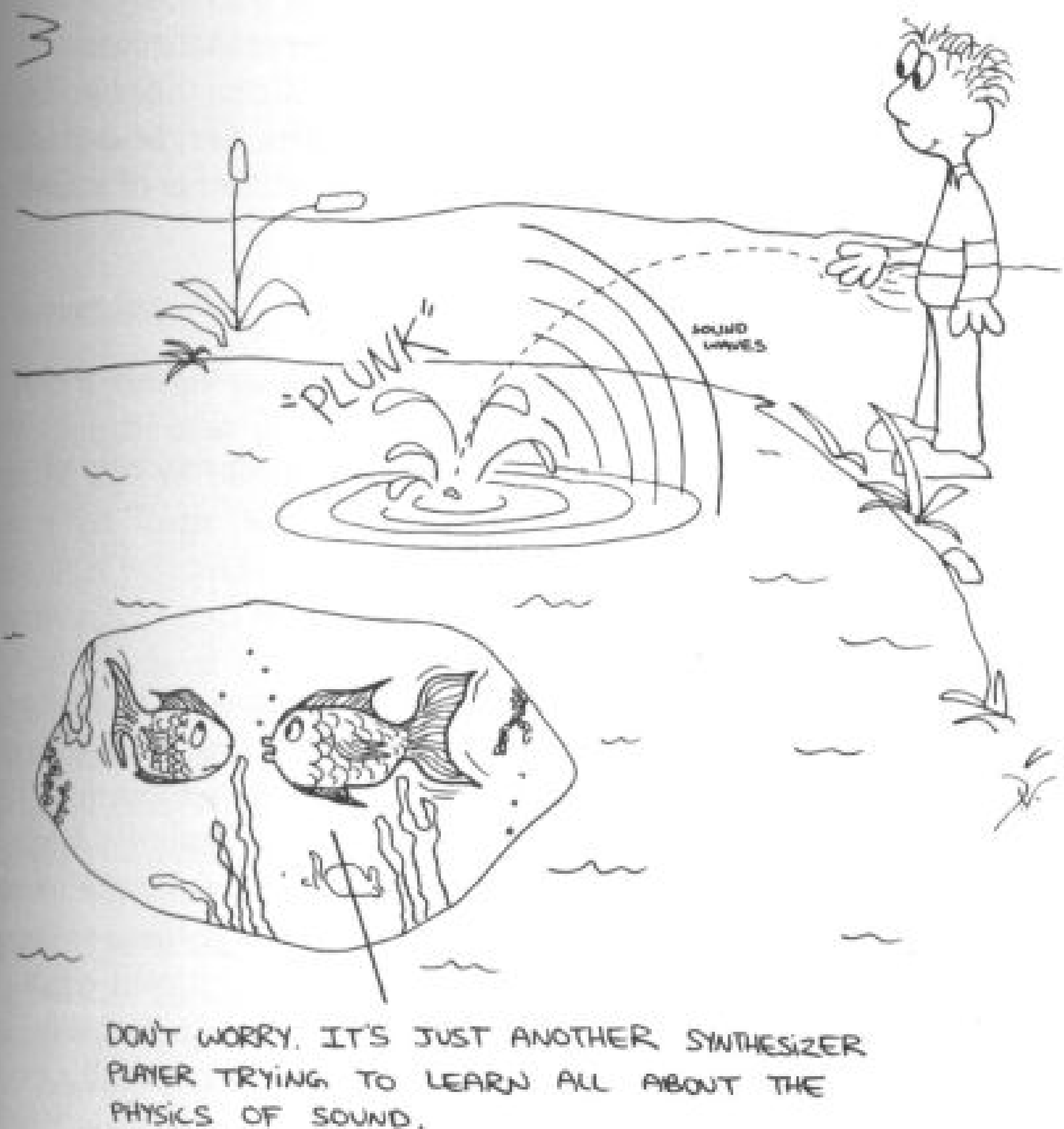
The Mini-Moog is a voltage-controlled synthesizer, that is, its sound is generated and modified by voltage-controlled electronic components such as oscillators, filters, and amplifiers. This type of synthesis is known as *subtractive synthesis*. Other types of synthesis are *additive synthesis* and *FM synthesis*. Each of these different kinds of synthesis will be explained in detail, but first, let's talk a little about sound.

SOUND AND WAVES

Drop a pebble into a pool of still water—*plunk*—and what happens?

You scare the fish. Also, ripples radiate outward from the center. The pebble, by briefly displacing a small amount of water, has set off a *vibration*. This vibration moves outward in all directions in the form of a *wave*. Keep in mind that it is not the water that is traveling across the surface of the pool. The molecules of water are simply moving back and forth. What you see is the energy created by the pebble's impact moving through the water in the form of a wave.

The same thing happens in the air. Remember that *plunk?* Just as the impact of the pebble set off vibrations in the water, similar vibrations are set off in the air. The molecules of

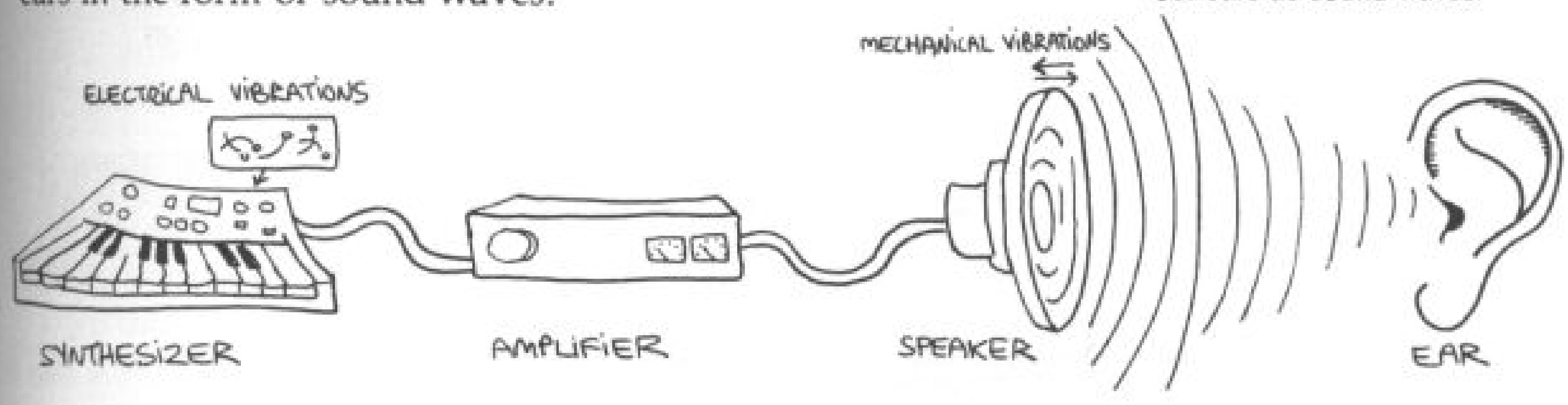


The physics of sound

air, just like the molecules in the water, move back and forth; and the energy created by the pebble hitting the water travels through the air in all directions in the form of waves. These waves are called *sound waves* because when they reach your ear you hear a sound—a *plunk*.

In a synthesizer electrical vibrations are set up in the circuitry. These electrical vibrations are sent to the speaker where they cause the speaker cone to move rapidly back and forth. Finally, just as with the pebble in the pool, the motion of the speaker cone sets up vibrations in the air that reach our ears in the form of sound waves.

Electrical vibrations set up in the synthesizer are converted by the speaker into air vibrations which eventually reach our ears as sound waves.



The process of synthesis involves generating and then modifying these sound waves. But before we can start tinkering around with these waves, we need to know a little more about what's inside them. In other words, we know that sound is made up of vibrations and that these vibrations travel in the form of sound waves; but just what, exactly, are the characteristics of these sound waves? What are the elements of sound?

The Elements of Sound

When you pluck a string, it causes the string to vibrate, which in turn causes the air around it to vibrate. The vibrations in the air are sound waves.

Traditionally, sound has been defined as being made up of three elements—*pitch, timbre,* and *loudness*.

Pitch

All musical instruments create sound by setting up vibrations in the air around them. When you pluck a string on a guitar, it causes the string to vibrate, which in turn sends vibrations through the air in the form of sound waves. Similarly, when you blow into a saxophone, the sound is created by the vibrations of the reed. The same thing occurs in a synthesizer except that instead of plucking a string or blowing on a reed what you are actually doing is tickling the electrons, so to speak; in other words, you are generating vibrations electronically.

The speed of these vibrations determines *pitch*. If the vibrations are slow, the pitch will be low. If the vibrations are fast, the pitch will be high.

A single unit of vibration is referred to as a *cycle*, and the number of cycles that occur in a second is known as the *frequency*. Another term used to describe the number of cycles in a second is hertz (Hz). In musical terms pitch is another word for frequency. When we say that a note played on a violin has a frequency of 440 Hz, we mean that that string is vibrating 440 times (or cycles) a second. And when we translate that technical jargon to the language of music, what we mean is that the pitch of the note that the violin is playing is A above middle C.

So, pitch equals frequency; hertz is another way of saying cycles per second; and all of these terms are simply a way of describing how fast or slow something is vibrating.

Timbre

Timbre is defined as the quality of a sound that distinguishes it from other sounds. Every voice or instrument has a different timbre, or tone color—for instance, the sound of an oboe has a different timbre from that of a clarinet. Even if both instruments are playing the same note, each has unique tonal characteristics that enable us to tell them apart. That is timbre. The main elements of sound that combine to create an individual timbre are called *harmonics*. The ratios and proportions in which these harmonics occur determine the timbre of a sound.

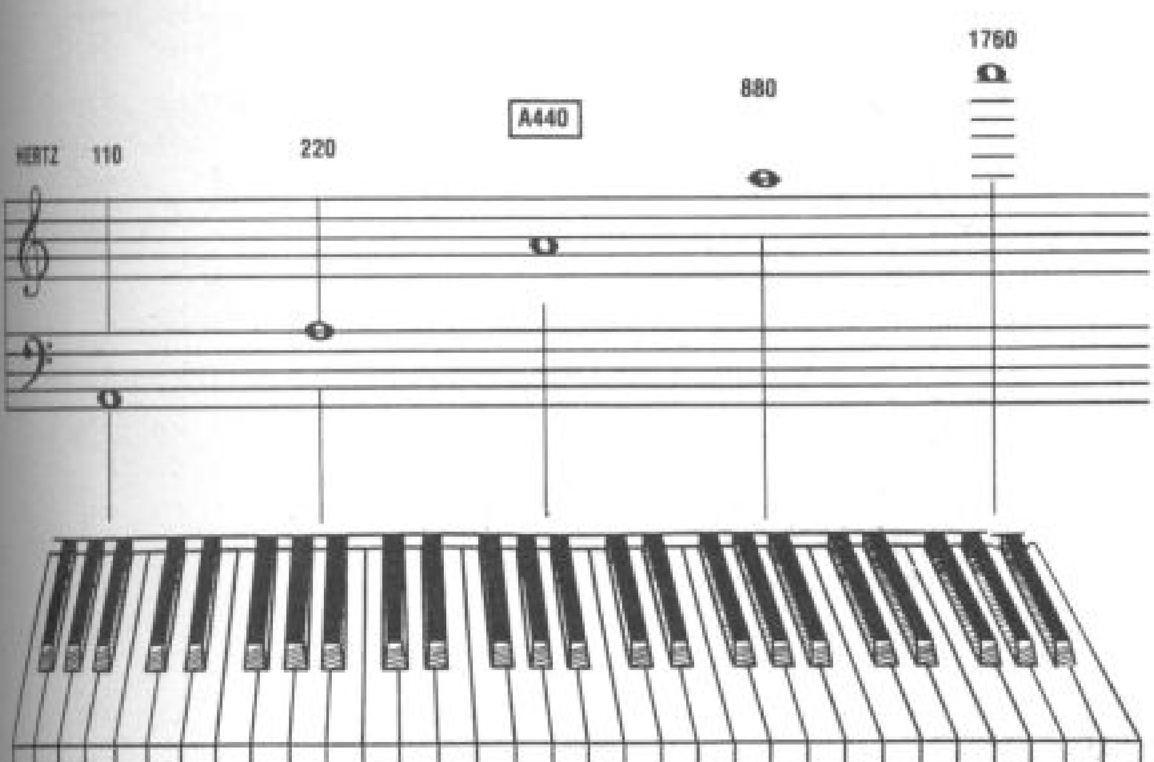
Harmonics

When a painter paints a picture of the sky, he doesn't reach only for the blue paint. Chances are good that he will also grab a tube of white, maybe some yellow, and perhaps even some red. Even though he may use more blue than anything else, it is by carefully mixing and combining all the other different colors and hues that the painter is able to achieve the precise shade of blue that he wants. True, the sky is still blue; blue is the main color in the painting. In reality, however, what we perceive as being just a blue sky is made up of many different colors which, when combined, contribute to the unique character of this particular "blue" sky.

Sound is color. In the same way that many colors can combine to make up a single color, many frequencies or pitches can combine to make up a single sound.

When you hear a C played on a guitar, you are not just hearing that C. You are hearing a whole series of mathematically related frequencies above and in addition to that C. In fact, almost every acoustic (as opposed to electric) note you hear is made up of a series of related frequencies that

FREQUENCIES OF A ACROSS THE KEYBOARD.

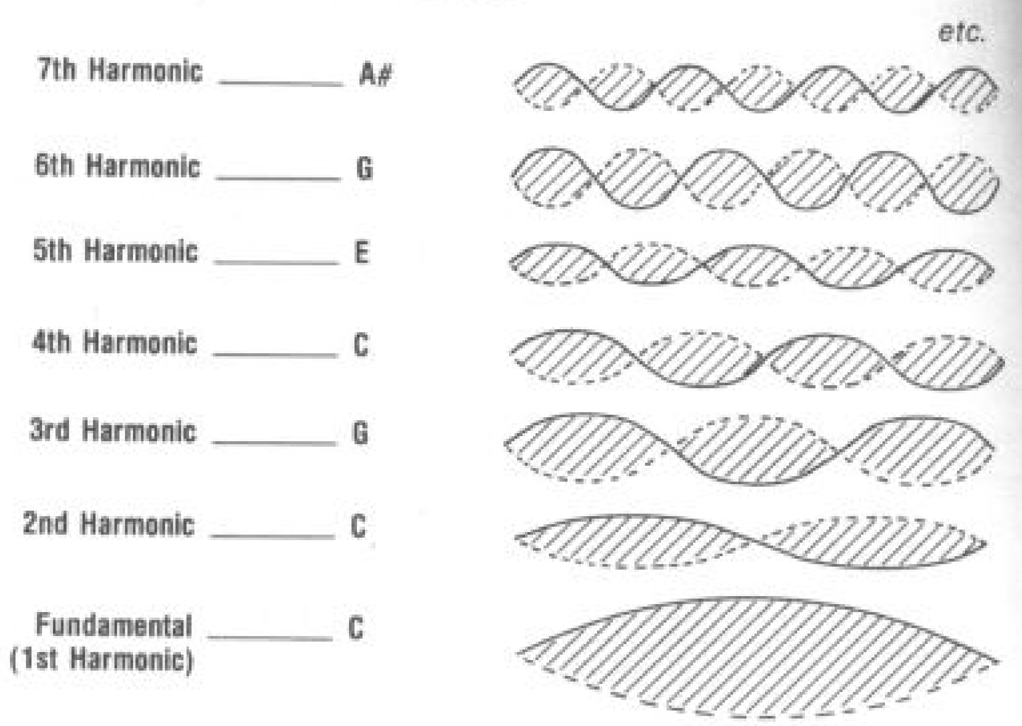


combine to create the characteristics (or timbre) of that particular note. These related frequencies are called *harmonics*, *partials*, or *overtones*. They are the result of a naturally occurring series of multiples of the basic vibration. A guitar string that is vibrating at a frequency of 264 Hz (middle C) will also contain vibrations, within the main vibration, that are two, three, four (and more) times the speed of the main frequency. These simultaneous vibrations yield frequencies of an octave above the main frequency (2 × 264 = 528 Hz), a twelfth above (3 × 264 = 792 Hz), two octaves above (4 × 264 = 1056), and so on. This predictable pattern is known as the *natural harmonic series*. It is the proportional strength of these harmonics relative to the main frequency that determines the timbre of a particular sound.

The harmonic content of a note is referred to as the note's *harmonic spectrum*. The main frequency, like the blue in the painting, is called the *fundamental* or the *first harmonic*. The fundamental is perceived as the most prominent frequency for two reasons: it is the lowest pitch in a harmonic series, and it has the greatest amplitude (it is the loudest).

In synthesis we change the color or timbre of a sound by adding or subtracting harmonics.

THE NATURAL HARMONIC SERIES



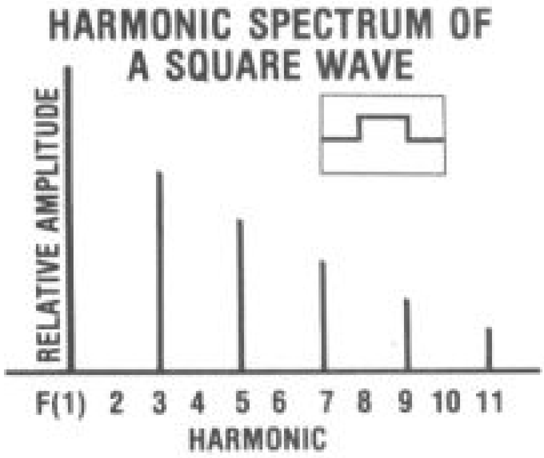
Every extra division of a vibrating string produces a higher harmonic.

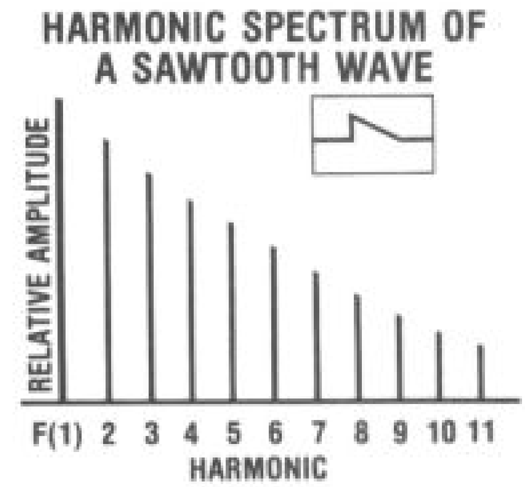
Loudness

Loudness, or volume or amplitude, affects sound in two ways. First, it defines the timbre of a sound by establishing the relative amplitude of the various harmonics in that sound. If the odd harmonics of a sound (3, 5, 7, 9, etc.) are emphasized, the sound will have a characteristic hollowness to it, such as a clarinet. If the upper harmonics of a sound are strong, it will have a nasal quality to it like an oboe.

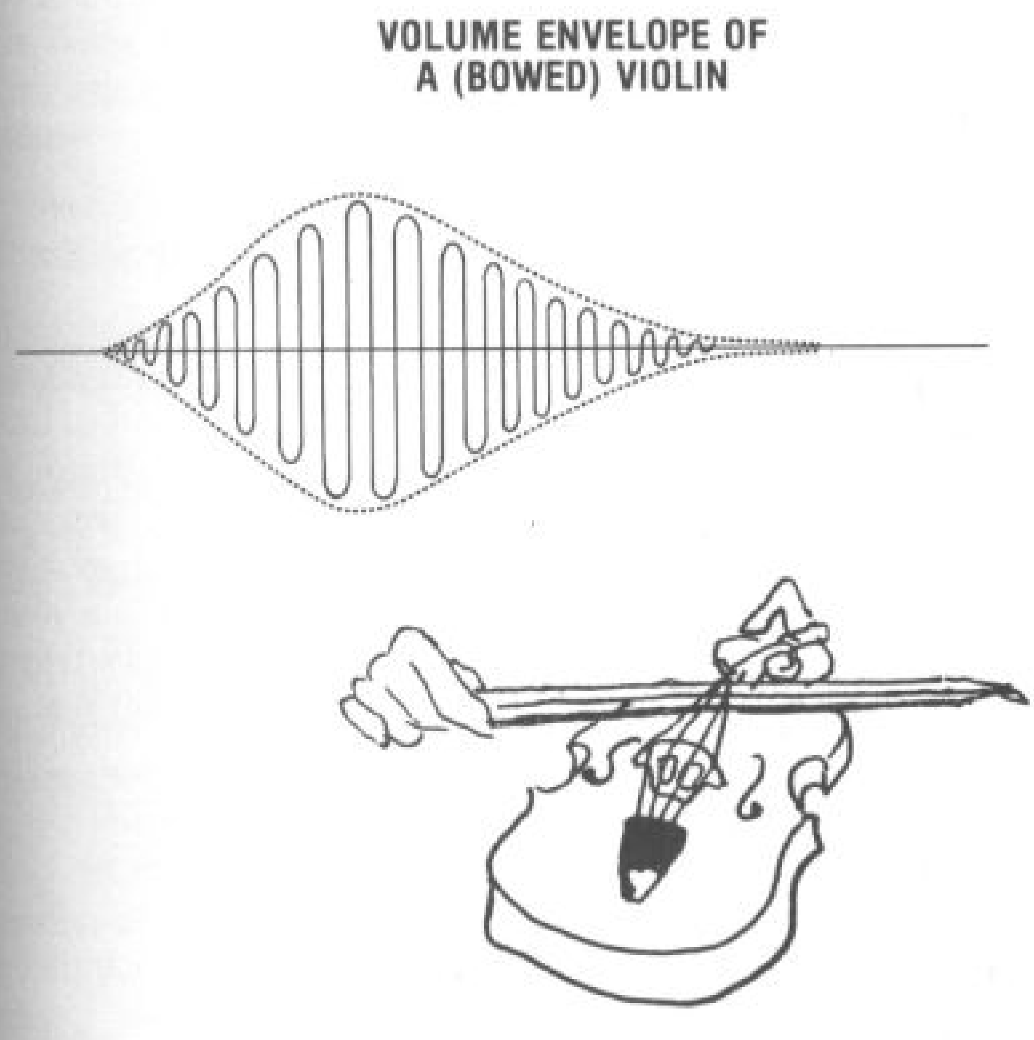
The second way in which loudness affects sound has to do with the way a sound changes "shape" over time. A note played on a violin has a very gradual *attack* (the beginning of a note), that is, it can take almost a full second for the note to reach maximum volume. On the other hand, the attack of a note played on a piano is very sudden; maximum or peak volume is reached almost instantly. The differences in the shapes of these notes have to do with how the volume levels of the notes change over time. The changing shape of the volume is called the *envelope*.

Pitch, timbre (harmonics), and loudness—these are the ingredients we use to synthesize sound.



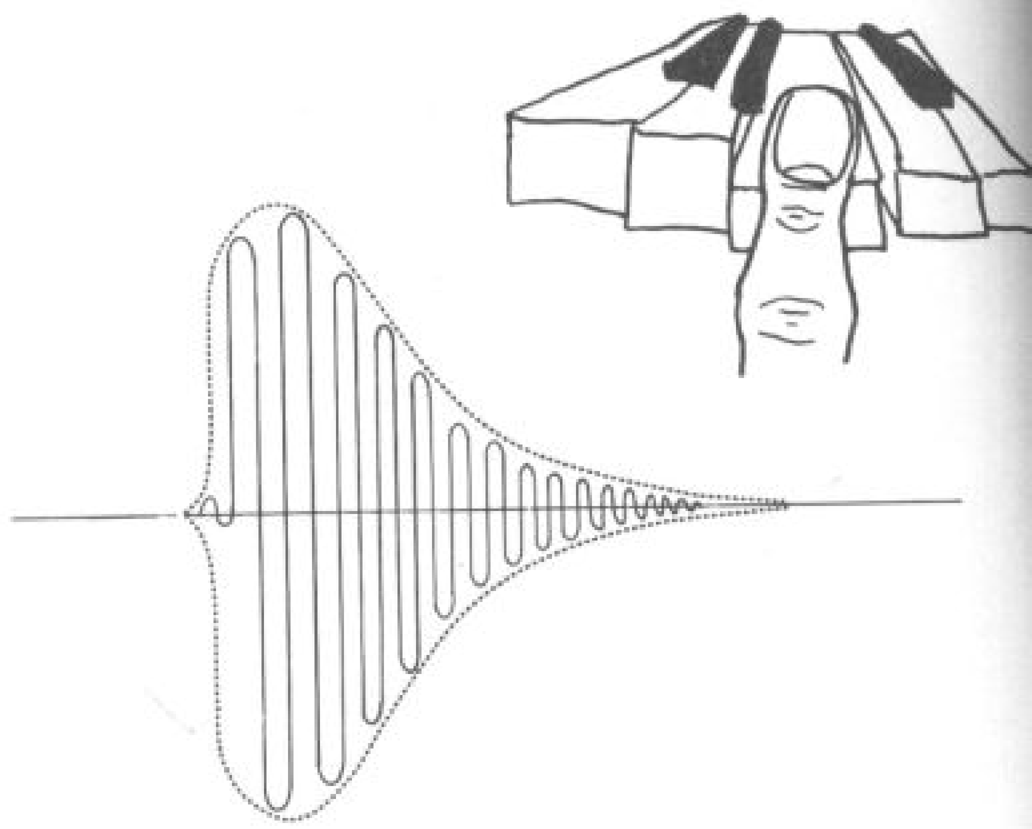


The relative amplitude of harmonics determines a sound's timbre.



A sound's overall shape is referred to as its volume envelope.

VOLUME ENVELOPE OF A PIANO



MORE ABOUT WAVES AND WAVEFORMS

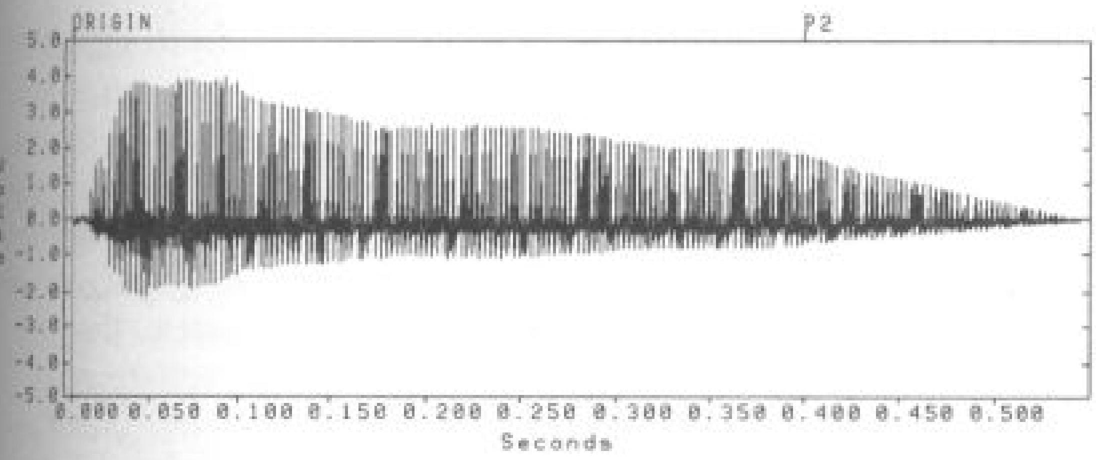
We discussed briefly how sound travels in waves. We hav also talked about the elements of sound: pitch, timbre (ha monics), and loudness. How do these relate?

If we could take a picture of these waves, we wo see that every sound creates its own characteristic shape c waveform.

The accompanying diagram is a computer-genera printout showing the waveform of a single note played or trumpet. Pretty wiggly, huh? You can see how complex e single note appears when you examine it closely.

If you play a C on a trumpet, its waveform, or waveshape, looks like the accompanying diagram. Pretty gly, huh? You can see how complex even a single note ap when you examine it closely.

All the elements of a sound are contained in its waveform. By examining a waveform, it is possible to ca its frequency, its amplitude, and the harmonics that cont to its timbre. We will be using these "sound snapshots" to identify the waveforms most commonly used in synthesis



Waveform produced by a trumpet.

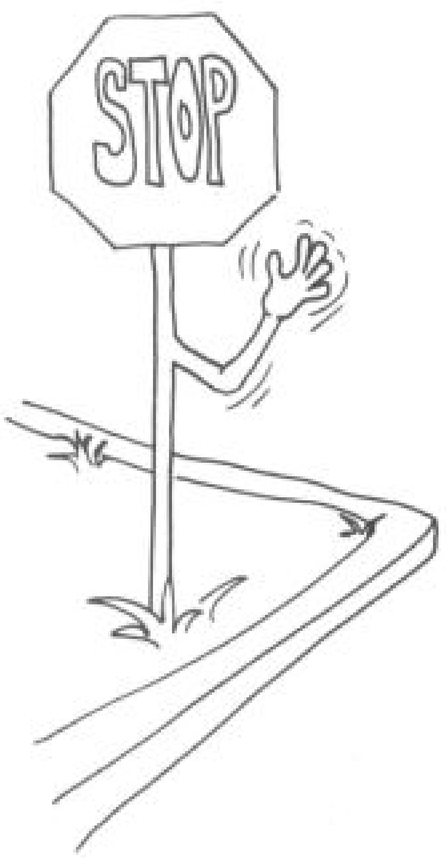
The Five Most Common Waveforms in Synthesis

In synthesis we generate waveforms electronically. Of course, we try to do it simply and cheaply and with as few components as possible. At the same time, we want those few components to provide us with the broadest possible variety of timbres. The following are the five most commonly used waveforms in synthesis. Together they fulfill the requirements mentioned: they are all relatively easy to produce electronically, and they each offer a different harmonic spectrum, or timbre. Some are brassy and bright; some are smooth and hollow. They represent the different colored paints of your synthesizer palette.

The Sine Wave

The simplest and purest waveform is the *sine wave*. But just as perfect crystals rarely occur in nature, a perfect sine wave is hard to find outside a laboratory (or synthesizer). Most natural or acoustic sounds are relatively complex and contain many harmonics. Sine waves contain no harmonics. A sine wave contains only the fundamental. The fact that a sine wave is actually a single pure tone, containing no harmonics, makes it an important building block in synthesis. The sound of a sine wave is typically smooth, pure, and almost flutelike.

THE FIVE MOST COMMON WAVEFORMS IN SYNTHESIS



Which of the five most common "you know whats" does this drawing represent?

The Square Wave

Of all the waveforms mentioned here, the easiest to generate electronically is the *square wave.* A square wave exists in only two states—high and low. In electrical terms this is as simple as turning a circuit on and off. Do it quickly enough and you generate a frequency which, when sent to a speaker, will be converted into an audible pitch. A square wave is made up of only the odd harmonics, that is, the third harmonic, the fifth harmonic, the seventh harmonic, and so on. A square wave has a characteristically hollow sound to it, very much like a clarinet.

The Sawtooth Wave

The *sawtooth wave,* also known as the ramp wave, is rich in both even and odd harmonics. This gives it a bright, almost buzzy timbre and makes it especially valuable as a sound source. (As a rule, the more sharp edges there are in a waveform, the "brighter" it will be.) The importance of the sawtooth wave will become even more obvious when we discuss subtractive synthesis, mainly because there is so much harmonic material to subtract from it.

The Pulse Wave

The *pulse wave* is a variation of the square wave. Like the square wave, it exists in a state of either high or low. The difference is that in a square wave the ratio of the high state to the low state is fixed in equal proportions—50 percent of the time it is high and 50 percent of the time it is low—whereas in a pulse wave the ratio is variable (sort of like adjustable rollerskates).

The percentage of time during which a pulse wave is either high or low is referred to as its *duty cycle.* A pulse wave that is high 60 percent of the time and low 40 percent of the time would have a duty cycle of 60 percent. (Note: A pulse wave with a duty cycle of 50 percent is, in fact, a square wave.)

What makes a pulse wave so useful is that by simply varying the duty cycle of a wave you drastically alter its harmonic content and therefore its timbre. The sound of a pulse wave can be either smooth and hollow or thin and reedy according to its duty cycle.

One thing to bear in mind is that a pulse wave with a duty cycle of 80 percent is identical to one with a duty cycle of 20 percent. The reason is that the waveform has simply been inverted; but because the proportions remain the same (20:80 or 80:20), the harmonic content of the two waves is identical.

The Triangle Wave

The *triangle wave* shares some things with both the sine wave and the square wave. Its shape is related to the sine wave in that it has some of the smoothness of that wave as well as an emphasis on the fundamental. But like the square wave, the triangle wave contains only odd harmonics, although in different proportions than the square wave. The resulting sound is hollow like the square wave but not as bright.

NO, NO! I SAID I WANTED TO HEAR SOME MORE 'HARMONICS'. NOT HARMONICAS!!

Exercise 1. Be sure your synthesizer is in neutral, or as close as you can get to it (see Introduction for instructions). Find the oscillator section of your synthesizer. This is usually labeled VCO or DCO. There should be some selectable control that enables you to choose between two or three different waveforms—usually a square wave and a sawtooth wave, but some synthesizers also include a pulse wave. Try to isolate each waveform and listen to the particular sounds they create, especially as they compare to each other. The sawtooth wave should be reedy. The square wave should sound hollow. If your instrument has a pulse wave, find the control that varies the pulse width (the duty cycle) of the wave. Listen to how the timbre of the wave changes as you vary the pulse width. Notice that at a certain point, as you are changing the pulse width, the sound will become smooth and hollow, exactly like a square wave. This is the point where the duty cycle of the pulse wave is 50 percent. At this setting the pulse wave is, in fact, the same as a square wave.

Familiarize yourself with the sounds these waves make individually as well as in different combinations. This is your basic source material. Knowing which waveform is best suited for a particular sound is the kind of thing that makes programming faster and easier. For bright sounds use a sawtooth wave. For a mellower sound use a square wave or a triangle wave. These are just general rules. Understanding the characteristics of each wave is the main thing.

THE DIFFERENT KINDS OF SYNTHESIS

There are several different kinds of synthesis, that is, different methods of generating sound electronically. The main ones are additive synthesis, subtractive synthesis, and FM.

Additive Synthesis

The very first experiments with synthesis in laboratories and universities around the world were done using *additive synthesis.*

Additive synthesis involves generating sine waves and then combing them to create pitches containing various harmonics. If you remember, harmonics are just frequencies that bear a certain mathematical relationship to a fundamental. The fundamental is defined as whatever frequency happens to be the lowest as well as the loudest. The fundamental is also called the *first harmonic.* In other words, harmonics and fundamentals are really the same—they are both frequencies; what distinguishes them is their relationship to each other. A sine wave is a frequency that contains no harmonics. If we combine sine waves in the correct proportions and amounts, what we are actually doing is designing our own harmonic spectrum. Essentially, that is what additive synthesis is all about: creating different timbres through the process of combining or "adding" sine waves.

Although today's technology has made this type of synthesis relatively inexpensive, in the early years of synthesis, generating sound in this manner was prohibitively costly. Not only that, the equipment necessary to perform these tasks could, at the time, easily fill a room. It was not until Bob Moog came along with his ingeniously simple alternative to additive synthesis—namely, subtractive synthesis—that the synthesizer gradually made its way into the hands of eager musicians everywhere.

By the way, Bob Moog (I know how to pronounce it but I'm not going to tell you!) in addition to having invented the synthesizer is actually a very nice guy, and if he got a nickel for every time he's been given credit for coming up with the voltage-controlled synthesizer he'd be rich. One of the most incredible things about him is that he's still alive! What I mean is that it's pretty amazing when one of the founding figures in any field hasn't been dead for a hundred years. In fact you can still spot him walking around at synthesizer conventions dispensing words of voltage-controlled wisdom to eager disciples. I mention this, in part, to illustrate just how young is the field of electronic music synthesis. Twenty years ago, except for obscure experiments in out-of-the-way laboratories, it didn't exist!

Subtractive Synthesis

In additive synthesis we combine sine waves to create sounds that are rich in harmonics. In *subtractive synthesis* we do just the opposite: we start by generating a waveform that is rich in harmonics and then proceed to filter out or subtract the unwanted harmonics. There is an old gag that goes as follows. Q: How do you make a sculpture of an elephant? A: Just take a large block of stone and carve away everything that doesn't look like an elephant. Basically, that's subtractive synthesis in a nutshell.

The way we put subtractive synthesis into practice leads us to our next topic: the voltage-controlled synthesizer.

I don't want to get bogged down with too many definitions right away, so take my word for the next few things. I will be explaining the basic principles of synthesis to you using a voltage-controlled model of an analog synthesizer. Even though many of today's analog synthesizers are not, technically speaking, voltage controlled, the operating principles remain the same. A voltage-controlled model is easier to understand and explains more about the entire process.

Also, later on in this book I will discuss other kinds of synthesis, specifically FM, a type of digital synthesis that has had enormous impact on today's synthesis. But again, for a clear understanding of synthesis it makes the most sense to start with the basics of analog synthesis. First—no matter what the digital camp may say—analog is here to stay! And second, almost all the principles involved are directly applicable to other kinds of synthesis. (For a complete definition of the terms *analog* and *digital* see p. 74). If these two paragraphs are confusing, skip it; the important stuff is coming up next.

How to make a sculpture of an elephant

2

THE VOLTAGE-CONTROLLED SYNTHESIZER

VOLTAGE CONTROL

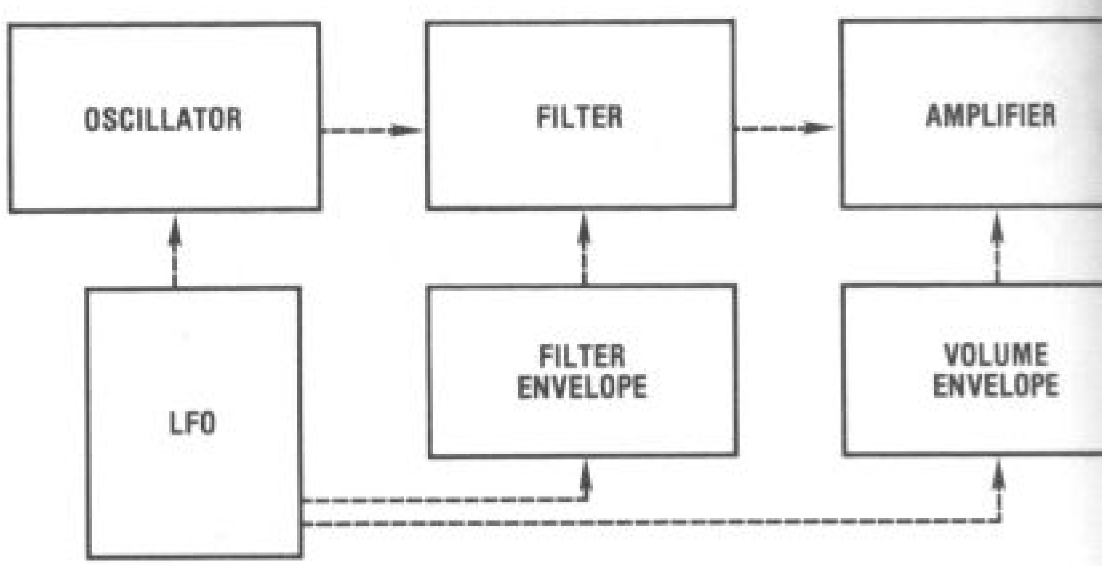
The single feature that made subtractive synthesis possible as well as practical was *voltage control.* A voltage-controlled device is one whose parameters can be controlled by a changing voltage.

If you twist the volume knob on your stereo back and forth very quickly, you can create a primitive kind of *tremolo.* (Tremolo is defined as a rapid fluctuation in volume.) The problem is that there is a limit to how fast you can spin the knob and also to how long you can keep it up without your hand getting tired. If a component is voltage controlled as opposed to hand controlled, it offers a number of important advantages. First, it frees your hands for other chores, like playing keys; and second, it is relatively easy to produce control voltages that are infinitely faster, more accurate, and more complex than anything you could do by hand.

THE SIX MAIN COMPONENTS

A typical voltage-controlled synthesizer consists of these main components: an oscillator, a filter, an amplifier, two envelope generators (the volume envelope and the filter envelope), and low-frequency oscillator. Each of these components is actually an electrical circuit that performs a specific function by either generating or modifying a sound. In addition, each of these components is voltage controlled.

SIX MAIN COMPONENTS



(Now, catch your breath because I'm about to give you a list of confusing acronyms and abbreviations. You d[o] have to memorize them right off the bat; you'll pick them u[p] soon enough.)

The abbreviations we use to identify these components are as follows:

OSC oscillator
VCO voltage-controlled oscillator
VCF voltage-controlled filter
VCA voltage-controlled amplifier
ENV envelope generator
LFO low-frequency oscillator

To confuse things a little, as more and more volta[ge-] controlled synthesizers begin incorporating digital technolo[gy] into their design, we start to come across components that the digital equivalents of the original voltage-controlled components. Whether these components are voltage controlled or digitally controlled, their functions remain essentially the same.

DCO digitally controlled oscillator (same as VCO)
DEG digital envelope generator (same as ENV)

Also, there are as many different abbreviations for envelope generators (ENVs) as there are different makes of synthesizers. The most common of these is ADSR, which stands for the four parts of an envelope: attack, decay, sustain, and release. You are also likely to run across ADRs or DADSRs. Each of these is a type of envelope generator.

ADSR attack, decay, sustain, release
ADR attack, decay, release
DADSR delay, attack, decay, sustain, release

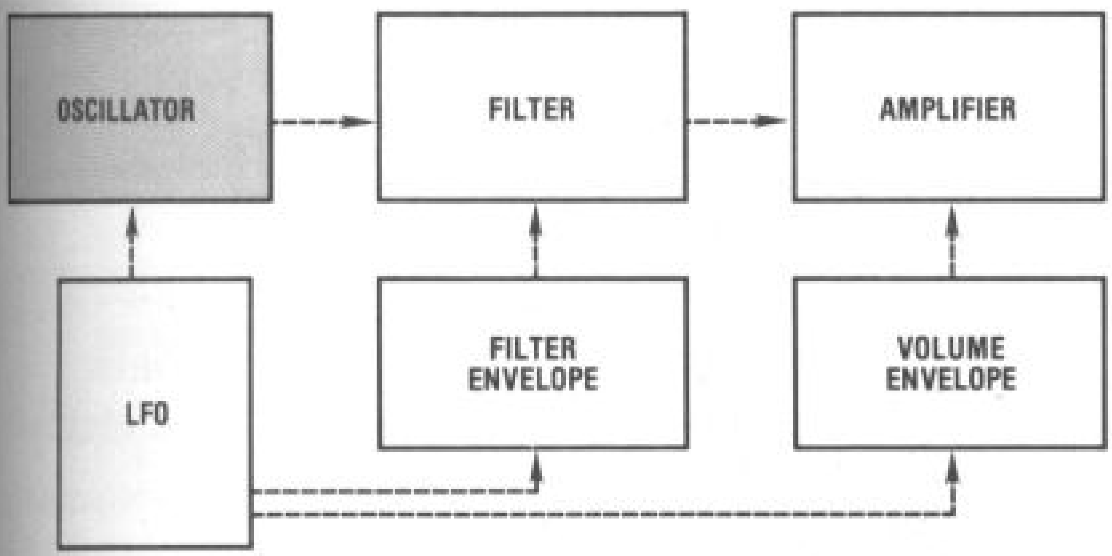
Two other terms for LFO are *sweep* and *MG* (modulation generator).

MG modulation generator (same as LFO; also sweep)

That takes care of terminology for the moment. Now I'll tell you what these components actually do.

The Oscillator

An *oscillator* is an electronic circuit that generates a constantly repeating waveform. This can be a square wave, a sawtooth wave, a sine wave, or any one or combination of different waveforms. It's like a sausage machine. It takes all that electricity lying around in the walls of your house and spits it out into a long string of discrete units (waves).

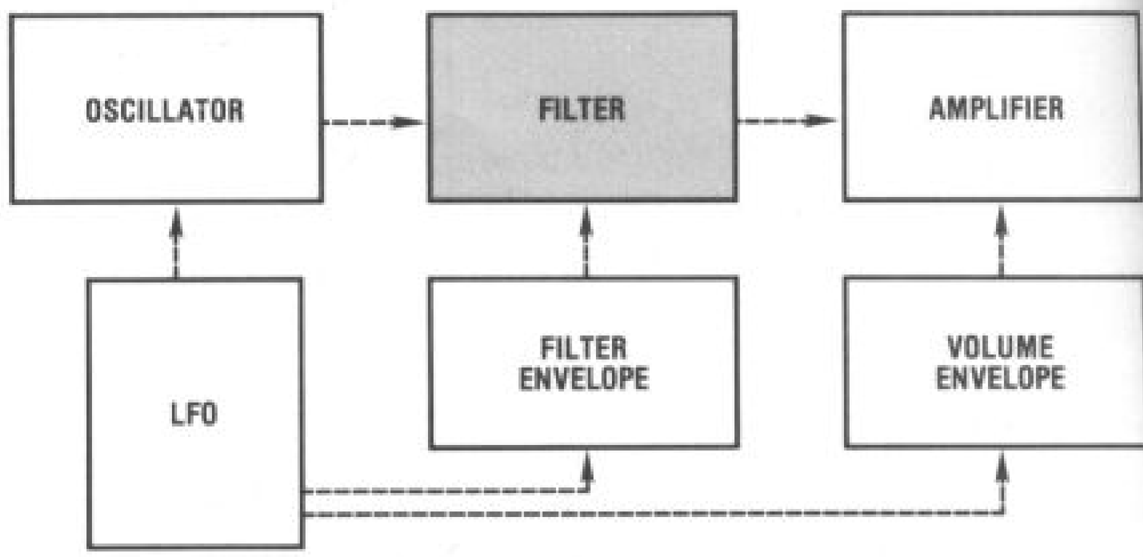


An oscillator acts as a *sound source*. The string on a guitar and the reed of a saxophone are also sound sources, that is, they initiate the vibration that we eventually hear as sound. In a synthesizer the oscillator is where the sound originates in the form of electrical vibrations (or waveforms). It is the beginning of the audio path. From here the signal travels to other components where it is filtered, modulated, and otherwise transformed into a finished sound.

> Remember, until it reaches the speaker cone, everything that happens inside the synthesizer is still in the form of electrical current or voltage. So when we talk about sound sources or audio paths or generating waveforms, we are talking about electrical signals that will eventually be transformed into audible form. As long as they are still stuck inside the synthesizer all they are are strings of dancing electrons. Even though we talk about altering the sound of a waveform, until it gets to the speaker there is no sound.

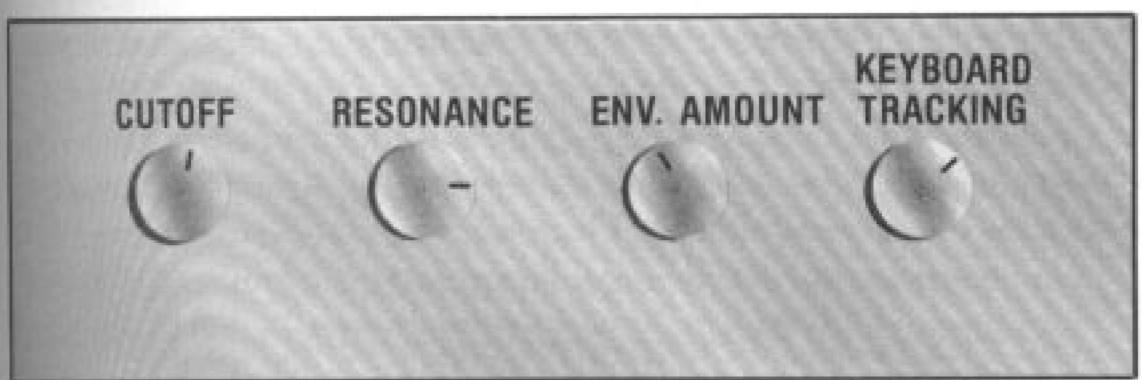
The Filter

Like the filter in a swimming pool or the filter in a cigarette, a filter in a synthesizer does exactly what its name suggests. It is an electrical circuit that filters out or blocks certain portions of the signal generated by the oscillator. It is not unlike the simple **treble/bass** controls you find on any stereo. With it you can affect the brightness and overall timbre of a sound. The filter accomplishes this by preventing all frequencies above a certain point from passing through the circuit. We refer to this point as the *cutoff frequency* or the *filter cutoff point* (Fc). When you cut out the upper frequencies (harmonics) of a sound you



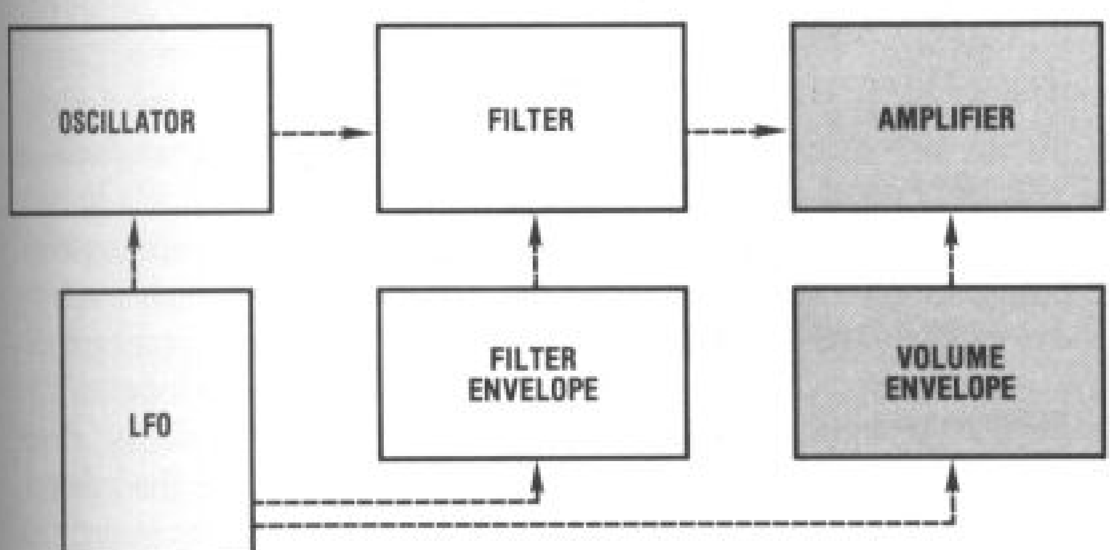
are in effect making the sound "duller." As you open the filter (or raise the filter cutoff point), you let more and more harmonics through; and the result is a brighter sound. This is the basic principle of subtractive synthesis. The oscillator acting as a sound source generates a rich, harmonic-filled waveform. This raw material is then routed to the filter, which proceeds to "filter" out the unwanted portions of the waveform allowing you to drastically modify the sound's timbre.

FILTER



The Amplifier and the Envelope Generators

There are two envelope generators in a typical synthesizer. They are exactly the same. The difference is in their application: one is used with the amplifier to affect volume, and the other is used with the filter to affect the timbre.



The Volume Envelope

One important way in which the sounds of various instruments are differentiated is by the differences in the "shapes" of the notes they produce. By shape we mean how the volume of the note changes over time. For instance, a note played with a bow on a violin starts very gradually whereas a plucked guitar string sounds immediately. A note played on a clarinet will end abruptly as soon as you stop the airflow whereas a note played by striking a triangle will sound for a long time and then fade out gradually. We refer to the shape of a note as its volume envelope.

Working together, the amplifier and an envelope generator determine a sound's volume envelope.

The amplifier is an electrical circuit that is capable of increasing or decreasing the amplitude of an audio signal. It is similar to the **volume** control on a stereo. The difference is that instead of being hand controlled the amplifier in a synthesizer is voltage controlled. The voltages that control this voltage-controlled amplifier (or VCA) are generated by one of the envelope generators (ENV). They work in tandem.

AMPLIFIER

The typical envelope generator, an ADSR, has four stages, as mentioned earlier. Attack, decay, sustain, and rele are the four parameters.

When you press down on a key, you are sending t voltage-control messages: one tells the oscillator which pitch produce; the other instructs the envelope generator to begin four-stage cycle. One stage after the other, the envelope generator runs through its cycle—first attack, then decay, t sustain. The envelope generator will then remain at the sust level until it receives its next instruction from the keyboard long as you keep the key held down, the note will continue sustain. Only when you release the key (i.e., lift your finge off it), will the envelope generator receive its instruction fr the keyboard and finally run through the last stage of its cycle—the release portion of the envelope.

Here are the four stages, one by one:

The *attack* portion of a volume envelope refers to the very beginning of a sound. It is defined as the time it takes for a sound to go from zero amplitude (or no sound) to peak amplitude (the loudest sound). A violin has a very slow attack. It reaches its peak volume gradually. A piano has a very sudden attack. It reaches its peak volume suddenly—as soon as a key is struck.

Decay refers to the next portion of the volume envelope immediately following attack. It is defined as the time it takes for a sound to go from peak amplitude to sustain amplitude. Many sounds reach a momentary peak or maximum amplitude during the very beginning of their volume envelope. As the sound continues, or sustains, the amplitude or volume levels off and remains at that fixed level until it reaches the release portion—the tail end of the envelope. A good example of peak would be the initial *blatt* of a trumpet note. The portion of that sound between the *blatt* and the eventual steady sustained note is what we call decay.

As described above, *sustain* is defined as that portion of the volume envelope that remains at a fixed level for as long as the note is held, that is, in terms of the synthesizer for as long as the key is depressed. A note on a violin played with a bow would have a very long sustain. The same violin string plucked by a finger would have hardly any sustain at all. (Don't confuse the sustain portion of an envelope with the **sustain** pedal on a piano. The **sustain** pedal on a piano has more to do with the release portion of an envelope, as described below.)

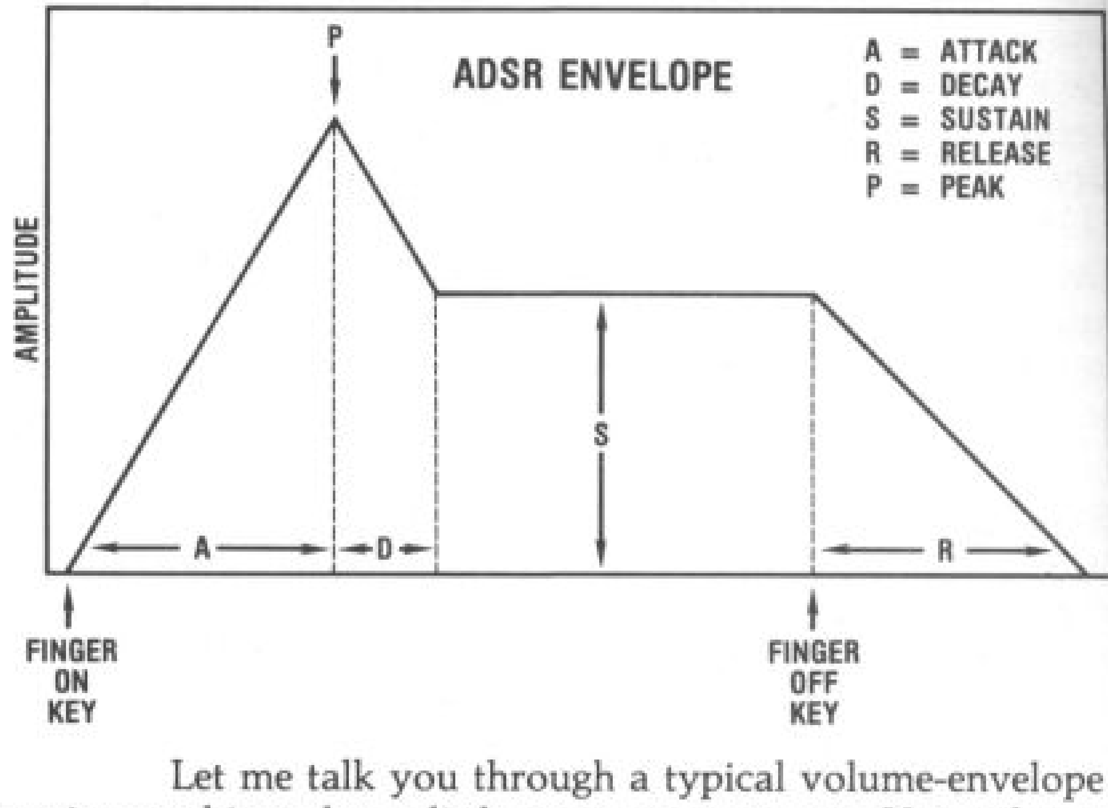
Release refers to the final portion of a sound's volume envelope. It is defined as the time it takes for a sound to go from the sustain portion of its envelope to zero amplitude. Once a key is released on the synthesizer, a sound enters the release portion of its envelope—that is, you hear the very tail end of the note. A woodwind instrument has a very short release time. As soon as you stop blowing into it, the sound stops. A xylophone, on the other hand, has a very long release time—that is, it takes a relatively long time for the note to die out.

Those are the four portions of your typical ADSR or envelope generator.

One important thing to remember is that the first, second, and fourth of these four envelope parameters (attack, decay, and release) determine *rate*—that is, the change in amplitude over time. The third, sustain, is simply a setting of level. The other important thing to be aware of is that the peak level (the point between attack and decay at which the envelope reaches maximum amplitude) is a fixed setting and cannot be controlled by the envelope generator. A good way to think of peak is as an invisible and permanent setting located between the **attack** and **decay** controls. There is an important relationship that exists between the peak of an envelope and the **decay** and **sustain** controls: If **sustain** is set at maximum, then regardless of where the **decay** is set it will have no effect on the sound. This is because with **sustain** set at the same level as peak there is no place for the sound to decay to. Only if the sustain level is less than the peak level will it be possible to hear the decay portion of the envelope.

These four parameters (five including peak) send out instructions in the form of control voltages to the voltage-controlled amplifier (VCA), and in so doing they determine the shape or volume envelope of a given sound. It's kind of like designing your own cookie cutter . . . every time you play a note it comes out in that particular shape.

ADSR VOLUME ENVELOPE

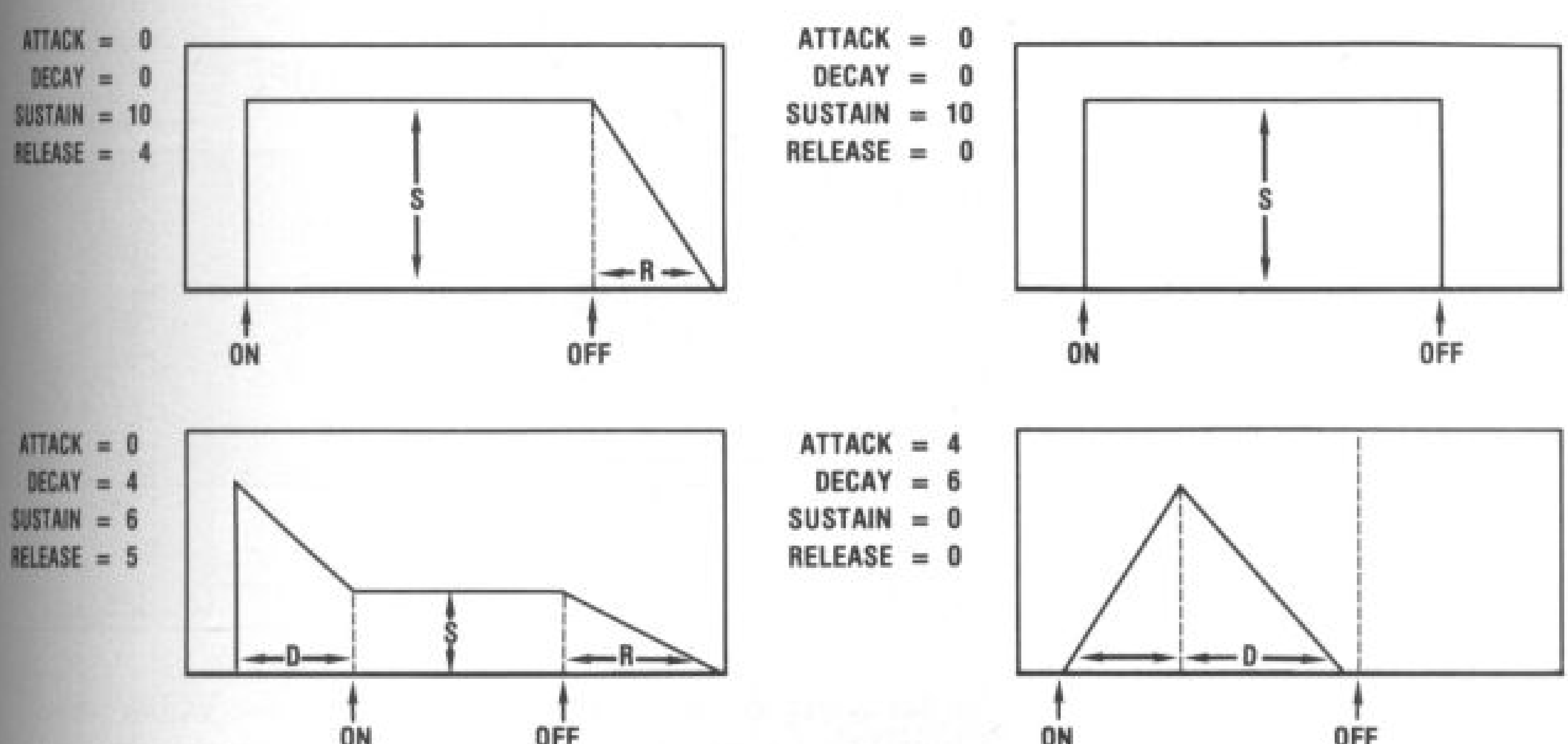


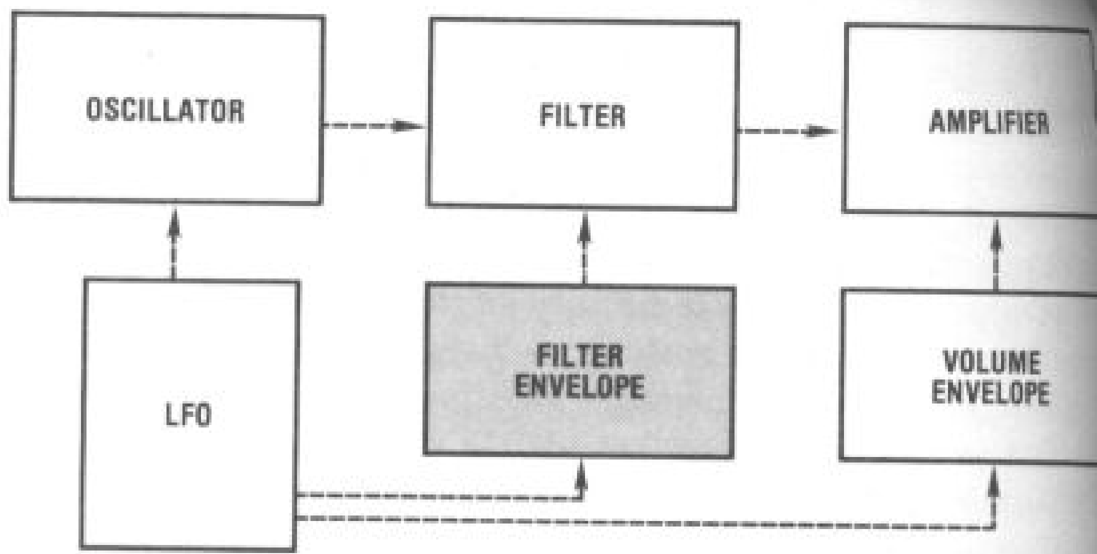
Let me talk you through a typical volume-envelope setting so this makes a little more sense to you. Using the accompanying diagram as our example, what we've done is set a slightly gradual attack, a slightly faster decay, a sustain at ¾ strength, and a relatively long release. When we press down on a key, this is what happens: First, the attack setting instructs the amplifier to gradually turn up the note's volume until it reaches its peak or maximum volume. This takes just under a second. Next the decay setting instructs the amplifier to turn down its volume until it reaches the level set by the sustain. It takes about ½ second for the volume to go from peak to sustain. Now, once the note reaches the sustain portion of its envelope, it stays there for as long as you hold down the key. It is only at the moment at which you take your finger off the key that the note enters the release portion of its envelope. At that point the time it takes for the note's volume to go from the sustain level back down to zero is determined by the release setting. In our example we set a relatively long release, so this will take around 3 or 4 seconds.

The Filter Envelope

Just as a volume envelope determines the shape of a sound and how the amplitude of that sound changes over time, a filter envelope determines how the effect of a filter changes over time. A filter envelope has the same parameters as a volume envelope—attack, decay, sustain, and release. When you set a gradual attack on a volume envelope, the beginning of the sound will fade in gradually. When you set a gradual attack on

OTHER POSSIBLE ADSR ENVELOPES





a filter envelope, the harmonics determined by the cutoff point will fade in gradually. With a filter envelope, you can subtly control the arrival and disappearance of harmonics over time. You can create a sound that starts "dark" but grows increasingly brighter, or vice versa. You are in effect superimposing two sound shapes, one over the other—the shape of a sound's timbre over the shape of the sound's volume. This is one important way we have of adding internal motion to a sound.

Remember, both of these envelope generators—the one for the volume envelope and the one for the filter envelope—are identical. The difference between them is that each is applied to a different component—one to the amplifier, the other to the filter—the result being that one shapes volume and the other shapes timbre.

FILTER ENVELOPE

The LFO (Low-Frequency Oscillator)

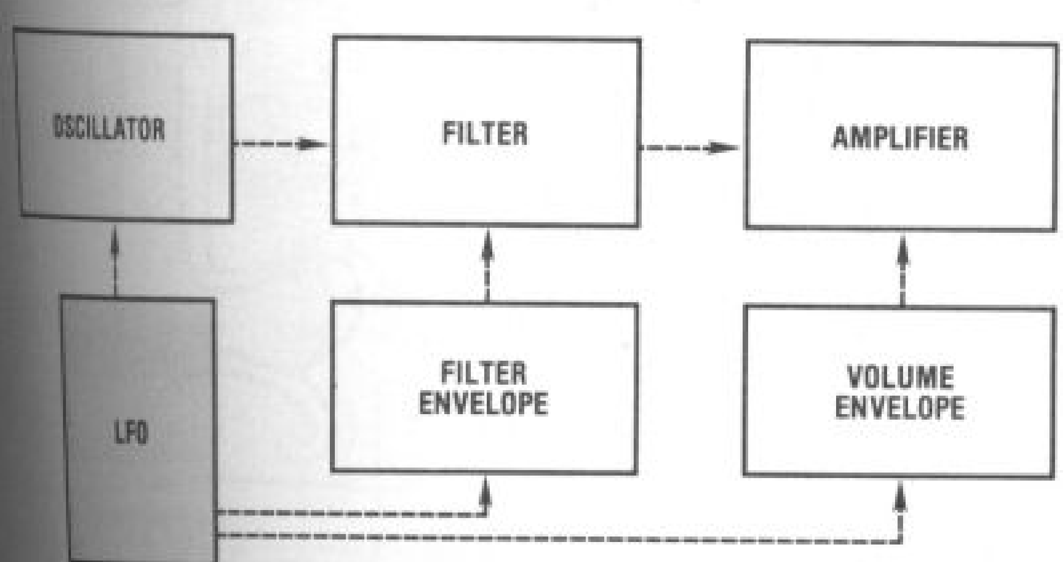
So far we've generated a waveform (with the VCO); we've filtered that waveform (with the VCF); and we've designed a shape for the volume of the waveform (with the volume ENV and the VCA) as well as one for the timbre of the waveform (with the filter ENV). Now we're going to take that waveform and make it wriggle across the floor on its belly like a reptile. That's right! We're going to add a little vibrato. Maybe even some tremolo.

Here's the most important thing to remember at this point. All the components that we've mentioned so far, the VCO, the VCF, the VCA, and even the two ENVs, are voltage controlled. That means that they will obey instructions sent to them in the form of a control voltage. And that is what we are going to do using the LFO. Ready?

An LFO is technically the same as an ordinary oscillator except for two important features: it oscillates at a lower frequency than does the main oscillator, and it is used not as a source but as a modifier (a modifier changes or alters a signal). Its function is to modulate (regulate or vary the frequency of) the other synthesizer components.

Now, there's a very important concept we're dealing with here—modulation—and it is easy to get confused. So bear with me, because I'm going to take it slow.

Remember that when we discussed the different waveforms and their properties I described the sonic qualities of each one—the sine wave has a pure sound and the square wave has a hollow sound, etc. Well, each of these waves, in addition to being able to produce an audible signal, can be used to modulate another voltage-controlled device such as an amplifier, a filter, or even another oscillator. In other words, the same signal or device can be used as a source or as a modifier depending on your needs. This is important to re-



member because as we go on we will be running into more and more examples of components doing this same kind of double duty.

As I said earlier, the other thing that distinguishes an LFO from your main source oscillator is the fact that it generates a much lower frequency. Let me explain. The human ear (as opposed to the rhinoceros ear or the porcupine ear) is capable of distinguishing audio frequencies somewhere in the range of 20 Hz to 20,000 Hz. (This range drops dramatically as

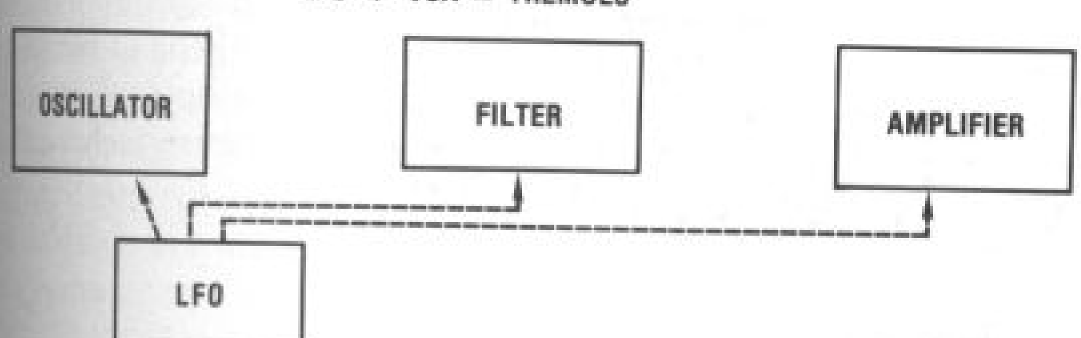
you get older and especially if you expose yourself to inordinately loud music. This is no joke. Don't forget Ted Nugent is now deaf in one ear. So watch it!) This is known as our *audible range*. Remember, Hz means cycles (or vibrations) per second. If an oscillator is generating less than 20 cycles per second, we will not hear it as a pitched tone but rather as a series of rhythmic pulses. However, if we use these slow waveforms to modulate another waveform that is within the audible range, its presence becomes immediately evident. This is how we get vibrato and tremolo. We use a very slow wave to modulate some aspect of a normal audio wave, and we wind up with a normal wave with a wiggle in it. Allow me to illustrate.

Imagine that you are sitting on a motionless train waiting to pull out of the station. You're getting very bored, so to help pass the time you take out a piece of paper and start drawing wiggly lines on it with a pencil. Suddenly the train pulls out of the station, but you are so into your drawing that you can't stop. When you finally do stop, you look down at your drawing and discover that what started out as a wiggly line turned into a squiggly line as soon as the train began moving. The basic contour of your line stayed the same but the vibration of the train influenced the overall shape of the wiggle by imposing its own movement onto it. The motion of your drawing hand was modulated by the jiggling motion of the train resulting in a modified wiggle or what we might now refer to as a squiggle. Or to put it another way: wiggle + jiggle = squiggle.

That is how an LFO works. The main oscillator generates its own version of a wiggly line—a waveform. The LFO generates a waveform similar to the main oscillator (usually a sine wave) but of a much lower frequency (i.e., there are only a few cycles per second). We don't hear the LFO's waveform because it is not sent directly into the audio path (also because its frequency is too low to be distinguished

THREE DIFFERENT LFO ROUTINGS

LFO + OSC = VIBRATO
LFO + VCF = FILTER MODULATION
LFO + VCA = TREMOLO



as a pitch); instead, it is used as a control voltage to control either the VCA, the VCO, or the VCF. By doing so, it is modulating those components by imposing its own waveshape characteristics on each of them.

Those are as follows: If an LFO generates a sine wave that is used to modulate the VCA you get *tremolo*. Tremolo is defined as a periodic fluctuation in amplitude.

If an LFO generates a sine wave that is used to modulate the VCO you get *vibrato*. Vibrato is defined as a periodic fluctuation in pitch.

If an LFO generates a sine wave that is used to modulate the VCF you get a *modulated filter*, in other words, a periodic opening and closing of the filter.

Another way of routing the LFO is to use it to modulate the pulse width. When we discussed the pulse wave, you learned that by altering the width of the pulse (the duty cycle) you change its timbre. Using a signal generated by the LFO, we can impose a periodic fluctuation on the pulse width thus adding continuous motion to the waveform's timbre.

In addition to the sine wave, most LFOs offer a choice of at least one or two other waveforms, usually a square wave, a sawtooth wave, or a triangle wave. Some systems offer a random waveform or what is also referred to as **sample and hold.** We will cover this in a later chapter.

The main thing to remember is that the LFO is the main source of modulation.

Recap

So, the oscillator (VCO or DCO) generates a waveform; the filter (VCF) changes the timbre of the waveform; and the amplifier (VCA) along with the envelope generators (ENVs or ADSRs) determines the shape of the waveform's volume and timbre. Finally, using the low frequency oscillator (LFO), we are able to impart a little vibrato or tremolo onto our friendly waveform.

Those are the basics.

You generate a waveform and then you modify it. There are many different ways of modifying the basic waveforms, and we'll try to cover as many of them as we can. Of course, the hard part of synthesis is coming up with interesting and subtle combinations of those different techniques.

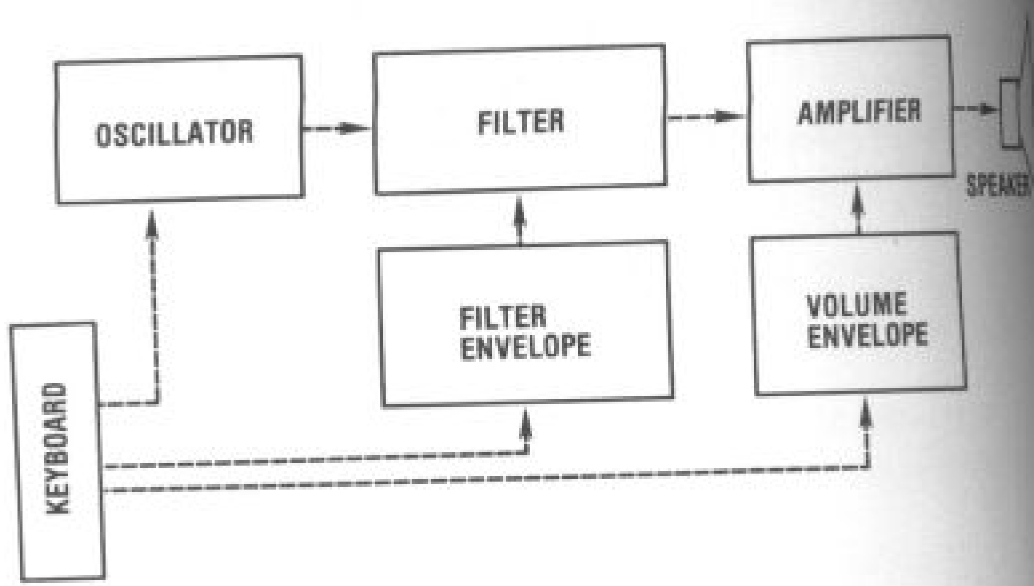
3

SYNTHESIZER COMPONENTS

A Synthesizer Flow Chart

Right about now would be a good time to familiarize you with signal-flow or block diagrams, a few of which you've already seen. These diagrams show you the signal flow of your synthesizer's voice module. In other words, they illustrate the signal path from the keyboard, to the oscillator, to the filter, to the envelope generator, to the amplifier, and finally to the speaker. It's not as complicated as it looks. Each component is represented by a block. The signal flows from left to right. It enters from the bottom or left side of a block and exits from the top or right side of a block. Lines that enter from the bottom are control signals (control voltages). Lines that enter from the left side are audio signals, that is, the waveform that originates at the oscillator.

BLOCK DIAGRAM

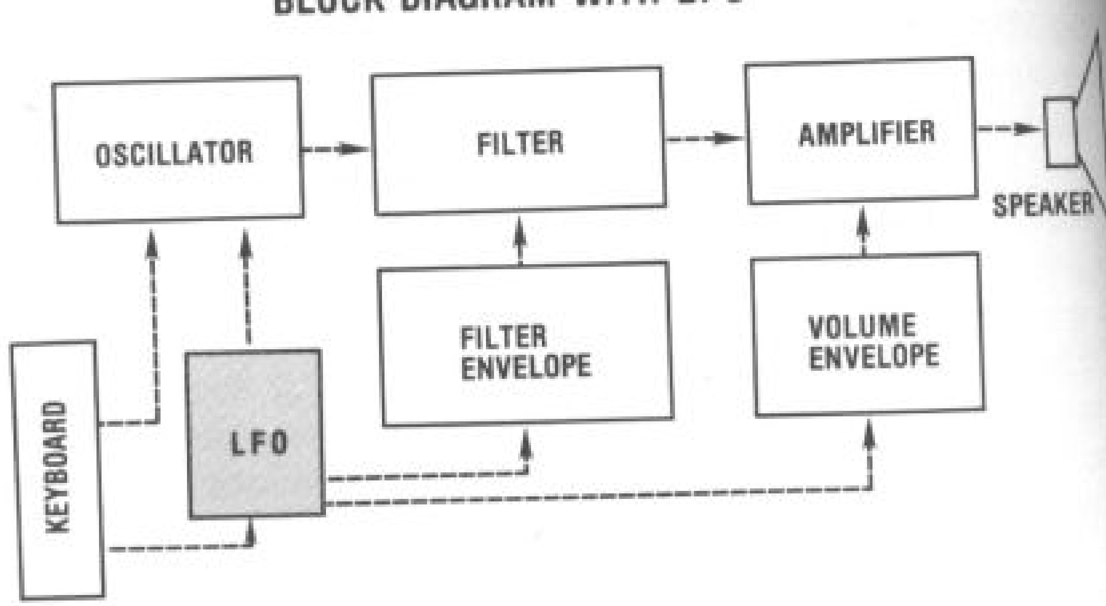


So, for instance, in the accompanying diagram the keyboard sends a control signal to the oscillator telling it to produce a waveform. At the same time, the keyboard sends control signals to both envelope generators instructing them in turn to send envelope instructions to the filter and the amplifier. The audio signal generated by the oscillator travels through the filter where it receives its filter envelope and then on to the amplifier where it receives its volume envelope. Finally it travels from the amplifier to the speaker to your ears. Got that?

Good. Now, let's make it a little more complicated by adding one other component, the LFO. See how the LFO sends control signals to either the VCO, the VCF, or the VCA?

Signal-flow or block diagrams are not that hard to figure out. They only look complicated because of all those intersecting lines. Take your time and follow the paths leading from each component to the others. This will help you to understand exactly what is going on there inside of your synthesizer. It is also the best way to get a sense of the multiple modulation routings that are possible on any given system.

BLOCK DIAGRAM WITH LFO



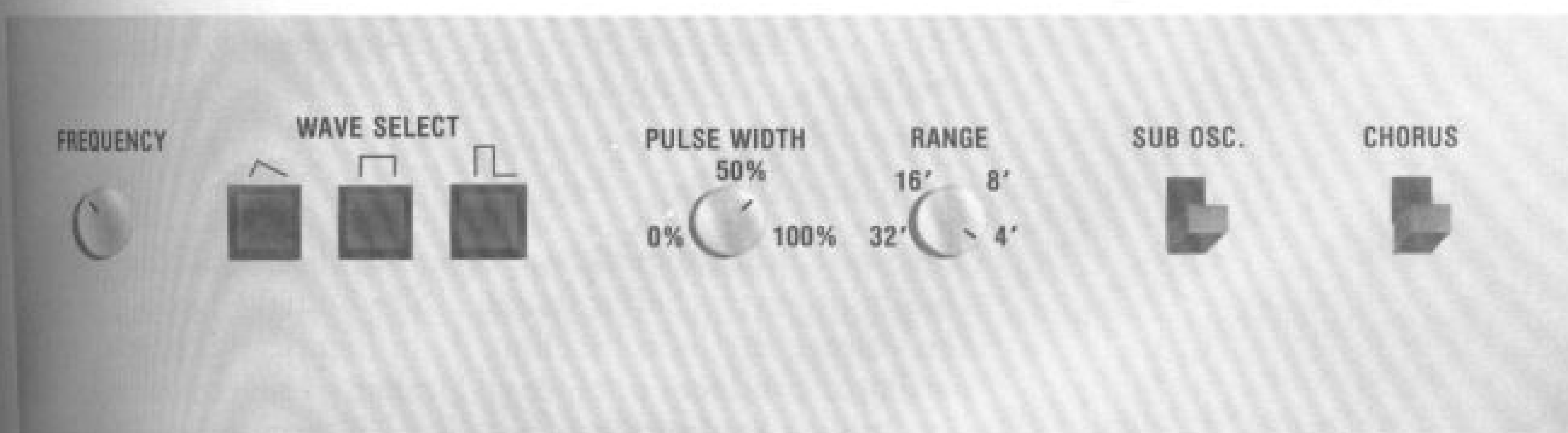
O.K., now we start getting down to the nitty gritty. If you've just finished this section, relax and let it sink in for a while. Go have a cigarette or chew on a carrot or something. Make sure that the basic functions of the components I've just described—the oscillator, the filter, the amplifier, the envelope generators, and the LFO—are clear in your in mind.

What I want to do now is start all over again and describe in even more detail how each of these basic components operates. Once I've done that I'll be able to begin introducing all the other components of a synthesizer and describing how they interact with each other. So, all ready? One more time. (Note: Any controls that are pictured in the accompanying diagrams but not explained in this chapter will be covered in Chapter 5, "Neat Extra Features."

More about Oscillators

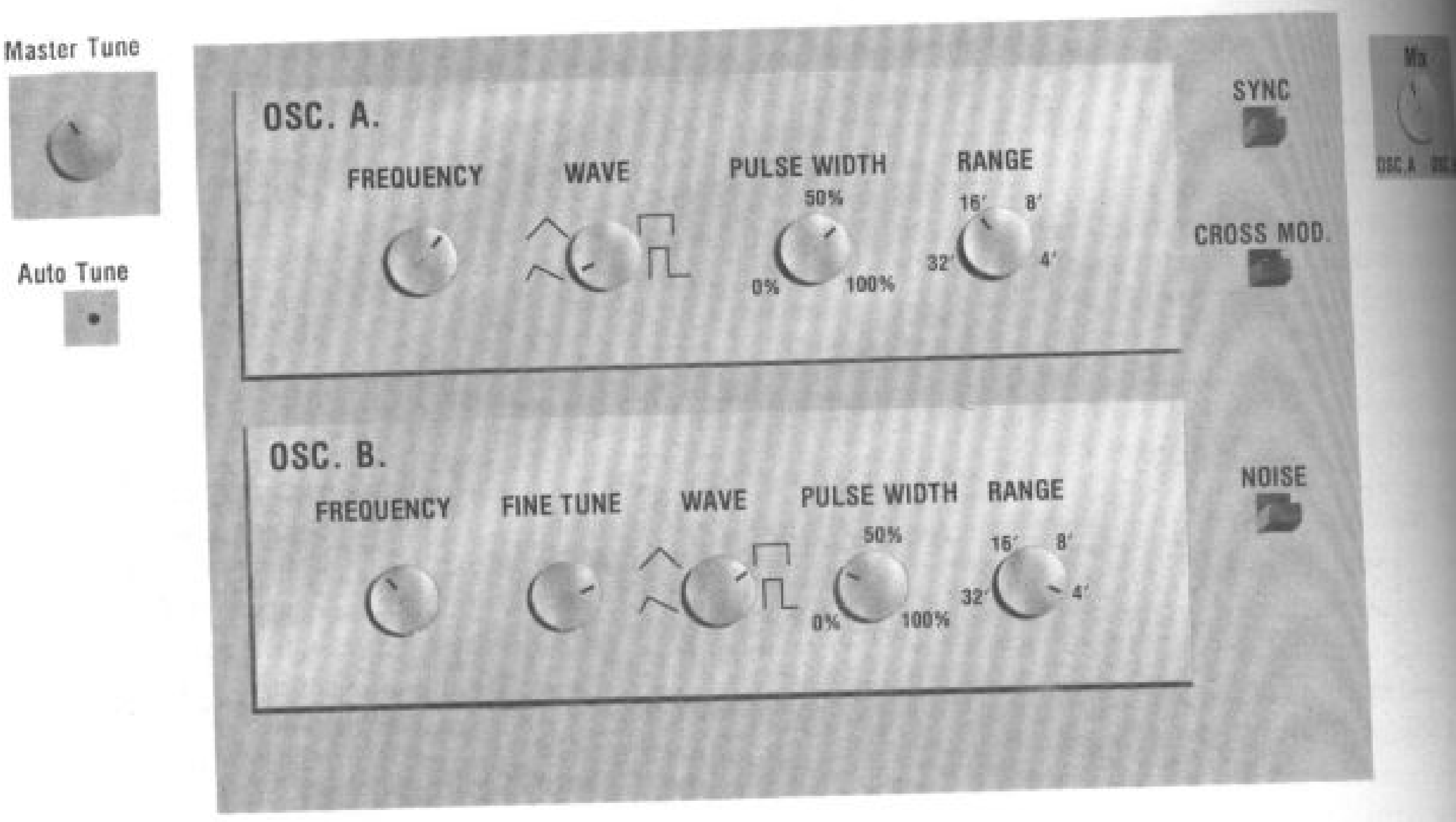
Let's look at our typical oscillator. (See accompanying illustrations.) What follows are some common selectable and variable features you'll find on most oscillator banks along with some more information about oscillators.

OSCILLATOR
SINGLE



Waveform Select

Most oscillator banks come with a switch that lets you choose between any one or a combination of several different waveforms—usually a sawtooth wave, a square or pulse wave, and a triangle wave. We've already learned how the type of waveform we start with has a lot to do with the timbre of a sound.



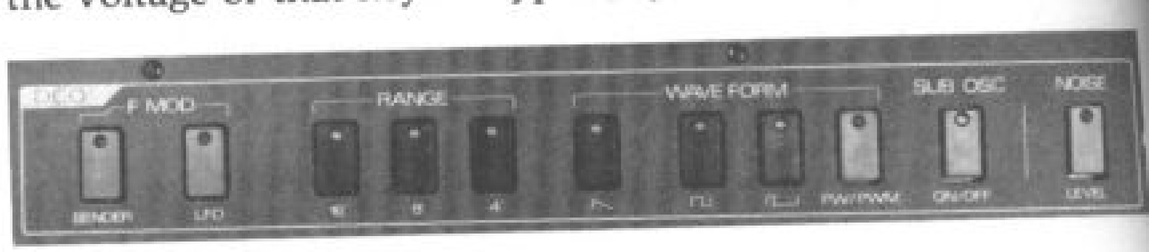
Dual Oscillator

Frequency, Tuning

Next we come to the oscillator's **frequency** or **tuning** control. This control allows you to change the frequency (the pitch) of the oscillator by as much as an octave or more. As we continue, you'll see how the ability to control the frequency of the oscillator gives you added flexibility when creating sounds.

Keyboard Control of the Oscillator

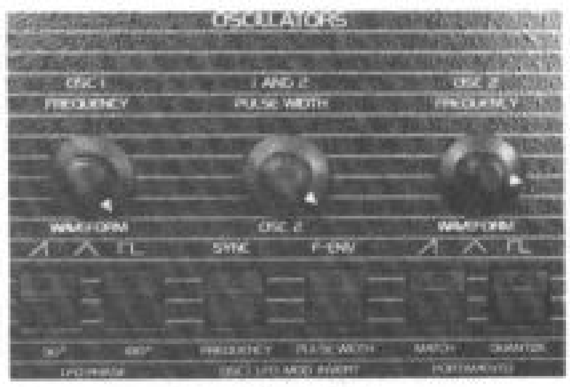
The great thing about a voltage-controlled oscillator (VCO) is that in order to change its pitch all you have to do is change the voltage that is controlling it. In fact, that is how a single oscillator produces pitch over a range of several octaves. When you press a key on a synthesizer, you send two voltage messages—one to the oscillator and one to the envelope generator. The message sent to the envelope generator signals that the note's envelope should begin. The message that is sent to the oscillator identifies what key is being played according to the voltage of that key. A typical synthesizer keyboard oper-



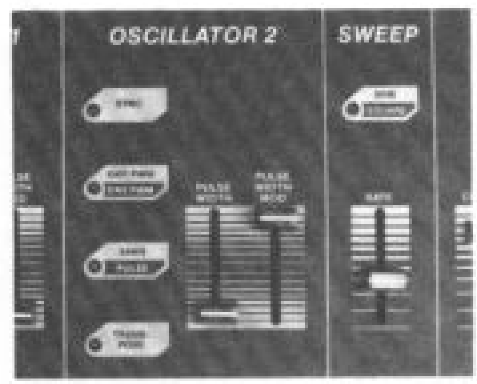
DCO (Oscillator). *Kawai SX-210*

ates on the basis of 1 volt per octave. If low C is represented by 0 volts, then the C an octave up would be represented by 1 volt. The C an octave above that would be 2 volts, etc. An increase in voltage of $\frac{1}{12}$ volt would represent a half step (there being 12 equal divisions in an octave). Using the principle of voltage control, the keyboard is controlling the oscillator and in so doing is determining its pitch. There are many other ways of controlling the pitch of an oscillator; we'll get to them all in due course.

DCO (Oscillator). *Roland Juno 106*



Oscillator. *Oberheim OB-8*



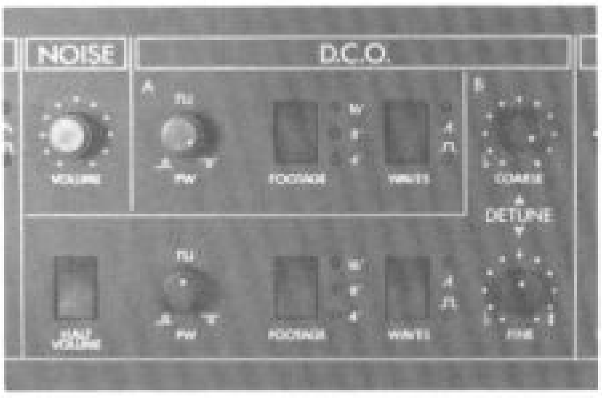
Oscillators 1 and 2. *Chroma Polaris*

Number of Oscillators per Voice

Most synthesizers today are *polyphonic*—that is, you can play more than one note at the same time on them. This means that every independent voice will require at least 1 separate oscillator so that, for example, an 8-voice polyphonic would have 8 oscillators. In fact, most synthesizers actually have 2 oscillators per voice (which would mean an 8-voice polyphonic would have 16 oscillators). By giving you 2 oscillators per voice, the synthesizer allows you to vary your source material in a number of useful ways even before you start modifying it with filters and so forth.

Tuning Oscillators

Another very important feature of having two oscillators per voice is that you can tune each of them separately. There are several applications of this technique. One is that you can tune the second oscillator to some harmonic interval of the first oscillator. Octaves would be the most obvious example of this. The right blend of an octave or even a fifth can add a lot of character and power to a sound. Even nonharmonic intervals can be useful for certain timbres such as bell-like effects.



DCO (Oscillator). *Siel 600*

Detuning or Fine Tuning

Still another extremely useful tuning application is referred to as detuning. Detuning means tuning one oscillator just slightly sharp or flat (less than a semitone) of the other oscillator. The result is a kind of chorus effect. It fattens up the sound considerably and can make a boring sound quite interesting. (Some single oscillator synthesizers have a separate chorus button just for this purpose.)

Range/Octave

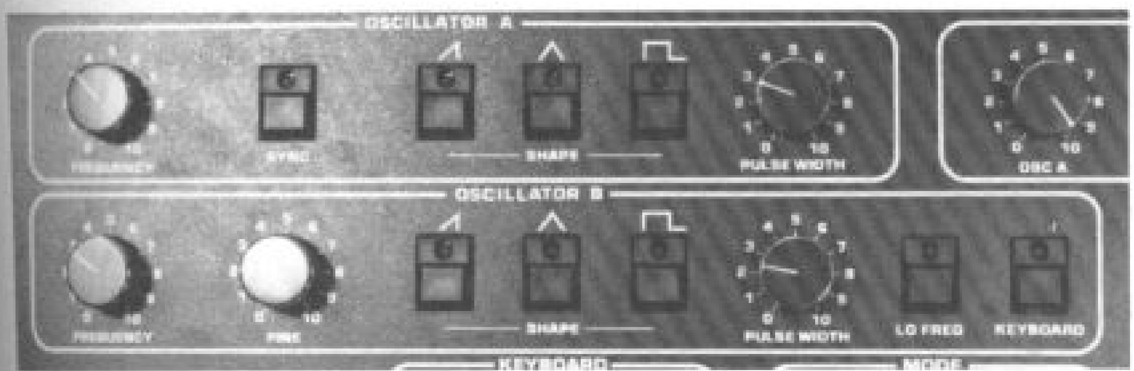
Many oscillators include a **range** or an **octave** switch. This moves the frequency range (pitch) up or down by octaves. Its function is similar to the octave drawbars on a Hammond organ, and in fact the **range** switches on today's synthesizers still use organ shorthand to label the octave steps. The markings 32′, 16′, 8′, 4′, and 2′ represent the different lengths (in feet) of the actual pipe on the early pipe organs. Even though we have traded in our pipes for oscillators, the meaning of the numbers remains the same. Every time you double a frequency (or cut a length of pipe in half) you jump up an octave.

Suboscillator

Although most synthesizers today have two oscillators per voice, there are still some that offer only a single oscillator. These single-oscillator synths often come equipped with what is called a *suboscillator*. A suboscillator is a circuit that essentially doubles the oscillator's pitch an octave below. It acts almost like a separate oscillator, but it derives its pitch directly from the main oscillator, so you have no independent control over it.

Pulse Width and Pulse Width Modulation

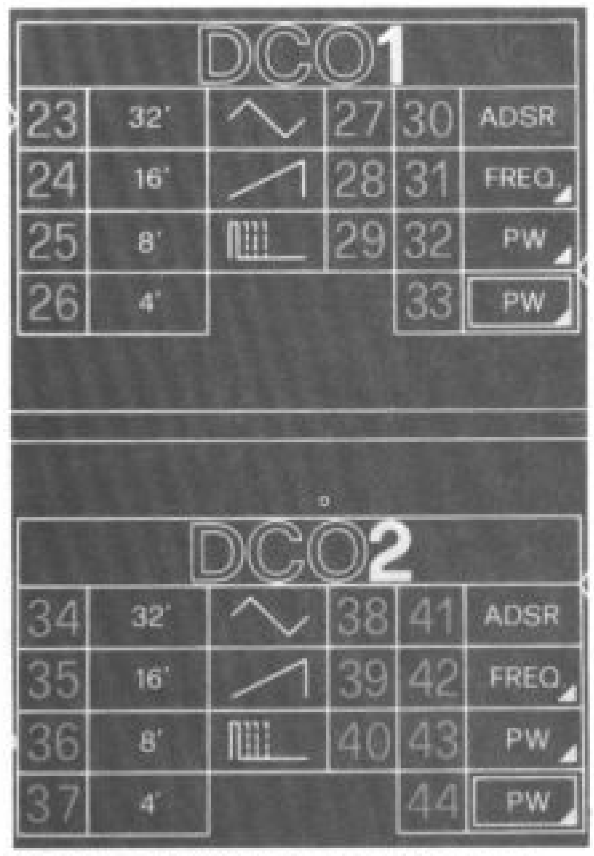
If a synthesizer offers a choice of a pulse wave, then it will also include some kind of control for changing the pulse width (PW). If you weren't watching television at the same time you were reading the section on waveforms, you will remember that the timbre of a pulse wave will change according to its duty cycle or pulse width. Very often, immediately adjacent to the **pulse width** control you will find a control labeled **pulse width modulation** (PWM). This control determines the amount of modulation from the LFO. By modulating the pulse width with the LFO, what you achieve is a periodic change in timbre. This change in timbre is similar in some ways to what happens when you modulate the filter with the LFO, but the effect is not quite as dramatic. It is relatively subtle but useful nonetheless.



Oscillator. *Sequential Prophet T8*

Mix or Balance

Any synthesizer that offers a pair of oscillators will also include a knob or *pot* for the purpose of mixing or balancing the relative amplitude of the two oscillators. This allows you to set the outputs of the two oscillators at equal strengths or, if you choose, to have one oscillator louder than the other.



DCO1 and DCO2 edit maps (Oscillator). *Bit One*

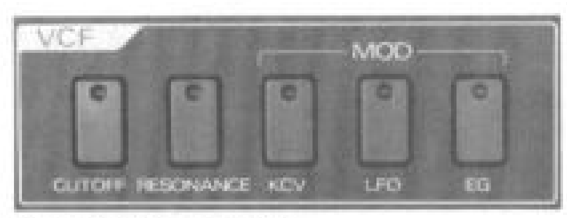
Source Material

With the exception of pulse width modulation, all the typical oscillator features just described—and those include waveform select, tuning and detuning (chorus), range, suboscillator, pulse width, and mix—represent different ways of varying your basic source material. That is, even before we start to apply filters or modulations to the basic waveforms, these features provide us with the ability to begin designing specific sounds just by mixing and blending the oscillator's output in various configurations.

Now let's apply the filter again . . .

More about Filters

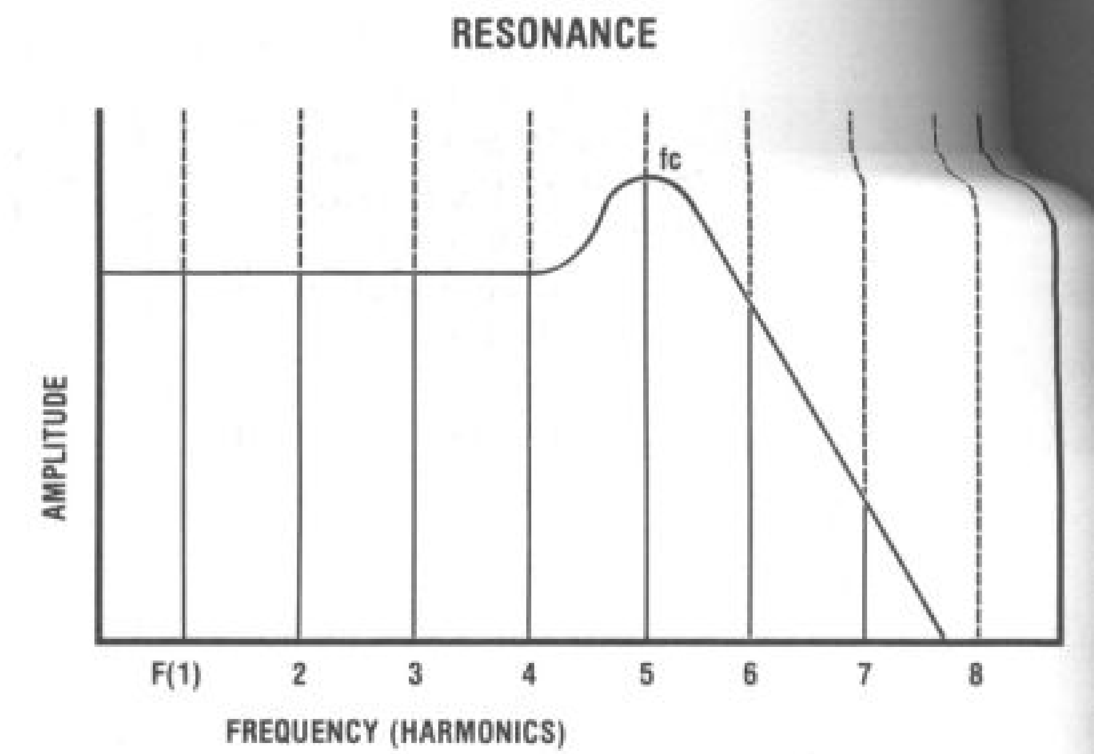
The two most important controls on a typical filter bank are **filter cutoff** and **resonance.**

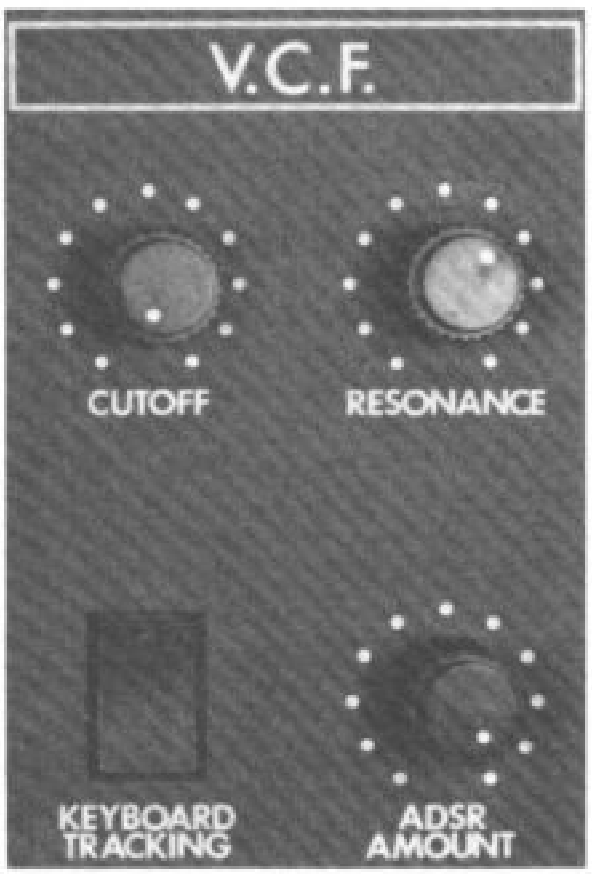


VCF (Filter). *Siel 600*

Filter Cutoff

The filter **cutoff** knob or lever is what you use to determine at which frequency in the harmonic spectrum the filter will take effect. If it is all the way down or to the left the filter is closed and only the fundamental frequency will be heard. As you turn the knob to the right, you open the filter and thus allow more and more harmonics to pass through. The more you open the filter, the brighter the sound.





VCF (Filter). *Siel 600*

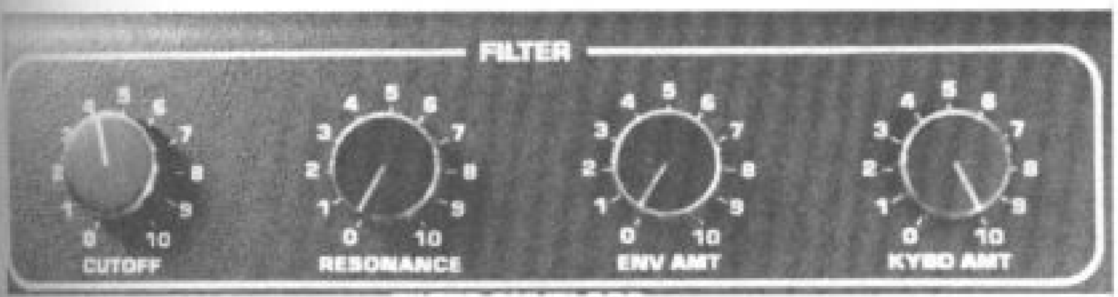
Resonance

Immediately adjacent to the **filter cutoff** control you will find the **resonance** control. *Resonance,* also known as *emphasis, regeneration,* or *Q,* controls the amount of feedback in the filter. When we take part of the already filtered signal and fee it back into the filter, it has the effect of emphasizing the frequencies right around the filter's cutoff point. This adds a "roundness" and "depth" to the sound and has many useful applications.

One interesting result of resonance is that as you increase the amount of feedback to the filter you reach a point where the filter itself begins to oscillate, creating its own audible frequency just as an ordinary oscillator would. This mean that if you choose to you can use the filter as another sound source instead of just as a sound modifier. The particular sound of an oscillating filter has a ringing quality to it and is useful for many different effects.

Envelope Amount

Most filter banks have a control labeled **envelope amount.** Th control determines the amount of effect that the filter envelop will have on the filter. With the envelope amount set at minimum, the filter envelope will have no effect at all on the filte so that even if you design a very dramatic filter envelope it will not be heard. With the **envelope amount** control set at maximum, the rising and falling effect of the filter envelope o the filter will be evident and the filter will perform according to the rate and level parameters set by the filter envelope.



Filter. *Sequential Prophet T8*

Keyboard Tracking

Keyboard tracking or *key follow* is a function that enables you to determine whether a sound's timbre will be the same or different as you move from one end of the keyboard to another.

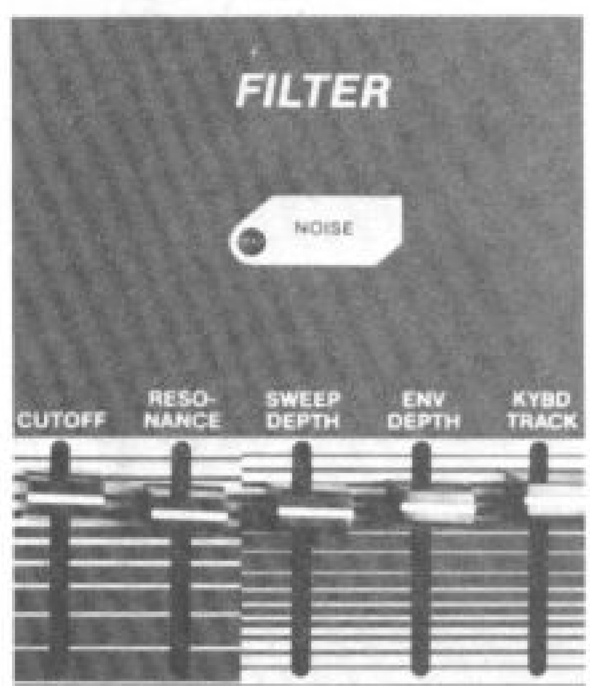
When you play a key on a synthesizer keyboard, the keyboard sends a voltage-control message to the oscillator instructing it as to the appropriate pitch. At the same time, a message is sent to the filter so that the filter cutoff point will change proportionally to the frequency of the note being played. This is referred to as keyboard tracking. If this were not the case, this is what would happen: Imagine that you have set the frequency cutoff point at 440 Hz. If keyboard tracking were not in effect, then all the notes below A 440 would be unfiltered, and conversely all the notes above A 440 would be filtered. For certain effects this might prove useful, but in most instances it would be preferable for all of the notes on the keyboard to have approximately the same relative amount of filtering. That way all of the notes of a given sound would have more or less similar timbres.

Now, as I said, there are instances where you might want the lower portion of the keyboard to have a slightly or even drastically different timbre from the upper portion of the keyboard. For instance, most acoustic instruments do in fact have a considerable change in timbre as they go from a low register to a higher register. Being able to control keyboard tracking gives you the ability to replicate that characteristic. Most synthesizers allow you to turn the keyboard tracking either on or off, some offer a halfway setting, and there are a few that are variable.

Different Kinds of Filters

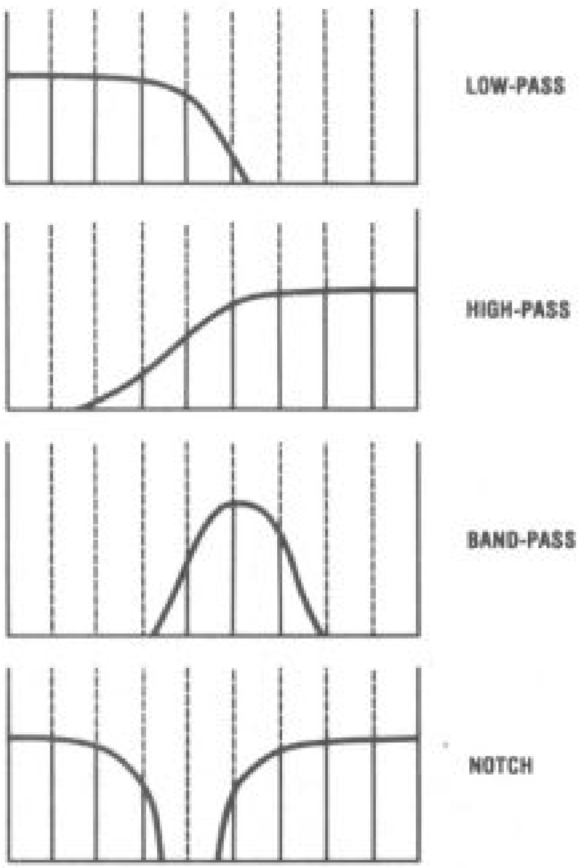
There are several different types of filters. The most common of these is the one I have already described—known as a *low-pass filter*. Although most synthesizers use a low-pass filter, there are some models that offer one- or two-filter options. I will describe them all for you now.

A low-pass filter is a circuit that allows the low fre-



Filter. *Chroma Polaris*

FOUR KINDS OF FILTERS



quencies of a signal to "pass" through while at the same time stopping all frequencies above a certain point.

A *high-pass filter* does just the opposite. It allows only that part of signal that is above a certain frequency to pass through the circuit, while blocking the lower frequencies.

A *band-pass filter* allows only those frequencies at a certain point to pass through, while blocking the rest of the signal on either side of that point.

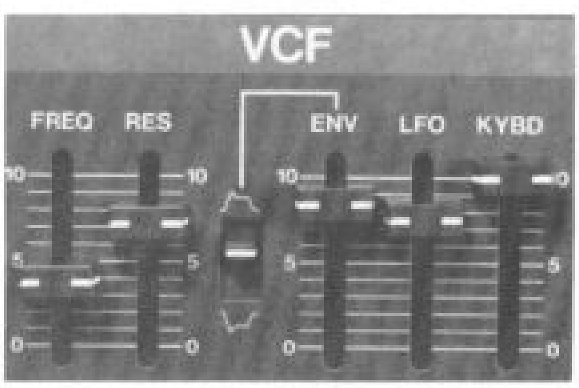
A *notch filter* does the opposite of a band-pass filter. It blocks the signal at a certain frequency while allowing the rest of the signal on either side of that frequency to pass through.

Each of these different kinds of filters offers a different way of changing the timbre of a waveform, but they all function in the same way—by subtracting a portion of the original audio signal.

As I indicated above, the vast majority of filters on synthesizers are low-pass filters, so for convenience's sake, unless I specify otherwise, when I use the word filter, from now on, I will be referring to a low-pass filter.

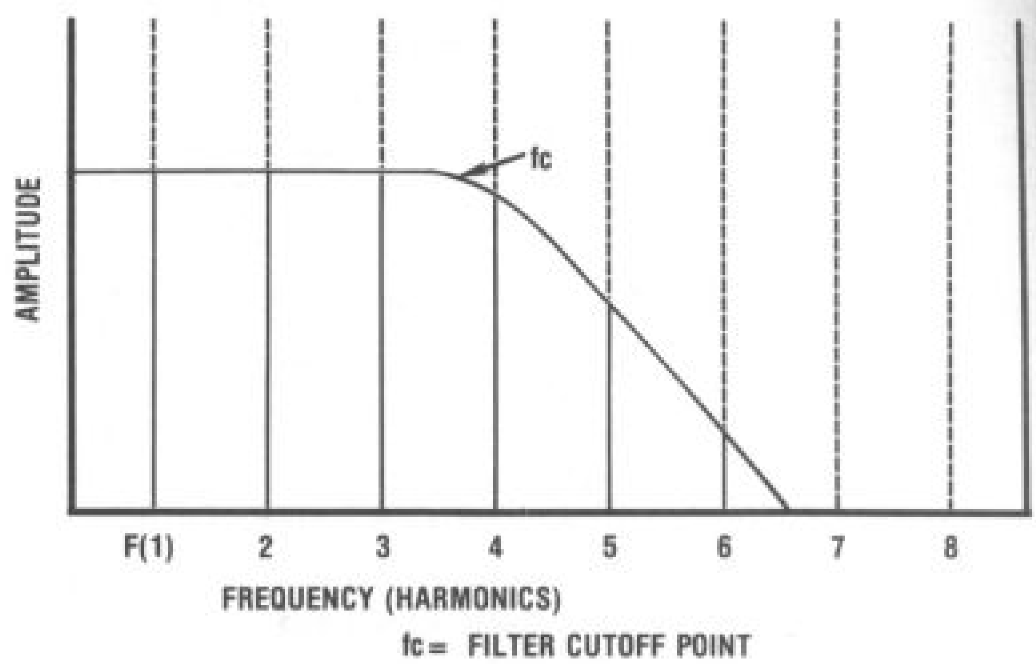
Two-Pole, Four-Pole

A fact of life about filters is that they are less than perfect. The diagram pictured illustrates how a typical filter affects a signal. The point at which the filter begins to attenuate (make weaker) the signal is called the filter cutoff point (Fc). The rate of attenuation is known as *rolloff*. If filters were perfect, we would see an immediate and abrupt rolloff, such that all frequencies above the filter cutoff point would completely disappear. In fact what occurs is a gradual attenuation or rolloff. The higher the frequency, the less signal is let through by the filter, until



VCF (Filter). *Roland Juno 106*

LOW-PASS FILTER

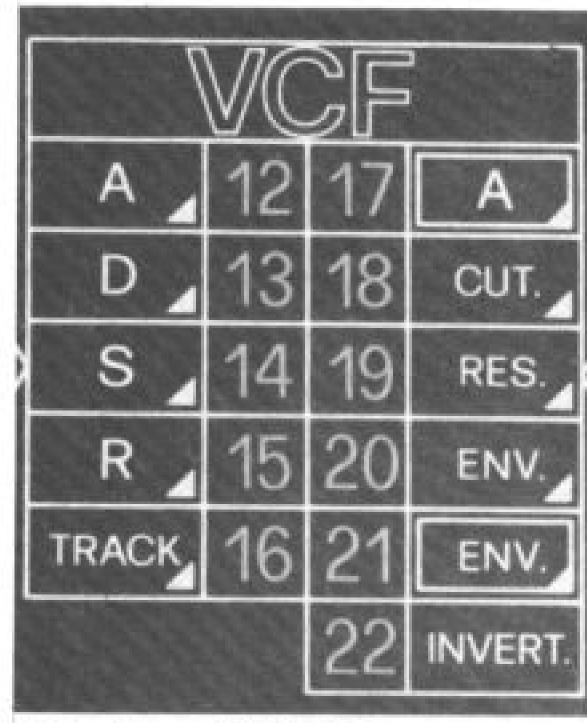


finally the signal is completely blocked. Rather than being a problem, this imprecision on the part of the analog filter (digital filters will be discussed later on) is very useful. In fact it is this imprecision that is responsible for the characteristic "warm" sound of analog synthesizers.

Different filters have different rates of attenuation or rolloff. We measure these differences by the change in perceived amplitude (dB) over the course of an octave. (For a more comprehensive definition of dB, see Glossary.) In other words, if a filter's cutoff point is set at 440 Hz and the perceived amplitude at 880 Hz (one octave higher) is 12 dB less than what it was at 440 Hz, we say that the filter has a −12 dB/octave rolloff characteristic. Every 6 dB of rolloff is referred to as a *pole*—so a −12 dB/octave filter and a −24 dB/octave filter are referred to respectively as a two-pole and a four-pole filter.

Most synthesizers today employ a four-pole filter (−24 dB/octave), although some, like the Oberheim series, offer a choice of either two- or four-pole. Actually, the only reason I have bothered to explain the difference between two-and four-pole filters is because for some reason these are the kind of terms that are bandied about in conversations between synthesists, and to a beginner the distinction might seem important. It's not. As I said, almost all synthesizers today use a four-pole filter, so if you were confused by anything in the last three paragraphs, forget about it. Pay attention to what's next though.

Filter. *Oberheim OB-8*



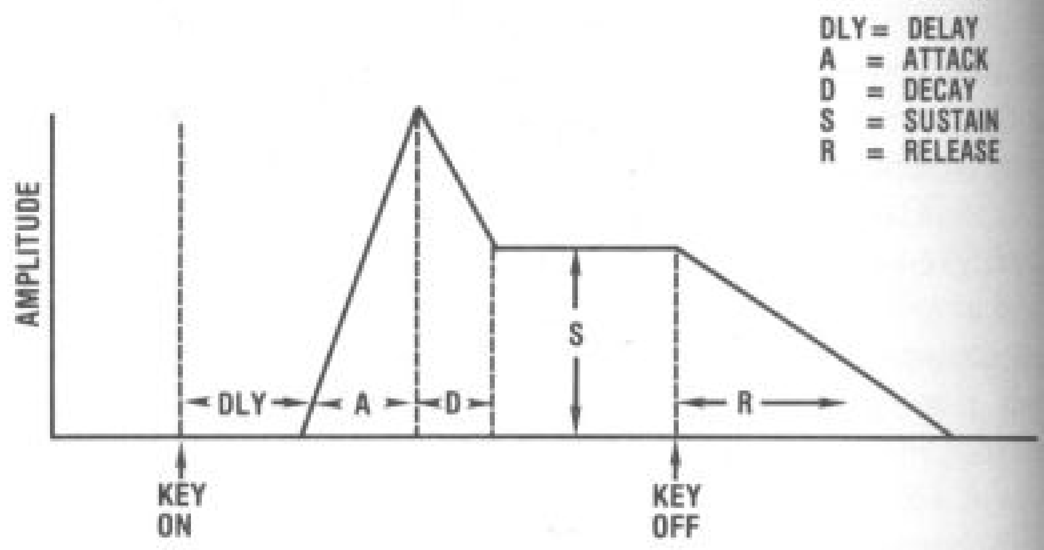
VCF edit map (Filter). *Bit One*

More about Envelopes

In our first discussion about envelopes, we talked about the most common one—the four-stage ADSR envelope. I'd like to introduce you to some other types of envelopes that you are likely to run into. The first is similar to the ADSR and is known as the DADSR. It is a five-stage envelope that is virtually identical to the ADSR except for the addition of a single stage called *delay* that occurs at the very beginning of the envelope. This added feature enables you to set a delay of as much as 10 seconds (on some systems) between the time you press a key and when the rest of the envelope actually begins. This can be useful for creating double attacks on a multitimbral system. (A multitimbral system is one that lets you play two distinct sounds upon pressing a single key.)

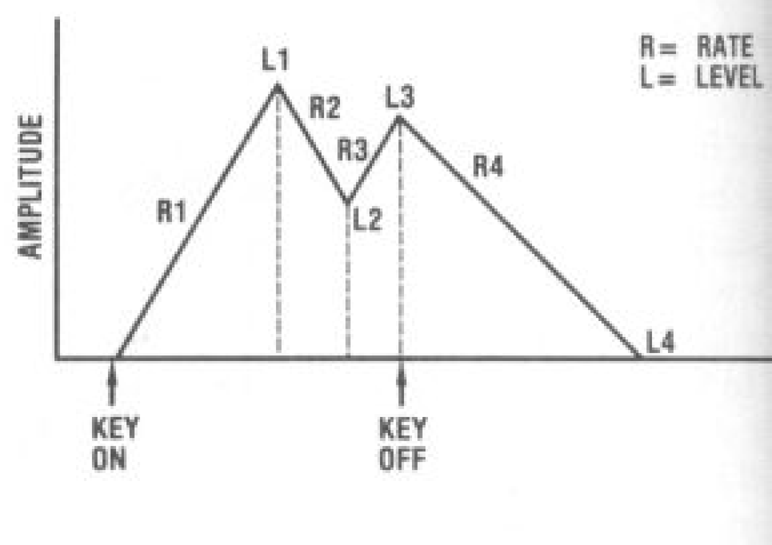
Another type of envelope that we are seeing more of is the eight-stage envelope, introduced originally by Yamaha in the DX7 and picked up by Casio for their CZ101. The eight-stage

DADSR FIVE-STAGE ENVELOPE



envelope allows you to set four rates and four levels. If you remember, the common four-stage envelope (ADSR) allows y to set three rates and one level. As you can see, the eight-sta envelope offers four extra parameters for designing an envelo An easy way of visualizing the eight-stage envelope is to imagine that the four level settings are the poles of a circus tent How you set the four rates will determine how close togethe those four poles will be. Remember, *level* represents the amplitude (or volume) of a signal (or sound) and *rate* determines how long it takes a note to reach each new level. The main thing to keep in mind is that all these different kinds o envelopes do approximately the same thing—they allow you describe the shape or contour of a sound.

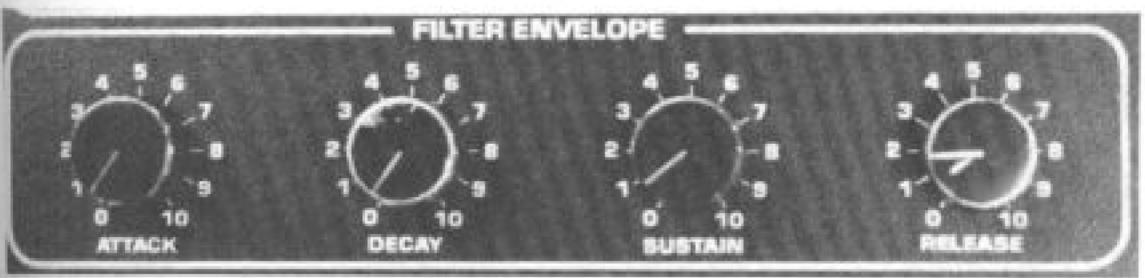
EIGHT-STAGE ENVELOPE



A Third Kind of Envelope

We have discussed the two main types of envelopes—the volume envelope and the filter envelope—but on some synthesizers there is a third kind of envelope.

This third envelope can be referred to as the *pitch envelope*. The pitch envelope is created by an envelope ger



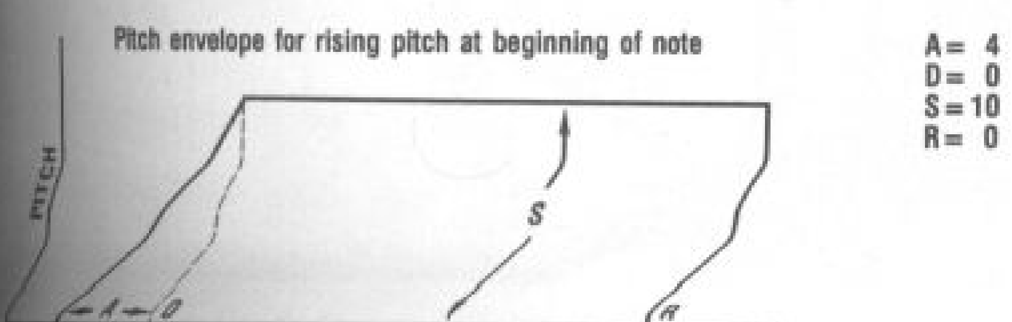
Filter envelope. *Sequential Prophet T8*

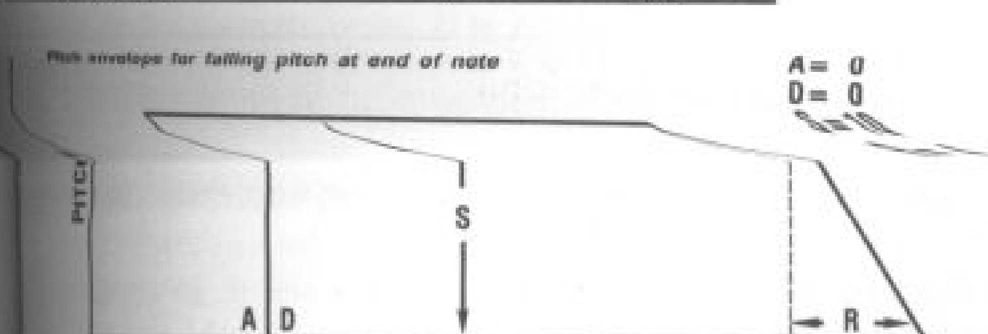
tor identical to the ones used to create the volume envelope and the filter envelope, the only difference is that in this instance the envelope generator is being used to control (with its changing voltage) the frequency (pitch) of the oscillator (as opposed to the amplifier or the filter).

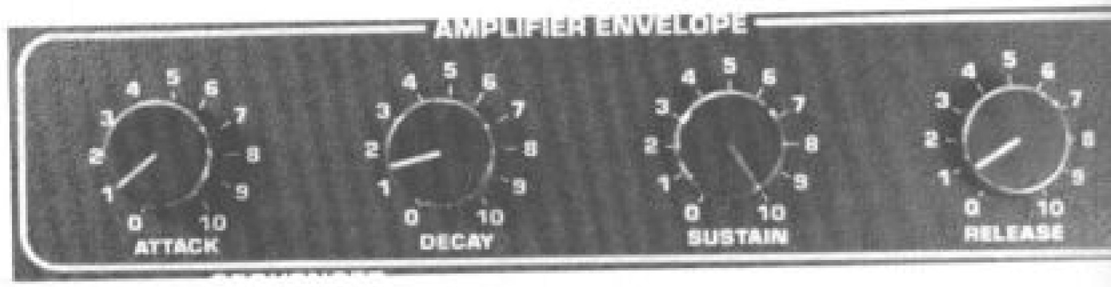
As you can see, this is very similar in some respects to the manner in which the LFO is used to modulate the oscillator, filter, or amplifier. As with the LFO, we can use an envelope generator to control either the amplifier, which gives us a volume envelope; the filter, which gives us a filter envelope; or the oscillator, which gives us our pitch envelope. The principle behind each of these envelopes is the same. As soon as we press a note on the keyboard, we trigger the envelope generator, which in this case will send a changing voltage to the oscillator instructing it to first raise its frequency (pitch) at a certain speed (attack), then return from its peak frequency to its sustain frequency at a certain speed (decay), then remain at the sustain frequency for as long as the key is depressed (sustain), and then finally return to its original frequency at a certain speed (release).

What we are able to achieve with all this is the ability to program a sound with a built-in pitch bend. This can be useful in trying to re-create a flute, for example, which has as

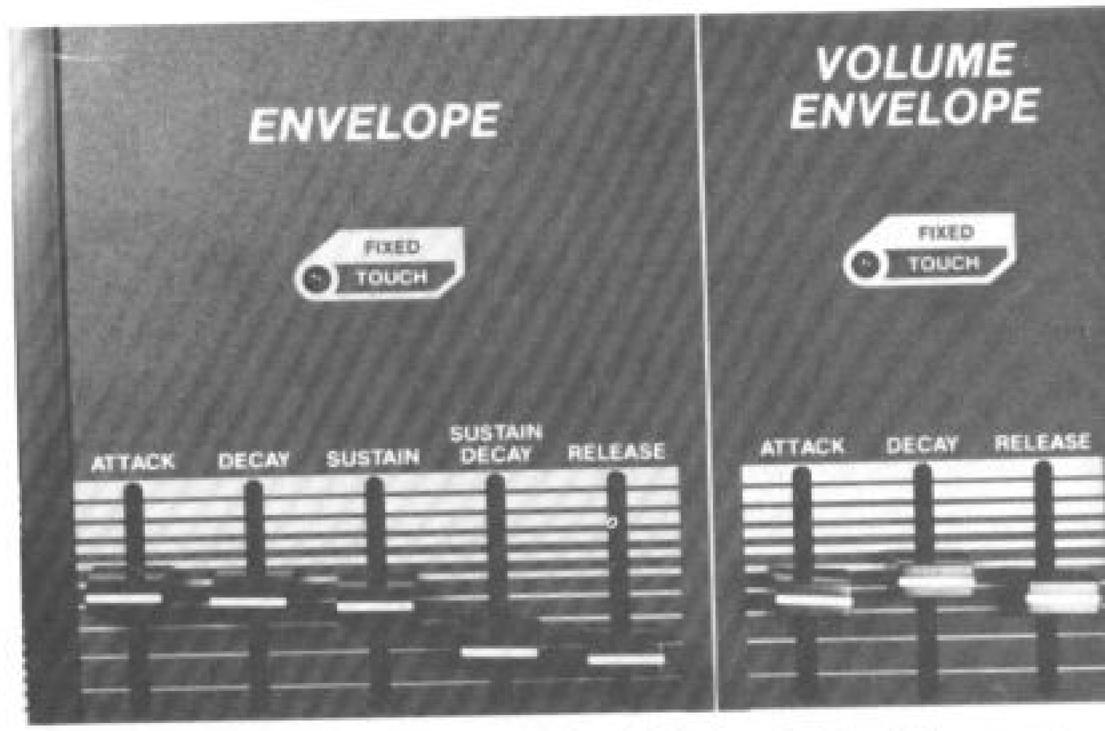
PITCH ENVELOPES







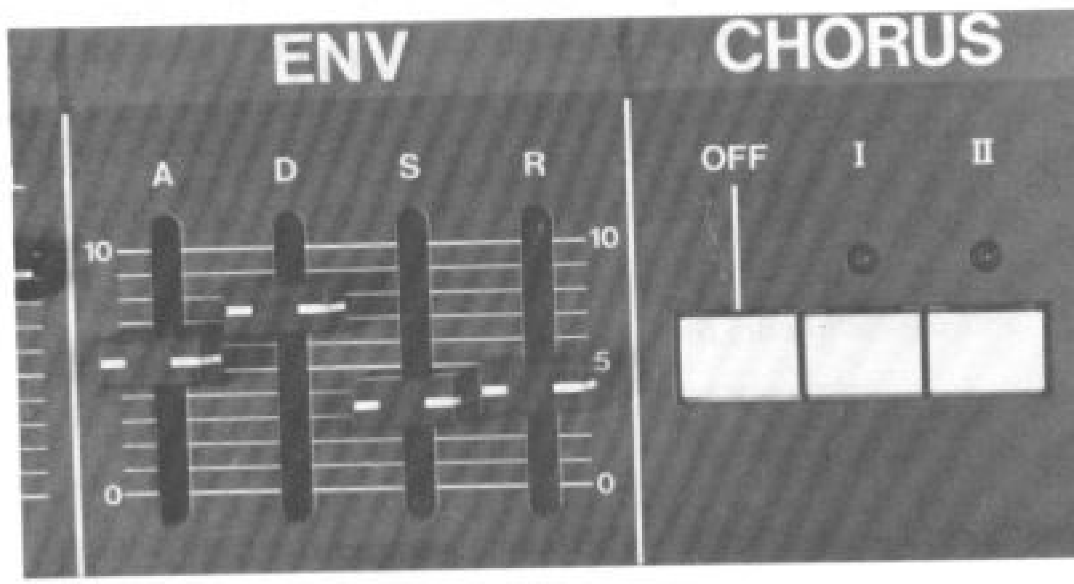
Amplifier envelope (Volume envelope). *Sequential Prophet T8*



Filter envelope. *Chroma Polaris*

Volume envelope. *Chroma Polaris*

one of its characteristics a slight shift in pitch at the very beginning of each note. Another application might be in desi ing a tom-tom with a dropping pitch. Some of these maneu may seem subtle to you at first, but you will soon discover that it is these very subtleties that make all the difference.

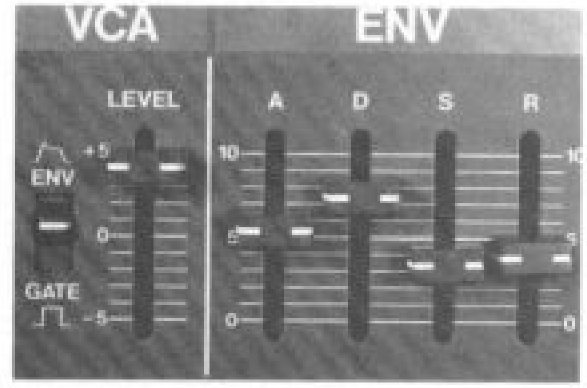


Volume envelope. *Roland Juno 106*

More about the Amplifier

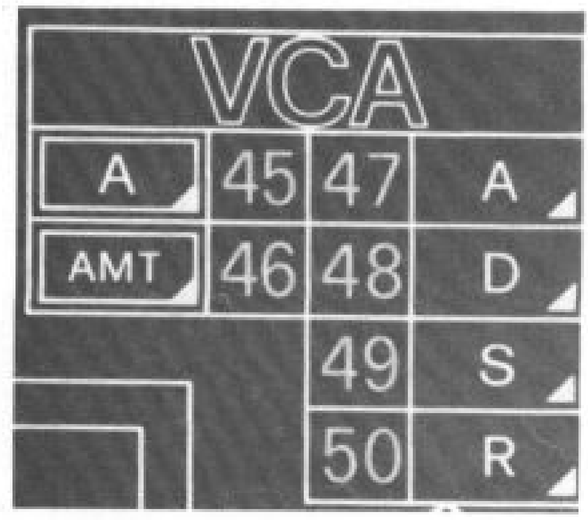
Actually, there's not a hell of a lot more to say about the amplifier, except, don't let the fact that it doesn't always a pear on all synthesizer consoles confuse you too much. Its main function is to be controlled by either the volume env or the LFO. It's there, you just can't always see it. In any it's important to know what it does – it regulates the volu

One feature that does sometimes crop up in the **VCA** section is a **gate/envelope** selector. The term *gate* refers to the opening and then closing of the signal path that occurs when you press a key. On a synthesizer we are able to determine the precise shape of a note's volume using an envelope generator working in tandem with the amplifier. Early electronic organs had no such envelopes. Their sounds did not fade in or out gradually. They were either on or off. And when they were on, their volume remained at a constant level. Instead of their volume being controlled by a changing envelope, it was controlled by an opening and closing gate. Some synthesizers have a switch that lets you select between a regular programmable envelope and a simple fixed gate. You would use a gate if you were going for a sound that approximated an electric organ feel. When you use a gate instead of a programmable envelope, the sound will have a short (zero) attack, no decay, steady sustain, and no release.



VCA (Amplifier). Roland Juno 106

One other thing: Do not confuse the amplifier (VCA) with the master **volume** control on the synthesizer. In fact, the master **volume** control is just another amplifier; but when we talk about synthesizer components, there is a tendency to take it for granted just as we take for granted the fact that the **mix** and **balance** controls on a synthesizer also employ amplifiers. We reserve the term amplifier (VCA) to refer to that particular amplifier that plays a crucial role as one of the six main components in the basic synthesizer voice. The main difference between it and the other amplifiers is that it is a voltage-controlled amplifier and is capable of changing the volume of a sound automatically over time (as in a volume envelope or for vibrato). Understand? Good.



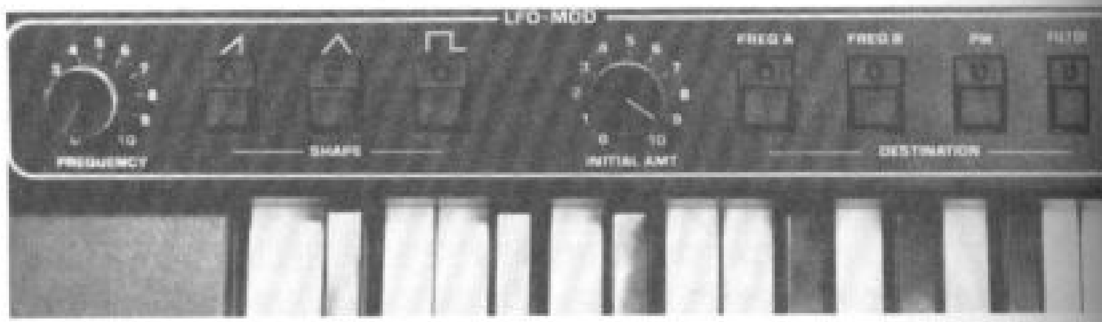
VCA edit map (Amplifier). *Bit One*

More about the LFO

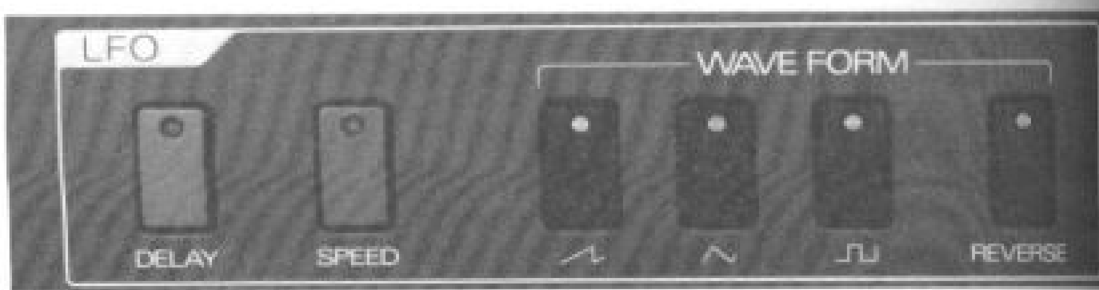
Well, here we are again. It seems like only yesterday I was explaining the basic principles of modulation to you. Are you sure you want to know more? O.K., O.K.

We know that the LFO is used to modulate a source in several different ways. If it is routed to the VCA, it affects the volume giving you tremolo. If is routed to the VCO, it affects the pitch giving you vibrato. If it is routed to the filter, it affects the frequency cutoff point giving you filter modulation.

There are three main controls on most LFO banks. The first one is **rate**, the second one is **amount**, and the third one is **delay**.



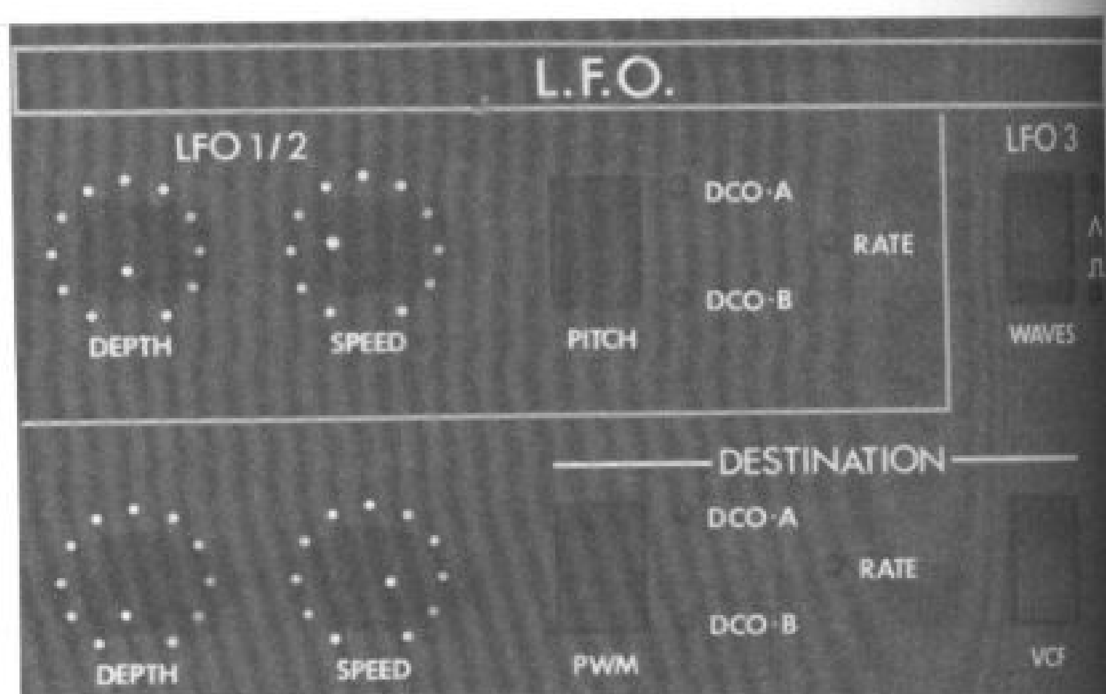
LFO. *Sequential Prophet T8*



LFO. *Kawai SX-210*

Rate

Rate (also known as **speed** or **frequency**) is the knob or lever that you use to determine the frequency (speed) of the low frequency oscillation. This rate usually falls somewhere between once every 4 seconds and twenty times a second. Typical vibrato is somewhere around six times (cycles) a second.



LFO. *Siel 600*

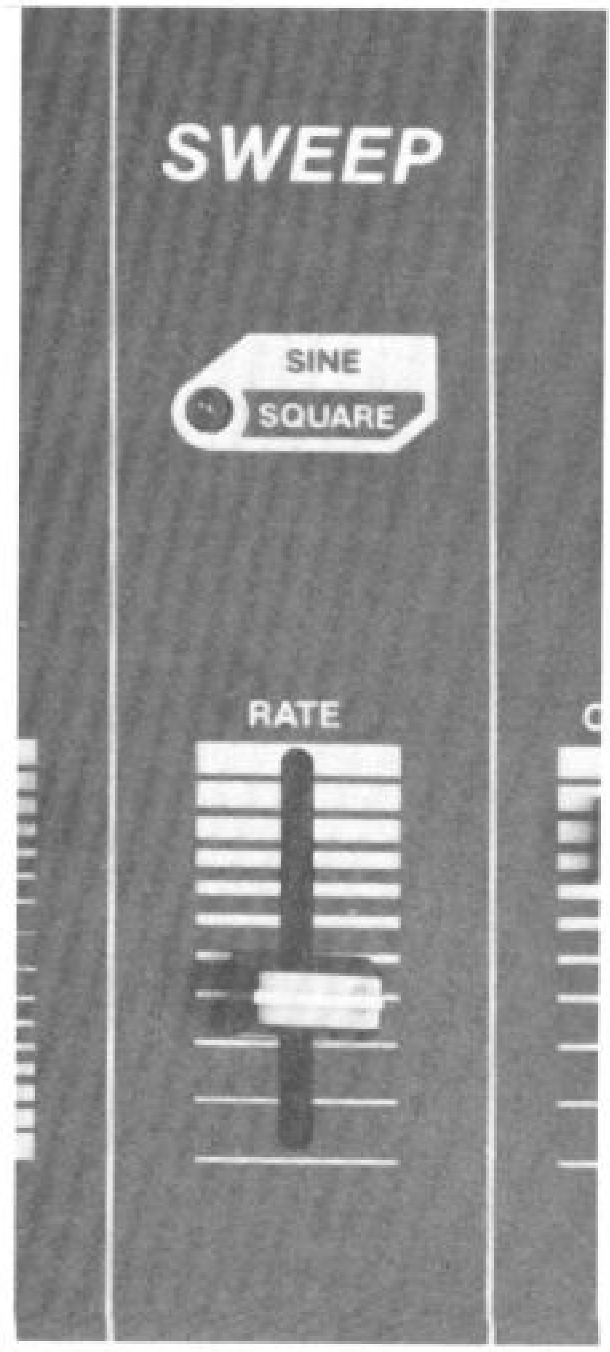
Amount

Amount is a control that determines the amplitude of the LFO. Now, if the LFO were an audible signal, amplitude would translate into volume. But in this case, that's not necessarily so. Because the LFO is being used as a modulator and is itself not directly audible, how its amplitude is expressed depends on what component is being modulated. For instance, when you increase the amplitude of the LFO modulating the oscillator

(VCO), you wind up with an even greater change in pitch as the oscillator sweeps up and down. Likewise, when you increase the amount (the amplitude) of modulation to the filter (VCF), you increase the range of the filter's sweep. And, similarly, when you increase the amount of modulation to the amplifier (VCA), you increase the degree to which the volume changes.

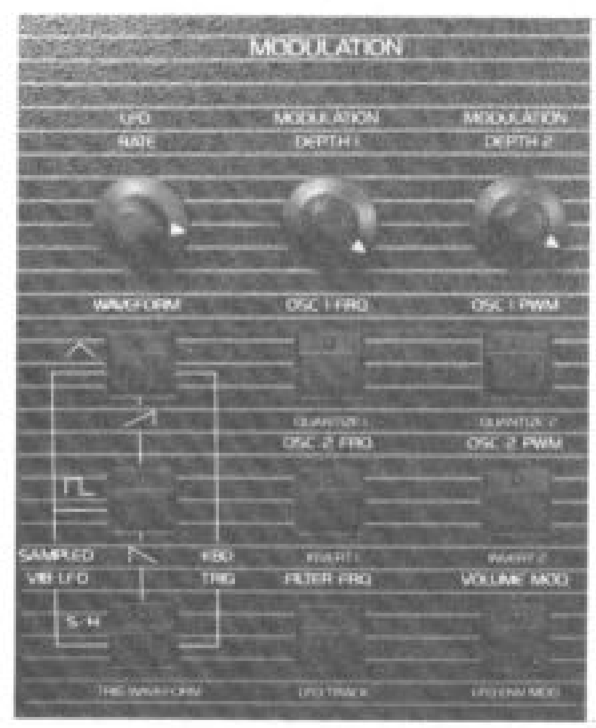
Delay

Delay is a control that allows you to set up an automatic delay of anywhere between 0 and 5 seconds so that when you press a key the effect of the LFO will not be heard instantly, rather it will fade in gradually. The delay's effect on the LFO is identical to the attack portion of an envelope's effect on the VCA. It introduces the signal slowly. This effect contributes enormously to the real-time performance characteristics of a synthesizer. It enables you to more realistically re-create the natural vibrato of instruments like violins or clarinets, and all this helps make the synthesizer more of a human instrument. It is just one more way of making the sound you produce that much more interesting.



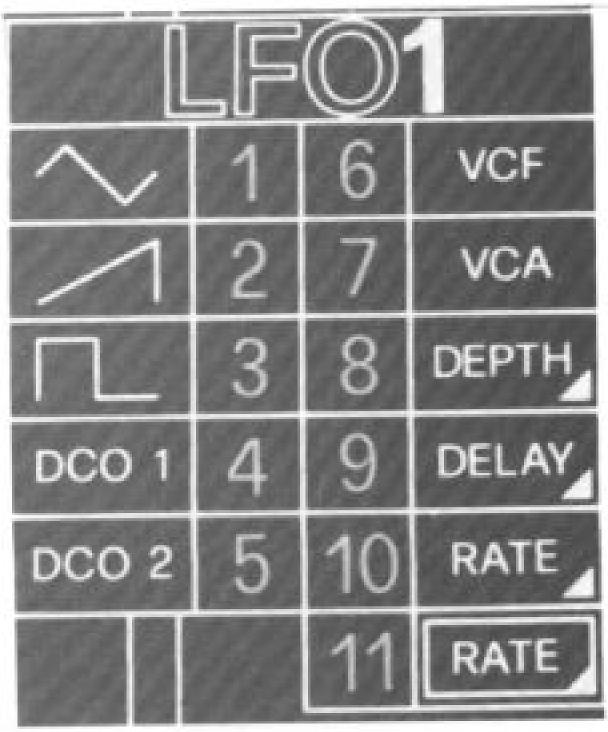
LFO (Sweep). *Chroma Polaris*

> Now is as good a time as any to talk a little more about the vast differences among synthesizers, especially where LFOs are concerned. As far as the other five components go—the VCO, the VCF, the VCA, and the two ENVs—although you will find variations from system to system, for the most part they are pretty similar. However, when you start talking about LFOs you begin to discover enormous differences between systems. Some let you route the LFO only to the VCA and the VCO. Others offer no **delay**. And not all systems locate the LFO controls in the same place. For example, some synthesizers provide a single **depth** or **modulation amount** right next to the LFO **rate** control. In this case there is normally an LFO **on/off** switch adjacent to each component that is to be modulated, i.e., the VCO, VCF, or VCA. Other synthesizers, instead, have a separate LFO **amount** control located in the vicinity of each component that is to be modulated. In other words, some systems let you set one LFO amount and then turn it on or off at each destination and others give you a separate **amount** control at each destination.
>
> I will talk more about these differences as the need arises. Obviously, it is a major difficulty in a book such as this; but for now just be aware that these differences exist and that the only way to deal with them is to first understand the basics and then apply them to whatever system you happen to be working with. Don't worry, it will all become very obvious to you after a while.



LFO. *Chroma Polaris*

O.K., let's make some noise.



LFO edit map. *Bit One*

Exercise 2. The Oscillator Start in neutral. Locate the oscillator bank on the panel. If your instrument has two oscillators per voice, identify the separate controls for each. Find the **mix** or **balance** control that enables you to balance the levels between both oscillators. As in Exercise 1, play around with different combinations of waveforms, except this time experiment with being able to blend the waveforms of both oscillators together. Not only can you mix different waveforms together, but with the **balance** control you can see how they sound together at different strengths.

Next, experiment with changing the frequencies of either one or both oscillators. Listen to the effect of different intervals—octaves, fifths, etc. While you're doing this, remember that you can control the relative levels of the oscillators using the **balance** control.

Try detuning both oscillators ever so slightly. Start off by getting them as close to unison as possible, and then slowly tune one of the oscillators a little sharp or flat. Listen to the different kinds of chorusing that can be achieved depending on how far apart the two frequencies are set. This is a simple effect which can yield a thick, fat timbre.

Try to get a feel for the enormous variety of different timbres that it's possible to achieve in the oscillator section alone. Remember, this is just your source material; and you haven't even begun to mess with it yet.

Exercise 3. The Filter Continue from where you left off in Exercise 2. First fool around with the **filter cutoff** control (also labeled **frequency** or just **filter**). Turn it one way and the sound gets bright. Turn it the other way and it gets dull. That's it folks—subtractive synthesis. At first this control may seem coarse; but as more elements come into play, you'll see that the slightest adjustment of the **filter cutoff** will make all the difference in the world.

Next, play with the resonance . . . but, BE CAREFUL! Resonance is feedback and at high levels this can hurt your ears. Don't be a chump. Now, **resonance** is a tricky control. The degree to which it affects a sound can be dependent on a number of different elements. For instance, resonance will be especially pronounced when it is introduced briefly as part of the filter envelope. You'll see what I mean when we get to envelope generators. For now, try to get a sense of what it adds to the sound and how it interacts with the **filter cutoff** control.

After you've spent some time doing that, try this: Turn your resonance off. Disengage (turn off) all of the selectable waveforms from both oscillators. This should leave you with a voice path that is open but that contains no waveforms, i.e., you should hear no sound. Now, carefully (watch your ears) put your **filter cutoff** control somewhere around 6 or 7, and then slowly turn **resonance** to the max. What should happen is that the filter will begin to oscillate, giving off a high-pitched ringing tone. This is a usable sound. (If none of this is happening, check to see that the **envelope amount** control and the **sustain** knob on the filter envelope are both fully open, or raise the filter cutoff point some more.)

Exercise 4. The Volume Envelope Make sure you've turned that resonance down and pick yourself a nice waveform or two. Play around with the ADSR controls for the volume envelope. Remember everything I said about the relationship between peak and sustain; unless **sustain** is set below max (peak), you will be unable to hear decay. Try to get a sense for the contour of the envelope and how each control affects a different portion of the envelope.

Design some simple envelopes. Make an envelope with a gradual attack and an even more gradual release (like a violin). Design an envelope with a short attack and no release (like an organ).

Make sure you understand the function of decay. Set a slight attack and then put **sustain** at 4. Now, as you move the **decay** control, listen to how the peak portion of the volume envelope drops to the relatively low sustain level. This will happen slowly or suddenly depending on where you've set the **decay**.

Here's another confusing instance you should be aware of: If **sustain** is at 0, you will never actually get to hear the release portion of the envelope. The note will end as soon as it reaches the end of the decay portion of the envelope. In this instance the **decay** control acts almost as if it were a **reluase** control.

One more thing to think about: If you set a long decay and take your finger off the key before the end of the decay, you will never actually hear the sustain or release portions of the envelope. This can be used for an interesting effect. If you set a long decay with a medium sustain and no release, you can achieve two different envelopes depending on when you remove your finger from the key. If you lift your finger before the sustain portion of the envelope, then the note will trail off according to the **decay** setting as if it had a long release. If, on the other hand (or the other finger), you keep your finger on the key into the sustain portion of the envelope, then when you lift your finger off the key the note will end abruptly according to the short **release** setting. What this means is that you can design an envelope that will perform two different ways according to how you play a note. Try it. It may seem confusing at first, but it will help you understand exactly what is happening in the envelope generator. Also, this is the kind of tidbit that is liable to come in very "handy" one day.

Horrible puns aside, try to get clear in your head how the different portions of an envelope interact. Some of it is obvious, and some of it is not so obvious. Take your time. It's not like you have a train to catch.

Exercise 5. The Filter Envelope The best way to demonstrate the filter envelope is to make sure that you have some resonance applied to the filter. Set **resonance** at about 5 or 6. (The reason for using resonance in this exercise is that it exaggerates the effect of the filter.) Now make sure that your volume envelope is not too short, in other words, see that the sustain is not too low and that you have some release. Finally, check that the **envelope amount** control is all the way up. Now you can get cooking.

First, experiment with different filter envelopes superimposed over your simple volume envelope. Bear in mind that you can extend the filter envelope beyond the boundaries of the volume envelope—you just won't hear it. For instance, if your volume envelope has no release, then no matter how long a release you set for the filter envelope it won't be heard. By the same token, if your volume envelope has a long release and your filter envelope has no release, you are likely to hear a dramatic drop-out of the harmonics in the filter envelope just as the volume envelope goes into its release. I point these things out to make you aware of the boundaries of these envelopes and how they interact.

Spend some time examining the relationships between the **decay, resonance,** and **filter cutoff** controls. If you put the filter envelope's **sustain** at zero and the **resonance** and **filter cutoff** controls at 7 or 8, then by slowly adding a small amount of decay in the filter envelope you'll be able to introduce a sort of *wow* into the beginning of your sound. It may take some adjusting of the parameters I mentioned to achieve this, but this should provide you with a good example of how each of these different functions affects and influences the others.

In Exercise 3 we used the **resonance** control to create feedback that caused the filter to oscillate. Do this again, and this time apply the filter envelope to the oscillation. (Make sure all the waveforms in the regular oscillator are turned off and that only the filter is oscillating.) Much to your surprise, what should be happening is this: When you apply a filter envelope to an oscillating filter, you cause the pitch of that oscillation to change. This shouldn't be that much of a surprise, if you think about it. All that is happening is that the filter envelope is changing the filter cutoff point, that is, the frequency at which the filter takes effect. **Resonance** emphasizes those frequencies, and in this case its setting is so high that the filter is oscillating at those frequencies. The filter envelope is therefore controlling the frequencies of the oscillation. In this instance the filter envelope is performing as if it were a pitch envelope. Hearing how an envelope affects pitch will give you another perspective on the actual shapes and contours that an envelope generator creates.

This is just another one of those peculiar patches that you should file into the back of your brain as it has numerous applications for creating sound effects, pitched percussive sounds, etc. But more importantly, understanding why this occurs will give you insight into how each of these components operates in relation to the others and that will make you a better synthesist. Put back your regular waveforms into this patch and fiddle around some more.

Note: Some synths use a single envelope generator for both the filter and the volume envelopes. Since very often we design filter envelopes that follow the basic contour of the volume envelope anyway, this is not such a tragedy; but obviously these systems offer no real flexibility in designing a filter envelope. Some of these systems compensate somewhat by providing a *gate envelope* (a simple on/off envelope that has no level changes) for the volume envelope thus leaving the ADSR controls free to control the filter envelope. In any case, if your system only has a single envelope generator a lot of this will not apply—tough noogies.

Exercise 6. The Pitch Envelope Not all synths offer a separate pitch envelope. However, many are set up so that the filter envelope can alternatively function as a pitch envelope. This is different from using the filter envelope to change the pitch of an oscillating filter as we did in the last experiment but only in that, in this instance, we would be applying the filter envelope to the main oscillator (the VCO) and thereby controlling the pitch of the principal sound source. (If your system doesn't have a separate pitch envelope, you can do most of this exercise using a filter envelope applied to an oscillating filter as in Exercise 5.) Whether your system has a separate pitch envelope or whether the filter envelope is doubling as a pitch envelope, you are accomplishing the same thing—controlling the pitch of the oscillator.

Try to determine how the **sustain** control on a pitch envelope affects the ceiling of the pitch and also how this relates to the envelope's peak. If there is an **amount** control for the envelope, this will also affect the pitch parameters. The principles for controlling the range and direction up and down of pitch bend are the same as they were for our last experiment in which we controlled the pitch of the oscillating filter. If you set **attack** at zero, the pitch will start at its peak frequency. If you set a gradual attack, it will take a short while until the pitch reaches its peak frequency. If you set **sustain** below maximum, the pitch will either drop suddenly or gradually to the lower frequency depending on how much delay there is. Any amount of release will cause the pitch to fall from the sustain point.

Try creating a pitch envelope whose pitch rises when you strike the note. Keep in mind the relationship between peak and sustain to avoid any unwanted pitch jumps.

Good, now try to create a pitch envelope that drops in pitch as soon as you release the key. Go ahead, you can do it. It's easy. Spend some time examining how all the other parameters we've discussed, like the **amount** control and the volume envelope, affect the pitch envelope. Having fun?

Exercise 7. The LFO Go back to neutral. As I warned you before, different systems have different LFOs; but we'll try our best.

First, let's apply the LFO to the pitch of the oscillator. Find the LFO **amount** control and turn it to 10. If you don't find it in the LFO section, look for it at the destination; in this case that would be the oscillator section (the VCO). Look to see if you can select your destination; for instance, on some synths there is a switch in the LFO section that allows you to choose either VCO, VCA, VCF, or PWM. For this example we want to modulate the pitch so we'll route the LFO to the VCO. Got that? (On other systems you choose your destination *at* the destination as in the above when we found the **amount** control in the oscillator section.)

If you have a choice of LFO waveforms (usually triangle, sine, or square), choose a triangle or a sine wave. If there is an LFO **delay** control, turn it to zero. Now when you play a note, you should hear the pitch of the note wobble dramatically.

Using the **rate** control (also called speed or LFO frequency), change the speed of the modulation and listen to how that affects the sound. Try changing the LFO **amount** and see what that does to the pitch range. The more LFO you apply, the broader the pitch range. Now, try to add a little LFO delay so that *the LFO enters gradually once you've played a note.* This is, you guessed it, vibrato. Not bad. Now try some tremolo. (Hint: Use the LFO to modulate the VCA.) Next, route the LFO to the filter (the VCF) for some lovely filter modulation. Take your time doing all this. Get a feel for what the LFO can do.

Finally, if your system is capable of it, apply the LFO to the pulse width. This is PWM or pulse width modulation. Some systems have an independent LFO just for PWM. In any event, set it up and then listen to how it compares with filter modulation.

One thing I should point out here is the psychoacoustic (how we interpret sound) phenomenon by which we very often perceive an increase in a sound's harmonic spectrum as an increase in volume. In other words, when we open the filter of a sound and let in higher harmonics, the sound is not necessarily getting any louder though it sounds as if it is. I mention this so that you don't get too confused when comparing the effect of the LFO on the VCA with its effect on the VCF. In both instances you will hear what you perceive to be a change in volume in the modulation. The difference is that when you modulate the filter (VCF) you will hear a dramatic change in harmonic content as well.

Try all these different kinds of modulations—vibrato (LFO to VCO), tremolo (LFO to VCA), filter (LFO to VCF), and PWM—using a square wave instead of a triangle. Modulate the filter with a square wave and play a chord. Adjust the speed to your liking. Do the same, except modulate the VCO. This time, because you are using a square wave to modulate the VCO, the pitch will jump up and down abruptly. All of these are really, you should excuse the expression, "neat" effects. File them away.

O.K., now that you have familiarized yourself with the LFO (and if you haven't spent a long time on this exercise, you are definitely rushing) what you should do is go back and see how to incorporate the LFO into the kinds of patches we've already covered. For instance, what happens if we apply LFO to a filter envelope that has a very long attack? What does it sound like to apply an LFO to an oscillator whose pitch is already being modulated by a pitch envelope? You see, we can now begin to apply these few components in any number of different combinations. The result is an infinite variety of sounds and effects.

Great, let's put it all together.

Exercise 8. Design a patch whose volume envelope has a short attack and a long release.

Exercise 9. Design a patch whose volume envelope has a gradual (long) attack and a sudden release. And add some slight chorusing.

Exercise 10. Design a patch with a dramatic decay in the filter envelope (use resonance).

Exercise 11. Design a patch that uses an oscillating filter as its sound source.

Exercise 12. Design a patch, with chorus, that has a volume envelope with a short attack and long release, and a filter envelope with a gradual attack and long release. Apply the LFO to the filter using a square wave.

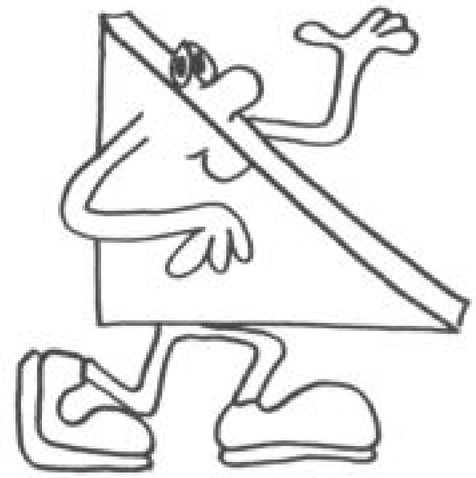
Exercise 13. Design a patch using a pulse wave and set its oscillators an octave apart. Add a filter envelope that enters at the beginning of the note but fades away before the end. Modulate the pulse width with the triangle wave of the LFO.

Exercise 14. Design a patch whose pitch rises when you play the key. (If you have no pitch envelope, then oscillate the filter.)

Exercise 15. Design a patch whose pitch falls when you play the key.

Exercise 16. Design a patch whose pitch first rises and then falls.

Exercise 17. Design a patch whose pitch falls during the release portion of its envelope. Set a very long attack on the filter envelope. Modulate the filter (VCF) with the LFO using a square wave.



Exercise 18. Design a patch with chorus, and apply the LFO to both the VCF and the VCA. Set the LFO rate at a very slow speed. Create a volume envelope with a very long attack and release. Open the filter up. Have a cup of tea.

Exercise 19. Design a bass patch. Make the frequency low. Give the volume envelope a short attack and a long decay. Use little or no sustain. Raise the filter cutoff point, add some resonance and find a **decay** setting on the filter envelope that gives you a nice little burp at the beginning of your sound. Boogie.

Exercise 20. Design a string patch. Find the right combination of waveforms (for example: triangle for body, sawtooth for edge—mix to taste). Add some chorusing. Give the volume envelope a long stringlike attack and release. Design a filter envelope so that the harmonics enter briefly during the beginning of the note. When you do this, try to approximate the sound that the rosin and bow make on the string of a real violin. Add some vibrato (LFO to the VCO) with a 1 second delay. If after following this recipe you've managed to come up with a sound that is somewhere in the vicinity of a stringed instrument, then I sincerely hope you are prepared for the dire consequences of your seemingly innocent actions. You've just put 60 hard-working classical musicians out of work.

Where We Are

So, we have covered the six main components of the voltage-controlled synthesizer. Where does that leave us?

Well, first of all, we're more than 50 percent of the way toward being able to make whatever sounds we want. If you've been doing the exercises, you know the extent to which these six components allow you to define the characteristics of a given sound. Next we are going to take a brief excursion away from those main components and talk a little bit about performance controls—those are the levers, pedals, and joystick that allow you to tailor your performance, in real-tim such a way as to add that essential human element (or Vul element, as the case may be) to the sounds you create. Foll ing that, we will return to explore the plethora (that means there are a lot of them) of extra features that you are likely find on your typical synthesizer. This may seem a little ou sequence, but I don't want to numb your brains with all th conceptual stuff first. Performance controls are hands-on (foots-on) devices that are fun to use. So . . .

4

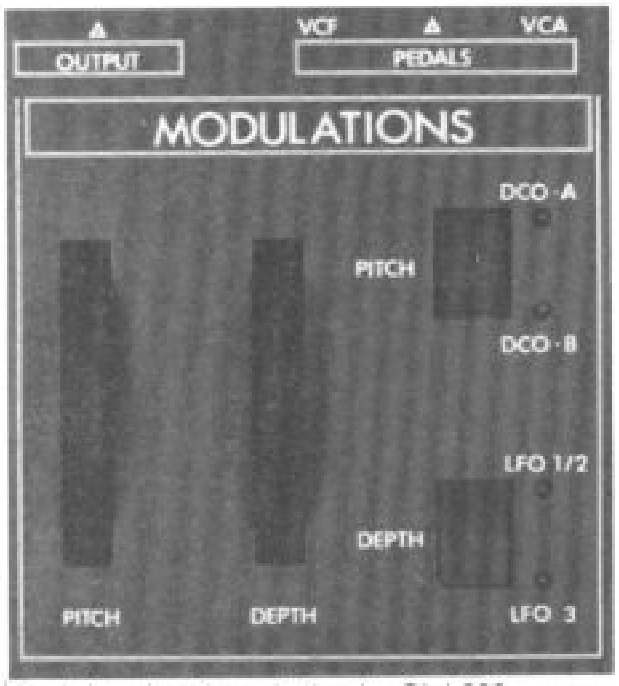
PERFORMANCE CONTROLS

Wheels and Levers

If you're wondering what those levers to the left-hand side of your synthesizer keyboard are doing there, well, I'm going to tell you. They are performance controls. We use them to change and alter a sound as it is being played. A good analogy would be a guitar player bending a note during a solo. He plucks the string with the pick in his right hand and bends the string with the finger of his left hand. We can do approximately the same thing on a synthesizer; we play a note with our right hand and bend or modulate the note with our left hand. The only difference is instead of bending a metal string we are "bending" a VCO or a VCA or a VCF. Here's what is going on.

There are typically two separate performance controls, usually located to the left of the keyboard. Sometimes they will be in the form of two wheels, and sometimes they

Performance controls and arpeggiator.
Oberheim OB-8



Pitch bend and mod wheels. *Siel 600*

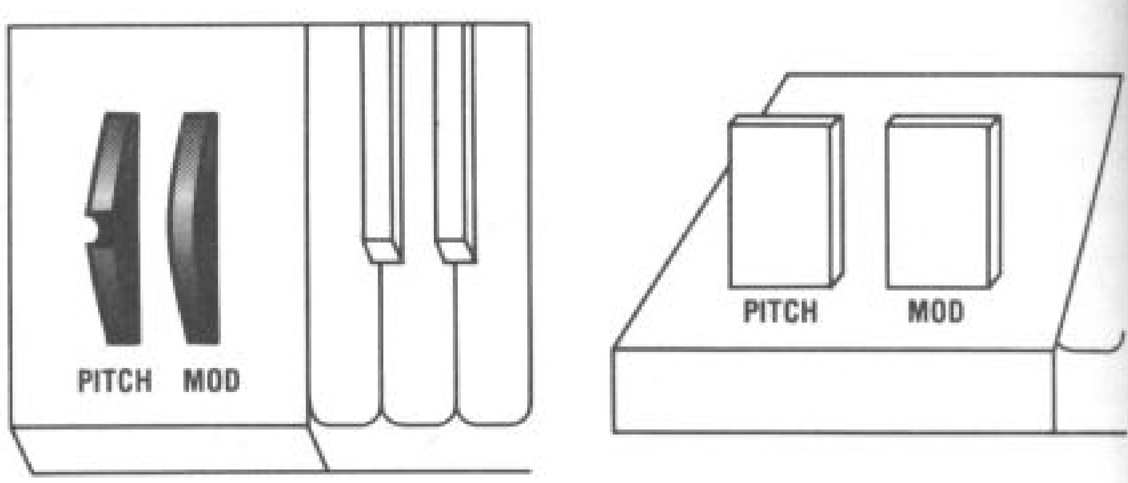
will be in the form of two levers. Usually one of them will be spring loaded, that is, it will be connected to a spring mechanism that will return it to a neutral position after you move it even slightly in either direction. The other will be free moving and stay wherever you leave it.

When we pull back or push forward on a wheel or lever, we produce a changing voltage. This voltage is then used to control those good old basic components, specifically the VCO, the VCA, and the VCF.

Typically one wheel or lever is assigned to control the frequency (pitch) of the VCO. (This would be the spring-loaded one.) This is called the *pitch bend wheel* (or lever). It can bend a note sharp or flat by as much as an octave. On some instruments the pitch bend range is fixed at a major or minor third, and on other instruments the range can be variable, that is, you can set it at a whole step or a major second or a fifth, etc.

The other wheel (or lever) is used to control the amount of LFO that is being sent to modulate the VCA, the VCF, or the VCO. This is called the *mod wheel* or the *modulation lever.* If you move the modulation wheel forward in the middle of a note, you will introduce tremolo, or vibrato, or filter sweep, depending on whether you are modulating the VCA, the VCO, or the VCF. Just as different systems offer different pitch bend ranges, systems vary when it comes to modulation wheel routings. Some only offer control of the VCA or the VCO. Some offer an independent LFO just for the modulation wheel, including its own **rate** control. Most systems operate by increasing the amount of LFO that is set by the main LFO. It might help to realize that a modulation wheel is nothing more than another LFO **amount** control, except instead of being located on the panel in the LFO section it is located to the left of the keyboard in a shape that is easy to manipulate in a performing situation.

PITCH AND MOD WHEELS AND LEVERS

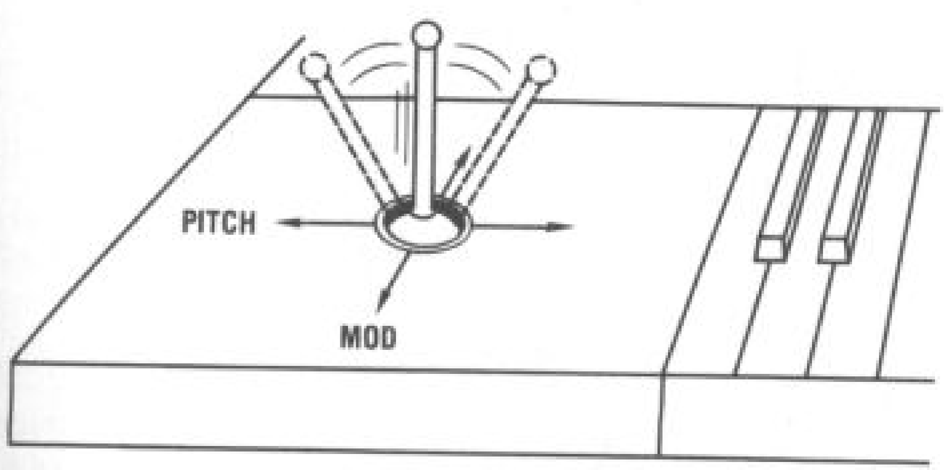


Both of these controls, the pitch bend and the modulation wheel (lever), give you the ability to turn the synthesizer into a real-time performing instrument. That is, they give you the ability to change the character of a note *as it is sounding,* as opposed to having to program it in advance. This is one of the few places in which it is possible to add that indispensible human element to an instrument that can otherwise run the risk of sounding lifeless and mechanical. The best thing about performance controls is that they're fun to use. It takes just a little practice to get used to their *throw* (how far you have to move them), and then you can swoop, dive, and gurgle to your heart's content.

The X,Y Joystick

Some synthesizers use a single *X,Y joystick* instead of two wheels or levers. One axis of the joystick will control pitch bend and the other will control modulation. A joystick can do virtually everything that a pair of levers can. The only difference is that it is completely spring loaded, so it would not be possible to introduce a modulation and then leave it set—as you could with a springless modulation wheel. If you are asking which of these performance controls is better, what it really comes down to is a matter of personal preference. They all do the job.

X, Y JOYSTICK

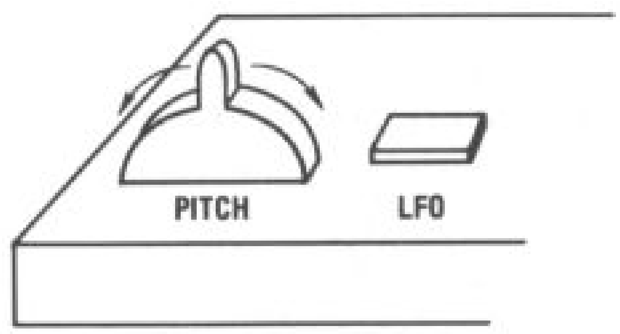


Other Types of Hand Controls

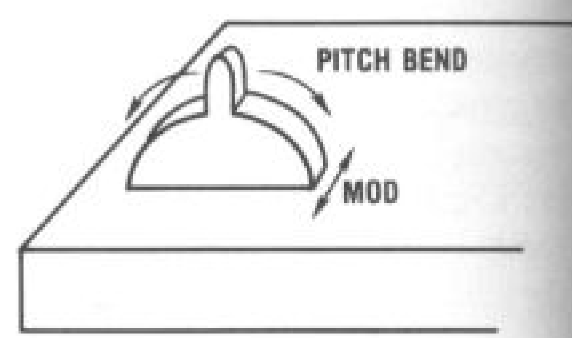
Another type of two-in-one performance control along the lines of the X,Y joystick is the *side-to-side push lever.* That is not really its name but it explains how it operates. You'll find them on many Roland synthesizers. It is a lever mechanism that moves side to side (as opposed to back to front) giving you pitch bend. If you push the entire mechanism forward, you get modulation.

Some synthesizers have an *LFO trigger switch*, which is usually found immediately adjacent to a side-to-side pitch bend lever within easy thumb's reach. This enables you to introduce modulation with a simple press of the thumb. If the system also has a **delay** control for the LFO, then you have the ability to closely approximate the natural movement of a wheel. It doesn't offer the degree of subtlety that a wheel would, but it is faster and simpler to use and is in many ways just as effective.

SIDE-TO-SIDE WITH LFO TRIGGER



SIDE-TO-SIDE PUSH LEVER



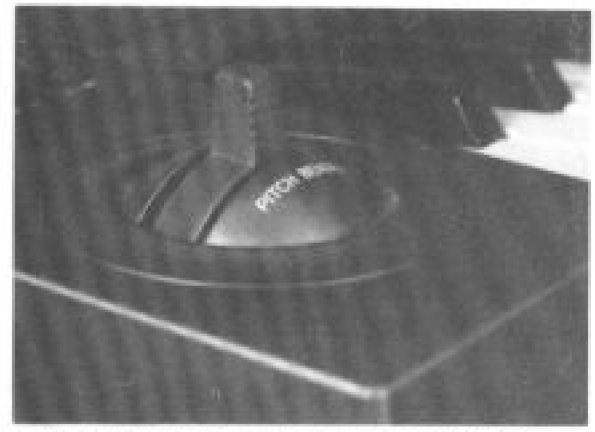
> **Exercise 21. The Pitch Bend Wheel** Practice bending a note up a whole step. Once you can do this with an isolated note, try it on a note that comes at the end of a three-note phrase. Try different intervals (if your system allows it). Try to hit the pitch dead on.

> **Exercise 22.** Practice bending a note down a whole step. Continue as in the above.

> **Exercise 23.** Practice adding a "slur" to a note. In other words, play the note and then quickly bend it a whole step (or half step) up and then back to its original pitch. Do the same thing again, but have the slur bend down and back this time.

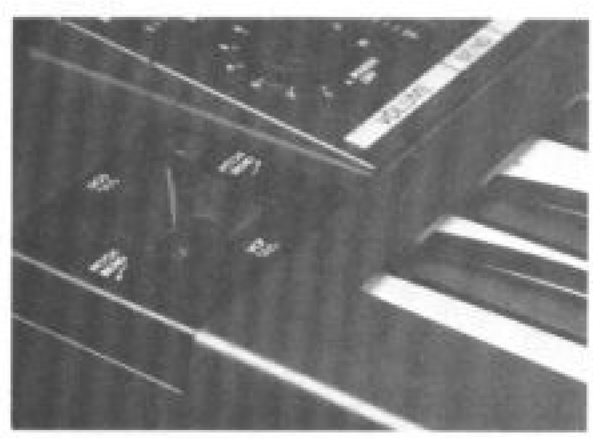
Exercise 24. Practice bending into a note from below. In other words, first pull back on the pitch bend wheel and then return it to center as you play the note. Do the same thing again except this time bend into the note from above. Get your timing right so that the bend is smooth.

One confusing aspect about pitch bend is that when you bend the pitch of a single note you also change the pitch of the entire keyboard. In other words, if you are in the key of C and you bend the last note of a phrase up a whole step, then until you return the **pitch bend** control to its original position all the notes in the key of C will sound in the key of D. What this means is that you'll have to give some thought to how you execute phrases that incorporate pitch bend. After a while you'll get the hang of it.


Side-to-side/push lever. *Roland JX-8P*

Exercise 25. The Modulation Wheel Start with a patch that contains no LFO modulation. Set your LFO so that the modulation wheel controls vibrato (LFO to VCO). If your system doesn't have an independent LFO for the modulation wheel, you will have to set the LFO as part of the patch and then turn the **amount** control to zero. Now the modulation wheel will introduce whatever LFO you set. (Remember, the modulation wheel is just another **amount** control.)

Wiggle those notes. Modulate a whole chord. Get down. Now, here's the deal: The modulation wheel will control *whatever you assign the LFO to.* In other words, if you route the LFO to the VCA, the mod wheel will control tremolo, if you route the LFO to the VCF, the mod wheel will control filter modulation, etc. Got that? So, fool around with your mod wheel and check out what it sounds like to introduce all those different types of modulation into a note or chord.


X,Y joystick. *Korg Poly 800*

Exercise 26. Combining the Two It takes some practice, but a nifty effect is to bend a note and then, in the middle of the bend, to apply some modulation using the mod wheel. You have to do both of these moves with the left hand. If you have wheels, you can use your palm and thumb or two fingers or whatever works for you. If you have a joystick or side-to-side lever with an LFO trigger, it won't be as hard; but like I said it takes practice to get it right. You want to be able to introduce a brief modulation right after the note has been bent. But, you don't want to leave the modulation going, nor do you want the note to return to its original pitch. The hard part is pulling the mod wheel back down again without moving the pitch bend wheel from its new position.

These are all basic moves. Once you are comfortable with them, go ahead and alter them to suit your style.

The Ribbon Controller

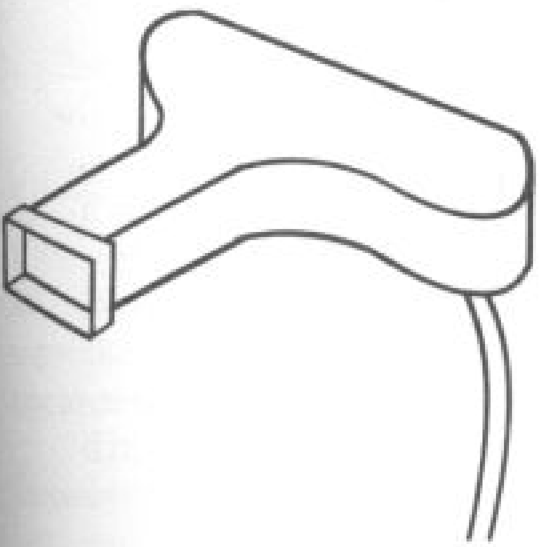
An unusual but popular performance control, used for pitch bend, is the *ribbon controller.* It consists of a thin metal strip, covered in either felt or plastic, that generates a voltage control that changes as you draw your finger across its length. Some people find it a little awkward as compared to a lever, but it still has its adherents. It is often found on hand-held remote keyboard controllers (like the Yamaha KX5) because the angle at which you need to hold them closely approximates the left-hand pitchbending grip of an electric guitar.

The Breath Controller

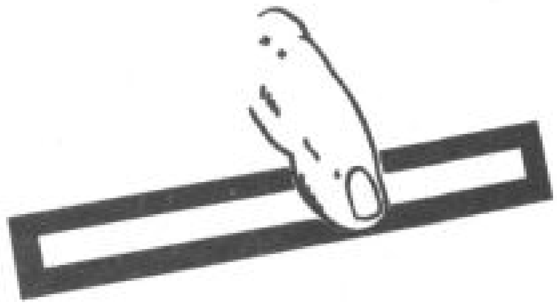
A *breath controller* is basically a pressure-sensing device that is held in your mouth between clenched teeth and responds to how hard you blow into it. It then transforms that pressure information into a control voltage that can be used to control any voltage-controlled parameter. Its unique features are (1) that you can play it without using your hands or feet and (2) that it enables you to more closely approximate the actual phrasing of wind instruments like the clarinet, the trumpet, and the saxophone. Because it can send out a voltage control that duplicates the contour of your blowing, it lends authenticity to a sound when controlling parameters such as the attack portion of the volume envelope or the frequency cutoff point of the filter.

RIBBON AND BREATH CONTROLLERS

BREATH CONTROLLER



RIBBON CONTROLLER

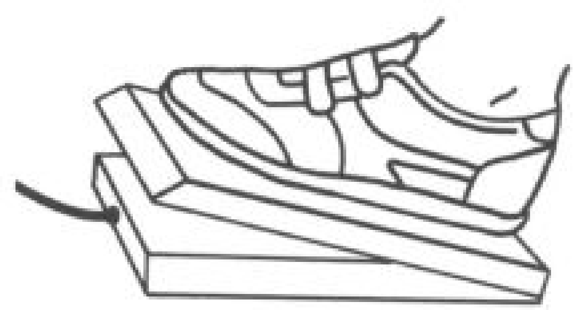


Footpedals and Footswitches

There are two kinds of performance controls that were designed for us to play with our two feet.

One, the footpedal, generates a changing control voltage just as do the lever, the ribbon, and the wheel. The other kind, the footswitch, is simply an on/off switch that can turn on or off whatever function it is assigned to.

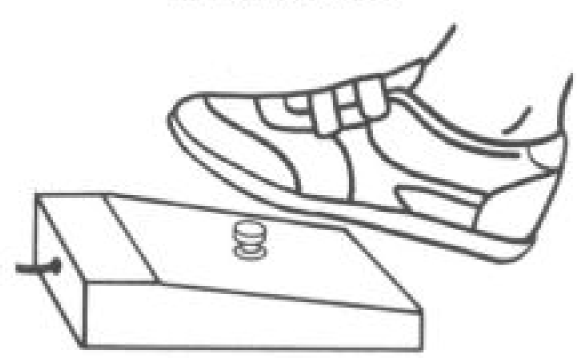
FOOTPEDAL



Any parameter that can be controlled by a changing voltage can be controlled by a footpedal. That includes volume (through the VCA), pitch bend (the VCO), and modulation (the LFO). Similarly, any parameter that can be controlled by an on/off switch is a candidate for being controlled by a footswitch.

Most synthesizers generally have anywhere from two to six different inputs for footpedals and/or footswitches. The parameters controlled by these inputs are usually "fixed." However, some of the more expensive synthesizers permit you to assign each footpedal or footswitch to a parameter of your choosing.

FOOTSWITCH



The most common fixed assignment for a footpedal is as a volume control. This is useful for fading in and out string parts and the like. The footpedal is also often used for adding vibrato effects. This is done exactly the way a modulation lever is used to control the amount of LFO modulating the VCO. It can also be used to control the filter cut-off point as well as the rate (speed) of the LFO. Some synthesizers let you use a footswitch to step through consecutive patches in a prearranged chain. This is a feature that enables you to go from song to song without having to search for the right patch.

Many synthesizers provide an input for a **sustain** pedal. **Sustain** pedals work differently from system to system but essentially they try to approximate the effect of a **sustain** pedal on an acoustic piano. A **sustain** pedal on a synthesizer is more of a switch than a pedal. It sends instructions to the volume envelope to either extend the sustain portion of the envelope or to extend the release portion of the envelope.

Most synthesizers offer some kind of input for a footswitch or footpedal, but the actual device itself (the footswitch or pedal) is generally sold separately. The result is that many beginning synthesists buy the synthesizer without a pedal and therefore miss out on a very important aspect of a synthesizer's performance capabilities. Buy the footswitch and/or footpedal.

Exercise 27. Footpedals Experiment with the footpedal as a **volume** control. Call up a string patch and practice fading notes in and out. This is harder than it sounds. You have to get used to playing the note before you actually hear it.

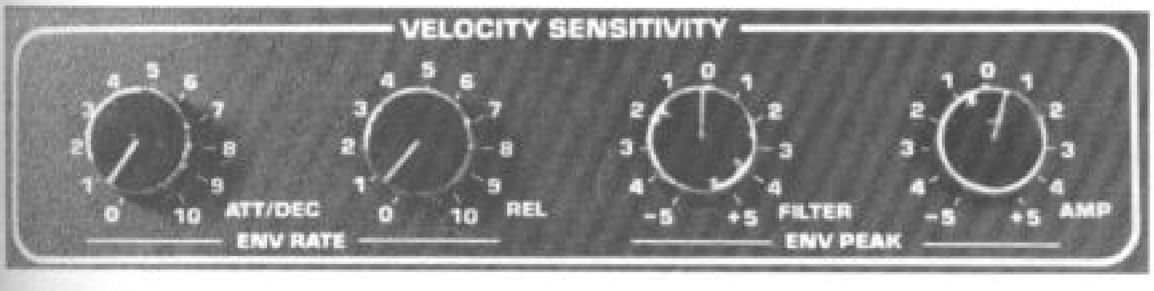
Use the footpedal as a modulation control. Remember there is absolutely no difference between the footpedal controlling modulation and the modulation wheel doing it—same principle, different appendage.

The Touch-Sensitive Keyboard

There is one more very important performance control that may not at first seem as if it falls within the same category as levers, wheels, and pedals. It is the touch-sensitive keyboard.

A touch-sensitive keyboard is one that has the ability to measure certain differences in the way you play the keys. It can tell whether you strike a key fast or slow. And it can respond to how hard you press down on the key after you have played a note. Some can also sense side-to-side movement of a key—the sort of motion one would use on a stringed instrument to achieve vibrato. The sensing devices in a touch-sensitive keyboard then translate these measurements into voltages or digital instructions (binary code) that in turn control the same sort of parameters that all other performance control devices control—namely, the VCO (pitch), the VCA (volume), the VCF (timbre), and the LFO (vibrato).

There are basically two kinds of touch sensitivity. They are velocity sensitivity and pressure sensitivity.



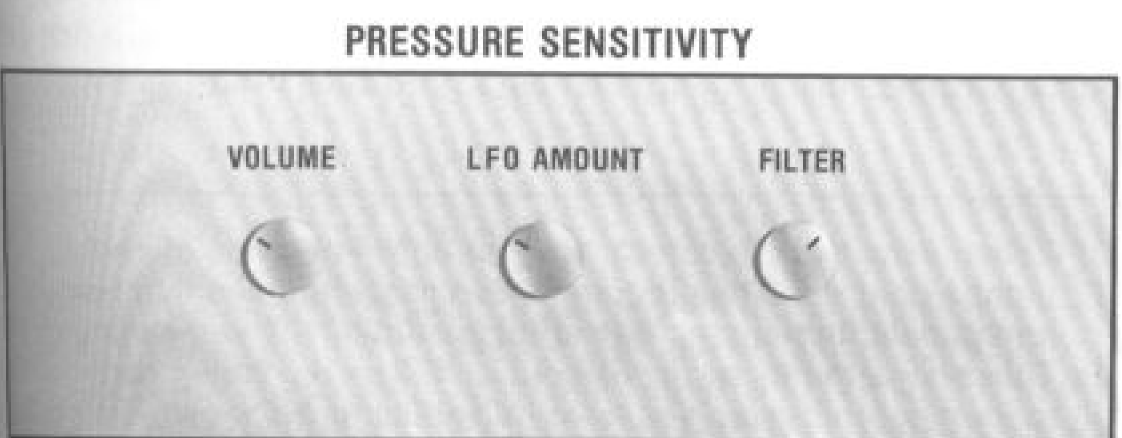
Velocity sensitivity. *Sequential Prophet T8*

Velocity Sensitivity

Velocity sensitivity measures the speed with which you strike a key. The effect will change according to which parameter is being controlled. For example, when this function is assigned to control the VCA, the faster you play, the louder the note.

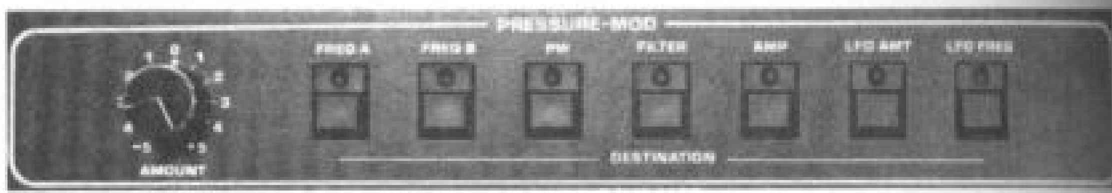
It is important to understand that velocity sensitivity measures velocity, not force or strength. There is a relationship between velocity and force. We naturally play harder when we try to accent a note on an acoustic piano for instance. But, on a synthesizer, what is actually being measured is speed, not strength.

Like a piano a velocity-sensitive keyboard on a synthesizer can control changes in volume. As I've already said, it does this by sending a message to the VCA. It can also control various other parameters. If the same velocity information is sent to the attack portion of the volume envelope, you can control the very beginning of a note's envelope—whether it has a short attack or a long attack—simply by changing the speed with which you depress the key. If you decide you want the velocity sensitivity to affect timbre, what you do is assign it to the filter cutoff point in the VCF. Then, the faster you strike a note, the brighter it will be.



Pressure Sensitivity

Pressure sensitivity, also called *aftertouch*, measures how far down you press a key after it has been struck. This can be useful in imparting some kind of modulation into the middle of



Pressure sensitivity (Aftertouch).
Sequential Prophet T8

a note (vibrato or tremolo) or as way to create a volume "swell." As with all these performance controls, the specific effect on the note is determined by the routing. If pressure sensitivity is assigned to the VCA, it will control volume. If it is assigned to the LFO, you get vibrato or tremolo, and so on.

Release Velocity

One other type of velocity sensitivity, which exists to date only on the Prophet T8, is called release velocity. *Release velocity* is similar to normal velocity sensitivity except that instead of measuring the speed with which you strike a key it measures the speed with which you release a key. This can be useful in controlling the release portion of a volume envelope. For instance, when playing a stringlike part, you can determine the length of the note's release by either lifting your fingers off the keys quickly or slowly.

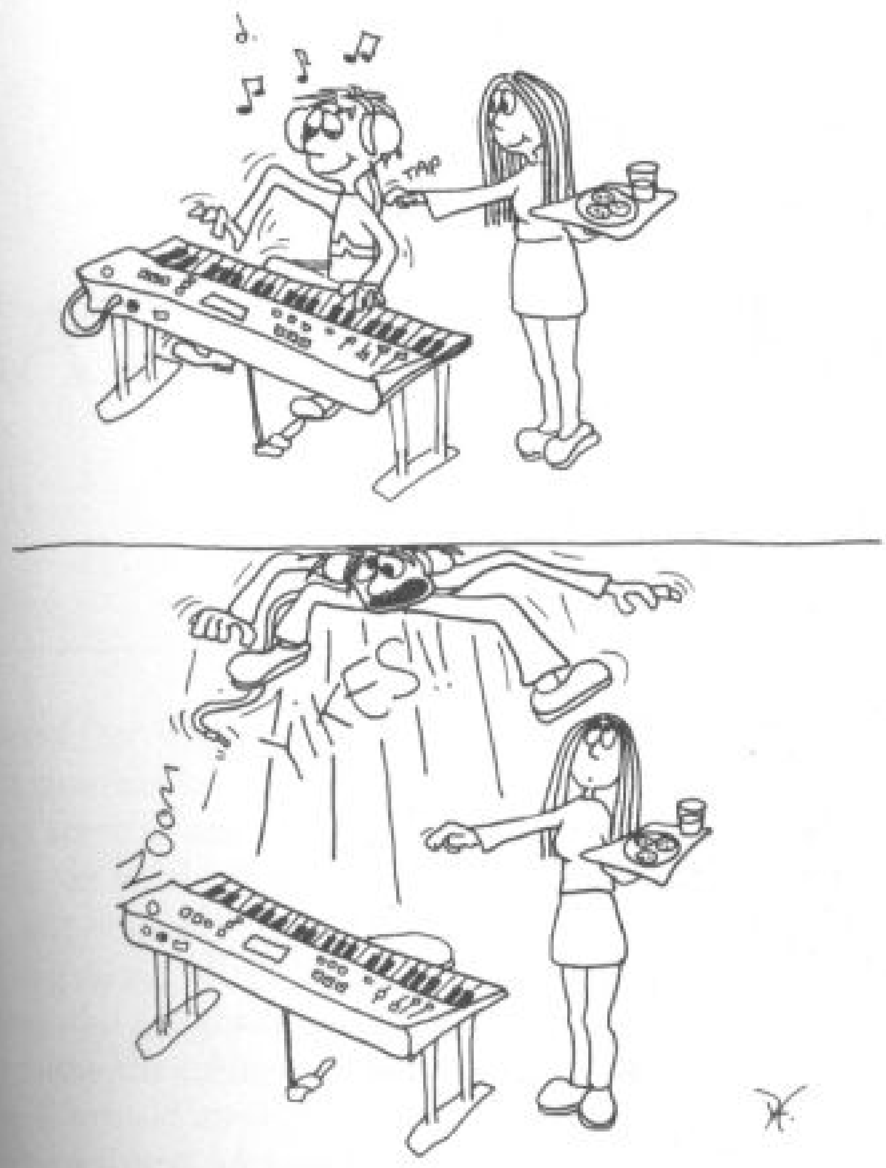
Side-to-Side Sensitivity

There is one more kind of touch sensitivity. It doesn't have a name but it is a feature involving a sensor that responds to a side-to-side or rocking motion of a finger on a key. It is similar to pressure sensitivity, but it measures movement left to right as opposed to up and down. It is normally used to control the VCA (volume) or the VCO (pitch) in order to give you the ability to create the same kinds of vibrato and tremolo one gets on a stringed instrument such as a guitar or violin.

> Touch sensitive keyboards are just different kinds of performance controls. They require different techniques than wheels and pedals, but again, they all do basically the same things. This brings me to an important point. The preceding performance control exercises were examples of the most common and basic moves. But, don't forget, anything's possible. Remember that almost every performance control has its equivalent on the main panel, i.e., the pitch bend wheel is just a handy **frequency** control, the modulation wheel is just another LFO **amount** control, and so forth.

The point I want to make is this: Under the right circumstances, almost every regular control knob or fader on the main panel of your synthesizer has the potential to be used as a performance control. In other words, just because it is not labeled **performance control** doesn't mean you can't use it as one. Use your imagination. Play a chord and listen to what happens to the timbre as you change different panel controls like the **filter cutoff, resonance, LFO speed,** etc. The synthesizer was designed to produce amazing and unusual sounds and if you can get a great effect by playing it upside down, or with your feet, or while spinning control knobs with your tongue, then do it. Just don't tell anybody I said it was O.K.

All of these performance controls simply offer different ways of being able to impart a little life into your synthesizer playing.



WOW! NOW, THAT'S WHAT I CALL 'TOUCH SENSITIVE'!

5

NEAT EXTRA FEATURES

BUT FIRST . . .

As promised, having already covered the different types of performance controls in the preceding chapter I will now talk to you about some of the various special features you are likely to run into on your typical or even not so typical synthesizer.

But first, it is time I explained a few things. Before we start talking about all of those neat extra features, I need to explain what *step programming* is. And before I can do that, I have to define the difference between analog and digital. Believe me, if I could avoid all this I would, but then you'd be hopelessly confused and probably ask for your money back. So, listen . . .

Analog versus Digital

The difference between analog and digital synthesizers is a source of great confusion for most beginning synthesists.

Ever since the word *digital* became a consumer buzzword, it has been a constant source of confusion to people. What does it actually mean? Unfortunately, it means something different to everybody. I will do my best to give you an all-around definition and then a definition as it applies to synthesis.

Digital refers to numbers, specifically the numbers in computer binary code. Every computer and microprocessor uses this binary code, which is made up of the numbers 0 and 1. Strings of these 1s and 0s represent the language of computers, and it is the elegant simplicity of being able to translate all kinds of information into numerical form that makes computers so powerful.

Digital clocks, digital calculators, digital stereos—all operate using the binary code. They each contain a *digital circuit* or *microprocessor* that serves to coordinate and perform their multiple tasks. These digital circuits and microprocessors are in fact miniature computers and it is their incredible speed as well as their ability to store enormous amounts of information and to execute complex instructions that is changing the world around us so dramatically. Actually it's pretty cosmic stuff. In any case, when you see the word digital all it means is that there's a computer in there.

The word *analog* is not as easy to define. It means *similar in some way*, but it also implies that while something is similar it is not exactly the same. Let me explain.

Before there were digital circuits, there were electronic circuits. These circuits were made up of electronic devices like capacitors and resistors. In their day they were considered pretty state-of-the-art, but by today's digital standards they actually seem slow and clumsy. I'm exaggerating the differences, of course, but it helps to illustrate the distinctions between digital and analog. Both types of circuitry use electricity, but digital technology employs it in a subtle and elegant way using strings of perfect numbers while analog technology uses it in an almost primitive fashion by forcing raw current through an obstacle course of transistors.

So, a digital device uses numbers, and an analog device operates by the direct application of current.

In reality, almost all electronic devices today use a combination of these two technologies. Therein lies the confusion.

For our purposes we can define an analog synthesizer as one that generates its sound source using analog technology—oscillators. By the same definition, a digital synthesizer would be one that generates its sound source using digital technology—digital tone generators.

According to those definitions, every voltage-controlled synthesizer is an analog device because it generates its sound source using analog technology. This is true even though every one of today's voltage-controlled synthesizers employs a certain amount of digital technology for almost every operation.

The only true digital synthesizers are instruments like the Yamaha DX7 and the Casio CZ101 and a few super-expensive systems such as the Synclavier and the Fairlight—all of which generate their sound source using some type of digital tone generator as opposed to a regular oscillator. This is true even though you don't actually hear that digitally generated sound source until it is converted into audible form through the use of a digital to analog converter. Got that? I'll say it again. All analog instruments employ some digital components, and all digital instruments employ some analog components.

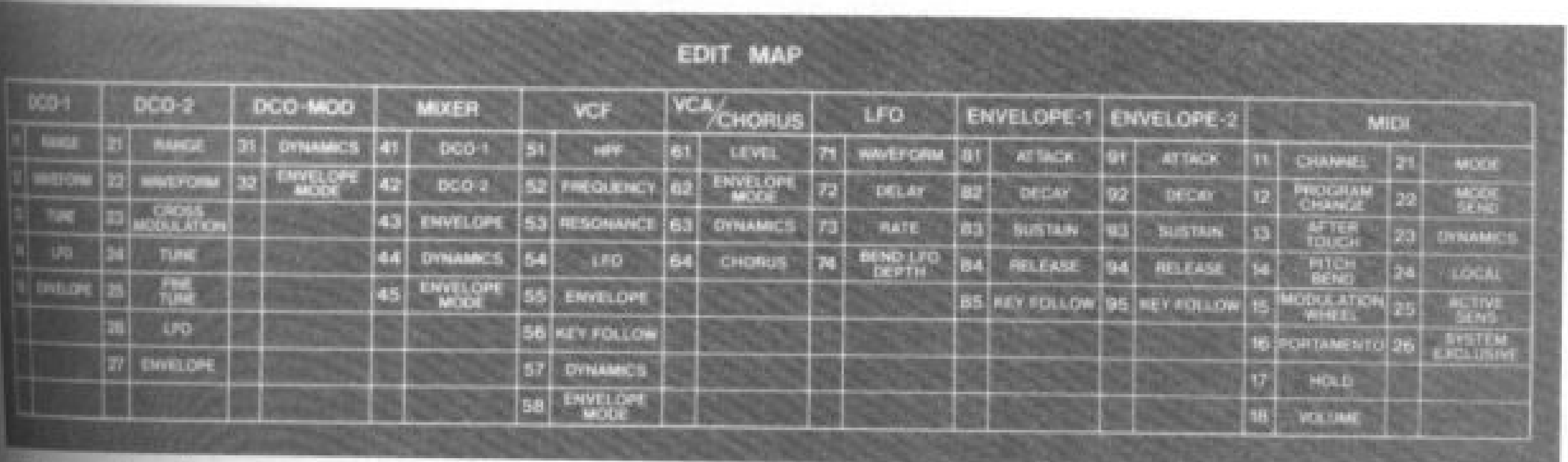
Now that hopefully I've cleared up this confusing issue, let me reiterate an important point that I made earlier on. That is, that even though we are continuing to discuss various aspects of the analog synthesizer, almost every one of these features exists or has some equivalent in a digital synthesizer. So, even if you never intend to own an analog synthesizer, you still need to learn all this stuff because although there are some important differences between analog and digital synthesizers a great many of the basic sound-creating features are exactly the same.

Now let me tell you about step programming.

Step Programming

Years ago all synthesizers were covered with switches, levers, and knobs. Lots of 'em. Each of these switches, levers, and knobs controlled some parameter on the synthesizer—whether it was to set the frequency of one of the oscillators or to raise or lower the volume. Along came digital technology.

One of the important features of digital technology is



Step-programming edit map. *Roland JX-8P* that it enables you to perform a great number of operations in a very small amount of space. Manufacturers were able to replace all those levers and knobs with a single all-purpose numeric keypad. By assigning a number to each programmable parameter (VCO tuning, filter cutoff point, ENVs, etc.), you can simply punch up (access the code number of) whatever parameter you want to edit and then, using the same numeric keypoard, punch in the appropriate values from a scale of 1–10 or 1–100 or 1,0 (on/off). This system makes it possible to reduce the number of switches, knobs, and levers to as few as three or four and still be able to program each and every parameter on a synthesizer.

Anyone that has ever programmed a digital alarm clock or watch is familiar with this process. In synthesis it is called *step* programming because it requires you to edit each parameter one at a time (step by step).

Like most technological advances, step programming has advantages and disadvantages. The manufacturers love it because it enables them to dispense with a lot of those expensive levers and knobs. It also means that they can cram more and more exciting features into less and less space (and weight). This brings the cost down and that's good for the consumer, but there's one problem. Step programming is a pain in the ass. Yes, I used a three-letter word, in this otherwise very serious and somber text (ha, ha), and for a very good reason.

True, step programming makes the instrument cheaper and it takes up less space, but having to program each parameter one at a time is an extremely time-consuming process, compared with levers and knobs which are much faster and easier to use. Also, with levers and knobs you can jump very quickly back and forth between two different parameters to hear how one affects the other—a crucial part of sound modeling. With step programming this takes forever.

Even more important than the question of speed is the problem of knowing where you are. With levers and knobs, once you have started editing and have activated all your controls, you can tell at a glance exactly what's going on—how much resonance there is, which waveshape is being used, and so on. With step programming this is not the case. In order to find out the settings for a particular patch, you need to access each parameter one at a time (step by step) before you can get a complete picture of how that sound was programmed. This is really confusing and makes it extremely difficult for even an experienced programmer to keep track of where he or she is while editing (programming) a sound.

I'm not trying to scare you; I'm just trying to make an important point. If you are just buying your first synthesizer, try to get something like the Roland 106 or anything that has good old levers and knobs. It will make learning a hell of a lot easier. I'm also hoping that if enough people complain about it the manufacturers will respond by providing us with an intelligent compromise. Of course microprocessors are good for us, but there needs to be some kind of balance. Synthesizers were meant to be twiddled and tweaked.

Now that that's out of the way . . . on to the features.

FINALLY . . . ALL THOSE NEAT EXTRA FEATURES

Now, remember that we started the entire section on the voltage-controlled synthesizer by discussing the six main components of a synthesizer—the oscillator (VCO), the amplifier (VCA), the filter (VCF), the two envelope generators (the filter ENV and the volume ENV), and the LFO. Almost every one of the following features involves the modulation, the manipulation, or some kind of control over one or more of those same basic components. In other words, all we are doing is finding more elaborate and intricate ways of fiddling around with the good old VCO, the VCF, the VCA, and the ENVs, etc. Got it? Good.

Transpose

This feature is pretty self-explanatory. It allows you to transpose the entire keyboard, usually within a range of plus or minus one octave. Some synthesizers offer fixed range settings, that is, you can choose between only a few predetermined settings, usually octaves or whole steps. In fact, some synthesizers include a separate **octave** switch in addition to a transpose switch. The **octave** switch is just a quick way of transposing up or down an octave in order to change registers. Many synthesizers have transpose functions that are variable, that is, they allow you to determine transposition by the interval of your choosing. The most common way of programming the transposition interval of your choice is by hitting the transpose switch and then pressing any key above the lowest C on the keyboard. Whatever the interval between the note you play and the low C will be the transposition interval.

Octave/Range

A specialized **transpose** switch, the **octave** switch, also known as the **range** switch, lets you change the register of your keyboard up or down an octave with the press of a button (or the turn of a knob). The settings on the **octave** switch often have standard organ stop markings—16′, 8′, 4′, etc.—which correspond to the actual pipe lengths of the original pipe organs. Every time you doubled the length of the pipe, you lowered the pitch one octave. Sometimes the octave switches are simply marked **up** and **down.**

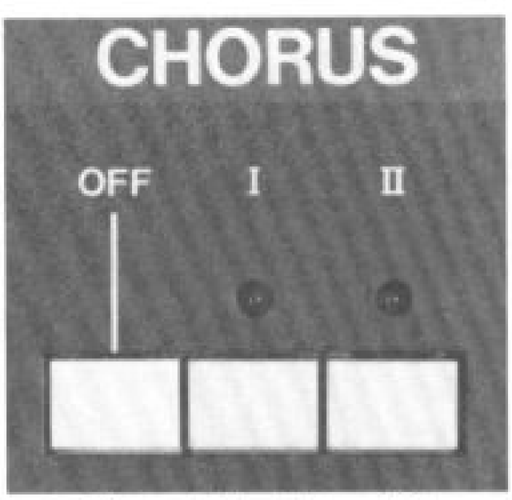
Master Tune

This is another tuning control that enables you to change the overall tuning of a two-oscillator system. It is a global tuning control that overrides the individual frequency (tuning) knobs on each oscillator. Master tuning is used to tune your instrument to concert pitch or to other instruments.

Auto-Tune

This is a switch that some systems, gratefully, include as one of their neat extra features. Analog synthesizers, especially older ones, are notorious for drifting gradually out of tune. The auto-tune function automatically recalibrates all the oscillators, putting the instrument back into tune with itself.

Chorus



Chorus. *Roland Juno 106*

This is an easy one. We already talked about the fact that most synthesizers have two oscillators (VCOs) per voice and that when you tune them slightly apart the result is a thickening or fattening of the sound. Some synthesizers have a feature that does exactly this at the flick of a switch. *Chorus,* also known as *ensemble,* is an extremely useful effect and can often turn a flat, boring sound into a beautiful and usable sound. One thing to be careful of is this: because chorusing involves the doubling of oscillators it very often will mean a halving in the number of available voices. For instance, if an eight-voice synthesizer gives you one oscillator per voice and you hit the **chorus** switch, you are likely to wind up with only four usable voices. This is because four of the original eight voices are being used to double the other four.

Some systems have two chorus settings. This simply means that one setting has the oscillators tuned farther apart than the other.

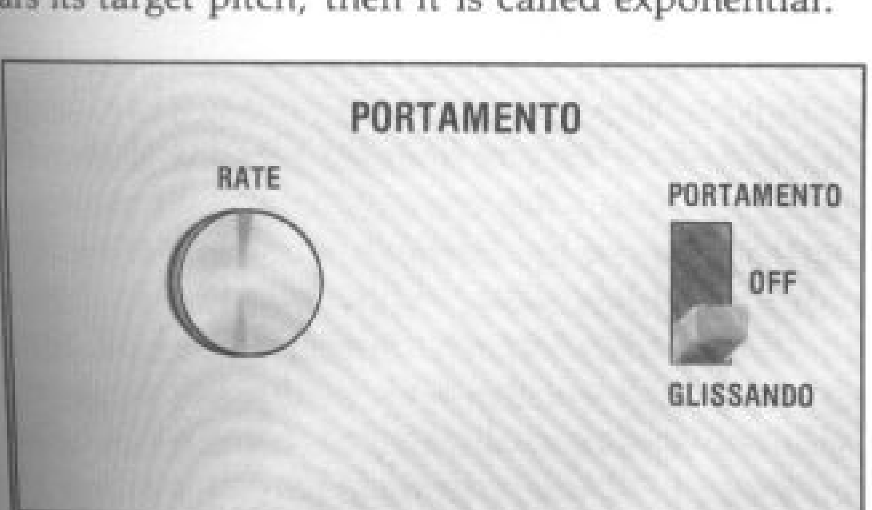
Portamento

Portamento also known as *glide* is one of the coolest effects on a synthesizer. It is a function that causes the pitch of a note being played to slide up or down from the pitch of the preceding note played. This is the effect that is responsible for those characteristic sliding synthesizer solos of the sixties and seventies (Yes, I know, that's a lot of *s*'s) that marked the beginning of the synthesizer era.

If at this point you understand all the things I've been saying about voltage control, then you should have no problem figuring out how this effect occurs. Remember how the keyboard sends a message to the oscillator telling it what pitch to produce? When the portamento function is engaged, the same thing occurs, only with a slight delay. In other words, the oscillator, instead of changing instantly from C to G, takes its time and moves gradually in pitch from the last note played (C) to the new note (G). This relatively gradual change in pitch is heard as a swooping pitch jump and that is portamento.

Most synthesizers offer a portamento **rate** control. This control lets you set the speed at which the portamento glide will occur. It can take anywhere from a fraction of a second to a full 10 seconds or more.

Some systems allow you to determine the slope of portamento, that is, whether the speed is constant or not. If the rate of portamento remains the same during the glide, the slope is referred to as linear. If the rate of portamento increases as it nears its target pitch, then it is called exponential.



When synthesizers were monophonic, that is, you could only play one note on them at a time, it was easy to predict from which direction the pitch slide would come—it would always come from the preceding note. Even on today's polyphonic instruments, as long as you play a monophonic line (only one note at a time, as opposed to chords) the same principle holds true. It is only when you start playing chords or more than one note at a time, with portamento on, that things start to get a little confusing. The reason for this is that, even though each voice (oscillator or pair of oscillators) slides up or down from its preceding pitch, the fact that you are dealing with several voices traveling in independent paths makes it almost impossible to predict the directions from which the slides (portamento) will occur. Not that it can't be done. It's just that it may require some thought and experimentation to be able to replicate a particular chordal effect using portamento.

Exercise 28. Portamento First find a patch with a lot of sustain. Experiment with different speeds of portamento. Remember, you need only a very small amount of it to hear the effect. (Warning: The portamento on some systems works only when you play legato, that is, when you play with no separation between the notes.) See what happens when you play chords with portamento. As long as everyone moves in the same direction, it's cool; but approach the chord from different directions and . . . watch out.

Glissando

Glissando is almost the same as portamento except that instead of producing a smooth continuous slide from one note to the next what results is a slide that is broken up into semitone steps.

Hold

This function enables you to play a note or chord and then sustain it for as long as you like. By pressing the hold switch once before you play the key(s), you cause the note(s) to sustain for as long as the **hold** switch is on. This is useful for creating drones or low sustained parts, and it frees your hands to play moving parts over whatever is being held. It also enables you to duck out the back for a quick sandwich during the set without anyone realizing you're gone. (Of course this would only be possible if you were playing a tune with very few key changes.)

Chord Memory

Chord memory, also called *chord latch,* is an interesting function that allows you to play chords with one finger. It does this by memorizing the intervals between the notes of whatever chord was held or captured by the **hold** function (see above). Then, using the pitch of whatever single key is pressed as the root, it re-creates the intervals of the held chord. So, for example, if you **hold** a chord consisting of the notes C, E, G and engaged the **chord memory** function, you could then press the note F with one finger and the chord F, A, C would sound. This is another finger- and hand-saving function that not only gives you the ability to play complicated and hard-to-reach chord progressions with a single digit but also makes it possible to pick your nose and play through changes at the same time. Technology is a wonderful thing.

Exercise 29. The *Chord Memory* Switch Even though I make fun of features like **hold** or **chord memory,** I don't want you to think that they are useless . . . they're not. However, they do serve a purpose only under very specific circumstances. These circumstances may be rare, which is why I poke fun at them. What I'm trying to say is this: don't wait for the circumstances, create them. You may not often encounter a piece of music that requires you to play a rapid, chromatically ascending series of major seventh chords, but so what? It's such a neat sound, and the fact that it would be impossible to realize in any other way makes it particularly unique. So, use it. Build a song around it. Use it as the basis for a riff. In other words, if you come across a sound or feature on your synthesizer that strikes you as odd or unusual, don't pass it by thinking it's not practical. Take advantage of it. Make it practical. Who knows? You're liable to come up with something really interesting. Then again . . .

Polarity Switch

This is a function that Roland made popular and that you will find on more and more synthesizers today. It is usually located near the envelope **amount** control. If you remember, an envelope generator gives a shape to the volume of a note as well as its timbre, and in some instances (when it controls the VCO) it affects its pitch. A *polarity* switch inverts the shape of an envelope. It literally turns it upside down. The result is that all your envelope controls have almost the opposite effect.

For example, take a typical filter envelope that is designed to gradually introduce the upper harmonics into a sound as it unfolds and then just as gradually reduce those upper harmonics until they completely disappear during the note's release. If you were to invert that filter envelope, this is what would happen: First, instead of the harmonics entering gradually, they would be there from the very beginning of the note. Then, instead of gradually increasing, they would gradually decrease, until they disappeared. However, what would happen during the release portion of the note's envelope is that they would suddenly reappear and begin to increase, at the same time that the volume envelope was reaching the end of its cycle.

This is an unusual effect—having harmonics re-enter as a note is ending—and would not be possible using a regular envelope. Only by inverting it can you take advantage of these peculiar upside-down properties. The same is true of the volume envelope and the pitch envelope. Being able to invert any of these envelopes gives you that many more options for designing a particular sound. It's like *Alice in Wonderland.* With the polarity switch everything is topsy-turvy.

Exercise 30. Inverted Envelopes If after all this time you have finally achieved some understanding of how the different portions of an envelope relate, this should serve to hopelessly confuse you all over again. No, I'm just kidding. It's not that hard to get. But it will require that you re-orient yourself. (No, that doesn't mean you have to go to Japan—although that is where most synthesizers are made today.)

A good idea is actually to draw a picture of a normal envelope on a piece of paper (or an envelope) and then turn the paper upside down. This will help you to keep track of the direction in which your envelope is going. You will then be able to predict, somewhat, the effect an inverted envelope will have on pitch, volume, or timbre, depending on which component you apply it to. This takes some getting used to, but the results are well worth the effort.

Try designing a patch whose harmonics disappear and then re-enter as in the above example by applying an inverted envelope to the VCF. Try designing a volume envelope that has a double attack.

Unison

When you hit the **Unison** switch on a polyphonic synthesizer, you assign all the oscillators to a single key. So instead of having an eight-voice system with two oscillators per key, you wind up with a one-voice system with sixteen oscillators per key. The result of this is that that single key is able to produce a very rich and powerful sound. This is useful for playing leads or simple one-note parts that require a certain amount of punch.

Single/Multiple Triggering

For normal operation most synthesizers function in a multitrigger mode. This simply means that every time a new note is played it triggers the volume envelope to begin its cycle. There are some situations where it is useful to be able to override this normal mode of operation. For this purpose some synthesizers include a feature that allows you to choose between single and multiple triggering.

When a synthesizer is in a single-trigger mode, it means that the volume envelope will only begin its cycle when a note is played without any other notes already being held down. In other words, if you were to play a note and then play a second note without first releasing the first note, the volume envelope would only trigger on the first note. The second note would be heard, but it would have no separate attack. It would be as if it shared the first note's envelope and started in the middle of the envelope cycle during the sustain portion.

Like the unison and portamento features, this effect is especially useful when playing solo lead lines. It enables you to add very discernible legato to your phrasing. *Legato* refers to a way of phrasing two notes so that one leads smoothly into the other as opposed to *staccatto* which means just the opposite—a way of phrasing two notes that exaggerates their separateness. In a single-trigger mode, all the notes in a phrase that are played legato will in fact share the same volume envelope. Only by playing without legato (lifting your finger between notes) can you retrigger the volume envelope.

Cross Modulation

By now you should be accustomed to the idea that almost every synthesizer component is capable of being modulated or controlled somehow by another component. A component such as an oscillator can even be controlled by another oscillator. LFO modulation, as you recall, is the result of a low-frequency oscillator modulating another oscillator. In cross modulation both oscillators fall within the audio range. The output of one oscillator is used to modulate a second oscillator. The result of this modulation is that new frequencies (harmonics) are created. These are called sidebands. Sidebands are also created when an LFO modulates a regular oscillator, but because an LFO is below the audible range the sidebands it creates are also inaudible. This is a primitive form of frequency modulation (FM), which we will discuss in greater detail later on.

Because the harmonics created by this process do not necessarily relate to each other as part of the natural harmonic series, the sound that results can often be metallic and clangorous. It is very useful for creating bell-like effects. This metallic quality is a characteristic shaped by the next feature—sync. In fact, the two are often confused because in spite of their differences they share one thing—they involve the combining of two oscillators.

Sync

Like cross modulation, sync involves the combining of two oscillators. The difference is in how they are combined. With *sync* the output of one oscillator (the slave) is locked in to the frequency of another oscillator (the master) in such a way as to repress the fundamental of the slave oscillator while causing it to accentuate certain harmonics according to where its own frequency is set.

What happens in synchronization is this: Every time the master oscillator begins a new cycle (generates a single wave), it forces the slave to do the same, regardless of where in its own cycle the slave happens to be. The result of this slightly sadomasochistic arrangement is that the slave's wave cycle is being continually abbreviated. It is this abbreviated wave that is responsible for the unusual harmonic properties of sync. Sync adds a metallic or bell-like quality to a sound. One interesting effect, using sync, is to modulate the pitch of the slave oscillator. This will yield an unusual harmonic sweep.

One word of caution: Some manufacturers still refer to sync as a type of cross modulation. They are, however, two different things. The point to remember is that they both produce a kind of metallic or ringing sound.

Exercise 31. Cross Modulation and Sync Both of these features involve the combining of two oscillators. Turn on one of the features and play a note over and over again. As you play the note, change the frequency of one of the oscillators. Listen to what happens to the note as you change first one oscillator and then the other. Find a sound with interesting upper harmonics and try to design a bell-like patch. Hint: You'll want a hard, short attack, so set it at zero. Don't use sustain. Let the decay take the place of release.

Noise

If you take a color wheel made up of the primary colors red, yellow, and blue and spin it quickly, the colors will start to blend and blur until all you see is the color white. White is defined as comprising all colors.

There are some sounds in nature—such as the wind, the ocean, and thunder, as well as some percussive sounds like drums and handclaps—that have no discernible pitch. In fact, like the color white, these sounds are made up of "all" frequencies spread equally across the audio spectrum. The best way we have to approximate these sounds is by generating a random signal resulting in a hissing staticlike sound that, like the sounds it is attempting to mimic, is made up of all frequencies in equal proportions. We call this *white noise*. White noise has a wide range of uses in synthesis, from generating frightening sound effects to imitating the properties of various percussion instruments. A little touch of white noise properly applied can also add a very authentic attack to a flute sound. You may also run into *pink noise* now and then (as in elephant). Pink noise is the same as white noise except with an emphasis on the lower frequencies, i.e., pink noise is not as bright as white noise.

White noise is an entirely independent sound source. It is actually a specialized oscillator, which is why it is generally found near or around the regular oscillators. However, a main difference between white noise and the regular waveform-generating oscillators is that it usually offers few if any selectable parameters. In other words, on most systems you can either turn it on or off, and that's it. Even so, it is still an invaluable sound source and fortunately today's newer synths are starting to offer a little more control over it.

Exercise 32. Noise Make an ocean. Use the LFO and a slow attack and release. Make thunder. Some resonance will help. Make wind. Brave the elements. (Don't forget your umbrella.)

Try creating a snare drum with a falling pitch. To do this, you want to combine your noise source with another sound source, either the main oscillator or, as an alternative, an oscillating filter. You'll need to apply an envelope to the VCO, or you'll have to use the filter envelope to control an oscillating filter in order to create a drop in pitch. Then you have to hope that your synth has some way of balancing the level between these two different sound sources. If not, at least you tried.

DESIGNING PATCHES AND PERFORMANCE TECHNIQUE

Here are just a few more exercises that should help you to incorporate some of the different things you've just learned. But first I want to make an important point. Sometimes we use synthesizers to mimic real instruments. (This is not to imply that synthesizers aren't real instruments. Of course they are. But let's leave the subject of reality aside for the moment.) What I'm trying to say is that even if you are able to design a patch that almost exactly duplicates the timbral qualities of a natural instrument, like a flute or a violin, that is still only half the battle.

Just as important as being able to duplicate an instrument's sound is being able to duplicate the performance characteristics of that instrument. For instance, two typical performance characteristics of an electric guitar are slides and bends. If you tried those same slides and bends on a violin patch, it wouldn't sound like a violin at all. Another factor is voicing. Brass instruments tend to be voiced more closely together than strings. How you voice a particular patch will go a long way toward implying what instrument you are trying to imitate. Finally, although there are always exceptions, the part itself will have a lot to do with whether a synthesized patch is convincing. For instance, you wouldn't play flute lines with a bass patch.

Some of this may be obvious, but it is worth mentioning. These subtleties can make all the difference, and it pays to keep them in mind whenever you are designing patches. Get out of the mindset of being a keyboard player. You're not. You are playing whatever instrument or sound you happen to have up at that moment.

Exercise 33. Design a Flute Patch Start with a very hollow-sounding waveform (square or triangle). Even if your system has two oscillators per voice, you may only want to use one. Set the frequency controls so that you are in a flutelike register (high). The volume envelope's attack should be short, but not too short. If you can control it, add a touch of noise to the very beginning of the sound. Add a slight amount of LFO applied to the VCO. Add delay to the LFO so that it is introduced gradually. If your instrument has a separate pitch envelope, try adding a very slight pitch bend at the very beginning of the note.

Exercise 34. Design an Electric Guitar This may take a lot of trial and error, but let's give it a shot. First, you know there is going to be a lot of high-end distortion and feedback, so to start off with choose a bright and buzzy waveform (a sawtooth wave should do it). Next open the filter all the way (you can back it off later, but you want to start out with a lot of rich harmonics). How about some resonance? (Not too much, or it may have the effect of rounding off some of the upper harmonics.) How about the volume envelope? You know you want a hard attack, and you'll also want to exaggerate the attack by making the sustain lower than the peak level (but not too much lower). A slight delay will do. And a small amount of release. Adjust the attack, decay, and sustain portions of the filter envelope so that it exaggerates the hard attack.

Now comes the tricky part. We want to put some more internal motion into the sound, and there are a couple of ways to do it. One would be to use a slow setting on the LFO and apply it to the filter. Another would be to sneak a slowly modulated pulse wave in there. We could even try to tune the oscillators apart just enough so as to create an exaggerated chorusing effect. Take your pick or maybe use them all. We should be pretty close at this point. Fine tune the overall sound with the **filter cutoff** control, the resonance, and the envelope amount.

Now we get to what I was talking about before, regarding performance characteristics. The only control on the synthesizer that can really get this patch to sound like an electric guitar is the pitch bend wheel. Practice bends. Try to imitate the kinds of moves that are part and parcel of an electric guitar solo. Remember, you are no longer playing keyboards. You are now a lead guitarist so you'd better act like one. Try mimicking a long slide down the fretboard. That's the idea. Rock and Roll. Now, just watch out for unscrupulous record company executives.

Exercise 35. Design a Spaceship Choose your galaxy and planet. I don't care. Create a patch that evokes the sound of a UFO hurtling at light speed through the cosmos, or maybe lifting off over a cornfield in the middle of Kansas. (Yes, I know that there is no sound in the vacuum of outer space. What do you think I am, stupid or something?) What I want to impress upon you is not to take weird sound effects for granted. Here are some suggestions, but seriously, do whatever you want. The idea is for you to become familiar with the outer reaches of your synthesizer.

Personally, I like a spaceship that rises in pitch as it disappears into the void. So, design a pitch envelope; and if you don't have a pitch envelope, try oscillating the filter with resonance and using the filter envelope to control pitch (we did this in Exercise 3). Another way of doing this would be to set up a gradual portamento. That way every time you hit a new note, the pitch would slide into it (see, there are many ways to approach a problem).

Next, why don't you try some LFO applied to the VCO and the VCF. In this case I would use the LFO's square wave if it has one, because it's so much more dramatic than a regular old sine or triangle wave. (Of course, that's just my opinion, and this is your spaceship.) A long release would probably be appropriate. Maybe a little noise, if you have it. You might want to fool around with the **sync** control or **cross modulation**. Got the idea? Enjoy yourself. Watch out for asteroids. And don't forget, never trust a Klingon!

6

ADDITIONAL FEATURES AND CAPABILITIES

O.K., so now we've covered most of the features that have to do with creating different types of sound. Next we're going to talk a little about different keyboard modes—the split keyboard, for example. From there we'll talk about two automatic playback features, the arpeggiator and the sequencer. Then we'll have a discussion about *data management*—that is, different systems for storing and manipulating sounds and sequences. After that we'll run through the typical inputs and outputs you are likely to find on the back of your synthesizer.

Get this stuff mastered, and you're on your way!

Keyboard Modes

There are two main keyboard modes: the split mode and the double mode. Each of these modes enables you to play two completely different sounds (patches) on the keyboard at the same time.

Split Mode

In *split mode* one patch is assigned to the left half of the keyboard and the other patch is assigned to the right. On some systems the split point is fixed somewhere around middle C. On other systems the split point is variable and is set much the way a transposition is set—by hitting the feature switch (**split keyboard**) and then pressing a key at the point you wish the split point to be. In a split mode, it is then possible to play a bass part with your left hand and at the same time a completely different-sounding melody line with your right. It's like having two synthesizers in one.

Double Mode

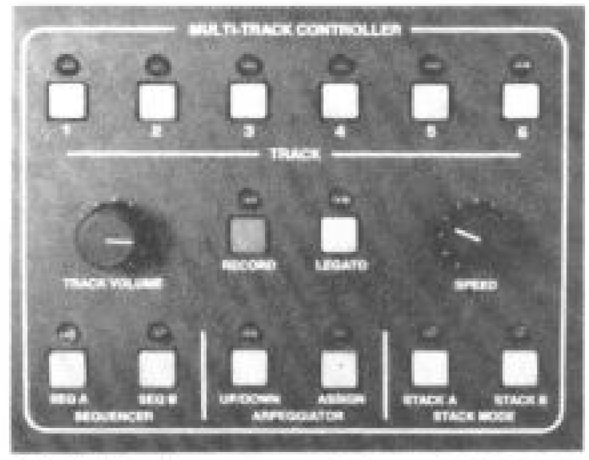
In *double mode* instead of being able to play two different patches on two separate parts of the keyboard, you are able to play two entirely different patches on the same key. Double mode, sometimes called *layered*, makes both patches playable simultaneously over the entire keyboard. By being able to combine two completely individual patches and trigger them off the same key, it is possible to create some very rich and complex sounds. Most synthesizers give you the capability for balancing the relative volume levels between the two patches to enable you to achieve just the right blend. Again, this amounts to being able to play two synthesizers at the same time.

One thing to keep in mind though is that, as with the chorus feature, doubling uses up twice the number of oscillators available to you. This, again, leaves you with half the number of polyphonic voices so that, for example, if you had an eight-voice polyphonic and switched it over to double mode you would then only be able to play four notes at a time.

The Multitimbral Synthesizer

While many polyphonic synthesizers offer a double mode feature, there are several models that offer you the ability to play as many as six or more different patches on the keyboard at the same time. These are referred to as multitimbral instruments. I am referring in particular to the Sequential Multi-Trak series. These instruments have a *stack* mode that enables you to play all six voices, each with a different patch, at the same time. Their limitation is that you can only play monophonically, that is, one note at a time. There are also some very expensive systems that offer multitimbral features of up to 32 voices and still remain polyphonic. These systems—like the Synclavier, the Fairlight, and the Kurzweil—also offer multiple split points. They give you the ability to assign as many as 64 different patches across the length of the keyboard. Incredible but true. More about these "super systems" later.

The following two devices are found on many of today's synthesizers. They have nothing to do with synthesizing sound per se. They are triggering devices that play notes automatically.

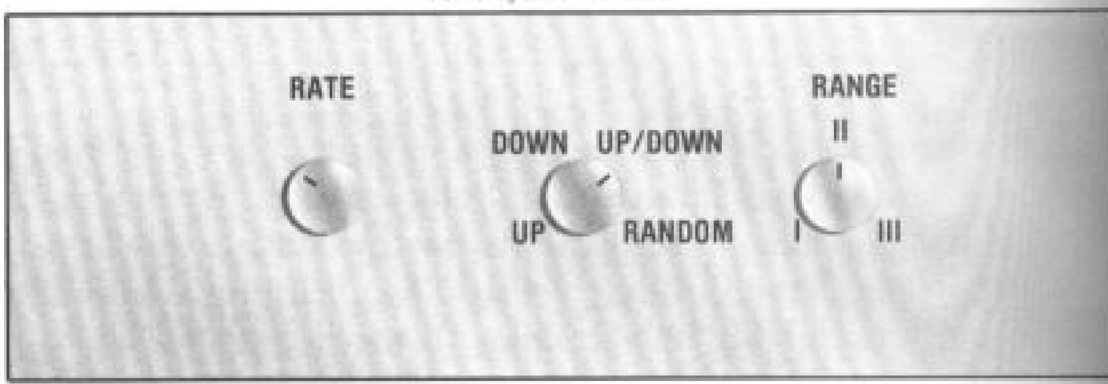


Multitimbral sequencer. *Sequential Six-Trak*

The Arpeggiator

An arpeggiator is analogous to the automatic machine guns you see on *Miami Vice*. It is a device on a synthesizer that automatically retriggers any note or group of notes for as long as you hold down the keys. The notes repeat in sequence (one after the other) and at a steady rate (in even beats). It has a rate control, which enables you to vary the speed of the repeats. Some systems allow you to choose the direction of the arpeggiation. For instance, if you were to hold down the triad C, E, G, you could choose to have it arpeggiate up (C-E-G) or down (G-E-C). Some systems offer a random arpeggiator so that the triad C, E, G might arpeggiate C-G-E-C-E-G, etc. Some systems offer a priority system. That means that the notes will arpeggiate in the order in which they were played, so that if you played the notes E then C and then G and held them they would arpeggiate in that order.

Many arpeggiators include a **range** switch. This function will automatically arpeggiate whatever notes are held, plus



the same notes up or down one or two octaves. In other words, you can be holding down three notes and the arpeggiator will play back as many as nine.

Some systems allow you to use the arpeggiator along with the **hold** function. This enables you to set up a sustained arpeggiation without actually having to hold down the notes. An arpeggiator is a wonderful performing and compositional tool. It permits you to develop parts that in terms of rhythmic accuracy and speed would otherwise be difficult if not impossible to play.

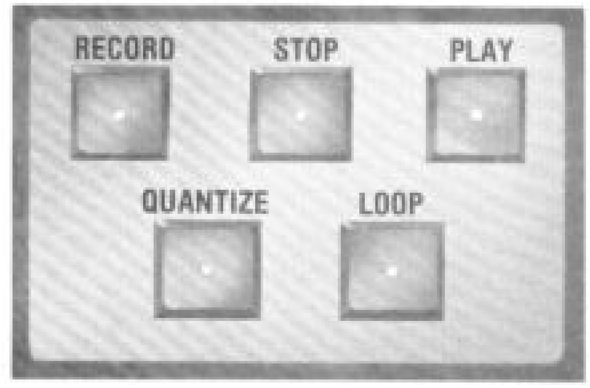
Exercise 36. The Arpeggiator Turn on the arpeggiator and play a single note (to start off with it is a good idea to use a patch with a short attack). Vary the **speed** control. Then play two notes at once. Then, a full chord.

Experiment with all the features of your arpeggiator using different chords with different voicings. Change the direction of the arpeggiation. Change the **range** setting. If your arpeggiator comes with a **hold** switch (sometimes called latch), turn it on. If your synthesizer has a **chord memory** switch, see if that will work with the arpeggiator (some do). Without making yourself dizzy, see how many different functions you can get to operate at the same time—hold, chord memory, portamento, etc.— that will also have some effect on the arpeggiator. Listen to how different patches work differently when they are arpeggiated. This is more than just a fun toy (although it is definitely a fun toy). It can also be a very useful tool once you become familiar with its full potential.

The Sequencer

A sequencer is like a tape recorder . . . without the tape. You play a piece on the synthesizer keyboard; and when you press **play** on the sequencer, it will play back exactly what you performed. The main difference is that instead of recording the actual sound of a performance it records only the key depressions. A good analogy would be the piano roll on a player piano. The piano roll, like a sequencer, doesn't record the actual sound of a piano piece, only the key depressions, yet it is able to reproduce the performance of the piano piece. That is how a sequencer operates.

SEQUENCER



Replacing Patches

One advantage of this mode of operation is that if you record a piece into a sequencer and you are happy with the performance but you don't like the sound of the patch you used you can simply replace the patch. The performance of the notes will stay the same, only their sound will change. For example, say you played a melody with a flutelike patch and you decide that you'd rather hear a trumpetlike sound. You don't have to play the part all over again. All you do is change the patch. This would be impossible to do on a tape recorder.

Changing Key and Tempo

Another advantage of sequencing is that if you have recorded a piece but find that it is in a difficult key you can transpose the key without changing the tempo as you'd have to on a conventional tape recorder. Conversely, with a sequencer you can change the tempo of a piece without changing the key. Both these features are made possible by the fact that the sequencer is recording key depressions, not sounds.

Time-Correction/Quantizing

One of the most important features of a sequencer is that many of them allow you to "correct" your time. This is also known as *quantizing* or *justification.* This feature enables you to play a part into the sequencer and, no matter how sloppily or out of time you played it, when it is played back it will be in perfect time. The sequencer actually moves the notes to the nearest correct beat. This feature alone has enabled many a keyboard player with a vague sense of time to earn a living in the music business.

Looping

One feature you are liable to find on any synthesizer with a sequencer is *looping.* Looping, as its name implies, is a function that automatically starts a sequence over again from the beginning after it has reached its end. It will repeat, or loop, over *and over* for as long as the looping feature is engaged. This can be great for setting up a repeating pattern and then improvising over it—an interesting way to write or just as a way of accompanying yourself.

Sequence Chaining

If a synthesizer has the ability to store several sequences in memory, you may encounter a feature called *sequence chaining*. This feature will automatically play one sequence after the other in a predetermined order resulting in one very long sequence made up of several different parts. Again, this type of feature has many applications compositionally.

Sequencers are normally sold as completely separate components from synthesizers, but a large number of synthesizers have incorporated a sequencer as another one of their regular features. We call these *on-board sequencers*, and those are the kind that we are discussing here. In general, these on-board sequencers are not as powerful as their independent counterparts, but they perform incredible feats nonetheless.

Some sequencers are capable of recording not only keyboard depressions but some performance control parameters as well, like velocity and pressure sensitivity, and pitch bend and modulation wheel moves.

All of these different features give you an enormous amount of flexibility as far as recording and then manipulating material.

Real-Time/By Step

There are two ways of programming, or loading, a sequencer: in *real-time* and *by step*. Programming in real-time involves playing your piece live into the sequencer, just as you would on a tape recorder. Programming by step means that you have to feed your piece into it note by note. The use of the term *step programming* as it relates to sequencers is almost identical to the step programming we discussed earlier. One refers to loading a sequence by step (note by note) and the other refers to editing or programming any synthesizer parameter by step (parameter by parameter).

One important consideration, if you have a synthesizer with an on-board sequencer, is whether or not it will sync up to external devices such as drum machines or other sequencers. Some sequencers have the ability to control, or "drive" external devices; some can be driven by external devices, and some are capable of doing both. This ability to sync up with an external device makes it possible for you to incorporate your synthesizer into an expanded network of instruments all working simultaneously as if they were a single instrument. We will discuss synchronization further when we talk about the wonderful world of MIDI.

Aside from all the impressive features described above, the most significant contribution of the sequencer is that it has had a dramatic influence on the process of composition as well as recording. In so doing, it has changed not only the way we make music but the kind of music we make. The randomly sequenced computerlike theme music for the 6:00 evening news is only one typical example of how technology has influenced style in a major way. Even if you have an onboard sequencer with very limited sequencing abilities, it can still be an extremely useful tool. At the very least, they can serve as invaluable electronic scratch pads for generating and then saving your musical sketches and ideas.

Exercise 37. The Sequencer If your synthesizer comes with an on-board sequencer . . . use it. I'm not just being funny when I say that. A lot of people tend to forget that a sequencer is much more than just a digital tape recorder. You can speed it up without changing the key. And you can change the key without speeding it up. Become familiar with these features and they will influence the kinds of pieces you write and the sorts of parts that you come up with. Try devising a riff (a series of notes) that outlines the changes (the chords) of a simple modal progression, in other words, something without too many chord changes—for instance, four measures of C, followed by four measures of B flat. Play this into the sequencer. Now, speed it up.

You can see how by simply speeding up the right selection of notes you can create very modern musical textures. This is similar to what you did before with the arpeggiator. This is because an arpeggiator can be thought of as a sort of primitive sequencer. A sequencer does more, but the point is that both of these devices play whatever notes you tell them to. Look, Ma, no hands!

Data Management and Patches

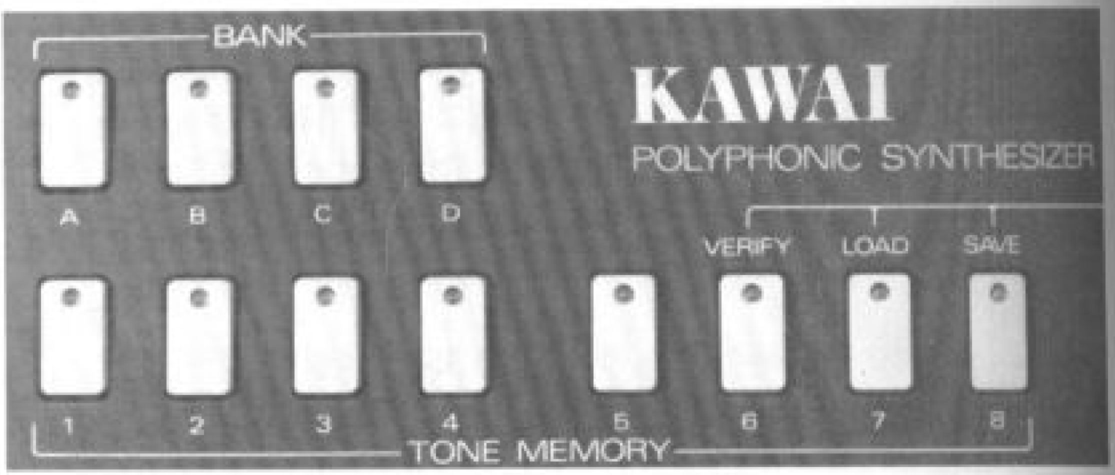
The Patch

Once you have designed a sound that you like after careful programming and editing (or by accident—which happens just as often), where do you put it?

All synthesizers have storage space for saving patches. On average, they allow you to store anywhere from 30 to 100 different patches in memory.

What is actually stored in a patch? Almost every programmable parameter is stored in digital form in memory as part of a patch, including oscillator frequency, filter and envelope settings, LFO rate, etc. The only exceptions might be more global (overall) keyboard settings such as transpositions, keyboard splitpoints, or some performance control settings such as pitch bend and velocity sensitivity. Some synthesizers store even this global information along with each patch. Others store much of this global information separately. When stored separately, this global information, which generally has more to do with performance than with the sound of the patch itself, is referred to as a *keyboard setup*.

There's really no point in my making up exercises for all of these data management routines. They are different for every instrument, so you'll just have to figure it out for yourself. I will stress the importance, though, of becoming familiar with how to store and save patches. Nothing's more frustrating than coming up with a great sound and then losing it because you weren't sure of the correct method for saving it.



Patch banks. *Kawai SX-210*

Banks

Very often, the storage space for patches is divided up into subdivisions called *banks*. Instead of having 40 switches for 40 patches, you might have 10 patch switches and 4 banks switches– in other words, 4 sets of 10. To call up a patch, you simply choose a bank and then a patch within that bank. On the other hand, if your synthesizer is step programmable, you just punch in the desired patch number on a numeric keypad. In any case, the process for calling up a patch is usually pretty self-evident. There all also set procedures for storing (saving) a patch, for erasing a patch, and for swapping patches (when two patches exchange places). These procedures vary from system to system, but they are normally well labeled and easy to figure out. Although in order to avoid erasing a good patch by accident, it is always a good idea to consult the manual.

Activating the Controls

When you call up a patch, you can begin editing it immediately. But on any system that is not step programmable, that is, any system that uses good old switches and knobs, there is one important thing to remember. Until you actually move them from their current positions, the controls on the panel will not accurately reflect the settings of the new patch. Only on/off switches with lights to indicate their status reflect the actual setting of the new patch. Not until you actually move each lever or knob in the process of editing will it reflect its actual setting. By moving a lever or knob, you activate it. Only once it has been activated is it a true gauge of its function.

This can be confusing at first. Even experienced synthesists will sometimes find themselves staring in bewilderment at a control setting that isn't doing what it says it's doing . . . until finally it dawns on them that the control hasn't been activated. So, don't feel too stupid when it happens to you. You'll soon get the hang of it. The other important thing to remember is that unless you save your newly edited patch (store it in memory) you will lose it as soon as you call up another patch.

Splits and Doubles

Systems that offer double and split modes also have set procedures for assigning pairs of patches. Again, these procedures may vary slightly from system to system, but the principles are the same. The thing to remember about splits and doubles is that they don't store the settings of the two patches. They only store the locations (the addresses) where the two patches were stored originally. This means that if you change either of the original patches you will wind up changing the doubles and splits as well.

Presets

Many synthesizers come with presets. A *preset* is a factory patch that is stored permanently in one location. It is possible to edit a preset and then store the edited version in a new patch location, but in most cases the original version will remain unaltered at its original location.

Edit/Compare

Some systems offer a feature called *edit/compare*. This is a feature that enables you to shift back and forth between an original patch and the currently-being-edited version of that patch without losing the version that is being edited. Normally if you switched back to the original version of a patch while you were in the middle of editing it, without having saved it first, you would lose the newly edited version. Edit/compare stores the edited patch temporarily, allowing you to hear how the newly edited version compares to the original.

Patch Shifting

Patch shifting, also called chain or step, is a feature that allows you to step through a series of from 10–50 different patches in a predetermined order by simply hitting a single switch repeatedly, usually a footswitch. This enables a performer onstage to quickly change from one sound to another without having to start searching through the patch banks for the correct patch. It also keeps his or her hands free.

Cassette Storage

Most synthesizers today have some capacity for storing all of the patches in their memories onto cassette tape. Being able to load all your patches onto cassette makes room in your synthesizer's memory for new patches and also insures that in case of accidental memory wipeout due to a power surge, or drugs, you will always have a backup of your favorite patches. The procedure for saving data onto cassette is described at the back of your instrument's manual. It involves hooking up a cassette player to an input and output in the back of your synthesizer.

It can be a little tedious at first, but it is worth getting into the habit of doing. Nothing is more frustrating than accidentally wiping your memory clean and not having a backup. Weeks of programming can vanish in an instant. In general, this only has to happen to you once and suddenly you'll find that you are very conscientious about storing data. The best thing about cassette storage, though, is that it effectively expands your synthesizer's memory to infinity (or as many cassettes as you can afford to buy at Radio Shack).

Cartridge and Disk Storage

Recently, more and more systems have begun offering cartridge and disk storage as alternates to cassette storage. The advantage to having a cartridge is that they are generally a lot sturdier than cassettes and they are also faster and easier to use. The advantages of disk storage are that they are also faster than cassettes as well as being generally more reliable.

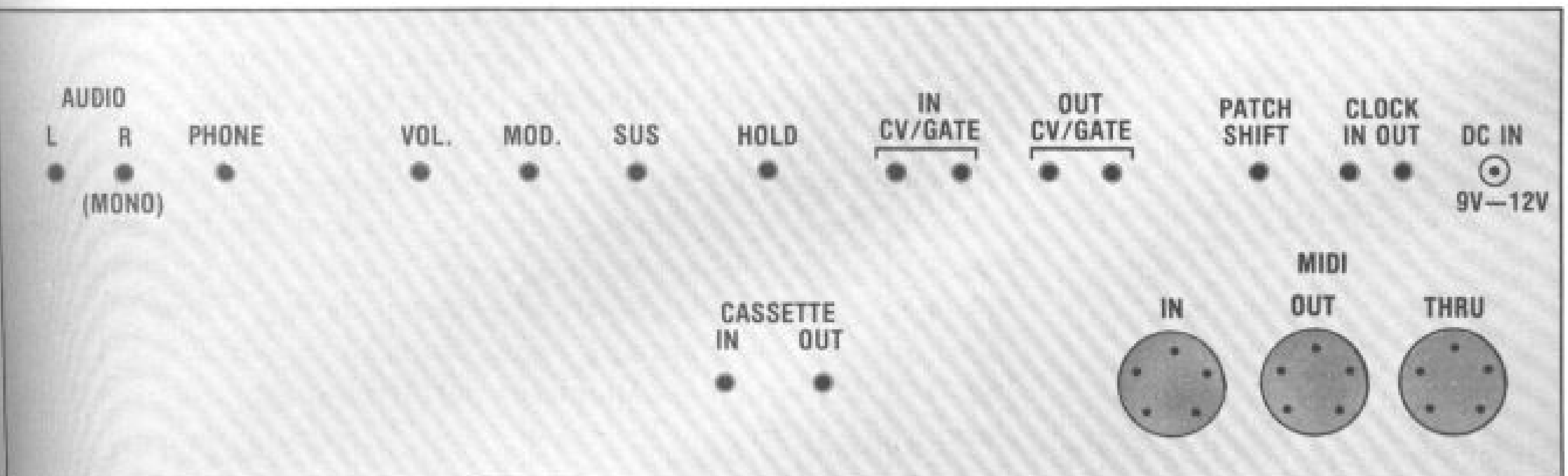
Sequence Storage

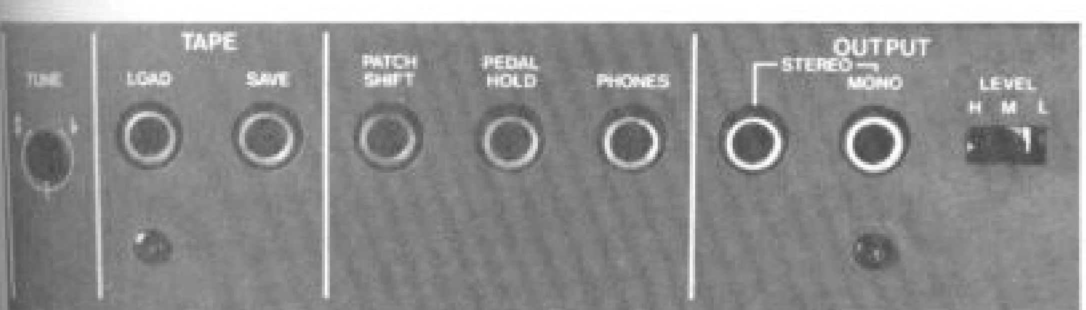
Most of the rules that govern patch storage and manipulation apply as well to sequences. Some systems have room for only 1 sequence in memory while others can store as many as 20 (the big guys I mentioned before can hold much more). The procedures for saving and erasing a sequence are similar to those used for patches. Like patches, sequences can be saved onto cassette, cartridge, or disk.

Inputs and Outputs

You are probably wondering what all those inputs and ouputs on the back of your synthesizer are for. Well, I'm going to tell you.

IN/OUT





In/out back panel access. *Roland Juno 106*

Stereo/Mono Audio Outs

All synthesizers have an audio output. This is the output that you need to connect to an external amplifier in order to hear the sounds coming out of the synthesizer. Some systems offer stereo outputs, which can deliver the two halves of a chorused signal or the two different patches of a split or double keyboard.

Phone Out

Some synthesizers have an output that enables you to plug a set of headphones directly into the instrument; thus allowing you to practice without waking up your lover or your dog.

Footpedals and Switches

These inputs were covered in the section on footpedals when we talked about performance controls. Some are assignable according to the required function, and others are labeled for a specific function. These include functions such as volume control, LFO modulation, filter cutoff control, sustain, hold (latch), and patch shift.

CV/Gate

Some systems are able to send and/or receive a *CV/gate* (also called *trigger*). A CV is a control voltage that can determine frequency, and a gate is an on/off signal that triggers a note. The two together are what a synthesizer's keyboard sends to its own voice module instructing it to produce a note of a certain pitch. A synthesizer with **CV/gate** inputs is capable of receiving those same note-triggering instructions from an external source, for example, a sequencer or another synthesizer. Conversely, a synthesizer with **CV/gate** outputs can send those note-triggering messages to an external device. These inputs and outputs are gradually being replaced by MIDI (see Chapter 9, "MIDI"), which is capable of performing the same functions and much more.

Clock In/Out

If your synthesizer has an on-board sequencer or an arpeggiator, very often it is possible to sync the clock of the sequencer or arpeggiator up to an external device such as another sequencer or a drum machine. This would enable you to program pieces and then play them back in perfect time with other instruments.

Some systems enable you to do the same thing with the LFO. Syncing the speed of the LFO to an external sequencer, for example, enables you to create patches whose modulations are perfectly in time with the rhythm of a piece. This can be a very interesting effect. As with CV/gates, MIDI is also capable of replacing this function.

Cassette In/Out

This input and output is for the purpose of saving (receiving) and dumping (sending) patch and sequence information to and from a cassette or other tape recorder (see "Cassette Storage" in this chapter).

MIDI In/Out/Thru

These inputs and outputs are for connecting any two MIDI devices together. (See Chapter 9, "MIDI.")

Summary

Well, that's it for exciting features. There are a handful of others that you may run into on some obscure synthesizer or another, but we've covered the most important ones. One comment, if I may: The problem with explaining one feature after another in a specific order is that it is easy to lose sight of the fact that all these different features and functions are interrelated. We followed a logical progression from the six main components onwards. Just keep in mind that complex relationships exist among all these extra features as well as among the six main components. Many of these relationships have been discussed, but there are just as many that you'll have to discover for yourself. For instance, what would happen if you were to design a patch with an extremely slow LFO rate, and then arpeggiate it? I said it before, but I'll say it again—experiment.

And now, folks, having thoroughly covered the ins and outs of voltage-controlled synthesis, it's finally time to figure out what all the fuss is about regarding FM synthesis. Ready?

7

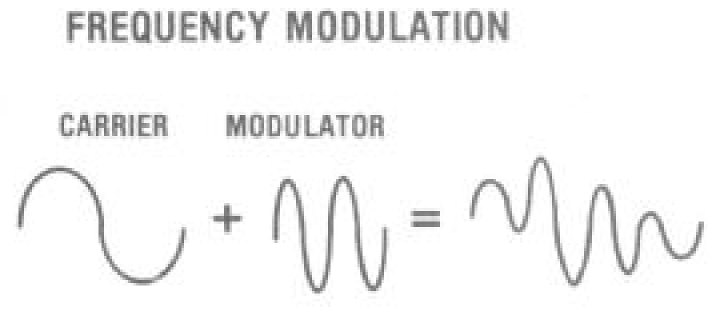
FM SYNTHESIS AND THE DX7

So far we have talked about two kinds of synthesis—additive and subtractive. As you recall, additive synthesis involves the combining (or adding) of many sine waves in order to create a timbre. Conversely, in subtractive synthesis you start with a harmonically rich waveform and filter out (or subtract) the unwanted harmonics to arrive at a desired timbre.

In the early 1970s, Dr. John Chowning, working out of Stanford University, came up with a relatively new kind of synthesis known as *FM*, which stands for *frequency modulation*.

FM is essentially this: When you take two sine waves in the audio range (the audible range of hearing, 20–20,000 Hz), and use one to modulate the other, the result is a waveform that is extremely rich in harmonics.

We call the wave that does the modulating the



One sine wave modulating another sine wave yields a waveform that is rich in harmonics.

modulator. The wave that is being modulated is called the *carrier.* The accompanying diagram illustrates what happens to a simple waveform, the carrier, when it is modulated by another waveform, the modulator. The result is a third waveform containing a relatively complex harmonic spectrum. Conceptually, it is almost like multiplication—you multiply one waveform by another to get a third waveform, the characteristics of which are the product of the first two waveforms. The actual mathematical formula for predetermining FM frequencies is somewhat more complicated than simple multiplication, but it is not really necessary to know the formula in order to achieve a practical understanding of FM. What is important to grasp is that FM is simply a kind of modulation.

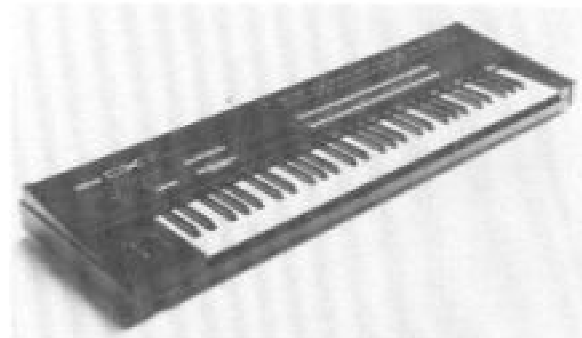
For example, when you use an LFO to modulate the frequency of an oscillator in order to achieve vibrato, that is a type of FM. Any time you use one frequency to modulate another you have, by definition, frequency modulation or FM.

The result of one frequency modulating another is the creation of new harmonics which occur on either side of the carrier wave. These harmonics are referred to as *sidebands.* If either of the two waveforms, the modulator or the carrier, is outside of the audio range, then the sidebands created will not be heard. It is only when both the carrier and the modulator are in the audio range that we can hear the resultant sidebands. It is these sidebands, in the audio range, that are responsible for the rich timbres of FM synthesis. So, even though LFO modulation is a type of FM, or frequency modulation, when we use the term *FM,* in regards to synthesis, we are really referring only to frequency modulation involving two audio waves.

An FM radio signal is another example of frequency modulation. Again, what distinguishes FM radio from FM synthesis is that an FM radio signal is made up of two frequencies of which one is far above the audio range. In order to create sound using FM synthesis, both frequencies must be within the audio range.

What is unique about this process are two things. One, the resultant waveform is more than *just* rich in harmonics: the particular nature of the complex harmonic spectra created by FM has an unusually acoustic and lifelike quality to it that is impossible to re-create using subtractive synthesis. Two, the ability to create such complex spectra using only two sine waves offers considerable advantages over additive synthesis, which would require as many as 100 individual sine wave generators to achieve close to the same harmonic complexity.

These two factors, the acoustic quality of FM sound and the fact that it is achieved using only two sine waves, explain why FM has had such an impact on modern synthesis. It has enabled the manufacturer to offer a completely new family of rich and vibrant sounds . . . at a reasonable price.



Yamaha DX7 Digital Synthesizer

When we talk about FM synthesis, we are really referring to a line of instruments made by a synthesizer company named Yamaha. In particular we mean the Yamaha DX7—the centerpiece of their FM line. The DX7 came out in 1983 along with its baby brother the DX9 (same FM, fewer options). These were essentially the first mass-produced, and affordable, FM synthesizers. (Yamaha actually introduced FM in 1980 on the GS1, but it was not programmable and was relatively high priced.)

Yamaha, a Japanese-based company (they also make motorcycles), deserves credit for recognizing the practical value of Dr. Chowning's research and for having the good sense to offer it in a beautifully designed instrument and at such a fair price. Yamaha owns the patent for FM as it applies to synthesis. (At present, the only other synthesizer that employs FM synthesis is the Synclavier, a powerful and high-priced music system that licenses the FM technology from Yamaha.)

So, for all practical purposes, this chapter describes the sound-generating principle behind a single instrument, the DX7, as well as the rest of its tight-knit family, members of the DX series—the DX9, the DX1, the TX816, and so on.

If you recall my definition of cross modulation, I described it as a primitive type of FM. That is the case. In cross modulation you have one audio oscillator modulating another audio oscillator. In fact, cross modulation on an analog synthesizer is capable of generating metallic tones that share some of the characteristics we normally attribute to FM synthesis. The basic principle is the same. The difference is in its application. The FM of cross modulation is extremely fixed and static. There are very few parameters for altering the frequency of both waves over time, and as a result only a very narrow range of sounds is possible. On the other hand, the DX7 offers an unlimited number of parameters for controlling and manipulating different aspects of the modulations, such that a broad range of timbres is possible.

The DX7 has taken the basic concept of FM synthesis and added to it in a number of different ways. First, instead of using only two sine waves to generate sound, it uses six. These six sine waves are generated by a digital circuit referred to as an *operator*. These operators are stacked in different configurations called *algorithms*. Depending on where it is placed in an algorithm, an operator will function as either a modulator or a carrier. (See the diagram.) The carrier is always the last in the chain. It is the output of the modulated carrier that you eventually hear. One algorithm might have five carriers, each being modulated by a single modulator. Another algorithm might have a single carrier, which is being modulated by a modulator that is in turn being modulated by another modulator and so on. The algorithms that contain more than one carrier are actually employing a combination of FM synthesis and additive synthesis. Each of these algorithms is capable of generating a different type of sound owing to their unique configurations.

One of the main characteristics of a natural acoustic sound is that its harmonics change dynamically over time. In analog synthesis this is approximated by using a filter envelope; but compared to FM synthesis, this single parameter is limited. In FM synthesis we have the ability to design separate volume envelopes for each of the six operators. As we increase the volume of an operator that is acting as a modulator, it increases the harmonics in the carrier's waveform. It is this ability to design complex multiple envelopes for the harmonics of a sound that enables FM synthesis to so realistically reproduce lifelike acoustic timbres. This is the essence of FM programming: being able to alter each of the six operators in an algorithm in order to achieve a desired sound. By controlling the frequency, the amplitiude, and the volume envelope of each individual operator, it is possible to create a virtually unlimited array of different timbres.

Another feature that the DX7 uses to enhance its FM sound is *feedback*. When you return a portion of an operator's output back to that same operator, you get a type of distortion or feedback that can be useful in creating sounds. In analog synthesizer terms, it is something like a cross between resonance and noise. Every algorithm has one operator that can be adjusted for feedback.

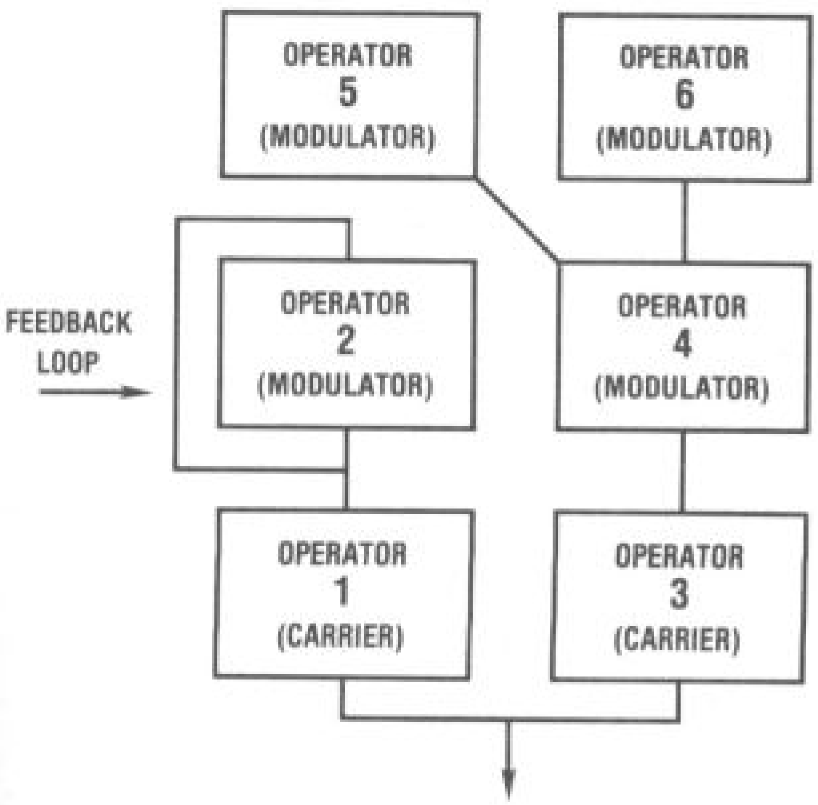
And that's it. That's FM. One audio wave modulates another and you get complex sounds.

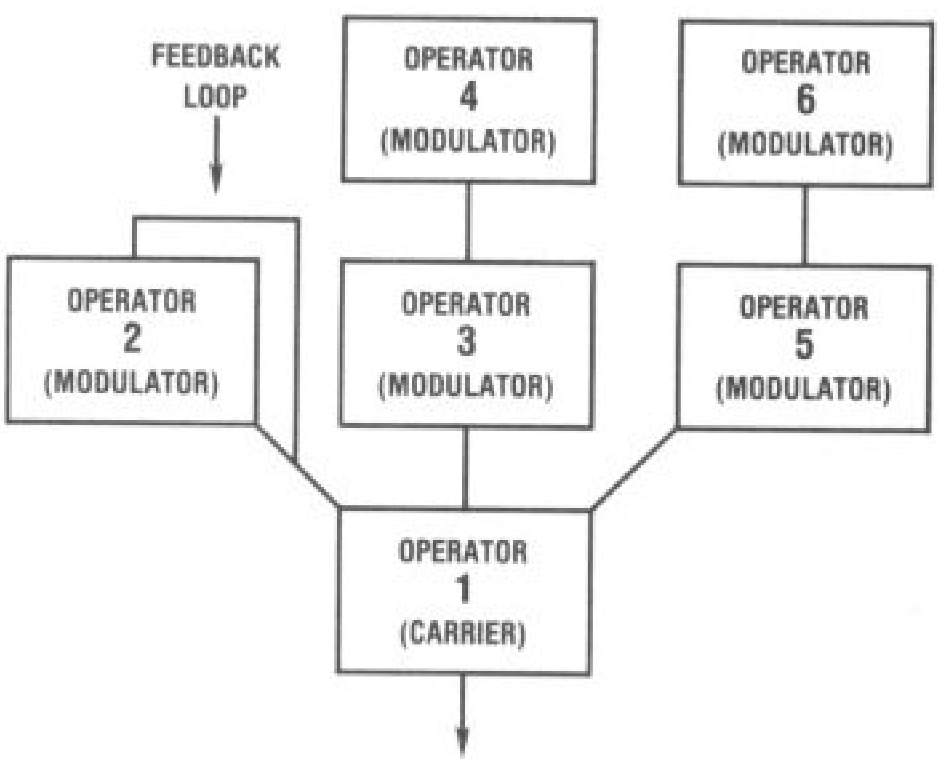
Of course, it's not that simple.

The DX7 evokes very strong reactions from people for two reasons. On the one hand, it is capable of getting truly beautiful sounds . . . woodwinds, percussion, bass, brass. The electric piano patch, alone, sounds so rich and clean that it has almost put the people that make the Fender Rhodes (the classic electric piano) out of business.

On the other hand, the DX7, as I indicated before, can be very difficult to program. This is for two reasons. First,

TWO DIFFERENT ALGORITHMS





it is completely step programmable, that is, each parameter must be entered one at a time via a single control. Second, the very nature of FM synthesis itself makes programming exceedingly difficult. Even though you can generate complex waveforms using only a pair of sine waves, predicting what the effect of altering one will have on the other can be difficult at best. When you then have to consider the relationships among as many as six individual operators, one modulating another, you can see how the entire process might seem daunting to some. It is difficult, but, not impossible. It is the fact that you have to program all of these subtle relationships among six separate operators, by step, that makes the process so tedious and confusing. I'm not saying this to intimidate you but to explain what all the fuss is about. Understanding the principle of FM synthesis is a snap; it is trying to remember the frequency, amplitude, and envelope settings on each of the six operators that can be a bitch.

That said, the DX7 contains so many useful and innovative features—even aside from how great it sounds—that it definitely warrants the extra effort involved in figuring it out.

It has an excellent touch-sensitive keyboard with very expressive aftertouch. It also offers, as an option, the breath-controller mouthpiece (see p. 66), a nifty little performance control feature that gives your mouth something to do while your hands and feet are busy. The DX7 also introduced the eight-stage envelope. Instead of the typical four-stage ADSR envelope, which lets you set three rates and one level for an envelope, the DX7's eight-stage envelope lets you set four rates and four levels for an envelope, giving you considerably more flexibility in determining the shape of an envelope.

The DX7 has two features called *keyboard rate scaling* and *keyboard level scaling*. These are similar to the keyboard tracking feature of an analog synthesizer except, instead of controlling the frequency cutoff point relative to where you are on the keyboard, they control the attack and decay rates and the amplitude levels across the keyboard.

Most of the other features on the DX7 are the same or similar to the features of an analog synthesizer. It has an LFO, which can be routed to modulate the pitch and/or the amplitude of an operator. There's portamento. There's a variable pitch bend wheel and an assignable modulation wheel—which can be made to control pitch, amplitude, and/or envelope amount (referred to on the DX7 as **EG bias**). Detuning, transpose, patch management (saving and recalling patches)—all of these familiar analog functions are found on the DX7.

I only want to add one more thing regarding FM synthesis and the DX7. People are constantly arguing the merits of FM synthesis versus analog synthesis, trying to determine which is better. This is silly. One is not better than the other. They are different. They are each capable of generating an entirely different family of synthesized sounds. Words are inadequate in trying to describe a type of sound, but, if I had to, I'd say that sounds generated by FM synthesis have an unusually natural and acoustic quality to them. They are also extremely clean sounding and at times can seem almost transparent. By comparison, sounds generated through analog technology tend to have a kind of rawness to them. They possess a power and bite (balls) that can sometimes (but not always) be hard to achieve with FM. They also have a richness and a type of warmth that can only be generated using analog oscillators and filters. My advice is to buy one of each and then MIDI them together. (Wait until Chapter 9 for an explanation of MIDI.)

8

OTHER SYNTHESIZERS, SAMPLERS, SYSTEMS

Phase Distortion and the Casio CZ101

There is one other type of synthesis that has been recently popularized by an instrument called the Casio CZ101. The CZ101 (along with its CZ relatives the CZ1000, the CZ5000, and so forth) generates its sound by a process called *phase distortion*.

Phase distortion can be considered a variation of FM synthesis in that it involves modulating one sine wave (or waves) with another.

The CZ101 contains eight selectable waveforms, which are analogous to the different algorithms used in FM. These waveforms contain groups of sine waves of different frequencies. The waveform that is used to modulate these selectable waveforms is called the *DCW* (digitally controlled waveform). When the DCW modulates the group of sine waves in one of the selected waveforms, it creates phase distortion between the different sine waves, due to the fact that each sine wave is of a different frequency. By altering the DCW, you change the quality and timbre of the sound.

In theory this is almost identical to FM synthesis. The main difference is simply in its application. The CZ101 is designed to more closely resemble the features and functions of an analog synthesizer and, in so doing, limits the amount of control you have over various aspects of the FM process. What the makers of the CZ101 have managed to achieve is an instrument that, although not as awe-inspiring as the DX7, has some of the most impressive sounds of any synthesizer on the market, is easier to program than the DX7, and, most incredible of all, costs less than $300.

In years to come you can expect to see more and more variations, such as this, of FM synthesis. Remember, all of this stuff is still brand new. So, keep your ears peeled. Or is it your eyes peeled and your ears clean . . . ? Or is it your nose clean and your ears open . . . ? You know what I mean. Just keep your head on straight, and don't get your nose out of joint because of these dumb jokes!

Sampling Devices

There are a number of electronic instruments on the market these days that are being referred to as synthesizers when in fact, strictly speaking, they are not synthesizers as we've come to define them. These devices would be more correctly referred to as *sampling devices* or *samplers*. They share many of the characteristics of a synthesizer, but they do need to be distinguished from the synthesizers we have been talking about in this book.

A sampling device is in fact a digital recorder. It makes an actual digital recording (or sample) of any natural sound and then allows you to trigger the playback of that

E-MU Systems' Emulator II

sound from its keyboard. At the same time, it will transpose the pitch of the sampled sound across the length of the keyboard, enabling you to play any sound—a whistle, a car horn, or a breaking glass—as if it were a pitched instrument. In other words, with a sampling device you could sample the *baaaa* of a real live lamb and then play the melody to "Mary Had a Little Lamb" on the keyboard as if the lamb were bleating out the melody. (I've heard this done and it's pretty hysterical.)

A more practical use of a sampling device would be to sample a regular instrument, say a trumpet or a violin, and then to go in and manipulate and alter the sample in order to change the original sound into something new and different. This is exactly what we do when we manipulate waveforms on a traditional synthesizer. In fact, many of the parameters for changing sampled sounds are the same ones that we find on any synthesizer, namely, *filter controls, envelope generators,* LFOs, and so forth. The only real difference, then, between most sampling devices and synthesizers are their sound sources. The sound source of a synthesizer is the VCO, which generates various waveforms electronically. The sound source of a sampler is whatever natural sound has been sampled. That natural sound—be it a bell ringing, a bird chirping, or a buzz-saw buzzing—then becomes the raw material for creating a new sound.

As you can see, because sampling devices do share so many of the characteristics of a synthesizer, there is some justification for calling them synthesizers. But they really are an entirely different ball game, so I think it is important to make the distinction between the two.

In any case, samplers are pretty incredible. They are capable of generating some of the most unusual hybrid sounds you've ever heard. But, rest assured. Almost all of the principles that you've learned in this book are directly applicable to playing a sampling device. If you understand a regular synthesizer, learning a sampler will be a breeze.

Super Systems

Throughout this book I've dealt mainly with the typical synthesizer, the kind that most of you are likely to own or have access to. These might cost anywhere from $200 to $5,000. But they all share a number of characteristics.

There are, however, a group of instruments that I refer to as the "super systems" that do not fall within the category of your typical synthesizer. First, these instruments cost anywhere from $10,000 to $100,000, and I suspect that most of you won't have one lying around your house. Second, these instruments are more than just synthesizers. They are complex, multifaceted computer-based music systems. There is a synthesizer somewhere in there, but these systems do much more.

The two most prominent of these super systems are the Synclavier and the Fairlight. It would take several volumes the size of the Manhattan Yellow Pages to explain all the things these systems are capable of, but I'll try to give you just a small idea of what they are about.

Both of these systems generate sound at least three different ways—additive synthesis, sampling, and something called resynthesis, which is a hybrid of additive synthesis and sampling. The Synclavier also has FM. This is the same FM that you'll find on the Yamaha DX7 (in fact, they license it from Yamaha). Oddly enough, neither of these systems has subtractive synthesis; but believe me, they already have enough sound generating capability to keep you busy for several generations.

Synclavier

Both of these systems offer up to 32 tracks of multitrack sequencing. This is almost equivalent to having a multitrack digital tape recorder on board your synthesizer.

Both of these systems include programs for printing out your music. All you have to do is sit down at the keyboard and play a piece. Then, with the touch of a button, whatever you played will be instantly printed out in perfect manuscript form, including the measures, ties, and rests.

There's much more, but suffice it to say that these systems are enormously powerful. The truth is that they are not so much synthesizers as they are miniature recording studios with keyboards attached to them. I wanted to mention them because they are pretty exciting, and also I wanted to give you something to aspire to while you're struggling away on the bandstand in the Captain's Lounge of the Holiday Inn. Who knows? You might win the lottery or something.

9

MIDI

I started out this book by saying that today's synthesizers were essentially keyboards attached to a computer. That is the case, even if that computer is in the form of a tiny microchip no bigger than your fingernail.

Now, within the last few years, the major synthesizer manufacturers got together and developed a way for all these computers to talk to each other. They created a common computer language designed especially for synthesizers. It is called *MIDI*, which stands for Musical Instrument Digital Interface.

Like any computer language, MIDI is a set of commands and instructions in digital form, that is, using the binary computer code of 0s and 1s. This digital language is designed to receive and send a set of instructions that can control almost all the parameters and functions of a synthesizer.

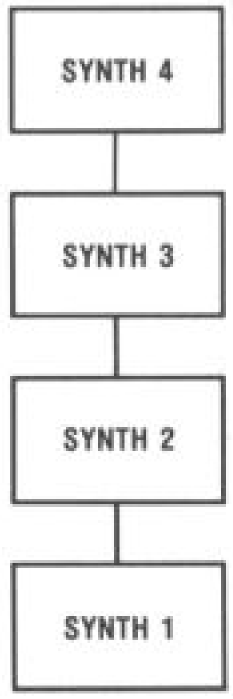
Here is a partial list of the type of information that MIDI is designed to receive and transmit: key depressions (what note is being played), key velocity (how fast a note is struck), pressure sensitivity (aftertouch), pitch bend and modulation wheel moves, footpedal and footswitch moves, patch changes, transpositions, and more. The list is only partial because as time goes on new operations are constantly being added.

All of this information, which encompasses almost all of the performance parameters on most synthesizers, can now be translated into the digital language of MIDI and then used in a number of extremely useful ways.

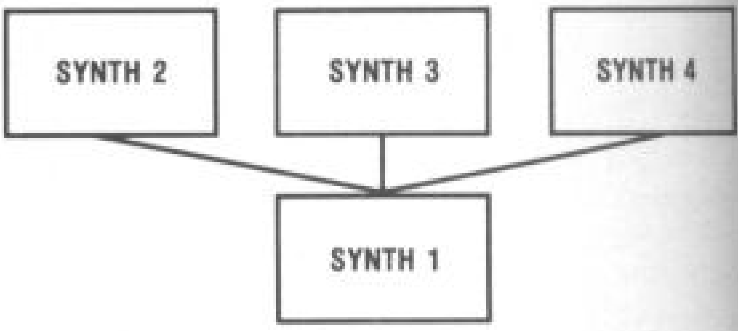
What follows are just a few of them.

MIDI-ing Synths Together

With MIDI it is possible to use one synthesizer to control another. (The type of cord used to connect one MIDI device to another is a standard five-pin DIN cable, otherwise known as a MIDI cable.) Once you have hooked two (or more) MIDI synths together, any notes or performance control moves (pitch bend, mod wheel, etc.) that you perform on one will be instantly played by the other(s). This means two things. First, it gives you the ability to play any number of different synthesizers from a single keyboard. Instead of having to jump from synthesizer to synthesizer, it is possible to access them remotely via MIDI. It also means that you now have the ability to mix timbres between completely separate instruments. The fact that



DAISY CHAIN



STAR NETWORK

any note you play on one synthesizer will be heard instantly on another gives the impression that the note was produced from a single source. This ability to combine timbres enables you to create sounds that would be impossible to realize on a single instrument.

Keyboard Controllers and Synthesizer Modules

One result of MIDI's ability to access a synthesizer remotely is the development of remote keyboard controllers and synthesizer modules. A keyboard controller is a synthesizer keyboard without the synthesizer. It is designed to control other synthesizers, but it is not capable of generating its own sound. It is the steering wheel, not the engine. A synthesizer module is just the opposite. It is a synthesizer without a keyboard—a synthesizer in a box. Since MIDI gave us the ability to access different synthesizers from a single keyboard, it obviated (made unnecessary) the need for multiple keyboards. All that was now necessary was the sound module of the synthesizer—its "innards." Being able to offer a synthesizer module without a keyboard has the obvious advantage of saving space, weight, and money. It is via MIDI that we are able to connect all these independent components of a synthesizer into a single musical network.

MIDI Sequencing

Another important use of MIDI pertains to sequencing. Prior to MIDI the only kinds of information that sequencers were capable of recording were key depressions (the notes played) and sometimes patch changes and transpositions. Those earlier sequencers were incapable of recording any performance control information such as pitch bend and mod wheel moves. Today's MIDI sequencers are capable of recording all those parameters, including velocity and pressure sensitivity, and as such have become in some ways almost superior to many multitrack tape recorders—at least as far as recording synthesizer parts. This is because, as was explained in the section on sequencers (see p. 72), a synthesizer performance recorded by a sequencer is infinitely more flexible than one recorded on tape.

Sixteen MIDI Channels

In addition to the set of coded instructions that controls a synthesizer's performance, the designers of MIDI included another important feature in the MIDI system. Instead of sending all MIDI information down a single path, they decided to give us 16 individual channels, each capable of controlling a separate synthesizer. This means that not only can synthesizers now talk to each other but that it is possible for them to carry on 16 separate conversations at once.

The concept of having 16 separate MIDI channels can be a little confusing at first, but here is an analogy that you may find helpful: Picture the Empire State Building. The TV transmitter on top of it is sending out 16 different TV shows at once, each on a different channel. There are 16 television sets spread out across the metropolitan area. If all 16 television sets are tuned to the same channel, then they are all going to show "Gilligan's Island." If they are each tuned to a different channel, then one is going to play "Gilligan's Island," one is going to show "The Price is Right," and another is going to play "The Partridge Family," and so on.

You see, in MIDI, the Empire State Building is the equivalent of a "mother" keyboard or a sequencer. (A mother keyboard is a keyboard that is capable of sending out MIDI information on various channels. A MIDI sequencer is capable of doing the same thing.) If you attach (MIDI together) a mother keyboard to an assortment of different synthesizers and assign a different MIDI channel to each, it then becomes possible to access each synthesizer simply by dialing up its MIDI channel number. If you set the mother keyboard to send on Channel 5, then whichever synthesizer has been set to receive on Channel 5 will be heard when you play the keys of the mother keyboard.

This gets even more remarkable when it comes to sequencing. When you play a synthesizer part into a sequencer, the sequencer will record not only the part but also the channel that you choose to send on. This means that you can record 16 different parts into a sequencer and, when it plays back, the sequencer will be able to drive 16 separate synthesizers—each playing a completely different part. Strange as it may sound, this is how more and more recordings are being made today.

Because a MIDI sequencer is capable of storing at least 16 separate keyboard performances, a large percentage of a recording can be programmed into a sequencer before you even step into a studio. This saves studio time, which in turn saves money. Once all of the parts have been stored in the sequencer, the sequencer and synthesizers are taken to the studio where the entire 16-part performance is dumped onto the multitrack tape recorder. The result is that all of those keyboard parts, which might otherwise have taken days to complete in the studio, are committed to tape in just a few hours. It also means that one person alone on a stage with a sequencer and a few synthesizers can now sound like an entire six-piece band.

Those are just a few examples of how MIDI is changing the way we make music. But, there's more.

MIDI Computer Interface

Once you have established a computer language for transmitting synthesizer information digitally, you open the door for any device that is capable of speaking that same digital language to become a part of the MIDI network. In other words, any device that can send digital commands using the MIDI language is now capable of controlling a synthesizer. For example, with the addition of a MIDI interface, any personal home computer can drive a MIDI synthesizer. This allows you to use the enormous power of a personal computer to greatly expand the capabilities of your synthesizer.

Computer software written expressly for this purpose enables you to do a number of amazing things by hooking up your synthesizer to a personal computer. For instance, there are sequencing programs that use the number-crunching power of your computer to turn it into a multitrack MIDI sequencer. There are programs that print out manuscript of whatever you play into the computer, so that all you have to do is play a piece, push a button, and you'll receive a printed version of what you just played, rests and all. There are programs that aid you in designing patches on instruments like the Yamaha DX7 and the Casio CZ101, which can otherwise be difficult to program. These programs provide a graphic representation of every changing variable in the patch and make it much easier to keep track of the dozens of interconnected programming parameters.

MIDI Tracking Devices

More recently, a new generation of MIDI tracking devices has been introduced to the consumer. These devices have the ability to read the pitch of any acoustic sound source—be it a voice, a flute, or a guitar—and then translate that acoustic sound into MIDI information, which is then capable of controlling any MIDI synthesizer. This means that it is now possible to play a synthesizer simply by singing into a microphone that is attached to a MIDI tracking device. This technology shifts the focus, for the first time, away from the keyboard as being the principal controlling device of a synthesizer. Now literally any instrument or sound-producing device has the potential to control a synthesizer, and to lend its own unique phrasings and characteristics to the already vast sound-generating abilities of a synthesizer.

MIDI is basically a manufacturer's standard. It originally evolved out of the manufacturer's decision to enable synthesists to plug two different makes of synthesizers together. As you can see, it has turned into something much more. It is a complex and living computer language for synthesizers; and as with all living languages, MIDI continues to evolve. As each new synthesizer feature or device arrives on the scene, it is added to the list of parameters controlled by MIDI and thus further expands the already enormous capabilities of this brand new synthesizer language.

10

A FINAL WORD ABOUT SYNTHESIS

This book was designed to provide an understanding of the fundamental principles behind synthesis and to enable you to apply that knowledge to create your own sounds. It was also designed to make a lot of money for me, your humble author; but my getting rich off this book is by no means a sure thing. What is a sure thing is this: No amount of book learning can replace actual time spent on a synthesizer. So, if you don't already own one . . . get one! Or borrow one. Or steal one. I don't care. Whatever you've got to do, get your hands on one. I guess what I'm trying to say is, now that you have the conceptual tools to work with, it's up to you to put them to use. All the exercises in this book are just a jumping-off point; you have to explore your instrument on your own. As I said in the beginning, each synthesizer is different. This can be frustrating, but it also gives you a chance to discover those particular features that make your instrument unique. So, play (in every sense of the word). That's how you'll really learn your instrument.

Understanding the function of the six main components on a voltage-controlled synthesizer will make you faster and more competent as a synthesist. A lot of time can be saved if you only think in advance about what it is you are trying to do. The goal of this book is to enable you to do just that: to imagine the kind of sound you are looking for, to calculate the kinds of settings it will require, and then to go for it. But, don't forget, just as important as the ability to calculate is the ability to intuit (to figure something out without thinking about it). In other words, you have to get *in to it* (ha, ha). Experiment, twiddle those knobs, slide those levers. If you are searching for a particular sound or effect, keep at it. Even if you are not precisely sure of the direction you are headed in, as long as you have something in mind, the chances are that eventually you'll arrive at a sound that is pretty close . . . or if not close, then just as good.

Anyway, I hope you enjoyed yourself. But, wait, before I go . . . as a special reward to those of you that bothered to read all the way through this book I will now give you the correct pronunciation of Dr. Robert Moog's last name. (Remember him? He invented the voltage-controlled synthesizer.) This is something that thousands of synthesizer players go to bed at night wondering about. Is it Moog as in a Bach *fugue?* Or is it Moog as in, miniskirts are in *vogue*. Well, which do you think? Care to take a guess? I'll sit here a moment while you mull it over . . .

On second thought, maybe I'll just wait until you buy a copy of my next book. Whatta you say?

ABOUT THE AUTHOR

As a songwriter and recording artist, Dean Friedman has sold millions of records in both the United States and England. Classic hits such as "Ariel" and "Lucky Stars" have earned him a devoted following on both sides of the Atlantic. As an author he has proven himself to be equally adept. The enormous popular and critical success of his first book, *The Complete Guide to Synthesizers, Sequencers & Drum Machines*, instantly established him as one of the foremost experts in the field of synthesis. Dean is currently the Director of the New York School of Synthesis located at 341 West 38th Street in Manhattan. A graduate of the Leonard Davis Center for the Performing Arts, he is presently preparing a series of educational videos covering all aspects of synthesis. Dean lives in upstate New York with his wife Alison, their two dogs, three birds, a monkey, and a pig.

ATTENTION SYNTHESISTS!

If you have any questions regarding synthesis that you would like answered, you may write to Dean in care of The New York School of Synthesis. Please send a self-addressed, stamped envelope to:

Dean Friedman
c/o The New York School of Synthesis
P.O. Box 878
Ansonia Station
New York, NY 10023

Oh, by the way, Moog is pronounced móg, as in miniskirts are in vogue.

Happy programming!

GLOSSARY

activate: The act of turning a knob on a synthesizer so as to make it "live," that is, to make it reflect its present setting.

additive synthesis: Synthesis involving the adding together of sine waves of different frequencies.

ADR: Attack, Decay, Release. A three-stage envelope generator.

ADSR: Attack, Decay, Sustain, Release. The most common type of envelope generator, a four-stage envelope that describes the shape of a sound.

algorithms: Different operator (oscillator) configurations used in Yamaha's FM sound-generating system.

amplifier: A device that increases the amplitude of a signal. See *VCA*.

amplitude: The amount of a signal is known as the amplitude. In the audio realm, amplitude corresponds to the perceived loudness of a signal. Loudness is the same as volume.

analog: Any device that operates through the use of direct and continuous electrical current. The opposite of digital. Digital units subtly coax sound out of a synth. An analog device hits it over the head with a hammer.

aperiodic: A random, irregular, or singularly occurring wave.

arpeggiator: A device that cycles continuously through a series of assigned notes.

attack: The first portion of a traditional ADSR envelope during which the amplitude of a sound goes from zero to maximum.

attenuate: To lessen or reduce a signal.

audible range: The frequency range of between 20 Hz and 20 kHz in which humans are capable of hearing sound.

audio range: See *audible range.*

audio wave: A wave whose frequency falls within the audible range.

auto-correct: When a sequencer or drum machine corrects the timing of a live performance by placing notes on the nearest beat. Also called time-correcting, and quantizing.

balance: A control that sets the relative amplitude between two oscillators or two voices. Also called mix.

band-pass filter: A filter that allows the frequencies within a certain range to pass through and only filters out the frequencies above and below that range.

bank: (1) A section on the front panel of a synthesizer pertaining to one of a synthesizer's main components, as in *oscillator bank.* Also called module. (2) A grouping of patches in the memory section of a synthesizer. (3) Where we would put money if we had any.

binary code: The 0s and 1s of base two, that make up the language of computers.

breath controller: A performance control, held in the mouth and blown into, that can apply various types of modulation to a note.

chain: The linking up of a number of patches or sequences. See *patch shift.*

channel: One of the 16 individual signal paths that are a part of the MIDI language.

chord memory: A feature that memorizes the intervals of a chord and allows you to play that chord by playing any single key. See *latch.*

chorus: The thickening or fattening of a sound achieved by tuning two oscillators slightly apart.

clock: A circuit that regulates the timing of a sequencer, an arpeggiator, or an LFO.

control voltage: See *CV.*

cross modulation: A primitive type of frequency modulation in which one wave modulates another.

cutoff frequency: See *Fc,* frequency cutoff point.

CV: Control Voltage. A voltage that is used to control the parameters of an electronic component such as an oscillator or an amplifier.

CV/Gate: A two-part signal for triggering an analog synthesizer; the control voltage determines pitch and the gate triggers the note.

cycle: A single unit of vibration.

DADSR: Delay, Attack, Decay, Sustain, Release. A five-stage envelope generator.

data: Digital information.

dB: Decibel. A unit for measuring the relative loudness of sound. For example, if a sound doubles in volume, you get an increase of 6 dB.

DCO: Digitally Controlled Oscillator. An oscillator whose frequency is regulated by a digital circuit.

DCW: Digitally Controlled Waveform. Casio's version of an FM modulating wave.

decay: The second portion of a typical ADSR envelope during which a sound's amplitude goes from maximum (or peak) to sustain level.

DEG: Digital Envelope Generator. A digital version of an envelope generator.

delay: (1) The portion of a five-stage envelope generator that precedes the attack portion of an envelope and that determines the time it takes before attack begins. (2) On an LFO, a control that determines the time it takes before the LFO begins.

detuning: The act of tuning the pitch of two oscillators apart. See also *chorus*.

digital: Any microprocessor-based device that operates through the use of numbers (computer binary code) that represent information; as opposed to analog, which uses direct and continuous electrical current.

double mode: A keyboard mode that allows you to play two different patches on the same note.

duty cycle: The width of a pulse wave expressed in percentages.

dynamics: The control of loudness via a velocity or pressure-sensitive keyboard or other triggering device.

edit: The act of designing and creating your own sounds (patches) or altering already existing patches. See *program*.

edit/compare: A feature that enables you to switch back and forth between an original patch and its edited version without having to save the version currently being edited.

emphasis: See *resonance*.

ensemble: See *chorus*.

ENV: Envelope. A shape that changes aperiodically over time; used to shape the amplitude, timbre, and/or pitch of a sound.

envelope: See *ENV*.

envelope amount: A control that determines the amount of effect an envelope will have on the filter.

envelope generator: A device that creates an envelope.

Fc: Filter cutoff point. The frequency at which a filter begins to take effect. Also called the cutoff frequency.

filter: A device that filters or blocks out portions of a waveform.

filter cutoff point: See *Fc*.

filter envelope: An envelope that, when applied to the filter cutoff point, determines the shape of a sound's timbre (its harmonic content).

five-pin DIN: Standard cable connectors used in MIDI.

FM: Frequency Modulation. Yamaha's patented sound-generating technique that creates complex waveforms through frequency modulation of sine waves.

footpedal: A performance control played with your foot that can apply various types of modulation, involving a changing voltage, to a note.

footswitch: A performance control played with your foot that can turn on or off various parameters on a synthesizer such as vibrato or sustain.

four-pole filter: A filter whose rolloff characteristics are −24 dB per octave. The most common analog filter; richer and warmer sounding than the two-pole filter.

frequency: The number of cycles per second that a signal vibrates (oscillates).

fundamental: The frequency in a sound that is both the loudest and the lowest of all the frequencies in the sound. Also called the first harmonic.

gate: The opening and closing of a signal path that occurs when you play and then release a note; the most basic envelope.

glide: See *portamento.*

glissando: A type of portamento in which one note glides to another in chromatic steps.

harmonics: Frequencies that bear a mathematical relationship to a fundamental frequency and combine to create the timbre of a sound. Also called partials or overtones.

hertz: See *Hz.*

high-pass filter: A filter that allows the upper frequencies in a waveform to pass through and only filters out the lower frequencies.

hold: A control that sustains whatever note or notes are played once it has been engaged.

Hz: Hertz. The number of cycles per second in a wave.

invert: See *polarity switch.*

joystick: See *X,Y joystick.*

keyboard controller: A keyboard that contains no synthesizer voice module but that can control another synthesizer or synthesizer module.

keyboard setup: Overall edit parameters—such as transpose, split point, and pitch bend range—which are separate from the patch itself and can be stored separately or alongside the patch.

keyboard tracking: Controlling the proportional cutoff frequency of the filter relative to where you play on the keyboard.

latch: (1) On an arpeggiator, a switch that continues the arpeggiation even after you remove your fingers from the keys. (2) Another name for chord memory.

legato: A way of phrasing notes so that they flow smoothly from one to another with no separation between them.

LFO: Low Frequency Oscillator. A modulating wave that operates in the subaudio range.

LFO trigger: A performance control switch that introduces LFO into a note.

load: To enter data.

looping: A feature that causes a sequence to repeat over and over again.

loudness: See *amplitude.*

low-pass filter: A filter that allows the lower frequencies in a waveform to pass through and only filters out the upper frequencies. Most synthesizers employ a low-pass filter.

MG: Modulation Generator. Digital equivalent of an LFO; not a sports car.

MIDI: Musical Instrument Digital Interface. A computer language for synthesizers.

mix: See *balance.*

modifier: A device or signal that changes or alters another device or signal. Similar to modulator.

modulation: (1) The application of control voltage to a voltage-controllable parameter in order to change the character of the audio signal. (2) Any fluctuation or change imparted by one synthesizer component onto another.

modulation wheel (lever): A performance control, in the shape of a wheel (or lever), that applies modulation to a note.

modulator: A device or signal that imparts a change or fluctuation onto another device or signal. Similar to modifier but having more to do with periodic waves.

module: (1) A section or bank of a synthesizer, as in a filter module. (2) A synthesizer without keys; a box containing all the components of a synthesizer except for the keyboard, made to be triggered via an external device such as a remote keyboard controller or sequencer.

mother keyboard: A keyboard in a MIDI setup that is used to control other synthesizers.

multiple triggering: A mode in which the envelope retriggers every time a note is played, regardless of whether they are played legato or staccato.

multitimbral: An instrument capable of producing more than one sound (patch) at the same time.

neutral: For the purpose of the exercises in this book, the most basic patch you can program; a single unmodulated waveform. See Introduction.

noise: Random fluctuations in sound involving all frequencies, perceived by the ear as hiss. Comes in different flavors—white and pink—which emphasize different parts of the frequency spectrum.

notch filter: A filter that only filters out the frequencies within a given range and allows all other frequencies above and below that range to pass through.

octave switch: A control that moves an oscillator's frequency up or down in octaves. Also called range.

operators: Yamaha's digital equivalent of an oscillator.

OSC: Oscillator. An electronic circuit that produces a constantly repeating waveform. Sort of an electronic sausage machine.

oscillator: See *OSC.*

patch: A sound created on a synthesizer.

patch shift: A feature that enables you to step from one patch to another in a predetermined order. Also called chain.

PCM: Pulse Code Modulation. A protocol for storing digital samples on a chip.

peak: The portion of an envelope during which a sound reaches its maximum amplitude. Peak occurs between the attack portion and the decay portion of the typical ADSR envelope.

performance control: Any device, such as a pitch bend wheel or footpedal, that is capable of altering some aspect of a note, in real-time (as it sounds).

periodic: A regularly repeating wave.

phase distortion: Casio's sound-generating method which generates complex waveforms by distorting the phase relationships between a group of sine waves. A variation of FM.

pitch: In music, the highness or lowness of a sound determined by its frequency.

pitch bend: A change in pitch that occurs during a portion of a note, either applied via a performance control or programmed into a patch with a pitch envelope.

pitch bend wheel (lever): A performance control in the shape of a wheel (or lever) that applies pitch bend to a note.

pitch envelope: An envelope that produces a change in pitch when applied to the oscillator.

polarity switch: A switch that reverses the polarity of an envelope; it turns it upside down. Also called invert.

polyphonic: An instrument capable of producing more than one note at a time.

portamento: A feature in which one note glides smoothly into another note; also called glide.

preset: A factory-made patch that is often permanently stored at a certain location in your synthesizer; or any patch already in your synthesizer.

pressure sensitive: Also known as aftertouch. A feature on a keyboard that measures the additional depression of a key after it has been struck and then applies that measurement to various performance control parameters.

program: To edit a patch or to enter data. Also called edit.

pulse wave: A variable square wave. See *square wave.*

pulse width: A control that allows you to change the duty cycle of a pulse wave.

PWM: Pulse Width Modulation. Varying the width of a pulse wave over time and thereby changing its timbre.

Q: See *resonance.*

quantize: See *auto-correct.*

RAM: Random Access Memory. A memory cartridge for storing data that you can record to as well as read from.

ramp wave: See *sawtooth wave.*

range switch: See *octave switch.*

rate: Any control that determines the speed or frequency of a component such as an LFO or an arpeggiator.

real-time: The ability to program a unit by playing live, as opposed to having to load it by step.

release: The final portion of a typical ADSR envelope during which a sound's amplitude goes from the sustain level back to zero.

release velocity: A keyboard feature that measures the speed with which you take your finger off a note and then applies that measurement to various performance control parameters.

remote keyboard controller: See *keyboard controller.*

resonance: The emphasis of the frequencies at a filter's cutoff point, which creates a kind of feedback.

resynthesis: A type of synthesis that is a hybrid of additive synthesis and sampling.

ribbon controller: A pressure-sensitive strip that puts out a control voltage as you drag your finger across its length; used for pitch bend.

rolloff: The rate of attenuation of a filter; how exactly a filter filters out harmonics at its cutoff point.

ROM: Read Only Memory. A memory cartridge for storing data. The data can be retrieved, but no new data can be recorded by the user.

sampler: A device that makes a digital recording of a sound and can then play it back when triggered.
sawtooth wave: A bright-sounding waveform whose shape resembles a sawtooth. Also called a ramp wave.
sequence: A unit of music, from a phrase to an entire composition.
sequencer: A recording device that remembers key depressions and digital events as opposed to the actual sound of an instrument.
signal: Electrical impulses.
sine wave: A pure wave containing no harmonics whose shape resembles a sine curve.
single triggering: A mode in which an envelope only retriggers when there is a separation between notes. When you play legato in a single-trigger mode, all the notes share the same envelope.
source: The place or component where a sound originates; the reed of a saxophone or the oscillator in a synthesizer.
speed: See rate.
split mode: A keyboard mode that splits a keyboard in half and assigns a different patch to each half of the keyboard.
square wave: A hollow-sounding waveform whose shape is square.
staccato: A way of phrasing notes that exaggerates the space between them, so that they are clearly heard as separate notes.
step programming: (1) The act of programming and editing on a synthesizer one function at a time, using common entry keys, as opposed to having a separate knob for each parameter. (2) The act of programming a sequencer or drum machine, note by note, as opposed to playing into it live.
suboscillator: A circuit that produces a waveform, derived from the main oscillator, which is one or two octaves below the main oscillator's frequency.
subtractive synthesis: Synthesis involving the filtering out of harmonics from a harmonically rich waveform.
sustain: The third portion of a typical ADSR envelope during which a sound's amplitude remains at the same level for as long as a key is held.
sustain pedal: A footpedal that lengthens the release portion of a note's volume envelope in order to duplicate the **sustain** pedal on an acoustic piano.
sweep: See *LFO*.
sync: When two oscillators are locked together at the beginning of each cycle.
synthesizer module: See *module*.
timbre: The qualities of a sound that distinguish one instrument or patch from another; for the most part, the perceived differences in harmonic content.
time-correct: See *auto-correct*.
touch-sensitive keyboard: A keyboard that is either velocity or pressure sensitive.
tracking device: A device that translates an acoustic pitch into a control voltage or MIDI signal which can then drive a synthesizer.
transpose: To change the frequency or pitch of the keyboard.
tremolo: A periodic fluctuation of volume that occurs when a voltage-controlled amplifier is modulated by an LFO.

triangle wave: A smooth-sounding waveform whose shape resembles a triangle.

trigger: (1) A signal sent by a keyboard or other controlling device that tells the envelope generator when to begin its cycle. (2) On an LFO, a performance control switch that introduces the LFO. (3) Roy Rogers's horse.

two-pole filter: A filter whose rolloff characteristics are −12 dB per octave. Less common than the four-pole filter and not as rich sounding.

unison: A feature whereby all the voices on a polyphonic instrument are assigned to a single key.

VCA: Voltage-Controlled Amplifier. A device that adjusts the amplitude of a signal proportionally to the control voltage applied to it.

VCF: Voltage-Controlled Filter. A filter whose cutoff frequency is proportional to the voltage applied to it.

VCO: Voltage-Controlled Oscillator. An oscillator whose frequency is proportional to the voltage applied to it.

velocity sensitivity: A keyboard feature that measures the speed with which you strike a key and then applies that measurement to various performance control parameters.

vibrato: A periodic fluctuation of pitch that occurs when an oscillator is modulated by an LFO.

voice: (1) The components that make up the sound source. (2) A unit of polyphony; the sound generated by a single key depression.

voltage-controlled synthesizer: A synthesizer that operates using components that respond to control voltages.

volume: See *amplitude.*

volume envelope: An envelope determines the shape of a sound's volume when it is applied to the amplifier.

wave: (1) The form in which energy, resulting from a periodic vibration, travels through a medium. (2) The form in which sound travels through air. (3) In synthesis, a periodic signal (such as a square or sine wave) generated by an oscillator. The terms *wave, waveform,* and *waveshape* are often used interchangeably.

waveform: See *wave.*

waveshape: How we refer to a wave when referring to its physical characteristics. See *wave.*

X,Y joystick: A performance control, that moves in two axes, that can apply pitch bend and modulation to a sound.

INDEX