

Chapter 16

DSP Functions

This chapter explains the DSP functions that can be inserted into the algorithms in the Program Editor. As you configure each algorithm, the DSP functions you select determine the type of synthesis you apply to your sounds. Deciding which algorithm to use depends on what you want to do; there's no hard and fast rule. If you want to create a classic analog sound, for example, you'll choose one of the algorithms containing one or more blocks that can have filter functions assigned to them. If you want real-time panning effects, choose an algorithm that includes the PANNER function in the F3 block. Your best approach is to study the algorithm charts in the *Musician's Reference*, and choose the algorithm that includes the functions you want to work with.

Before we get to the explanations of the DSP functions, we've included a brief discussion of a few general concepts of sound synthesis. This should help you understand the workings of the DSP functions. We'll refer to these concepts repeatedly as we go along.

Any single sound waveform is composed of numerous sine wave components, each at a different frequency. These components are called partials. The lowest frequency is perceived by the ear as the pitch of the sound, and is called the fundamental. The other components are called harmonics. The relative amplitudes (volume) of each of the partials in a sound determine its timbre, its most recognizable characteristic. When you think of the difference between the sound of a piano and a saxophone, you're thinking about their different timbres. A dull sound has a strong fundamental and weak harmonics, while a bright sound has strong harmonics.

Sound synthesis can be most simply described as the manipulation of either the amplitude or phase of one or more of the partials constituting a sound. The K2600's various DSP functions give you a variety of methods for manipulating those partials. We've grouped our explanations of the DSP functions according to the types of specialized manipulation they enable you to perform on a given sound. The categories are as follows:

Filters	Added Waveforms
Equalization (EQ)	Nonlinear Functions
Pitch / Amplitude / Pan Position	Waveforms with Nonlinear Inputs
Mixers	Mixers with Nonlinear Inputs
Waveforms	Synchronizing (Hard Sync) Functions

Introduction to Algorithm Programming

Programming the algorithms is a multi-step process. The first step is selecting an algorithm. Changing the algorithm of an existing program's layer is likely to alter the sound of the layer dramatically. As a rule, then, you won't want to change a layer's algorithm unless you're building a sound from scratch. Furthermore, when you change a layer's algorithm, the values for each of the DSP functions within the algorithm may be set at nonmusical values; you should lower the K2600's volume slider before changing algorithms.

Deciding which algorithm to use for a new sound is primarily a process of planning a layer's signal path through the sound engine. The real sound manipulation is done by the DSP

functions you insert into the algorithm. The algorithm simply lays a framework that determines how the DSP functions interact.

Once you know which algorithm you're going to work with, you'll assign various DSP functions to each of the stages of the algorithm. These stages, as you recall, are represented by the rectangular blocks you see on the ALG page. The arrows pointing down at the blocks represent control inputs that affect the behavior of the DSP functions. For each arrow, there's a page of parameters controlling some aspect of the DSP function's behavior. Every DSP function has at least one control input; several have two or three.

The ALG page is where you select algorithms and assign DSP functions to the algorithm's various stages. To assign a DSP function, move the cursor to select the stage you want to modify, then use any data entry method to scroll through the list of available DSP functions for that stage. You'll normally hear the effect of each selection as soon as you make it. If you don't hear a difference, it's because the function's control parameters aren't set to significant values. Once you adjust some of these parameters, the function will have a noticeable effect on the sound. Keep in mind that not all DSP functions are available at every stage of every algorithm.

When you have each stage of the current algorithm set up to your liking, you can begin to program the control inputs of each DSP function. This is done by selecting the control-input page(s) for the currently selected DSP function, and adjusting the parameters on the page. There are two ways to select the control-input pages: you can move the cursor to select the DSP function you want to tweak, and press **Edit**. The selected DSP function's control-input page will appear (if it's a multi-stage DSP function, its first control-input page will appear). Or you can use the soft buttons to select the pages. The **PITCH** soft button always selects the pitch control-input page, since the first stage of every algorithm is invariably the pitch control. The **F1-F4** soft buttons select the control-input pages corresponding to the remaining four arrows, which point down at the subsequent four variable control inputs.

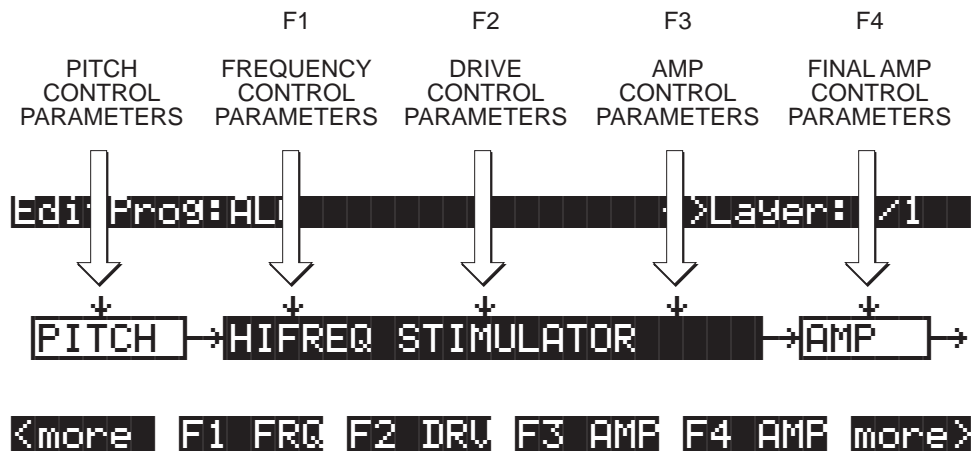


Figure 16-1 Input Control for DSP Functions

Each control-input page contains several parameters, which affect some aspect of the behavior of the DSP function named on the top line of the page. Most of these parameters are the common DSP control parameters; for a review, see *Common DSP Control Parameters* on page 6-14.

The possibilities are truly enormous, given the number of different combinations of functions you can assign to any particular layer (not to mention multi-layer programs, each layer of which has its own algorithm). You can create completely new sounds just by tweaking the parameters on the control-input page for a single DSP function. When you begin adjusting these

parameters, it's a good idea to start with all of them set to 0 (or the value that minimizes their effects), then adjust them one by one. This will help you understand exactly what effect each parameter has, and will give you an idea of the variety of effects each parameter can produce. Then you can start combining the effects of multiple parameters, and quite possibly take your sound in a whole new direction. You'll quickly become familiar with the control-input pages for the DSP functions, since most of them contain the same parameters, with just a few variations. You'll find that they all look much alike. The top line of each page, however, will indicate which DSP control input it represents (PITCH, or F1–F4).

For example, on the page below, the top line tells you that the currently selected DSP function is the high-frequency stimulator—its name is abbreviated and enclosed in parentheses. You can also see that you're looking at F1, which in this case controls the frequency of the high-frequency stimulator. So the top line of these pages always shows three things:

1. The currently selected control input (PITCH or F1–F4);
2. The aspect of the current DSP function controlled by the input (this varies depending on the current DSP function);
3. The currently selected DSP function (usually abbreviated, and in parentheses). Items 1 and 2 match the label of the soft buttons that select each page. The page below, for example, is selected with the soft button labeled F1 FRQ

```

EditProg:F1 FRQ(H)FRQ S 110<>LAYER:1/1
Coarse:C 4 262HZ Src1 :OFF
Fine :Oct Depth :Oct
Src2 :OFF
KeyTrk:Oct/key DptCtl:OFF
VelTrk:Oct MinDpt:Oct
Pad :dB MaxDpt:Oct
<more F1 FRQ F2 DRU F3 AMP F4 AMP more>
    
```

Additional Parameters

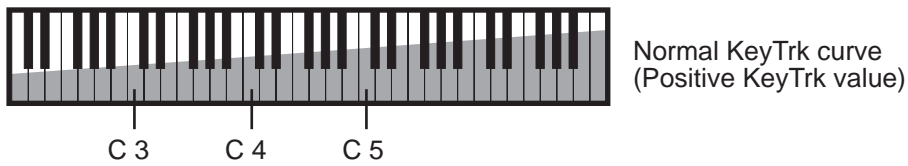
In addition to the common DSP control parameters you'll find on each page, you'll also see a few others on various pages. They're described here, since programming them is the same regardless of the page on which they appear. Depending on the DSP function they affect, they'll have different ranges of values and different units of measurement (% , dB, etc.).

Pad

Many of the DSP functions boost the signal as it passes through. Depending on the signal's input level and the amount of gain (boost) introduced by any given DSP function, its output may clip, which will alter the sound considerably. Clipping may also occur as a result of phase shifting, but this is not as common as clipping caused by gain. While you may find clipping to be a useful component of some sounds, you'll want to remove it from others. The Pad parameter, which appears on the control-input pages of many DSP functions, lets you attenuate (reduce the amplitude of) the signal by 6, 12, or 18 dB at the input of those functions. Use the Pad parameter to reduce or eliminate any undesired clipping caused by the currently selected DSP function.

Key Track Start (KStart)

This parameter appears on many control-input pages, and gives you added control over the effect of key tracking. For each note you play, it multiplies the value of the KeyTrk parameter by a number that varies with the note’s MIDI key number. If KeyTrk is set to 0, this parameter will have no effect. When KeyTrk is a nonzero value, KStart will modify the normal key tracking curve, which is shown in the diagram below. The effect of normal key tracking reaches its minimum at C -1, and its maximum at C 9. You can use KStart to dampen the effects of key tracking at one end of the keyboard. If key tracking causes a sound to clip or distort toward the high end of the keyboard, for example, you can use KStart to reduce the effect of the key tracking at the upper end without changing its effect on the lower end. To do this you would set a negative value for KeyTrk, and a unipolar value for KStart.



Unipolar Keystart

The range of values for KStart is C 1 to C 9 unipolar, and C -1 to C 9 bipolar. Unipolar and bipolar values have different effects on the key tracking. The next three diagrams illustrate the effect of three different *unipolar* keystart values on the key tracking curve when a positive value is assigned for the KeyTrk parameter. At a KStart value of C 4, for example, there is no key tracking effect below Middle C (it multiplies the key tracking amount by a key number value of 0). The key tracking value is multiplied by 0 at C 4 (normal key tracking), by 1 at C[#] 4, by 2 at D 4, and so on to a maximum of 64 at 5 1/3 octaves above the KStart value. For higher notes, the key tracking effect would still increase on its own, but the effect of keystart would not increase further. At a KStart value of C 3, the key tracking value would be multiplied by 0 for C 3 and all notes below, by 1 for C[#] 3, and so on. The key number value would reach its maximum of 64 before reaching C 9. When KStart is set above C 4, its effect on key tracking will continue to increase up to C 9, but will not reach full scale at C 9.

You’ll use unipolar values for KStart when you want to cancel the key tracking effect on a DSP function over a sizable portion of the keyboard, but have it increase or decrease throughout the rest of the keyboard’s range. Set high unipolar values for KStart when you want to remove key tracking from the lower notes, applying it only to the higher notes. If you have set a positive value for KeyTrk, set low unipolar values when you want to apply key tracking to the lower notes and pin it at its maximum throughout the upper range of the keyboard. You may want to use low values of key tracking in this case, depending on the DSP function you’re applying.

When the value of the KeyTrk parameter is negative, remember that the key tracking is at its minimum effect at C 9, and maximum at C -1. In this case, the key tracking effect will be reduced for notes above the KStart setting. For notes below the keystart note, the normal key tracking amount will apply.

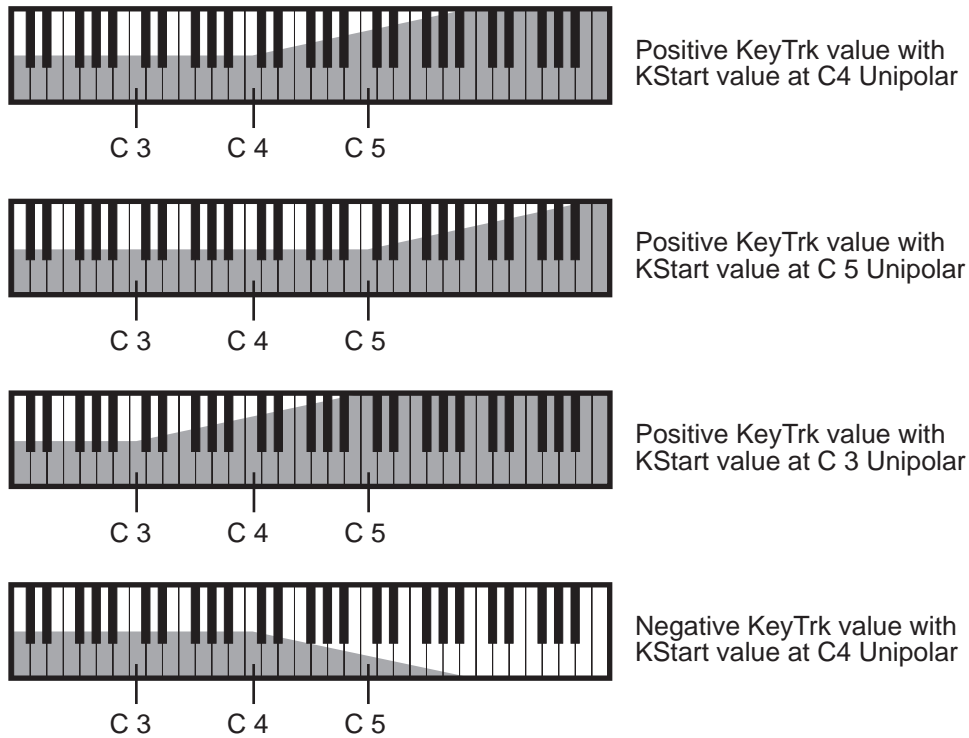


Figure 16-2 Unipolar Keystart

Bipolar Keystart

For bipolar KStart values with positive key tracking values, the effect on key tracking depends on whether the KStart parameter is set above or below C 4. When it's above, the effect on key tracking will be minimum at C -1, reaching its maximum effect on key tracking at the keystart setting. The normal key tracking curve applies above the keystart setting. When KStart is set below C 4, the effect on key tracking is maximum at C 9, decreasing with each successive note closer to the keystart setting, and remaining constant at the keystart setting and below. The normal key tracking curve applies below the keystart setting.

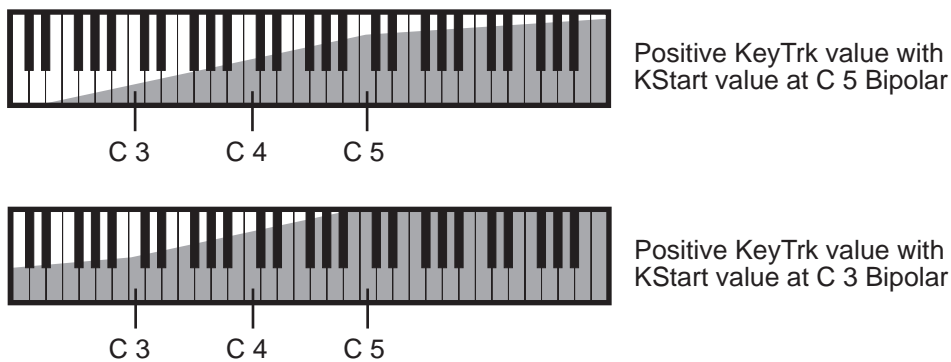


Figure 16-3 Bipolar Keystart

Use bipolar settings for KStart when you want to gradually increase or decrease the key tracking effect of the currently selected DSP function across the entire keyboard range. With KStart at C 4 bipolar, playing C 4 will apply the DSP function at the level you set with the Adjust parameter, and will increase or decrease with higher or lower notes, depending on your settings for KeyTrk.

When KeyTrk is set to a negative value, the effect on key tracking is reversed. For keystart settings above C 4, the effect on key tracking will be maximum at C -1, decreasing with each note closer to the keystart setting, and remaining constant at and above the keystart setting. For keystart settings below C 4, the effect on key tracking will be minimum at C 9, increasing with each note closer to the keystart setting, and remaining constant for notes at and below the keystart setting.

KStart is available for many of the *nonlinear DSP functions*, like SHAPER and WRAP. If you like the control it gives you, you can simulate its effect by using the FUNs. To simulate unipolar keystart, assign Key Number (KeyNum) as one of the inputs to a FUN, and select one of the diode equations for the FUN's Function parameter. To simulate bipolar keystart, assign Bipolar Key Number (BKeyNum) as one of the inputs of a FUN. Then assign those FUNs to some other control-source parameter.

The DSP Functions

Filters

One-pole Lowpass	One-pole Allpass
Two-pole Lowpass	Two-pole Allpass
Two-pole Lowpass, -6 dB resonance	Two-pole Notch
Two-pole Lowpass, +12 dB resonance	Two-pole Notch, fixed width
Four-pole Lowpass with separation	Double Notch with separation
Gated Lowpass	Two-pole Bandpass
One-pole Highpass	Two-pole Bandpass, fixed width
Two-pole Highpass	Twin Peaks Bandpass
Four-pole Highpass with separation	

Filters are widely used in synthesis to change the timbre of a sound by manipulating the amplitude of specific partials. When using filters, you always set a reference point (cutoff or center frequency) that determines which partials the filters affect. Here's a quick summary of the effects of the filter functions.

Lowpass filters cut the levels of all partials above the cutoff frequency without affecting the partials at or below the cutoff frequency (the low frequencies pass through). Highpass filters do the opposite; they cut the levels of all partials *below* the cutoff frequency without affecting the partials at or above the cutoff frequency.

Notch filters, as the name implies, cut the levels of partials in a range between high and low frequency. Consequently the "cutoff" frequency is referred to as the center frequency. With notch filters, the levels of partials at the center frequency are cut, while the levels of partials above and below the center frequency are unaffected. Bandpass filters are the opposite of notch filters; they leave the levels of partials at the center frequency unchanged, and cut the levels of partials above and below the center frequency.

The use of lowpass, highpass, notch, and bandpass filters is often referred to as subtractive synthesis, since the timbre of a sound is changed by removing certain partials.

Allpass filters, instead of cutting or boosting the partials of a sound, change the phase of the partials as their frequencies pass through the center frequency.

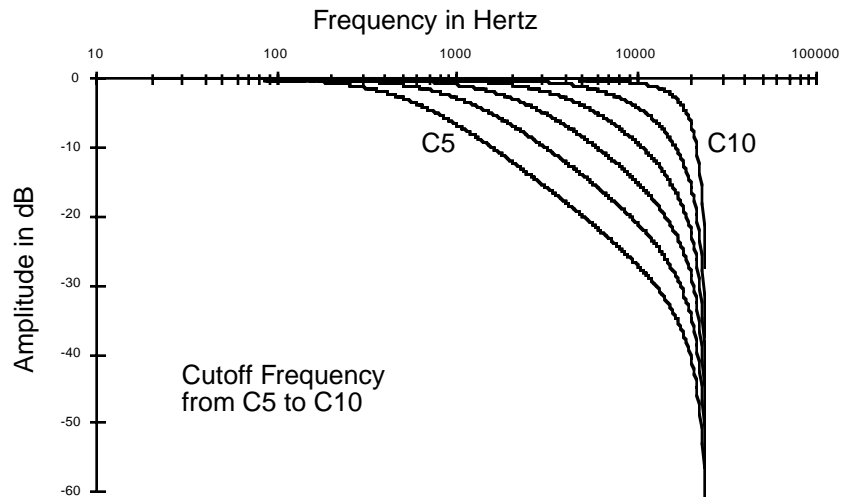
Filter Terminology

- Rolloff** Filters do not usually cut all frequencies precisely at their cutoff point. Instead, the amplitude of the frequencies above (or below, in case of a hi pass filter) the cutoff decrease by a fixed amount per octave—for example, 6 dB per octave. This curve of lessening amplitude is called a rolloff.
- Poles** The number of poles in a filter affect how sharp the rolloff is. The more poles there are, the sharper the rolloff, meaning that the cutoff will have a more dramatic effect on the sound. The K2600 has one-pole, two-pole, and four-pole filters available. A one-pole filter has a 6 dB per octave cutoff; a two-pole is 12 dB per octave; and a four-pole is 24 dB per octave.
- Resonance** In a filter that has resonance, the frequencies near the cutoff are given an increase or decrease in amplitude. If you decrease these frequencies, you are essentially creating a longer rolloff. But if you increase those frequencies thereby emphasizing them, it creates a distinctive sound that you will very likely recognize. Resonance is also sometimes called Emphasis or Q on various synthesizers. Resonance on the K2600 is implemented in one of two ways. On some filters, the resonance is fixed, adding or subtracting a specific amount of dB to the affected frequencies (the ones near the cutoff). On other filters, you can control the amount of resonance applied. In the case of these filters, there will always be a separate control page for the resonance.
- Separation** Four of the filters in the K2600 (both Four-Pole filters, the Double Notch, and the Twin Peaks) are actually two filters combined into one DSP function. For these filters, you will find a control page called Separation. This allows you to shift the cutoff frequency of the second filter, creating a separation in the cutoff frequencies of the two filters. In the case of the Notch and Band Pass filters, this can be used to create two separate notches or band passes. In the case of the four-pole filters, it affects the shape of the roll off. For the four-pole filters, separation set to 0 creates sharp rolloff of 24dB per octave.

How to Read the Graphs

The graphs show the rolloff curve, using several different values to show how they change the shape of the curve. Amplitude is always on the vertical axis. Frequency is always on the horizontal axis. You will notice on several graphs that the curve becomes more dramatic as the cutoff frequency is set at a higher value. This is because the highest frequency the K2600 can produce is 20Khz, so as the cutoff is set to higher values, there are fewer frequencies available before it is past the range of the K2600.

One-pole Lowpass Filter (LOPASS)



Frequencies below the cutoff frequency are unaffected by this filter. At the cutoff frequency, the signal is attenuated 3 dB. There’s a rolloff of 6 dB per octave above the cutoff frequency—that is, the signal is attenuated 6 dB with each octave above the cutoff. The resonance—the amount of cut or boost at the cutoff frequency—is fixed at -3dB. When the cutoff frequency is well below the lowest-frequency partials of a sound, lowering the cutoff further will not affect the timbre of the sound, but will reduce its overall amplitude.

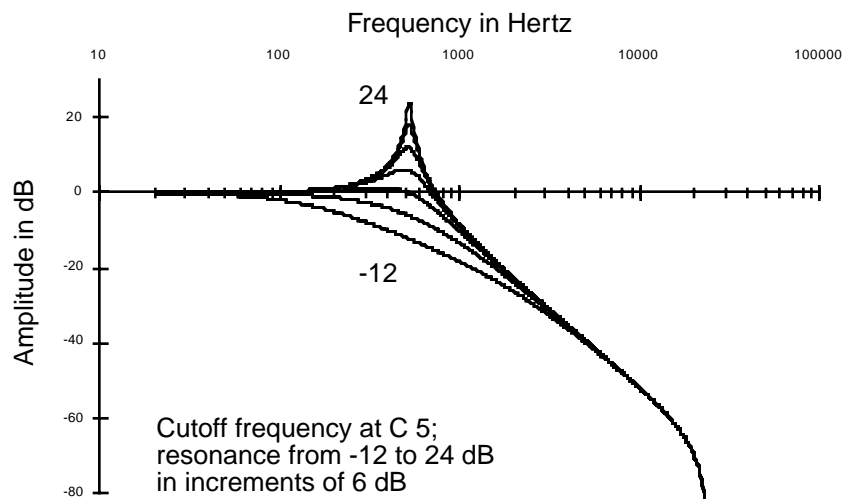
```

EditProg:F1 FRQ(LOPASS) <>Layer:1/1
Coarse:C 4 262 Hz Src1 :OFF
Fine :0ct Depth :0ct
KeyTrk:0ct/key Src2 :OFF
VelTrk:0ct DptCt1:OFF
Pad :0dB MinDpt:0ct
MaxDpt:0ct
<more F1 FRQ F2 F3 F4 AMP more>
    
```

Parameter	Range of Values
Coarse Adjust	C 0 16 Hz to G 10 25088 Hz
Fine Adjust	± 100 cents
Key Tracking	± 250 cents per key
Velocity Tracking	± 10800 cents
Pad	0, 6, 12, 18 dB
Source 1	Control Source list
Source 1 Depth	± 10800 cents
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 10800 cents
Maximum Depth, Source 2	± 10800 cents

The Coarse Adjust parameter sets the cutoff frequency in terms of a key name. The remaining parameters (except Pad) alter the cutoff frequency in increments of cents. You'll notice that positive values for key tracking have an interesting effect on the function of lowpass filters; positive key tracking values raise the cutoff frequency for high notes and lower it for low notes. More specifically, a value of **100 cents per key** on this page, when filtering a constant waveform like a sawtooth, would result in waveforms of exactly the same shape for all pitches of the waveform. The cutoff frequency moves in sync with the frequencies of the waveform's partials as different pitches are generated. Negative key tracking values will steepen the rolloff of lowpass filters above the cutoff. The Pad parameter, as always, attenuates the signal at the input to the function. These parameters affect all the lowpass filters similarly.

Two-pole Lowpass Filter (2POLE LOWPASS)



The two-pole lowpass filter has a rolloff of 12 dB per octave above the cutoff frequency. This is a two-stage function, so it has two control-input pages. The first affects the cutoff frequency, and has the same parameters as the one-pole lowpass. The second control-input page (F2 RES) affects the resonance of the filter. Resonance is a cut or boost in amplitude of the partials in the vicinity of the cutoff frequency.

Set the resonance with the Adjust parameter on the F2 RES control-input page, and use the other parameters to assign various controls to alter it. If a boost is applied at frequencies where there

DSP Functions

The DSP Functions

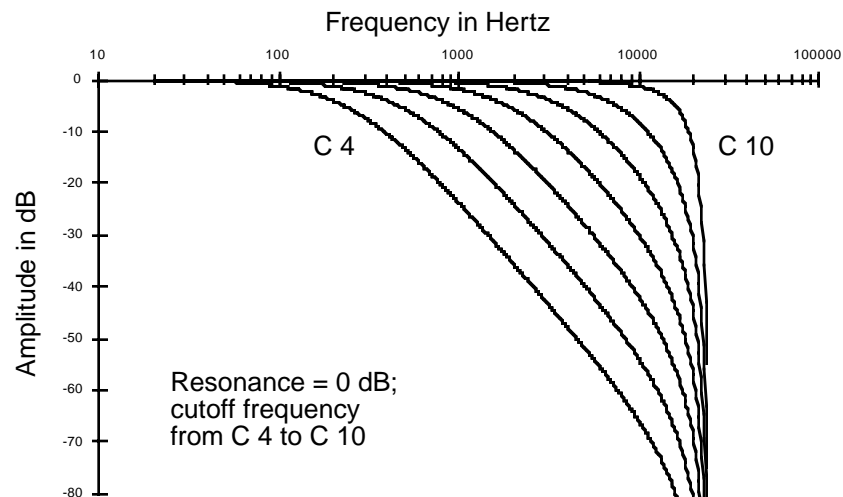
are high-amplitude partials, the signal may clip. The Pad parameter on the F1 FRQ page will reduce the clipping, but there's no harm in keeping it if you like the sound.

```

EditProg:F2 RES(2P LOPAS2) <> Layer:1/1
Adjust:0.0dB Src1 :OFF
Depth : 0.0dB
Src2 :OFF
KeyTrk: 0.00dB/key DptCtl:OFF
VelTrk: 0.0dB MinDpt: 0.0dB
MaxDpt: 0.0dB
<more F1 FRQ F2 RES F3 F4 AMP more>
  
```

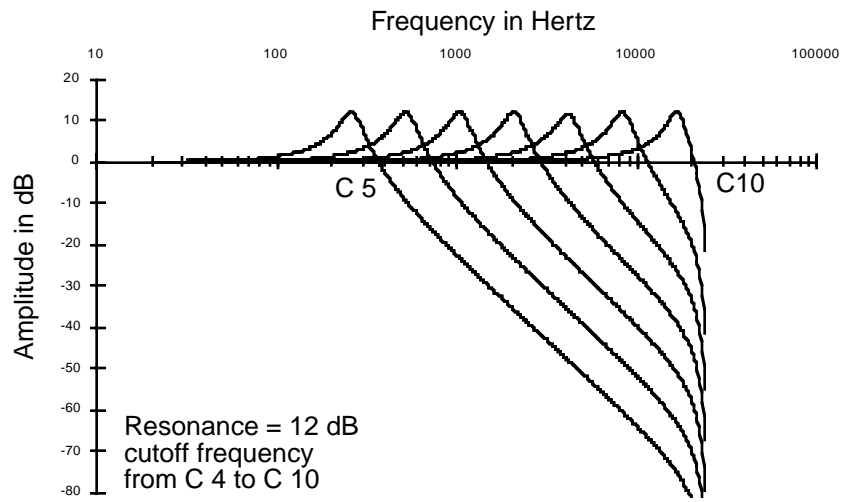
Parameter	Range of Values
Adjust	-12 to +48 dB
Key Tracking	± 2.00 dB per key
Velocity Tracking	-30 to +60 dB
Source 1	Control Source list
Source 1 Depth	-30 to +60 dB
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	-30 to +60 dB
Maximum Depth, Source 2	-30 to +60 dB

Two-pole Lowpass Filter, -6 dB Resonance (LOPAS2)



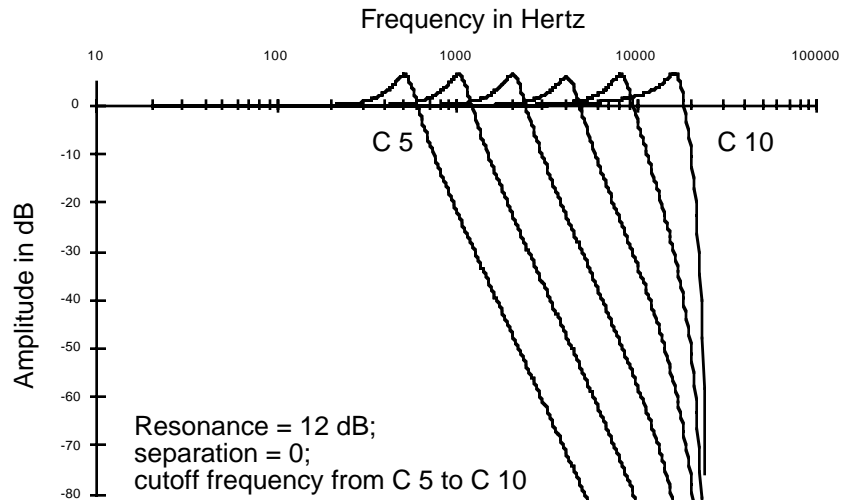
Using this filter is the same as using two one-pole lowpass filters in successive algorithm blocks. Since its resonance is fixed at -6 dB, it's also the same as using 2POLE LOWPASS with the resonance set to -6 dB. You'd use this filter when you want a 12 dB per octave rolloff but don't need to set a resonance level. This would leave you free to use another DSP function in the algorithm, since it's a one-stage function.

Two-pole Lowpass Filter, +12 dB Resonance (LP2RES)



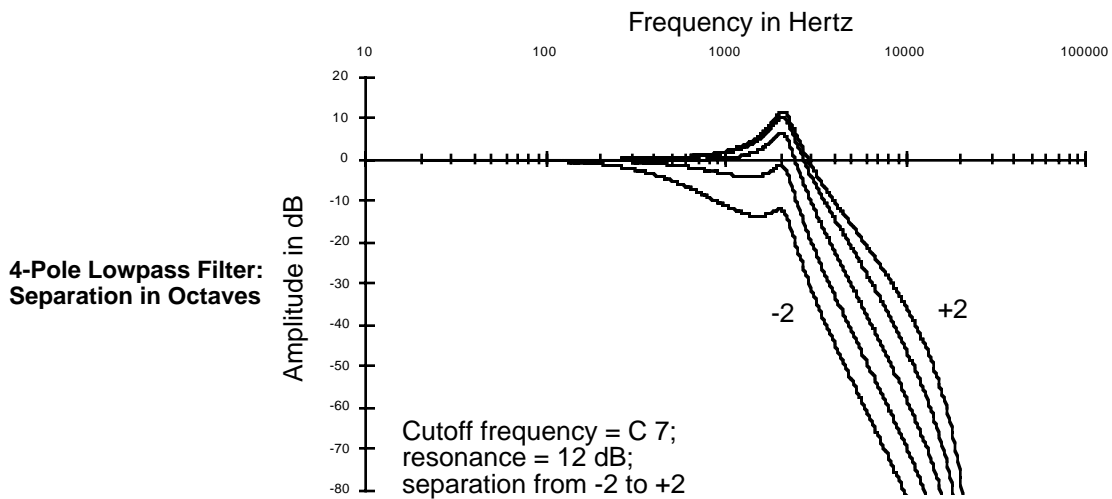
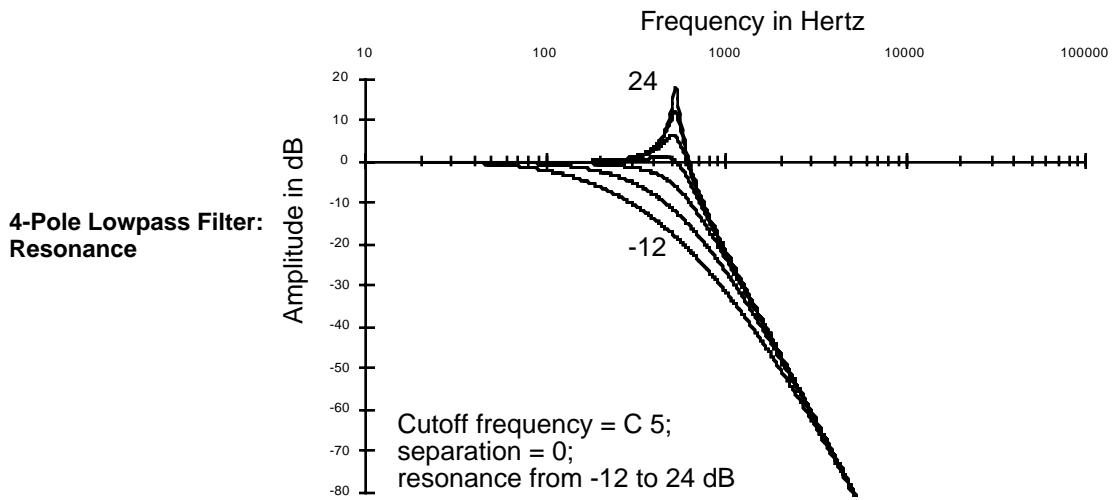
This is similar to LOPAS2; the only difference is that its resonance is fixed at +12 dB.

Four-pole Lowpass Filter with Separation (4POLE LOPASS W/ SEP)



DSP Functions

The DSP Functions



This combines 2POLE LOWPASS and LOPAS2 in one three-stage function. The parameters on the F1 FRQ control-input page affect the cutoff frequencies of both filters. The parameters on the F2 RES page affect the resonance of 2POLE LOWPASS. The parameters on the F3 SEP page shift the cutoff frequency of LOPAS2, creating a separation between the cutoff frequencies of the two

filters. Positive values raise the cutoff frequency of LOPAS2, while negative values lower it. If no separation is applied, there's a 24 dB per octave rolloff above the cutoff frequency.

```

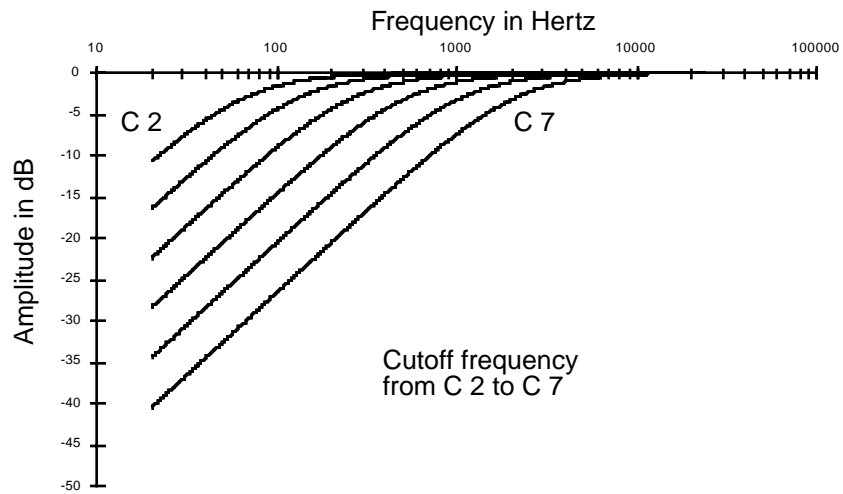
EditProg:F3 SEP(4F LOPASS) <>Layer:1/1
HdJust:0ct Src1 :OFF
Fine :0ct Depth :0ct
KeyTrk:0ct/key DptCt1:OFF
VelTrk:0ct MinDpt:0ct
MaxDpt:0ct
<more F1 FRQ F2 RES F3 SEP F4 AMP more>
    
```

Parameter	Range of Values
Coarse Adjust	± 10800 cents
Fine Adjust	± 100 cents
Key Tracking	± 250 cents per key
Velocity Tracking	± 10800 cents
Source 1	Control Source list
Source 1 Depth	± 10800 cents
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 10800 cents
Maximum Depth, Source 2	± 10800 cents

Gated Lowpass Filter (LPGATE)

You may be familiar with gates as applied to effects like reverb, where the effect shuts off abruptly after a specified time. The gated lowpass filter produces a somewhat similar effect in terms of the sound's amplitude. The filter's cutoff frequency is controlled by the AMPENV. When the AMPENV is at 100%, the cutoff frequency is high, so most of the partials are heard. When the AMPENV decays or releases to 0%, the cutoff frequency is low, so only the lowest partials are heard. You'll hear the distinct effect of the filter closing as the amplitude envelope releases.

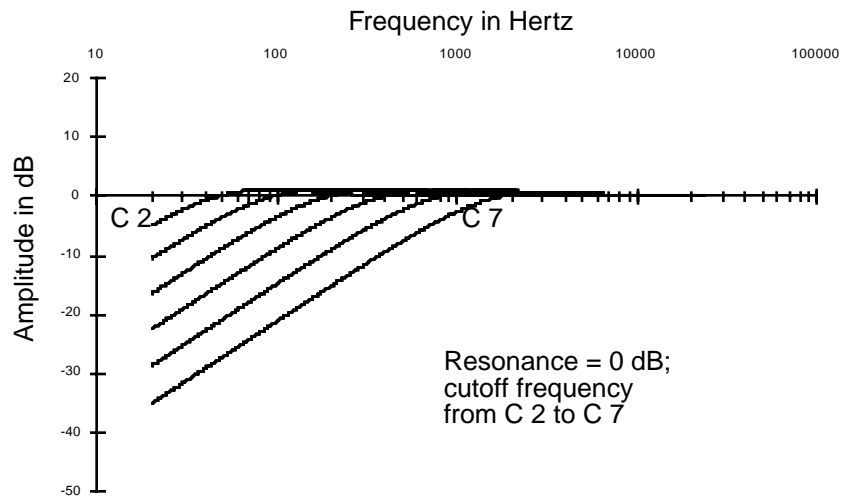
One-pole Highpass Filter (HIPASS)



High-frequency partials pass through this filter unaffected. At the cutoff frequency, the signal is attenuated 3 dB. There's a roll-off of 6 dB per octave below the cutoff frequency. The resonance is fixed at -3dB. When the cutoff frequency is well above the highest-frequency partials of a sound, raising the cutoff further will not affect the timbre of the sound, but will merely attenuate it further.

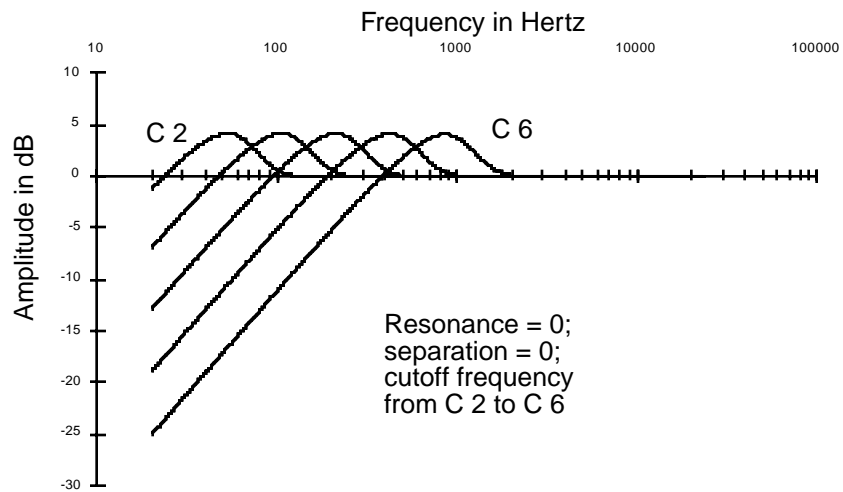
The Coarse Adjust parameter sets the cutoff frequency in terms of a key name. The remaining parameters (except Pad) alter the cutoff frequency in increments of cents. Positive key tracking values raise the cutoff frequency for high notes and lower it for low notes. More specifically, a value of 100 cents per key on this page, when filtering a constant waveform like a sawtooth, would result in waveforms of exactly the same shape for all pitches of the waveform. The cutoff frequency moves in sync with the frequencies of the waveform's partials as different pitches are generated. Negative key tracking values will steepen the rolloff of highpass filters below the cutoff. The Pad parameter, as always, attenuates the signal at the input to the function. These parameters affect all the highpass filters similarly.

Two-pole Highpass Filter (HIPAS2)



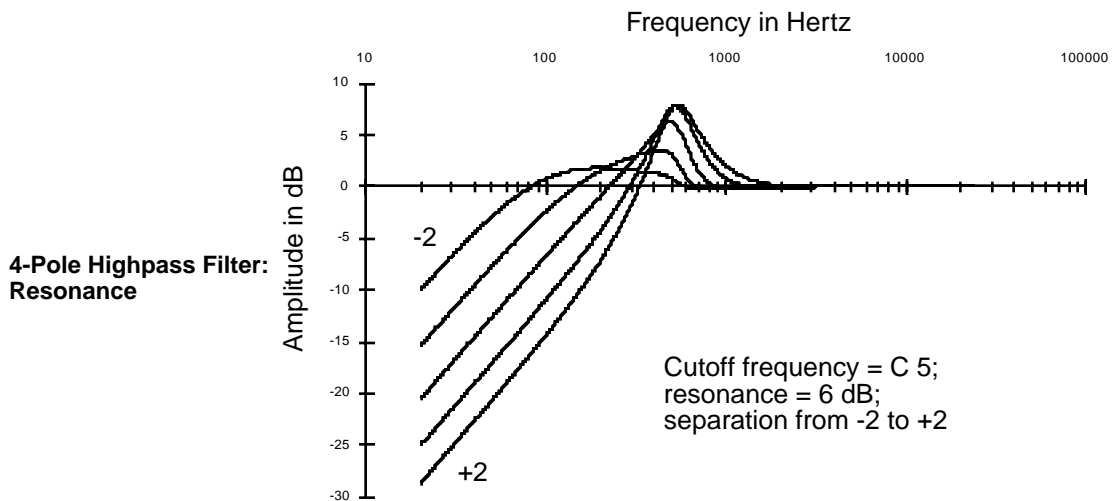
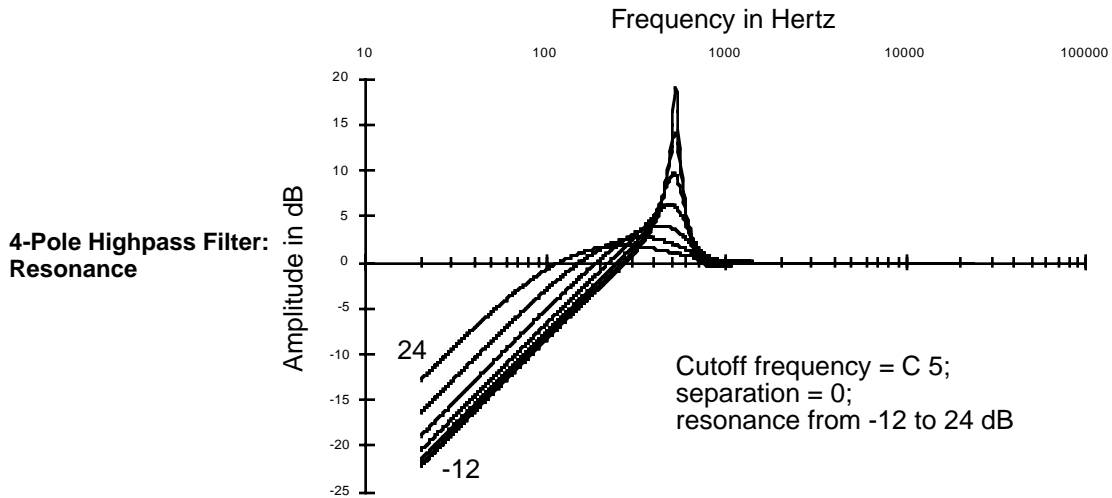
This is very similar to HIPASS. The primary difference is in the steepness of the rolloff at the cutoff frequency. Below the cutoff frequency, the rolloff is similar to that of HIPASS, except that there's a one-octave shift—that is, HIPASS with a cutoff frequency of C 3 will sound nearly the same as HIPAS2 with a cutoff of C 4. In other words, HIPASS gives you greater attenuation of low frequencies when set to the same cutoff frequency as HIPAS2.

Four-pole Highpass Filter with Separation (4POLE HIPASS W/ SEP)



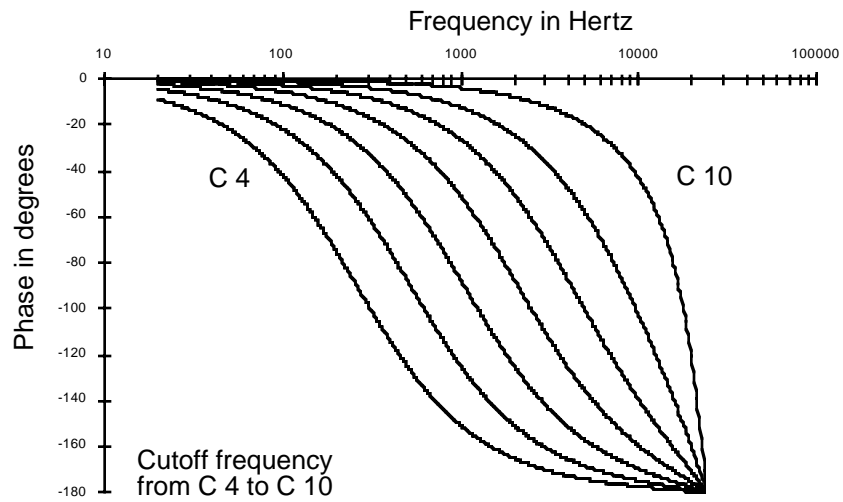
DSP Functions

The DSP Functions



This combines two of the 2POLE HIPASS filters into one three-stage function. It has a rolloff of 6 dB per octave below the cutoff frequency. The parameters on the F1 FRQ control-input page affect the cutoff frequencies of both filters. The parameters on the F2 RES page affect the resonances of the first filter. There will always be a slight extra boost of partials at the cutoff frequency, even at low resonance settings. The parameters on the F3 SEP page shift the cutoff frequency of the second 2POLE HIPASS, creating a separation between the cutoff frequencies of the two filters. Positive values raise the cutoff frequency of the second 2POLE HIPASS, while negative values lower it. If no separation is applied, there's a 24 dB per octave rolloff below the cutoff frequency. A variety of responses can be produced by adjusting the resonance and separation settings.

One-pole Allpass Filter (ALPASS)

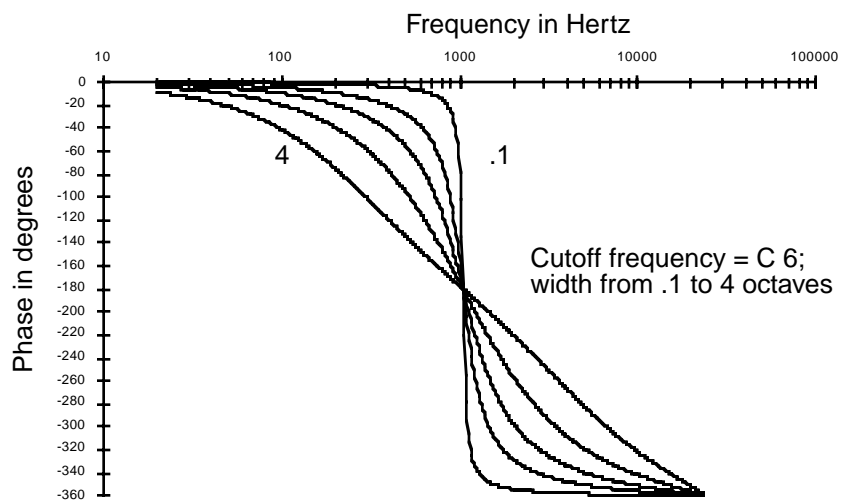
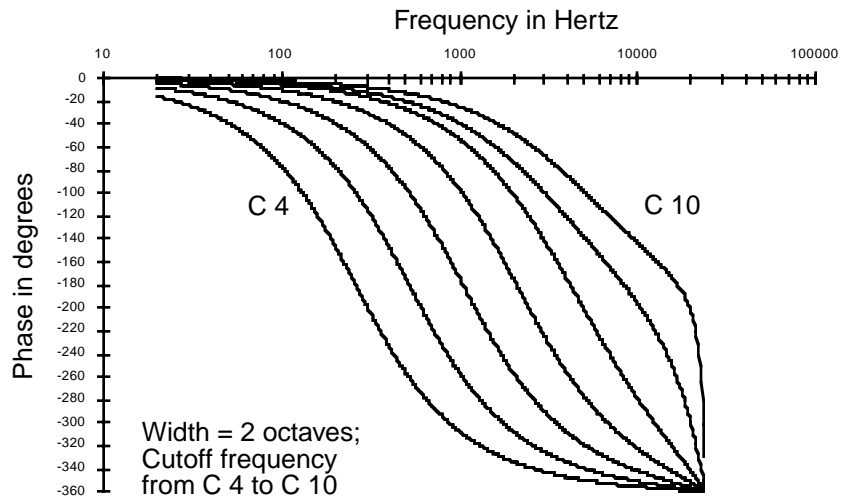


Allpass filters do not affect a sound’s frequency response (the amplitude of partials at various frequencies), but change the phase of each partial depending on its proximity to the center frequency. The phase shift is -90 degrees for partials at the center frequency. It rises toward 0 degrees for partials at frequencies below the center, and falls toward -180 degrees for partials at frequencies above the center. With low-frequency waveforms, you’ll be able to hear this phase shift. As a rule, however, the ear is not sensitive to phase shifts unless they’re changing, so you’ll usually want to use Source 1 or 2, and assign an LFO to sweep the center frequency up and down.

Periodic phase shifts like those induced by an LFO on the center frequency will cause a vibrato-like variation in the pitch of a sine wave input. This vibrato effect will be less regular for more complex partials. The greater the depth setting of the control source using the LFO, the greater the vibrato effect.

The amount of vibrato effect also depends on the speed and amount of the phase shift. Try adjusting the rate of the LFO controlling the center frequency. Another way to increase the amount of phase shift is to use the two-pole allpass filter, or to use the one-pole allpass filter in more than one algorithm block.

Two-pole Allpass Filter (2POLE ALLPASS)



Using 2POLE ALLPASS is very similar to using ALPASS, with two differences. First, the phase shift is -180 degrees for partials at the center frequency, approaching 0 degrees for partials at low frequencies, and approaching -360 degrees for partials at high frequencies.

Second, since this is a two-stage function, there's an additional control-input page (F2 WID) which controls the filter width. The parameters on this page affect the frequency range, measured in octaves, where most of the phase shifting occurs. Small values cause a drop from 0 to -360 in the phase shift to occur near the center frequency, while large values spread the drop in the phase shift over a broader frequency range. Small values tend to affect just a few partials, leaving others mostly untouched. The affected partials seem to become detached from the others, creating the illusion of an additional sound source.

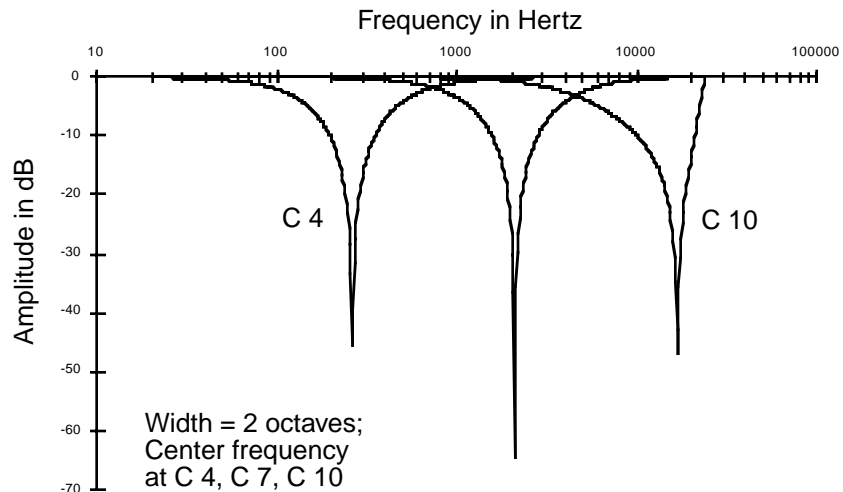
If you leave the center frequency constant and assign an LFO to vary the width, partials with frequencies above the center will shift their pitches in the opposite direction of partials below the center frequency.

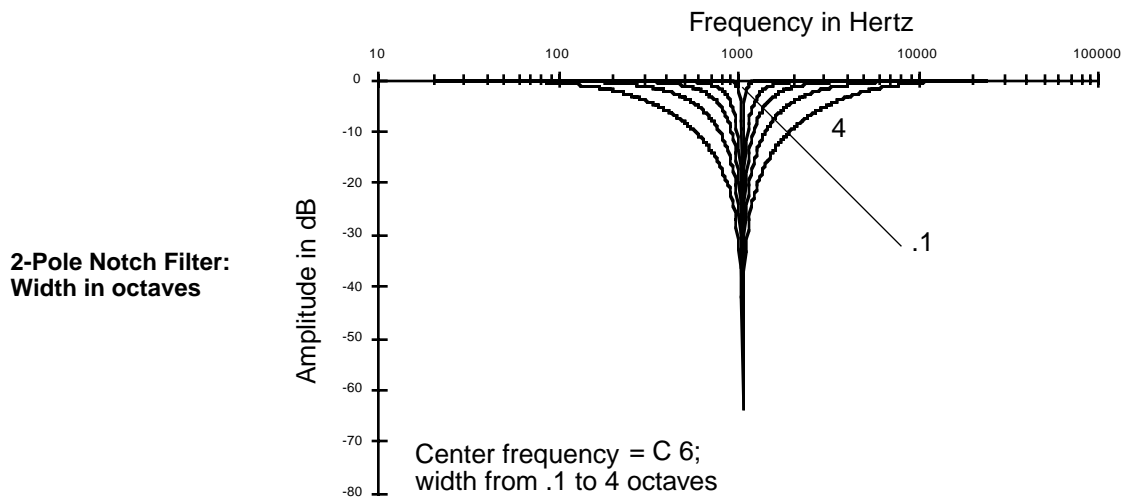
```

EditProg:F2 WID(2P HPASS) <> Layer:1/1
Adjust:0.010oct Src1 :OFF
Depth :0.00oct
Src2 :OFF
KeyTrk:0.000oct/key DptCt1:OFF
VelTrk:0.000oct MinDpt:0.000oct
MaxDpt:0.000oct
<more F1 FRC F2 WID F3 AMP F4 AMP more>
    
```

Parameter	Range of Values
Adjust	0.010 to 5.000 octaves
Key Tracking	± .200 octaves per key
Velocity Tracking	± 5.00 octaves
Source 1	Control Source list
Source 1 Depth	± 5.00 octaves
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 5.00 octaves
Maximum Depth, Source 2	± 5.00 octaves

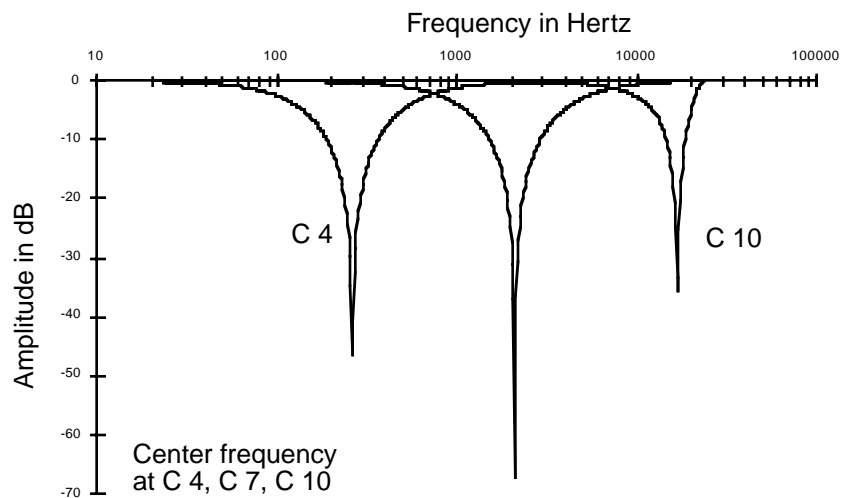
Two-pole Notch Filter (NOTCH FILTER)





The two-pole notch filter has two control-input pages, one for center frequency, one for width. Partials with frequencies above or below the notch will be unaffected. Within the notch, partials will be attenuated according to the width of the notch. The width is defined in terms of the number of octaves between the points on the signal's attenuation curve where the attenuation is 3 dB (see the explanation of F2 WID for the PARAMETRIC EQ function—page 16-26). For example, if the width is set at four octaves, then the attenuation will be 3 dB at two octaves in either direction from the center frequency. There's no attenuation of partials at more than two octaves in either direction from the center frequency.

Two-pole Notch Filter, Fixed Width (NOTCH2)



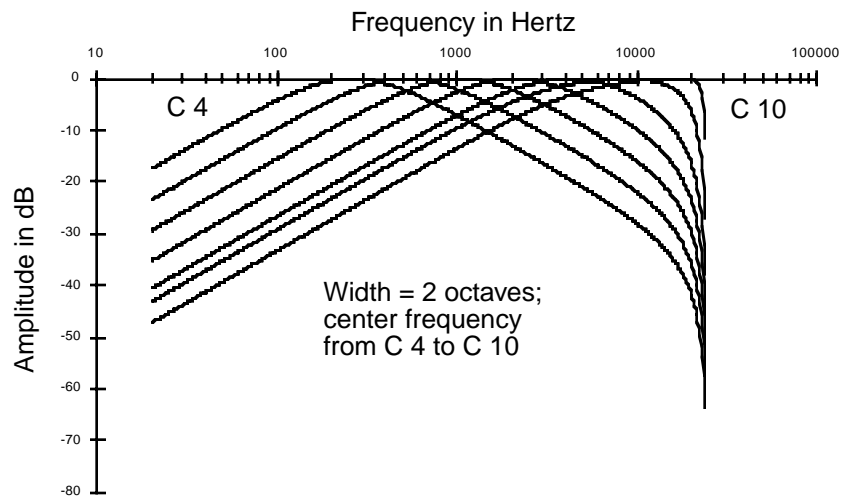
The only functional difference between NOTCH2 and NOTCH FILTER is that the width of NOTCH2 is fixed at 2.2 octaves. This gives you a one-stage notch filter function.

Two-pole Bandpass Filter (BANDPASS FILTER)

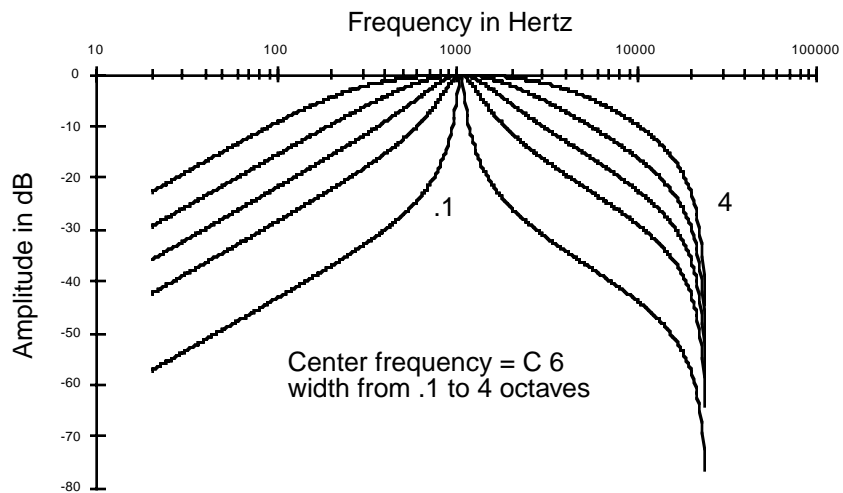
This is essentially the opposite of a notch filter; it passes all partials at the center frequency, and cut the levels of partials above or below the center frequency. The width is defined the same as for the double notch filter.

The gain at the center frequency is 0 dB. Small values for width (a narrow bandpass) may produce a very quiet signal unless the center frequency matches the frequency of a strong sine wave partial. Wide bandpasses may result in a quiet signal if they're centered in a region of the sound where the partials are weak. You can easily boost these quiet signals with the parameters on the F4 AMP page.

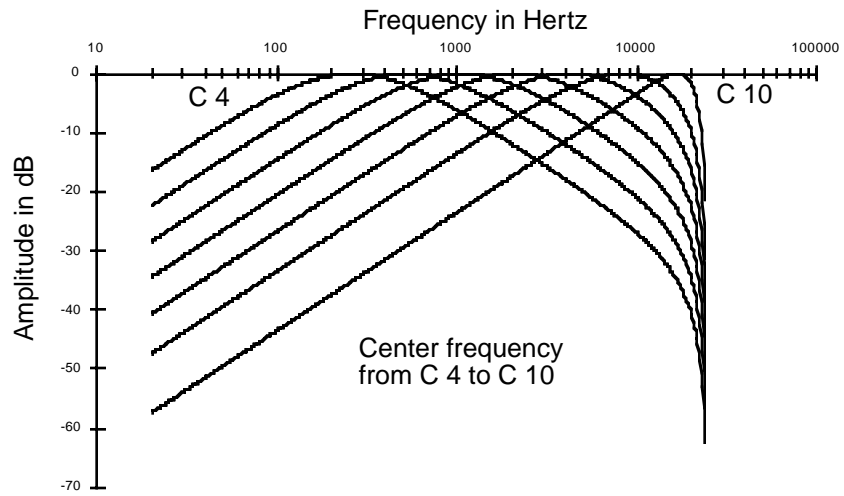
**Bandpass Filter:
Frequency**



**Bandpass Filter:
Width in octaves**

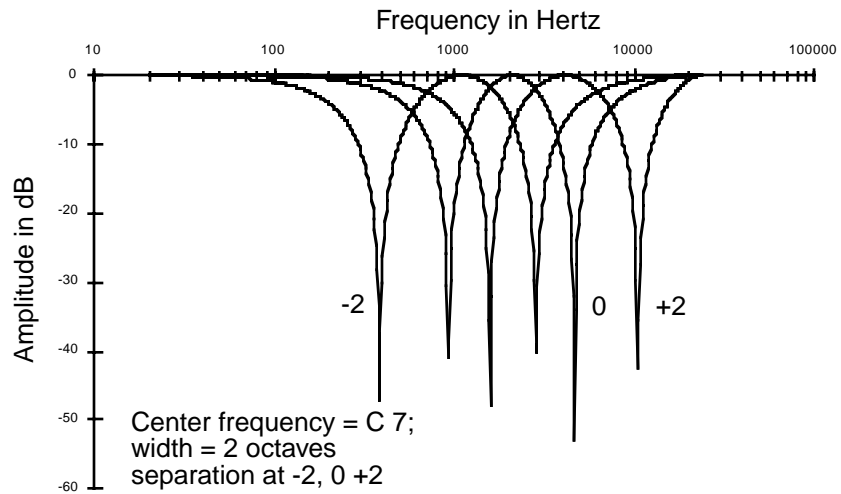


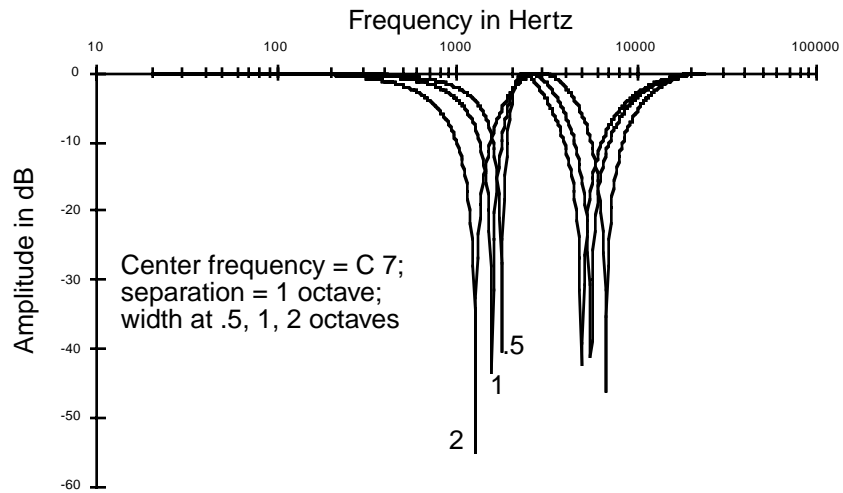
Two-pole Bandpass Filter, Fixed Width (BAND2)



The only functional difference between BAND2 and BANDPASS FILTER is that the width of BAND2 is fixed at 2.2 octaves. This gives you a one-stage bandpass filter function.

Double Notch Filter with Separation (DOUBLE NOTCH W/ SEP)





This is a three-stage function that puts two notches in the frequency response. As with NOTCH FILTER and NOTCH2, there are control-input pages for frequency and width. A third control-input page affects the separation of the notches.

Setting the center frequency on the F1 FRQ page defines the frequency that's halfway between the notches. The settings for the separation affect the behavior of the width control parameters. When the separation is 0, the notches are close to the center frequency, and the width control parameters control the widths of both notches equally. Positive values for separation move the notches apart, and cause the width control parameters to affect the width of the higher-frequency notch more than the width of the lower-frequency notch. Negative values for separation will move the notches apart to the same extent, but will cause the width control parameters to affect the width of the lower-frequency notch more than the width of the higher-frequency notch.

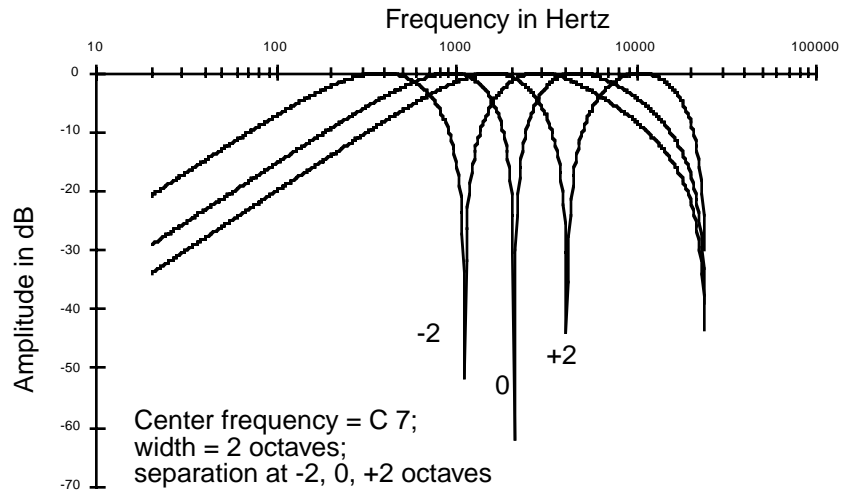
Twin Peaks Bandpass Filter (TWIN PEAKS BANDPASS)

The control parameters for TWIN PEAKS BANDPASS work the same way as for DOUBLE NOTCH FILTER, but of course, you get peaks instead of notches—that is, the amplitudes of partials near the center frequency are high, and the amplitudes are increasingly attenuated at frequencies farther from the center.

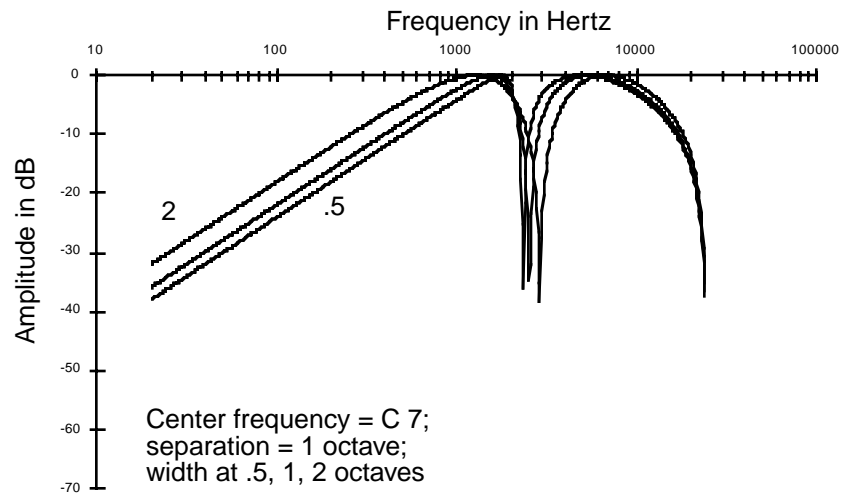
DSP Functions

The DSP Functions

**Twin Peaks
Bandpass Filter:
Separation**



**Twin Peaks
Bandpass Filter:
Width**



Equalization (EQ)

Equalization is a specialized filtering process that lets you boost or cut the amplitude of a specified range of frequencies.

Parametric EQ

Treble Tone Control

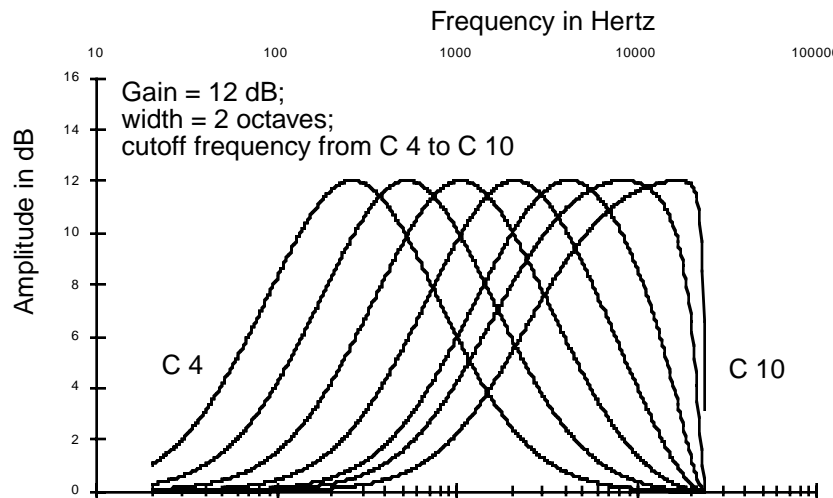
Mid-range Parametric EQ

Steep Bass Tone Control

Bass Tone Control

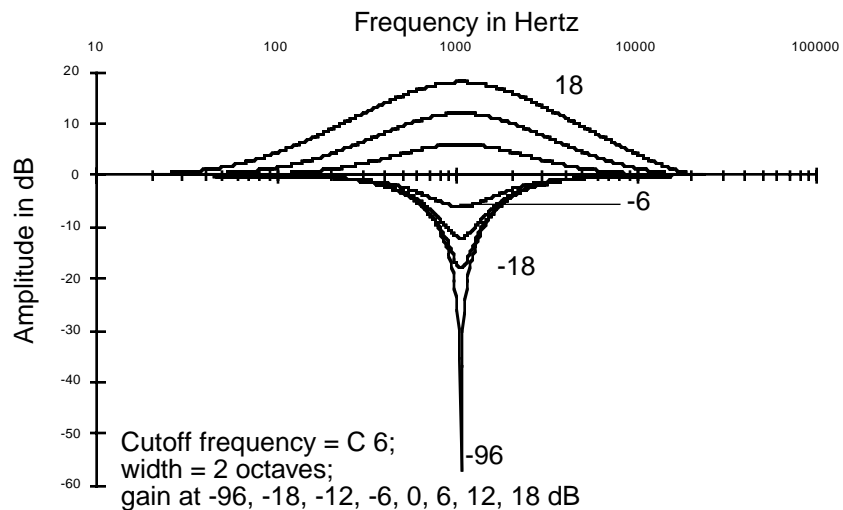
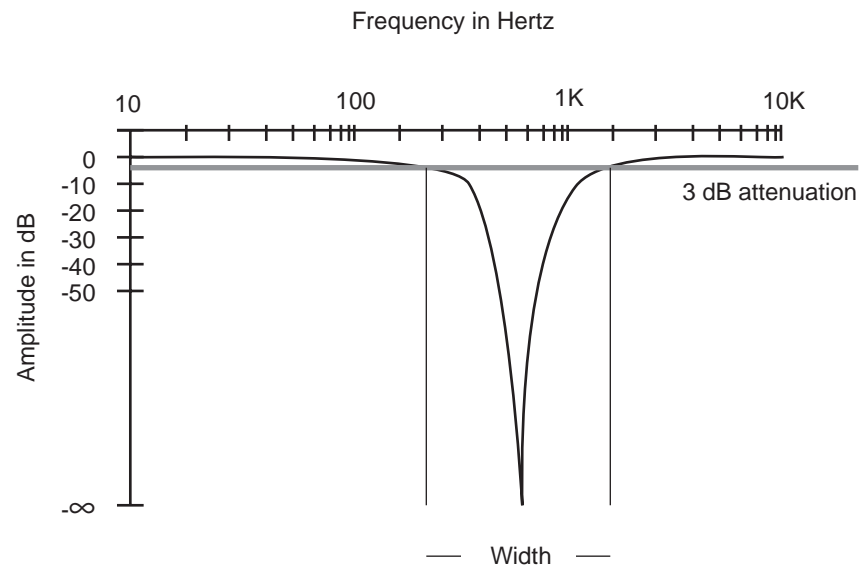
Parametric Equalizer (PARAMETRIC EQ)

This function has three interacting parameters, each with its own control-input page: center frequency, width, and amplitude. The center frequency is the center of the range of frequencies that will be boosted or cut by the amplitude setting. The width is the entire range of frequencies that will be affected by the amplitude setting. For the K2600, the width is defined by imagining an amplitude curve with a level (in dB) of -infinity (minus infinity) at the center frequency, then measuring the distance (in octaves) between the points on the curve where the amplitude is attenuated by 3dB. See the diagram below.



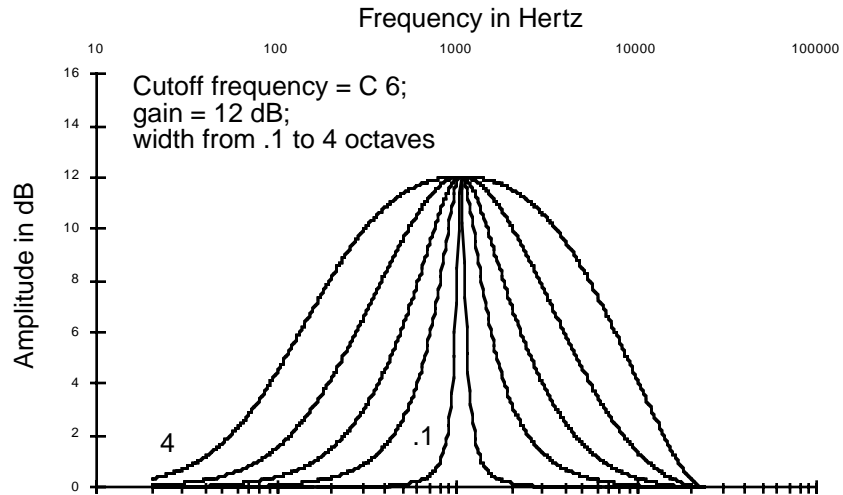
DSP Functions

The DSP Functions



When you're using the Parametric EQ, you might use the following sequence. Set the center frequency (press the **F1 FRQ** soft button to select its control-input page). The frequency is measured in terms of each note of the keyboard. The frequency in Hertz of each note appears with the note name as the value for the Adjust parameter. Next, select the width control-input page (the **F2 WID** soft button) to determine how wide a range will be affected by the amplitude adjustment. Then select the amplitude control-input page (**F3 AMP** soft button), and adjust the amplitude of the range you specified with the center frequency and width settings. You'll

probably jump back and forth between these three pages until your ear is satisfied with the sound.



```

EditProg:F1 FRQ(PARH EQ) <>Layer:1/1
Adjst:C 4 262Hz Src1 :OFF
Fine :Oct Depth :Oct
KeyTrk:Oct/key Src2 :OFF
VelTrk:Oct DptCtl:OFF
Pad :0dB MinDpt:Oct
MaxDpt:Oct
<more F1 FRQ F2 WII F3 AMP F4 AMP more>
    
```

Parameter	Range of Values
Coarse Adjust	C 0 16 Hz to G 10 25088 Hz
Fine Adjust	± 100 cents
Key Tracking	± 250 cents per key
Velocity Tracking	± 10800 cents (9 octaves)
Pad	0, 6, 12, 18 dB
Source 1	Control Source list
Source 1 Depth	± 96 dB
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 10800 cents
Maximum Depth, Source 2	± 10800 cents

DSP Functions

The DSP Functions

The Fine Adjust parameter gives you one-cent precision in setting the center frequency.

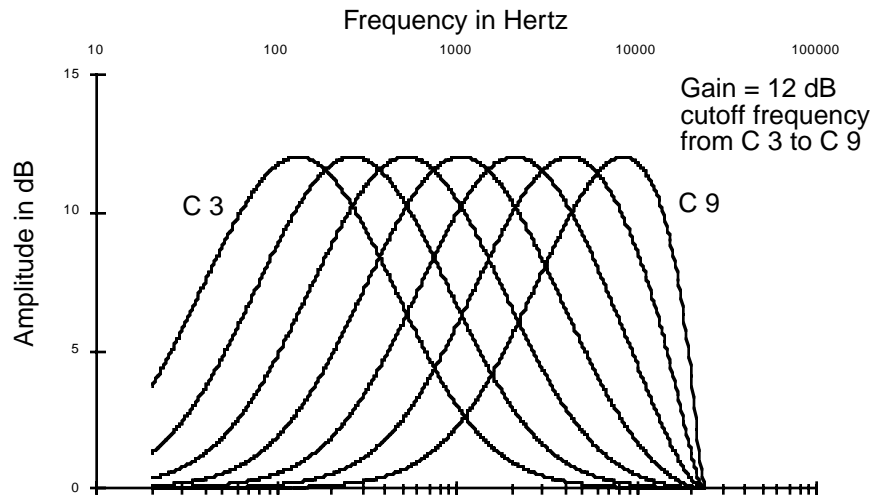
```
editProg:F2 WII(PARA EQ) <>Layer:1/1
Adjust:0.010oct Src1 :OFF
Depth :0.00oct
Src2 :OFF
KeyTrk:0.000oct/key DptCtl:OFF
VelTrk:0.00oct MinDpt:0.00oct
MaxDpt:0.00oct
<more F1 FRC F2 WII F3 AMP F4 AMP more>
```

Parameter	Range of Values
Adjust	0.010 to 5.000 octaves
Key Tracking	± .200 octaves per key
Velocity Tracking	± 5.00 octaves
Source 1	Control Source list
Source 1 Depth	± 5.00 octaves
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 5.00 octaves
Maximum Depth, Source 2	± 5.00 octaves

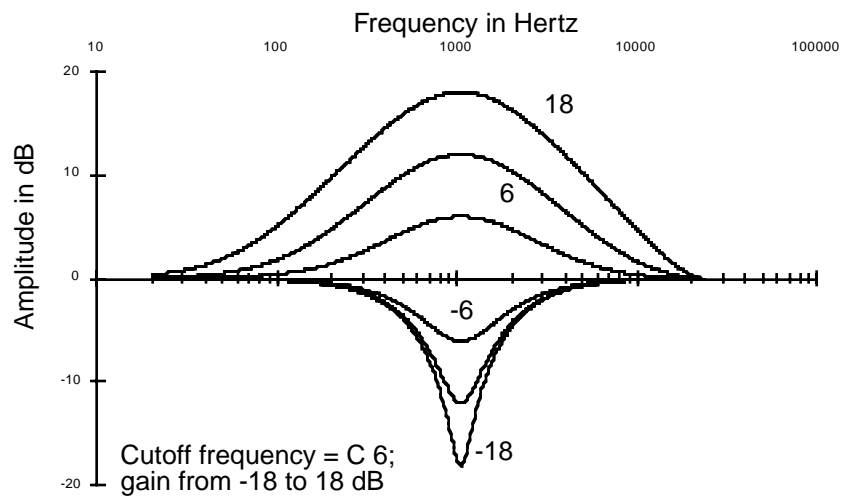
The control-input page for the amplitude stage (F3 AMP) is identical to the AMP page described previously, except that there's no Pad parameter.

Mid-range Parametric EQ (PARA MID)

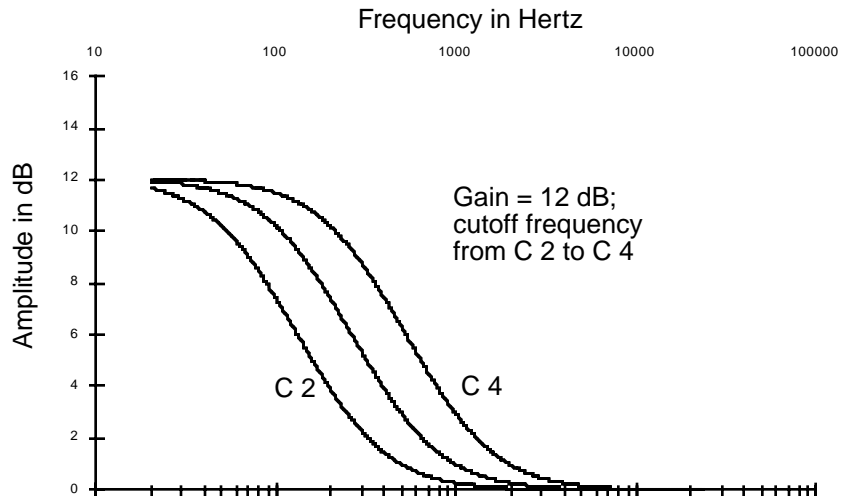
This two-stage function is almost identical to the three-stage Parametric EQ function. The only difference is that the width of PARA MID is fixed at 2.2 octaves. Consequently there's no control-input page for the width.



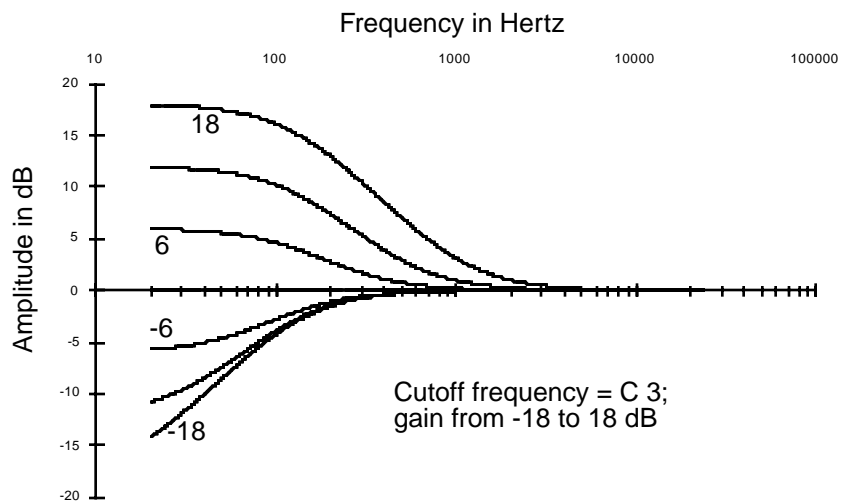
Para Mid: Gain



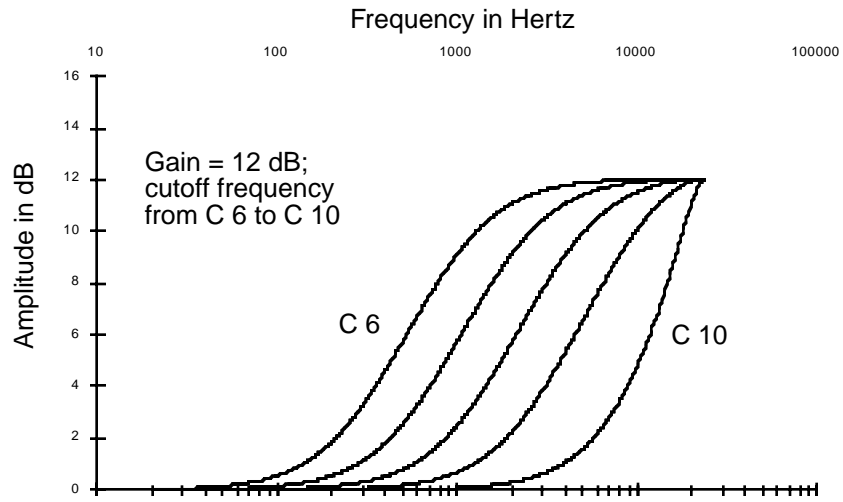
Bass Tone Control (PARA BASS)



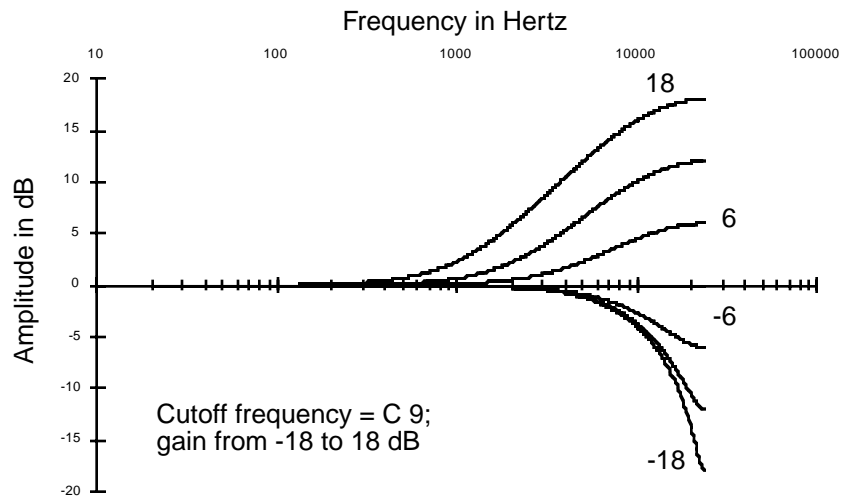
This is a two-stage function, with control-input pages for frequency and amplitude. These pages are the same as those for frequency and amplitude in PARA EQ. On the frequency control-input page, you'll set the cutoff frequency. For notes above this frequency, the amplitude setting has a diminished effect. On the amplitude control-input page, you'll set the amount of cut or boost that's applied to notes below the cutoff frequency. There's a gradual increase in the bass response for each successively lower note. The location of the cutoff frequency will change somewhat as you change the amplitude settings, although the value for the Adjust parameter on the frequency control-input page will not reflect the change.



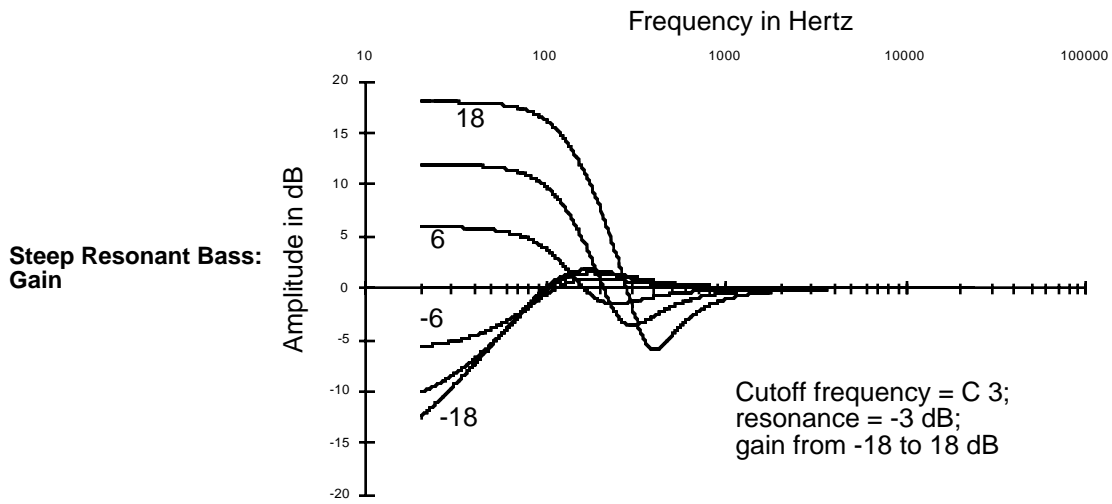
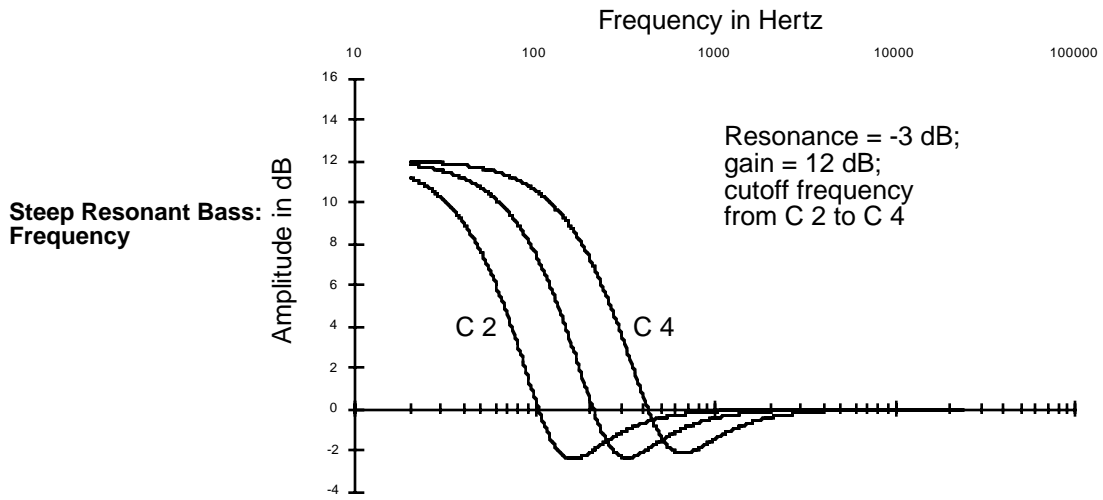
Treble Tone Control (PARA TREBLE)

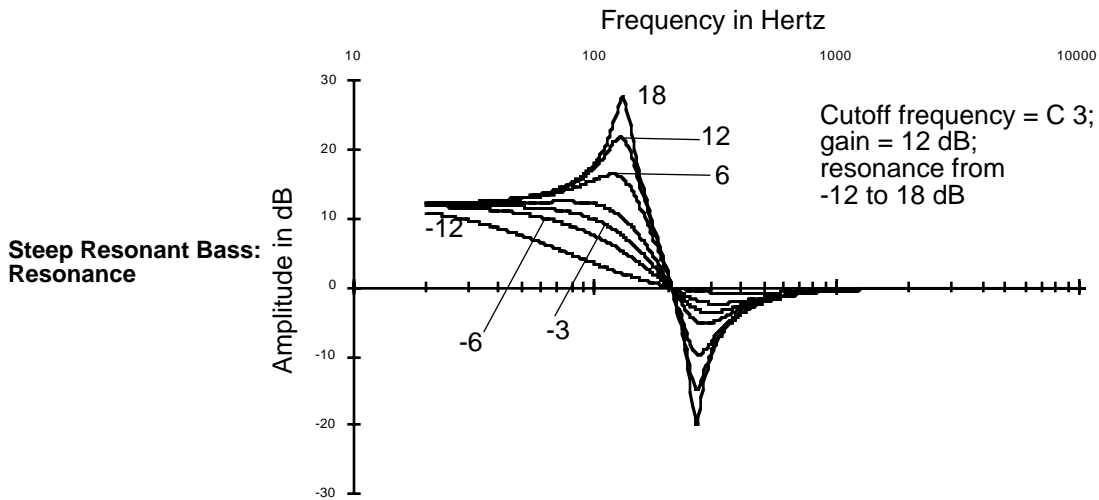


PARA TREBLE is very similar to PARA BASS; the only difference is that the amplitude setting affects notes *above* the cutoff frequency.



Steep Bass Tone Control (STEEP RESONANT BASS)





This function uses a two-pole lowpass filter to give you a sharper transition in bass response than PARA BASS. Like PARA BASS, there are control-input pages for cutoff frequency and amplitude, which are identical to those for PARA BASS. There is also a control-input page for resonance (also known as “q”), which can boost or cut the amplitude of the partials near the cutoff frequency.

```

EditProgHF2 RES(S) DE: BASS(X) Layer:1/1
Hdjust:0dB Src1 :OFF
Depth :0.0dB
Src2 :OFF
KeyTrk: 0.00dB/key DptCt1:OFF
VelTrk: 0.0dB MinDpt:0.0dB
MaxDpt:0.0dB
<more F1 FRQ F2 RES F3 AMP F4 AMP more>
    
```

Parameter	Range of Values
Adjust	-12 to 24 dB
Key Tracking	± 2.00 dB per key
Velocity Tracking	± 30 dB
Source 1	Control Source list
Source 1 Depth	± 96 dB
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 30 dB
Maximum Depth, Source 2	± 30 dB

You’ll get the best transition in bass response with a resonance setting of -6 dB. There’s a small frequency range above the cutoff frequency where the response reverses direction (if you’re cutting the amplitude, for example, you’ll get a slight boost just above the cutoff frequency). The higher you set the resonance, the larger this reversal will be, resulting in unusual—but possibly useful—response curves at high resonance values.

Pitch / Amplitude / Panner

PITCH	UPPER AND LOWER AMP
AMP	BALANCE AND AMP
PANNER	GAIN

PITCH

We used the PITCH control-input page as an example to introduce the common DSP control parameters in Chapter 6 (*Common DSP Control Parameters* on page 6-14), so we won't add much here. The PITCH function modifies the pitch of the layer's keymap as it passes through the sound engine. The PITCH stage of each algorithm is always the first stage. Algorithms 26–31, the Sync algorithms, don't show the PITCH stage on the ALG page, since these algorithms generate their own sawtooth waves, and do not use keymaps.

AMP

The AMP function is the final stage in every single-output algorithm, and controls the overall amplitude (volume) of the layer. This is an easy way to boost the signal to a more desirable level if it's not loud enough for your purposes. Large values for the Adjust parameter can cause a sound to clip, which will distort most sounds considerably. You may want this effect, and using it won't damage anything, but as a rule, you'll want to avoid clipping your sounds with the AMP (or GAIN) function. There are many other ways to distort your sounds, like DIST, SHAPER, and WRAP, to name a few.

The settings for the parameters on the F4 AMP page affect the gain level for the currently selected layer. So do the settings on the AMPENV page. Compare this with the effect of GAIN, described on page 16-37.

```

EditProg:F4 AMP(FINHL AMP) <>Layer:1/1
Adj:0dB Src1 :OFF
Depth :0dB
Src2 :OFF
KeyTrk: 0.00dB/key DptCt1:OFF
VelTrk:0dB MinDpt:0dB
Pad :0dB MaxDpt:0dB
<more F1 F2 F3 F4 AMP more>
    
```

Parameter	Range of Values
Adjust	-96 to 48 dB
Key Tracking	± 2.00 dB per key
Velocity Tracking	± 96 dB
Pad	0, 6, 12, 18 dB
Source 1	Control Source list
Source 1 Depth	± 96 dB
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 96 dB
Maximum Depth, Source 2	± 96 dB

PANNER

This single-stage function converts a single wire at its input into a double wire at its output, splitting the signal between an “upper” and “lower” wire. This creates a double-output algorithm, as discussed on page 6-31. The parameters on the PANNER page enable you to modify the signal’s routing through the upper and lower wires. By itself the PANNER doesn’t change the pan position of the sound. It just defines what percentage of the currently selected layer’s sound goes to each wire. When you select one of these double-output algorithms, the OUTPUT page for the layer changes to enable you to make pan settings for each wire independently. So when you use the PANNER function, you’ll also want to adjust the Pan parameters on the OUTPUT page, setting the upper wire’s pan fully right, and the lower wire’s pan fully left. This will enable you to hear the effect of the PANNER function.

The PANNER function is available only in algorithms 2, 13, 24, and 26, and always appears in the block before the final AMP function. Consequently, it will always be selected with the F3 soft button, which is labeled F3 POS (position).

```

EditProg:F3 POS(PANNER) <>Layer:1/1
Adjust:0% Src1 :OFF
Depth :0%
Src2 :OFF
KeyTrk: 0.0%key DptCtl:OFF
VelTrk:0% MinDpt:0%
Pad :0dB MaxDpt:0%
<more F1 F2 F3 POS F4 AMP more>
    
```

Parameter	Range of Values
Adjust	± 100%
Key Tracking	± 16% per key
Velocity Tracking	± 200%
Pad	0, 6, 12, 18 dB
Source 1	Control Source list
Source 1 Depth	± 200%
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 200%
Maximum Depth, Source 2	± 200%

The Adjust parameter sets the initial routing of the layer to the upper or lower wire. **-100%** is the lower, and **100%** is the upper. The KeyTrk parameter lets you shift the layer’s sound from one wire to the other based on the MIDI key number of each note. For positive values of KeyTrk, the higher above Middle C, the more sound goes to the upper wire.

The remaining parameters have ranges from **-200%** to **200%**. This lets you start with a sound that’s fully on the lower wire, for example, and shift it completely to the upper wire. The VelTrk parameter shifts notes between wires based on the attack velocity of each note. For positive values, the higher the attack velocity, the more sound goes to the upper wire. The Src1 and Src2 parameters let you assign controls to reroute the sound relative to the initial routing. Setting their depth parameters to positive values will shift the sound to the upper wire when the controls assigned to them approach their maximum values.

Upper and Lower Amp (AMP U AMP L)

This two-stage function is similar to the AMP function described above, but it appears in algorithms that have split the signal to two wires and has sent them through different DSP functions in the F2 and F3 blocks. This function enables you to set the final amplitude independently for each wire, and keeps the two signals separate at its output, giving you added flexibility for mixing and panning. Like the AMP function, UPPER AND LOWER AMP always appears as the last block in an algorithm. Since it's a two-stage function, it has two control-input pages. F3 selects the control-input page for the lower wire, and F4 the control-input page for the upper wire.

```

editProg: F3 AMP(AMP U/L) <> Layer: 1/1
Adjust: 0dB Src1 : OFF
Depth : 0dB
Src2 : OFF
KeyTrk: 0.00dB/key DptCtl: OFF
VelTrk: 0dB MinDpt: 0dB
Pad : 0dB MaxDpt: 0dB
<more F1 F2 F3 AMP F4 AMP more>
    
```

Parameter	Range of Values
Adjust	-96 to 48 dB
Key Tracking	± 2.00 dB per key
Velocity Tracking	± 96 dB
Pad	0, 6, 12, 18 dB
Source 1	Control Source list
Source 1 Depth	± 96 dB
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 96 dB
Maximum Depth, Source 2	± 96 dB

Balance and Amp (BAL AMP)

This function has a two-wire input and a two-wire output. The parameters on its control-input page affect the amount of gain applied to each wire between input and output. A value of 0% applies equal gain to both the upper and lower wires; at a value of 100% only the upper wire's sound will be audible, at -100%, only the lower wire's sound will be audible. This works like the balance control on any stereo system; as the gain increases for one wire, it decreases for the

other. It's also similar to the PANNER and XFADE functions. The F3 soft button selects the control-input page for the balance stage of this function.

```

EditProg:F3 PUS(BHL/AMP) <>Layer:1/1
Adjst:0dB Src1 :OFF
Depth :0%
Src2 :OFF
KeyTrk: 0.0%/key DptCt1:OFF
VelTrk:0% MinDpt:0%
Pad :0% MaxDpt:0%
<more F1 F2 F3 POS F4 AMP more>
    
```

Parameter	Range of Values
Adjust	± 100%
Key Tracking	± 16.00% per key
Velocity Tracking	± 200%
Pad	0, 6, 12, 18 dB
Source 1	Control Source list
Source 1 Depth	± 200%
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 200%
Maximum Depth, Source 2	± 200%

The AMP stage sets the overall amplitude applied to both wires, and is programmed exactly like the AMP function described above. Their control-input pages are almost identical, including the ranges of values. The only difference is that there's no Pad parameter for the AMP stage of the BAL AMP function. The F4 soft button selects the control-input page for the AMP stage.

GAIN

This function, like AMP, can boost or cut the amplitude of the signal as it passes through the algorithm. Unlike AMP, however, the layer's amplitude envelope doesn't effect the gain settings. GAIN can be used to introduce clipping into a signal, or for adjusting the amplitude of an added waveform. Use a gain function when you want to boost or cut all of a sound's partials uniformly. The control-input page for GAIN is identical to the one for AMP.

Mixers

- +AMP
- +GAIN
- CROSSFADE

The functions in the Mixer category combine two-wire signals in various ways. They have double wires at their inputs, and they mix and amplify the signals from the two wires, then combine them for output to a single wire. Depending on where you assign one of these

functions, they can be used to combine two-wire signals for the F4 AMP block, or to enable you to apply another DSP function to the combined signals before the F4 AMP block.

There's a Pad parameter on the control-input pages for these functions, which attenuates the lower wire's signal at its input to the function.

+AMP

The two input signals to this function are multiplied by .5 (to reduce the likelihood of clipping), then added together. The resulting signal is then multiplied by a gain factor (the combined values for the parameters on the control-input page), and multiplied by 2. Any clipping that occurs can be eliminated by lowering the value of the Adjust parameter. If the Adjust value is -6 dB or lower, the signal will never clip. The control parameters are affected by the settings and controls on the AMPENV page.

+GAIN

This function operates in almost the same way as +AMP, the only difference being that the signal is not affected by the settings on the AMPENV page, since it occurs before the final AMP block.

Crossfade (XFADE)

This function adds the signals from the upper and lower wires after evaluating the combined values of the parameters on its control-input page. If those values add up to -100% , only the lower wire's signal is sent to the output. If they add up to 100% , only the upper wire's signal is sent to the output. If they add up to 0% , both signals are attenuated 6 dB, then added and sent to the output.

Waveforms

SINE

LOW FREQUENCY SINE

SAWTOOTH

LOW FREQUENCY SAWTOOTH

SQUARE

LOW FREQUENCY SQUARE

In this category of DSP functions are three standard synth waveforms—Sine, Sawtooth, and Square—with high- and low-frequency variations of each. These are all one-stage functions. They can be assigned in several different positions and combinations in many of the algorithms.

One important fact to keep in mind is that assigning one of these waveforms to a layer's algorithm may remove the original sample from the signal, since they don't have input signals to send to their outputs (they send only the waveform that they generate themselves). If, for example, you were editing the Acoustic Piano program, and you assigned SINE in the F1 block, you would no longer hear the piano timbre, only the sine wave (unless the signal splits before the F1 block, as in Algorithm 10). Consequently, you'll tend to use these waveforms when you want to build a sound from scratch. If you want to add a waveform to the original timbre of a sound, use one of the added waveform functions described in the next section, or use one of the split signal algorithms.

These waveforms can range in frequency from .1 Hz to 20 KHz. They're not samples like the instrumental sounds and other waveforms; they're generated by oscillators. Since the DSP function waveforms aren't produced by playing back multi-sample keymaps, there are no sample root transitions as you play notes in different keyboard ranges. This makes them especially suitable for use with portamento and wide pitch bend ranges.

Since these waveform functions generate an output signal only, and don't receive an input signal to pass along, the algorithms are arranged so you won't inadvertently assign a series of waveforms that interfere with each other. You'll usually find, for example, that if you can assign a waveform in the F1 block, all subsequent blocks will allow you to assign only the added waveforms. Or, if the subsequent blocks allow you to assign the "regular" waveforms, it's because the wiring of the algorithm is split so that the two waveforms pass through in parallel (as in Algorithm 10).

This next point is a small one, but important, and may make it easier for you to understand the way the waveform functions operate, especially if you've been carefully studying the wiring paths of the algorithms. In several algorithms where the waveforms are available the wiring paths of the algorithms (the horizontal arrows) appear to send a signal to an input of the waveforms. This is *not* the case, and anywhere one of these waveforms is assigned, you should view the algorithm as if there were no horizontal arrow pointing to the left (input) side of the block where the waveform is assigned. The diagrams below will clarify this point. The only difference in the DSP function assignments is in the F1 block, where the first and second diagrams show the SAW waveform, and the third diagram shows the SAW+ added waveform (described in the next section). In the first diagram, the PITCH function's output (passing the sample signal from the keymap) appears to be connected to the input of the F1 block (the SAW function), as well as splitting and passing to the +GAIN function in the F2 block. This is what you would see on the ALG page.

In fact, the actual signal path does *not* pass from the PITCH function through the SAW function; it splits and bypasses the SAW function, as shown in the second diagram. The third diagram shows the same algorithm with the SAW+ added waveform assigned to the F1 block. In this case, the diagram is accurate; the signal passes from the output of the PITCH function, and splits into a two-wire signal. The upper wire passes through the F1 block where the sawtooth wave is added, and into the +GAIN function in the F2 block. The lower wire bypasses the F1 block, and passes directly to the F2 block, where it is combined with the upper wire signal.

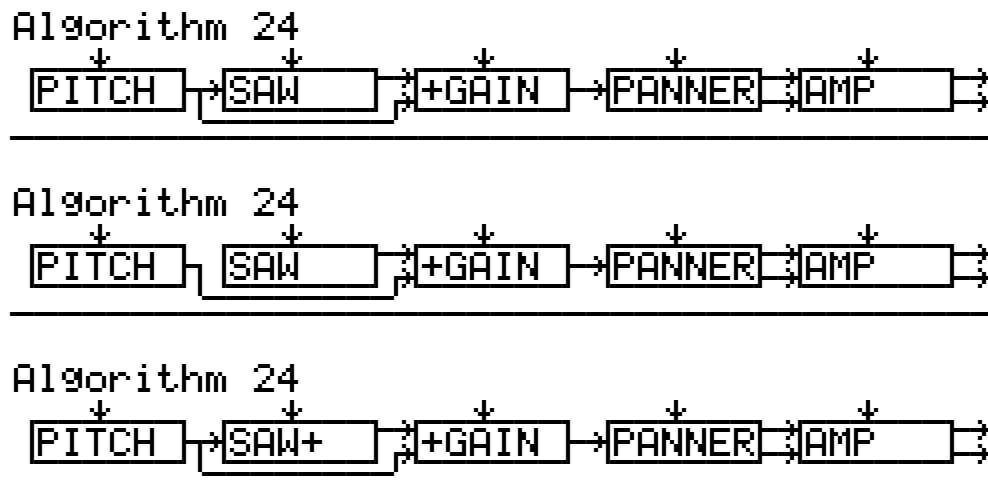


Figure 16-4 Understanding Waveform Wiring

The six waveforms in this category are Sine, Sawtooth, Square, Low Frequency Sine, Low Frequency Sawtooth, and Low Frequency Square. The control-input pages for all six waveforms affect the pitch of the waveforms. The control-input pages for the first three waveforms are identical, as are the control-input pages for the three low frequency waveforms are identical.

SINE, Sawtooth (SAW), SQUARE

There's only one parameter on this control-input page that may still be unfamiliar to you: Fine Hz. This is discussed on page 6-27. It can tune the pitch of the waveform in terms of its actual frequency in Hertz, as opposed to the usual method of tuning by key names. The advantage to using the Fine Hz parameter is that you can maintain constant beat frequencies across much of the keyboard when you have a program with slightly detuned multiple layers (or multiple waveforms in one layer).

```

EditProgB: F1 PCH(SINE) <> Layer: 1/1
Coarse: [051] Src1 : OFF
Fine : [0ct] Depth : [0ct]
FineHz: [0.00Hz] Src2 : OFF
KeyTrk: [0ct/key] DptCt1: OFF
VelTrk: [0ct] MinDpt: [0ct]
Pad : [0dB] MaxDpt: [0ct]
<more F1 PCH F2 F3 F4 AMP more>
    
```

Parameter	Range of Values
Coarse Adjust	-120 to 60 semitones
Fine Adjust	± 100 cents
Fine Adjust in Hertz	± 6.00 Hertz
Key Tracking	± 2400 cents per key (2 octaves)
Velocity Tracking	± 7200 cents (6 octaves)
Pad	0, 6, 12, 18 dB
Source 1	Control Source list
Source 1 Depth	± 7200 cents
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 7200 cents
Maximum Depth, Source 2	± 7200 cents

If you want the waveform to play with standard twelve-tone octaves, set the KeyTrk parameter to **100 cents per key**. Different values for KeyTrk will result in nonstandard tunings.

Low Frequency Waveforms: Sine (LF SIN), Sawtooth (LF SAW), Square (LF SQR)

These can be used like the waveforms above, since their frequency ranges are similar, but they're intended to be used not for their timbres, but for the shape of their waveforms. By using low frequency values for these waveforms, you're basically getting extra LFOs with very precise control parameters. They're intended to be used as inputs to drive the DSP functions in the subsequent algorithm blocks. They're especially useful with the nonlinear DSP functions, such as xAMP.

The parameters on this page affect the pitch of the low frequency waveform in a slightly different manner. They're all tied to the value of the Coarse Adjust parameter, so when you're working with this page, you'll want to set the Coarse Adjust first, then set the values of the other parameters to modify the initial setting. The Coarse Adjust value is multiplied by the values of

the other parameters to determine the effect on the pitch, as indicated by the “x” after the parameters’ values. More parameter descriptions follow below.

```

EditProgH1 PCH(LF SUN) <> Layer:1/1
Coarse:100.0Hz Src1 :OFF
Fine :4.00x Depth :1.000x
KeyTrk: 2.00x/oct DptCt1:OFF
VelTrk: 1.000x MinDpt:1.000x
Pad :0dB MaxDpt:1.000x
<more F1 PCH F2 F3 F4 AMP more>
    
```

Parameter	Range of Values
Coarse Adjust	0.1, 1.0, 10.0, 100.0, 1000.0 Hertz
Fine Adjust	1.00 to 20.00 x
Key Tracking	0.1 to 10.0 x per octave
Velocity Tracking	0.010 to 32.000 x
Pad	0, 6, 12, 18 dB
Source 1	Control Source list
Source 1 Depth	0.010 to 32.000 x
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	0.010 to 32.000 x
Maximum Depth, Source 2	0.010 to 32.000 x

Coarse Adjust (Coarse)

There are only five values to choose from. They’re expressed in terms of their frequencies in Hertz. Each value has a frequency ten times higher or lower than adjacent values.

Fine Adjust (Fine)

Set this parameter’s value to **1.00 x** if you don’t want it to affect the Coarse Adjust setting. Doubling this value (**2.00 x**, **4.00 x**, etc.) will raise the pitch in octaves. By using Fine Adjust in tandem with Coarse Adjust, you can achieve frequencies from .1 Hz (well below the audible range) to 20 KHz.

Key Tracking (KeyTrk)

A value of **1.00 x per octave** will keep the waveform’s pitch uniform across the entire keyboard. A value of **2.00 x per octave** will give you the normal twelve-tone octave. Other values will give you nonstandard tunings.

Added Waveforms

- SINE+
- SAW+
- NOISE+

There are three DSP functions that add waveforms to a layer's existing sample: SINE+, SAW+, and NOISE+.

The parameters on the control-input page for the SINE+ function affect the pitch of the sine waveform without affecting the pitch of the existing sample. The control-input page for the SINE+ function is similar to those for the regular waveforms above. There are parameters for coarse adjust, key tracking, velocity tracking, Source 1 and Source 2, and a pad. There are also parameters for fine adjust and fine Hertz adjust.

The SAW+ function is virtually identical to the SINE+ function; the only difference is in the shape of the waveform.

The NOISE+ function is tied to the level of the sample to which it's added. It will generate fairly white noise (that is, nearly equal amplitude at all audible frequencies) as long as the amplitude of the sample is nonzero. The amplitude of the noise is multiplied by its gain control (the Adjust parameter on its control-input page), then added to the signal. To add a short burst of noise at the beginning of a sound, assign ENV2 as the value of one of the Source parameters, then edit ENV2 to produce an envelope with a rapid decay.

The control parameters for NOISE+ are similar to those for SINE+ and SAW+, except that there are no parameters for fine adjust or fine Hertz adjust.

Nonlinear Functions

High Frequency Stimulator	Two-parameter Shaper
Distortion	Wrap
Shaper	Lowpass Filter with Clipping
Double Shaper	Pulse Width Modulation

The functions in this category have a variety of effects on the signal. What they have in common is that they can add partials to the signal that were not present at their inputs.

The nonlinear functions can produce dramatic changes in timbre, resulting in all sorts of new and modified sounds. One thing to keep in mind is that sounds with a large number of high-frequency partials can be subject to distortion at the high end of the keyboard, especially when you're using more than one of the nonlinear DSP functions. You might also hear a bit of *aliasing* with some sounds. Aliasing refers to unintended partials that occur below the fundamental pitch of a sound. The easiest way to remove this distortion or aliasing is to reduce the level of the Adjust parameter on the control-input page for whichever nonlinear DSP functions you're working with. When you're using PWM followed by DIST or SHAPER, you'd reduce the level of the Adjust parameter for the DIST or SHAPER function. You can also use key tracking (KeyTrk, usually with a negative value), and key tracking in combination with the Keytrack Start (KStart) parameter that's described on page 16-4.

Even with the damping effects of KeyTrk and KStart, you may come up with sounds that are fantastic in the low range, but gritty in the high range. You can transpose the keymap down to counteract this, but that's the nature of the nonlinear functions. In extreme cases, you can lower the HiKey of the layer to disable the high end completely.

High Frequency Stimulator (HIFREQ STIMULATOR)

The overall effect of this three-stage function is to boost the high frequency partials of the signal, and depending on the settings of the control inputs, it can add high-frequency partials to the signal as well. It's useful for building sounds that cut through the mix, and have a bright crisp nature.

There's more to the High Frequency Stimulator than meets the eye. It works like this: the signal is run through a high-pass filter, then through a distortion function, then through a second high-pass filter. Finally, it's mixed with the original signal after passing through the final AMP stage of the algorithm. The three control-input pages let you adjust the cutoff frequency of the first high-pass filter (F1 FRQ), the amount (drive) of the distortion function (F2 DRV), and the mix (relative amplitude) of the stimulated signal with the original (F3 AMP).

```

EditProg:F1 FRQ<H>F2 DRV<S>F3 AMP<LAYER:1/1
Coarse:C 4 262HZ Src1 :OFF
Fine :Oct Depth :Oct
KeyTrk:Oct/key Src2 :OFF
VelTrk:Oct DptCtl:OFF
Pad :Oct MinDpt:Oct
MaxDpt:Oct
<more F1 FRQ F2 DRV F3 AMP F4 AMP more>
    
```

Parameter	Range of Values
Coarse Adjust	C -1 16 Hz to G 10 25088 Hz
Fine Adjust	± 100 cents
KEY Tracking	± 250 cents per key
Velocity Tracking	± 10800 cents
Pad	0, 6, 12, 18 dB
Source 1	Control Source list
Source 1 Depth	± 10800 cents
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 10800 cents
Maximum Depth, Source 2	± 10800 cents

DSP Functions

The DSP Functions

```

EditProgBF2 DRU(H)FRQ S UNO<>LAYERB1/1
Adjust:0dB Src1 :OFF
Depth :0dB
KStart:C -1 unipola Src2 :OFF
KeyTrk: 0.00dB/key DptCtl:OFF
VelTrk:0dB MinDpt:0dB
MaxDpt:0dB
<more F1 FRQ F2 DRU F3 AMP F4 AMP more>
    
```

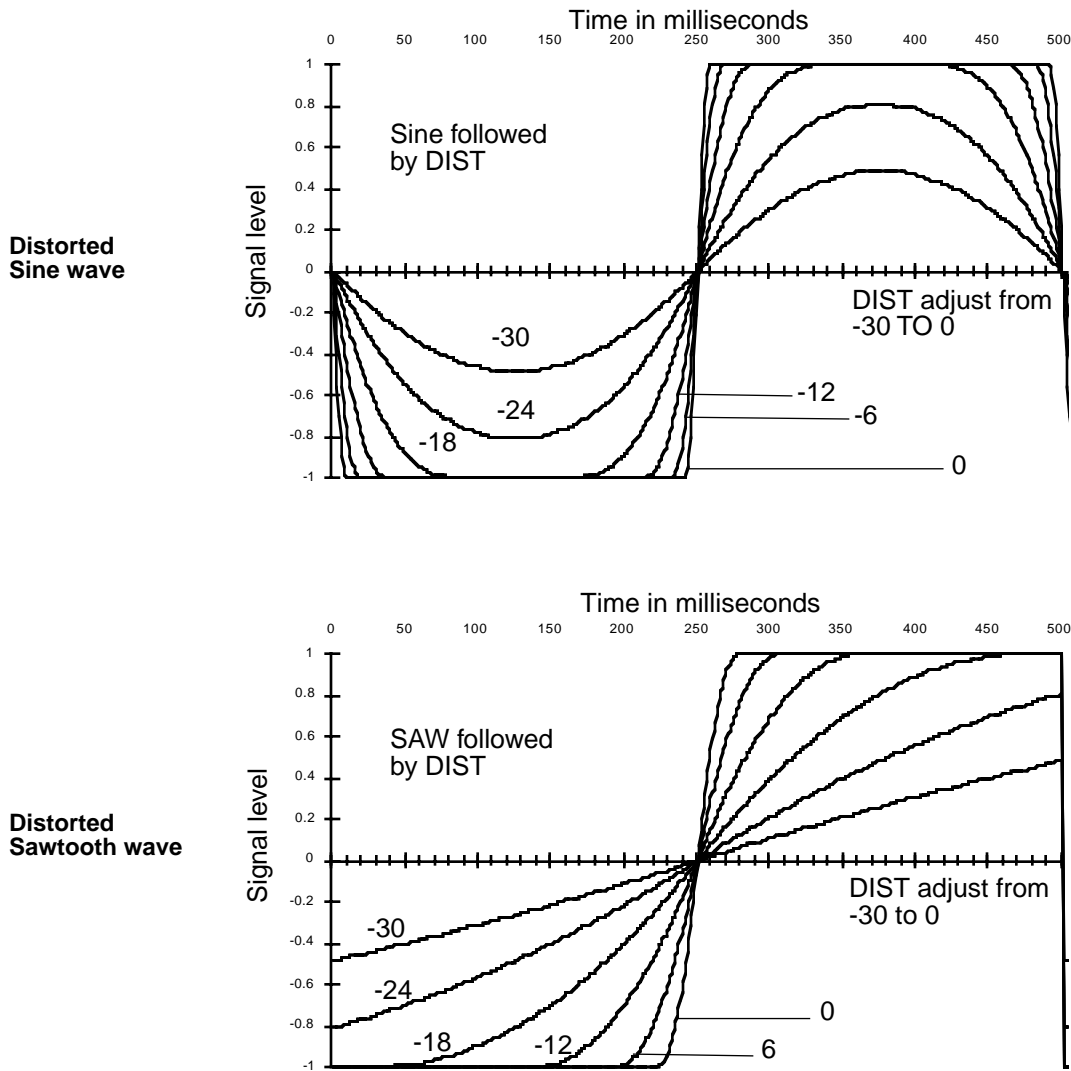
Parameter	Range of Values
Adjust	-96 to 48 dB
Keytrack Start	C -1 to C 9 unipolar, C -1 to C 9 bipolar
Key Tracking	± 2.00 dB per key
Velocity Tracking	± 96 dB
Source 1	Control Source list
Source 1 Depth	± 96 dB
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 96 dB
Maximum Depth, Source 2	± 96 dB

```

EditProgBF3 AMP(H)FRQ S UNO<>LAYERB1/1
Adjust:0dB Src1 :OFF
Depth :0dB
Src2 :OFF
KeyTrk: 0.00dB/key DptCtl:OFF
VelTrk:0dB MinDpt:0dB
MaxDpt:0dB
<more F1 FRQ F2 DRU F3 AMP F4 AMP more>
    
```

Parameter	Range of Values
Adjust	-96 to 48 dB
Key Tracking	± 2.00 dB per key
Velocity Tracking	± 96 dB
Source 1	Control Source list
Source 1 DEPTH	± 96 dB
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 96 dB
Maximum Depth, Source 2	± 96 dB

Distortion (DIST)



Using this function is much like overdriving an ordinary keyboard or guitar amplifier. The input signal is multiplied by the gain control (the combined values of the parameters on its control-input page labeled DRV, for Drive), then passes into a distortion mapper. Large Adjust values will cause serious amounts of distortion.

Different sounds are affected differently by DIST. Waveforms that are static (waveforms with shapes that repeat regularly and are not evolving) when they enter the DIST function will undergo more of a timbre change than the familiar sound of distortion.

The DIST function distorts each note separately, unlike a fuzz box, which adds several notes together then applies a uniform amount of distortion to all of them. Consequently your power chords may sound a little different from your expectations, but you can also get some great effects with key and velocity tracking (not to mention Sources 1 and 2!) that aren't possible with other distortion devices.

The page below shows the DIST function in the F1 block, but it can appear in other blocks as well.

```

EditProg:F1 DRUM(DIST) <>Layer:1/1
AdjJust:0dB Src1 :OFF
Depth :0dB
KStart:C -1 unipola Src2 :OFF
KeyTrk: 0.00dB/key DptCtl:OFF
VelTrk:0dB MinDpt:0dB
Pad :0dB MaxDpt:0dB
<more F1 DRU F2 F3 F4 AMP more>
    
```

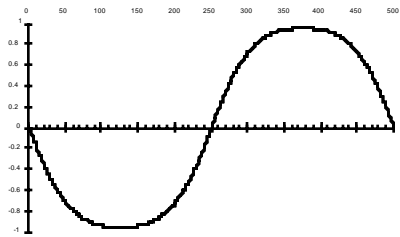
Parameter	Range of Values
Adjust	-96 to 48 dB
Keytrack Start	C -1 to C 9 unipolar, C -1 to C 9 bipolar
Key Tracking	± 2.00 dB per key
Velocity Tracking	± 96 dB
PAD	0, 6, 12, 18 dB
Source 1	Control Source list
Source 1 Depth	± 96 dB
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 96 dB
Maximum Depth, Source 2	± 96 dB

SHAPER

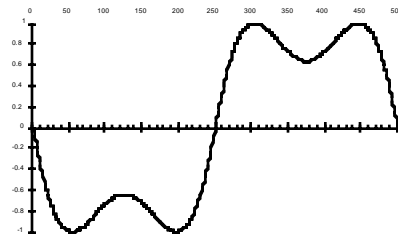
The effect of SHAPER can be very unpredictable, and the mechanics of its operation lend themselves toward explanations that are more numerical than verbal. The best way for you to get a feel for the SHAPER is to start with single-cycle waveform keymaps and experiment with different values for the parameters on its control-input page (labeled AMT, for Amount), and listen to the results. SHAPER tends to work best with the single-cycle waveform sounds (keymaps with IDs 112—166), and is usually less effective with acoustic instrumental sounds. SHAPER often produces numerous peaks throughout the frequency range, even at frequencies that didn't have much amplitude to begin with. These peaks can sound like resonant filters, and can even sound voice-like.

The two series of graphs that follow show the effect of SHAPER on two typical single-cycle waveforms. The first set of six graphs just below shows the evolution of a sine wave input as the value of the Amount parameter is increased. The following set of six graphs shows the effect when the Adjust parameter is increased. Each graph plots a 500-millisecond segment of

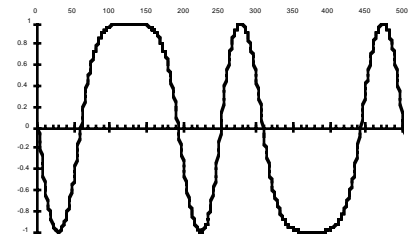
waveforms cycling at frequencies of 2 Hz. Of course, these are just a few of the countless modulations you can apply to different waveforms at different frequencies.



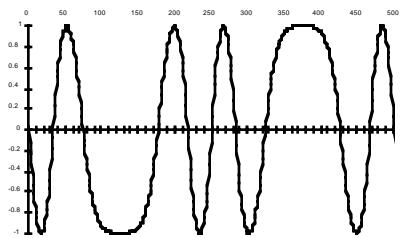
Amount = .1



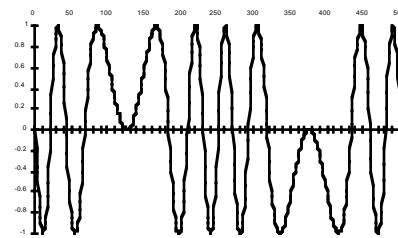
Amount = .2



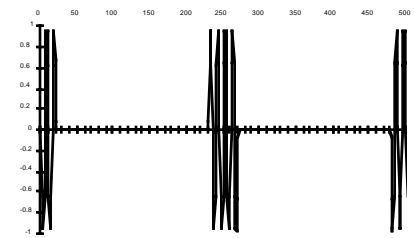
Amount = .375



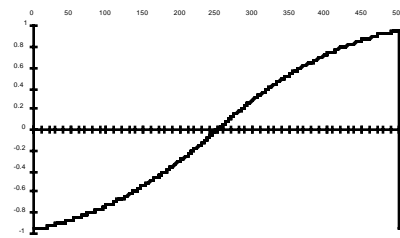
Amount = .625



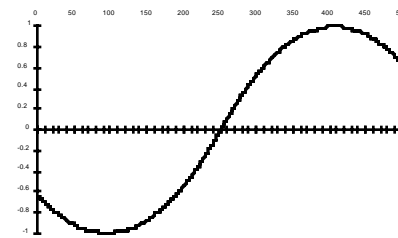
Amount = 1



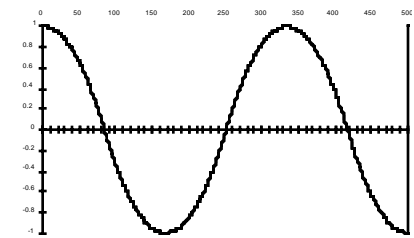
Amount = 4



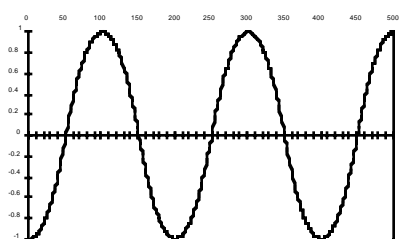
Adjust = .1



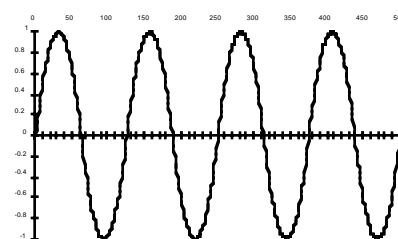
Adjust = .2



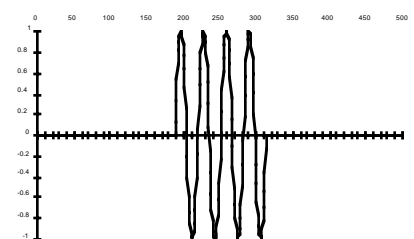
Adjust = .375



Adjust = .625



Adjust = 1



Adjust = 4

As the SHAPER receives input signals, it evaluates the signal's level according to its own internal scale. When the SHAPER's Adjust value is at .25, an input signal moving from negative full scale to positive full scale (a sawtooth) will map to an output curve with a single-cycle sine wave shape. At an adjust value of .5, the same input signal would map to a 2-cycle sine wave output signal. Adjust values of .75 and 1.0 for the SHAPER would map to 3-and 4-cycle sine wave output signals, respectively. Beyond values of 1.0, some portions of the output will pin at zero-scale.

Small Adjust values for the SHAPER can sound much like the DIST function, but larger values will introduce dramatic changes in timbre, while DIST will have a less pronounced effect on timbre.

The following F1 page shows the SHAPER function in the F1 block, but it can appear in other blocks as well.

```

EditProg:F1 AMT(SHAPER) <>Layer:1/1
Adjust: 0.100x Src1 :OFF
Depth : 0.00x
KStart:C -1 unipola Src2 :OFF
KeyTrk: 0.000x/key DptCtl:OFF
VelTrk: 0.00x MinDpt: 0.00x
Pad :0dB MaxDpt: 0.00x
<more F1 AMT F2 F3 F4 AMT more>
    
```

Parameter	Range of Values
Adjust	0.100 x to 4.000 x
Keytrack Start	C -1 to C 9 unipolar, C -1 to C 9 bipolar
Key Tracking	± 0.200 x
Velocity Tracking	± 4.00 x
Pad	0, 6, 12, 18 dB
Source 1	Control Source list
Source 1 Depth	± 4.00 x
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 4.00 x
Maximum Depth, Source 2	± 4.00 x

The values for each of the parameters on the SHAPER's control-input page are expressed in arbitrary quantities that represent a multiplication of the amount of shaping applied.

Double Shaper (SHAPE2)

This is simply a series of two SHAPERS. The first is programmed exactly like SHAPER. The values of the control parameters for the second are fixed at .75 times those of the first. This can produce effects that the single SHAPER can't. If, for example, you set the Adjust parameter of SHAPE2 to 1.000, it will process the input signal with a value of 1.000, then again with a value of .75. This is not the same as processing the input signal once with an Adjust value of 1.75.

Two-parameter Shaper (2PARAM SHAPER)

This function is similar to the SHAPERs described above, but it has two control-input pages instead of one. The F1 EVN control parameters enable you to add distortion to sine wave partials that are even harmonics of the input signal, and the F2 ODD control parameters let you add distortion to sine wave partials that are odd harmonics of the input signal.

In simpler terms, the control parameters behave like those of the regular SHAPER, but they can shape the signal about six times more than SHAPER can. 2PARAM SHAPER works by separately multiplying the input signal by the combined values of the two sets of control parameters, adding the resulting signals, and multiplying that sum by a constant, then wrapping the signal values that exceed positive or negative full scale (see the WRAP function below).

Experimentation is the key here. Start with very low values for each of the Adjust parameters, and increase them until you begin to hear an effect. Some values will create a DC offset (see diagram below) in the signal—that is, the signal won't oscillate around the normal zero-point of the scale, but will be shifted toward positive or negative full scale. This may cause a click or thump in sounds with rapid attacks, decays, or releases. To reduce the click or thump, you can edit the AMPENV to produce a more gradual envelope.

2PARAM SHAPER works best with the single-cycle waveform keymaps (IDs 112—166).

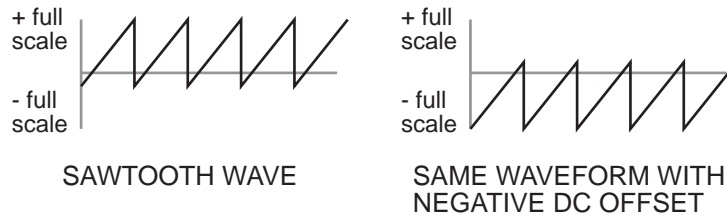
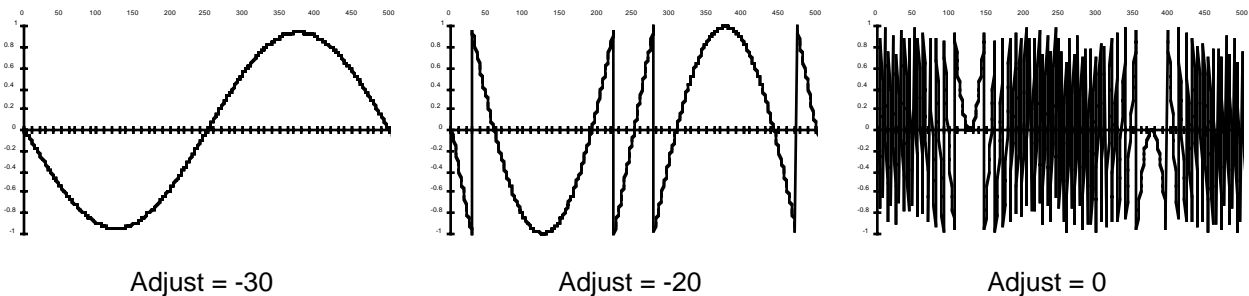


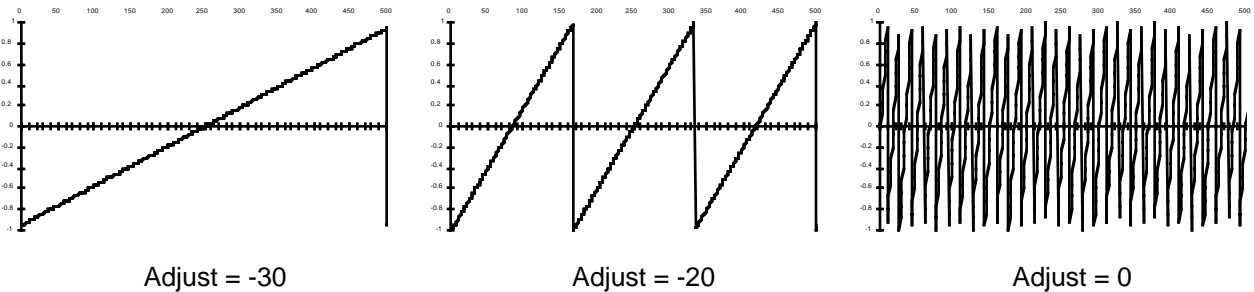
Figure 16-5 DC Offset

Waveform Wraparound (WRAP)

The next three graphs show the effect of various amounts of WRAP on a sine wave.



The following three graphs show the effect of WRAP on a sawtooth wave at the same frequency.



With this function you can completely mutilate a sound, and with large amounts of wrap, turn anything into white noise. At the input of the WRAP function, the signal is multiplied by the combined values of the parameters on the WRAP control-input page, then multiplied by an additional gain factor of 30. If the resulting value is greater than full scale (in other words, if it's sufficiently high to clip), then instead of clipping, the waveform "wraps" back around to negative full scale, and it continues to evolve from that point. Likewise, if the resulting value is less than negative full scale, it wraps to positive full scale and proceeds from there. For any waveform, several of these wraparounds can take place before the waveform fits into the allowable range.

You'll want to try different values of the Adjust parameter to get a feel for the results of different amounts of wraparound. Look for the value that introduces a very slight amount (it will tend to be well below 0). The sound will start to buzz here and there, as a few segments of the input wrap around. As you increase the Adjust value, the buzz will increase, and the pitch of the sound will begin to disappear. Keep adding to the Adjust value, and you'll end up with white noise, regardless of the starting timbre.

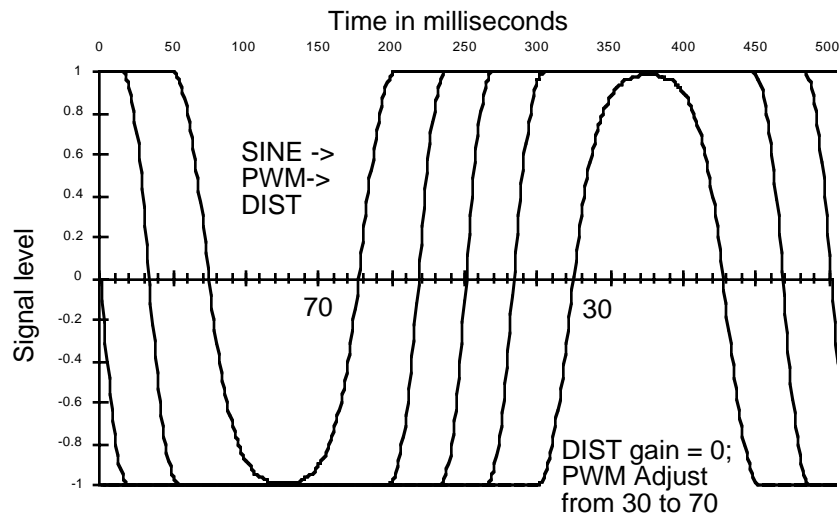
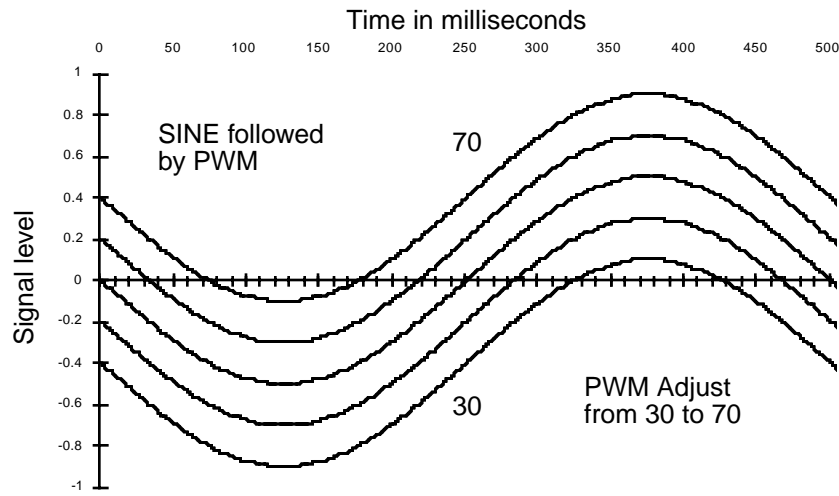
The bright buzzy nature of wrapped sounds is due to the discontinuities in the waveforms of the partials as they wrap around from positive full scale to negative full scale and vice versa. If you want, you can reduce or eliminate the buzz by sending the signal through SHAPER after it goes through WRAP (assign SHAPER as the DSP function in the following algorithm block). Set the SHAPER's Adjust parameter to .25. This will map both positive full scale and negative full scale amplitudes to a level of zero, eliminating the wraparound discontinuities, but preserving the alterations in the waveform produced by WRAP. SHAPER will add its own effects as well.

The control-input page for WRAP uses the same set of parameters and ranges of values as DIST.

Lowpass Filter with Clipping (LPCLIP)

This is a one-pole filter, which is programmed just like LOPASS. The difference with LPCLIP is that the amplitude of the input signal is multiplied by 4 before the filter. This can cause the signal to clip, which can produce interesting results. Naturally, you'd use this function only when you want to induce clipping intentionally as a component of your sound.

Pulse Width Modulation (PWM)



Pulse width modulation can produce some classic synth sounds, and can break new sonic ground as well. Its operation is very simple; it adds an oscillating DC offset to the input signal (shifts it toward positive or negative full scale). Unless it causes the signal to clip, it won't have much effect on the signal. It's designed to be followed by DIST, with the Adjust parameter for DIST set to a fairly high level. The DIST function will drive all positive elements of the signal toward positive full scale, and negative elements toward negative full scale. The result is a rectangular wave with a width that varies according to the Adjust level of PWM.

PWM won't affect a square wave input. Sine and triangle waves will produce familiar PWM sounds. More complicated waveforms will result in discontinuous rectangular waveforms.

You can also follow a PWM algorithm block with SHAPER, since SHAPER's output is affected by the DC level of the signal.

DSP Functions

The DSP Functions

The parameters on the PWM control-input page affect the DC offset of the signal, in terms of the percentage of shift from no offset to maximum offset. At a value of 0%, there is an offset positive full scale. At 100%, the offset is negative full scale, and at 50%, there is no offset.

A typical control configuration for PWM has the Coarse Adjust set to 50%, an LFO assigned to Src1, and Src1's Depth parameter set to a value of 25%.

The page below shows the PWM function in the F1 block, but it can appear in other blocks as well.

```

EditProg:F1 W10(PWM) <>Layer:1/1
Adjust:50% Src1 :LFO1
Depth :25%
Src2 :OFF
KeyTrk: 0.0%/key DptCtl:OFF
VelTrk:0% MinDpt:0%
Pad :0dB MaxDpt:0%
<more F1 W10 F2 F3 F4 AMP more>
    
```

Parameter	Range of Values
Adjust	0 to 100%
Key Tracking	± 8.00% per key
Velocity Tracking	± 100%
Pad	0, 6, 12, 18 dB
Source 1	Control Source list
Source 1 Depth	± 100%
Source 2	Control Source list
Source 2 Depth Control	Control Source list
Minimum Depth, Source 2	± 100%
Maximum Depth, Source 2	± 100%

Waveforms Combined with Nonlinear Functions

Added Sawtooth with Nonlinearity (SW+DST)	x Shape Mod Osc
Added Sawtooth plus Shaper (SW+SHP)	+ Shape Mod Osc
Shape-modulated Oscillator	Amp-modulated Oscillator

The six functions in this category do one of two things: they combine samples and waveforms with a nonlinear DSP function, or they use a waveform or sample as inputs to drive nonlinear functions.

Added Sawtooth Wave with Nonlinearity (SW+DST)

This function starts by adding a sawtooth wave to the layer's sample input. When the input signal from the sample is added to the sawtooth, the signal may exceed full scale, so a wraparound function similar to WRAP is performed. The result is then squared to remove any discontinuities from the wraparound. The resulting signal has a large DC offset, so a constant of 3/8 is subtracted.

The parameters on the control-input page for SW+DST control the pitch of the sawtooth wave.

Added Sawtooth Wave Plus SHAPER (SW+SHP)

For this function, the sample input is combined with a sawtooth wave, then passed into the SHAPER function. The SHAPER has a constant Adjust value of .25. First, the sample input is multiplied by a constant, which may cause it to clip. Any clipping becomes part of the signal. This result is added to the sawtooth wave, which may cause the waveform to exceed full scale. If it does, the signal wraps around as in the WRAP function. This result is then put through the SHAPER.

You may want to use the Pad parameter with this function, depending on the sample you use. Pastorius fans should try this function using one of the Electric Bass keymaps. Use Algorithm 8, and start with the second and third algorithm blocks set to **NONE**. Set the fourth block to **SW+SHP**. Try setting the keymap transposition to **-12**, and the SW+SHP key tracking to **100 cents per key**.

Shape-modulated Oscillator (SHAPE MOD OSC)

This function combines the sample input with a sine wave at 1/4 scale, does a combination of additions and multiplications on both signals, then puts the result through a SHAPER function. The amount of shaping depends on the levels of both the input signals. First, the SINE value is multiplied by the sample input value, then multiplied by a constant—any samples exceeding full scale will wrap around. The result is added to the wrapped product of the SINE value times a constant. The entire resulting waveform is then passed through the SHAPER, whose Adjust value is set by the level of the sample input. You might think of this function as an oscillator whose shape is controlled by the sample input signal.

Even if the depth value goes to zero, there will still be a nonzero final output, due to the addition of the multiplied SINE value. In this case, the output would be a slightly distorted sine wave. As the value of depth increases, the harmonic content of the output signal will rise, assuming the pitches of the sample input and the sine wave are simply related—unison, octaves, etc. Slight detuning between the two pitches will cause a slow beat frequency.

The parameters on the F2 PCH control-input page affect the pitch of the sine wave. Those on the F3 DEP page affect the level of the sample input, and consequently the amount of SHAPER that's applied. If the DEPTH values exceed +5 dB, then the product of SAMPLE INPUT x DEPTH may clip, adding further harmonics through a mechanism that differs from the addition of harmonics below the +5 dB DEPTH level.

x SHAPE MOD OSC

Available only in Algorithm 18, this function is similar to SHAPE MOD OSC, except that it multiplies its two input signals and uses that result as its input.

+ SHAPE MOD OSC

Also available only in Algorithm 18, + SHAPE MOD OSC is similar to x SHAPE MOD OSC, except that it adds its two input signals and uses that sum as its input. With this and all the modulated oscillators, let your ears be your guide.

Amp Modulated Oscillator / Final Amp (AMP MOD OSC)

This function is available only in Algorithm 17. The sample input is multiplied by the output of a sine wave oscillator. This result is scaled by the parameters on the F3 DEP control-input page, and the result is added to the original sample input and sent to the final AMP function. The

parameters on the F2 PCH control-input page affect the pitch of the sine wave, and consequently all subsequent results.

Mixers with Nonlinear Inputs

x AMP

! AMP

x GAIN

Amplitude Modulation

x AMP

This function can be used in the final algorithm block when it mixes two input wires into a single output. The two input signals are multiplied. The control-input parameters affect the gain of the multiplied signals. The final amplitude is also affected by the settings for AMPENV and ENVCTL. Multiplying the two signals can result in outputs that differ dramatically from the input signals. You can get a wide variety of effects from this function, for example, turning an acoustic sample and a waveform (or two waveforms) into a sound that has little resemblance to the input sounds.

When two signals are multiplied, the resulting signal consists of the sums and differences of the frequencies of each partial of each signal. The frequencies of the original signals do not come through, unless they have one or more DC components (nonoscillating partials). And of course, if one of the signals has zero amplitude, the resulting signal also has zero amplitude.

If the fundamental frequencies of the two input signals are related by simple fractions (that is, if the ratios between their frequencies are something like 1/1, 2/1, 3/1, 4/1, 1/2, 1/3, 1/4, 2/3, 3/2) the resulting signal will be a harmonic sound. Its partials will be multiples one of the original fundamentals, or possibly a new fundamental. If the frequency ratios of the original signals are nearly but not quite one of these fractions, some beat frequencies will be perceived, which may or may not be useful. Of course, with equal temperament, the ratios given above are not perfectly precise (a perfect fifth, for example, has a frequency ratio of 1.4983, not 3/2). If the frequencies of the original signals are not at least closely related, the result of X AMP will be, shall we say, less than harmonious.

If the frequency of one of the original signals is below the audible range, then the result of X AMP is not a matter of harmony, but of amplitude. In this case, a tremolo effect (amplitude modulation) would be heard, because the resulting signal would periodically dip below the audible range. In fact, when you're using X AMP in the final algorithm block, you can use any sample as an LFO source by setting the Adjust parameter on the PITCH page to its minimum. To make this work, the algorithm must use one of the waveform functions in one of the blocks, and the sample signal must be routed to the x AMP block. Results will vary.

x GAIN

This function operates like X AMP, except that it is not affected by the settings for AMPENV, since it occurs before the F4 AMP block.

SHAPER / Final Amp (! AMP)

This function also appears in the final algorithm block when it mixes two input wires to a single output wire. The two inputs are added, then put through the SHAPER function with a fixed Adjust value of .25, then amplified according to the values for the parameters on the F4 AMP control-input page.

Amplitude Modulation (AMP MOD)

The AMP MOD function multiplies its two input signals, and the result is multiplied by a gain value that is determined by the parameters on the AMPMOD's control-input page. This result determines the balance between the upper and lower wires. AMP MOD can clip the signal, so you may need to use the Pad parameter.

Hard Sync Functions

SYNC M AND SYNC S

These two functions appear in Algorithms 26–31, and always work in tandem. Each is a rising sawtooth oscillator. SYNC M is the “master” waveform, and SYNC S is the “slave.” These terms stem from the fact that the pitch (frequency) of the master waveform determines the repetition rate, and thus the shape, of the slave's waveform. These functions generate their own waveforms, and do not pass the sample input through the algorithm. Consequently the PITCH function does not appear for these algorithms.

Every time the master waveform falls from positive full scale to negative full scale, the slave waveform is forced to negative full scale. You can create a wide variety of timbres by adjusting the pitches of the waveforms relative to each other. This is done with the parameters on the F1 PCH and F2 PCH control-input pages. F1 is for the master, and F2 is for the slave. Pitch control is really a bit of a misnomer for the slave waveform, because its pitch is determined by the pitch of the master waveform. The fundamental of the slave waveform is forced to be the same as that of the master, since they always have the same frequencies, although the shapes of their waveforms differ.

To clarify this, assume for now that the pitch of the master waveform remains constant. When you trigger a note, both waveforms start at negative full scale. If the slave's “pitch” control is set to a much lower value than that of the master, the master waveform will reach positive full scale before the slave. So the shape of the slave waveform will be that of a more slowly rising sawtooth wave with a relatively large negative DC offset (most of the waveform will be in the negative portion of the scale).

If the slave's “pitch” control is set to a value only slightly lower than that of the master, the waveforms will be very similar, and the slave waveform will have a small negative DC offset. When the pitch settings are identical, the waveforms are identical.

If the slave's pitch setting is higher than that of the master (which usually gives more interesting results), the slave's waveform will alternate between a complete sawtooth cycle and a fraction of the subsequent cycle. At twice the frequency of the master, the waveform will have twice the frequency, and only the even harmonics of the master frequency will be pronounced. When the slave/master frequency ratio is nearly, but not exactly three, all harmonics will be present, but the 3rd, 6th, 9th (all multiples of 3) harmonics will be much louder than the others. This will sound like a resonant filter with multiple resonance peaks.

DSP Functions

The DSP Functions

Because the pitch of the slave waveform is forced to be nearly that of the master, you can adjust the key tracking of the slave to values less than **100 cents per key** without affecting the pitch. This will help reduce some of the harshness at the high end of the keyboard.

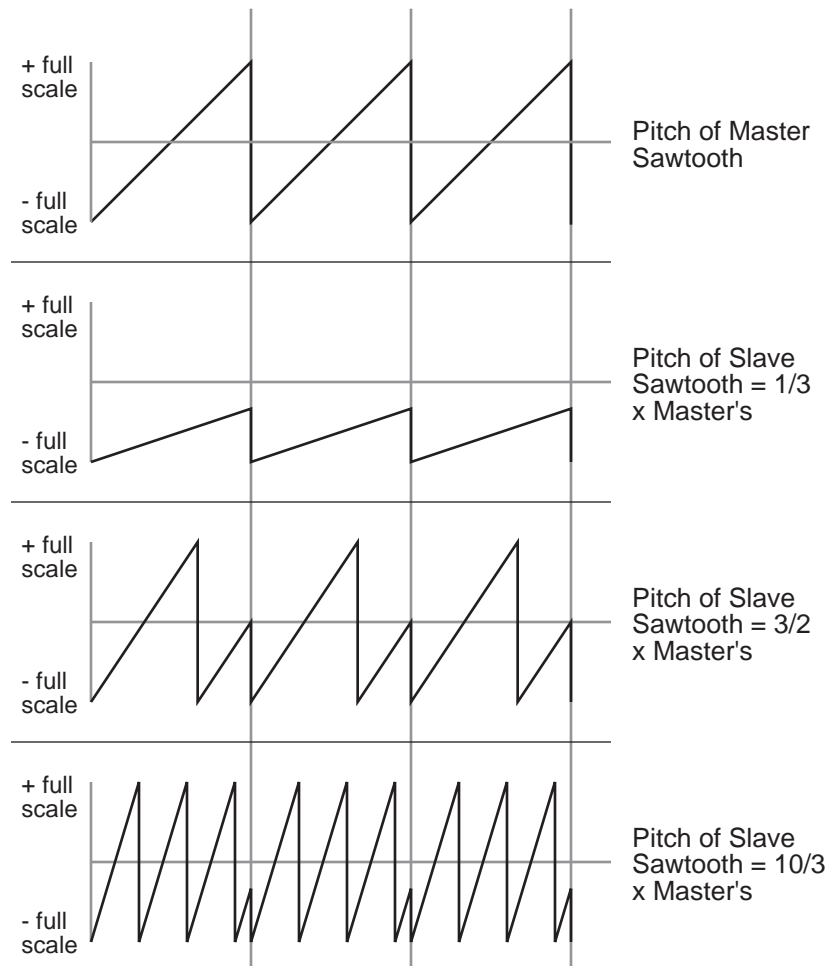


Figure 16-6 Hard Sync Functions