Summer 8-10-2016

Practical Wired Digital Communications Link Analysis

Raymond Matthew Schmelzer
Portland State University

Practical Wired Digital Communications Link Analysis

by

Raymond Michael Schmelzer

A thesis submitted in fulfillment of the
requirements of the degree of

Master of Science
in
Electrical and Computer Engineering

Thesis Committee:
Y.C. Jeng, Chair
Richard Campbell
Branimir Pejcinovic

Portland State University
2016

ABSTRACT

This thesis deals with the analysis of a Wired High Speed Serial Data Link (PAM2) which is commonly used throughout the data-communications and tele-communications industry. The goal of this study is to build a scalable simulation tool using Matlab that ultimately uses Receiver Bit Error Ratio (BER) as the metric for data link health. This study is also designed to aid in link specification development.

The Matlab and theoretical development is broken up into three sections being Transmitter (TX), Channel (Hs) and Receiver (RX). Realistic noise impairments can be added to each section along the signal path creating signal stresses commonly seen in data center applications. The TX function is designed to create random and periodic timing jitter, voltage noise and deterministic pre-distortion filtering effects. For the channel response s-parameters are used as the model result for many commonly seen channel loss and reflection scenarios. The RX model uses signal to noise ratio and vertical eye margin to determine the equalized link BER.

The study results show many tradeoffs between noises, RX Equalizer, RX gain and RX BER. The simulation results also reveal that there is no closed form solution for converging the modern closed-eye PAM2 detector.

CONTENTS

FIGURES

TABLES

1 INTRODUCTION

The modern network server room sometimes consists of vertical, scalable racks of network servers which in many cases connect to a ToR (Top of Rack) data switch through a lossy passive copper cable utilizing relatively simple high speed (10Gbps and greater) binary communications. The next communication link from the ToR switch to what is sometimes called the EoR (End of Row) switch can also potentially be linked through the same style of lossy copper cable and signaling, but this data center configuration can drive the simple 2 level detectors currently available in the market beyond their capabilities. 10GBaseT Ethernet will be briefly discussed as a non-fiber optic method to cover the channel distance to the EoR switches, but first the failure of the simpler link will be analyzed. These passive copper communications cables can range from 0.5m to ~10m in length. This range of channel lengths force the need for a receiver to properly set an Electronic Dispersion Compensation (EDC) or simply put, equalizer to compensate for a wide range of frequency dependent channel losses without over amplifying noise. The receiver equalizer also has to adapt its settings in a non-Gaussian random noise environment. Many other complex engineering tasks have to happen to create a high speed link, but bit detection at a specified Bit Error Rate (BER) must occur first and equalizing the frequency dependent physical layer effects is key to this goal.

Since the lossy copper communications cable directly connects two network controller PHY (physical layer) devices through a passive channel they are sometimes referred to

as a Direct Attach Cables (see SFF-8431 specification). Non-copper communications links such as single-mode and multi-mode optical links are not in the scope of this study. In this thesis many of the communications impairments present in the architecture of *figure1* are analyzed and simulated in Matlab using receiver BER as the ultimate

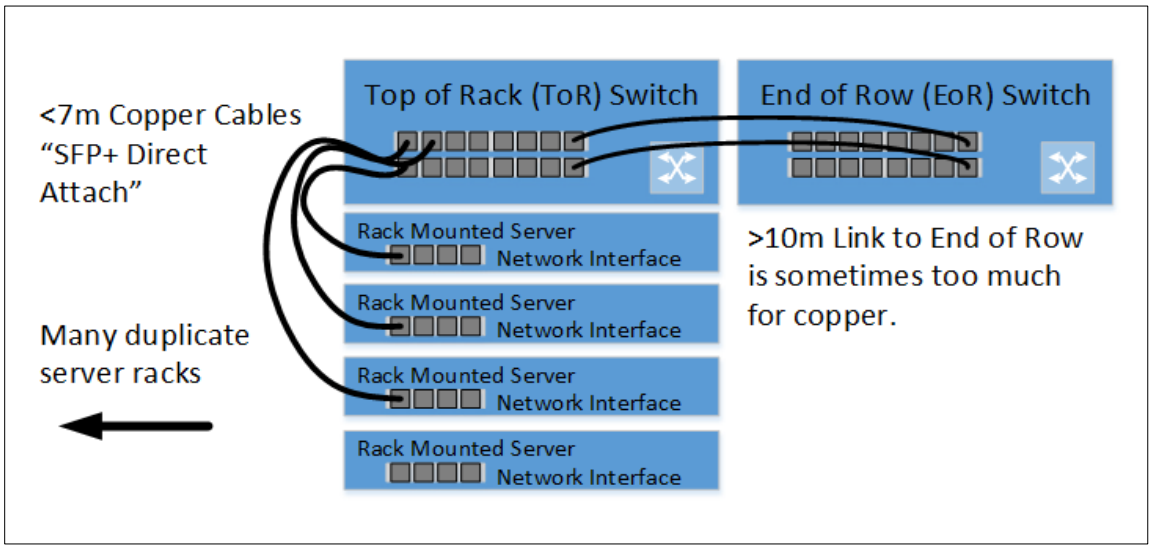


Figure 1, Rack mounted data servers connected to data switches via lossy copper cable.

measure of link quality. The problem is broken down into three sections, transmitter (TX), channel (H(s)), receiver (RX) and each is related to increasing BER and relevant lab/conformance measurements. Where applicable industry specifications such as 10GBaseKR, SFP+ and 40GBaseCR4 are used for reference and to solidify points. Finally, after a practical Matlab link simulation environment is developed and proven it is used to improve BER performance through noise, ISI (Inter Symbol Interference) reduction. Each section will go into enough detail required to make specification requirement decisions, but not so detailed as to lose focus on the communications system.

This study's main contribution is to provide a specification writer a framework to properly budget a wired communications link while taking into account real signal impairments. It is not meant to be a tool to model analog communications circuits.

1.1 System Description

The following diagram (*figure2)* is a high level model of one direction of a 10Gbps high speed serial link. Along the signal path measurement points are defined so signal stresses can be analyzed at points along the transmission path.

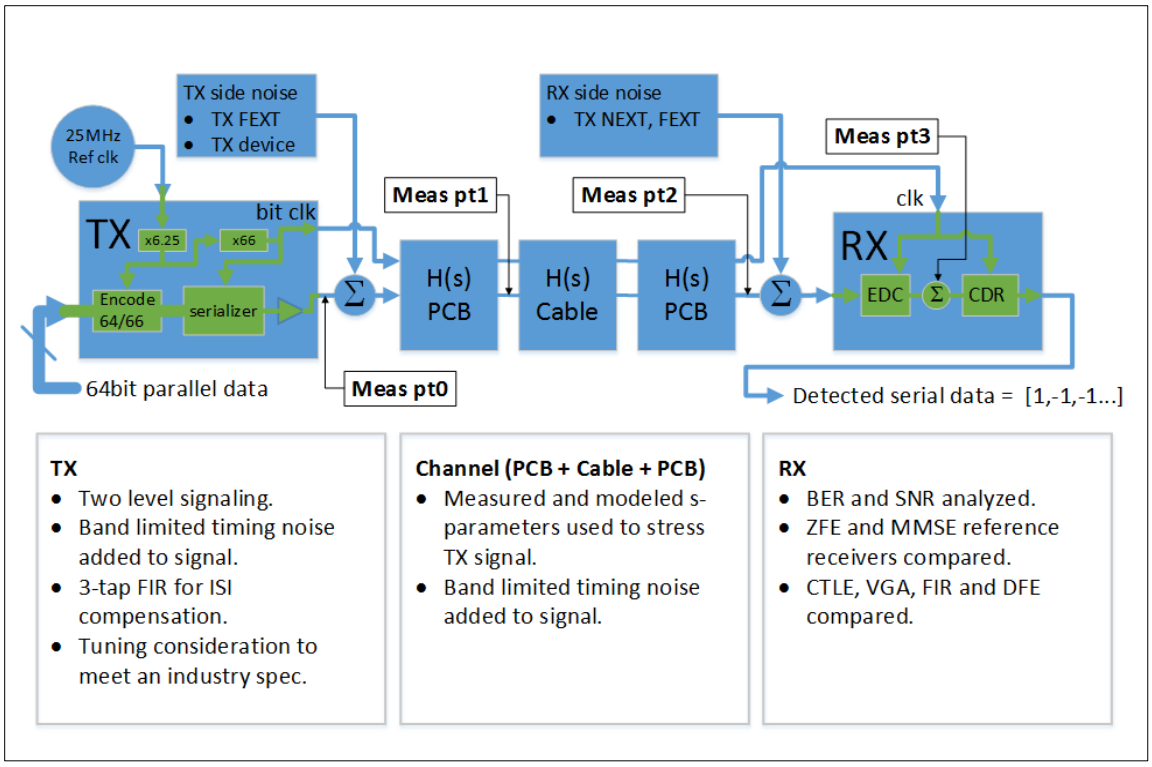


Figure 2, Detailed system diagram of physical layer wired communications system.

The measurement/analysis points are defined as follows:

Meas pt0 – This measurement point is the signal after the band limited nature of typical 3tap FIR transmitter. Timing noise (jitter) and device amplitude noises are included at this point.

Meas pt1 – Includes all of pt0 with the addition of printed circuit board (PCB) stresses, which includes Inter-Symbol Interference (ISI) and Far End Cross Talk (FEXT).

Meas pt2 – Includes all of pt0 and pt1 with the addition of twin axe cable ISI and link partner PCB stressors. Link partner PCB stresses include Near End Cross Talk (NEXT) from adjacent TX functions and ISI.

Meas pt3 – Includes all of pt0 thru pt2 with the addition of noise amplification of the EDC. SNR and or distance from bit decision threshold is measured at pt3.

This is a large complex system with a career full of late nights working thus simplifications such as knowing the proper TX clock (noise and delay) at the RX have to be made to keep the scope of this study reasonable.

2 TRANSMITTER (TX)

In this section a scheme is introduced to transmit 66 bit parallel data out to a 10.3125Gbps serial stream with a 25MHz clock as the stable reference [Derickson]. Also, a noisy (jitter and amplitude) PAM2 signal with an n-tap Finite Impulse Response Filter (FIR) is described, analyzed and created in Matlab. Referring to system description section in chapter 1 this signal will be analyzed at measurement points pt1 and pt2.

2.1.1 TX Signal from Reference Clock to Data Transmission

In 10Gbps High Speed Serial Signaling the timing of the 10Gbps TX signal generally begins with a very accurate, but lower frequency crystal oscillator circuit. A 25MHz reference clock is assumed for this study. The 25MHz reference has to be multiplied in such a way that a parallel bit stream from a higher network layer (Media Access Control) can be properly encoded, scrambled and ultimately in this case shifted out a serial register at the serial data rate of 10.3125Gbps. *Figure3* shows a possible way to implement the 10GBaseR encoding that is used in many 10Gbps High Speed Serial (HSS) Ethernet specifications [7].

Before going further into Matlab TX signal synthesis a brief description of the 10GBaseR encoding is in order. The parallel data path from the upper layer of the 10GBaseR encoded data can be 64bits wide so a 25MHz ref clock needs to be multiplied by 6.25 using a PLL based frequency synthesizer in order to get a bit rate (br) of 10.0000Gbps.

$$br\ parallel = 25MHz * 6.25 * 64bits = \ 10\frac{Giga\ bits}{second}$$

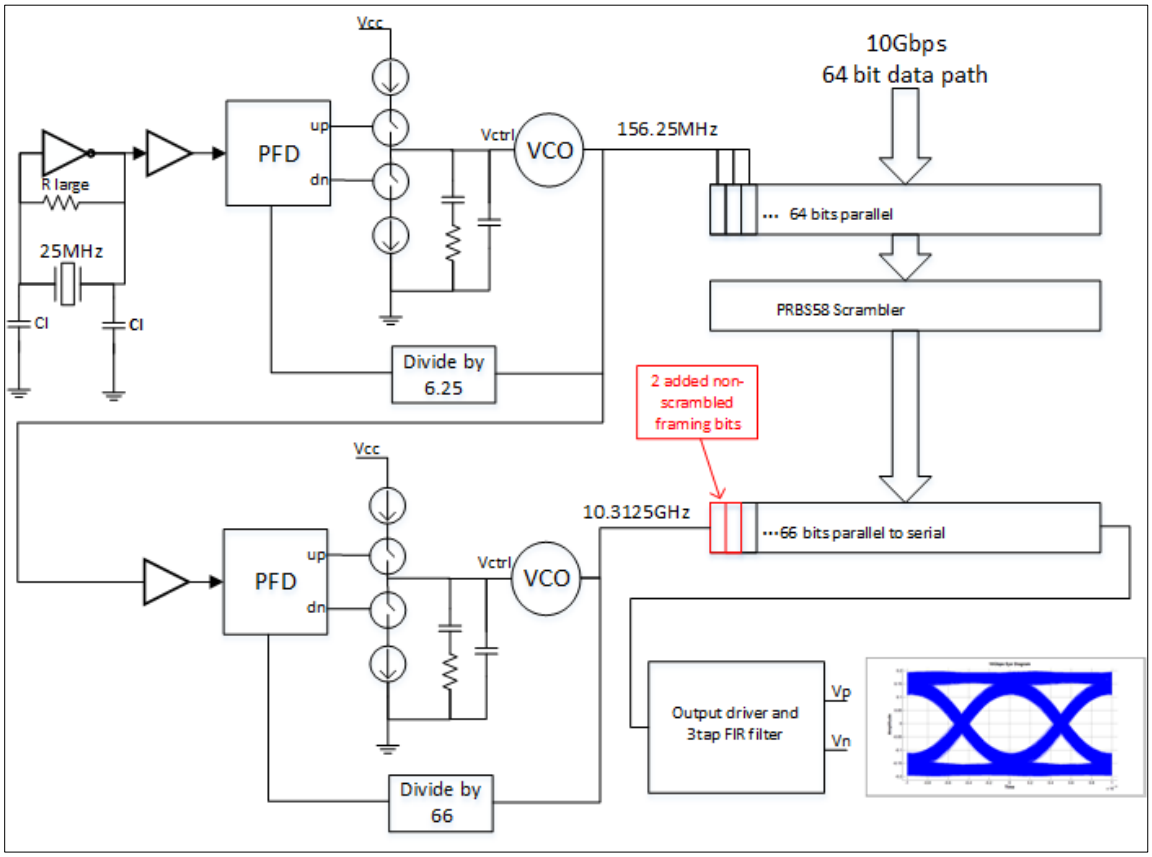


Figure 3, Parallel to serial data path and clocking diagram.

In the case of 10Gbps Ethernet 2 sync or framing bits are added by way of a 64b/66b encoder block which makes the bit rate 10.3125Gbps in order to keep the 10Gbps data rate.

$$br\ parallel\ encoded = 25MHz * 6.25 * 66bits = \ 10.3125\ Gbps$$

To turn the 66 bits of parallel data into a serial bit stream a 10.3125GHz clock can be used to drive a serializer which creates the 10.3125Gbps serial bit stream. This clock can be created by multiplying in frequency the 156.25 MHz clock by 66.

$$br\ serial\ encoded = 25MHz * 6.25 * 66 = \ 10.3125\ GHz$$

Ultimately the 10.3125 Gbps serial data drives an analog output buffer circuit which utilizes a multi tap FIR filter used to assist with link equalization [7].

The TX function description in this section reveals many avenues for random and non-random noises (power supply, frequency synthesizer phase noise, crystal circuit phase noise) to cause timing noise (jitter) on the HSS TX signal. Reference clock noise transfer through to the TX will be discussed in a later section. In the next section the breakdown of TX jitter is discussed.

2.1.2 TX Jitter Breakdown into Components

TX jitter is signal timing noise which in some cases diminishes the BER margin at the far end RX. "In some case" means that some jitter can be equalized and some jitter cannot thus certain jitter components don't necessarily have to raise BER. Inter-symbol interference is a good example of a jitter component that is fully equalizable with the proper filtering.

In this section jitter is broken down into its sub components and described. Reasonable simplifications are made with respect to jitter components to keep the scope of the study manageable. First it is useful to start with the basic jitter breakdown diagram a shown in *figure4* and a description of each jitter component.

Random Jitter (Rj) is the timing noise based on random effects which can be largely based on the VCO of a clock synthesizer and reference clock noise [2].

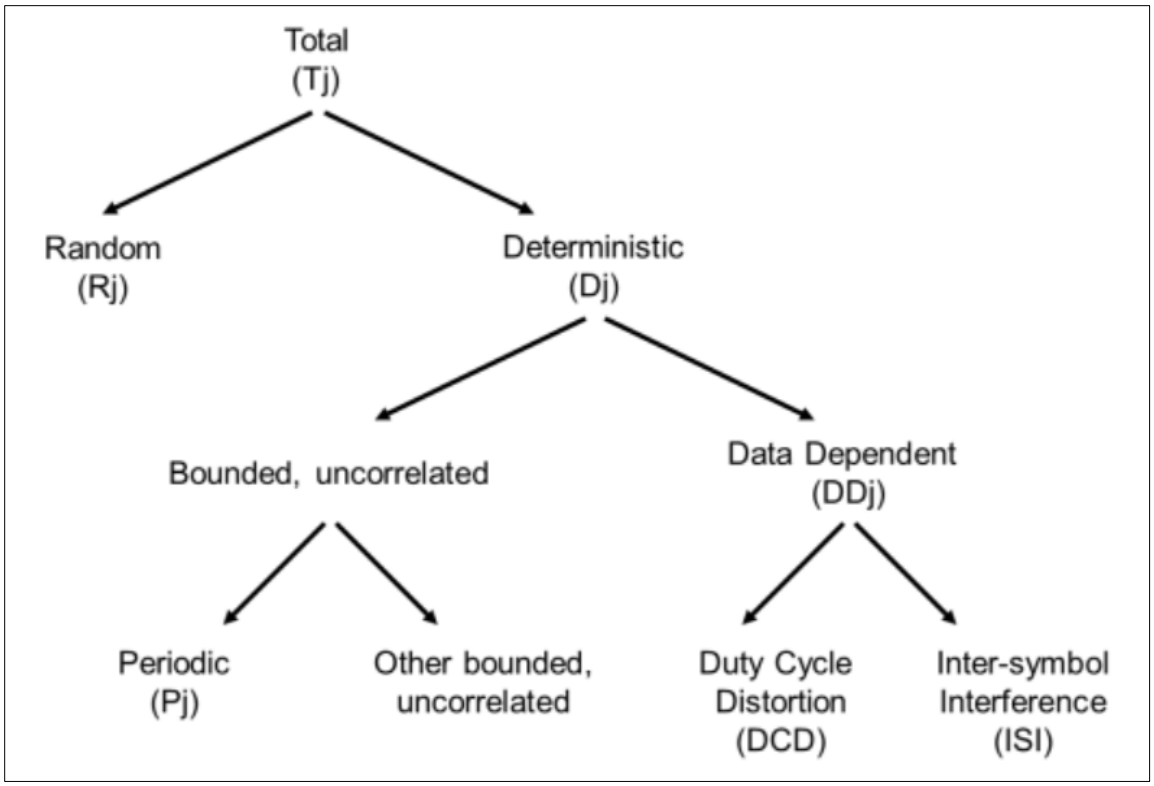


Figure 4, Breakdown of timing noise at the transmitter [2].

Deterministic Jitter (Dj) is made up of all the jitters that aren't truly random. Dj measurements are generally made very accurately with a quality sampling oscilloscope if the specification [2].

Inter-Symbol Interference Jitter (ISI) is a component of Dj and is in theory completely equalizeable by either the TX or RX filters and is only present due to the broad spectral content of a data pattern and frequency dependent nature of the output driver and channel. If the TX jitter analysis is done using a repeating (101010….) pattern then no ISI is present thus a simpler analysis can be made [2].

Duty Cycle Distortion (DCD) is commonly defined as the percent difference between a positive pulse time and a negative pulse time. Physical reasons for DCD can include a

non-symmetric reference level and different propagation delays for negative and positive edges. For this study it is assumed positive and negative pulses are perfectly symmetric for a 101010 pattern thus DCD is not present [2].

Bounded Uncorrelated Jitter (BUJ) timing noises are generally considered un-equalizable for the simple HHS link and must be budgeted into the link BER as such. Physical reasons for BUJ can be lack of isolation from and adjacent HSS channel giving rise to cross talk noise and noise couple into the HSS signal from power supplies. BUJ can be added to the Matlab functions created for this study, but to for now BUJ is assumed zero [2].

Total Jitter (Tj) is the combination of random and deterministic jitter defined at the same probability as the specification BER which in this case is 1E-12. A practical problem in industry is measuring Tj even though the measuring instrument did not directly measure 1/BER number of bits. The reason to extrapolate a higher probability measurement to a lower probability is that a more usable piece of equipment (oscilloscope) can make the measurement rather than the more expensive Bit Error Ratio Tester (BERT). This of course means that jitter components must be separated until a truly random jitter is obtained, σ, and thus extrapolated using the Q function to the low probability [2]. The equation below is commonly used in HSS Tj analysis and it shows that all the deterministic elements must be accurately separated from the random or an inaccurate Tj will result

$$Tj(1E-12) = Dj + (2 * 7.035 * \sigma) \text{ [2]}$$

Some specifications will clearly state that the Tj measurement should come from a direct measurement made to the probability of the BER (1E-12). This means that the no separation needed. In this case the peak to peak value of the jitter distribution is the Tj value.

In stressed RX testing and simulation the different types of jitters are relatively easy to add to a nearly ideal signal thus the components of noises are understood, but as described above the case where an engineer has to measure a real TX signal and separate jitter components in order to meet a specification is not completely agreed on in industry. If a specification is poorly defined and too much is left to interpretation the method that makes the TX look the best to customers is sometime the one used even though it really is not the method used in the system BER budget. Specifications where measurement methods are well defined, like SFF-8431 (SFP+), results in similar measurement values from most labs even another company's lab. A quality specification definition including validation measurement methods with reference to TX, channel and RX also leads to one vendor's device interoperating with another vendor's device at a system level to a proper BER. This makes engineers, managers and most of all end users happy. The next section starts the Matlab signal synthesis discussion.

2.1.3 TX Signal Analytical Description

The binary data stream can be described by starting with a ones and zeros sequence and superimposing unit interval delayed pulse responses over one another to represent a discrete analog signal. Other noises and impairments are added using this foundation.

$$seq = [0,1,1,0,0,0 \ldots \ldots .]$$

$$TX\_pt0(t) = \sum_{n=0}^{seq\ length} seq[n]g(t - nT) + TX\_noise(t)$$

Where $g(t)$ is the band-limited single bit pulse response seen at measurement pt0 described by the following convolution equation:

$$g(t) = \int h\_BT4(\tau)g(\tau - t)d\tau$$

Where $h\_BT4(t)$ is the impulse response for a 4th order Bessel Thomson low pass filter [1].

To get the signal at pt1 the signal at pt0 is convolved with PCB response impulse response, h_pcb(t).

$$TX\_pt1(t) = \int h\_pcb(\tau)TX\_pt0(\tau - t)d\tau + TX\_NEXT(t)$$

In the next several sections the concepts above are expanded to include synthesis of a jittered signal.

2.1.4 Matlab TX Function with Band Limited Jitter

The Matlab algorithm used to create the TX waveform does not exactly follow the equation in section 2.1.1, but the end goal of a noisy and band limited 10Gb signal is achieved. Even though Matlab probably has a single function to create this signal only basic functions are used to preserve fundamental thinking and a clearer understanding of possible solutions to problems when they occur.

First, as in section 2.1.1 a ones and zeros data sequence and bit rate is defined.

$$br = 10.3125\ GHz \qquad seq = [1,1,-1,1,-1,-1 \ldots\ldots]$$

It is convenient to define the zeros as -1's for this signaling. Next the number bits in the seq array plus one is used to calculate the number of clock edges used to create signal timing. A noise free clock is created from

$$t\_clk\_clean = 0 : \frac{1}{br} : \left[length(seq) * \frac{1}{br}\right]$$

where the 1/br is defined as the unit interval or UI. Now that a clean clock has been created it is simple to create a random noise distribution scaled to the amount of random jitter desired using the standard normal function in Matlab.

$$Rj_{array} = Rj_{rms} * RANDN(1, length(t\_clk\_clean)$$

A typical measured 10Gb random jitter is $Rj_{rms} \cong 1ps$. The same method is used for other desired timing noises. To get the noisy clock, t_clk_clean is summed with all timing noise arrays created.

$$sine\_noise = J\_sine\_amp * sin(2 * pi * J\_sine\_freq * t\_clk\_clean)$$

$$t\_clk\_noisy = t\_clk\_clean + Rj_{array} + sine\_noise$$

Now that a noisy clock has been developed it's used to create edges based on transitions in the sequence array. Two points are created on either side of the transition reflecting the 0% to 100% signal rise time and fall time and the signal amplitude. This signal is called signal_frame and can easily be linearly interpolated using the Matlab interp1 function to achieve a constant sample rate Ts. Ts is used to determine total pattern time and number of sample points desired and is also used to calculate the new time array, t_interp. The interpolated signal is called signal_interp.

$$signal\_interp = interp1(t\_frame, signal\_frame, t\_interp, 'linear')$$

This signal becomes more realistic and convincing after it is propagated thru a low pass filter and displayed as an eye diagram which is the subject of the next section.

2.1.5 Band Limited TX Signal with added Jitter

The details of the method used to propagate signals through transfer function is discussed in chapter 3, but it used here in order to show a realistic signal. For this section only a statement what of has been done to the signal will be made. Chapter 3 is dedicated to signal and channel convolution.

In this section eye diagrams with various BWs and added jitter are shown and discussed. First, a quick note on how the TX eye diagrams show jitter. The Matlab eye diagram function written for this study uses a clock recovered from the open eye data signal. The function calculates all signal crossings and determines an average bit rate from which a clean constant comparison clock is created. Most real TX jitter measurements are made with a reference receiver where jitter is calculated after a PLL based clock and data recovery circuit filters out jitter within the bandwidth of the PLL. Later in this study a simplification is made in that the clock is not recovered from the closed eye data rather the clock of the TX is assumed known at the RX. This effectively cancels the TX jitter at the RX and is a simplification that should be fixed to make the simulation more realistic. To recover a clock from closed eye data requires a non-standard algorithm to take guesses at EQ setting and sample phase and test if the correct direction has been taken.

The signal created in section 2.1.4 has some abrupt edges thus unreasonable bandwidth. A linear phase 4th Order Bessel Thomas low pass transfer function is used to filter the signal to 15GHz. This filter is meant to roughly mimic the finite bandwidth of an output driver. *Figure5* shows jitter free signal eye diagrams for PRBS9 signals created

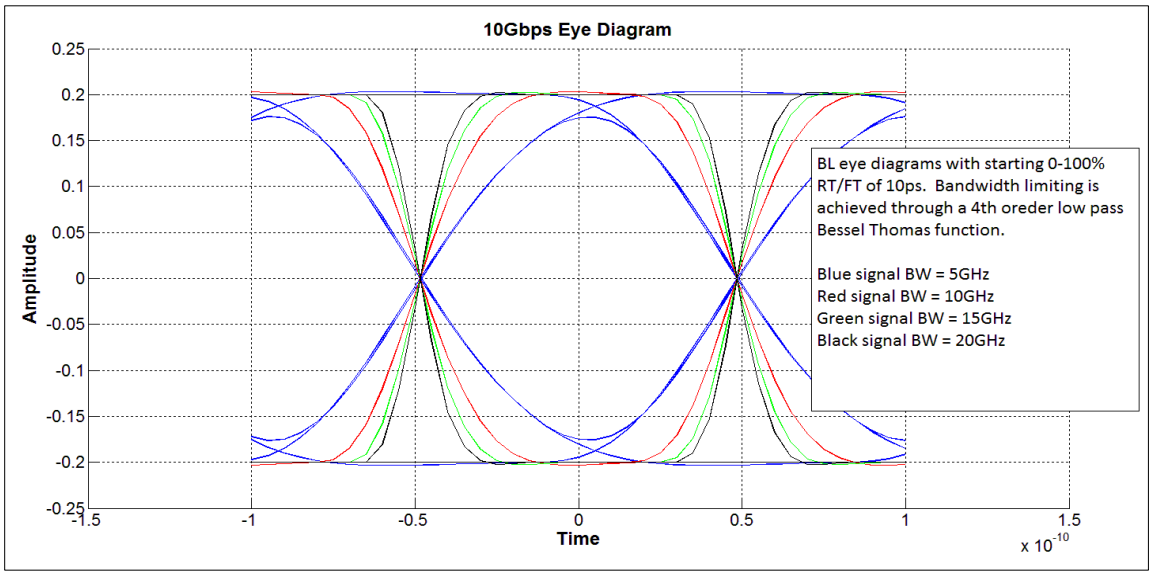


Figure 5, Eye diagrams with decreasing signal bandwidth.

at measurement point 0 with 0%-100% rise and fall times of 10ps filtered with 5GHz, 10GHz, 15GHz and 20GHz BW. The 5GHz blue plot above will clearly begin to effect link margin especially with open eye RX systems which have very little RX equalization capability. Based on lab measurement experience with 10Gb transmitters 15GHz seems to be a reasonable target to use for the measurement point 0 BW of signals in this study.

It is easy to see thru intuition that timing jitter added to the lower BW signal will have more effect on the vertical eye opening than the signals with more BW [2]. Using the

Matlab function from section 2.1.4 a 2ps_rms of 5GHz band limited normally distributed jitter is added to the same signal as in the signals in *figure6*. Since the component of jitter is known and only consists of random jitter (Rj) its ok to call any timing noise measured in *figure6 purely* Rj. Once deterministic channel effects are added this assumption no longer makes sense and becomes the jitter separation problem discussed in section 2.1.2.

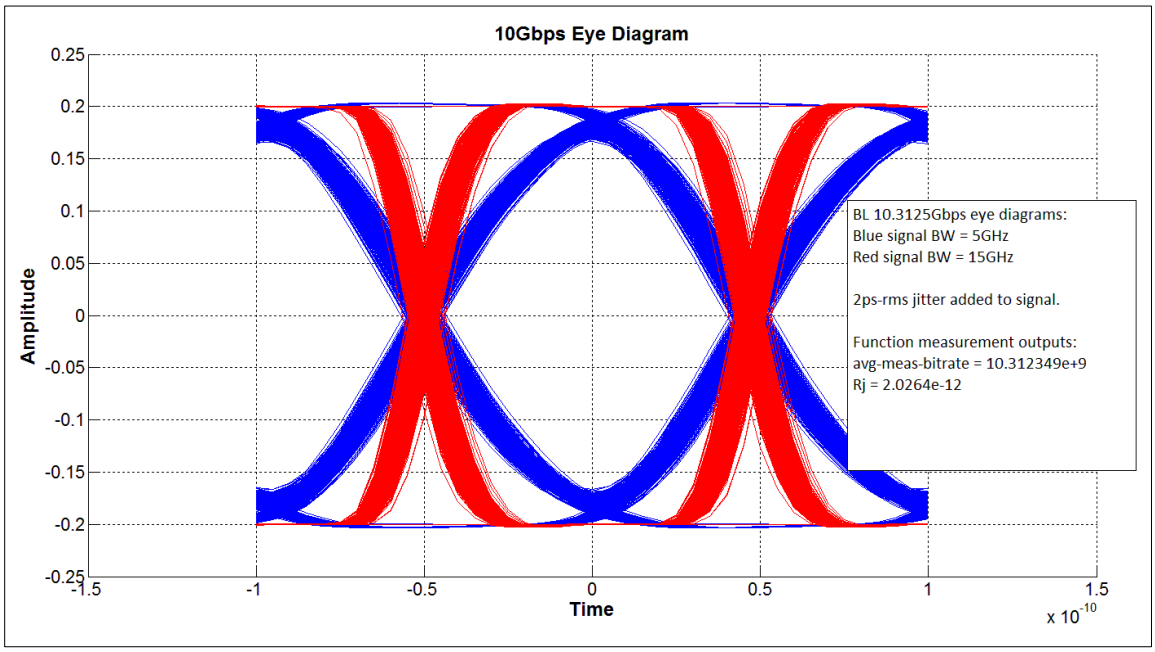


Figure 6, Eye diagrams with decreasing signal bandwidth and added random jitter.

It is clear that the jitter effects the vertical eye opening of the 5GHz BL signal at the sample point which can effect BER of the system. The vertical eye opening distribution with the 5GHz with both 0 jitter and 2ps of jitter are compared in *figure7*. The vertical opening shows ~8mV of vertical eye closure due to timing jitter when combined with low signal bandwidth. TX Rj and the truth of its randomness is important to know

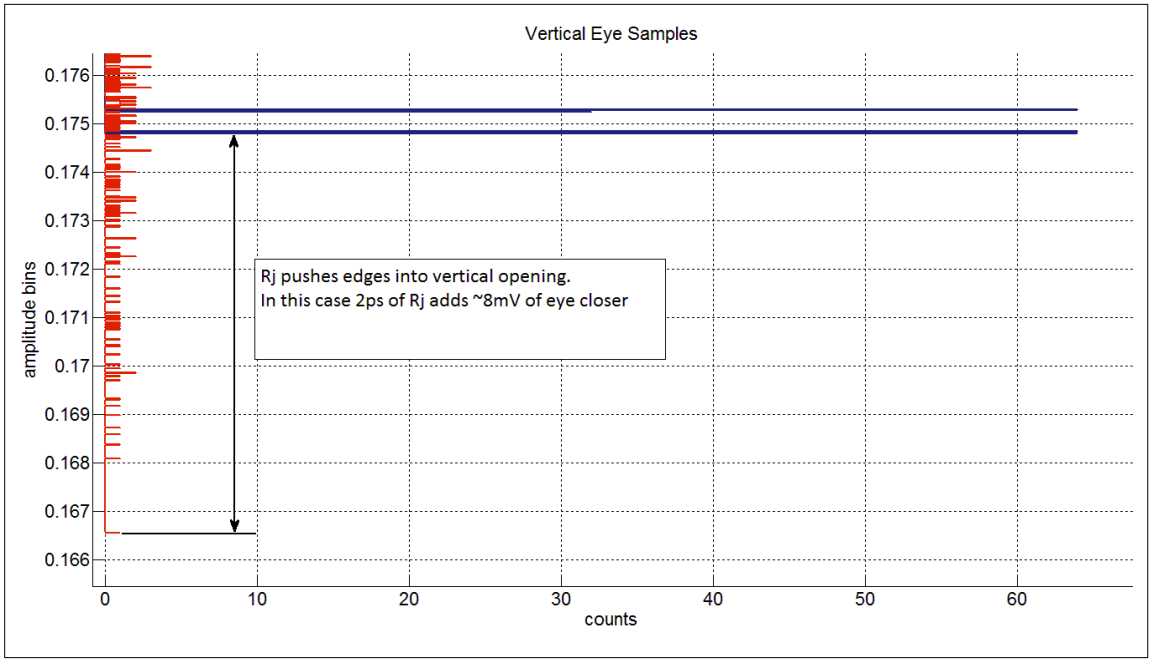


Figure 7, Vertical eye closure at sample point due to Rj and band limited signal.

because the PLL in the CDR (Clock and Data Recovery) circuit cannot track this type of timing noise thus it translates to the vertical margin at the sample point in the RX and this eats into BER margin. In the next section a 3tap transversal filter is calculated, synthesized and used to tune a signal after it is convolved with a host PCB channel.

2.2 TX Signal After PCB Channel

In the channel section it is demonstrated how the frequency dependent loss of a channel creates deterministic signal distortion or ISI which has to be compensated for before a 2 level detector can properly detect the data sent from a link partner. In this section a 3tap transversal filter is used to pre-distort the TX signal in order to tune out ISI creates by a typical host channel. This ability to tune the TX is important due to the fact that some TX specification measurements are defined after a section of host PCB

and some type of connector thus TX FIR filter tuning is required in most cases to meet specification conformance values.

2.2.1 3 Tap TX Filter

Compensation or equalization generally has a high pass filtering effect on the signal and can be very easily accomplished as a pre-distortion at the transmitter function. *Figure8* below shows a 1UI spaced 3-tap FIR TX filter which is regularly used in industry as a way

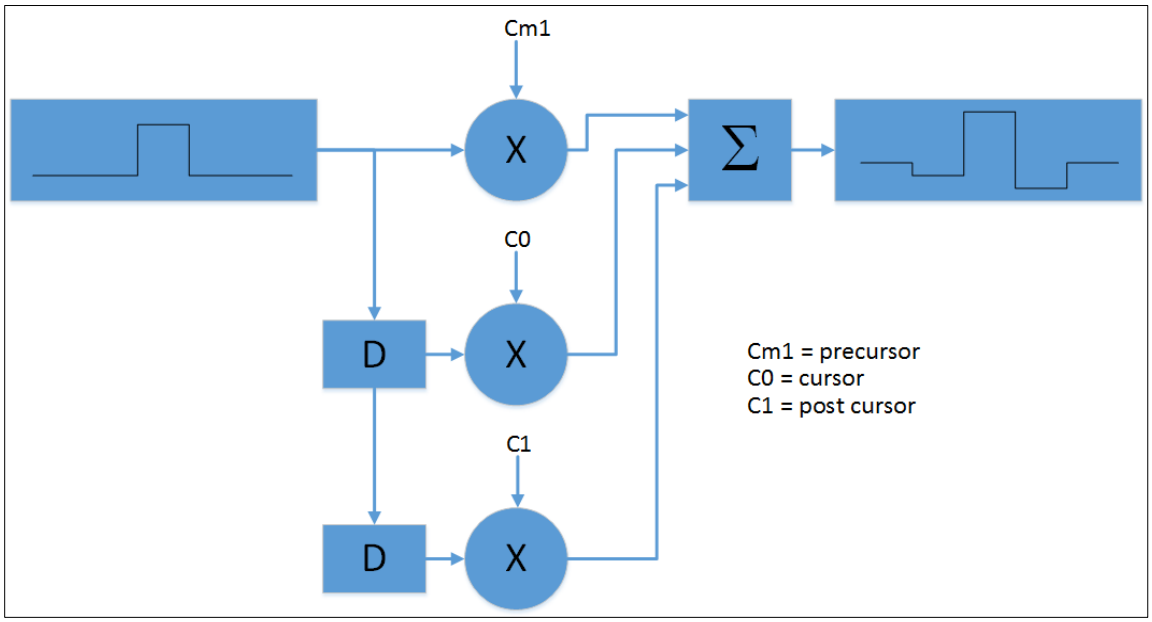


Figure 8, Bit spaced 3-tap FIR TX filter.

to assist in link equalization. This method of EQ is very attractive because the clock is generated at the TX thus the delay (D) can be accomplished using high speed D-type Flip Flops controlled by the bit clock in the digital domain. This means the TX EQ coefficients can be set high without risk of noise amplification. As a result, some specifications will allow for high levels of TX pre-distortion in order to relieve the noise amplifying RX EQ

needs. Before using this FIR filter it is first analyzed analytically and compared to the Matlab simulation. The Z Transform can easily be used to calculate the frequency domain response of the filter [5]. The frequency domain response can be convolved with the uncompensated channel to yield an effective channel showing the idea of pre-distortion at the TX assisting with the overall equalization of the channel [4].

Starting with the IO difference equation,

$$y[n] = Cm1 * x[n] + C0 * x[n-1] + C1 * x[n-2]$$

Taking Z transform and solving for transfer function,

$$Y(z) = Cm1 * X(z) + C0 * z^{-1} * X(z) + C1 * z^{-2} * X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = Cm1 + C0 * z^{-1} + C1 * z^{-2}$$

Substituting, $z = e^{2*\pi*j\frac{f}{fs}}$ where $fs = \frac{1}{10.3125e9}$

In the case of 10.3125Gbps signaling the D element in the filter is 1/10.3125Gbps = 97.96ps which is 1UI (Unit Interval). [5]

$$H(f) = Cm1 + C0 * e^{2*\pi*j\frac{f}{fs}^{-1}} + C1 * e^{2*\pi*j\frac{f}{fs}^{-2}}$$

$$H(f) = Cm1 + C0 * e^{-2*\pi*j\frac{f}{fs}} + C1 * e^{-4*\pi*j\frac{f}{fs}}$$

By inspection it is clear that this filter can only compensate to just over 5.15GHz since the sample rate Ts = 96.97ps thus the Nyquist Frequency, fn = 1/(2*Ts) = 5.1562GHz. Usually these filters are used to do high pass filtering meaning Cm1 and C1 are negative and C0 is positive. *Figure9* is a frequency domain plot of the equations above. If the goal of the TX filter is to stress a receiver with added channel an n-tap TX device can be

used with positive coefficients to create a low pass filter in order to add channel and margin the BER of a receiver under test. *Figure9* shows a low pass response plotted with the high pass response.

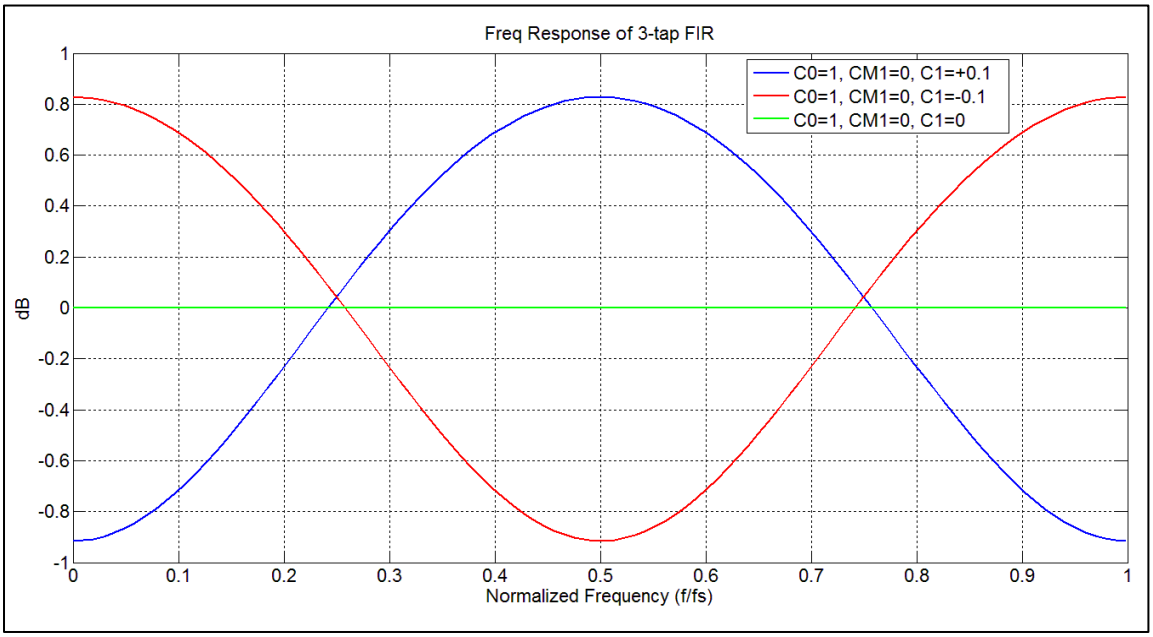


Figure 9, Frequency domain representation of 3-tap filter with positive and negative tap coefficients.

2.2.2 TX Signal FIR Matlab

The Matlab function created in the previous section will also create a 1UI spaced filtered TX signal. The signal is created by adding (LP) or subtracting (HP) signals created from rotated versions of the original sequence to the signal created from the original sequence. The same clock is used in all signals, so all bits from the added or subtracted signals are all on the same time base. *Figure10* is an example of a high pass filtered signal. The frequency response of the time domain signal in *figure10* is calculated by the

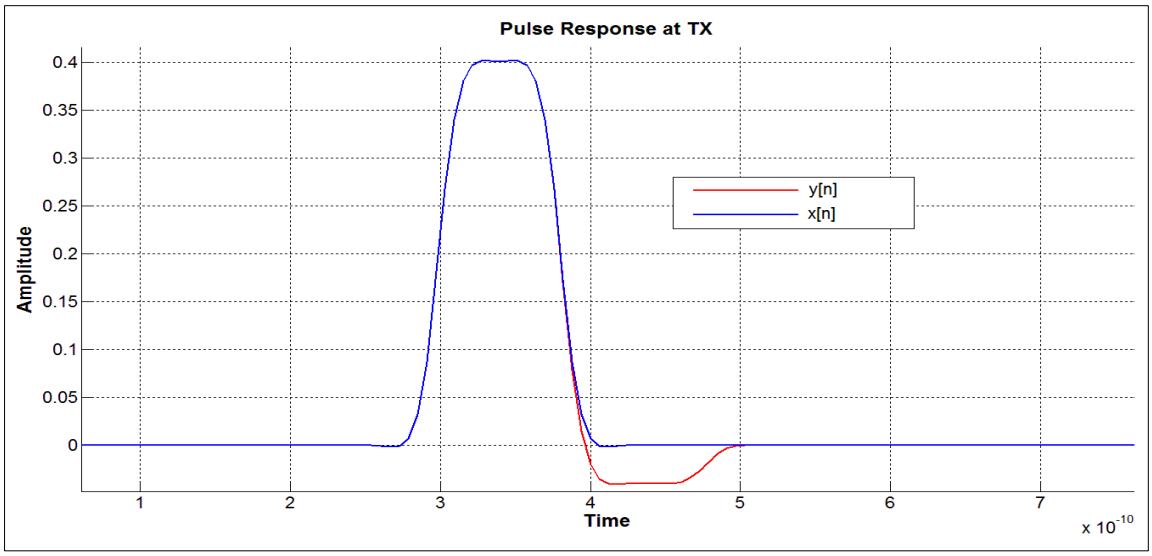


Figure 11, Single bit pulse response with flat filter (red trace) and slight post cursor high pass filter (red trace).

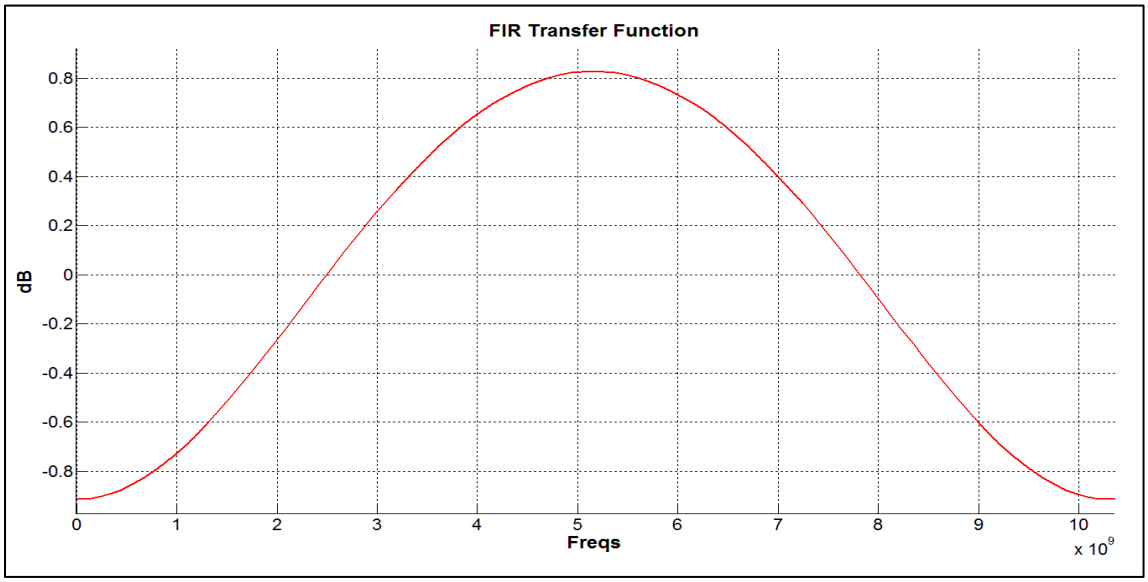


Figure 10, 3-tap FIR filter transfer function determined by input and output pulses.

following equation and is plotted in *figure11*. Note that the high pass filter plot below matches the one from section 2.2.1.

$$H(f) = \text{fft}(y[n])/\text{fft}(x[n])$$

2.2.3 TX Signal Synthesized, Stressed and Tuned

In this study it is assumed that the TX signal could be specified after a host PCB (Printed Circuit Board) channel and a high frequency surface mount pluggable connector as the interface between server PCB and copper DA cable. The circuit below (*figure12)* is a

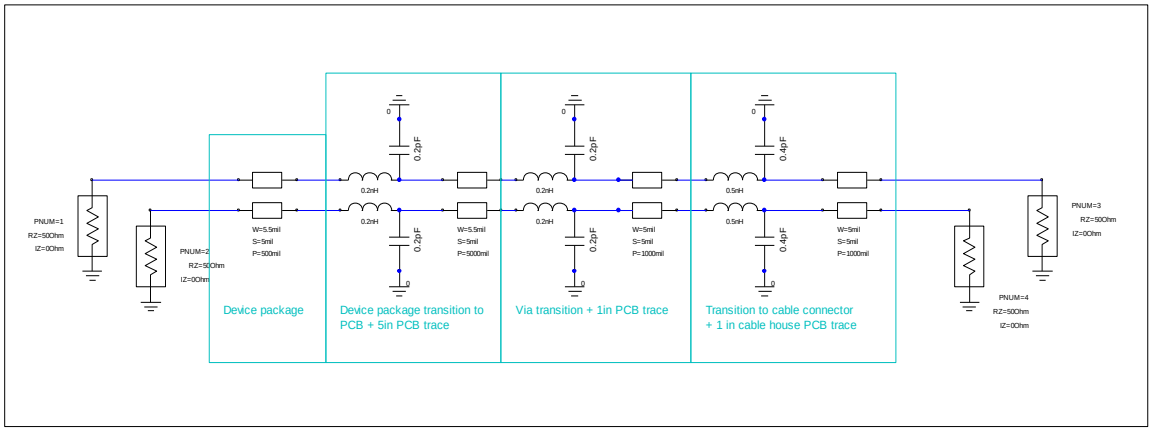


Figure 12, PCB channel model up to measurement point 1.

typical model used to describe this electrical interface. The frequency dependent loss through the channel (ILoss) and the reflected signal (RLoss) are the primary deterministic responses used to determine the deterministic signal at the TX compliance point. This channel's frequency response is shown in *figure13* and the s-parameters are used to create the TX signal. The channel is simulated to 40GHz to allow simulation of higher bit rates with the same s-parameter file.

The eye diagram in *figure14* is generated using the same signal generation function as the previous section except this time more stressors are added to make a more realistic signal. Each stressor can be added ideally and independently, so the resulting signal

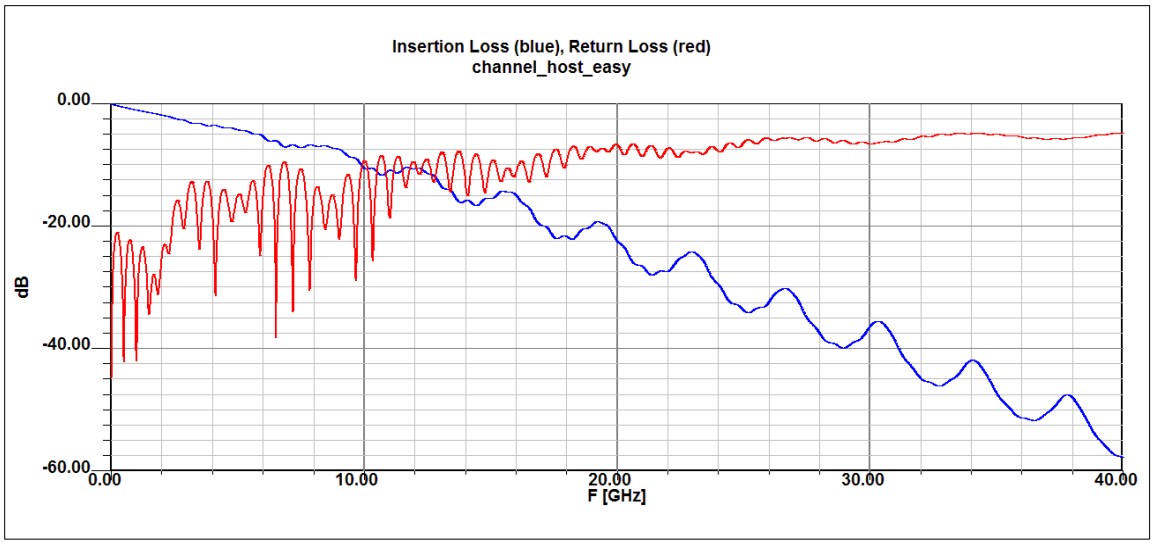


Figure 14, Simulated response of channel model in figure12.

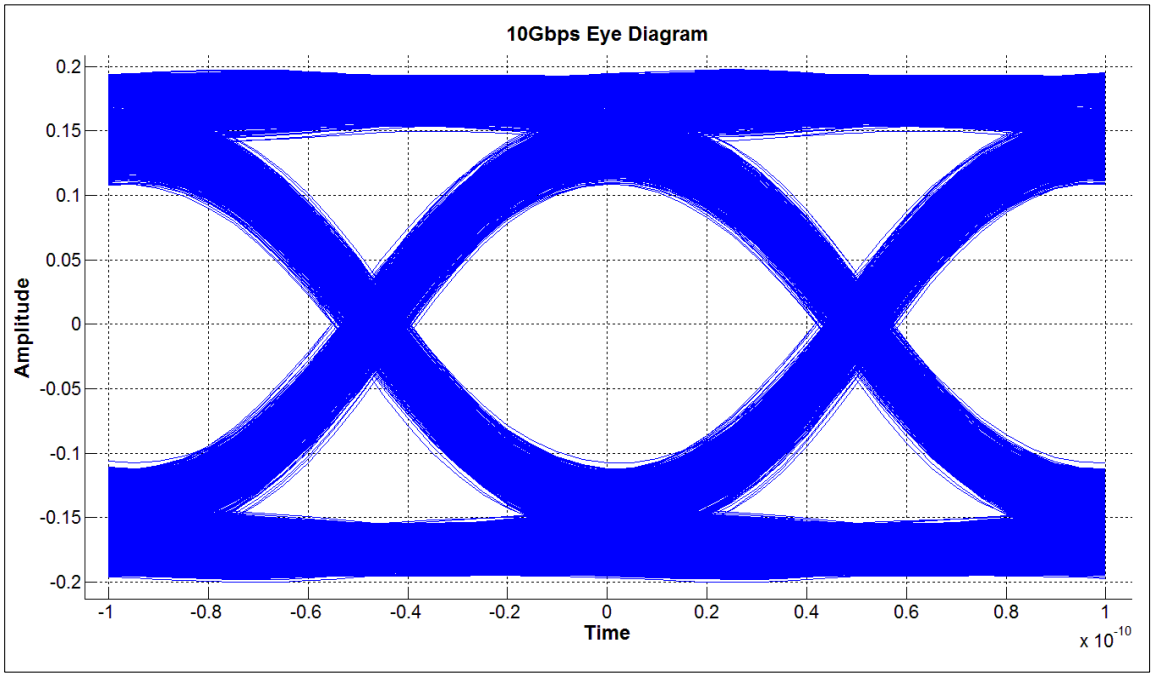


Figure 13, Eye diagram of un-tuned and noisy TX signal.

bound for the RX function is fully described. This eye diagram signal in *figure14* has the following input properties and measured values (see table1).

Signal input values	**Measured signal values**
Rise time fall time = 30e-12	DDJ = 15.1 ps
cm1 = 0.0, c0 = 1, cp1 = 0.0	Rj = 1.82 ps rms
Random jitter = 2e-12 rms in 5GHz BW	
Signal BW=15e9	
TX random amplitude noise=10e-3 rms	

Table 1, Un-tuned TX signal characteristics

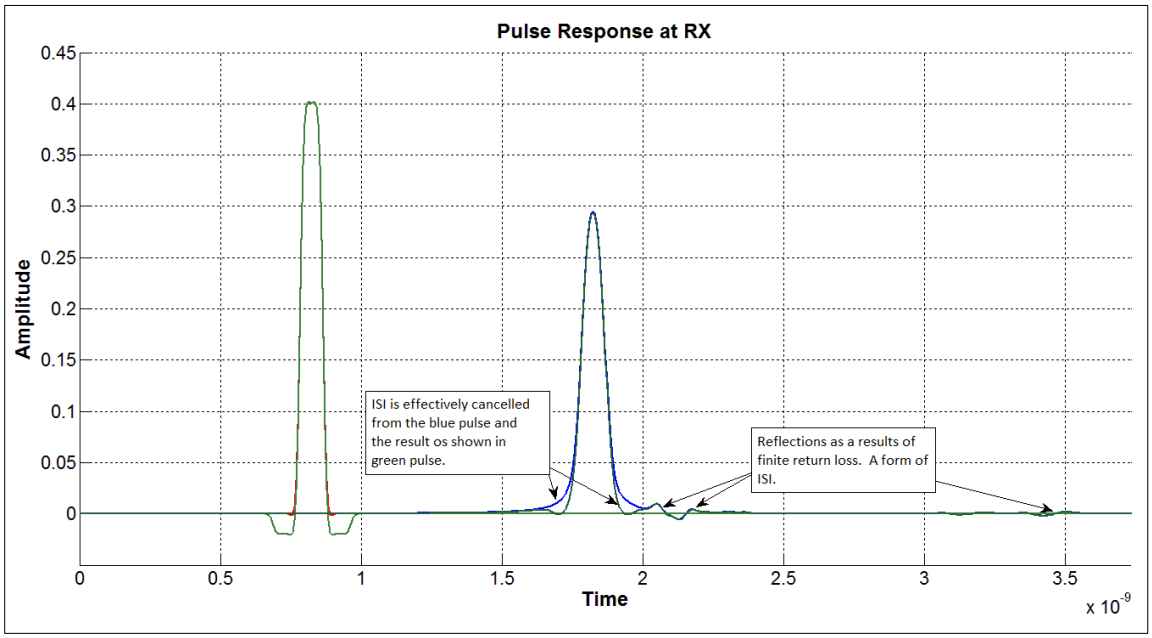


Figure 15, Single bit pulse response illustrating visual tuning capability of 3-tap TX filter.

A pulse response with the same signal generator properties shows Cm1 and C1 high pass filtering could help the eye opening. *Figure15* shows the pulse with C1=0, Cm1=-

0.05 and C1=-0.05 and a pulse with a flat filter. This shows how useful the TX 3-tap FIR filter can be for TX conformance tuning. The tap values could be calculated, but knowing the visual trends from the previous calculations is usually enough to converge to a reasonable answer. Better methods to adapt the EQ for the channel will be discussed in chapter 4. Since the eye diagram is the easiest way to see the overall signal quality the analysis is re-run with the tuned FIR tap values and shown in *figure16*. Approximately 50mV of eye opening is gained from this simple and visual pulse response tuning (see table2) below.

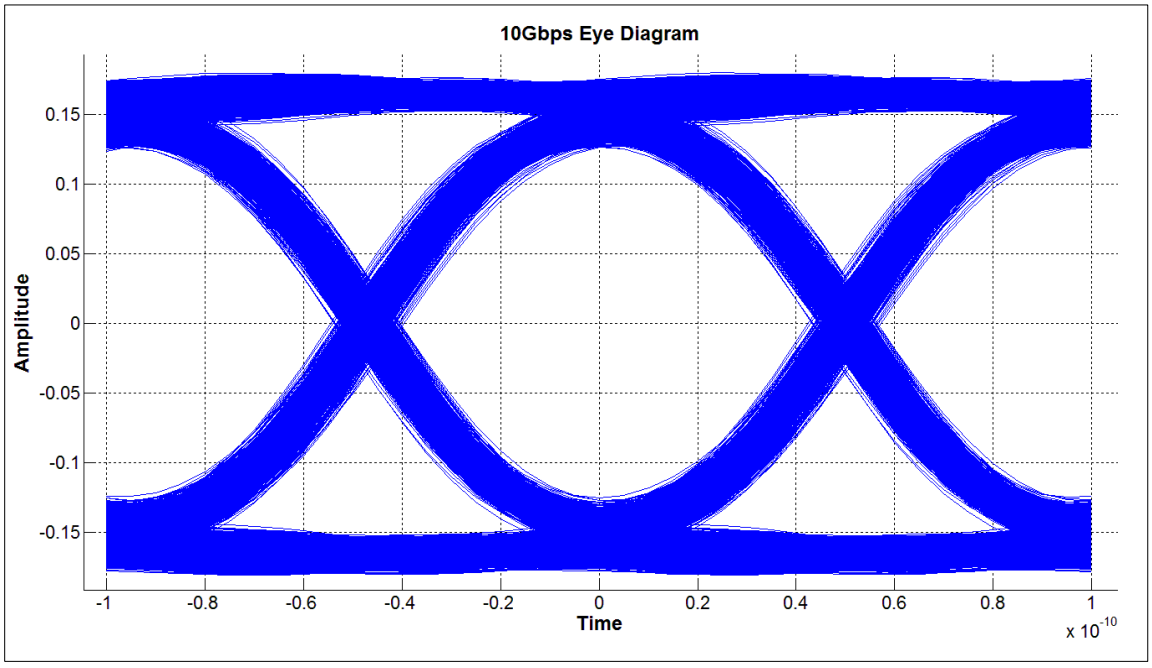


Figure 16, Eye-diagram of TX signal tuned with 3-tap TX FIR filter.

Signal input values	**Measured signal values**
Rise time fall time = 30e-12	DDJ = 10.3 ps
cm1 = -0.05, c0 = 1, cp1 = -0.05	Rj = 1.91 ps rms
Random jitter = 2e-12 rms in 5GHz BW	Vertical eye opening improvement ~ 50mV
Signal BW=15e9	
TX random amplitude noise=10e-3 rms	

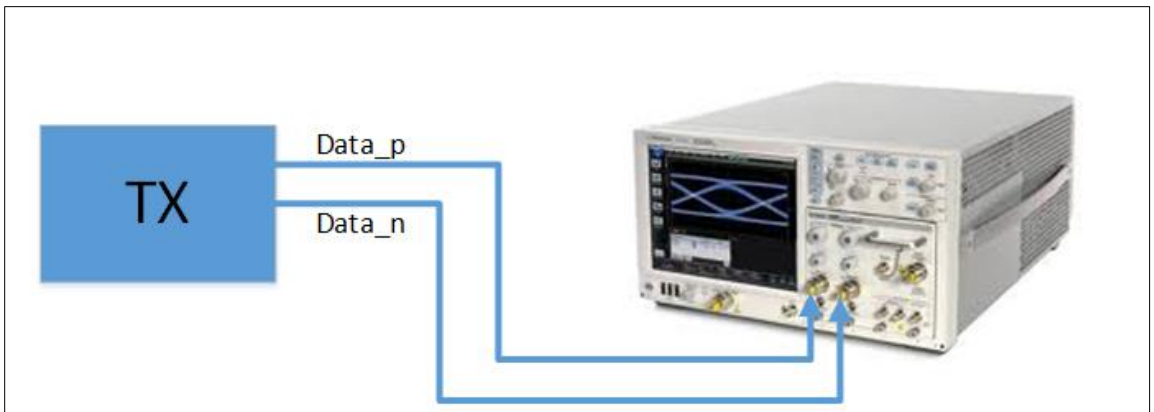
Table 2, Tuned TX signal characteristics

2.2.4 TX Filters Tuned By Far End RX

What a perfect concept in theory. The RX gets to tell the TX how to set its 3tap filter. This concept is used in many specifications including 10GBaseKR, 40GBaseCR4, 10GBaseT and the newer 40G and 100G specifications. The great part of this concept is that the RX EQ at least in the 10G case can be purely post cursor cancelling and can be implemented with a non-noise enhancing Decision Feedback Equalizer (DFE). Since the RX can tell the TX FFE to pre-distort and equalize 1 tap of pre and post cursor and in addition increase and decrease c0 amplitude the RX can be simplified and reduce BER reducing noise enchantments. In reality the complexity of the training phase can sometimes cause more problems with vendors interoperating than would occur if phy designers planned to make RX paths more robust to a properly specified TX signal.

2.2.5 TX Signal Measured Eye

n the previous section transmitter concepts were analyzed and simulated. The overall goal of this study is to analyze and simulate TX, channel and RX in order to create a lab measureable specification for each. The basic TX lab measurement is made with either a sampling or real time oscilloscope. Each scope has it strong points, but for this signal analysis the sampling scope is selected for its 10bit vertical resolution and its very accurate and stable time base. Most 20GHz and above real time scopes have 8bit vertical resolution which is not enough to properly measure some specification signal properties. See figures 17 and 18.



Agilent DCAx Sampling Oscilloscope.

- Scope recovers clock from data using configurable CDR.
- CDR is specification TX reference RX for many industry standards.
- Sampling scope great for signal integrity measurements and conformance testing, but does not do well for general HSS debug.
- Can measure: EYE diagrams, waveforms, timing and amplitude noises with great accuracy relative to large amplitude HSS signals.

Figure 17, Sampling Oscilloscope TX measurement setup.

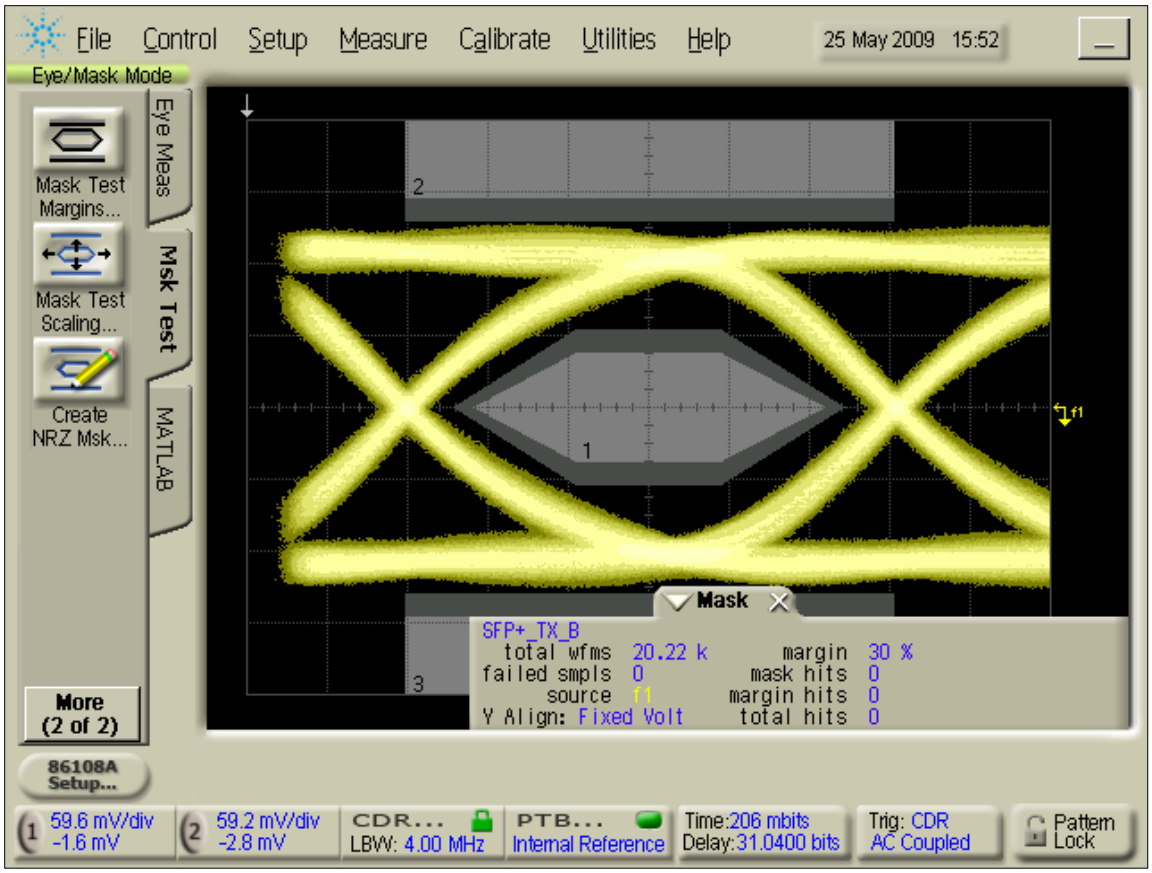


Figure 18, Sampling Oscilloscope measured eye diagram example.

3 CHANNEL

In this section a synthesized signal is propagated through a low, medium and high loss direct attach (twin coaxial cable) channel. The goal is to properly use measured and simulated channel s-parameters in order to predict signal stresses that the RX will have to equalize in order to make a correct bit decision. Other noises which play a role in higher BER such as Unbounded (TX thermal transistor noise), bounded (adjacent channel cross talk) random noises and TX timing based noise will be considered in chapter 4.

3.1 S-PARAMETERS

3.1.1 Brief Description

S-parameter analysis is a great way to capture the response of a passive linear communications channel. For the passive linear channel the measurement using a Vector Network Analyzer (VNA) is fairly straight forward and the resulting s-parameter data file is portable across many simulation tools. Channels with active components such as the active CTLE can be measured with the VNA, but care must be taken to understand power calibration and linear range of the CTLE.

The VNA is a complicated device that measures voltage waves incident on and reflected from a port of a device under test (DUT). *Figure19* shows the incident voltage waves on a port are defined as a_x and the voltage waves flowing away from a port are defined as b_x. The port is defined by the subscript x and the number of potential ports is

Figure 19, Incident and egressing voltage waves.

unlimited. Both a_x and b_x waves are defined as voltage wave normalized by the square root of the waves source impedance, Zo.

$$a_x = \frac{V^+}{\sqrt{Z_o}} \qquad\qquad b_x = \frac{V^-}{\sqrt{Z_o}}$$

The VNA systematically drives a swept sinusoid from a Zo=50ohms port into each DUT port while terminating and listening to all other ports with Zo=50ohms. The driving port also measures the reflected wave reveling the b wave from the same port that is driven. The individual dependent (b_x) variables are built as follows,

$$b_1 = S_{11} * a_1 + (S_{12} * a_2)|_{a_2=0} + (S_{13} * a_3)|_{a_3=0} + (S_{14} * a_4)|_{a_4=0}$$

If $a_2 = a_3 = a_4 = 0$ then b_1 can be determined by driving a_1 and listening to the reflected signal. This yields the coefficient, $S_{11} = b_1 \,/\, a_1$. S_{21} , S_{31} and S_{41} can be determined in the same measurement.

$$b_2 = S_{21} * a_1 + (S_{22} * a_2)|_{a_2=0} + (S_{23} * a_3)|_{a_3=0} + (S_{24} * a_4)|_{a_4=0}$$

$$b_3 = S_{31} * a_1 + (S_{32} * a_2)|_{a_2=0} + (S_{33} * a_3)|_{a_3=0} + (S_{34} * a_4)|_{a_4=0}$$

$$b_2 = S_{41} * a_1 + (S_{42} * a_2)|_{a_2=0} + (S_{43} * a_3)|_{a_3=0} + (S_{44} * a_4)|_{a_4=0}$$

Thus knowing $a_1$and holding all the other independent variables ($a_2 = a_3 = a_4 = 0$) zero gives,

$$S_{21} = b_2 \,/\, a_1\,,\ S_{31} = b_3 \,/\, a_1 \text{ and } S_{41} = b_4 \,/\, a_1$$

After the VNA is finished driving each port and listening to all others a matrix, [S], of coefficients is built from the a_x and b_x values [6]. The 4x4 matrix below is an example of the s-parameter result.

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} S_{11}\ S_{12}\ S_{13}\ S_{14} \\ S_{21}\ S_{22}\ S_{23}\ S_{24} \\ S_{31}\ S_{32}\ S_{33}\ S_{34} \\ S_{41}\ S_{42}\ S_{43}\ S_{44} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix}$$

Since the magnitude and phase is known for all ports to all ports it is a full linear description of the system. Under the assumption the channel is linear and the VNA has enough measurement ports the engineer can know the complete response of the signal path and all cross talk paths.

The s-parameter description above only deals with the single ended responses of the channel however most digital communications circuits make use of balanced signals thus the mixed mode version of the s-parameter values must be calculated. Using the single ended coefficients above the channel response as seen by a differential signal (sddxy) can be calculated by the following equation assuming the following port diagram in *figure20*.

$$Sdd21 = 0.5 * (S31 + S42 - S41 - S32)$$

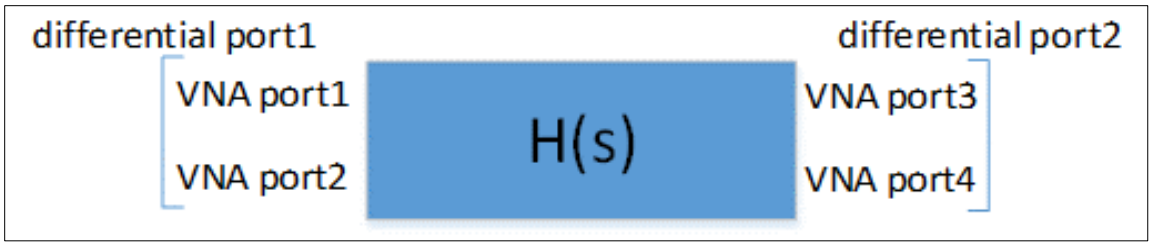


Figure 20, Differential s-parameter port setup.

Sdd21 is used as the differential signal thru response thought out this study. Mode conversions and common mode responses can also be calculated by knowing the single ended response, but are not currently part of this study.

Before an accurate measurement is achieved there are many details to consider while setting up the VNA. A few of the details are Intermediate Frequency Band Width (IFBW), number of frequencies, port power levels and proper calibration so the 50+0j ohm reference plane is correctly placed. Once a passive structure is measured or modeled the resulting s-parameters should be tested for passivity and causality however reasonable pulse or impulse response can usually reveal a good measurement/simulation or a problem [8].

The next section details a measurement of a real direct attach cable and a reasonable simulation. Both yield s-parameters as the channel response.

3.1.2 Channel Definition and Response

This section describes the channel in detail along with a channels measured and another channels simulated results. Perfectly matching simulated results with measured results is not in the scope of this study. The simulated results are simply used as a way to look

at trends and quickly understand what channel variations do to the BER of an adaptively equalized link.

The channel specification is designed from package pad to package pad. *Figure21* shows the complete signal path from TX to RX in both directions. Note that *figure21* shows a network connection between a network server and a switch.

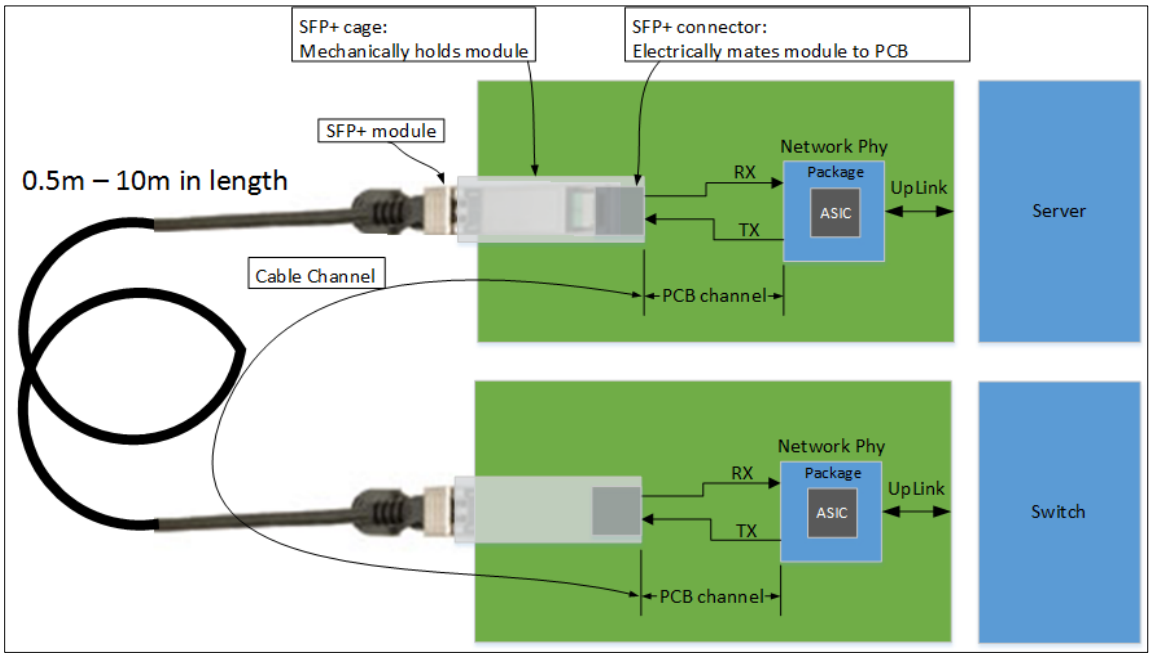


Figure 21, Physical channel diagram including package, PCB and cable.

The measurement previously mentioned is of a 7 meter Direct Attach Cable measured with a VNA set to sweep from 10MHz-20GHz in 10MHz steps and the Intermediate Frequency BW (IFBW) set to 300Hz. The IFBW set to 300Hz creates a very narrow band measurement lowering the instrument noise floor which results in a more accurate cross talk measurement. This particular cable uses two twin-axe cables for the signal paths thus 8 VNA ports can be used to fully describe the channel including insertion loss,

cross talk and all mode conversions. Figures 22 and 23 show an Octave (free Matlab) based analysis of the measured response described above with respect to the cables compliance to the SFP+ (SFF-8431) specification. *Figure22* shows the Sdd21 insertion

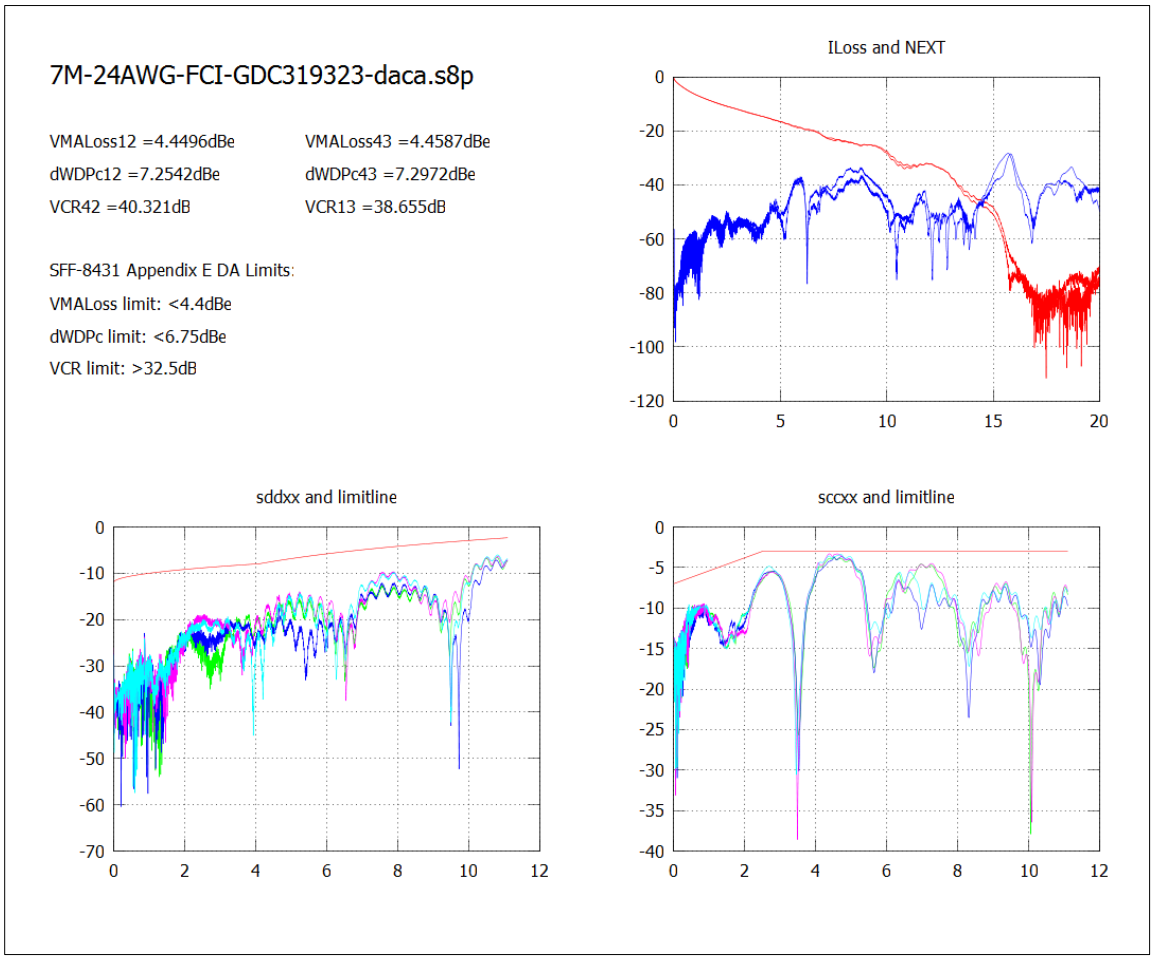


Figure 22, Measured s-parameter response of 7 meter twin-axial communications cable.

loss and the Near End X-Talk (upper right plot). The NEXT frequency domain measurement is used to quantify how much an aggressor signal (with certain amplitude and BW) effects victim RX signal. If the VNA IFBW were set too high (<2kHz) then the measured NEXT would be artificially high due to instrument noise floor thus potentially

leading to a false failure. This example shows that careful setup of the VNA is a must in order to get an accurate analysis. The pulse response in *figure23* is included to show an engineer looking at the data that the frequency to time domain conversion and the

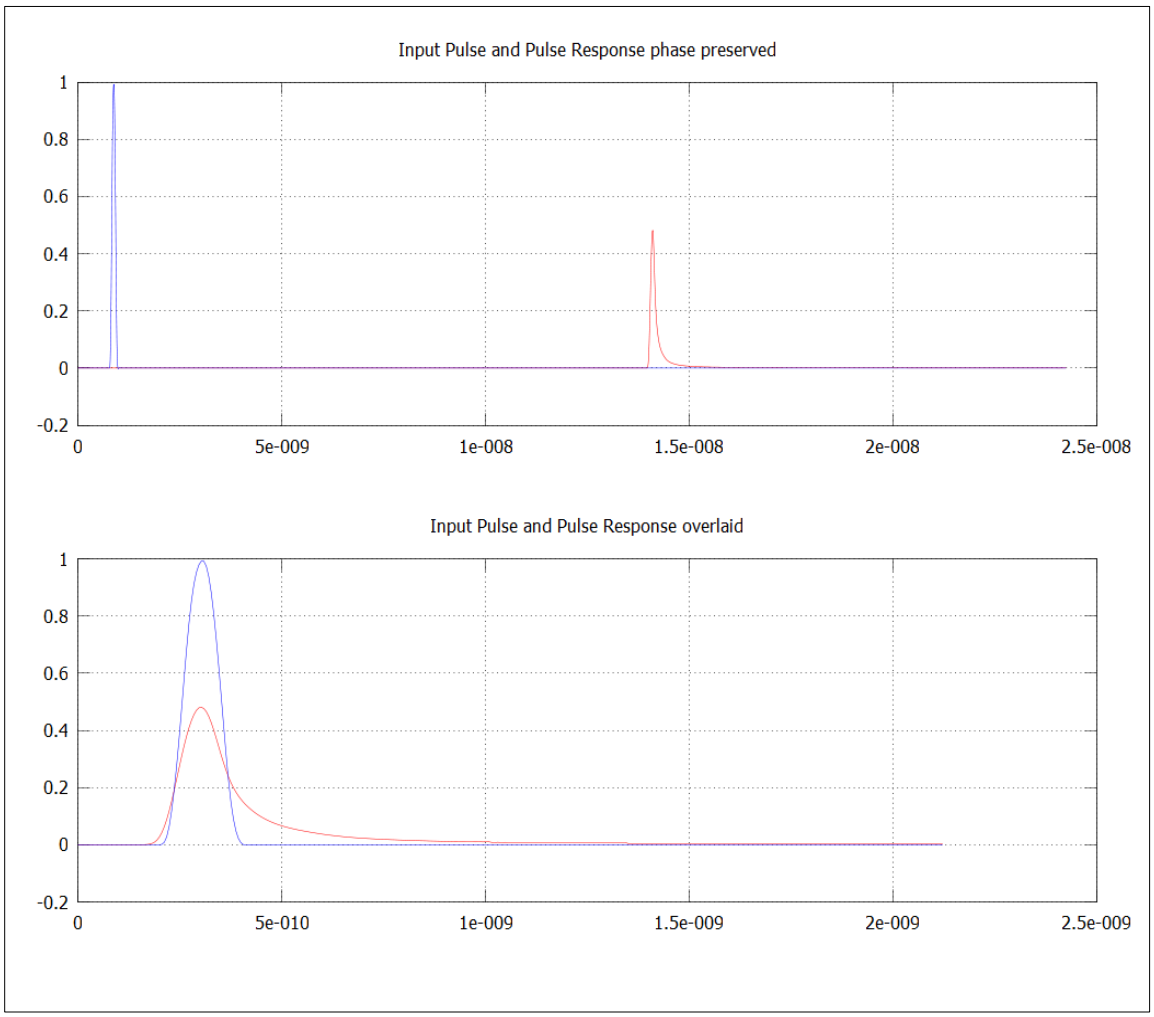


Figure 23, Single bit pulse response channel phase preserved and pulses overlaid.

measurement itself is reasonable.

3.2 TIME DOMAIN TRANSFORM VERIFICATION

The channel measured data is very interesting and when properly applied is very useful for design, but it is very easy to make mistakes and improperly transform this frequency

domain data into time domain signal data. For now, the measured channel parameters will be put aside and a very simple calculated channel model will be used to verify signal convolution and Matlab functions. The circuit in *figure24* is setup to act as a network analyzer for measuring a single direction of a 2-port insertion loss and return loss which

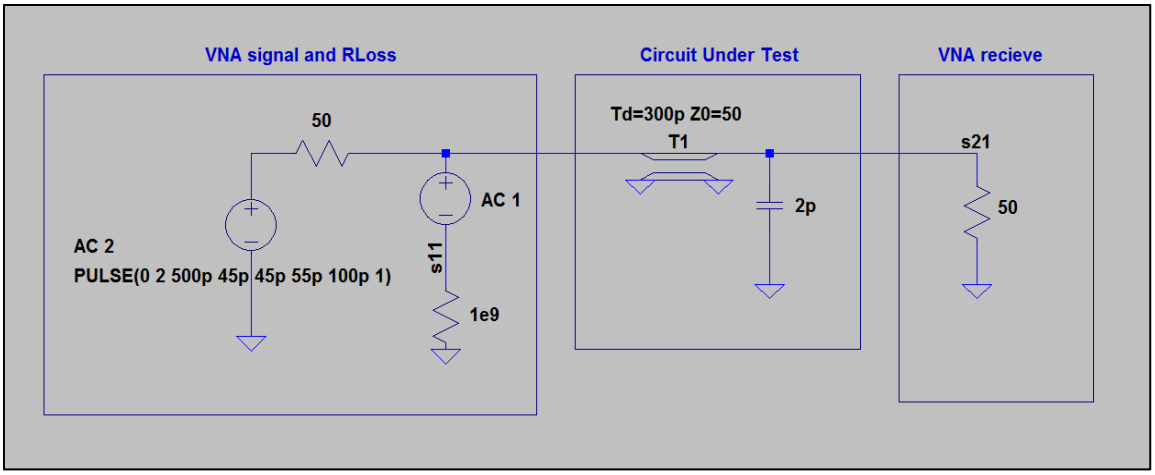


Figure 24, Simple RC and delay model used to compare symbolic calculation with s-parameter time domain conversions.

in s-parameters terms is s21 and s11 respectively. The model can very easily be reversed to find s12 and s22. Later mixed mode combinations of 50ohm single ended s-parameters will be calculated from n-port measurements in order to see the channel response to any number of differential signals, but for now proof will continue with the simple 2- port model.

The step response transfer function of the 50ohm s21 parameter for the low pass filter plus delay is very simple and is of the form,

$$H\_unit\_step(s) = \frac{N}{s+a} e^{-d*s} \frac{1}{s} \qquad \text{Where } a = \frac{2}{Zo*C} \text{ and } N = \frac{2}{Zo*C}$$

Taking the inverse Laplace Transform gives system step response and differentiating gives h(t),

$$h\_unit(t) = \frac{d}{dt}(N * e^{-a*t}u(t-d))$$

This function, h_unit(t), can now be used with arbitrary inputs.

The plots, *figures25, 26 and 27,* are made with C=2pF, Zo=50ohms and d=300ps with H(s) evaluated over the same frequencies present in a synthesized input signal, x. In this case the input signal, x, has 240 samples with Ts=6.0606ps, frequency range Fstart=343.75MHz, Fnyq=82.50GHz and Fs=165.GHz. After the system impulse response *(figure26)* is calculated a 10.3125Gbps (96.97ps wide) single bit pulse signal is created and propagated through the system by linearly convolving the time domain

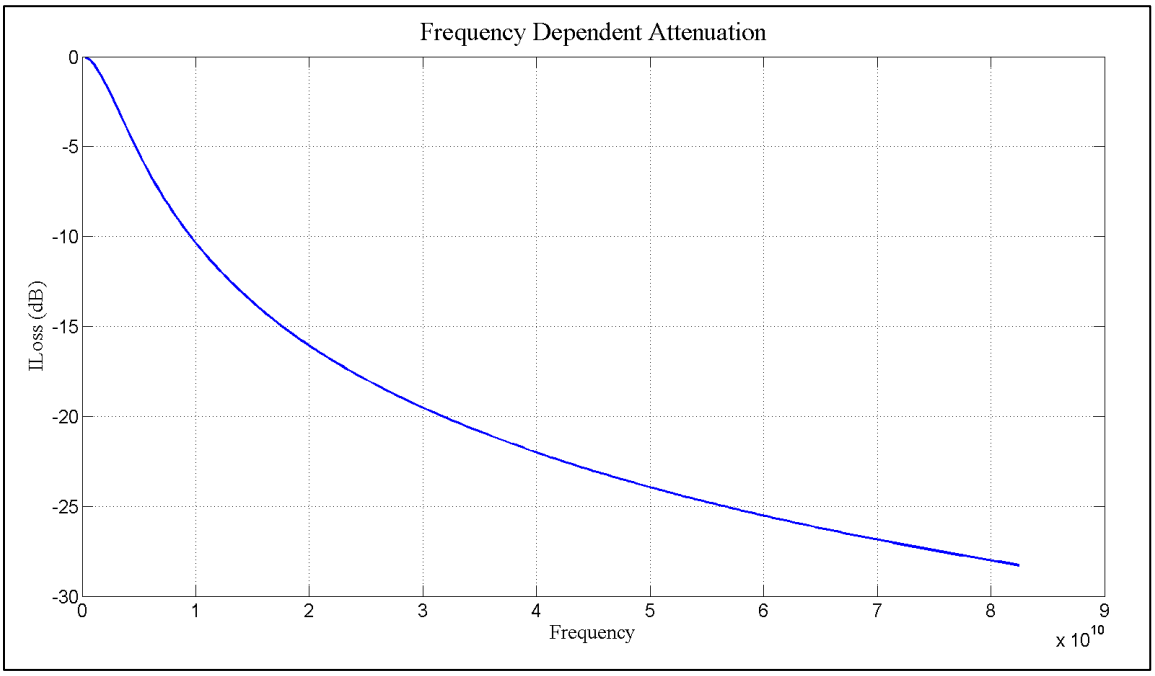


Figure 25, Frequency domain response of 2pF shunt capacitor in Zo=50ohms (figure24).

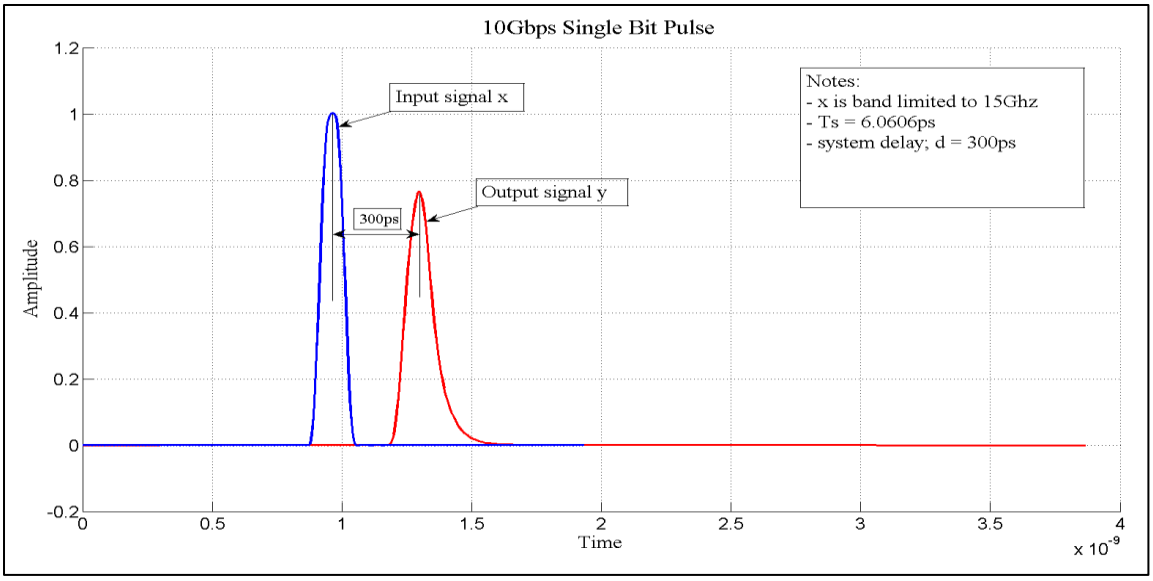


Figure 27, Input pulse (blue) and output pulse (red) convolved with response from circuit in figure24.

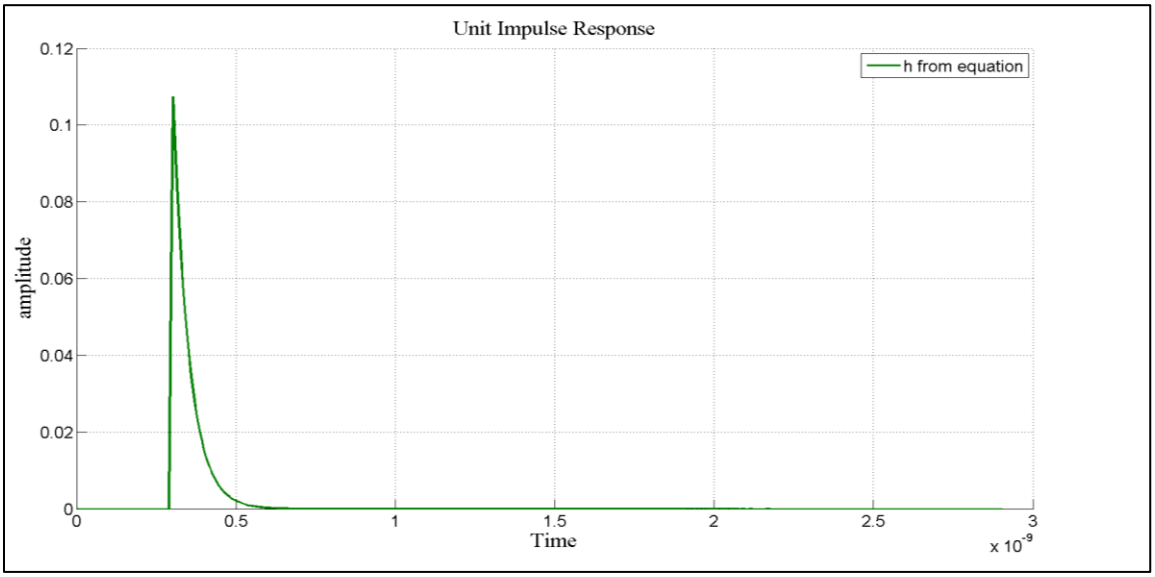


Figure 26, Impulse response of 2pF shunt capacitor in Zo=50ohms.

input signal with the normalized impulse response. This creates the output signal y[n]=conv(x[n],h[n]). At this point the TX source and RX load impendences are assumed

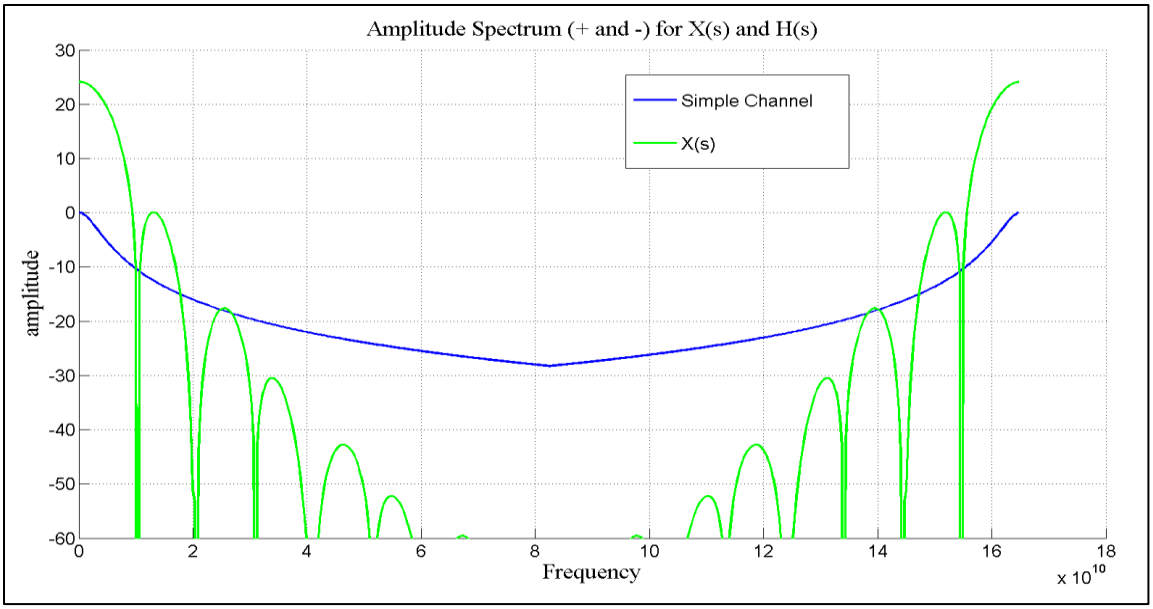


Figure 29, Amplitude spectrum of input signal plotted with simple channel response.

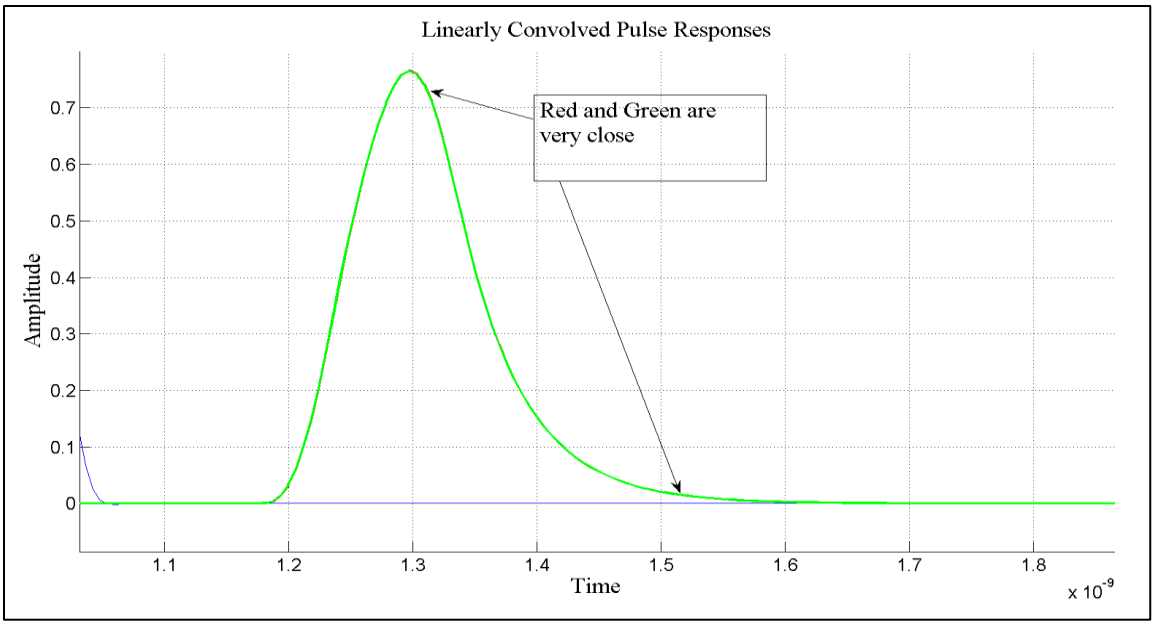


Figure 28, Comparison of pulse calculated from equations and h(t) from S21.

to be matched to the normalizing impedance of the network analyzer. The pulse responses in *figure27* were calculated analytically by knowing the transfer function and

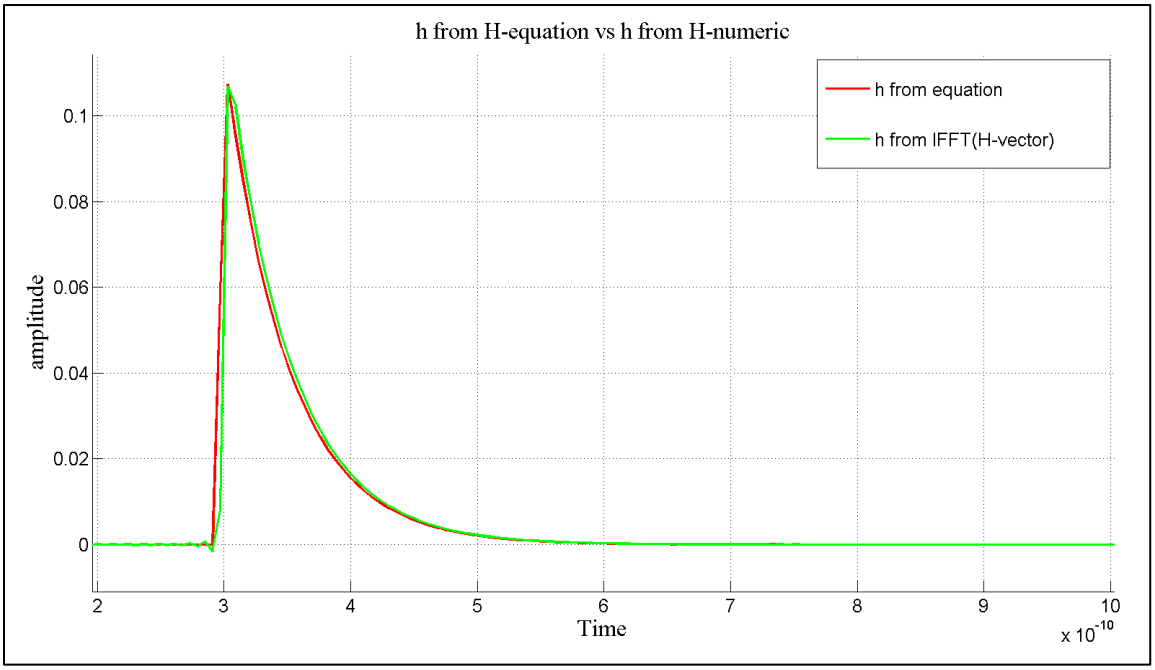


Figure 30, Comparison of h(t) calculated from equations and h(t) from S21.

applying basic signals and systems techniques.

Next a method is created where the pulse response signal can be attained by only knowing the H(s) in complex numerical form which is the format of the VNA measured results. A portion of the numeric H(s) vector is listed below as an example.

H(s) = 0.4189 - 0.8938i, 0.6029 - 0.7359i, -0.8764 + 0.2020i, -0.2074 + 0.8132i, 0.5724….

The impulse response from the numerical H(s) data can be calculated with the IFFT. Since it is assumed the VNA only measures to the Nyquist frequency the negative frequencies are created by taking the complex conjugate of the mirror image of the positive frequencies which are then appended to the end of numeric transfer function H(s). This is needed to create a proper time domain impulse and *figure28* shows this along with the spectrum of the signal x. Next the IFFT will be used to find the impulse

response, h_from_sparam, from the numeric vector H_vector_full [4]. This result should be the same as the impulse calculated with the inverse Laplace Transform in *figure26*. *Figure30* shows the linear convolution of the pulse x with the equation based impulse and the data based impulse responses.

Now that a proven function has been created to stress a TX signal with numeric channel data it will be used in the next chapter to stress a receiver functions capability.

4 RECEIVER (RX)

4.1 RECEIVER (RX) INTRODUCTION

The receiver is the most complex part of the serial link. It has to make a proper decision about the transmitted data in a noisy and distorted environment without the benefit of having a TX reference clock at the RX or details about exact channel response. In a 2 level system if a channel creates so much ISI that a 1 bit's peak amplitude falls below the decision threshold the system is referred to as a closed eye system. In this case the RX

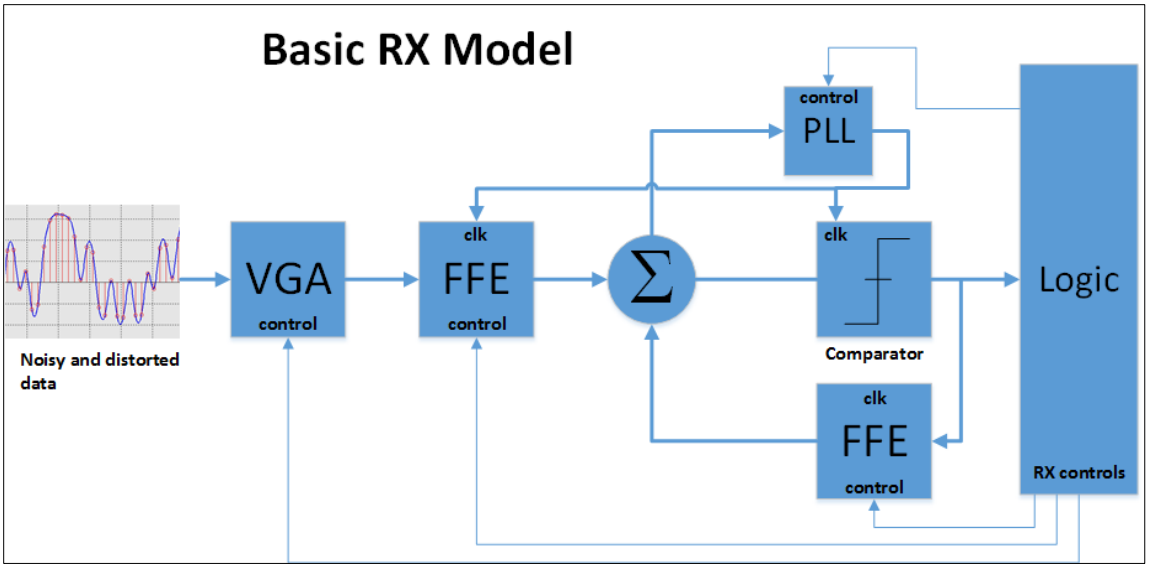


Figure 31, High level RX block diagram.

has to do something to electrically correct the signal before the RX decision circuit detects the bit. The modern serial receiver generally has to recover a clock and adaptively equalize the signal. This is no easy task since in some cases the recovered clock is needed for the equalizer and the equalizer is needed for the clock recovery. The elements of the receiver can vary widely, but a simple model will sometimes include a

FFE, VGA, summing node, PLL, comparator and DFE. Continuous Time Linear Equalizers are also sometimes used in the RX path. The block diagram in *figure31* is a generic description of a typical PAM2 receiver. *Figure31* also shows the distorted post channel signal and that it has no clear level to set as a single threshold for a data bit decision. The first circuit element in the RX path can be a FIR filter or specifically a Feed Forward Equalizer (FFE) which consists of clocked analog sample and hold circuits with a scaling factor summed at a common point in time. Very complex, but for this model the in line FFE is the only block that can cancel pre-cursor ISI. The FFE can also cancel post-cursor, but since the down side of the FFE analog block is noise amplification, post-cursor EQ is usually left to the Decision Feed Equalizer DFE. The DFE is a non-linear equalizer that feeds back the scaled and delayed decision to the summing node just before the decision comparator. The DFE block is sometimes referred to as a canceler due to its ability to equalize without enhancing noise. Next in the path is the Variable Gain Amplifier (VGA). The VGA is used to boost the signal away from the decision threshold. BW and noise amplification are obviously problems that can occur with this block. The next block is the summing node and its function is to sum all the equalization effects at 1 point in time in order to allow the comparator to sample the analog signal and ultimately turn it into a digital signal. The PLL block is used to recover a clock from the data and act as the sample clock for all blocks that are sampled. The FFE in the feedback path around the comparator acts as a Decision Feed Equalizer (DFE) and is “by design” the non-linear part of the RX model. The DFE can only cancel post-cursor ISI, but can do

this without amplifying the noise present on the input signal. The down side of the DFE is that the bit decision has to be correct for the equalization or cancellation to be optimum. In the extreme errors can propagate around the DFE loop thus causing receiver adaptation algorithm to diverge. This shows the IIR nature of the DFE filter [1]. The logic section of the RX model stores data bits and calculates an error signal used to update all the blocks of the RX.

In this study only gain and FFE will be used to compensate the signal prior to the bit decision.

4.2 RX SAMPLING AND SNR

This section covers simulated sampling, SNR calculation of the equalized signal and estimated RX BER. To keep the simulation from becoming too lofty of a goal it will be assumed the TX clock and its proper phase is known by the RX sampler. In reality the RX has to recover the clock as it's trying to converge to EQ settings to achieve a specified BER. For this study the TX edge clock is delayed by the channel impulse response delay and then delayed again by roughly ½*UI which sets the sample point close to optimum. This sample phase is further optimized by sweeping the clock across 1 UI and looking for the best RX SNR and Eye Margin (EM) at **Meas_pt3.** This is true for every RX simulation. After sorting through sampling details the signal at the decision circuit can be analyzed for BER performance.

The goal is to know BER at a specified level, in this case 1e-12, and to predict BER performance without unreasonably long simulations. If the computer and program

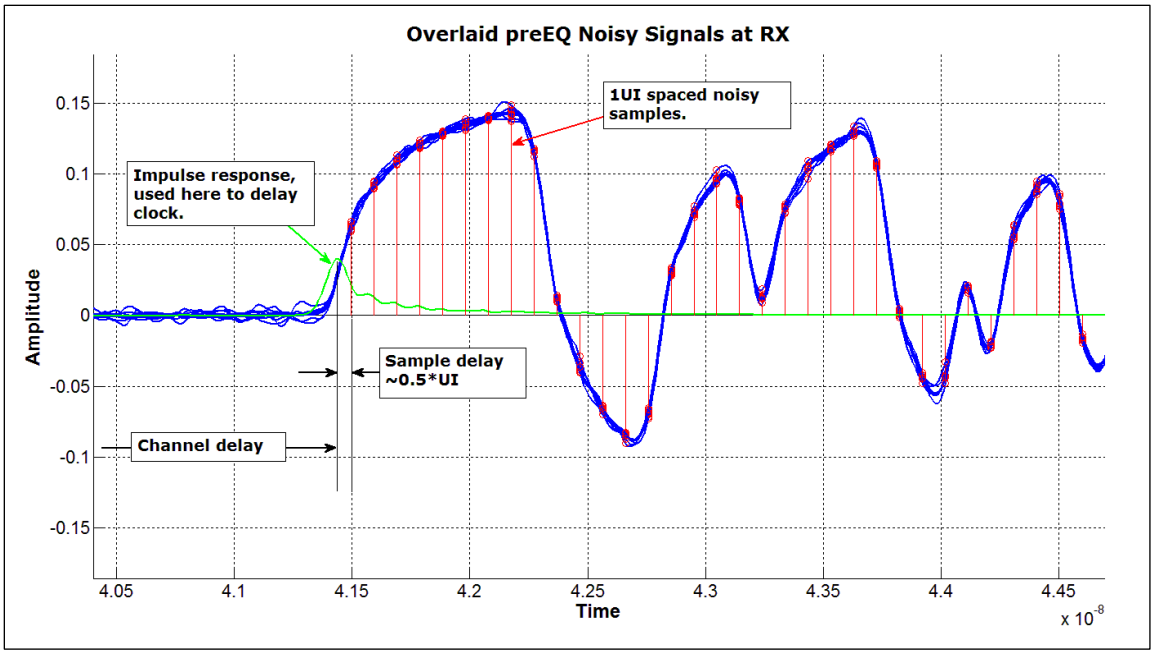


Figure 32, Example of many PRBS9 signals overlaid and sampled by delayed TX clock.

were fast enough 1e12 bits could be simulated and BER could be measured directly to the specified depth. Since this is not the case the residual distortion and the bounded and unbounded random effects will have to be extrapolated to the BER probability. The following diagram shows an RX signal distribution after a ZFE EQ stage. BER will be analyzed and thus RX performance gauged from this distribution across various channels and added RX noise. From *figure33* it is not clear what makes up the sample distribution meaning some could it be actual Gaussian, but some of the noise comes from residual ISI which is bounded in nature and other components come from cross talk sources which are bounded as well. Any way this problem is viewed an assumption will have to be made about the distribution in order to extrapolate to the specified BER without running impractically long simulations. The analysis in this paper assumes all the

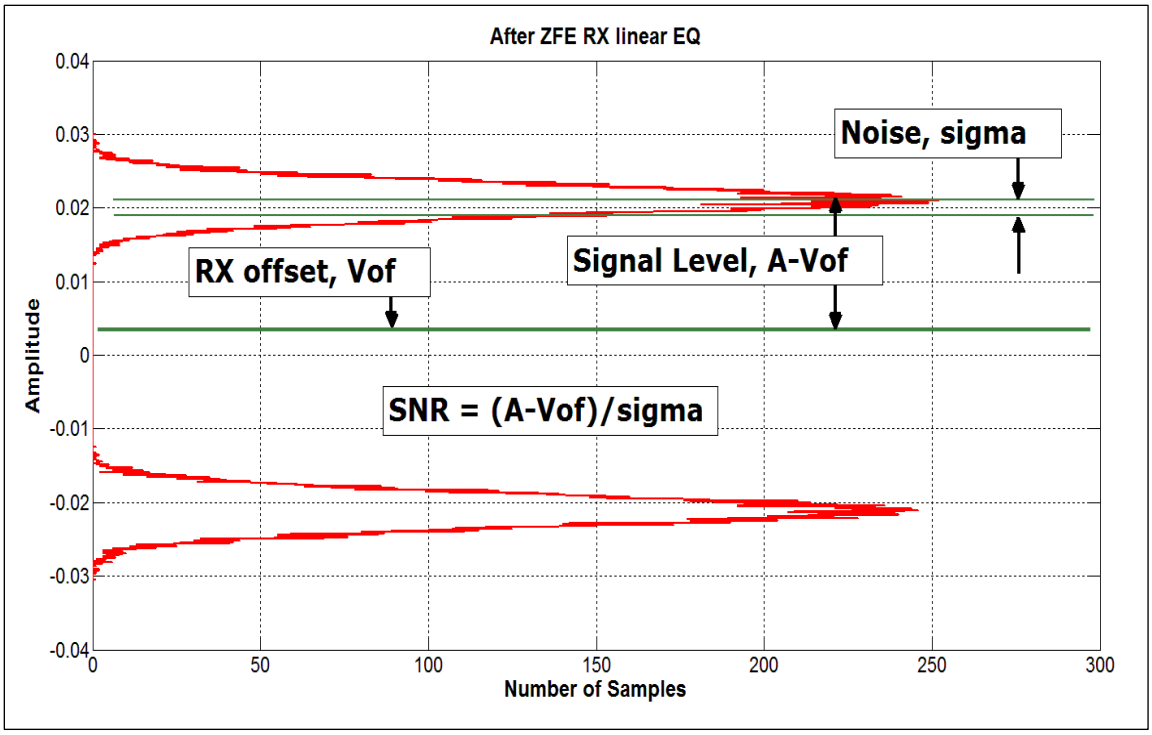


Figure 33, Distribution of RX signal after sampling at the RX comparator.

sources of residual or post EQ noise fits into with reasonable standard error to a Gaussian distribution. It is also understood that simulated BER will be pessimistic with respect to directly measured BER.

$$SNRatBERprobability = erfcinv(2 * 1e - 12) * sqrt(2) = 7.0345$$

Simulated SNR calculation,

$$SNRsimulated = \frac{(A - Vof)}{\sigma}$$

Receiver margin is calculated as,

$$RXMargin = \frac{SNRsimulated}{SNRatSpecBER}$$

A result of 0dB obviously just meets the BER spec and positive dB is margin better than spec and a negative values is worse than spec [1].

In the next section several sections a simplified model is calculated to assist with equalizer convergence algorithm.

4.3 SIMPLIFIED COMPLETE SYSTEM WITH GAIN

The complete detailed TX, H(s) and RX system is very complex. In this section the system is simplified to understand what can be done at the RX in order to gain BER margin beyond simply demanding more power from the link partner TX. *Figure34* below is used to support calculation in the next several sections.

4.3.1 Simplified System Calculation up to FFE Output

In this section a very basic and rough calculation is made for the signal flow from the TX, thru the channel and into the RX up to the output of the equalizer. From *figure34* the EQ output is described by the distribution with mean signal level Sa1. The following equation assumes an ideal amplifier and that the FFE has converged and ideally compensates the signal for channel ISI.

$$Sa1 = A_{VGA1} * Vtx * 10^{\frac{-20}{20}} = A_{VGA1} * 0.1 * Vtx$$

This equation shows an estimate of the available mean signal level at the input of VGA2 for a well behaved -20dB loss channel at the signal Nyquist frequency. This concept is used throughout this study to sanity check simulation results.

4.3.2 Simplified System Calculation from VGA2in to VGA2out

The model in *figure34* is purposely simplified in order to drive intuitive understanding and hopefully a better design. The purpose of *figure34* is to figure out what can be done at the RX to gain BER margin or, put differently, increase the distance the peak of the signal noise is from peak of the noise associated with the threshold level.

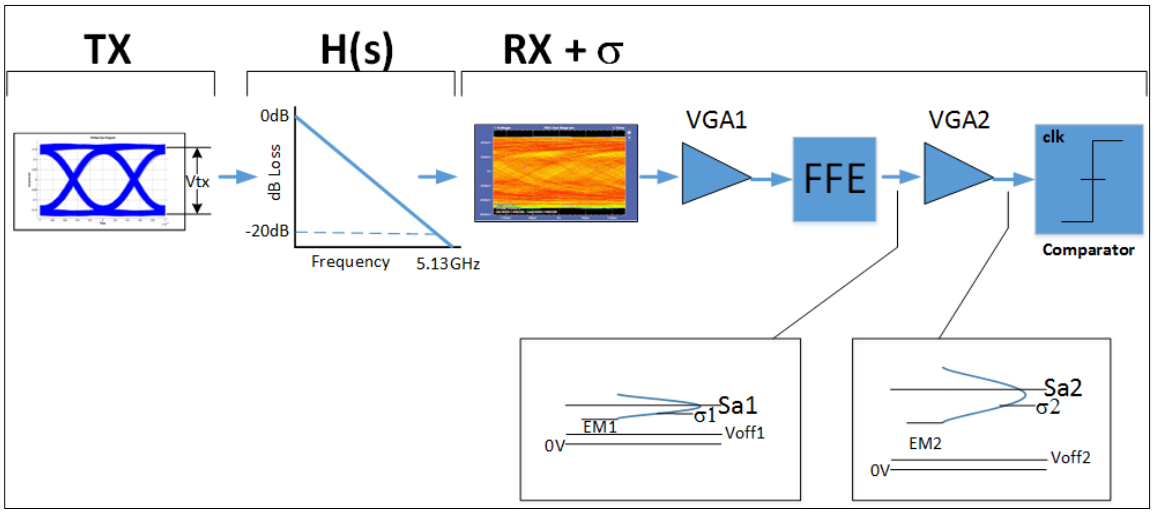


Figure 34, Simplified system block diagram with simplified calculations.

An easy BER improvement solution turns out to be placing properly designed and set Variable Gain Amplifier(s) in the RX path to push the bottom of the noise distribution away from the bit decision threshold. The following BER equations are developed from the distribution diagrams before and after VGA2 from *figure34*.

$$EM1 = Sa1 - \sigma1 * 7.03 - Voff1$$

Where EM1 is Eye Margin before VGA2. Assuming a perfect amplifier for now EM2 shows improvement by the following equation,

$$EM2 = A_{VGA2} * (Sa1 - \sigma1 * 7.03) - Voff2$$

Where EM1 is Eye Margin after VGA2. If hypothetically Sa1=1V, $\sigma 1 = 1\text{Vrms}$ and Voff1=0.01 then,

$$EM1 = 1 - 0.1 * 7.03 - 0.01 = 0.287V$$

and if VGA2 is set to 6.02dB voltage gain then,

$$EM2 = 2 * (1 - 0.1 * 7.03) - 0.01 = 0.584V$$

The 7.03 multiplier is to extrapolate the sigma of a Gaussian distribution to the 1e-12 probability. The EM is also directly measured from the equalized and gained distribution, but is generally an optimistic measurement since there is no practical way to simulate 1e12 bits.

When channels are high loss and signals are very small the noise figure (NF) of amplifiers becomes very important for digital communications systems. In this calculation a perfect amplifier is used to figure out EM2, but the result can still be very useful in specifying a NF for VGA2 that's based on BER margin that can be given up.

4.3.3 Setting Gain

In this section a gain element is developed to increase the mean of the equalized signal to a desired level. For the ZFE case the amplifier gain needed to get the sample point distribution mean to an acceptable level is very easy to calculate. First the signal level, A, before the gain stage is known because the ZFE is a calculation. The desired level, A_desired, is known because it's an input. It follows that the gain needed to get to A_desired is:

$$RX\_ZFE\_Gain = \frac{A\_desired}{A} \quad \text{Where } A\_desired = 150mV$$

The 1UI spaced samples are then multiplied by the RX_ZFE_Gain value.

In a later section the LMS receiver is analyzed. For the LMS RX EQ the pulse response of the channel is not known so there is no easy calculation to get at the available signal at the comparator before the equalizer is done converging. An algorithm needs to be developed to run the EQ for a short time and see what can be gleaned from the signal level with an incomplete RX EQ convergence then set an amplifier gain to increase BER margin.

Of course there is a hundred ways to do one thing so with that in mind a method to equalize the RX signal and apply proper amounts of gain while running an LMS algorithm could be achieved by the diagram in *figure35*.

The first step in the receiver algorithm could be to make all amplifiers and equalizers 0dB and use the comparator as a peak detector in order to find a reasonable setting for

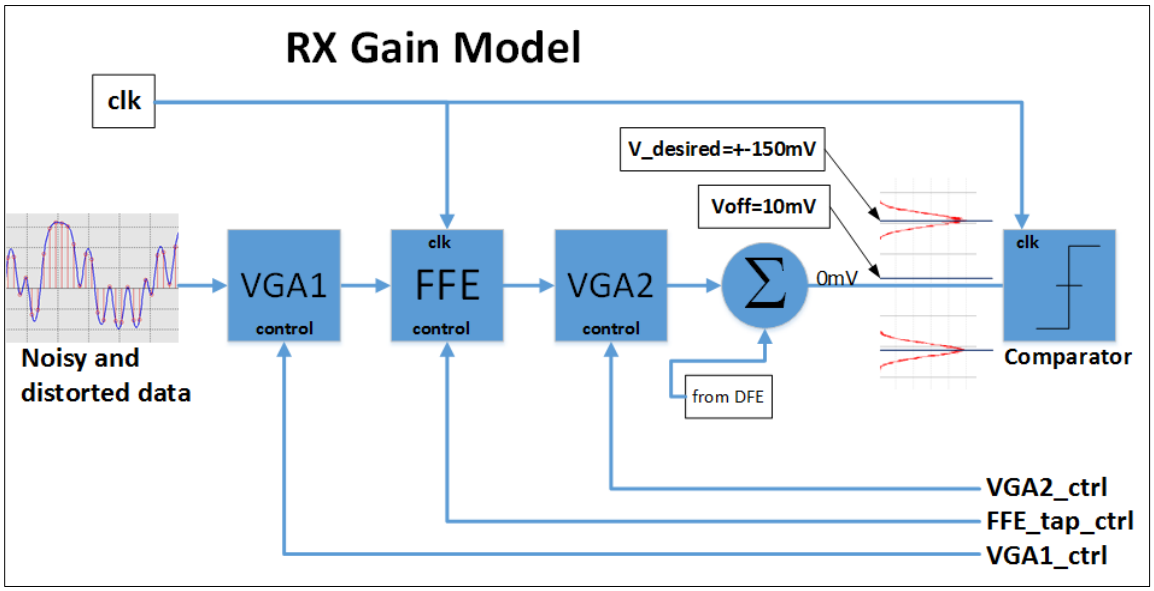


Figure 35, RX block diagram showing tap and gain controls coming from DSP algorithm.

VGA1. For low loss channels the signal is of course larger so VGA1's gain control must be set low enough to keep this first gain stage as linear as possible. For the high loss channels a 101010 pattern could have as much as -30dB attenuation thus getting max signal into the FFE stage might not be as important as controlling added amplifier noise. At this stage an optimum distortion free output level for VGA1 is considered to be 300mVp so in the simulation VGA1 is set by,

$$VGA1\_Gain = \frac{V\_desired\_VGA1\_output}{V\_meas\_peak}$$

where,

$$V\_desired\_VGA1\_output = 300mV$$

This gain should allow the LMS RX function to run faster without adjusting the LMS feedback. The next step could be to let the LMS run and converge to tap values resulting in a signal distribution that is open for the ideal two level comparator, but still needs VGA2 set to get sampled signal at the summing node away from the decision threshold of a more realistic comparator.

$$VGA2\_Gain = \frac{V\_desired\_VGA2\_output}{V\_meas\_peak}$$

where, $V\_desired\_VGA2\_output = 150mV$

The 150mV level at the comparator should keep the BER to the 1E-12 specification even for the high loss channel.

This is obviously just treating gain as a simple multiplier, but as a first pass it shows there can at least be a positive effect. How positive the effect is analyzed in a later section and includes Noise Figure, Linearity, BW and Gain Control.

4.4 RX ZFE

There are many types of equalizers each having distinct positive and negative aspects for a given application. The type that is most direct and easiest to understand is the Zero Forcing Equalizer (ZFE). The ZFE must have knowledge of a properly sampled pulse response at the RX in order to directly determine the taps of an FIR filter. The samples of the pulse around the cursor are directly forced to zero thus all the ISI is cancelled. The cost of forcing the ISI to zero in this way is noise amplification. Some equalizers purposely relax ISI cancellation in order to minimize noise enhancement [4]. Other negative sides to this EQ are requirements for a single bit pulse response somewhere in a training sequence and a detector that can detect many levels rather than simply two levels.

Figure36 shows a pulse response after a measured 7 meter twin axe cable. The pulse is obviously sampled with the taps defined as 1UI space pre-cursor, cursor and post-cursor. The number of taps is directly related to the amount of ISI introduced into the signal by the channel. Visual inspection of this particular pulse shows this channel could make good use of 2-3 pre-cursor taps and 12-15 post-cursor taps both 1UI spaced. Cursor gain can be added for BER margin, but is the subject of another section.

$$PreCursorTaps = 2 \qquad PostCursorTaps = 15$$

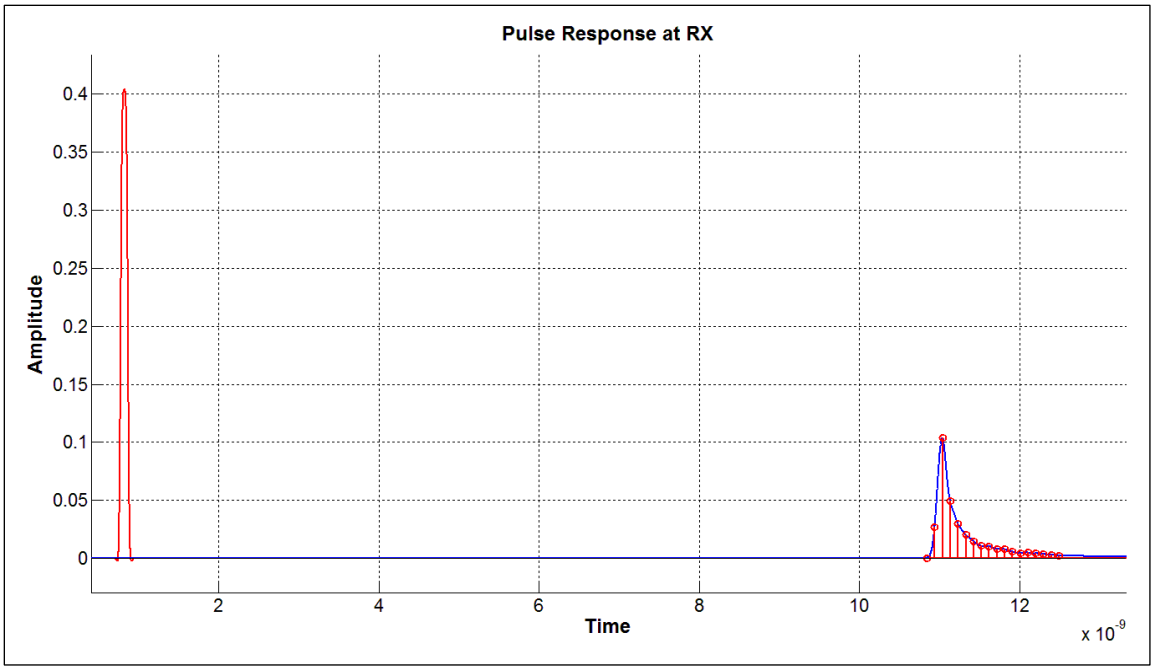


Figure 36, Channel single bit pulse response including 1UI spaced samples.

The basic idea behind the ZFE begins with the following convolution matrix equation. Vector y becomes the desired signal, usually a 1 surrounded by 0's and the matrix x is the pulse sample values and finally c is the filter coefficients [4].

$$[\boldsymbol{y}] = [\boldsymbol{x}\,][crx]$$

This equation has to be solved for c which means matrix x has to the invertible. If the number of pre and post cursor taps are the same then the matrix is square thus inversion can be straight forward. In this case the x matrix is not square thus a least squares estimate must be used.

$$[\boldsymbol{crx_{ZFE}}] = [\boldsymbol{x^{-1}}][\boldsymbol{y}] \text{ , where } \quad [\boldsymbol{y}] = \begin{bmatrix} \boldsymbol{y(-pre) = 0} \\ \boldsymbol{y(..) = 0} \\ \boldsymbol{y(-1) = 0} \\ \boldsymbol{y(cursor) = 1} \\ \boldsymbol{y(1) = 0} \\ \boldsymbol{y(..) = 0} \\ \boldsymbol{y(post) = 0} \end{bmatrix}$$

$$[\boldsymbol{crx_{ZFE}}] = inv(\boldsymbol{x'} * \boldsymbol{x}) * \boldsymbol{x'} * \boldsymbol{y}$$

In most cases this estimate is good enough to create a reasonable filter. For this filter to be used it must be normalized by the sum of the absolute values of the taps. [4]

$$[\boldsymbol{crx_{ZFE}}] = \frac{[\boldsymbol{crx_{ZFE}}]}{sum(abs([\boldsymbol{crx_{ZFE}}]))}$$

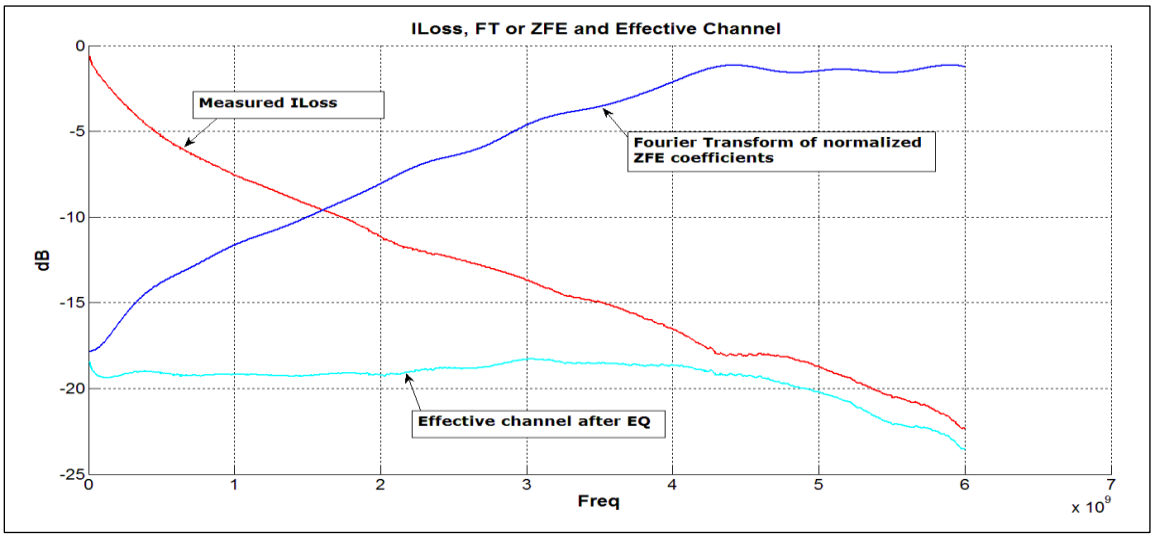


Figure 37, Calculated ZFE coefficients in the frequency domain convolved with measured channel loss.

The filter can now be transformed to the frequency domain and plotted with the measured s-parameter insertion loss revealing the high pass nature of calculated ZFE and how it nearly cancels all the channel ISI at least in the deterministic sense. (See

figure37) It's interesting to note that the effective channel after EQ plot in *figure37* could be made even flatter and extended out to 10.3125GHz with 2 samples per UI filter instead of 1 sample per UI.

In the next sections an adaptive LMS filter is analyzed and simulated.

4.5 RX LMS

In this section the same FIR filter (PreCursor = 3 and PostCursor = 15) as in the previous section is improved by using a Minimum Mean Squared Error (MMSE) algorithm to adapt the filter taps. This type of LMS filter still would require a training sequence to adapt and a multi-level detector, but the concept, with respect to the ZFE, points in the direction of the simplest two level signal link.

From the analysis it is clear to see that the LMS algorithm takes time to converge to the optimum tap values and in fact the time to converge is based on channel complexity (ILoss and RLoss) and noise levels in the signal [1].

The first experiment in the section is run with the same channel as in the ZFE case with 3.6mVrms AWGN band limited to 5GHz summed to the signal at the RX. Since the noise is summed at the RX it does not get filtered by the channel and thus can be used to approximate x-talk noise coupled to the RX signal from an adjacent TX. The first plot (*figure38)* will be in the frequency domain and the converging EQ (green traces), channel ILoss (red trace), converging effective channel (black traces) are all plotted with increasing convergence times. As *figure38* shows the longer the LMS algorithm is

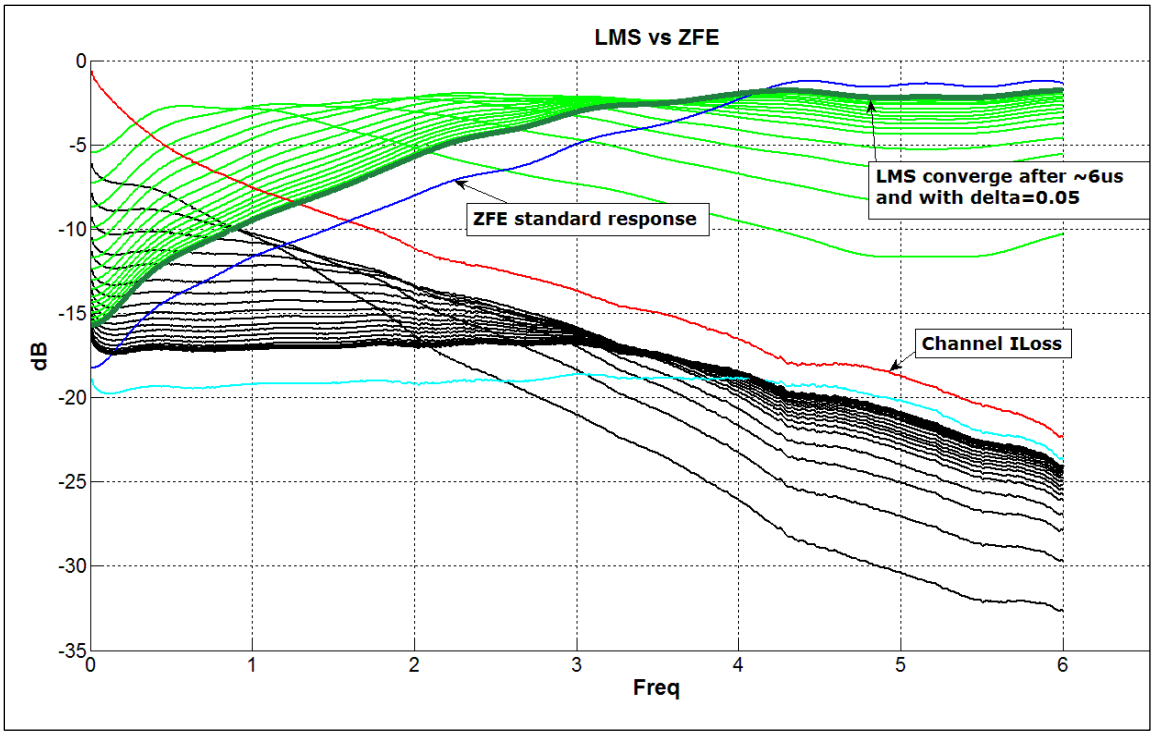


Figure 38, MMSE vs ZFE with MMSE feedback scale set to 0.05. Never converges to ZFE solution.

allowed to run the closer it converges to taps values similar to the ZFE taps. For this experiment the LMS feedback scale is set to 0.05.

It is clear from *figure38* that the simulation needs to run longer or the LMS feedback scale needs to be increased in order to get closer to the calculated ZFE tap values. With the 3.6mVrms of added noise the optimum LMS tap convergence should be about the same as the ZFE tap calculation, but as the noise increases the LMS will tend to relax the tap values to reduce noise amplification and thus increase SNR over the non-adaptive ZFE EQ.

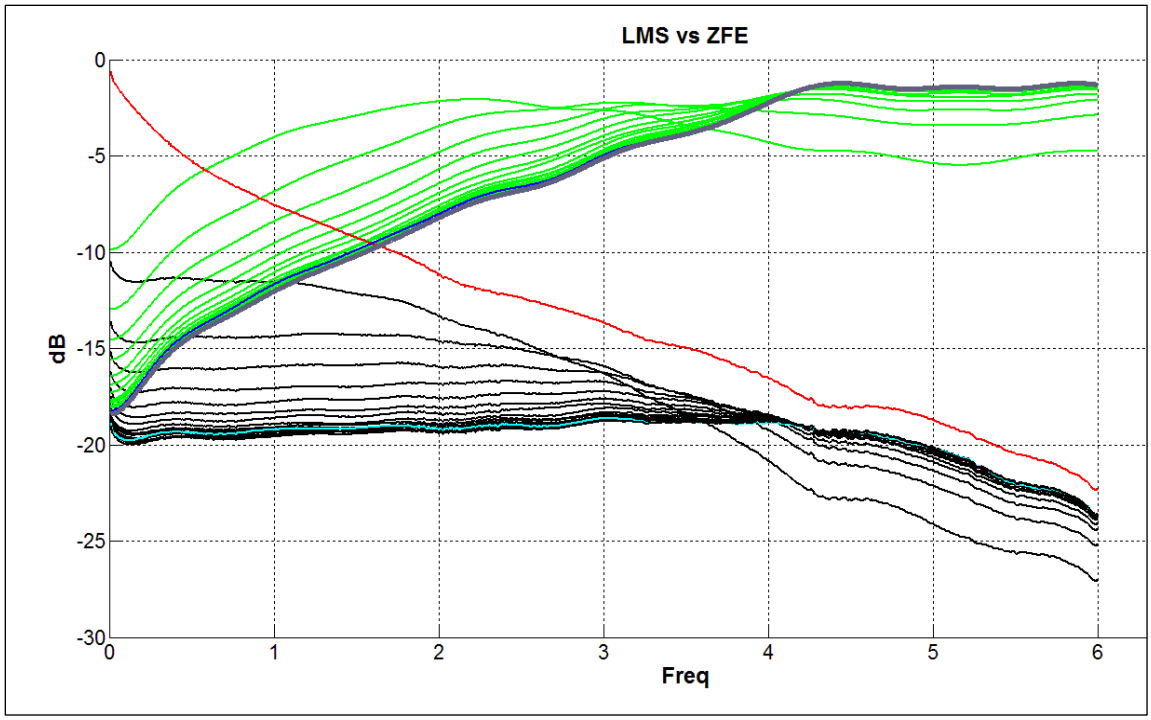


Figure 39, MMSE vs ZFE with MMSE feedback scale set to 0.5. Converges to ZFE solution.

The next experiment (*figure39)* is the same as the first except the feedback scale is set to 0.5. This is very aggressive, but it is clear that the within the same convergence time the LMS converged to practically the same tap values as the ZFE taps. The danger with speeding the convergence up in this manner and this aggressively is that it probably wouldn't be hard to create a channel where the LMS EQ would diverge instead of converge.

The next experiment (*figure40)* is done with the same channel as the previous experiments, but now the ratio of simulated SNR to reference SNR at the BER probability (1E-12) is plotted vs convergence time and the noise is varied from

0.2mVrms to 5.8mVrms. The vertical axis shows SNR margin in that anything above 0dB is positive margin and results below 0dB is negative margin. The blues traces in are the ZFE and as expected the value doesn't change over time however the red traces are the LMS and they reveal better SNR margin with slight noise after ~3us of convergence time. Even without the added noise the LMS would generally converge to better SNR margin due the fact that the ZFE leaves some amount of ISI residual noise at the sample point. The LMS will continue to optimize the tap values beyond the ZFE values until it converges to a better SNR for the added noise less than 2.8mVrms. For noises greater than 2.8mVrms *figure40* shows the LMS SNR margin to be about the same as the ZFE

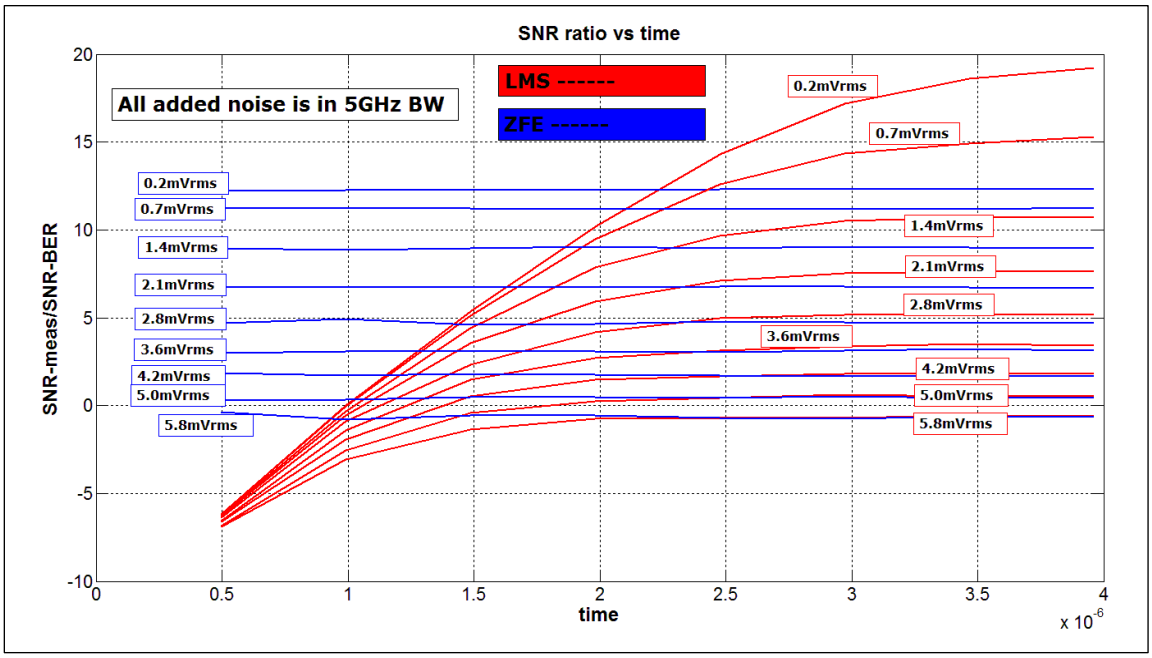


Figure 40, RX convergence time with various RX random noise normalized to 1e-12 BER.

margin. It could be that as the random noise gets larger it becomes the dominant factor in the SNR margin and thus the LMS converges to about the same margin as the

ZFE case. This is of course for a pretty easy channel, 15dB@5.1GHz ILoss and good RLoss, so in the next section a series of increasingly difficult channel will be compared with the same RX EQ settings and added noise.

SNR is good for looking at convergence time but is it is difficult to determine a systems health or BER margin from SNR alone. In the next section extrapolated and directly measured Eye Margin (EM) will be used to determine RX BER margin.

4.6 LMS RX WITH VARIED CHANNEL CHARACTERISTICS

In this section 5GHz BW Added White Gaussian Noise is varied from 0mVrms to 2.9mVrms and the LMS Eye Margin (EM) is plotted for the four channels in the plot below (*figure41*). Two of the four channels vary ILoss and RLoss stresses to create unique signal properties that the RX has to be able to equalize and sample correctly. The 20dB and 5dB ILoss (*figure41*) channels with good RLoss (*figure42*) are meant to be easy channels and they are expected to show better EM then the two stressor channels. Stressor channels are -30dB and -18dB ILoss and both have equivalently poor RLoss. The poor RLoss creates reflected signal which ends up being out of reach of the EQ taps thus adds to the residual noise. Floating RX EQ taps could help equalize reflected signal, but is beyond the scope of this study.

After some experiments a good starting point for the LMS RX controls is as follows:

LMS feedback scale = 0.08
Desired signal after VGA1 = 300mV
Desired signal after VGA2 = 150mV
TX signal level = 400mVpp
FFE taps pre-cursor = 5
FFE taps post-cursor = 15

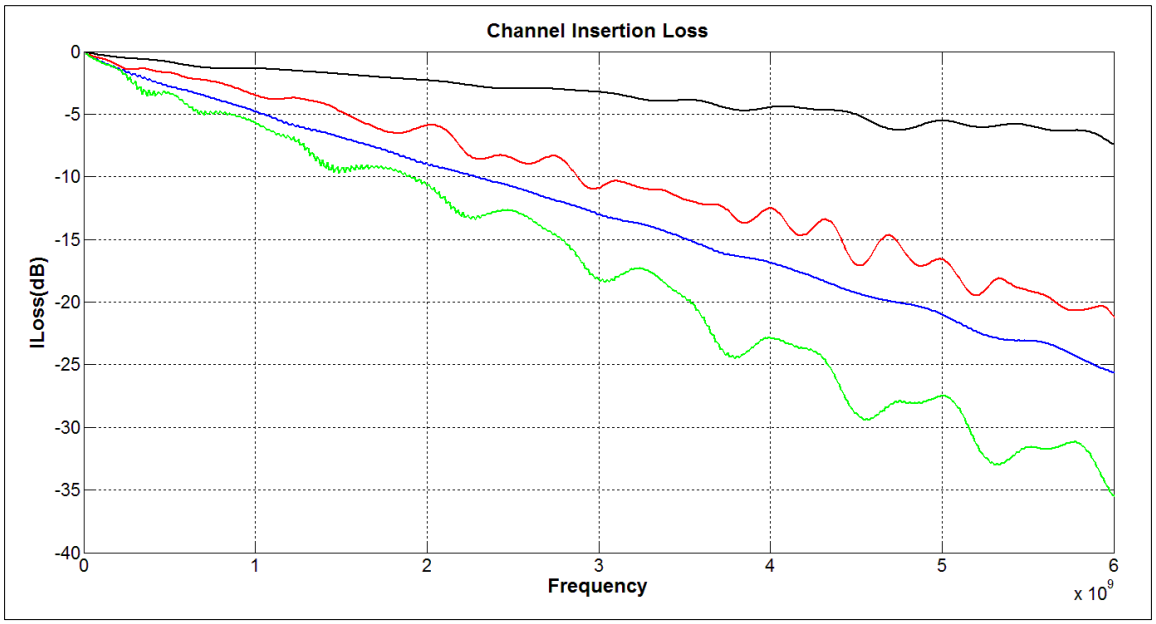


Figure 42, Insertion Loss of test channels.

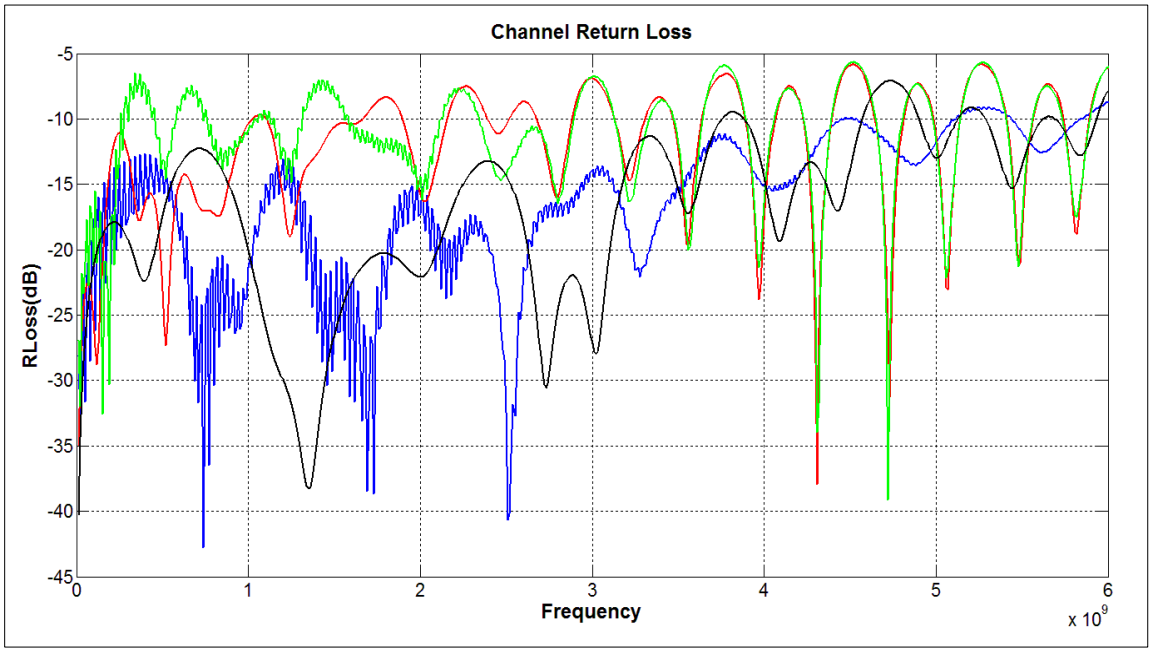


Figure 41, Return Loss of test channels.

4.6.1 -5dB Channel with Good RLoss

The -5dB ILoss channel really doesn't even need an equalizer and in fact it becomes a stress case due to RX amplifier linearity. If either controls on the VGAs are set too high, then amplifier distortion effects could adversely affect EM.

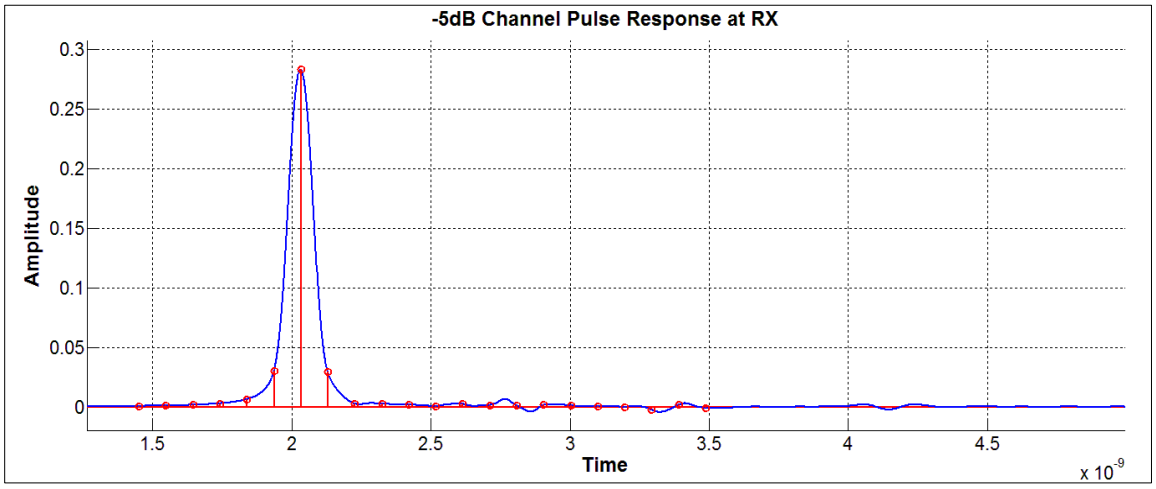


Figure 43, Single bit pulse response of -5dB channel.

Figure43 shows that the LMS converged well before 2us which is expected for a signal that has very little ISI. The pulse response in *figure43* shows a two tap EQ would have been sufficient and that the un-attenuated reflections in the signal are small enough to ignore. The VGA gain settings are below and are set as expected. The gains below can be determined by the measured RX peak, desired levels and the EQ DC attenuation. Working backwards to sanity check and using the VGA1 and VGA2 gains the desired level of 150mV is calculated below.

Nyquist_Gain_required_VGA1 = 3.6976 dB
Nyquist_Gain_required_VGA2 = -0.70836 dB
Vpeak_RX = 190 mV
EQ_DC_Attenuation = -4.82 dB

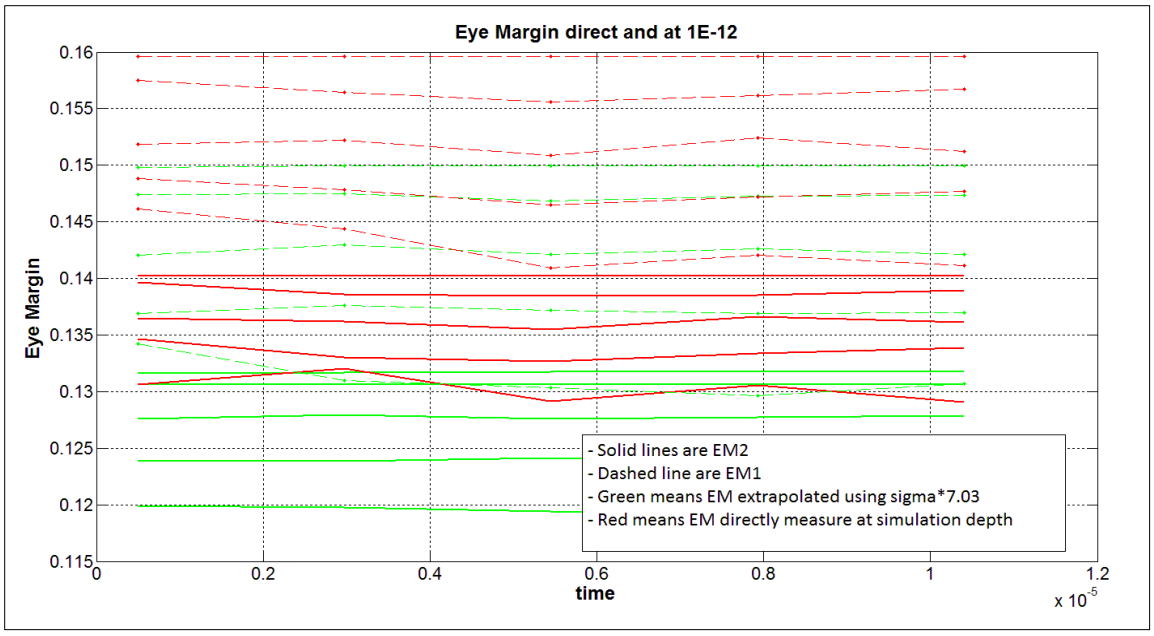


Figure 44, RX convergence time vs Eye Margin with easy -5dB channel.

V_desired = 0.190*10^(3.7/20)*10^(-4.82/20)*10^(-0.71/20) = 155 mV

The positive EM effect with an ideal amplifier and from *figure44* is only about 10mV of RX eye opening improvement over the 120mV opening prior to VGA2. For this channel the signal shows up at the RX with plenty of signal swing and very little ISI. So much so that amplifier linearity is concern over added amplifier noise. In this case the amplifier is probably not needed but it has to be present due to the other channels that need to be supported.

4.6.2 -20dB Channel with Good RLoss

The -20dB channel uses more capabilities of the RX, but the channel is still relatively easy to equalize to a good EM. *Figure45* shows a reduction in EM relative to the previous channel. As the RX added noise increases both the extrapolated margin (green

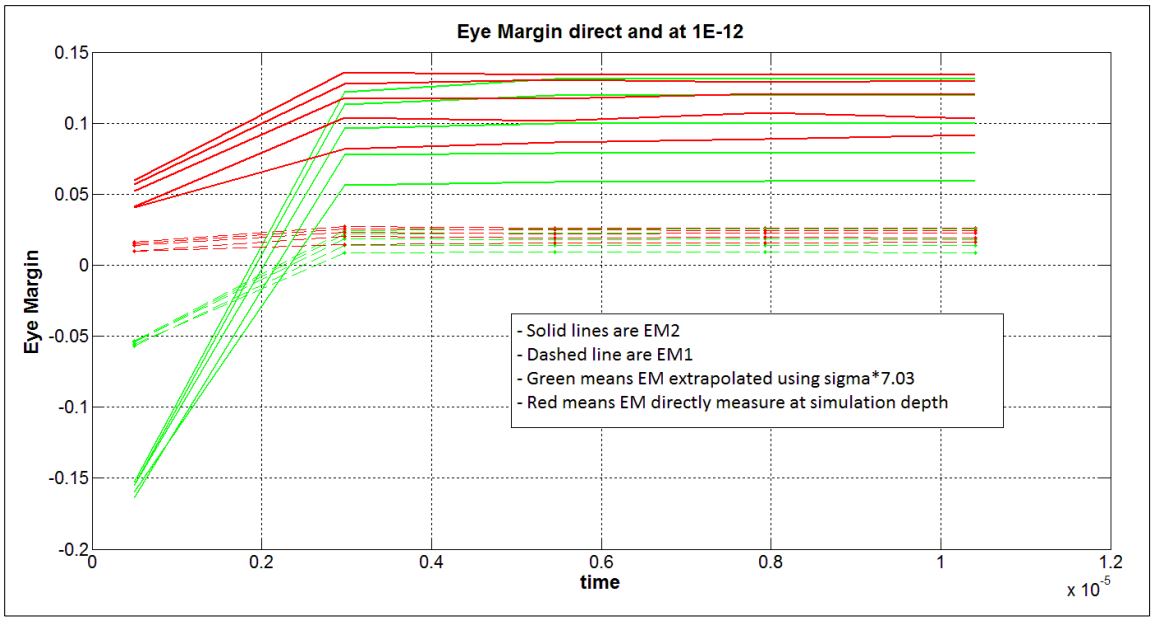


Figure 45, RX convergence time vs Eye Margin with -20dB good return loss channel.

traces) and the directly measured (red traces) margin decrease and as expected the extrapolated shows pessimistic margin.

The simulation is again sanity checked for reasonable results.

Nyquist_Gain_required_VGA1 = 5.033 dB
Nyquist_Gain_required_VGA2 = 13.171 dB
Vpeak_RX = 170 mV
EQ_DC_Attenuation = -19.9 dB
V_desired = 0.170*10^(5.033/20)*10^(-19.9/20)*10^(13.171/20) = 140 mV

For this -20dB channel the positive EM effect with an ideal amplifier and from *figure45* is about 25mV of RX eye opening improvement over the 25mV opening prior to VGA2 with 2.9mVrms of added random noise. For this channel the signal shows up at the RX with a signal swing that needs to be amplified. In the -5dB channel case noise figure was not the concern, but for this case added amplifier noise should be considered. In the next

section the most stressful channel is simulated and is where the added noise of an amplifier should be specified.

4.6.3 -30dB Channel with Bad RLoss

The -30dB channel with poor RLoss should stress this system beyond a practical specification. From *figure46* below it is plain to see that the LMS RX takes longer to converge with the increased ILoss and decreased RLoss. This is expected for the LMS algorithm.

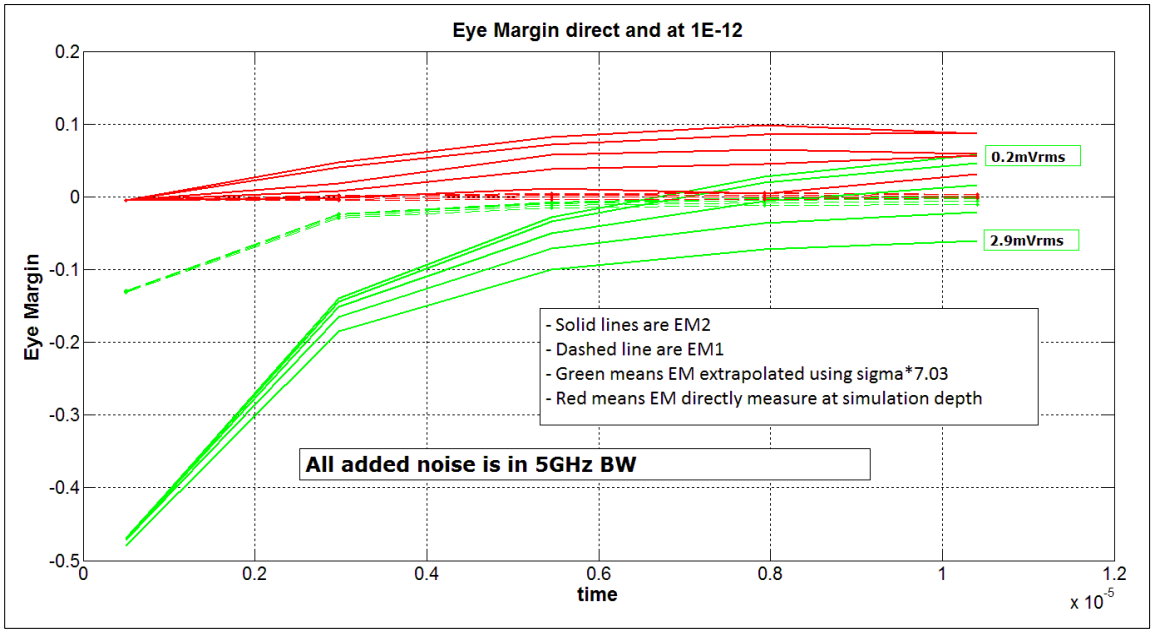


Figure 46, RX convergence time vs Eye Margin with -30dB channel.

This channel breaks the BER margin for this system even with the ideal amplifier. With the 2.9mV AWGN at the RX input the extrapolated EM shows nearly 60mV of negative eye margin. This translates to a BER of greater than 1E-12 even with a noiseless comparator circuit.

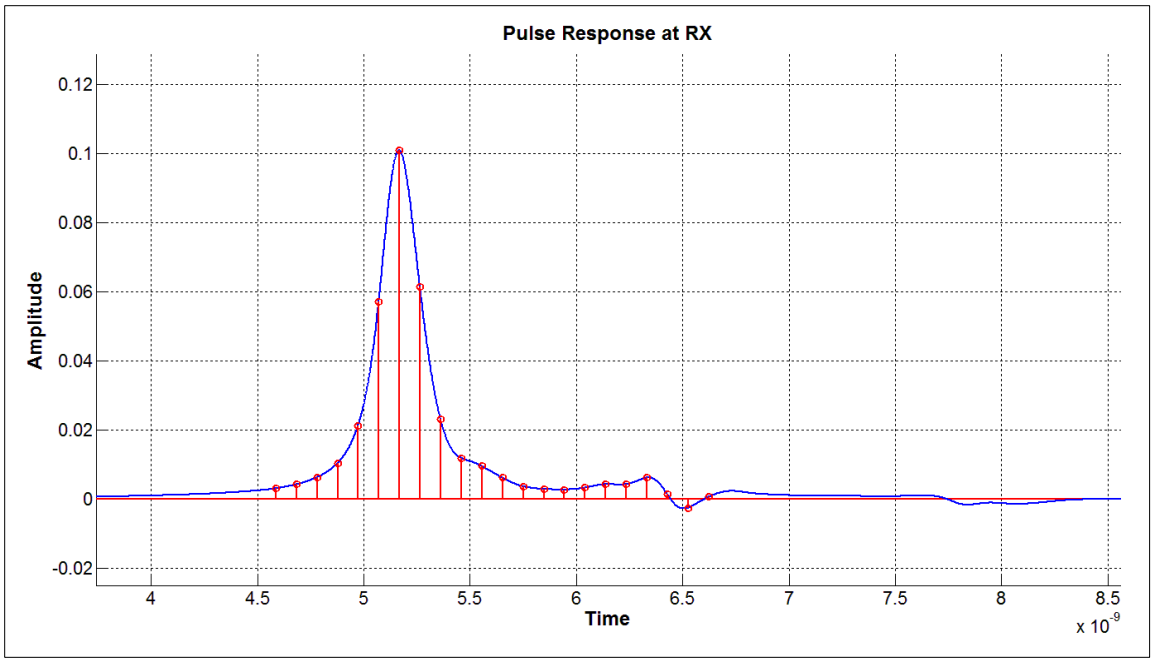


Figure 47, Single bit pulse response with -30dB channel.

There are some options to gain more margin, but none seem to completely fix the problem. A potential fix could be to lower the noise that shows up at the RX with the signal. The value of 2.9mVrms of added noise models cross talk noise from adjacent transmitter so a better isolation could be specified reducing the noise to 1.5mVrms. With 1.5mVrms of AWGN the BER<1E-12, but leaves little room for RX added noises for the VGA and comparator.

The pulse response in *figure47* is included just to show how stressful this channel is to the RX.

4.6.4 Specifying Amplifier (VGA2) Noise Figure

The -30dB channel was too much for this system as designed so specifying an amplifier property for it would not be useful. The -20dB channel had some margin to give up to real circuit effects so it is used to determine a noise figure allowed.

Noise figure is defined as by the ratio of the input SNR to the output SNR [3].

$$NF = \frac{SNR_{input}}{SNR_{output}}$$

$$NF = \frac{\frac{Sa1^2}{\sigma 1^2}}{\frac{Sa2^2}{\sigma\_extrapolated^2}}$$

To calculate NF $\sigma 2$ is increased until its extrapolated SNR reaches the specification limit of 1E-12 probability. This can be found by,

$$\sigma\_extrapolated = \frac{Sa2}{7.03}$$

For the max AWGN case and the -20dB channel the maximum NF for VGA2 (assuming ideal VGA1) is calculated by,

$$NF\_dB = 10 * log10\left(\frac{\frac{Sa1^2}{\sigma 1^2}}{\frac{Sa2^2}{\sigma_{extrapolated}{}^2}}\right)$$

$$NF_{dB} = 10 * log10\left(\frac{\frac{0.044^2}{0.0036^2}}{\frac{0.10^2}{0.013^2}}\right) = 4.3\ dB$$

This is the NF for only VGA2 while holding the other elements ideal. In reality margin would have to be taken from this calculation and given to other elements of the receiver.

5 MATLAB FUNCTIONS

As noted throughout this paper a group of Matlab function were developed from scratch using only native Matlab functions. As much as possible functions are designed to be scalable to any bit rate, but are currently always restricted to 2 level signaling.

5.1 TX FUNCTIONS

The TX function is designed to make a signal vector look as if it could have been measured by a real time oscilloscope in the lab. The function arguments are pat type, bandwidth, bit rate, voltage noises, timing noises and 1UI spaced Feed Forward Equalizer. All added noises are band limited and the band widths are user selectable.

```
%Input arguments to TX function
cm1=0;                  %pre-cursor EQ tap.
c0=1;                   %amplitude tap.
cp1=0;                  %post-cursor EQ tap.
bit_rate=10.3125e9;     %bit rate of signal.
SampPerUI=16;           %number of voltage points per bit.
BW_Signal=10e9;         %BW of 4th order linear phase low pass filter.
seq_manual=[-1,-1,1,-1……];      %set pat_type to “none1” to use seq_manual.
Zdc=0.0001;             %DC resistance of thru channel.  If xtalk use 1e10 ohms.
pat_type='prbs9';       %prbs pattern length.
pat_repeats=10;         %only used for prbs patterns.
J_rand_rms=0;           %random jitter added to signal.
J_rand_BW=5e9;          %random jitter band width.
J_sine_amp=0*UI_time;       %sinusoidal jitter amplitude.
J_sine_freq=4e6;        % sinusoidal jitter frequency.
No_TX=0e-3;

%TX function
[seq, x, t, t_clk_noisy, t_clk_clean, Ts_return] = pat_gen_jitter_noise_func_rev4(…
,pat_type…
,UI_time…
,SampPerUI…
,pat_repeats…
,seq_manual…
```

```
,v_high…
,v_low…
,RTFT…
,cm1…
,c0…
,cp1…
,No_TX…
,J_rand_rms…
,J_rand_BW…
,J_sine_amp…
,J_sine_freq);
```

5.2 CHANNEL FUNCTION

The channel function convolves the TX signal created described in section 5.1 with a measured, simulated or calculated set of s-parameters. This function band limits the signal to max frequency in the s-parameters. To preserve sample rate from the input signal the signal is zero padded after signal is brick wall limited.

```
%Input arguments to channel function.
Freqs = [10e6,20e6,….];        %frequencies loaded from parsed s-parameter file.
sdd_thru = [0+0*I,….];         %ILoss loaded from parsed s-parameter file.
x_BL = [0.012, 0.1,….];        %voltage samples of signal to be convolved.
Ts_return = 6.08e-12;          %Ts value returned from noisy TX function.
Zdc = 0;                       %manually added DC value to channel.

%channel function
[y,t_y,h,t_h,h_delay,freqs,sdd_thru]=propagate_x_thru_sparam_linear_rev0p2_WIP(…
,freqs…
,sdd_thru_total…
,x_BL…
,Ts_return…
,Zdc);
```

6 SUMMARY

6.1 GENERAL

This practical study of wired digital communications highlights a system from reference clock to RX bit decision. It is clear that the simplest high speed digital communication link is a huge subject and thus it has proven very difficult to keep the complete system in mind while analyzing one block with more depth. Seems like this is where a team of engineers comes in handy.

As this study progressed it became apparent that this is just a framework to use as a start point to develop a specification or a high level description before a real design begins.

6.2 RX CONVERGENCE WITH ASYNCHRONOUS CLOCKING

Throughout this paper assumptions and simplifications were made and of them the lack of a RX recovered clock is the one that deviates from what really happens in a modern digital RX. As previously stated the clock recovery circuit needs a relatively open eye signal to recover the clock and a good clock is needed to sample and converge an EQ. It does not seem like there is any close form solution to this besides a very specific training pattern within a specified training period. If there is no training as assumed in this paper then the RX would almost have to start taking guesses at start points for the signal and begin running various RX blocks until an SNR or BER monitor shows something reasonable. After the RX thinks it is sampling at the correct phase and frequency then the fine tuning can begin. This is just the beginning of the RX

convergence issues and from here it really starts to seem like a combination creative recipe building in order to cover all the channels that could be in a system.

6.3 LMS Assumes High Resolution ADC

The simple two level CDR based RX is just a 2bit ADC, but throughout this paper it is assumed that the precise signal level is known at the RX for the LMS to generate an error that get passed back through the LMS feedback loop, but that would seem to require an ADC with more than 2 bits. The fix here is to implement the sign-sign LMS algorithm. Sign-sign LMS converges an EQ by simply knowing the sign of the error signal which is perfect for a 2 level RX. This will be a future addition to this study.

6.4 LMS Noise and DFE

The RX EQ used is an FFE linear equalizer which adds some of its own noise to the signal, but most of all it amplifies the noise on the RX signal. In most modern designs good use is made of a Decision Feedback Equalizer (DFE) which does not amplifier the noise on the signal, but it is more complicated in that the theoretically noiseless bit decision is fed back thru and FFE filter to the comparator summing node. This presents many problems related to feeding back incorrect decisions and stability.

6.5 Ideal Amplifier

In section 4 ideal amplifiers were used to gain the RX signal and increase Eye Margin (EM). This was used as a start point in order to eventually define a complete RX added noise specification. More work needs to be done related to this, but it was shown that in the most stressful case (-30dB channel) that there was no room for intrinsic RX noise

due to the unavoidably noise circuits. At this point a more transistor level approach would need to be taken to see what can be achieved in the RX circuits and then trade off with channel budget, TX power or even BER.

6.6 MATLAB ANALYSIS CODE

Once the number functions used for analysis start become more than 10 and each has several revisions it's best to be vigilant about revision control. Many hours were wasted debugging issues that root caused to lack of revision control of Matlab functions.

6.7 LTSPICE

Using LTSpice for the simulation and then Matlab for signal analysis may have been a better approach for the study. If behavioral models in LTSpice were used for the communications system simulation it would have been feasible to insert more detailed transistor level circuits for future study.

7 REFERENCES

[1] Proakis, John G. and Salehi, Masoud. Digital Communications. NY: McGraw-Hill, 2001

[2] Peng Li, Mike Jitter. Noise and Signal Integrity at High-Speed. Boston MA: Prentice Hall, 2007

[3] Razavi, Behzad. RF Microelectronics. Boston MA: Prentice Hall, 2012

[4] Hall, Stephen H. and Heck, Howard L. Advanced Signal Integrity. Hoboken NJ: Wiley, 2009

[5] McMahon, David. Signal and Systems Demystified. NY: McGraw-Hill, 2007

[6] Pozar, David M. Microwave Engineering. Wiley, 2005

[7] Derickson, Dennis and Muller Marcus. Digital Communications Test and Measurement. Boston MA: Prentice Hall, 2008

[8] Bogatin, Eric. Signal and Power Integrity Simplified. Boston MA: Prentice Hall, 2010