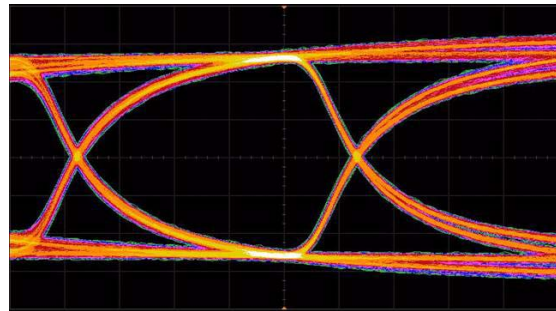




Agilent N5461A Infiniium Serial Data Equalization



User's Guide

Notices

© Agilent Technologies, Inc. 2009

No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Agilent Technologies, Inc. as governed by United States and international copyright laws.

Manual Part Number

1N5461-97000

Edition

Second edition, December 2009

Agilent Technologies, Inc.
1601 California Street
Palo Alto, CA 94304 USA

Warranty

The material contained in this document is provided “as is,” and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied, with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement shall control.

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

Restricted Rights Legend

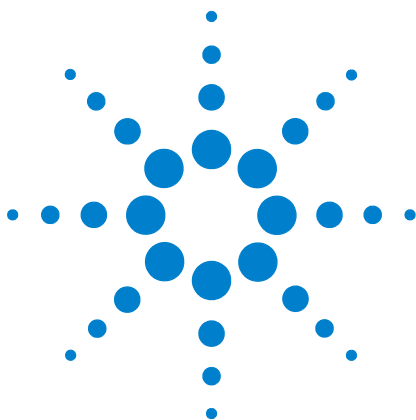
If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as “Commercial computer software” as defined in DFAR 252.227-7014 (June 1995), or as a “commercial item” as defined in FAR 2.101(a) or as “Restricted

computer software” as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause. Use, duplication or disclosure of Software is subject to Agilent Technologies’ standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

| | | |
|----------|--|----|
| 1 | License Information | |
| 2 | Theory / Background | |
| | The Problem | 7 |
| | Inter-symbol Interference (ISI) | 9 |
| | The Solution: Equalization | 10 |
| 3 | Feed-Forward Equalization Theory | |
| | Introduction | 13 |
| | Mathematical Description of FFE | 14 |
| | Description of FFE Acting on a Signal | 15 |
| | FFE Decision Diagram | 17 |
| | Understanding FFE Through Frequency Response Plots | 18 |
| 4 | Continuous Time Linear Equalization | |
| | Introduction | 21 |
| 5 | Decision Feedback Equalization Theory | |
| | Introduction | 23 |
| | Mathematical Description of DFE | 24 |
| | Description of DFE Acting on an Eye Diagram | 26 |
| | DFE Decision Diagram | 28 |
| | Understanding DFE Through Frequency Response Plots | 29 |
| 6 | Taps | |
| | Introduction | 33 |
| | How Many Taps Should You Use? | 33 |
| | How to Enter More Than Sixteen Taps | 34 |
| | How to Determine Which Values to Use For Your Taps | 36 |
| 7 | Equalization Wizard | |
| | Introduction | 45 |
| | Step 1: Select Configuration | 45 |
| | Step 2: Reference Configuration | 47 |
| | Step 3: Clock Recovery | 47 |
| | Step 4: Thresholds | 48 |
| | Step 5: Acquisition | 50 |
| | Step 6: FFE/CTLE Setup | 52 |
| | Step 7: DFE Setup | 58 |
| | Step 8: Complete | 59 |

8 Manual Equalization Setup

| | |
|-------------------------------------|----|
| Introduction | 61 |
| Getting Your Signal Set up | 61 |
| The Equalizer Setup Dialog Box | 63 |
| The Available Equalization Setups | 64 |
| Equalizer Setup Dialog Box Controls | 65 |



1 License Information

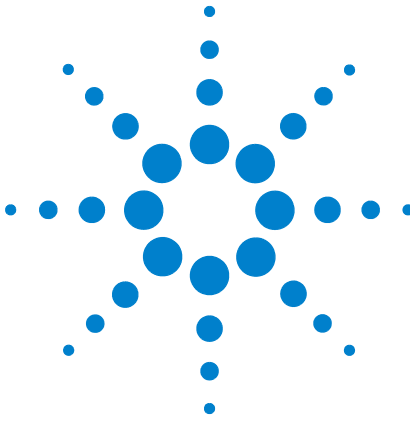
The license information for the Infiniium Serial Data Equalization application is:

- N5461A: Infiniium Serial Data Equalization
- DSO90000A-012: Infiniium Serial Data Equalization (installed on oscilloscope during purchase)
- N5435A-025: Infiniium Serial Data Equalization server license

Additionally, the Infiniium Serial Data Equalization application is dependent upon having the E2688A Serial Data Analysis option installed on your oscilloscope.



1 License Information



2 Theory / Background

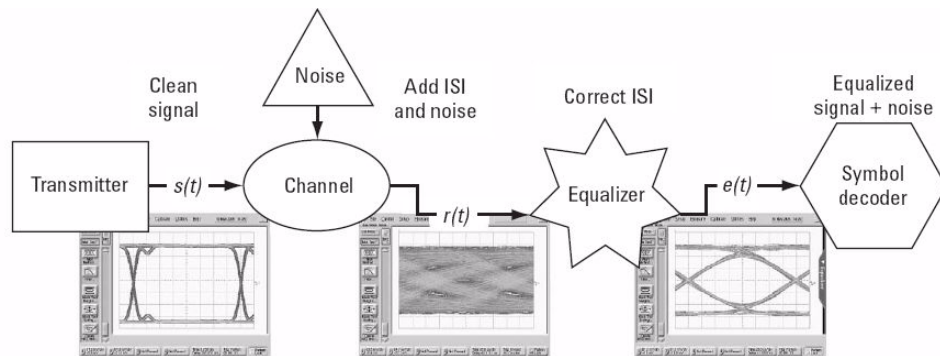
The Problem

In order to understand how equalization works, you first need to understand the problem it corrects.

A serial signal consists of a transmitter sending a signal over a transmission channel (examples: backplane, cable) to a receiver. As the signal rate increases, the channel the signal travels through distorts the signal at the receiver. This can result in a partially or completely closed eye diagram where the clock/data cannot be extracted by the receiver. In order to recover a clock or data from the eye diagram, it must be re-opened. This is where equalization can help.

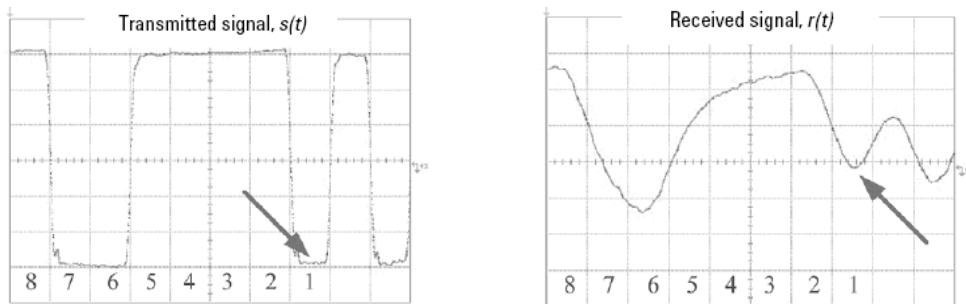
Look at the diagram below. You can see that an open, clean eye leaves the transmitter and is sent through the channel. As it passes through the channel, random noise, crosstalk, and inter-symbol interference (ISI, see [page 9](#)) distort the signal. All of these factors can cause the eye to close.

You can use equalization to re-open the eye by correcting for the ISI because ISI is deterministic (in other words, the ISI can be determined prior to transmission).



The random noise that was introduced cannot be corrected for by equalization because it is non-deterministic. Therefore, you can re-open the eye, but maybe not completely due to the inability to correct for this random noise.

To see why ISI (and the closing of the eye) causes problems when analyzing serial data, look at the following two screen shots showing a data signal before and after traversing a backplane.



Looking at the five bits labelled **5**, **4**, **3**, **2**, and **1**, you see the logical pattern is 11110 (decision threshold of $V = 0$). Also, notice even though the bit indicated by the arrow (bit **1**) is a logical zero when it is transmitted, the logic level of the bit as seen by the receiver is right around the decision threshold. It may, therefore, be inaccurately counted as a logical 1. This occurred because the frequency response of the backplane filtered out the high frequency content of the waveform.

It is vital that your receiver measures the same pattern the transmitter sends. Equalization is a technique used to ensure this occurs.

Inter-symbol Interference (ISI)

Inter-symbol interference (ISI) is caused by the non-flat frequency response of the channel the signal travels through (high frequencies exhibit more loss than low frequencies) and results in the distortion of pulse shapes in your signal.

Look at the following simplified frequency response plot.

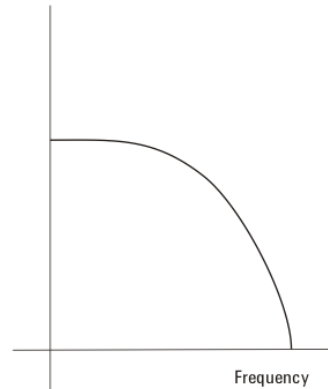


Figure 1 Simple example of a non-flat frequency response plot

You can see that at low frequencies, the frequency response is flat (like you want), but as the frequency increases, there is a drop in the amplitude. If your signal contains frequency components in this region, it will be distorted.

For example, imagine you are trying to send a serial data sequence (11010 . . .) from a transmitter, through a channel, and to a receiver.

Theoretically, you could send each one of these bits as square pulses from the transmitter. In real life, however, you are limited by bandwidth and, therefore, they are somewhat rounded.

As this data sequence is sent through the channel, the medium it is travelling through distorts the rounded pulses by smearing and elongating them (the degree of this distortion depends on the frequency components). This can cause each bit to have some of its energy overlap with neighboring bits, thus impacting the voltage level of those bits. When the signal exits the channel and enters the receiver, the receiver may measure different bit voltage levels as a result.

If you were to plot an eye of a serial signal with high enough frequency components that ISI is a significant problem, you would see that the low frequency trajectories of the waveform would still be accurate, but the high frequency components would be attenuated, resulting in either the complete or partial closing of the eye.

ISI is caused by a combination of:

- The geometry of the circuit (i.e. the design of the trace)
- The medium from which it is composed (i.e. the conductor or dielectric)

Each of these factors can be determined prior to signal transmission. Hence, ISI is deterministic and due to this, you can correct for it with equalization techniques.

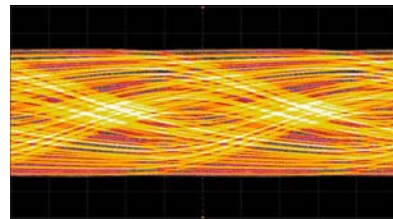
The Solution: Equalization

Now that the problem has been established, you can see how equalization is used as a solution.

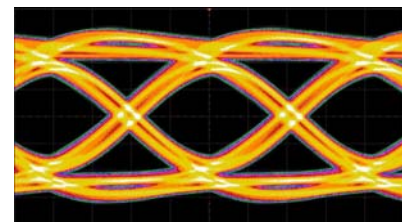
The main purpose of equalization is to correct for the problems caused by the transmission channel. Equalization techniques provide a way to discern the original signal (the signal coming out of the transmitter) given a distorted signal at the receiver.

In other words, equalization corrects for the high frequency component voltage levels and, in the process, corrects the trajectories of these components in the corresponding eye diagram (i.e., opens up the eye).

- **Unequalized 6 Gb/s Eye Diagram**
(notice the eye is completely closed)

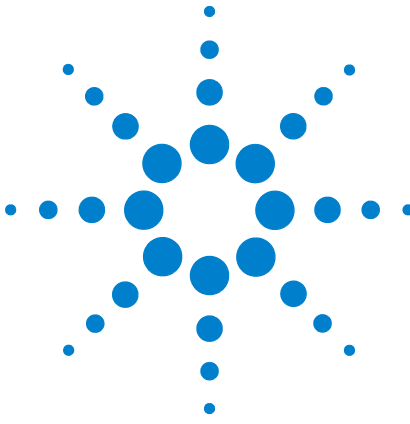


- **Equalized 6 Gb/s Eye Diagram**
(notice the eye is now open after equalization was applied)



There are three types of equalization methods used in the Infiniium 90000A and 9000A Series oscilloscopes: Feed-Forward Equalization (FFE), Continuous Time Linear Equalization (CTLE), and Decision Feedback Equalization (DFE). These three methods will be discussed in detail in [Chapter 3](#), [Chapter 4](#), and [Chapter 5](#).

2 Theory / Background



3 Feed-Forward Equalization Theory

Introduction

Feed-Forward Equalization (FFE) is an equalization technique so named because unlike Decision Feedback Equalization (discussed in [Chapter 5](#)), FFE corrects the received waveform with information about the waveform itself and not information about the logical decisions made on the waveform. FFE basically acts like a FIR (finite impulse response) filter and uses the voltage levels of the received waveform associated with previous and current bits to correct the voltage level of the current bit.

One key thing to remember when working with FFE is that the equalization is performed on the actual waveform. At no point in the FFE algorithm are logical decisions made (is this bit a 1 or a 0?). Instead, FFE is only concerned with correcting voltage levels in the waveform. You will see in [Chapter 5](#) that this is in direct contrast to Decision Feedback Equalization (DFE).

This chapter will describe FFE in four different ways. Each section will contain the same basic content, but will enable you to choose the viewpoint that best suits your understanding.



Mathematical Description of FFE

For the purpose of this discussion, assume the FFE algorithm we are using has three taps (refer to [Chapter 6](#) for more information on taps and how to determine the number to use). Taps are unitless correction factors applied to voltage levels in order to correct them. One way to think of these correction factors is to view them as the ratio of the voltage the receiver should have seen to the voltage the receiver did see.

The mathematical description of a three tap FFE is as follows:

$$e(t) = c_0 \cdot r(t - (0 \cdot T_D)) + c_1 \cdot r(t - (1 \cdot T_D)) + c_2 \cdot r(t - (2 \cdot T_D))$$

where:

- $e(t)$ is the corrected (or equalized) voltage waveform at time t .
- T_D is the tap delay.
- $r(t - nT_D)$ is the uncorrected input waveform n tap delays before the present time.
- c_n is the correction coefficient (tap) multiplied with the version of the uncorrected waveform that has been time-advanced by n tap delays.

So FFE obtains the corrected (or equalized) voltage level at the location of interest on the waveform (time t) by forming a sum of the taps and voltage levels of the previous two tap-delayed locations as well as the location of interest before being equalized.

Notice that if the FFE method above used a tap delay equal to one unit interval then it would use three bits to correct one bit (it uses the bit of interest and the previous two bits). This is because we used three taps in the algorithm. If we had used five taps then the equation above would have two more terms added to the sum [$c_3r(t-3T_D)$ and $c_4r(t-4T_D)$] which would mean the FFE algorithm would have used the bit of interest and the previous four bits to correct the bit of interest's voltage level.

Description of FFE Acting on a Signal

Another way to understand FFE is to see how it acts on a waveform. Assume the unequalized waveform measured at the receiver looks like the following:

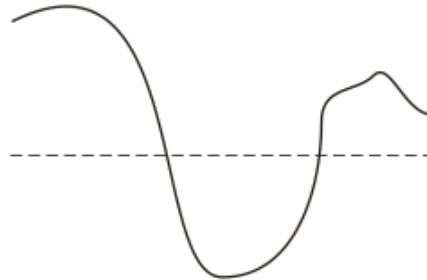


Figure 2 Unequalized waveform used for this example

Now, apply FFE and see how it progresses along the signal to correct it. Again, assume we are using three taps in our FFE algorithm for the purpose of this example.

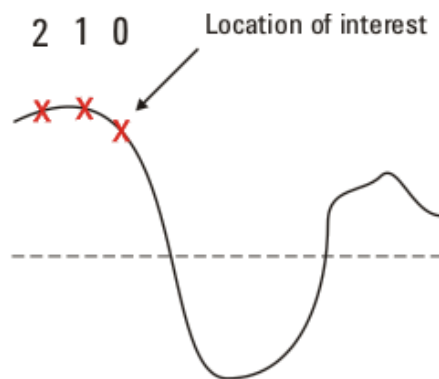


Figure 3 Location of interest and the two previous locations used in the equalization

The part of the waveform labeled as the “location of interest” is the voltage we are trying to correct (or equalize). The locations on the waveform marked **1** and **2** are determined by the tap delay you specify. Location **1** is one tap delay away from the location of interest and location **2** is two tap delays away from the location of interest.

It is important to note that we are just working with voltage levels in the waveform. At no point in the process are we going to use logical decisions (1's or 0's) in the FFE algorithm.

First we need to know the tap values for the distortion present in our transmission channel. Determining these values is discussed in [Chapter 6](#). For now, assume these three tap values have been determined (call them Tap0, Tap1, and Tap2 corresponding to c_0 , c_1 , and c_2 in the equation above).

The oscilloscope measures the voltage of the location of interest (location 0) and then multiplies it by the Tap0 value. Then the oscilloscope measure the voltage of location 1 and multiplies it by the Tap1 value. Next, the oscilloscope measures the voltage of location 2 and multiplies it by the Tap2 value. Finally, the oscilloscope adds each of these voltage/tap products together to obtain the corrected (or equalized) voltage for location 0. The voltage at the location 0 will then be adjusted to this corrected value.

Now the three Xs move along with the sample rate to their next locations on the waveform.

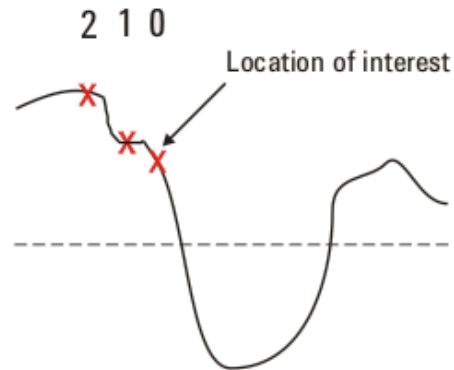


Figure 4 The new locations used for FFE

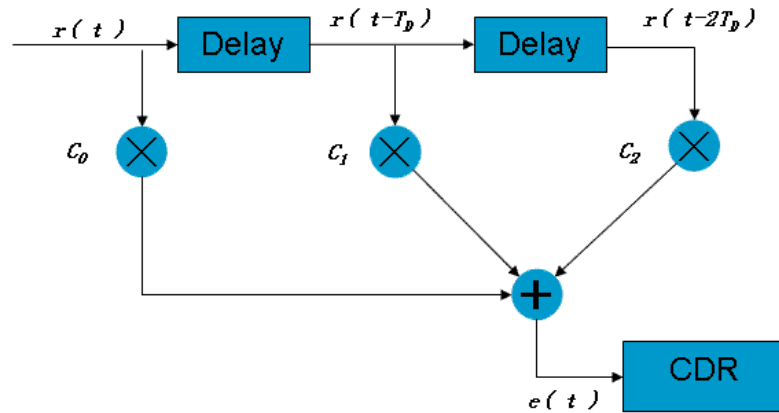
Notice the old “location of interest” (now labeled 1) has its new equalized voltage level. You can see the tap values caused the voltage level to drop for that location.

The same process is repeated for the new location of interest. Location 0’s voltage level is multiplied by Tap0 and then added to the product of location 1’s voltage and Tap1. Finally, this is added to the product of location 2’s voltage and Tap2 to give bit 0’s new corrected voltage.

The three Xs will then move along with the sample rate again to their new locations and repeat the entire process. This continues until the entire waveform has been traversed.

FFE Decision Diagram

It may also be useful to see the decision diagram for a three tap FFE.



where:

- $r(t - nT_D)$ is the input waveform n tap delays before the present time
- C_n is the n^{th} coefficient (tap)
- T_D is the tap delay
- $e(t)$ is the equalized waveform at time t
- CDR is Clock/Data Recovery

This is just a different way of looking at the process described in the previous two sections. Again you see that each voltage level is multiplied by its corresponding tap value and then all of these products are summed together to give the new equalized voltage for the location of interest.

Understanding FFE Through Frequency Response Plots

One final way to understand FFE is in terms of the frequency response plot of your signal.

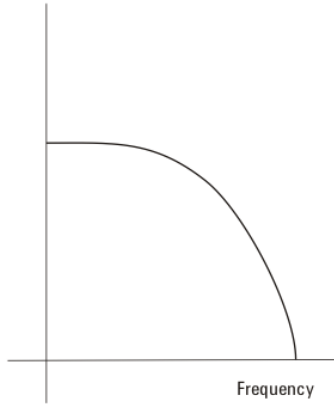


Figure 5 Simple example of a non-flat frequency response plot

Ideally, you would like to transmit a signal whose frequency response was flat over the full bandwidth of the signal so every frequency component of the signal was equally represented. However, as you saw in [Chapter 2](#), the response sometimes attenuates the signal at higher frequencies (as shown in the plot above).

To correct for this attenuation, we can apply a peaking filter via equalization. With FFE, we apply a filter response as shown in the plot below (dashed line) such that the sum of the peaking filter response and the channel response yields a flat frequency spectrum (red line in plot below).

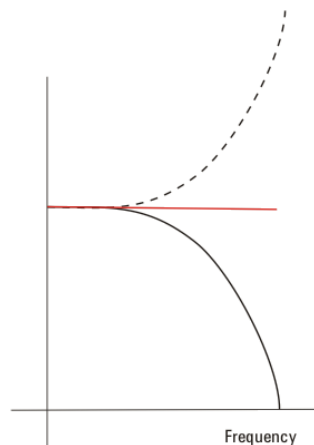


Figure 6 Plot showing the response generated by FFE (dashed) to correct for the non-flat frequency response of the channel (black)

Remember ISI is a result of the non-flat response of the channel. When we flatten the response, we are eliminating the ISI.

3 Feed-Forward Equalization Theory



4 Continuous Time Linear Equalization

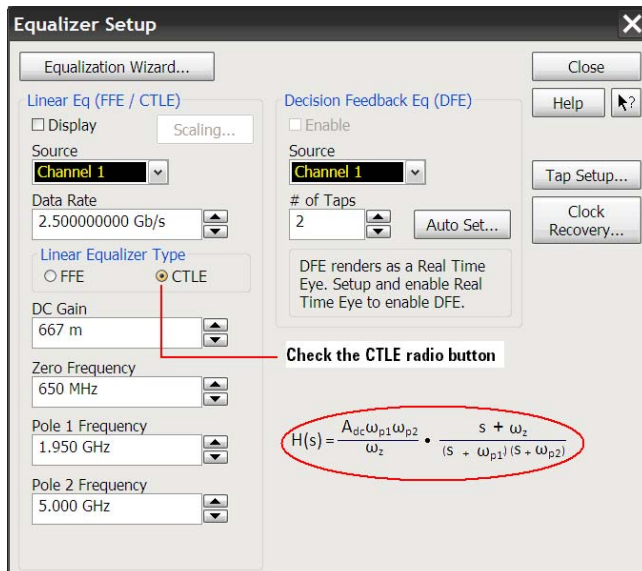
Introduction

Continuous Time Linear Equalization (CTLE) is another linear equalization method available via the Infiniium 9000A and 90000A Series oscilloscopes. When performing equalization on your Infiniium oscilloscope, you choose whether you want to use FFE or CTLE (or neither) for your linear equalization method.

The filter applied to your signal via CTLE is described by the equation shown below:

$$H(s) = \frac{A_{dc} \omega_{p1} \omega_{p2}}{\omega_z} \cdot \frac{s + \omega_z}{(s + \omega_{p1})(s + \omega_{p2})}$$

This equation is also shown in the dialog box when you check the **CTLE** radio button in the **Equalization Setup** dialog box.



You must specify four variables in this equation in order to generate the filter. These variables are:

- DC Gain (A_{DC})
- Zero Frequency (ω_z)
- Pole 1 (ω_{p1})
- Pole 2 (ω_{p2})

Set these variables using the four fields to the right of the CTLE filter equation in the Equalizer Setup dialog box (again, the CTLE radio box must be checked to see the CTLE settings).

For more information about using the other Serial Data Equalization generic controls or about using the Equalization Wizard, please refer to subsequent chapters.



5 Decision Feedback Equalization Theory

Introduction

Decision Feedback Equalization (DFE) is an equalization technique so named because it makes logical decisions (whether a bit is a 0 or a 1) and then feeds that information back to help determine whether the current bit is a 1 or a 0.

There are multiple ways to implement DFE. This chapter will discuss the DFE algorithm used in the Infiniium 90000A Series oscilloscope.

DFE will be described in four different ways. Each section will contain the same basic content, but will enable you to choose the viewpoint that best suits your understanding.



Mathematical Description of DFE

For the purpose of this discussion, assume the DFE algorithm we are using has two taps (refer to [Chapter 6](#) for more information on taps and how to determine the number to use). Taps are unitless correction factors applied to voltage levels in order to correct them. One way to think of these correction factors is to view them as the ratio of the voltage the receiver should have seen to the voltage the receiver did see.

Before looking at the mathematical description of DFE, it is important to understand the results of the algorithm. Generally, DFE calculates a correction value that is added to the logical decision threshold (the threshold above which the waveform is considered a logical high and below which the waveform is considered a logical low). Therefore, DFE results in the threshold shifting up or down so new logical decisions can be made on the waveform based upon this new equalized threshold level.

The mathematical description of the correction value added to the decision threshold for a two tap DFE is as follows:

$$V(k) = c_1 s(k-1) + c_2 s(k-2)$$

where:

- $V(k)$ is the correction voltage added to the decision threshold used when determining the logic value of bit k .
- $s(k-n)$ is the logic value of the data bit located n bits prior to bit k . The logic high value of $s(k-n)$ is equal to the Upper Target field in the GUI while the logic low value of $s(k-n)$ is equal to Lower Target (see [page 37](#) for more information on the Upper Target and Lower Target fields).
- c_n is the correction coefficient (tap) for the bit n bits prior to the bit of interest.

So, in order for DFE to obtain the corrected voltage offset for the threshold level at the bit of interest, it first needs to be seeded with the correct bit values for the first several bits to get started. Assuming the logical decisions for the first several bits are correct, the algorithm can feed them forward to determine the logical value of the current bit.

For a two tap DFE, the two bits previous to the current bit need to have their bit levels already determined. Then the algorithm multiplies their bit levels by their corresponding

tap values. The sum of these two tap/bit level products gives the amount the decision threshold should be shifted. However, instead of shifting the threshold, the Infiniium oscilloscope leaves this constant and shifts the waveform relative to it.

Notice the DFE method above used two previous bits to correct the current bit. This is because we used two taps in the algorithm. If we had used four taps then the equation above would have two more terms added to the sum ($c_3s(k-3)$ and $c_4s(k-4)$) which would mean the DFE algorithm would have used the previous four bits to correct the bit of interest's voltage level.

Also notice the bit of interest is not used in the summation. FFE uses previous bits and the current bit in its sum, but DFE only uses previous bits.

Description of DFE Acting on an Eye Diagram

Another way to understand DFE is to see how it acts on an eye diagram.

First, look at the following segment of a serial data signal.

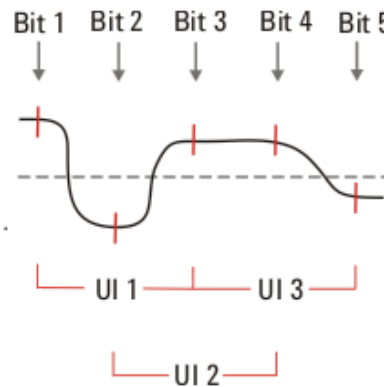


Figure 7 Example of a serial data stream

The vertical red bars on the waveform correspond to the clock edges for each bit and the red segments at the bottom of the figure delineate the three unit intervals (UI) that will be overlapped on the display to form an eye diagram. The gray line is the logical threshold level (bits with voltages above this are considered logical 1s and those with voltages below this are considered logical 0s).

Now if you overlap these three unit intervals to form an eye diagram, the logical high voltage levels for unit intervals 2 and 3 will not be as high as unit interval 1. In other words, the trajectories from unit intervals 2 and 3 will result in the partial closing of the eye.

Let's equalize the voltage for bit 4 using two taps to correct unit interval 3.

First, we need to know the tap values for the distortion present in our transmission channel. Determining these values is discussed in [Chapter 6](#). For now, assume these two tap values have been determined (call them Tap1 and Tap2 corresponding to c_1 and c_2 in the equation on [page 24](#)).

The DFE algorithm is first seeded with the correct bit values (high or low) for the first several bits in order to get it started. Assuming the algorithm has correctly determined the initial bit values, it can feed forward previous bit values to determine the current bit. So, for our example, we assume the oscilloscope has already determined that bit 2 is low and bit 3 is high.

The DFE algorithm would then take the ideal high value for the third bit (+1, for example) and multiply it by Tap1. Then it would add this product to the product of the ideal low value for the second bit (-1, for example) and Tap2.

This sum would then give the amount the decision threshold should be shifted. The resulting threshold shift might be the shift marked by the blue line in the plot below.

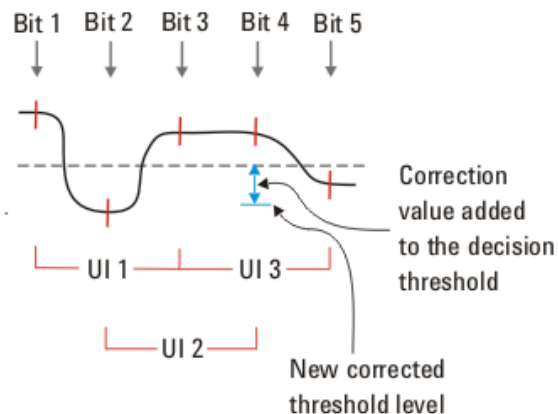


Figure 8 Blue arrow shows the new threshold level after DFE was applied to the fourth bit

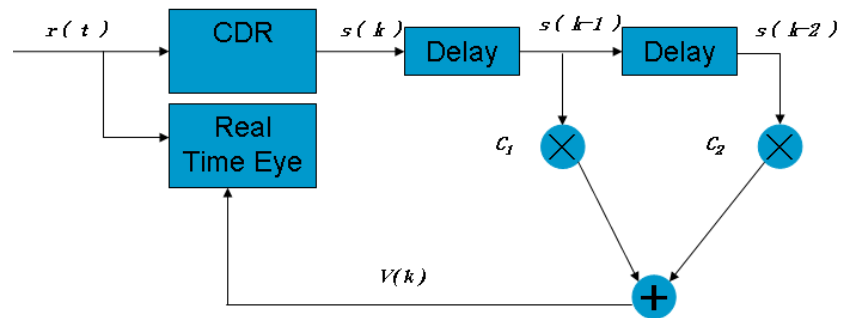
Many DFE algorithms would then shift the voltage threshold (the dashed gray line) to this new voltage level at the location of the fourth bit, but the Infiniium 90000A Series oscilloscope does the opposite. Instead of shifting the voltage threshold down in this example, it keeps the threshold constant and shifts the bit four voltage level up by the same amount (the amount marked by the blue arrow).

This would have the effect on the eye of raising the corresponding trajectory, opening up the eye in the process.

The algorithm would then shift forward one index so the new bit of interest would be bit five and the two bits used to equalize it would be bits three and four. This process repeats itself until the entire signal has been traversed.

DFE Decision Diagram

It may also be useful to see the decision diagram for DFE.



where:

- $r(t)$ is the unequalized analog waveform voltage at time t
- $s(k-n)$ is the logic value (either Upper Target or Lower Target) for the bit n tap delays prior to the current bit.
- C_n is the n^{th} coefficient (tap).
- CDR is Clock/Data Recovery.
- $V(k)$ is the correction voltage added to the decision threshold used when determining the logic value of bit k . (Remember that Infiniium actually leaves the threshold constant and shifts the waveform by this amount in the opposite direction).

This decision diagram is just another way of looking at the process described above.

The DFE algorithm is seeded with several correct bit values to get started. Then there is a tap delay (corresponding to each clock edge) between each bit. Each previous bit used in the algorithm is determined to be either high or low and is then multiplied by its corresponding tap value. These voltage/tap products are summed to determine how much to shift the waveform relative to the logical decision threshold.

Understanding DFE Through Frequency Response Plots

One final way to understand DFE is in terms of the frequency response plot of your signal.

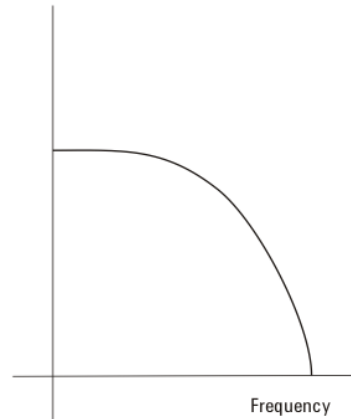


Figure 9 Simple example of a non-flat frequency response plot

Ideally, you would like to transmit a signal through a channel whose frequency response was flat over the full bandwidth of the signal so every frequency component of the signal was equally represented. However, as you saw in [Chapter 2](#), the response sometimes attenuates the signal at higher frequencies (as shown in the plot above).

To correct for this attenuation, we can apply a peaking filter via equalization. With DFE, we apply a peaking filter response as shown in the plot below (dashed line) such that the sum of the peaking filter response and the channel response yields a flat frequency spectrum (red line in plot below).

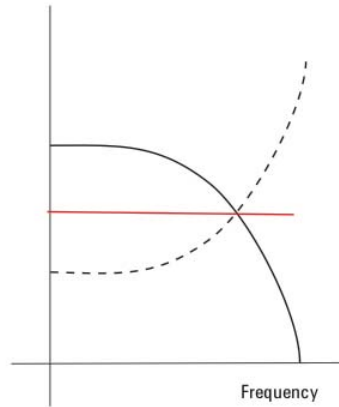


Figure 10 Plot showing the response generated by DFE (dashed) to correct for the non-flat frequency response of the channel (black)

Remember ISI is a result of the non-flat response of the channel. When we flatten the response, we are eliminating the ISI.

In order to get the eye at the receiver to be the same size as the eye at the transmitter, we need to apply a gain term. To understand why the gain term (A) is needed, remember what the frequency response looks like for FFE.

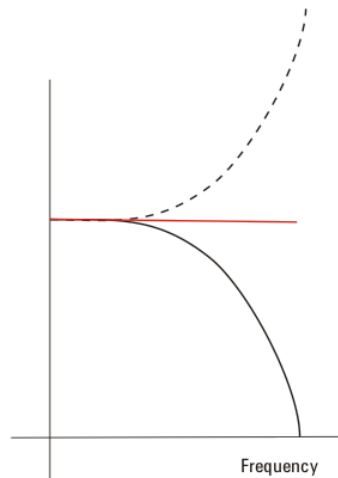


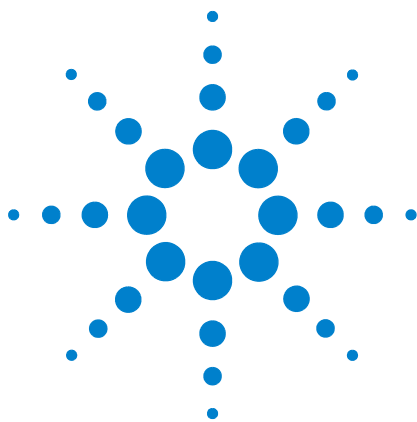
Figure 11 Plot showing the response generated by FFE (dashed) to correct for the non-flat frequency response of the channel (black)

You can see we did not attenuate the low frequencies, but instead left them alone.

However, with the DFE response, we are attenuating the low frequencies because we have shifted the entire FFE response down (Figure 10). This means the flat response we obtain in the end will be shifted down as well. You can use the gain factor (A) to shift the flat response back up to its correct value.

This makes the eye at the receiver the same size as the eye leaving the transmitter.

5 Decision Feedback Equalization Theory



6 Taps

Introduction

Taps are dimensionless correction factors applied to the bit voltage levels in serial data patterns during equalization. One way to think of these correction factors is to view them as the ratio of the voltage the receiver should have seen to the voltage the receiver did see.

To see how taps are actually implemented in equalization, refer to the *Mathematical Description* sections in [Chapter 3](#) and [Chapter 5](#).

How Many Taps Should You Use?

Generally, two taps will open your eye diagram significantly. However, there may be times when the signal error may span over more than two bits and in these cases, more taps will be required.

A simple way to see how many taps you need is a trial-and-error approach. Simply set the number of taps (perhaps start with 3-4) and then have the oscilloscope optimize the tap values for your signal (optimization is discussed later in this chapter). Look at the tap values after they have been optimized and see if the higher tap values (for example, start with tap 4, then 3, ...) are contributing much or if they are near zero. If there is a tap value near zero, this tap is unnecessary. If all the tap values are significant then add more taps, re-optimize, and investigate again. This is a good approach if you do not know how many tap values to use for your specific setup.

Notice the GUI allows you to enter up to 16 taps. However, you can actually use up to 40 taps (see [page 34](#)). More than likely, you are not going to have an error that spans 40 bits, but the oscilloscope allows you to use this many taps in case you want to use more than one tap per bit. For instance, if



you found you needed four taps when you used 1 tap/bit, then you would probably need eight taps if you used 2 taps/bit.

How to Enter More Than Sixteen Taps

This section only applies if you are manually entering tap values. If you are going to have the oscilloscope determine the tap values for you through optimization then all you need to do is follow the instructions in the *Tap Optimization* section on [page 38](#).

If you are manually entering your tap values, the GUI only allows you to enter up to 16 taps. However, you can actually use up to 40 taps. In order to use more than 16 taps, follow these instructions:

- 1 Open the **Equalizer Tap Setup** dialog box (**Analyze -> Equalization ...** and then click on the **Tap Setup** button).
- 2 Choose the number of taps you want to use by entering a number (up to 40) in the **# of Taps** field. If you are only going to use FFE then you only need to use the **# of Taps** field under the FFE section and likewise, you need to use the **# of Taps** field under the DFE section if you are using DFE. If you are using both then you will need to fill in the fields under both sections.
- 3 Even though you may enter a value higher than 16, you will see that only 16 tap value fields are shown in the GUI. To access the rest, you need to first press the **Save Taps ...** button. This will save your current taps setup to an .xml file (choose a save location on the scope where you can easily find this .xml file).
- 4 Minimize the oscilloscope application and navigate to the location of the .xml file. Open it. The file will look similar to the one seen below.

```
<?xml version="1.0" encoding="utf-8" ?>
- <Equalization
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
Version="1" SwVersion="01.39.9057">
- <FFE>
<Delay>0.0E+00</Delay>
<NumTaps>40</NumTaps>
<Coefficients>1.000 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0</Coefficients>
</FFE>
- <DFE>
<UpperTarget>1.000</UpperTarget>
```

```

<LowerTarget>-1.000</LowerTarget>
<Gain>1.000</Gain>
<Delay>0.0E+00</Delay>
<NumTaps>40</NumTaps>
<Coefficients>0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0</Coefficients>
</DFE>
</Equalization>

```

- 5 Notice that there are two sections in the .xml file, one for FFE (enclosed between the starting tag <FFE> and ending tag </FFE>) and one for DFE (enclosed between the starting tag <DFE> and ending tag </DFE>). Also notice that the other parameters in the dialog box are also saved to this file (examples: gain, number of taps, upper target, etc.). To edit FFE tap values, simply change the numbers in the <Coefficients> section within the FFE section.

For example, if you wanted to change the seventeenth tap from 0.0 to -0.230, you would make the following change to the FFE section of the .xml file:

```

- <FFE>
<Delay>0.0E+00</Delay>
<NumTaps>40</NumTaps>
<Coefficients>1.000 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 -0.230 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0</Coefficients>
</FFE>

```

- 6 So, if you are using seventeen taps, the FFE section might look something like this when you are done entering all your values:

```

- <FFE>
<Delay>0.0E+00</Delay>
<NumTaps>17</NumTaps>
<Coefficients>1.000 -0.786 0.684 0.559 -0.491 0.473 0.430
-0.402 -0.391 0.355 -0.330 -0.319 0.297 -0.276 0.272 0.254
-0.230
</Coefficients>
</FFE>

```

- 7 If you are using DFE as well then you will also need to adjust the tap values in the DFE section. If you are only using DFE or FFE then you only need to adjust the tap values in the section you are using.
- 8 Save the .xml file and then bring the oscilloscope application up. Go back to the **Equalizer Tap Setup** dialog box and press the **Load Taps ...** button. Then

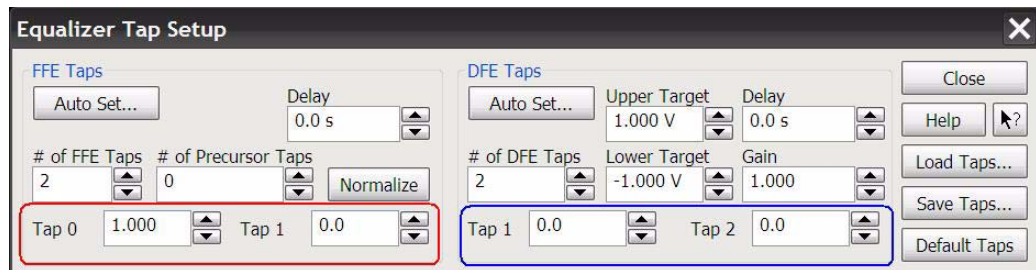
choose the .xml file you just changed. Press **OK** and the tap values you entered will be used in the equalization.

How to Determine Which Values to Use For Your Taps

There are two ways to determine the tap values for your specific application: manually or by optimizing them with your oscilloscope.

Manual Method

This method assumes you know what tap values to use for your system. To enter these tap values, enter the **Equalizer Setup** dialog box (**Analyze -> Equalization...**) and then click on the **Tap Setup** button. The following **Equalizer Tap Setup** dialog box will be displayed.



The area circled in red is where you can manually enter FFE taps and the area circled in blue is where you can manually enter DFE taps. The number of taps can be adjusted with the **# of FFE Taps** or **# of DFE Taps** fields.

Notice the GUI only allows you to manually adjust up to 16 tap values when in fact you can use up to 40 taps. To see how to access the rest of these taps, refer to page [page 34](#).

FFE Taps Section

Equalization may cause your eye diagram to drift so the eye is no longer centered on the clock. The **Delay** field allows you to specify this shift in order for the eye to be brought back to its original location relative to the unequalized signal.

The **Normalize** button makes the tap values sum to one in order for the output to have unity gain at dc. This allows you to see the same size eye diagram at the receiver as you

do at the transmitter. If you want to use this feature, first enter your tap values and then press the **Normalize** button. Your tap values will be adjusted so they sum to one.

You can also specify the number of precursor taps you want to use by using the **# of Precursor Taps** control. Remember that with standard FFE, the current bit is equalized using its tap value along with tap values associated with bits prior to it. Precursor taps are tap values associated with bits in front of the current bit. For example, if you set **# of Precursor Taps** to zero and you were using three taps then the equalization would use the current bit's tap value, the previous bit's tap value, and the bit's tap value that was two behind the current bit. Now, if you instead set the **# of Precursor Taps** control to 1 then the current bit's tap value, the previous bit's tap value, and the bit in front of the current bit's tap value would be used. The **# of FFE Taps** is the total number of taps you want to use and the **# of Precursor Taps** is the number of taps from this total that will be associated with bits in front of the current bit.

DFE Taps Section

The **Upper Target** and **Lower Target** fields dictate the high and low logical values used in the DFE algorithm. For example, in DFE, when a bit is determined to be a logical high, its value will be equal to the Upper Target. Similarly, if a bit is determined to be a logical low, its value will be equal to the Lower Target.

Equalization may cause your eye diagram to drift so the clock is no longer centered on the eye. The **Delay** field allows you to specify this shift in order for the eye to be brought back to its original location relative to the unequalized signal.

Remember in the discussion on DFE that the eye drawn after DFE is applied is attenuated. To amplify the eye back to its original size (so you can directly compare the eye at the receiver to the eye at the transmitter) a gain factor needs to be applied. The **Gain** field allows you to enter a constant that is applied to both the unequalized input signal and the corrected decision threshold value. Setting this field to 1 means there is no gain applied to the DFE eye diagram (like multiplying by 1).

Tap values can also be saved and loaded by using the **Save Taps ...** or **Load Taps ...** buttons. These buttons save and/or read your taps from a .xml file that can then be shared with colleagues or used by you at a later date so you do not have to enter the taps again manually.

Optimizing the Tap Values

If you do not know which tap values to use, you can have the oscilloscope figure them out for you through optimization. The optimization process works differently for FFE and DFE.

NOTE: The menu paths included in the following discussion will assume you are not using the Equalization Wizard. If you are using the Wizard, you will automatically have a chance to optimize the tap values during one of the steps. You will still need to read the following discussion in order to understand the optimization, but ignore the menu paths.

Optimizing FFE Taps

If you are not using the Equalization Wizard, go to the **Equalizer Setup** dialog box (**Analyze -> Equalization...**), select the number of taps to use, and then click on **Auto Set ...** under the *FFE* section. This will bring up the **Equalizer Auto Tap Setup** dialog box.

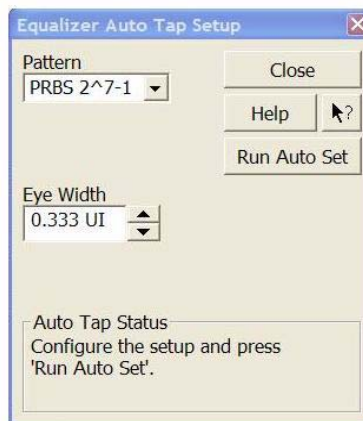
The optimization algorithm needs to know the pattern being input to the oscilloscope in order to work. You can train the oscilloscope to a specific pattern signal and then use those tap values to equalize live traffic.

It is possible to train the tap values from a reference signal because the tap values are only dependent on the type of distortion, not the specific pattern. Therefore, you need to input a specific pattern signal with similar ISI characteristics to your live traffic signal. For example, a PRBS 2^6-1 signal has five successive 1's and 0's similar to 8b/10b traffic.

It is best to use a short PRBS (pseudorandom binary sequence) as the reference signal for training. This is because:

- PRBS signals are rich and diverse in that they contain many different combinations of ones and zeros. The more combinations the optimizer "sees", the better it is able to optimize the tap values. This is why CJT patterns do not work well as reference patterns.
- Using a short pattern is key because it does not require you to know the exact data rate. For instance, if you use a long pattern length and are off on the data rate, by the time the optimizer reaches the end of the pattern, the bits may not be in their correct locations.

The **Equalizer Auto Tap Setup** dialog box gives you several different PRBS patterns to choose from under the **Pattern** field. You can also set this field to **Custom** and then import a .ptrn file that describes the pattern you are using.



The **Eye Width** field lets you choose how much of the eye to use in the optimization. This field ranges from zero to one. Entering zero means the eye is only optimized at the location of the clock. Entering one means the entire acquisition is used. Typically with FFE, the receiver samples data not only at the location of the clock, but also in its vicinity. The default is set to 0.333 UI (unit interval) in order to account for this variance in the sample location. Typically values between 0.0 UI - 0.5 UI are the most useful.

If you are unsure about what value to enter here, but do not want to use the default, think about whether you have more margin vertically or horizontally (where is your

uncertainty?). If you use a narrow portion of the eye to optimize the tap values, your eye will have a greater peak, but only near its center.

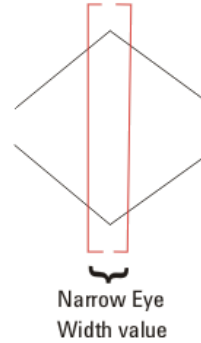


Figure 12 When a small Eye Width value is used for optimization, the eye diagram is greatly peaked only near its center.

If you use more of the eye to optimize the tap values, your eye will not be peaked as high, but will be more spread out horizontally.

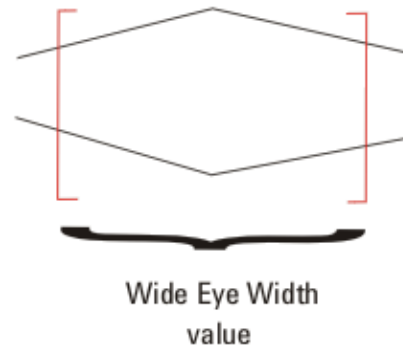
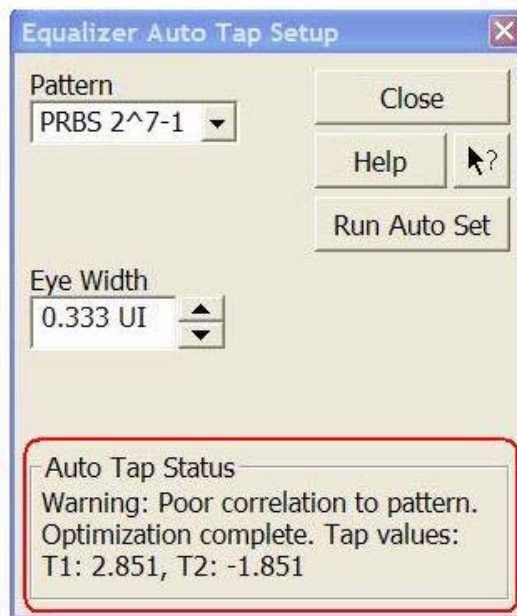


Figure 13 When a larger Eye Width value is used for optimization, the peak is not as great, but the eye diagram is more spread out horizontally.

Once you have these two fields set, press the **Run Auto Set** button. The optimizer will run and report the tap values to you in the **Auto Tap Status** box. These taps will also be automatically inserted into the **Equalizer Tap Setup** dialog box (**Analyze -> Equalization -> Setup Taps** button).

The optimizer also looks for correlations between the input signal and the pattern you specify in the **Pattern** field. If it does not find a strong correlation, it will report a warning like the one circled in red in the screen shot on the next page.



Notice the optimizer still generated tap values in case there was a weak correlation but everything was entered correctly. However, if you receive this warning, it is advised you double check the following two things:

- Make sure your input pattern to the oscilloscope is the same pattern you specified in the **Pattern** field
- Make sure the data rate is set accurately

Optimizing DFE Taps

If you are not using the Equalization Wizard, to access the tap optimization for DFE, go to the **Equalizer Setup** dialog box (**Analyze** -> **Equalization...**), select the number of taps to use, and then click on **Auto Set ...** under the *DFE* section (circled in screen shot on the next page). This will bring up the **Equalizer Auto Tap Setup** dialog box.

Some serial data standards have limits on the tap values that can be used for DFE (you can look up these limits in the standards for your technology). Therefore, this dialog box allows you to enter maximum/minimum values for the generated taps.

There is an **Eye Width** field which dictates how much of the eye to use in the optimization (see previous discussion in the *Optimizing FFE Taps* section). For DFE, true receivers only sample data at the location of the clock so a value of 0.0 UI (unit interval) is used as the default. If you want to include more of the eye, simply set this number to a higher value (0.0 UI - 0.5 UI would make sense for DFE).

If you are unsure about what value to enter here, but do not want to use the default, think about whether you have more margin vertically or horizontally (where is your uncertainty?). If you use a narrow portion of the eye to optimize the tap values, your eye will have a greater peak, but only near its center.

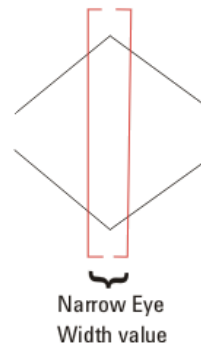


Figure 14 When a small Eye Width value is used for optimization, the eye diagram is greatly peaked only near its center.

If you use more of the eye to optimize the tap values, your eye will not be peaked as high, but will be more spread out horizontally.

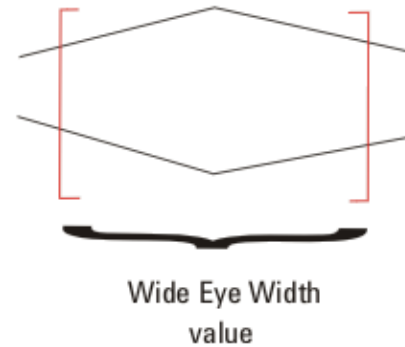
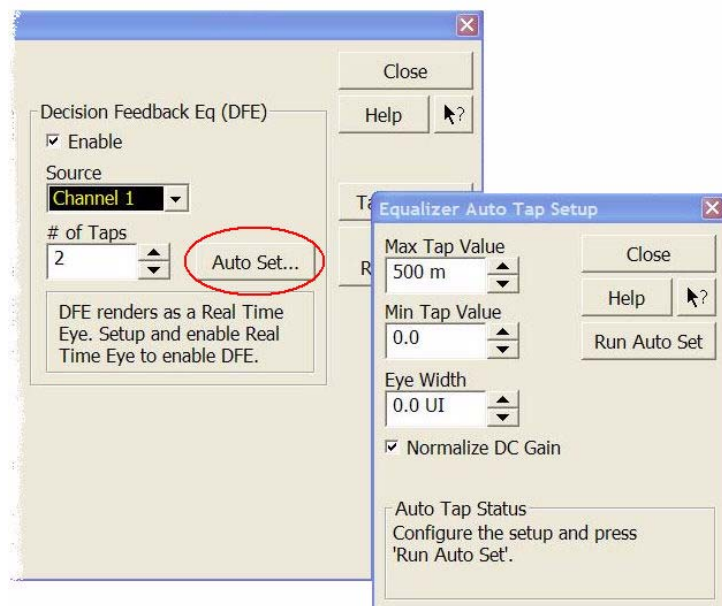
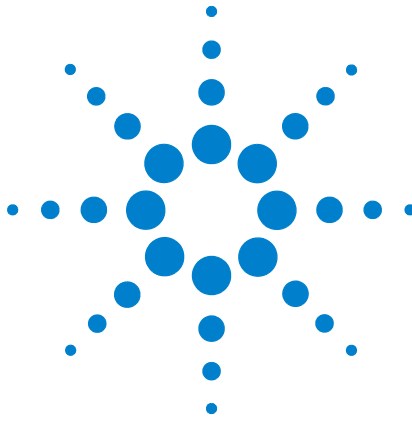


Figure 15 When a larger Eye Width value is used for optimization, the peak is not as great, but the eye diagram is more spread out horizontally.

Remember in the discussion on DFE that the eye drawn after DFE is applied is attenuated. To amplify the eye back to its original size (so you can directly compare the eye at the receiver to the eye at the transmitter) a gain factor needs to be applied. Checking the **Normalize DC Gain** box automatically scales the eye diagram so it is the same size as the transmitted eye.



Once you have these two fields set, press the **Run Auto Set** button. The optimizer will run and report the tap values to you in the **Auto Tap Status** box. These taps will also be automatically inserted into the **Equalizer Tap Setup** dialog box (**Analyze -> Equalization -> Setup Taps** button).



7 Equalization Wizard

Introduction

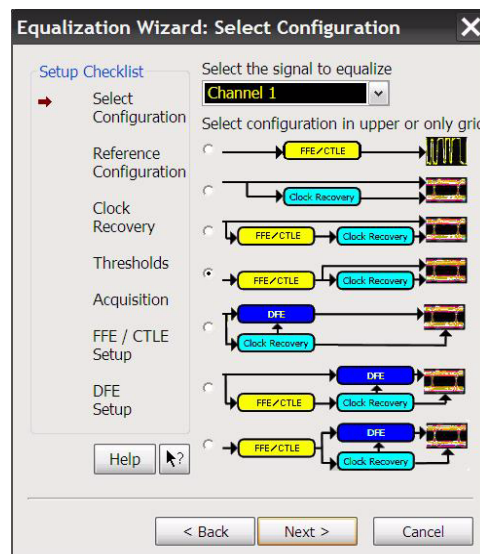
The Equalization Wizard is the recommended setup method as it walks you through the process in a step-by-step fashion. The other setup method, Manual ([Chapter 8](#)), should only be used if you are comfortable with the manual controls, are familiar with equalization, and know what all your parameter values should be.

To access the Equalization Wizard open the **Equalizer Setup** dialog box (**Analyze -> Equalization**) and then click on the **Equalization Wizard . . .** button.




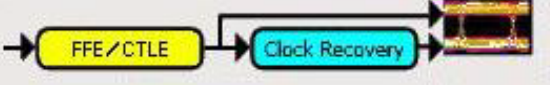
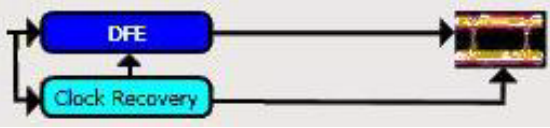
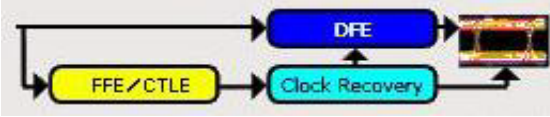
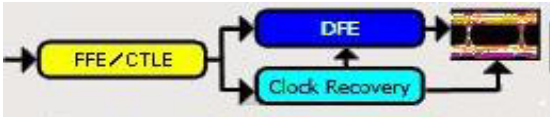
Step 1: Select Configuration

The first step after the **Welcome** page, is the **Select Configuration** page. This page offers you a wide selection of how you want to set up the equalization.

The Equalization Wizard does not set up your signal so you need to have whatever channel, function, or waveform memory you want to equalize displayed on the oscilloscope. **The Wizard assumes the signal is already on the display.**



Simply click on the radio button next to the desired selection. The equalization setup selected will be used for the eye drawn in the top grid of the oscilloscope display whereas the next step will have you select the equalization method to use and display in the bottom grid (in case you want to compare or have a reference eye). The following is a table with details for each setup.

| Image from Dialog Box | Description |
|---|---|
|  | <ul style="list-style-type: none"> • FFE/CTLE is applied, but the real-time eye is not shown. Instead, the waveform resulting from applying FFE/CTLE is displayed. |
|  | <ul style="list-style-type: none"> • No equalization is implemented and the unequalized real-time eye is displayed. This is useful if you want to have a reference to compare to. For instance, you can put the unequalized eye in the top grid and the equalized eye in the bottom grid so you can compare them on the same screen. |
|  | <ul style="list-style-type: none"> • FFE/CTLE is only used to recover the clock, but the actual displayed real-time eye is unequalized. This is useful if you want to have a reference to compare to, but your unequalized eye is completely closed. You cannot recover a clock on a closed eye, so FFE/CTLE is applied to open the eye so the clock can be recovered. In other words, this options serves the same purpose as the selection immediately above it, but is used with closed eyes. |
|  | <ul style="list-style-type: none"> • A FFE/CTLE eye is displayed on the oscilloscope. This is a standard FFE/CTLE implementation and can be used on partially or completely closed eyes. |
|  | <ul style="list-style-type: none"> • A Decision Feedback Equalized eye is displayed on the oscilloscope. This structure cannot be used on completely closed eyes as the clock cannot be recovered with this method. If the eye is closed and you want to implement DFE, use the option immediately below this. |
|  | <ul style="list-style-type: none"> • FFE/CTLE is used to recover the clock (but is not implemented in the real-time eye display) and a Decision Feedback Equalized eye is displayed on the oscilloscope. This allows you to perform DFE on completely closed eyes. |
|  | <ul style="list-style-type: none"> • FFE/CTLE is applied and then DFE is applied to the real-time eye. Additionally, since FFE/CTLE is also used to recover the clock, this structure can be used on completely or partially closed eyes. |

This step also requires you to select the channel input, function, or waveform memory you want to equalize. **Be sure you choose the correct signal.** It is recommended that only one signal be displayed prior to running the Wizard. It is difficult to discern one signal from another with real-time eye diagrams.

Step 2: Reference Configuration

This step involves selecting which equalization setup to use on the eye drawn in the lower grid of your oscilloscope's display. This is useful if you want to have a reference eye to compare to.

For instance, you could draw an unequalized eye in the lower grid and an equalized one in the upper grid for comparison. Or you could draw a DFE eye in the lower grid and a FFE/CTLE eye in the upper grid.

You can also choose to display nothing in the bottom grid (by clicking on the radio button next to **None**) and instead display the setup from Step 1 using the entire display.

The choices you have available in this step depend on your selection in Step 1 and will be a subset of the ones listed on the previous page.

Step 3: Clock Recovery

The clock recovery algorithms available in this step will depend on your settings in Steps 1 and 2. It will be a subset of the following list:

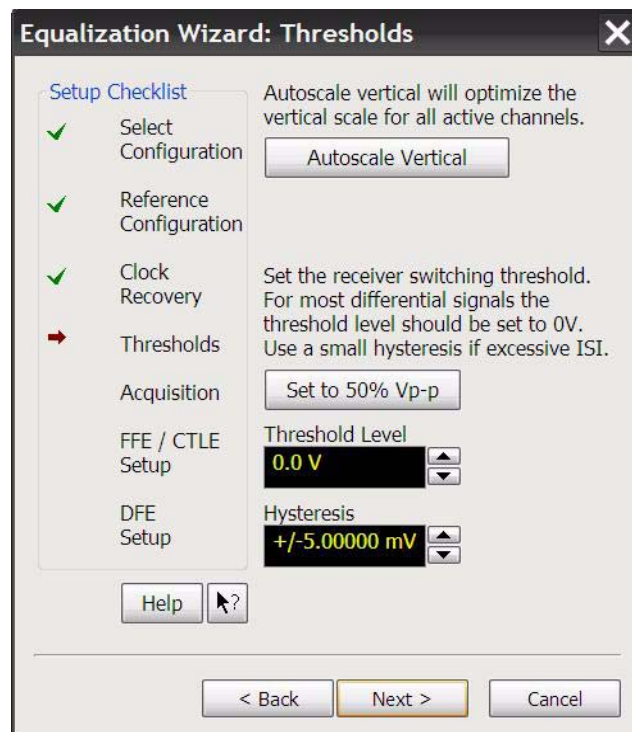
- First Order PLL (Phase Lock Loop)
- Second Order PLL
- Equalized First Order PLL
- Equalized Second Order PLL
- Constant Frequency
- Explicit
- Explicit First Order PLL
- Explicit Second Order PLL
- Fibre Channel

Generally, you want to choose the method that approximates the method your receiver uses. After you choose a clock recovery method, there will be a second window where you fill in the needed parameters for that method.

Equalized First and Second Order PLL means first FFE/CTLE is applied to the closed eye in order to open it (because the oscilloscope cannot extract a clock from a closed eye) and then the corresponding PLL algorithm is used to extract the clock.

Step 4: Thresholds

This step sets up various parameters associated with the logical threshold applied to the serial data pattern.

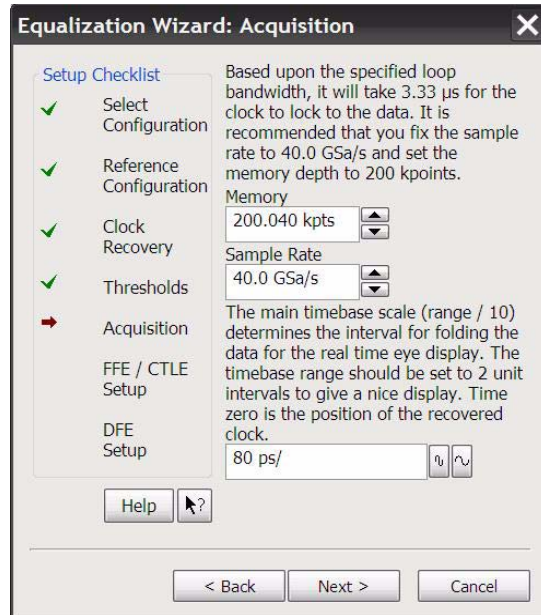


- **Autoscale Vertical:** this button automatically sets the vertical V/division scale and vertical offset to the optimal values for all channels.

- **Set to 50% Vp-p:** Clicking this button automatically sets the logical threshold level to be $(V_{max} + V_{min})/2$. The logical threshold level is defined to be the voltage level above which the signal is considered to be a logical 1 and below which the signal is considered to be a logical 0. Pressing this button also sets the hysteresis to be 5% of the source waveform's amplitude.
- **Hysteresis:** The hysteresis helps define what is considered an edge by the oscilloscope. For example, the default hysteresis is set to ± 5 mV. This means the signal must go through a 10 mV swing (-5 mV to 5 mV) in order to be considered an edge. If you make this value too large, the oscilloscope may skip edges and incompletely recover the clock. If you make this value too small, the oscilloscope will mistake noise for edges. In most cases, ± 5 mV is a good setting, although you may need to increase or decrease this setting based upon your specific signal and the amount of noise.

Step 5: Acquisition

This step sets up various parameters associated with the oscilloscope's acquisition of the signal.



The fields in this step will automatically be filled in based upon the settings you chose in the clock recovery step. How these values were determined is explained in the text above each field in the GUI and is also discussed below.

- **Timebase Scale:** The timebase range (the width of the display) should be set to two unit intervals in order to produce a nice display. Since the horizontal axis has ten divisions, the timebase scale would then be calculated via:

$$\text{Timebase scale} = \frac{(2 \cdot \text{unit interval})}{10}$$

As an example, if you set the data rate in the clock recovery step to 6.0 Gb/s, the unit interval would be 167 ps (1/data rate). Multiplying 167 ps by two and then dividing by ten gives a timebase scale of 33.4 ps.

- **Sample Rate:** The sample rate is set to the maximum value possible for your specific oscilloscope model.
- **Memory Depth:** There are actually two ways this value is set:

- If you set Constant Frequency as your clock recovery method, the oscilloscope will determine how many samples per unit interval are acquired (it will tell you this value in the text at the top of the dialog box). The oscilloscope will then set the memory depth such that the oscilloscope displays 1,000 unit intervals [(samples /unit interval) x 1000]. Displaying 1,000 unit intervals is arbitrary. If you want to display a different number of unit intervals, multiply the number you want times the samples per unit interval.
- If you use a PLL or Fibre Channel clock recovery algorithm, the oscilloscope will calculate how much time it takes for the clock to lock to the data (assumes five time constants for PLL to set up so this time is calculated via $5/(\text{loop bandwidth})$). The oscilloscope will then recommend the memory depth based upon the following equation:

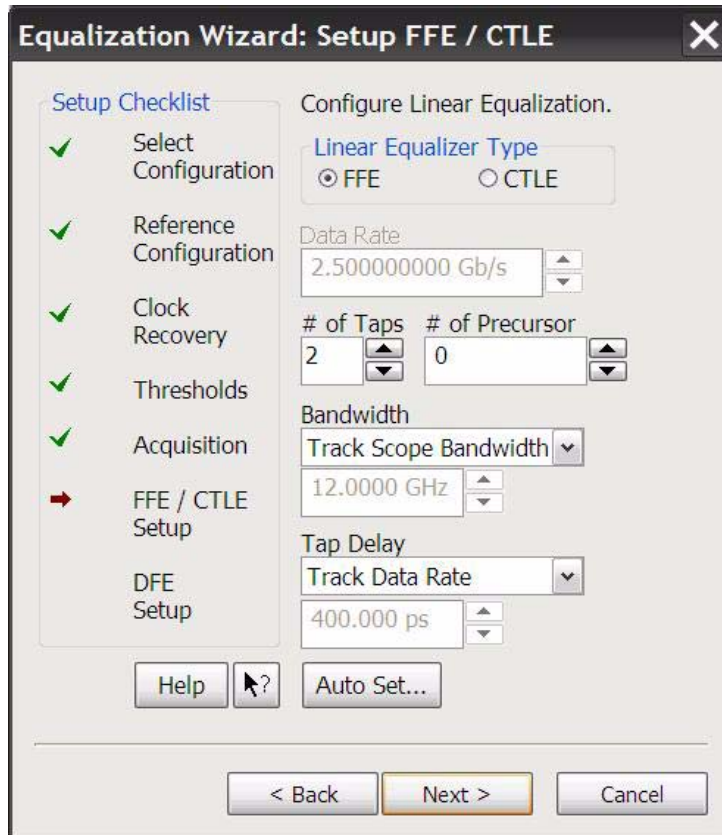
$$\text{Memory depth} = \left(\frac{5}{\text{loop bandwidth}} \right) \cdot \frac{3}{2}$$

where five is the number of time constants for PLL to settle and $3/2$ is used so some data after the PLL settles is included.

Step 6: FFE/CTLE Setup

This step allows you to set various parameters associated with FFE or CTLE. If you selected choices in Steps 1 and 2 that did not involve FFE/CTLE, this step will be skipped.

Here is what the dialog looks like when FFE is selected.



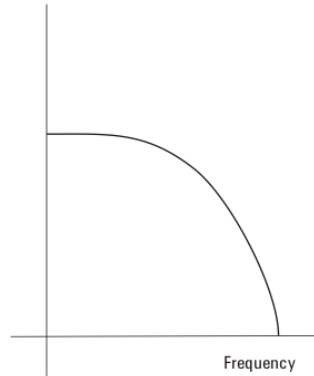
- **Data Rate:** This field is automatically filled in based upon the settings you provided in the clock recovery step.
- **# of Taps:** This field allows you to enter the number of taps to use (up to 40). For more information on what taps are, how to determine the number of taps to use, and how to set their values, refer to [page 33](#).

- **# of Precursor Taps:** You can also specify the number of precursor taps you want to use by using the **# of Precursor Taps** control. Remember that with standard FFE, the current bit is equalized using its tap value along with tap values associated with bits prior to it. Precursor taps are tap values associated with bits in front of the current bit. For example, if you set **# of Precursor Taps** to zero and you were using three taps then the equalization would use the current bit's tap value, the previous bit's tap value, and the bit's tap value that was two behind the current bit. Now, if you instead set the **# of Precursor Taps** control to 1 then the current bit's tap value, the previous bit's tap value, and the bit in front of the current bit's tap value would be used. The **# of FFE Taps** is the total number of taps you want to use and the **# of Precursor Taps** is the number of taps from this total that will be associated with bits in front of the current bit.
- The **Auto Set ...** button automatically determines the tap values for you through optimization (the Equalization Wizard assumes you do not know the tap values and are using the oscilloscope to optimize the values. If you want to enter the taps manually, you must set up the equalization manually instead of using the Equalization Wizard.). For a more in-depth description of this optimization, refer to page [page 38](#).

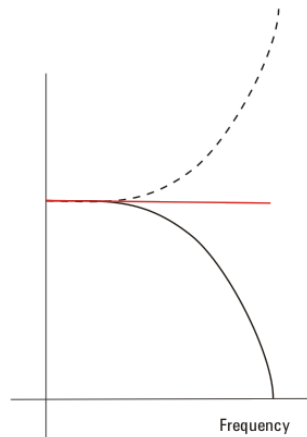
- **Bandwidth:** To understand this parameter, we must first discuss some aspects of equalization.

As discussed in [Chapter 2](#), one way to think of equalization is in terms of the frequency response of the channel your signal travels through. Ideally, you would like to transmit a signal through a channel whose frequency response was flat over the full bandwidth of the signal so every frequency component of the signal was equally represented.

In practice, your frequency response plot may look something like the plot below where the signal attenuates at higher frequencies (this is a very simplified example). This is what causes ISI and makes the eye diagram close.

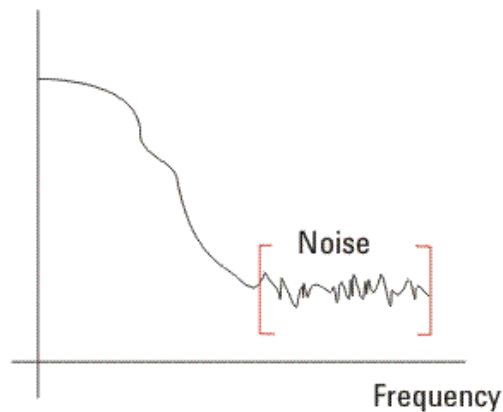


To correct for this attenuation, we can apply a peaking filter response (dashed line in plot below) such that the sum of the peaking filter response and the channel response yields a flat frequency spectrum (red line in plot below).



Figuring out what this response should be and applying it is what you are doing when you perform equalization.

Now, in the above example, the unequalized response plot was very simplistic. A more realistic frequency response plot might look more like the following:



Notice after a certain frequency (or bandwidth) there is very little data and mostly noise. If you apply a response via equalization as discussed above, you will end up amplifying the noise as well as the data. The **Bandwidth** field in this dialog box allows you to minimize this noise gain by rolling off the response above the frequency where there is mostly just noise.

The three possible choices for the **Bandwidth** field are

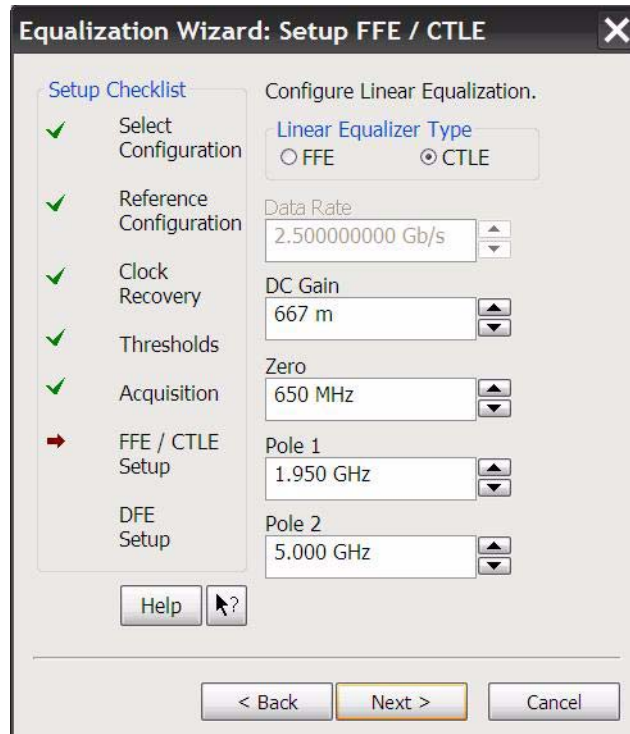
listed on the next page.

- **Track Scope BW** (default) - This option simply generates a response across the entire bandwidth of the oscilloscope. Therefore, if you are using a 12 GHz oscilloscope, the response will go to 12 GHz before rolling off. This is the method that maximizes the noise gain because the noise may be the dominant factor before the scope BW.
 - **Track Tap Delay** - This forces the response to roll off at a typical ratio of bandwidth to data rate (0.75 x data rate). This will minimize the noise and is best used if you are unsure where in the frequency spectrum the noise is the dominant factor, but would like to minimize the noise gain.
 - **Manual** - If you know at what frequency you want the response to roll off, you can enter it manually.
- **Tap Delay:** Usually when you set the number of taps, it will be at a rate of 1 tap/bit. If you are indeed using 1 tap/bit then the tap delay is defined as (1/data rate) and you can set it to this by selecting the **Track Data Rate** choice for this field.

However, you can also set up the equalization such that multiple taps/bit are used. To do this, first set the **Tap Delay** field to **Manual**. Then compute how large to make the tap delay via the following equation:

$$\text{Tap delay} = \frac{1}{[(\text{data rate}) \cdot (\# \text{ of taps per bit})]}$$

Here is what the dialog looks like if CTLE is selected.



The filter applied to your signal via CTLE is described by the equation shown below:

$$H(s) = \frac{A_{dc} \omega_{p1} \omega_{p2}}{\omega_z} \cdot \frac{s + \omega_z}{(s + \omega_{p1})(s + \omega_{p2})}$$

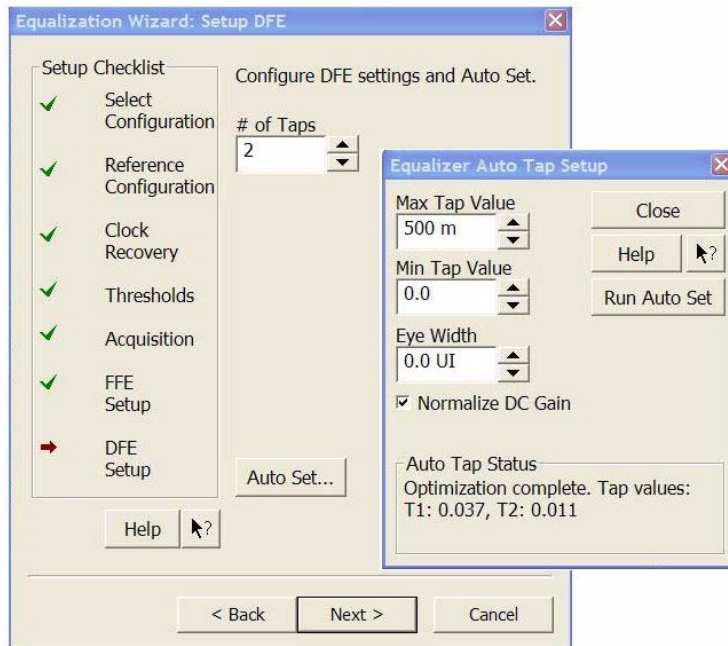
You must specify four variables in this equation in order to generate the filter. These variables are:

- DC Gain (A_{DC})
- Zero Frequency (ω_z)
- Pole 1 (ω_{p1})
- Pole 2 (ω_{p2})

Step 7: DFE Setup

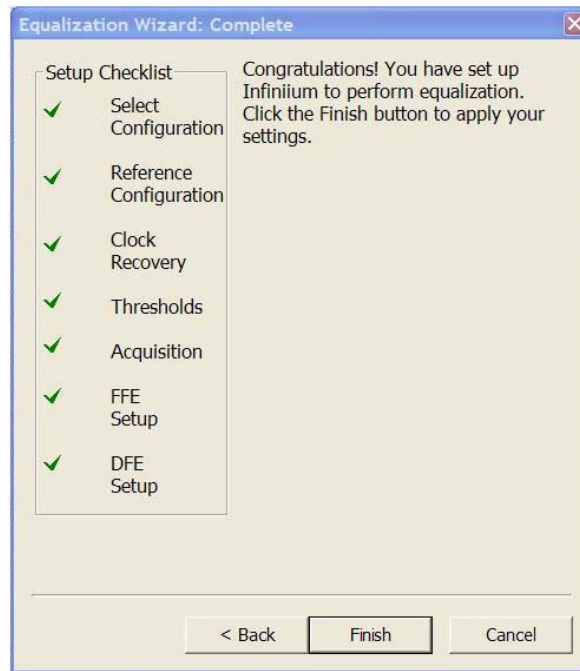
This step allows you to set various parameters associated with Decision Feedback Equalization. If you selected choices in Steps 1 and 2 that did not involve DFE, this step will be skipped.

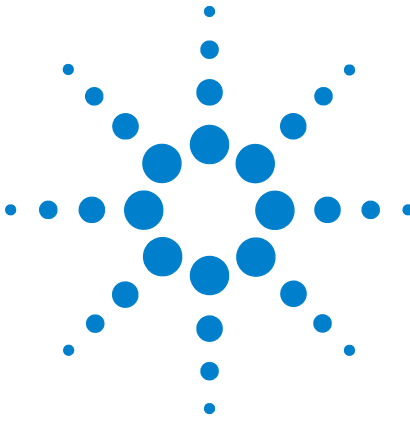
- **# of Taps:** Specify the number of taps to use in the Decision Feedback Equalization. For more information on taps and how to choose the correct number of taps, refer to page [page 33](#).
- **Auto Set ...:** After you set the number of taps, press this button to have the tap values determined by the oscilloscope via optimization. After pressing this button, the **Equalizer Auto Tap Setup** dialog box will appear (see screen shot below). For more information on the optimization process and on using the **Equalizer Auto Tap Setup** dialog box, refer to page [page 41](#).



Step 8: Complete

Once you see the following dialog box, the Equalization Wizard is complete. Press the **Finish** button and after several seconds, you should see the eye diagram(s) displayed on your oscilloscope.





8 Manual Equalization Setup

Introduction

Manually entering the settings is one of the two ways to set up the equalization feature (the other way is to use the Equalization Wizard discussed in [Chapter 7](#)). The Equalization Wizard is the recommended setup method as it walks you through the process in a step-by-step fashion. Only use the manual setup method if you are comfortable with the manual controls and are familiar with equalization.

Getting Your Signal Set up

Before entering the Equalization dialog box, set up the channel, function, or waveform memory you want to equalize and turn on the real-time eye feature via the **Serial Data** dialog box (**Analyze -> Serial Data**). You will also need to set the sample rate, memory depth, and timebase scale using the following recommendations.

- **Sample Rate:** (accessed via **Setup -> Acquisition**) This is generally set to the maximum possible value.
- **Timebase Scale:** (accessed via **Setup -> Horizontal**) The timebase range (the width of the display) should be set to two unit intervals in order to produce a nice display. Since the horizontal axis has ten divisions, the timebase scale would then be calculated via:

$$\text{Timebase scale} = \frac{(2 \cdot \text{unit interval})}{10}$$

As an example, if your signal's data rate is 6.0 Gb/s, the unit interval would be 167 ps (1/data rate). Multiplying this by two and then dividing by ten would give you a timebase scale of 33.4 ps.



- **Memory Depth:** (accessed via **Setup** -> **Acquisition**) If you are going to use Constant Frequency as your clock recovery method, you need to first determine how many samples per unit interval will be acquired based upon the data rate of your signal and the sample rate of your oscilloscope.

As an example, assume your signal's data rate is 6 Gb/s (so your unit interval is [1/data rate] or 167 ps) and your sample rate is 20 GSa/s. You could determine the samples per unit interval by multiplying your unit interval times your sample rate and then converting units as in the following:

$$\frac{\text{Samples}}{(\text{Unit interval})} = \frac{(20\text{GSa})}{\text{s}} \cdot \frac{(10^9 \text{ Sa})}{(1\text{GSa})} \cdot \frac{(1 \text{ s})}{(10^{12} \text{ ps})} \cdot \frac{167 \text{ ps}}{1 \text{ unit interval}}$$

$$\frac{\text{Samples}}{(\text{Unit interval})} = 3.34 \text{ Sa/unit interval}$$

Determine how many unit intervals you want to display and multiply the number you want times the samples per unit interval to calculate your needed memory depth.

If you use a PLL or Fibre Channel clock recovery algorithm, the recommended memory depth is based upon the following equation:

$$\text{Memory depth} = \left(\frac{5}{\text{loop bandwidth}} \right) \cdot \frac{3}{2}$$

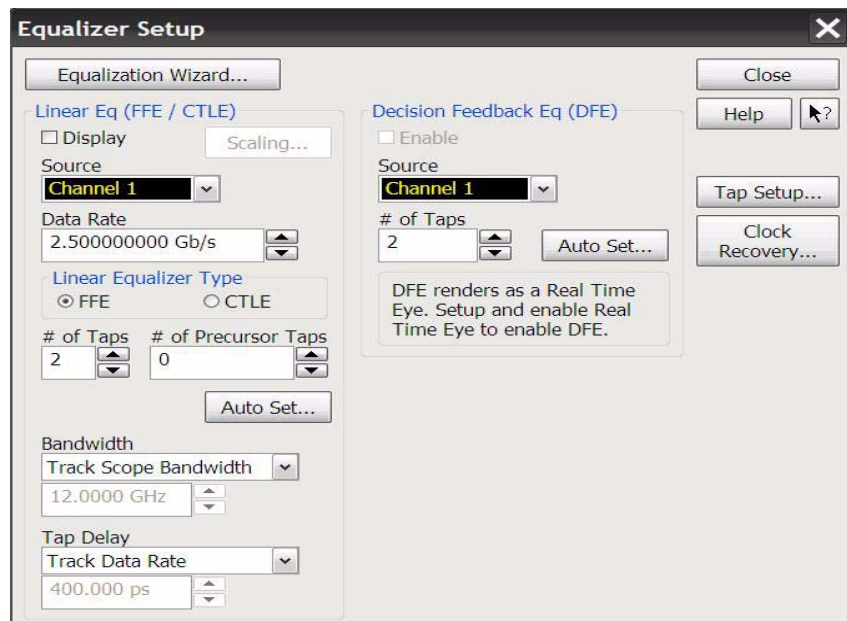
where five is the number of time constants for PLL to settle and 3/2 is used so some data after the PLL settles is included.

- **Thresholds:** (accessed via **Analyze** -> **Serial Data** and then clicking on the **Thresholds** button) In the **Measurements Thresholds** dialog box, set the **Thresholds** field to **Custom: Level +/- Hysteresis**. For most differential signals, set the **Threshold Level** to 0 V. You can also manually set the threshold level to whatever value you want or set it to 50% of the peak-to-peak voltage.

- **Set to 50% Vp-p:** Clicking this button automatically sets the logical threshold level to be $(V_{max} + V_{min})/2$. The logical threshold level is defined to be the voltage level above which the signal is considered to be a logical 1 and below which the signal is considered to be a logical 0. Pressing this button also sets the hysteresis to be 5% of the source waveform's amplitude.
- **Hysteresis:** The hysteresis helps define what is considered an edge by the oscilloscope. For example, the default hysteresis is set to ± 5 mV. This means the signal must go through a 10 mV swing (-5 mV to 5 mV) in order to be considered an edge. If you make this value too large, the oscilloscope may skip edges and incompletely recover the clock. If you make this value too small, the oscilloscope will mistake noise for edges. In most cases ± 5 mV is a good setting, although you may need to increase or decrease this setting based upon your specific signal and the amount of noise.



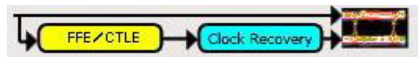

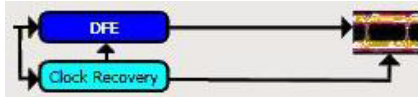
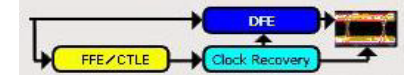
The Equalizer Setup Dialog Box

When you select **Acquire -> Equalization** from the top menu bar on the display, you will see the following **Equalizer Setup** dialog box. This dialog box will be where you manually enter your equalization settings.



The Available Equalization Setups

There are several different equalization setups you can apply to your signal. Each of these are discussed in the table below.

| Graphical Representation of Setup | Description of Setup | How to Manually Set Controls to Obtain Setup |
|---|---|---|
|  | <p>FFE/CTLE is applied, but the real-time eye is not shown. Instead the waveform resulting from applying FFE/CTLE is displayed.</p> | <ul style="list-style-type: none"> • In the Serial Data dialog box (Analyze -> Serial Data), uncheck the Real-time Eye box. • In the Equalizer Setup dialog box, check the Display box under the <i>FFE</i> section and select either FFE or CTLE. • Click on the Clock Recovery button and choose your clock recovery method. |
|  | <p>No equalization is implemented and the unequalized real-time eye is displayed. This is useful if you want to have a reference to compare to. For instance, you can put the unequalized eye in the top grid and the equalized eye in the bottom grid so you can compare them on the same screen.</p> | <ul style="list-style-type: none"> • Do not check the Display box under the <i>FFE</i> section or the Enable box under the <i>DFE</i> section. • Click on the Clock Recovery button and choose your clock recovery method. |
|  | <p>FFE/CTLE is only used to recover the clock, but the actual displayed real-time eye is unequalized. This is useful if you want to have a reference to compare to, but your unequalized eye is completely closed. You cannot recover a clock on a closed eye, so FFE/CTLE is applied to open the eye so the clock can be recovered. In other words, this option serves the same purpose as the selection immediately above it, but is used with closed eyes.</p> | <ul style="list-style-type: none"> • Do not check the Display box under the <i>FFE</i> section or the Enable box under the <i>DFE</i> section. • Click on the Clock Recovery button and choose either Equalized First Order PLL or Equalized Second Order PLL as your clock recovery method. |
|  | <p>A FFE/CTLE eye is displayed on the oscilloscope. This is a standard FFE implementation and can be used on partially or completely closed eyes.</p> | <ul style="list-style-type: none"> • Check the Display box under the <i>FFE</i> section and choose either FFE or CTLE. • Click on the Clock Recovery button and choose your clock recovery method. |
|  | <p>A Decision Feedback Equalized eye is displayed on the oscilloscope. This structure cannot be used on completely closed eyes as the clock cannot be recovered with this method. If the eye is closed and you want to implement DFE, use the option immediately below this.</p> | <ul style="list-style-type: none"> • Check the Enable box under the <i>DFE</i> section. • Click on the Clock Recovery button and choose your clock recovery method. |
|  | <p>FFE/CTLE is used to recover the clock (but is not implemented in the real-time eye display) and a Decision Feedback Equalized eye is displayed on the oscilloscope. This allows you to perform DFE on completely closed eyes.</p> | <ul style="list-style-type: none"> • Check the Enable box under the <i>DFE</i> section. • Click on the Clock Recovery button and choose either Equalized First Order PLL or Equalized Second Order PLL as your clock recovery method. |

| Graphical Representation of Setup | Description of Setup | How to Manually Set Controls to Obtain Setup |
|-----------------------------------|---|---|
| | <p>FFE/CTLE is applied and then DFE is applied to the real-time eye. Additionally, since FFE/CTLE also is used to recover the clock, this structure can be used on completely or partially closed eyes.</p> | <ul style="list-style-type: none"> • Check the Enable box under the <i>DFE</i> section and select FFE/CTLE as your source in the <i>DFE</i> section. • Click on the Clock Recovery button and choose your clock recovery method. |

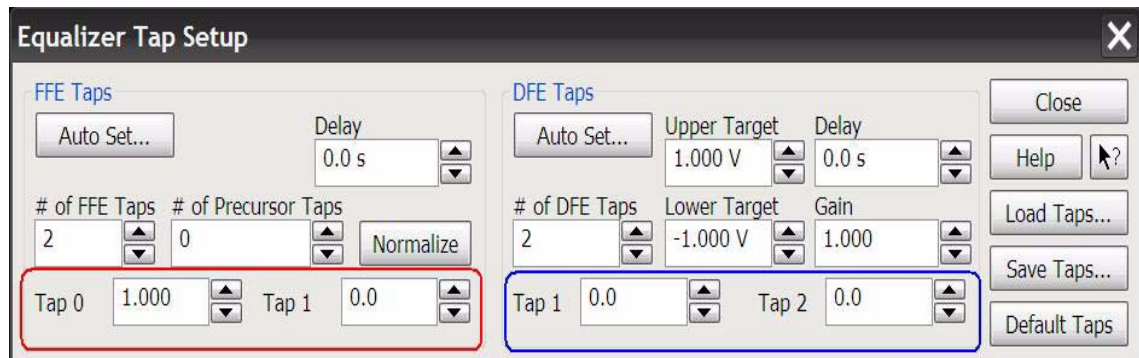
Equalizer Setup Dialog Box Controls

After you have your initial setup ready from following the directions in the table above, you can set the rest of the parameters in the **Equalizer Setup** dialog box.

Taps Control

The **# of Taps** control lets you set how many taps to use in the equalization (for more information on how to determine the number of taps to use refer to [page 33](#)).

You can also choose to have the oscilloscope determine the tap values via optimization or you can manually enter the values.



To optimize the tap values, click on the **Auto Set ...** button in either the *FFE* or *DFE* section (depending on which equalization method you are using). Please refer to [page 38](#) for more information on the optimization process and how to set up the controls in the **Auto Set** dialog box.

To set the tap values manually, click on the **Tap Setup ...** button on the right hand side of the **Equalizer Setup** dialog box. You will then see the following dialog box.

The area circled in red is where you can manually enter FFE taps and the area circled in blue is where you can manually enter DFE taps. The number of taps can be adjusted with the **# of FFE Taps** or **# of DFE Taps** fields.

Notice the GUI only allows you to manually adjust up to 16 tap values when in fact you can use up to 40 taps. To see how to access the rest of these taps, refer to page [page 34](#).

FFE Taps Section

Equalization may cause your eye diagram to drift so the clock is no longer centered on the eye. The **Delay** field allows you to specify this shift in order for the eye to be brought back to its original location relative to the unequalized signal.

The **Normalize** button makes the tap values sum to one in order for the output to have unity gain at dc. This allows you to see the same size eye diagram at the receiver as you do at the transmitter. If you want to use this feature, first enter your tap values and then press the **Normalize** button. Your tap values will be adjusted so they sum to one.

You can also specify the number of precursor taps you want to use by using the **# of Precursor Taps** control. Remember that with standard FFE, the current bit is equalized using its tap value along with tap values associated with bits prior to it. Precursor taps are tap values associated with bits in front of the current bit. For example, if you set **# of Precursor Taps** to zero and you were using three taps then the equalization would use the current bit's tap value, the previous bit's tap value, and the bit's tap value that was two behind the current bit. Now, if you instead set the **# of Precursor Taps** control to 1 then the current bit's tap value, the previous bit's tap value, and the bit in front of the current bit's tap value would be used. The **# of FFE Taps** is the total number of taps you want to use and the **# of Precursor Taps** is the number of taps from this total that will be associated with bits in front of the current bit.

DFE Taps Section

The **Upper Target** and **Lower Target** fields dictate the high and low logical values used in the DFE algorithm. For example, in DFE, when a bit is determined to be a logical high, its value will be equal to the Upper Target. Similarly, if a bit is determined to be a logical low, its value will be equal to the Lower Target.

Equalization may cause your eye diagram to drift so the clock is no longer centered on the eye. The **Delay** field allows you to specify this shift in order for the eye to be brought back to its original location relative to the unequalized signal.

Remember in the discussion on DFE that the eye drawn after DFE is applied is attenuated. To amplify the eye back to its original size (so you can directly compare the eye at the receiver to the eye at the transmitter) a gain factor needs to be applied. The **Gain** field allows you to enter a constant that is applied to both the unequalized input signal and the corrected decision threshold value. Setting this field to 1 means there is no gain applied to the DFE eye diagram (like multiplying by 1).

Tap values can also be saved and loaded by using the **Save Taps ...** or **Load Taps ...** buttons. These buttons save and/or read your taps from a .xml file that can then be shared with colleagues or used by you at a later date so you do not have to enter the taps again manually.

Data Rate Control

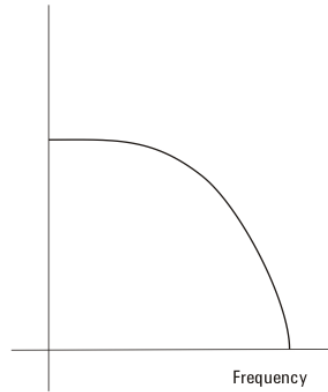
Enter the data rate of your signal into this field.

Bandwidth Control

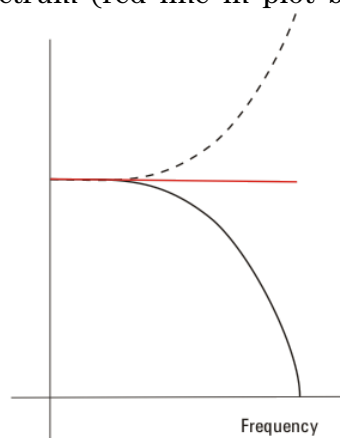
To understand this parameter, we must first discuss some aspects of equalization.

As discussed in [Chapter 2](#), one way to think of equalization is in terms of the frequency response of the channel your signal travels through. Ideally, you would like to transmit a signal through a channel whose frequency response was flat over the entire bandwidth of the signal so every frequency component of the signal was equally represented.

In practice, your frequency response plot may look something like the simplified plot below where the signal attenuates at higher frequencies. This is what causes ISI and makes the eye diagram close.

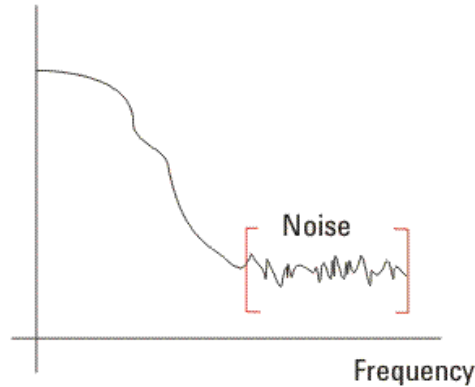


To correct for this attenuation, we can apply a peaking filter (dashed line in plot below) such that the sum of the peaking filter response and the channel response yields a flat frequency spectrum (red line in plot below).



Figuring out what this response should be and applying it is what you are doing when you perform equalization.

Now, in the above example, the unequalized response plot was very simplistic. A more realistic frequency response plot might look more like the following:



Notice that after a certain frequency (or bandwidth) there is very little data and mostly noise. If you apply a response via equalization as discussed above, you will end up amplifying the noise as well as the data. The **Bandwidth** field in this dialog box allows you to minimize this noise gain by rolling off the response above the frequency where there is mostly just noise.

There are three options for the Bandwidth field:

- **Track Scope BW** (default) - this option simply generates a response across the entire bandwidth of the oscilloscope. Therefore, if you are using a 12 GHz oscilloscope, the response will go to 12 GHz before rolling off. This is the method that maximizes the noise gain because the noise may be the dominant factor before the scope BW.
- **Track Tap Delay** - This forces the response to roll off at a typical ratio of bandwidth to data rate ($0.75 \times$ data rate). This will minimize the noise and is best used if you are unsure where in the frequency spectrum the noise is the dominant factor, but would like to minimize the noise gain.
- **Manual** - If you know at what frequency you want the response to roll off, you can enter it manually.

Tap Delay Control

Usually when you set the number of taps, it will be at a rate of 1 tap/bit. If you are indeed using 1 tap/bit then the tap delay is defined as (1/data rate) and you can set it to this by selecting the **Track Data Rate** choice for this field.

However, you can also set up the equalization such that multiple taps/bit are used. To do this, first set the **Tap Delay** field to **Manual**. Then compute how large to make the tap delay via the following equation:

$$\text{Tap delay} = \frac{1}{[(\text{data rate}) \cdot (\# \text{ of taps per bit})]}$$

www.agilent.com

© Agilent Technologies, Inc. 2009

Second edition, December 2009



N5461-97001