rdq-1 instruction manual

introduction

Congratulations on your purchase of the rdq-1 digital reference equalizer. The rdq-1 is the result of many years of research and development in digital audio technology, and when used properly, it will give you listening pleasure you may not have thought possible from your system. Thus, the purpose of this document is to give you a complete understanding of the theory and operation of the rdq-1 so that you may apply it properly in your home playback system.

The simplest description of the rdq-1 is given by its name: it's a digital equalizer. That is to say, it performs the functions of a conventional equalizer, only on digital input signals instead of analog signals. This is shown schematically in Figure 1. The rdq-1 is placed between your digital sources and your digital-to-analog converter (DAC).

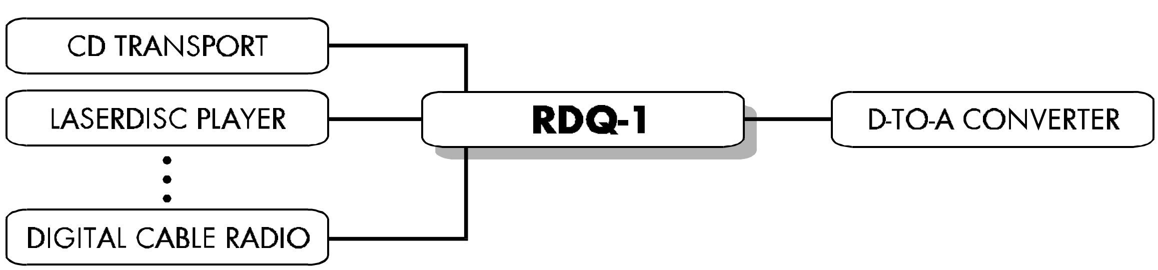

FIGURE 1

The input signals to the rdq-1 are the digital outputs of your digital sources (*e.g.*, the S/PDIF digital stream from your CD transport). Once set up in this manner, the rdq-1 allows you to perform a number of useful functions directly in the digital domain, including input selection, dither selection, tone control, and volume control. In addition, the rdq-1 has a number of useful memory features that make it extremely versatile, flexible, and easy to configure. The following pages will give you a simple “walking tour” through the rdq-1. Appendices A, B, and C provide more detailed information.

• input selection

The rdq-1 allows you to choose between two different digital inputs. There are a number of reasons why input source selection is best performed in the digital domain. The most significant is the reduction of crosstalk. An analog equalizer will have crosstalk between the different analog input sources. There is infinite rejection between input channels on the rdq-1; crosstalk is simply not possible.

How to do it. The factory-default configuration for the rdq-1 is with 1 AES/EBU input on an XLR connector and 1 S/PDIF input on an RCA connector. There are also 2 digital outputs: 1 XLR and 1 RCA. Regardless of which input is active, both outputs are active simultaneously. To select which input is used by the rdq-1, press the **system** button. The display will appear as shown in Figure 2. To change the active input, simply rotate the knob below the input selection display.

• dither selection

For a more thorough discussion of dither, please refer to Appendix A.

The rdq-1 allows you to control the dither it uses so that its output signal is compatible with your DAC. The rdq-1 carries out its computations in 40-bit floating-point arithmetic, which then must be converted back to a fixed-point representation usable by your DAC. Internally, the 40-bit result is first converted to 24-bit fixed-point. The dither control then determines how many bits appear at the outputs of the rpd-1 and how those bits are obtained. You can choose from:

- 24 bits
- truncated to 20 bits (no dither)
- dithered to 20 bits
- truncated to 16 bits (no dither)
- dithered to 16 bits

The majority of DACs today use 20-bit digital-to-analog chips as their core conversion technology; hence, we tend to recommend the 20-bit dithered output as the most useful across the broadest spectrum of source material and DACs today. As 24-bit converters and 24-bit digital filter chips become more commonplace, we will recommend the 24-bit output setting.

How to do it. To select which dither mode is used by the rdq-1, press the **input** button. The display will appear as shown in Figure 2. To change which dither mode is used, simply rotate the knob below the dither selection display.

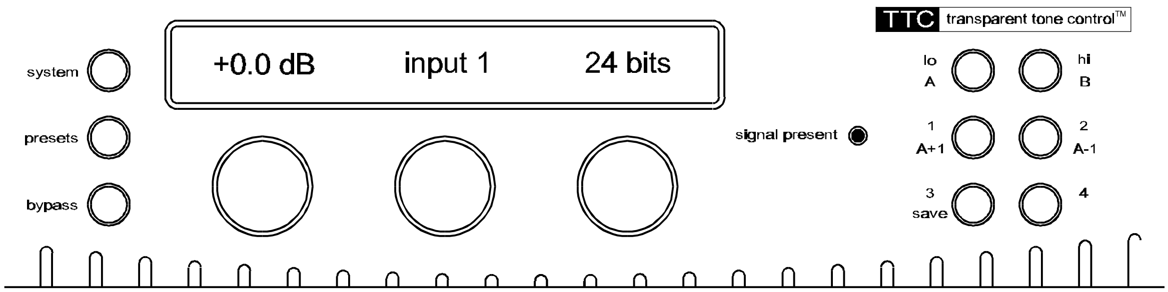

FIGURE 2

• transparent tone control

For a thorough discussion of equalization, please refer to Appendix A. There, you will find a complete explanation of concepts such as center frequency, bandwidth, gain, boost, bell filters, and shelving filters. The reader who is familiar with these concepts may skip Appendix A and proceed with the following.

Of all the rdq-1's functions, the transparent tone control merits the most in-depth discussion. Most high-end audio equipment does not include any form of tone control. There are several reasons for this, but the simplest is that a tone control would require additional circuitry in the signal path, which would lead to audible degradation of the audio signal. This reasoning is indeed correct for tone controls, which work in the analog domain. The rdq-1, however, works entirely in the digital domain. Z-Systems has developed a very sophisticated tone control algorithm we call Transparent Tone Control (TTC) and the TTC facility built into the rdq-1 is its most important feature. TTC is actually more than a tone control; it's a complete digital-domain parametric equalizer that can be used subtly for speaker correction, dramatically for room correction, or artistically, to sculpt the signal and make it sound the way you want it to sound.

You may have a negative impression of digital-domain tone control; indeed, there have been many poor implementations in the past. Rest assured: we're confident you'll find TTC to be free from any non-beneficial coloration and to be the finest tone control you've ever heard. TTC is a direct implementation of the digital equalization algorithm used in our line of professional disc mastering equalizers. In fact, there's a good chance many of your compact discs were prepared using one of our equalizers and that TTC was used in the preparation of many of your favorite compact discs.

The TTC provides you with 4 completely independent bell filters, a high-shelf filter, and a low-shelf filter. Let's first talk about the bell filters. You have a fairly wide range of control on each of the parameters for the bell filters. The center frequencies range from 28 Hz to 18 kHz, giving you 56 possible values spaced uniformly on one-sixth octave centers. The gain at the center frequency ranges from a maximum of +12 dB boost to a minimum of -12 dB cut. Finally, you have 12 values of slope which typically correspond to values of Q between 0.4 and 8.0. These parameters allow you to generate fairly complicated frequency-domain profiles, ranging from precisely located notches to broad, sweeping curves. The low- and high-shelf filters can have their corner frequencies placed at the same numerical values as the center frequencies for the bell filters. The boost and cut similarly range from +12 dB to -12 dB.

How to do it. With the rdq-1 in normal operating mode, simply press one of the buttons labeled **lo**, **hi**, **1**, **2**, **3**, or **4**. The **lo** button corresponds to the low-shelf filter and the **hi** button corresponds to the high-shelf filter. Buttons **1**, **2**, **3**, and **4** correspond to the four bell filters, respectively. Figure 3 shows the rdq-1 as if button **1** had been hit (ignore for now the particular values assumed by the various parameters). Observe that the first display window shows the number "1," indicating that filter 1 is active. In this state, the three knobs control the various parameters for bell filter 1; the knob below the gain display controls the amount of boost/cut, the knob below the frequency display controls the center frequency, and the knob below the slope display controls the slope.

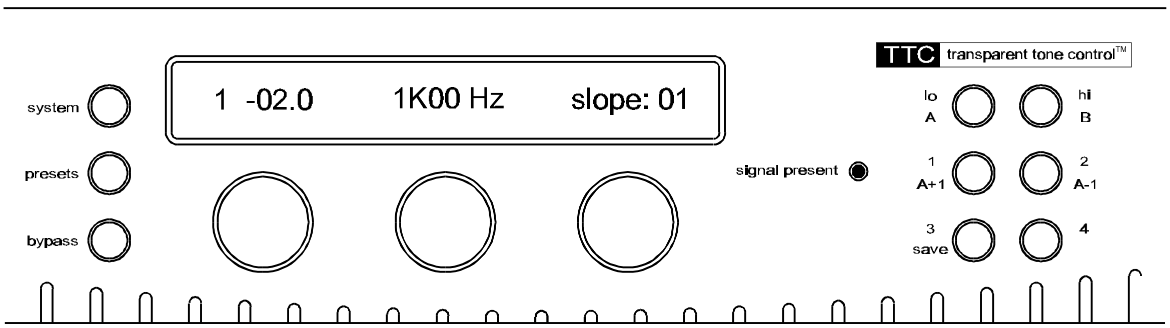

FIGURE 3

Next, press the **lo** button. The displays change as shown in Figure 4. Observe the "L" in the first display window. Notice there is no value of slope shown; this is because low-shelf filter has a fixed slope, which cannot be modified by the user. The corner frequency and gain parameters for the low-shelf filter are adjusted in the same manner as with the bell filter. The same is true for the high-shelf filter. If you press the **1** button again, the parameters you dialed in earlier for bell filter 1 will be displayed again. All six filters can be set in this manner; press the button corresponding to the filter of interest in order to display its parameters and bring the focus of the knobs to that particular filter.

It is important to realize that the display will change whenever the **volume, lo, hi**, **1**, **2**, **3**, or **4** buttons are pressed. This is because the display is not large enough to show the parameters for all six filters and the volume at once. Think of the buttons as serving to "bring the focus" of the knobs and display to the function represented by the button. For example, suppose you have dialed in parameters for bell filter **1**. When you press the **2** button, the parameters on the display will change, reflecting what is happening with bell filter **2**. Bell filter **1** is still in the signal path with its corresponding parameters; you've simply moved on to control bell filter **2**.

Whenever the gain of a filter is set to 0.0 dB, that filter is bypassed algorithmically, regardless of the center frequency or slope. Therefore, it is sufficient to dial in a gain of 0.0 dB to "null" a filter.

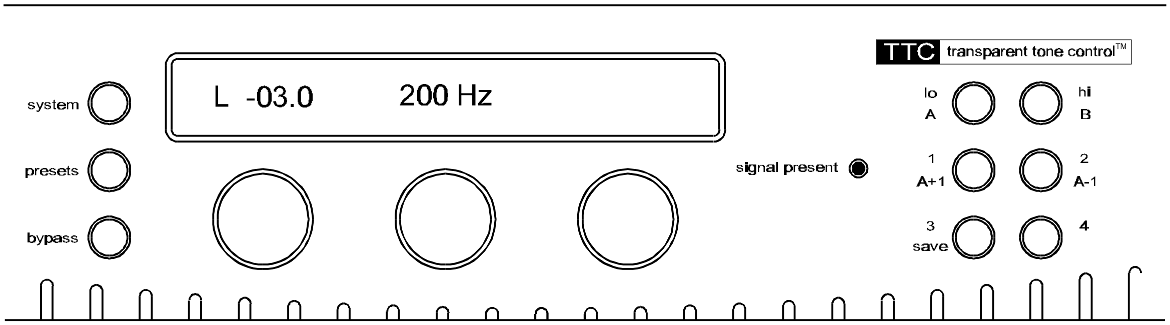

FIGURE 4

• digital volume control

The rdq-1 features a digital-domain volume control. This control allows you to apply between +12 dB of boost and -12 dB of cut to the incoming digital signal. You may have heard arguments that digital-domain volume control is undesirable because it leads to loss of resolution. This may be true for poorly implemented digital volume controls but certainly is not the case for the rdq-1. Remember the discussion of dither: we can achieve resolution well below the least significant bit. This is the key to the rdq-1's digital volume control (please refer to appendix B for a more detailed explanation).

How to do it. With the rdq-1 in normal operating mode, press the **system** button, which will result in the display as shown in Figure 5. To change the volume, turn the knob below the numeric display showing the volume.

A special note about the rdq-1's volume control. The rdq-1 always powers up to a default volume of 0 dB. Furthermore, volume is the only parameter not stored or recalled by the load and save functions, nor is it affected by the **bypass** control. This is for an important reason. Dangerous output levels could result when using the rdq-1 in a system with no analog preamplifier. For example, suppose we designed the rdq-1 so that the bypass control returned the volume to 0 dB. Hitting **bypass** during a loud passage would then result in the system volume going to full-scale. The same could happen while comparing presets. It is for this reason that the volume setting is unaffected by **presets** and **bypass**; whatever the value of volume is dialed in will remain in place during preset loads or during bypass.

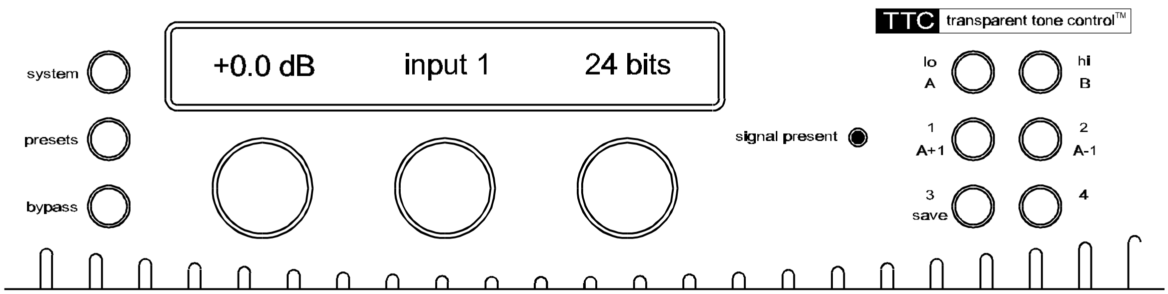

FIGURE 5

• preset control

The rdq-1 gives you the ability to store and recall up to 100 configurations. Furthermore, there are a number of useful functions that allow you to compare presets against one another. This section describes how to use the rdq-1's preset facility.

<u>How to do it.</u> Suppose you have dialed in the rdq-1's controls to a setting you find pleasing: you are happy with the bell filters, the shelf filters, the input, and the dither. You can save this setting so it can be recalled at a later time or so that you can experiment further without worrying about not being able to return to this state. To save this setting, press the **preset** button twice. The rdq-1's display will appear as in Figure 6. Use the knob beneath the numeric display to choose a preset number. Once you select a number, press the **save** button (this function is indicated in blue below the **3** button) to complete the process. To return to normal operation, press the preset button one more time.

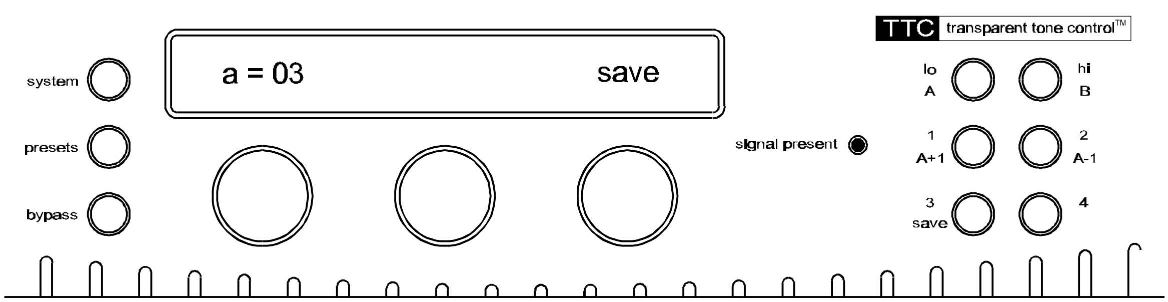

FIGURE 6

Now that we know how to store presets, what about recalling them? Assuming we are in the normal mode of operation, press the **preset** button once to get to "load preset" mode. The display will appear as in Figure 7. Notice that there are two numbers shown: **a** and **b**. These two numbers allow you to compare two presets against one another. For example, suppose you previously created two presets that you liked and saved them as presets 7 and 4, respectively. To compare them with one another, use the knobs beneath the displays to set **a** to 7 and **b** to 4. To listen to preset 7 (the **a** preset), press the **a** button. This will load preset 7 and put an arrow next to the a display. To listen to preset 4, press the **b** button. This will load preset 4 and put an arrow next to the **b** display. You can toggle between presets 7 and 4 by pressing the **a** and **b** buttons.

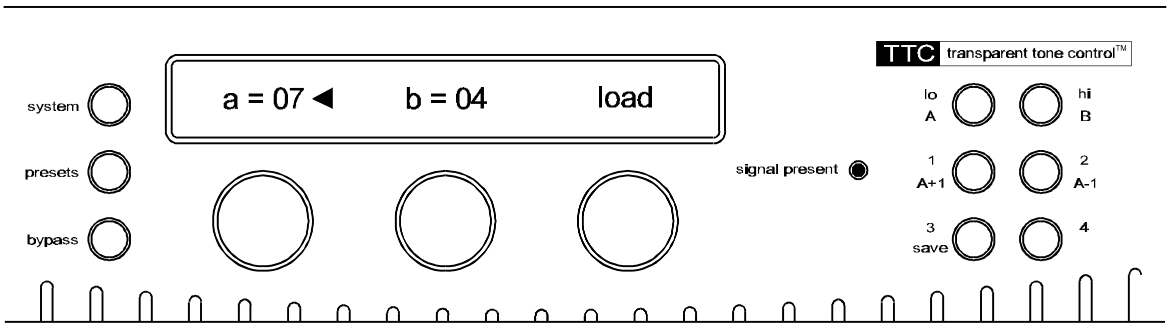

FIGURE 7

There are two other useful functions in the load preset mode: increment preset and decrement preset. Suppose you have the **a** preset set to 4. To listen to the next preset, press the **a+1** button. The **a** display will change to 5 and preset 5 will be loaded. To return to preset 4, press the **a-1** button. This will return the **a** display to 4 and preset 4 will be loaded.

It is important to note that **a** and **b** are nothing more than "placekeepers." They can be set to any values, even the same values. The only reason we have provided the **b** preset is for simple comparison between two settings. For example, suppose you wish to compare preset 1 with preset 99; this would be difficult without the a/b controls.

You should make note of two particular preset numbers. Loading preset number 0 will return the rdq-1 to a "flat" setting. The flat setting is characterized by:

- *all bell and shelf filters disengaged*
- *24-bit output*

Note that preset 0 will not change the active input from the one presently engaged. All other presets will change the active input to the input indicated by the chosen preset.

Preset 99 is important because it is the user-default preset. In other words, preset 99 will be loaded automatically when the power is turned on. Therefore, you should store in preset 99 the settings you want to be in effect (active input, dither, and filters) when the unit wakes up.

• bypass mode

Hitting the **bypass** button results in the state shown in Figure 8. The bypass mode sets all of the filters flat and cancels the dither; only the volume is left unchanged. When in bypass mode, none of the rdq-1's controls are functional; you must press **bypass** again to return to normal mode.

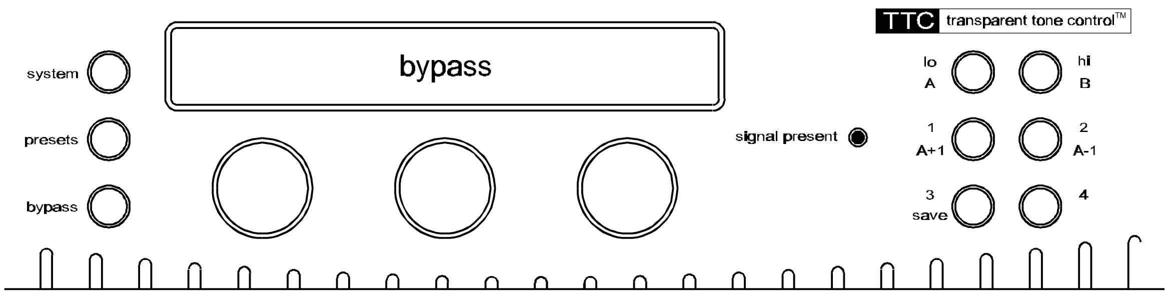

FIGURE 8

a special note on performing A/B comparisons.

The italicized text above makes it clear that the bypass control returns the output wordwidth to 24 bits, canceling the dither that may have been selected. Perhaps most significantly, the bypass mode does not affect the volume. We do this because we want it to be impossible for the user to accidentally bring the volume to full-scale. Because the bypass mode cancels the dither, hitting bypass when the volume is set to anything other than 0 dB can result in very subtle sonic differences due to truncation, depending on the program material and the volume. Therefore, doing a level-matched comparison between a given EQ setting and flat EQ using the bypass control is not really meaningful unless the volume is set to 0 dB. A better approach is to set up a preset with all of the filters set flat and the dither set as desired and use that preset in the A/B-compare mode.

listening to the rdq-1

The rdq-1 is designed to be placed between the digital output from your CD transport and the digital input to your digital-to-analog converter (DAC). It can also be placed in the processing loop of a one-piece player, assuming the player allows its digital output to be connected to its digital input. You may either of two digital interfaces with the rdq-1: AES/EBU on balanced XLR or S/PDIF on unbalanced RCA. The rdq-1 can be used to convert between these digital formats; both of its digital outputs are active whenever a valid input is selected.

▪ input/output

With the power turned off, determine which digital input and output you will be using, and connect the digital cable from the CD transport's digital output to the rdq-1's digital input. Be sure to note the input number you're using. Then connect the appropriate digital output of the rdq-1 to the digital input of your DAC, make sure your DAC is set to accept input from the rdq-1, and press **play** on your transport. Turn the power on, but don't worry if you don't hear any audio yet. Press the **system** button and you will see that the middle knob acts as an input selection knob. Turn this knob until the input display shows the input number corresponding to the input to which you connected your transport. The signal present LED on the rdq-1 should light up, giving you visible indication that you've configured the input correctly.

Similarly, your DAC should give visible indication that it is receiving a digital signal correctly. Assuming your preamplifier and amplifier are turned on and your speakers are connected, you should be hearing signal. The right knob will show the output wordwidth setting, which is initially configured for 24 bits. Without getting into a long discussion, we recommend you turn the knob so that the display shows **20 dith**, which stands for "20-bit dithered." The sonic differences between output wordwidths are subtle, but the 20-bit dithered setting works well with the current state-of-the art in DACs. You can play with the output wordwidth control at your leisure and pick a setting that sounds best to you.

▪ volume

The default volume when you first powered up the rdp-1 was 0.0 dB. To increase or decrease the volume, simply hit the **input** button and turn the volume knob up or down.

▪ equalization

(Please consult the appendix A for a more thorough discussion of equalization.) You are probably already familiar with the graphic equalizer, which is very popular in car stereo applications. A graphic EQ consists of a bank of filters with fixed center frequencies, fixed slopes, and variable gain. The graphic EQ in your car probably has five or six sliders. Each slider allows you to control the gain of one of the EQ's filters, in essence giving you coarse control over the bass, midrange, and treble. Because the slopes of the filters and their center frequencies are fixed, a graphic EQ gives you very limited control over a system's tonal balance and is of very limited use for room correction.

Unlike a graphic EQ, the rdp-1 has two types of filters (bells and shelves) and gives you full parametric control over these filters' slopes, gains, and center frequencies. Bell filters are used to emphasize or de-emphasize energy at a range of frequencies centered about a center frequency. Shelf filters are used to emphasize or de-emphasize all of the energy at frequencies above or below a corner frequency. For example, a bell filter could be used to boost by 2 dB a 500 Hz-wide region of the spectrum centered at 1 kHz. Or, a shelf could be used to cut by 3 dB all of the spectrum below 200 Hz. The best way to become familiar with the sonic attributes of equalization filters is by experimentation.

The rdp-1 can simultaneously process four bell filters and two shelf filters (one low shelf and one high shelf). The parameters for each filter are displayed by pushing the corresponding filter selection button from the rectangular group of six buttons on the right side of the front panel. Pushing a filter selection button lights an LED next to that button and also "brings the focus" of the three knobs to that corresponding filter. When you push a different filter selection button, the display changes accordingly and the knobs now affect the new selected filter. *This does not mean the other filter is no longer in the signal path. Rather, the display and knobs can only control one filter at a time*.

For this demonstration, we recommend a good recording of a small acoustic jazz combo. Let's begin with the hi shelf filter. Hit the hi button and you will see two parameters on the screen: the gain and the corner frequency. Turn the gain knob until the display shows –12 dB then turn the corner frequency knob until the display shows 1 kHz. The rdp-1 is applying a filter to the audio signal that is flat from 0 Hz to somewhere near 1 kHz, drops to –3 dB at 1 kHz, then slopes downward until it hits –12 dB, where it then levels off. You should hear a distinct “muffling” effect, similar to a blanket being placed over the speakers. In particular, notice how “blurred” transient attacks sound, how indistinct cymbals sound, and how the saxophone loses that “breathy” quality that makes it sound real. Now begin to turn the corner frequency knob clockwise, increasing the corner frequency. You will hear the sound begin to “open up” as the corner frequency moves higher and higher. This is because you are moving to a higher point in the spectrum the frequency at which the shelf filter begins to attenuate. This allows progressively more high-frequency information to get through. Be sure to pay attention to how the sound stage begins to snap into focus as you let more of the highs through. You can perform a similar experiment with the low shelving filter; cut the low frequencies by 12 dB and listen to what happens as you move the corner frequency backwards from 1 kHz to 28 Hz: you’ll hear progressively more and more bass.

Next we’ll experiment with a bell filter. First, make sure you’ve returned the high and low shelf filters to 0 dB; successively press the **hi** and **lo** buttons and turn the respective gains to 0 dB. Press the **1** button to invoke bell filter 1. Adjust the filter so that it shows a gain of –12 dB, a center frequency of 1 kHz, and a slope of 1. Turn the center frequency knob counter-clockwise. As you progressively move lower from 1 kHz, you’ll hear a “suckout” that moves with the filter’s center frequency. Notice, for example, how an acoustic bass loses all of its body once the center frequency gets to 200 Hz. Now turn the frequency knob clockwise until the filter is centered at 2.8 kHz. Listen carefully to the effect this has on the piano. Notice how you can get a mental picture of the "shape" of this filter. You should still be able to hear low frequencies; the bass should sound similar to when the filter is not engaged. You should still be able to hear high frequencies; cymbals will still have a “shimmer” to them. Only a region centered about 2.8 kHz has been affected -- the highs and lows are still present.

The **slope** parameter controls how *wide* a region of frequencies is affected by a bell filter. For example, a bell filter centered at 1 kHz with 6 dB of cut (i.e., -6 dB of gain) and a slope of 1 will have -3 dB points at roughly 450 Hz on the low side and 2.2 kHz on the high side. With a slope of 3, the -3 dB points change to roughly 700 Hz and 1300 Hz. As you can see, the -3 dB points for the bell filter move closer to the filter's center frequency as the slope increases. In other words, a high value of slope means that the filter moves from "flatness" to the point of maximum boost or cut very quickly. Let's try an experiment to illustrate this effect. Adjust a bell filter for a center frequency of 1.6 kHz, a gain of -12 dB, and a slope of 1. Now, use a test disc to play a 1 kHz sine wave. If the center frequency of the bell filter were at 1 kHz, the sine wave would be attenuated by exactly 12 dB. Because the sine wave is located in frequency below the 1.6 kHz center frequency of the bell filter, it will be attenuated by less than 12 dB. As you increase the slope of the filter, the 1 kHz sine wave will get progressively louder. This is because as the slope increases, the -3 dB points move closer to 1.6 kHz. For a slope of 12, the attenuation at 1 kHz isn't more than a few tenths of a dB.

Now play some musical material. Bring the center frequency back to 2.8 kHz with a slope of 1 and a gain of -12 dB. Increase the slope to hear what this does sonically -- it sounds as if the filter is being removed from the signal path. As the slope increases, more of the midrange returns and so do the high frequencies. Once the slope is increased beyond a value of 5, it’s difficult to hear any substantial tonal difference; higher values of slope are more useful for surgical treatment of room problems than for tonal adjustment. We recommend using low values of slope for tone control.

Return the slope to a value of 1 and listen to the effect of changing the **gain**. As the gain is turned lower and lower, we are removing more and more of the audible spectrum around 2.8 kHz. If we bring the gain past 0 dB and into the positive region, we're increasing the amount of energy in the spectrum around 2.8 kHz. This will tend to add more presence, but beyond a certain point the sound will get overly glaring.

There is no substitute for listening and experience. Much in the same fashion that you grow accustomed to the keystrokes required of a new piece of software, you'll soon grow accustomed to the rdp-1's ergonomics. And believe it or not, although the concept of equalization may seem somewhat intimidating at first, you will be able to adjust the tonal balance to your own liking in a matter of days.

appendix A – equalization and dither

• equalization

The TTC consists of two types of filters: bells and shelves. These filters derive their names from their frequency-domain shapes. The bell filter is characterized by three parameters: the center frequency, the amount of cut (or boost), and the bandwidth. Bell filters are typically used for applications such as resonance control, room mode control, and narrow-band noise reduction, to name a few. The bell filter has a typical gain of 0 dB (unity gain) with a maximum cut of -g dB occurring at the center frequency. The bandwidth, expressed in Hertz, is defined as the width of the filter at its -3 dB points. It is typical to express the bandwidth of a bell filter in terms of its *quality factor*, Q. The Q parameter is defined as the ratio of the filter's center frequency, fc, to filter's bandwidth, BW. Formally, we have the relation

$$Q = \frac{f_c}{BW}$$

We can see from this equation that for a given center frequency, as we make the bandwidth smaller, the value of Q increases. The same discussion holds valid for bell filters in boost mode rather than in cut mode. Rather than describing the bell filters' bandwidths in terms of Q, we have chosen to display on the front panel a parameter we call "slope." Slope is related to Q in a straightforward fashion, as detailed in Appendix B.

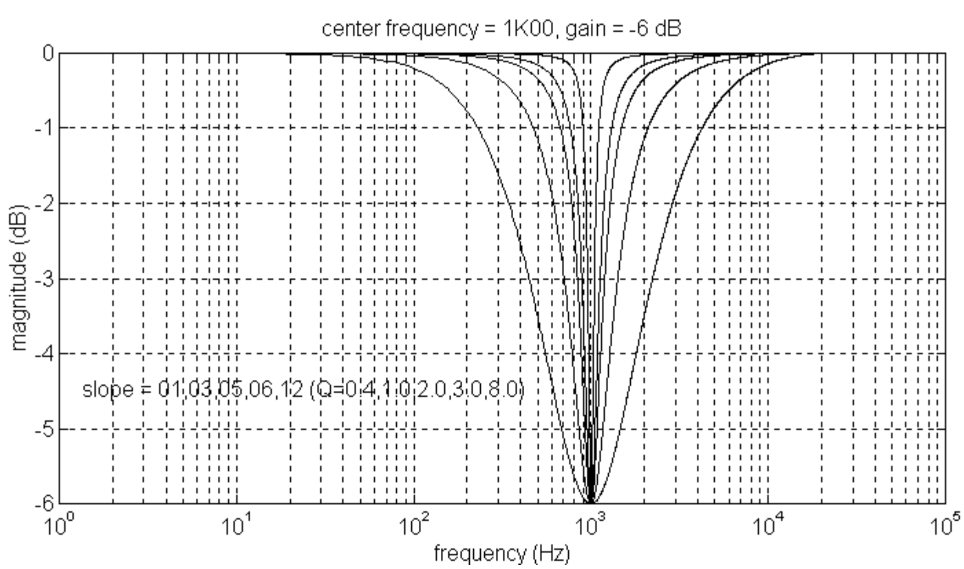

FIGURE 9

For most users, it will suffice to understand how the slope parameter works in a qualitative sense. Consider the widest filter shown in Figure 9, which has a center frequency of 1000 Hz and a cut of –6 dB. As we vary the slope while keeping the center frequency and gain constant, the bandwidth of the filter changes. As the slope increases, the bandwidth decreases (that is, the filter gets narrower). On the rdq-1, the slope varies from 1 to 12, with a slope of 1 giving the widest bandwidth, and a value of 12 giving the narrowest bandwidth. Figure 9 illustrates this relationship clearly -- as we increase the slope parameter, the 3-dB points will gradually move toward 1000 Hz from the left and from the right and the filter gets narrower and narrower.

Whereas Figure 9 showed the effect of changing the slope while holding the center frequency and gain constant, Figure 10 shows the effect of holding the slope and gain

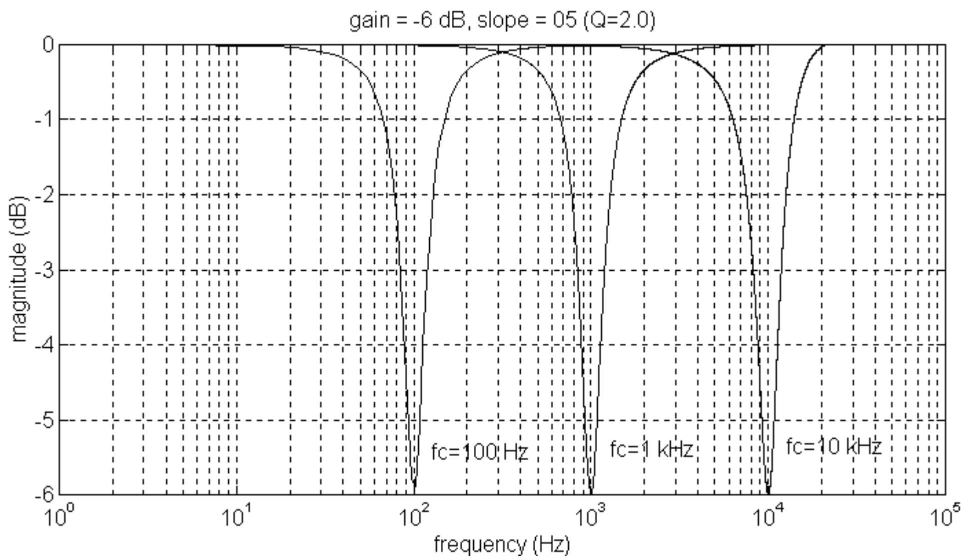

FIGURE 10

constant while moving the center frequency. Figure 10 shows a family of frequency response curves for a slope equal to 05, a gain of –6 dB and center frequencies of 100 Hz, 1 kHz, and 10 kHz, respectively. Notice that the filters have identical shapes when viewed on a logarithmic frequency axis.

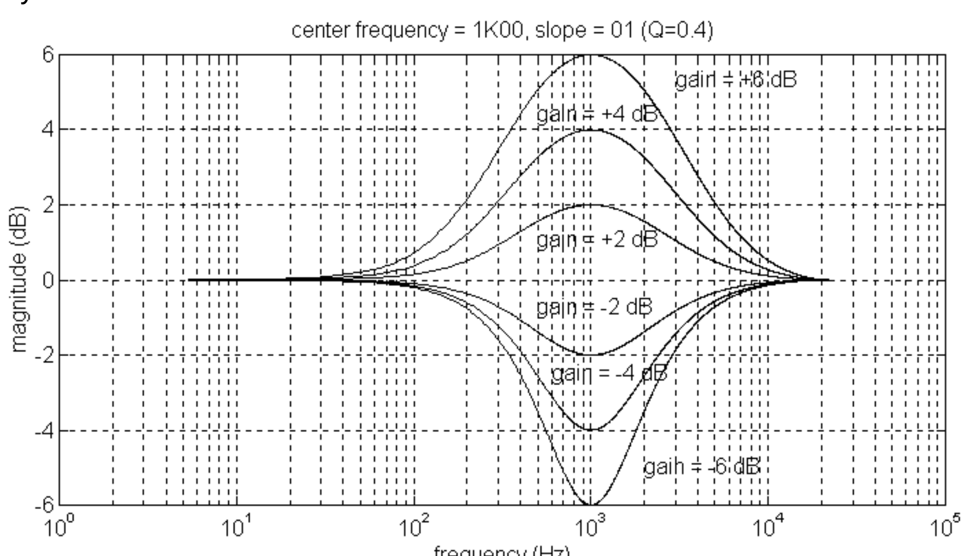

FIGURE 11

Finally, Figure 11 shows the effect of holding the slope and the center frequency constant while varying the gain. In particular, the center frequency is 1000 Hz, the slope is the minimum allowed, 01, and the gain is varied between –6 dB and +6 dB.

So far, we've only discussed bell-shaped parametric filters. We will now turn our attention to shelving filters. Whereas bell-shaped filters are used to emphasize or de-emphasize a range of frequencies in a bell-shaped region about a center frequency, a shelving filter is used to emphasize or de-emphasize *all* of the frequencies above or below a desired corner frequency. A shelving filter is characterized by a pair of parameters: the corner frequency and the gain. For example, consider the shelving filters shown in Figure 12. These curves correspond to a low-shelving filter with a corner frequency of 500 Hz and gains of -2 dB, -6 dB, and -12 dB, respectively. Ideally, a low-shelving filter would be flat until the corner frequency and drop to a value specified by the gain for all frequencies below the corner frequency. In practice, these filters have a finite slope, as shown in Figure 12.

The corner frequency is tough to "eyeball" from these plots. Loosely speaking, it is the frequency at which the magnitude response deviates by 3 dB from the 0-dB reference (which only makes sense for filters that have shelves lower than -3 dB). For example, the bottom curve in Figure 12 is down by 3 dB at roughly 500 Hz. Below 500 Hz, the curve continues to slope downward, approaching flatness at a level of -12 dB. We have not shown a low shelving filter with boost instead of cut, but this too is permissible.

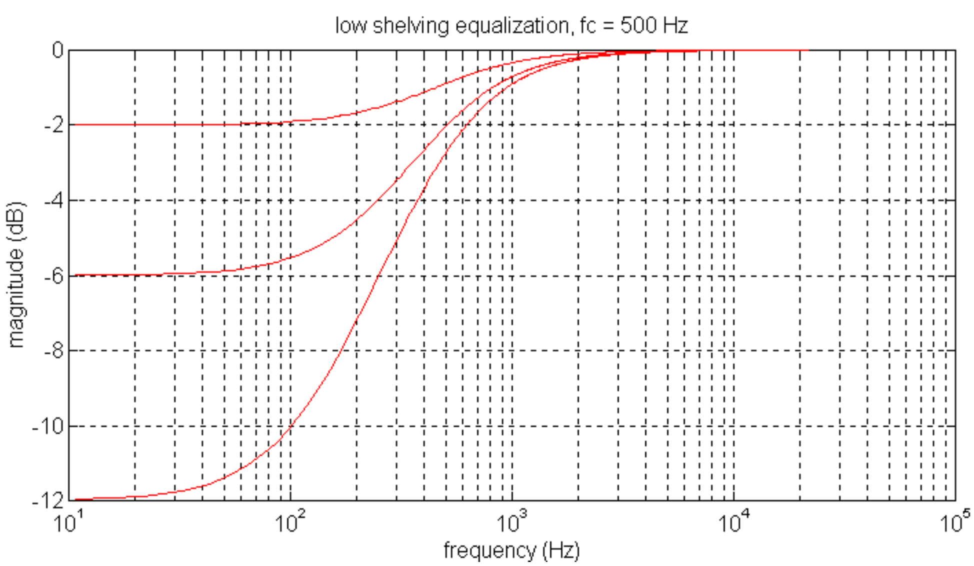

FIGURE 12

A high-shelving filter is similar to a low-shelving filter. The only difference is that frequencies *above* the corner frequency are boost or cut. Figure 13 shows a family of curves for a high shelving filter with a corner frequency of 5 kHz.

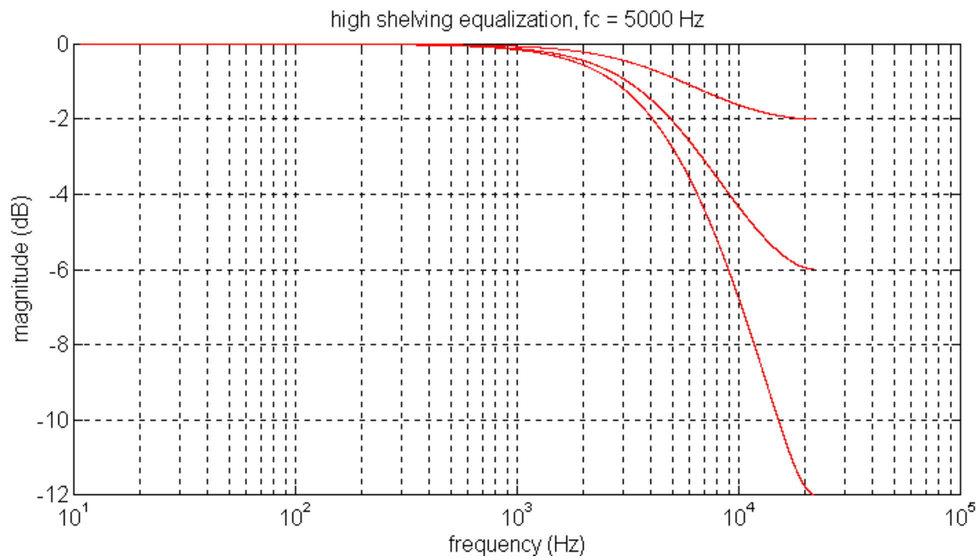

FIGURE 13

• dither

In order to understand dither, we must briefly discuss the problems associated with quantization and re-quantization. First, quantization is the process whereby a continuous-valued analog signal is converted into a discretely valued digital signal. In the case of compact disc digital audio, an audio signal is represented by 16-bit samples taken 44,100 times per second. A 16-bit system comprises 2^{16}=65,536 quantization steps. Thus, 16-bit quantization consists of sampling the analog waveform uniformly in time and approximating each sample by the nearest of the 65,536 quantization steps. Conventional quantization begins to show its weakness on source material with wide dynamic range (i.e., a very high ratio between the levels of the loudest sounds and the softest sounds, typical of orchestral music). In order to make room for the loudest sounds, the quietest sounds are "pushed" below the smallest quantization step, leading to signal-dependent quantization noise. This distortion can manifest itself in many different ways including harshness, edginess, loss of reverberant detail, and alteration of timbre. Loosely speaking, then, a digital audio system's performance is bounded by a level that is established by the size of the quantization step used.

This same type of distortion can occur when doing digital-to-digital operations, as well. For example, to ensure a high-quality end product, many compact discs are mastered at 20 bits or 24 bits as an intermediate format for editing, equalization, compression, and level adjustment. One of the final steps of the mastering process is to convert the 20- or 24-bit digital signal to a 16-bit signal for final release on the CD format. The problems here are more or less the same as those associated with quantization of an analog signal. The 20- or 24-bit signal has much wider dynamic range than the 16-bit signal and signal-dependent distortion can occur when making the transfer to 16 bits. In this case, the mastering engineer must deal with the problems associated with *re-quantization* noise.

Any type of DSP algorithm such as volume control or tone control involves mathematical operations on an input signal which result in the accumulation of extra bits. For example, the internal result of the rdq-1's volume control is, in general, 40 bits, whereas the CD source material feeding it is only 16 bits. Ideally, we would like to present the digital-to-analog converter (DAC) with a signal having the proper bit-width. For today's 20-bit converters, we need to somehow "map" the 40 internal bits to a 20-bit result that can be used by the converter. How we do this can impact the sound dramatically. In fact, one of the keys to successful audio DSP is dealing with the accumulated extra bits.

This is where dither comes in. Dither is an extremely low-level signal that is used to ameliorate some of the sonically disturbing effects of quantization or re-quantization. There are many different "flavors" of dither, all of which have their own strengths and weaknesses. In its simplest form, dither is a very small noise signal that is added to the original signal before quantization or re-quantization. The net result of dither is to decorrelate the quantization or re-quantization error from the signal itself, which helps improve the low-level characteristics of the system. It may seem paradoxical that the key to eliminating digital distortion is through the addition of noise, but this is indeed the case. Dither will increase the system's overall noise level, but the resulting noise floor is much more like an analog noise floor. Interestingly, in a properly dithered system signal details far smaller than the smallest quantization step can be heard, giving an effective dynamic range well beyond the 96 dB commonly attributed to the compact disc medium. This has profound sonic implications, including but not limited to timbral neutrality, stereo image, and reverberant detail. Judicious use of dither is one of the keys to the rdq-1's unrivaled sonic performance.

Let's look at a simple example. Figure 14 shows the spectrum of a 40-bit 5512 Hz sine wave sampled at 44.1 kHz, with an amplitude of –96 dB from full-scale digital. Notice that all of the

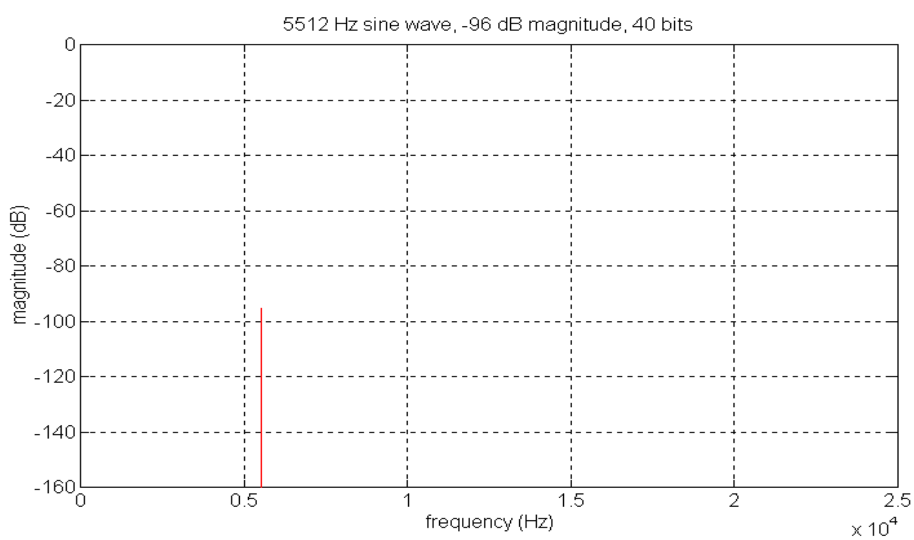

FIGURE 14

signal's power is concentrated at the frequency of the fundamental: 5512 Hz. There is actually a small amount of power across the rest of the audio band due to the quantization to 40 bits, but it is *well* below –160 dB in magnitude.

What happens when we convert the 40-bit sine wave to a 16-bit sine wave by means of truncation (that is, by throwing away the lower-order bits)? Figure 15 shows the spectrum of the truncated sine wave. Observe that there are harmonically related features all up and down the spectrum. When we listen to this signal, it will sound harsh and "buzzy," like a square wave. This makes sense intuitively: -96 dB undithered from full scale leaves us with essentially two quantization steps, which looks like a very low-level square wave. Now, suppose that before truncating to 16 bits we add a random noise signal (the dither) to the 40-bit sine wave where the noise has a peak magnitude of approximately –84 dB from full scale. This has the effect of "stimulating" more quantization levels. The result is the spectrum shown in Figure 16. Observe that the harmonic distortion evident in Figure 15 is no longer present.

It is important to understand that we haven't gotten something for nothing. We've traded harmonic distortion for an elevated noise floor. In Figure 15, the non-harmonic components are, for the most part, below –140 dB in magnitude. In Figure 16, we can see that the noise level has risen to roughly –125 dB. This will be perceived as a very slight hissing sound, much like analog tape noise. This hissing, however, is much more benign than the harmonic distortion induced by truncation. Upon listening, the signal in Figure 15 is not recognizable as a sine wave. The signal in Figure 16, however, sounds like a sine wave in a low-level hissing background. The same can be said for other low-level signals, not just laboratory-produced sine wave test signals.

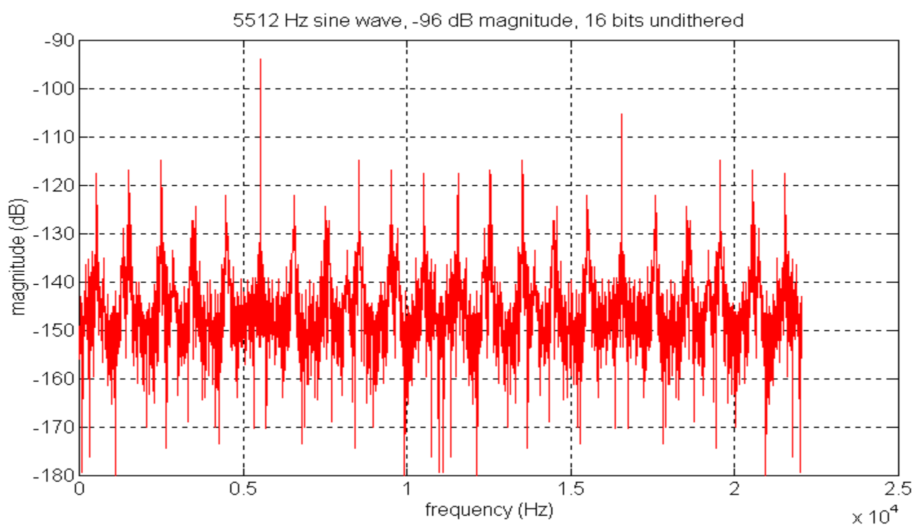

FIGURE 15

This example only showed the effect of "flat" dither, which is dither with a flat power spectrum. There are other types of dither that have different shaped power spectra. For example, it is possible to "frequency-shape" the dither so that most of its power is contained in the high frequencies where the ear is less sensitive to it. There are also other dithers that have more complicated spectral shapes. Whatever the flavor of dither used, its purpose is the same: to reduce the distortions incurred whenever a signal needs to have its number of bits reduced.

This brings us back to the discussion of digital volume control. How is it possible, then, for the rdq-1 to control the volume in the digital domain without any significant loss of resolution? The answer has to do with dither. Suppose you are playing a 16-bit CD through the rdq-1 and

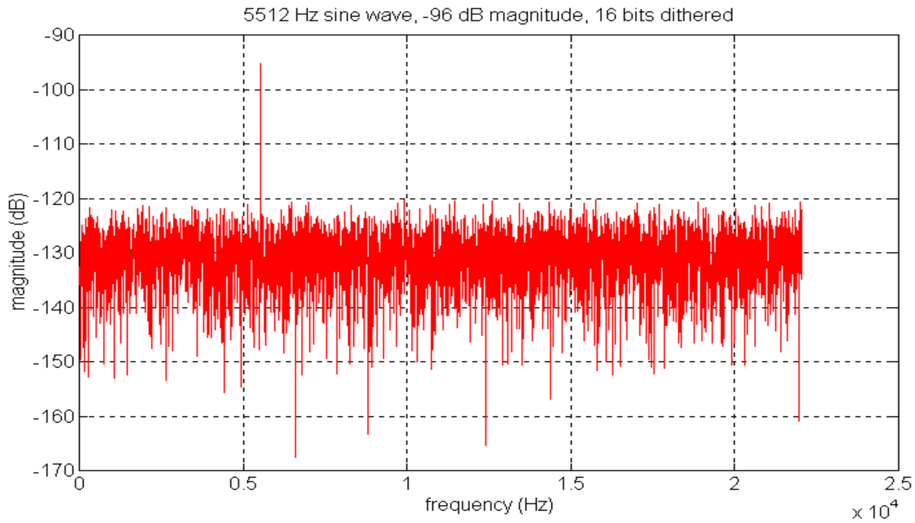

FIGURE 16

into a 20-bit DAC. With the volume set to 0.0 dB, the 16 bits of the CD are aligned with the 16 most significant bits (MSBs) of the DAC. Roughly speaking, attenuating the signal corresponds to "shifting the bits" to the right. Those against digital volume control would argue that the loss of resolution begins to happen when the least significant bit (LSB) of the CD data "slides" past the LSB of the DAC; we call this condition *underflow*. In the early days of digital, when 16-bit DACs were the norm, even the smallest amount of attenuation could cause the data to underflow the DAC. Today, with 20-bit DACs in common use, up to 24 dB of attenuation can be made without DAC underflow. And, as the dither discussion illustrates, resolution well beyond the LSB is possible; we typically claim a number of 12 dB beyond the LSB. This "extension" beyond the LSB, in conjunction with the 20-bit DAC, allows for a full 36 dB of attenuation before any loss of resolution begins to occur. This is a *significant* amount of attenuation. With a 24-bit DAC, a full 60 dB of attenuation is possible without any loss of resolution.

appendix B: the relationship between Q and slope

The relationship between slope and Q is fairly straightforward but requires a bit of preliminary explanation. Appendix A gives some background on the unique digital filter structure that the rdq-1 uses for its parametric equalizer filters. This structure leads to the lowest-noise, best–sounding digital filter possible. One of the few complications this structure imposes, however, is a limitation on the range of possible bandwidths that can be realized.

With Q defined as

$$Q = \frac{f_c}{BW}$$

where BW is the 3-dB bandwidth in Hertz and f_c the center frequency in Hertz, we start with a *nominal* minimum Q_0, equal to 0.4. The first slope value displayed by the rdq-1, “slope: 01,” corresponds to a Q equal to Q_0. Subsequent values of slope give multiples of Q_0. Specifically, we have

slope: 01 corresponds to Q_0
slope: 02 corresponds to $2Q_0$
slope: 03 corresponds to $2.27Q_0$
slope: 04 corresponds to $3.25Q_0$
slope: 05 corresponds to $5Q_0$
slope: 06 corresponds to $6.25Q_0$
slope: 07 corresponds to $7.5Q_0$
slope: 08 corresponds to $10Q_0$
slope: 09 corresponds to $12.5Q_0$
slope: 10 corresponds to $15Q_0$
slope: 11 corresponds to $17.5Q_0$
slope: 12 corresponds to $20Q_0$.

For the first 49 values of center frequency, Q_0 is equal to 0.4. For the last eight center frequencies, Q_0 is different. This is because of the limitations placed on the range of possible bandwidths for our special filter structure. For reference, we have

For f_c < 8K00, Q_0 = 0.40
For f_c = 8K00, Q_0 = 0.44
For f_c = 9K00, Q_0 = 0.48
For f_c = 10K0, Q_0 = 0.55
For f_c = 11K2, Q_0 = 0.61
For f_c = 12K5, Q_0 = 0.68
For f_c = 14K1, Q_0 = 0.76
For f_c = 16K0, Q_0 = 0.83
For f_c = 18K0, Q_0 = 0.90.

appendix C: finite wordlength digital filtering

Two of the primary challenges facing designers of digital filters are dealing with quantization error and dealing with roundoff noise. Quantization error refers to a digital filter's sensitivity to coefficient quantization within a fixed-wordlength system, as occurs any time a real-world processor is used to implement the digital filter. Roundoff noise is a more difficult problem to treat because of the cumulative numerical precision problems that arise whenever a recursive or "recirculant" algorithm is used to implement the filter. Both of these problems can occur with fixed- and floating-point processors.

In a nutshell, the types of digital filters that found in audio equalization are implementing a difference equation of the form

$$y(k) = b_0x(k) + b_1x(k-1) + b_2x(k-2) – a_1y(k-1) – a_2y(k-2) \qquad \text{(Equation 1)}$$

where x(k) is the input and y(k) is the output [1]. The frequency-selective behavior of the filter is determined by the set of parameters $\{b_0,b_1,b_2,a_0,a_1\}$ and "recipes" for determining these parameters as a function of center frequency, bandwidth, and gain are readily available. One can readily see from Equation 1 that the difference equation is recursive – there is feedback. In the presence of an impulse (a signal, x(k), that is equal to 1 for k=0 and equal to 0 for all other times), the difference equation will continue to produce a non-zero output signal since future values of y are computed as a function of past values of y. This is known as an *infinite impulse response* (IIR) filter. One can also see how computational errors as a result of finite precision can recirculate through the filter.

There are many different ways to compute the difference equation given by Equation 1. For example, there are different ways to arrange the order of summation and products, or there are alternative formulations to realize the same output y(k) as a "rearrangement" of different intermediate sums and products. Whatever the realization, the order and structure of computations is referred to as a "filter architecture."

The textbook architecture for realizing Equation 1 is known as the direct-form architecture. The direct form architecture has two principal advantages. First, it is simple to derive from the canonical difference equation above. Second, it is simple to implement and can be run in real-time with very few instructions. Unfortunately, the direct-form has some significant drawbacks: abysmal quantization and roundoff noise performance. In many applications, the roundoff noise is so large that the direct-form filter architecture is all but unusable. For example, consider implementing a peaking filter. It turns out that the roundoff noise power is a function of the bandwidth, center frequency, and gain. As the bandwidth decreases, the noise increases. As the gain increases, the noise increases. As the center frequency decreases, the noise increases. In fact, implementing a narrow filter at low center frequencies with any peaking is all but impossible with the direct-form architecture.

There is a counterintuitive axiom that experienced digital filter designers are familiar with. Filter architectures that are simple to implement (few multiplies and adds) tend to have poor finite-wordlength performance. Architectures tend to need to be complicated in order to perform well from the standpoint of roundoff noise. This was recognized in the early days of digital signal processing when only small-wordwidth fixed-point processors were available and still holds true today, even for larger-precision floating-point processors. Other filter architectures, such as the wave structure, the cascade structure, and the lattice-ladder structure arose in an attempt to make better filters.

A logical question arises. Is there any way to derive a filter structure that can produce the guaranteed minimum roundoff noise power and minimum sensitivity to coefficient quantization given a fixed, known wordwidth with which to work? The answer is in the affirmative, although it is complicated to derive or even explain (the interested reader is referred to [2] or [3]). Suffice it to say that it is possible to derive an architecture that is optimal in the above sense. There are several solutions to this problem, and they differ slightly as a function of the optimality criteria they use.

The solution we have derived has a number of useful properties. Most significantly, our architecture realizes the absolute minimum possible roundoff noise power and sensitivity to coefficient quantization. Equally interesting, our architecture *always* produces this minimum roundoff noise power, *independent* of gain, center frequency, or bandwidth. For certain highly reasonable filters (e.g. center frequency of 28 Hz, cut of 10 dB, with a Q equal to 8), the roundoff noise power of our architecture is as much as *4 or 5 orders of magnitude* better!

1. Oppenheim, A. V., and Schafer, R. W., *Discrete-Time Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, 1989.
2. Zelniker, G. S., and Taylor, F. J., *Advanced Digital Signal Processing: Theory and Applications*, Marcel Dekker, NY, NY, 1994
3. Roberts, R. A., and Mullis, C. T., *Digital Signal Processing*, Addison-Wesley Publishing Company, Reading, MA, 1987.

specifications

- Inputs: 1 AES/EBU (transformer-isolated, 110-ohm terminated), 1 S/PDIF (transformer-isolated, 75-ohm terminated).
- Outputs: 1 AES/EBU, 1 S/PDIF
- Input/output precision: up to 24-bits
- Gain control: from -12 dB to +12 dB
- Gain resolution: 0.2 dB increments from +12 dB to -12 dB
- Filter types: 4 parametric, 2 shelving
- Center frequency resolution: 1/6-th octave ISO from 28 Hz to 18 kHz
- Filter gain/cut: from -12 dB to +12.0 dB
- Filter bandwidths: 12 steps from Q = 0.4 to Q = 8.0 (nominal)
- Shelf filter slopes: 6 dB/octave
- Dither types: 20-bit and 16-bit proprietary floating-point techniques
- Digital filter architecture: proprietary minimum roundoff-noise structure
- Dynamic range: better than 144 dB
- THD+N: better than -135 dB
- Processor type: TMS320C31 32-bit floating-point DSP
- Processor performance: 60 MFLOPS
- Digital audio demodulator/modulator: Crystal Semiconductor CS8412/CS8402
- Digital audio signal transformers: Scientific Conversions SC916-01, SC944-05
- Sample rates supported: 32 kHz, 44.1 kHz, 48 kHz. 88.2 kHz/96 kHz upgrade available from factory
- Auxiliary-bit status handling: unit is transparent to channel status, validity, and auxiliary bits
- Stereo-linked and dual-mono operation
- Number of presets: 99
- Power supply: 110/220VAC, 50/60Hz