Equalizers for Communications Satellites

by

Leon Fay

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2008

Author ....
Department of Electrical Engineering and Computer Science
August 22, 2008

Certified by ....
David Materna
Assistant Group Leader, Advanced Satcom Systems and Operations
MIT Lincoln Laboratory
Thesis Supervisor

Certified by. ....
Joel L. Dawson
Carl Richard Soderberg Professor of Power Engineering, EECS
Thesis Supervisor

Accepted by . ...
Arthur C. Smith
Professor of Electrical Engineering
Chairman, Department Committee on Graduate Theses

This work was sponsored by the Department of the Air Force under Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the author and are not necessarily endorsed by the United States Government.

Equalizers for Communications Satellites

by

Leon Fay

Submitted to the Department of Electrical Engineering and Computer Science
on August 22, 2008, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

This thesis investigates equalization for advanced protected satellite communications systems in development at MIT Lincoln Laboratory. Equalizers facilitate high data rate communication by correcting dispersion in the transmitter and receiver signal chains. An automated calibration procedure for finding optimal equalizers was developed. Repeated testing addressed questions about noise amplification, filter complexity requirements, and narrow band performance degradation. After examining various architectures, it was determined that the FIR filter was the best equalizer structure given the nature of the channel. The basic calibration procedure was also extended for use at high RF frequencies by using a spectrum analyzer as a tuned receiver.

Thesis Supervisor: David Materna
Title: Assistant Group Leader, Advanced Satcom Systems and Operations
MIT Lincoln Laboratory

Thesis Supervisor: Joel L. Dawson
Title: Carl Richard Soderberg Professor of Power Engineering, EECS

Acknowledgments

I would first like to thank my thesis advisers Dave Materna and Professor Joel Dawson. Their guidance and advice helped me sort through tough problems, stay grounded in reality, and maintain focus. I would also like to thank Paul Lawson, Jim Vian, and the many other people at Lincoln Laboratory who were more than willing to take time out of their day whenever I needed help.

Most of all, I would like to thank my parents, Victor and Larisa. They have always inspired me to do my best and never give up. I owe everything to their love, support, and hard work.

Contents

List of Figures

Chapter 1

Introduction

1.1 Protected Communications Satellite Systems

MIT Lincoln Laboratory is currently participating in the development of advanced protected communications satellite systems. These next generation military satellites will provide secure, high speed connectivity almost anywhere in the world. However, higher data rates are always accompanied by a loss in reliability. To compensate for this unavoidable performance degradation, non-fundamental sources of errors need to be reduced. Chief among these sources is the communication channel, which includes any elements between sender and receiver that affect the signal.

Although non-ideal channels are a common problem of many communications systems, protected satellites present several unique challenges. Much of the difficulty comes from the high frequency of operation. At a 20 GHz or 44 GHz carrier frequency, it is hard to make ideal analog components for the transmitter and receiver. The signal bandwidth can also be several hundred megahertz, which makes baseband processing difficult.

Another unique aspect of protected satellites is that their service must be secure and resistant to jamming. Many systems make the energy requirements for jamming impractical by spreading communication over a wide range of frequencies. This is accomplished by periodically changing the carrier frequency while sending data. The overall spectrum of the signal appears much larger than what is actually needed.

This technique is called frequency hopping, and each change in frequency is called a hop. Frequency hopping makes communication more secure, but also complicates the system design. Rather than a fixed upconverter and downconverter, the system requires variable frequency conversion over a wide range, with fast settling time.

When high data rate communication is not possible due to bad weather or partial obstruction, communication satellites use lower data rate modes. These modes have lower symbol rates and less dense constellations in order to increase reliability. Whatever method is used to compensate for the channel and help the high data rate modes, the other modes should not be degraded.

1.2 Linear and Nonlinear Channels

Most channel imperfections can be categorized very generally as either dispersion or nonlinear distortion. A dispersive channel has gain and phase that changes as a function of frequency. Note that sometimes dispersion only refers to variations in phase (and group delay) as a function of frequency. Here, it will include frequency dependent changes in amplitude as well. In terms of linear systems theory, a dispersive channel has a non-flat frequency response.

The main problem with dispersion is that it causes intersymbol interference (ISI). ISI is the result of pulses that are spread out in time and non-zero for more than one sampling instance. Thus, a single sampling instance can be affected by many surrounding pulses, introducing data dependent errors. In the frequency domain, we can understand ISI in terms of the Nyquist criterion [1]. The Nyquist criterion states that no ISI occurs if the Fourier transform of the pulse, $H(f)$, satisfies

$$\sum_{k=-\infty}^{\infty} H\left(f - \frac{k}{T_s}\right) = T_s \tag{1.1}$$

where T_s is the symbol period. The simplest example of a no ISI pulse is one that looks like a low pass filter in the frequency domain, and the symbol rate is equal to the filter's bandwidth. In this case, it is clear that the shifted copies of $H(f)$ add up to a constant and satisfy the Nyquist criterion. If $H(f)$ had the same bandwidth but was non-flat, then the symbol rate would need to be lowered in order to maintain no ISI. This is because the copies of a non-flat $H(f)$ would have to be shifted closer together to add up to a constant, which corresponds to a reduced symbol rate. Figure 1-1 illustrates this point. Dispersion in the communication channel can change the pulse shape, causing it to violate Nyquist. This introduces ISI unless the symbol rate is reduced.

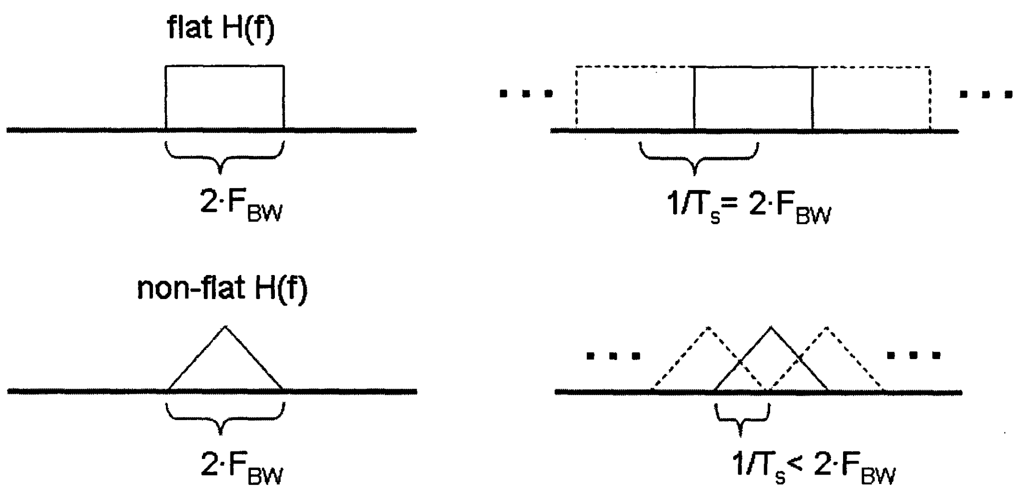


Figure 1-1: Nyquist Criterion example.

Nonlinear distortion encompasses a broader category of non-ideal channels. In addition to possible frequency dependent effects, nonlinear systems are characterized by a dependence on amplitude. This dependence greatly complicates analysis and makes most of linear systems theory inapplicable. One of the unique characteristics of nonlinear distortion is signal expansion in the frequency domain, called spectral regrowth. This is a serious problem in communication systems where the channels are closely spaced. Similarly to the way that nonlinear functions create harmonics of a signal that were not originally present, spectral regrowth introduces side lobes that could interfere with adjacent channels.

If distortion in a channel also depends on frequency, it can create symbol errors that appear similar to ISI but are much more difficult to correct. In this case, the nonlinearity has a "memory," since the output depends on the current and past values of the input in a nonlinear way [2].

1.3 Channel Examples

Many common communications systems have dispersing and distorting channels. Modems sending data through telephone cables are one of the most common examples of dispersive channels. Telephone cables have echoing due to imperfect termination and gradual frequency roll-off due to conductive and dielectric losses. Echoing is actually similar to filtering, since it involves different copies of the signal interfering with each other. For example, a system where the round trip time of the echoes is constant and causes a 50% power loss would have an impulse response like the one shown in Figure 1-2. The first impulse represents the main signal, and the impulses that come after represent how the echoes differ in amplitude and phase from the original. Since both echoing and roll-off can be modeled as linear filters, the telephone line itself can also be called linear. Compensating for dispersion on telephone lines has led to drastic increases in the data rates of cable modems [3].

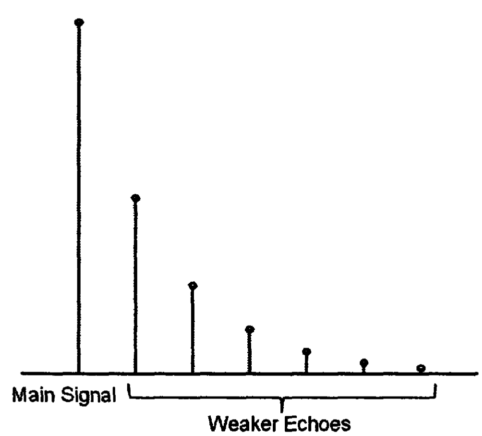


Figure 1-2: An impulse response that models echoing.

The channel for cell phones is more complex, exhibiting both linear and nonlinear behavior. The linear component comes from multipath, which in some ways is similar to the echoing in telephone lines. Multipath arises from crowded environment where there are many possible routes between the transmitter and receiver. The signal goes through all these routes and many different versions arrive at the receiver, each with its own amplitude and phase. The main difference between echoing and multipath is that multipath varies quite rapidly in time. Any scheme that attempts to compensate for multipath must therefore be able to adapt quickly. Also, it is likely that the short wavelengths used in cellphones will interfere, resulting in deep nulls in the channel [4].

The nonlinear component of the cell phone channel comes mostly from the power amplifier (PA). All amplifiers have nonlinear compression if driven hard enough. In many applications it is acceptable to simply avoid the nonlinear regime. The compression can be moved further out by increasing supply voltages and/or bias. However, in portable systems where energy is scarce, there is a strong incentive to move as close as possible to the nonlinear regime without completely distorting the signal. Fortunately, a PAs nonlinearity is approximately memoryless and can be fully characterized as changes in gain and phase shift as a function of input power [2].

There has been a great deal of research focused on compensating imperfect cell phone and modem channels. To make use of this prior work, we will examine how the channel for the satellite systems might look in comparison. Just like cell phones, satellites have limited energy sources, so they also share the problems of nonlinear PAs. As we will see in the next section, the PAs in satellite systems are substantially more difficult to compensate because of the high signal bandwidths. Unlike cell phones, there is usually a direct line of sight between the transmitter and receiver in satellite communication, so multipath is not a major issue. This means that the ability to continuously adapt to the channel is not crucial for satellites, although perhaps still useful due to temperature drift and aging.

The main source of dispersion in satellite systems is the complex signal chain in the transmitter and receiver, which must support high frequencies, high bandwidths, and frequency hopping. The resulting frequency response can be much more difficult to compensate than the gradual roll-off of cables. However, it is unlikely that the response will include and any of the deep nulls that commonly occur in multipath.

1.4 Correcting Linear and Nonlinear Channels

Channel distortion and dispersion is corrected mainly by analog or digital techniques. Analog approaches are interesting because of their potential speed, low cost, and low power. Their main weakness is addressing complex functionality, which fortunately digital solutions can handle. Digital approaches, however, do have trouble reaching the extremely high speeds required for some schemes due to the limitations of DACs.

Both analog and digital circuits can correct for dispersion by synthesizing a filter that approximates the inverse frequency response of the channel. This technique is called equalization, and can significantly reduce ISI. The analog implementation works in only a limited number of cases due to the inability to arbitrarily set the the filter's response. When a simple high pass filter is required to overcome a gradual roll-off, for example, an analog circuit could prove to be extremely efficient. Such a circuit could even be complemented with digital feedback and control so that some properties of the filter could be tuned to an optimum. However, an analog equalizer does not easily scale up to meet the challenges of a complex channel. In this case, digital solutions shine. Although requiring a power hungry DSP, an FIR filter can easily take on shapes that would be difficult to reproduce in analog. For this reason, in any application where equalization is not simple, digital equalizers are preferred [2].

Correcting nonlinear distortion can also be accomplished in either analog or digital circuits by applying a function to the signal that is the inverse of the channel's nonlinearity. This technique is called predistortion. A typical digital predistortion architecture is shown in Figure 1-3. The baseband signal is generated digitally and is multiplied by a factor selected from a lookup table. The multiplication factor is a function of the signal's power, and would ideally cancel out any nonlinearities in the PA. After multiplication, the signal is converted to analog through a high-speed DAC, upconverted to passband, and finally sent to the PA. The lookup table can be gradually filled and corrected by comparing the PAs output to the desired signal [2].

This architecture can be very effective but does not work well for wideband sys-

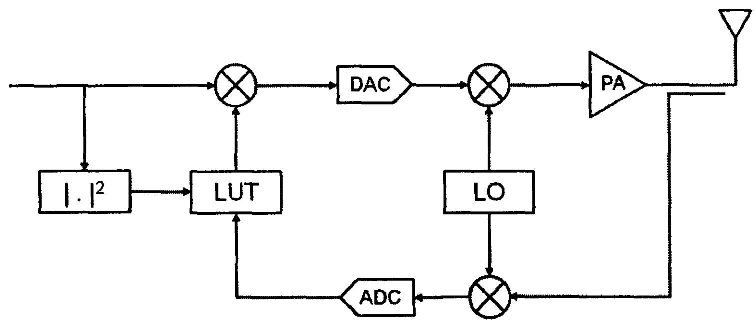


Figure 1-3: Digital predistorter.

tems. This is because the DAC must operate near its limits to faithfully reproduce the baseband signal. Adding the harmonics to cancel nonlinearities usually pushes the required DAC speed to beyond what is possible. For example, if the PA has a cube root nonlinearity, then the predistorter must cube the signal. Thus, the DAC must be able to reproduce three times the highest baseband frequency. The performance could be even more severe depending on the PAs distortion.

One possible analog architecture from [5] that overcomes this speed disadvantage is shown in Figure 1-4. It takes advantage of the fact that analog components are themselves nonlinear and can be designed to be the inverse of the PA. Once this is accomplished, the predistortion scheme can be scaled to very high speeds. The particular circuit has a gain of less than one for small signals, and a gain of almost one for large signals, thus reversing the typical PA compression effect. For small inputs, the diode's impedance is large and approximately constant. When the input grows, the diode's impedance becomes disproportionately small at the signal peaks thanks to the exponential nonlinearity. This causes the average value of the output voltage and the diode bias current to increase. The average impedance becomes smaller, and the circuit gain moves closer to one.

Just as in the case of analog equalizers, it is difficult to fine tune analog predistorters. Digital feedback can be used to adjust biases and switch in appropriate components, but the architecture is still not nearly as flexible as its digital counterpart. However, its simplicity and speed make it worth investigating.

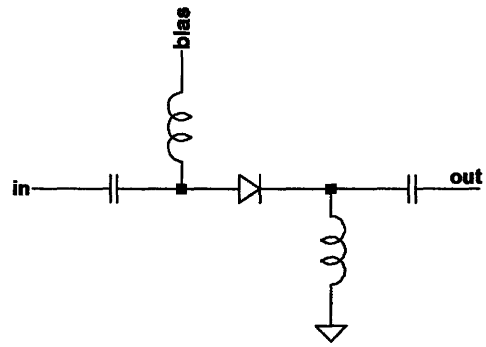
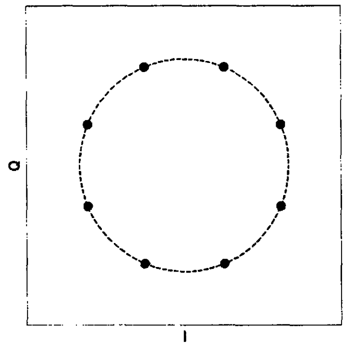


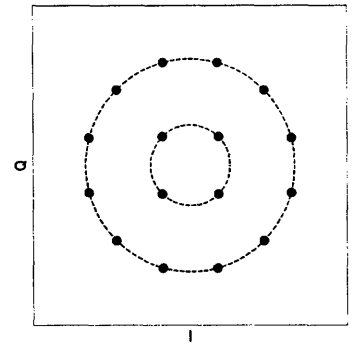
Figure 1-4: Diode based analog predistorter.

One way to solve the problem of distortion without analog or digital techniques is to use phase shift keying (PSK). If the symbols were not pulse shaped, then the 8-PSK constellation in Figure 1-5(a) would have constant power. The PA input power would never change, so there would be no nonlinear effects. This is only a partial solution, since PSK does not fully utilize the IQ plane. Most systems actually do use pulse shaping, so that even PSK would have changing power levels. However, PSK constellations still have a lower peak to average ratio (PAR) than their QAM equivalents. PSK symbols are less spread out than QAM symbols, so the average symbol magnitude for PSK is higher, making the PAR smaller. Lower PAR boosts PA efficiency since the signal is less likely to enter the PA's nonlinear regime for a given input power. The 12/4-QAM constellation, shown in Figure 1-5(b), is used in the high data rate satellite modes since it packs symbols more densely than 8-PSK, but has a lower PAR than 16-QAM.

Communication satellite systems requires solutions to both the dispersion and distortion problems. The simplest solutions would use the digital approach because of the powerful signal processing FPGAs already on board. The clock rates of the associated DACs are too slow to implement predistortion for a 240 MHz wide signal. By focusing on digital equalization, we took advantage of all the existing hardware and made large strides in improving the system.



(a) 8-PSK



(b) 12/4-QAM

Figure 1-5: 8-PSK and 12/4-QAM constellations.

1.5 Lab Setup

The lab setup used to support development of satellite systems is shown in Figure 1-6, and can also be used to investigate equalization. Each rack consists of a terminal on the left and satellite simulator on the right. The two are connected by a transmission line channel and can communicate at either the 3.1 GHz IF, or at the much higher RF. The additional upconverters, downconverters, and synthesizers to translate from IF to RF and back are not shown. The entire setup is called a Signaling Standards Test and Verification Environment (SSTVE).

To create a more realistic channel, a noise generator is inserted in the signal path. We used the UFX-EbNo 3100 from NoiseCom, which allows precise control over the SNR. The UFX-EbNo works by measuring the power of the carrier at the input and generating noise that has the same power level. The noise is then attenuated according to the desired SNR and finally combined with the carrier. This method eliminates any nonlinearities in the power measurement, since signal and noise start out at the same level. The attenuators could introduce errors as well, but it is easier to make them linear.

The lab setup gives us the capability of directly controlling the transmitted signal and then analyzing the received signal. This is accomplished using various "snapshot" memories along the signal chain. These memories can be both written to and read from a workstation. To send an arbitrary waveform, we simply need to write to the

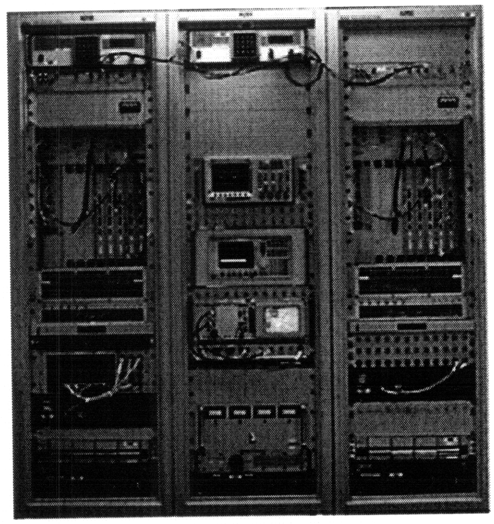


Figure 1-6: SSTVE lab setup.

memory on the transmit side that feeds into the baseband I and Q DACs. A similar memory on the receive side can be used to load the received waveform.

Chapter 2

Equalizer Background

Digital equalizers can be categorized as either linear or nonlinear. Linear equalizers remove ISI through filtering that compensates for a channel's non-flat frequency response. Nonlinear equalizers use some other form of signal processing to remove ISI that cannot be described with a transfer function. For both types of equalizers there are also many optimization methods. Different optimization methods require different degrees of knowledge about the channel. Equalizer architectures are compared mainly on the basis of minimizing signal errors. However, it is also important to consider the required computational resources.

2.1 Linear Filters

There are many different ways to choose the frequency response of a linear filter equalizer. The most intuitive method is to set the equalizer to be the exact inverse of the channel. The cascade of the channel and the equalizers would then be an ideal delay, which has the necessary flat frequency response. This technique is called zero forcing and although it theoretically eliminates ISI, it also greatly reduces the SNR so overall errors can actually increase. This is because a zero forcing filter will amplify any part of the spectrum with nulls, which are also the parts of the spectrum with the lowest SNR. For this reason, zero forcing is rarely used in practice [1]. The opposite of a zero forcing filter is the matched filter. A matched filter has the same amplitude response as the channel and the opposite phase. By attenuating or amplifying parts of the signal in proportion to their SNR, the matched filter maximizes the overall SNR. At the same time, it does not help the original problem of ISI. All linear equalizers fall somewhere in between the zero forcing and matched filter, and must choose in some sense between reducing ISI or noise.

2.1.1 Wiener Filters

The Wiener filter finds the optimum choice between the zero forcing and matched filters by accounting for both noise and ISI. Instead of trying to perfectly equalize the channel, the Wiener filter minimizes the overall difference between the received and desired signals. It has an especially favorable form if implemented with a finite impulse response (FIR) filter. In this case, the equations possess some very useful properties because they are linear in the filter taps.

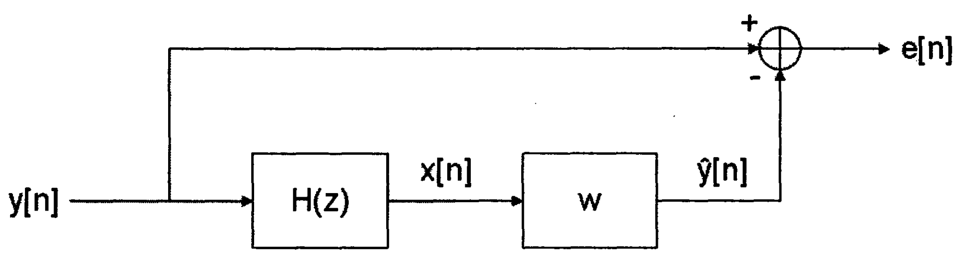


Figure 2-1: Block diagram for deriving the Wiener filter.

The setup for deriving the Wiener filter is shown in Figure 2-1. A discrete signal $y[n]$ is filtered by an unknown $H(z)$ to produce $x[n]$. The column vector $\mathbf{x}[n]$ represents the last N elements of $x[n]$. $x[n]$ then passes through an N tap FIR filter $\mathbf{w}$ of our choosing. The output of the FIR is $\hat{y}[n]$, which is an estimate of the original $y[n]$. Then we subtract $\hat{y}[n]$ from $y[n]$ to get the error $e[n]$. The goal of the filtering process is adjust $\mathbf{w}$ to minimize the error in some sense. For mathematical simplicity, usually the mean-squared error (MSE) is minimized.

If the FIR is causal, then its output at time n is simply $\mathbf{w}^T\mathbf{x}[n]$. We can then write an expression for the MSE, denoted by $\xi[n]$.

$$\begin{aligned}
\xi[n] &= E[e^2[n]] \\
&= E[(y[n] - \hat{y}[n])^2] \\
&= E[(y[n] - \mathbf{w}^T\mathbf{x}[n])^2] \\
&= E[y^2[n]] + \mathbf{w}^T E[\mathbf{x}[n]\mathbf{x}^T[n]]\,\mathbf{w} - 2\mathbf{w}^T E[\mathbf{x}[n]y[n]] \qquad (2.1)
\end{aligned}$$

If we assume all the processes are wide sense stationary, then ξ is no longer a function of time. We can then also write $E[\mathbf{x}[n]\mathbf{x}^T[n]]$ as a constant $N \times N$ autocorrelation matrix ϕ_{xx}, and $E[\mathbf{x}[n]y[n]]$ as a constant $N \times 1$ cross correlation vector ϕ_{xy}. ϕ_{xx} and ϕ_{xy} give all the information we need about how $y[n]$ becomes dispersed by $H(z)$, but only give partial information about $H(z)$ itself. If $y[n]$ is a narrow band signal that only excites a small part of $H(z)$, then the resulting FIR equalizer will compensate only that part.

To find the optimal filter $\mathbf{w}_{opt}$, we take the derivative of ξ with respect to each filter tap and set it to 0. Combining these equations together we get

$$\begin{aligned}
\frac{\partial \xi}{\partial \mathbf{w}} &= 2E[\mathbf{x}[n]\mathbf{x}^T[n]]\,\mathbf{w}_{opt} - 2E[\mathbf{x}[n]y[n]] = 0 \\
\frac{\partial \xi}{\partial \mathbf{w}} &= \phi_{xx}\,\mathbf{w}_{opt} - \phi_{xy} = 0 \\
\therefore \mathbf{w}_{opt} &= \phi_{xx}^{-1}\,\phi_{xy} \qquad (2.2)
\end{aligned}$$

2.1.2 Adaptive Algorithms

In spite of the simple form of equation 2.2, $\mathbf{w}_{opt}$ is difficult to calculate in practice because ϕ_{xx} and ϕ_{xy} are usually not known exactly. Adaptive FIR filters takes on these issues by approximating ϕ_{xx} and ϕ_{xy} and iteratively solving equation 2.2. Adaptive techniques are very powerful because the linearity of the Wiener approach guarantees that any locally optimum filter must also be globally optimum. Therefore, as long as we know how to step in the direction of $\mathbf{w}_{opt}$, we will eventually converge to it [6].

Ideally, during each step in the iteration process we would change the taps of the filter so that the MSE was reduced. This type of optimization is called gradient descent and would require adding some fraction of $-\frac{\partial \xi}{\partial w_i}$ to the i^{th} tap. Least mean squares (LMS) is one of the simplest adaptive filtering technique that approximates $\frac{\partial \xi}{\partial \mathbf{w}}$ in terms of known quantities. LMS is derived as follows, starting with the standard formula for $\frac{\partial \xi}{\partial \mathbf{w}}$

$$\begin{aligned} \frac{\partial \xi}{\partial \mathbf{w}} &= 2E[\mathbf{x}[n]\mathbf{x}^T[n]]\,\mathbf{w}_i - 2E[\mathbf{x}[n]y[n]] \\ &= -2E[\mathbf{x}[n]\,(y[n] - \mathbf{x}^T[n]\,\mathbf{w}_i)] \\ &= -2E[\mathbf{x}[n]e[n]] \\ &= -2\phi_{xe} \\ &\approx -2\mathbf{x}[n]e[n] \end{aligned} \tag{2.3}$$

The last step is the key to LMS. Only the current values of the input $\mathbf{x}[n]$ and the error $e[n]$ are needed to get a rough estimate of ϕ_{xe}. This level of accuracy is sufficient for convergence in most well behaved systems. The overall LMS iteration is then

$$\mathbf{w}_{i+1} = \mathbf{w}_i + 2\mu\mathbf{x}[n]e[n] \tag{2.4}$$

where μ is a step size factor that determines the convergence rate and stability. Equation 2.4 is straightforward to implement but takes many iterations to arrive at $\mathbf{w}_{opt}$. As the channel changes, the LMS algorithm will track those changes and adjust the filter accordingly. It is also convenient that as the FIR taps are adjusted, it always remains stable because it has no poles.

Infinite impulse response (IIR) use both poles and zeros to synthesize a transfer function, and could therefore be more efficient than FIRs. They are much more difficult to use, however, since the filter could easily become unstable during optimization if one of the poles were to leave the unit circle. In addition, the equations for minimizing MSE for IIR filters are not linear. Rather than the one global minimum we had for FIR filters, IIRs could have multiple local minimums. As a result, the usual strategies for searching for the best IIR filter usually have suboptimal results [7]. In some situations, IIR filters are still useful despite these disadvantages. For example, transfer functions with sharp peaks can be easily synthesized with some combination of second order sections. The same degree of sharpness would require many more taps in an FIR filter. If we were implementing our filter in analog, then IIR filters are definitely the best choice since poles are much easier to add than zeros in the analog domain. In our application, sharp peaks or dips are completely absent from the channel [6]. Given these problems with IIRs, we tend to focus on the flexible and successful FIR filter.

There are many different adaptive filtering methods that are much more sophisticated than LMS. One of the most common and powerful alternatives to LMS is recursive least squares (RLS). RLS chooses filter taps that minimize the squared errors for all previous times. This is reminiscent of curve fitting (hence the least squares in the name), where there are a large number of data points and a small number of parameters to adjust (the filter taps). The RLS approach is very different from LMS, which crudely estimates the MSE and then tries to minimize it. The recursive part of RLS comes from the way the taps are updated with the arrival of new data. RLS does not recompute the filter taps from scratch, but rather finds an adjustment factor for the current filter taps for each new data point.

For adaptive applications, a weighting factor can be used to make recent errors more important than older errors. RLS can then process data in real time, optimizing the filter taps to reflect the most recent shape of the channel. For calibration applications where the channel is static, RLS is still useful because it provides an efficient way to handle large amounts of data. We can process small segments of data at a time until the filter converges. Even with all the data available upfront, it would not be clear how to efficiently generate and invert the matrices to find the optimal Wiener filter using equation 2.2. Although RLS does require much more computation than LMS, it converges much faster and the resulting filter is usually better [6]. Figure 2-2 shows the convergence of LMS and RLS adaptive filters from a computer simulation.

We can speed up filter calibration on by running the RLS algorithm and terminating after the error flattens out.

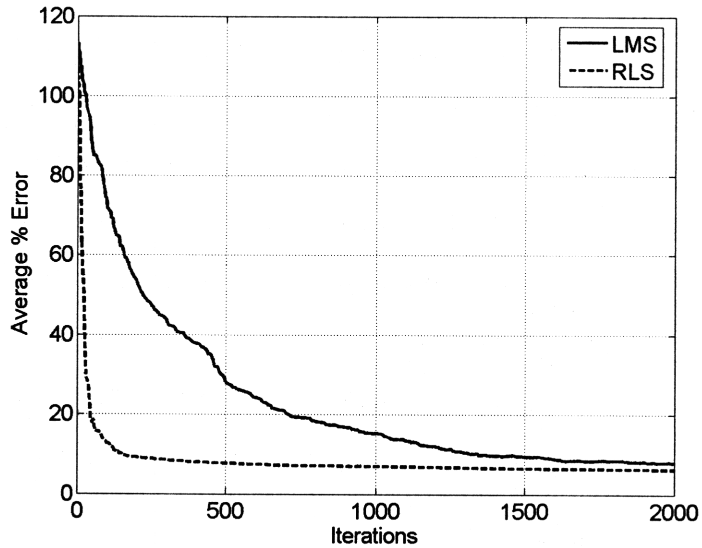


Figure 2-2: Convergence of LMS and RLS algorithms.

The main problem with all these adaptive algorithms is that they require knowing both the sent and received symbols. In a typical communications system the transmitted symbols are known only to the transmitter and received symbols is known only to the receiver. To overcome this difficulty, training sequences known to both transmitter and receiver can be used to periodically adjust the filter.

If the channel changes so slowly that training sequences are rarely needed, adaptive filtering can be very effective. However, if the channel is constantly changing, then the training sequences would waste a substantial amount of bandwidth. So called blind algorithms are ideal in these dynamic environment since they do not require training. Unlike the Wiener approach that uses second order statistics, blind algorithms use higher order statistics of the signal. However, blind algorithms are much more complex and converge more slowly than standard methods [8].

2.1.3 Fractionally Spaced Equalizers

The relationship between the symbol rate and the tap spacing is an important aspect of equalizer design. If there is one tap per symbol, then the equalizer bandwidth is exactly enough to correct the channel. A fractionally spaced equalizer (FSE) has multiple taps (usually two) per symbol, so it equalizes over a larger bandwidth than necessary. FSEs also require a higher clock rate.

Despite these disadvantages, most practical equalizers are FSEs, including the FIR filter currently implemented in the communication satellite hardware. The main strength of the FSE is that it is resistant to phase errors. A small phase shift would cause a symbol spaced equalizer to have artificially introduced nulls near the edges, but would have almost no effect on an FSE. Another advantage of FSEs is that the equalizer can filter out frequencies outside the band of interest. The FSE can therefore combine the equalization and matched filtering operations into one block [9].

2.2 Nonlinear Filters

All the equalizers discussed so far use linear filtering to remove ISI, which means they also amplify noise. Unlike linear filters, nonlinear filters do not have to compromise between zero forcing and matched filtering. Instead, some other form of processing is used to remove ISI without enhancing the noise. This can sometimes be an enormous advantage for channel's with deep nulls. In low noise cases, nonlinear filters can better approximate the inverse of a channel with a limited number of taps.

2.2.1 Decision Feedback Equalizers

The most common nonlinear filter is the decision feedback equalizer (DFE). The DFE works by attacking the root of ISI, the channel impulse response. ISI is caused by the impulse response having non-zero values at intervals of the symbol period. This means that a single symbol affects multiple sampling instances. An impulse response exhibiting ISI is shown in Figure 2-3. The parts of the impulse response that affect past sampling instances are called postcursors. The parts that affect future sampling instances are called precursors. Large errors can result from the combined effect of precursors and postcursors from many symbols on a particular sampling instance [9].

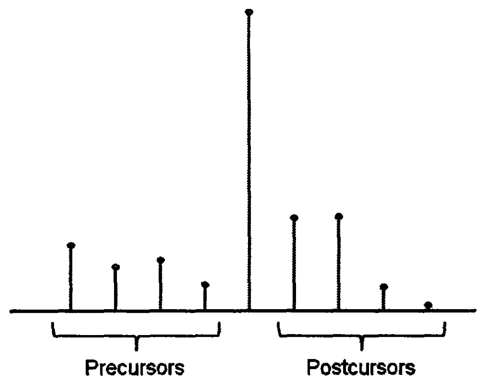


Figure 2-3: An impulse response that causes ISI.

The DFE attempts to estimate what interference is being added to the current symbol from past symbols (postcursors), and then subtracts out the interference term from the sampled value. Symbols that have not yet arrived at the receiver can also cause errors (precursors), but cannot be predicted by the DFE. The key to the DFE is that once it recognizes a symbol, it can use knowledge of the channel impulse response to predict how that symbol will interfere with soon to arrive symbols. This requires making a symbol decision, which is the nonlinear operation that makes the DFE a nonlinear filter. The block diagram of a DFE is shown in Figure 2-4.

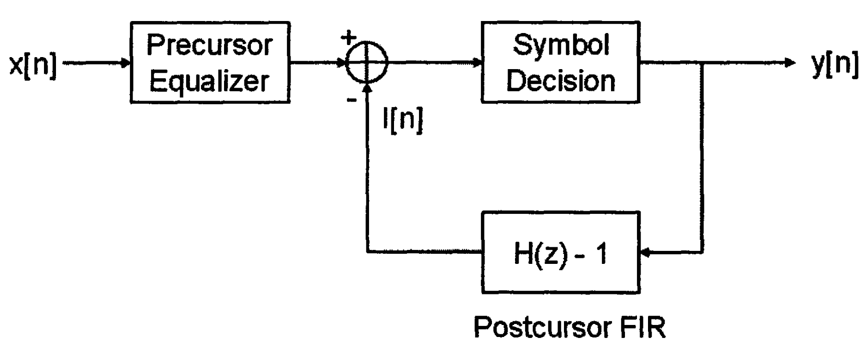


Figure 2-4: Block diagram of a DFE.

The precursor filter is a linear equalizer designed to correct for ISI from the precursors only. It is limited by the same noise and ISI trade-offs as the linear equalizers discussed earlier. After the precursor filter comes the DFE loop, which continuously makes symbol decisions. These decisions depend on the constellation of the communication mode. Before making a decision, the interference terms from the previous N symbols are subtracted. This operation perfectly removes ISI without any noise amplification, given accurate knowledge of the channel and N previous correct symbol decisions. Note that we call the postcursor part of the channel transfer function $H(z)$, so the postcursor FIR block must be $H(z) - 1$. The -1 term means that the DFE does not consider the current symbol when subtracting out ISI.

Even when noise amplification is not an issue, the DFE can perform extremely well by closely approximating the inverse of the channel. Consider what the DFE would look like if the symbol decision block was replaced with a gain of A. Using Black's formula to analyze the now linear negative feedback loop we get

$$\frac{A}{1 + A(H(z) - 1)} = \frac{1}{\frac{1}{A} + H(z) - 1} \tag{2.5}$$

If A is close to 1, then loop of the DFE becomes the inverse of the causal part of the channel. We would never actually do this because this inverse is most likely unstable and would amplify noise. However, this exercise does show how the DFE loop is a pseudo-inverse of the channel with a nonlinear element introduced to make the whole thing stable. In practice, this means that a DFE might better remove ISI from the channel than a linear equalizer even when noise is not an important factor [1], [9].

A DFE does not perform well if it somehow makes a wrong decision. It then calculates an incorrect interference term to subtract from the next symbol. It is possible that subtracting this incorrect term will cause the next decision to be wrong as well. This cycle could repeat for a some time, meaning that errors in DFEs usually occur in bursts. Most coding schemes, however, perform best when errors are randomly distributed. Therefore the advantages of using a DFE must be carefully weighed against the possible losses in coding performance. The DFE's ability to make decisions requires knowledge of the current communications mode. This could be a problem if the system architecture calls for separation of the equalizer from the other steps in demodulation. Furthermore, when advanced coding schemes are used, the symbol decision block can be very complex, making it difficult to apply feedback due to delays.

2.2.2 Tomlinson-Harashima Precoding

Tomlinson-Harashima precoding, or just precoding, is another type of nonlinear filter that has many of the advantages but almost none of the problems of DFEs. Whereas the DFE is designed for the receiver since it removes already added ISI terms, precoding is designed for the transmitter. It preemptively removes ISI that the postcursors will add during communication [10], [11].

During transmit equalization, all the symbol values are known with complete certainty. This means that precoding does not require a symbol decision block, and can completely avoid the clustering errors of DFEs. A block diagram of precoding is shown in Figure 2-5.

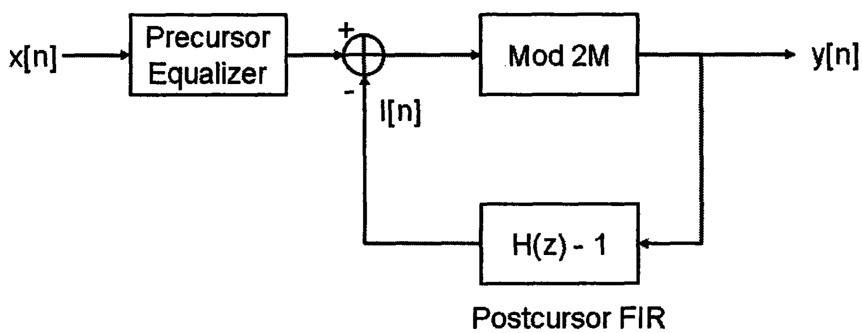


Figure 2-5: Block diagram of a Tomlinson-Harashima Precoder.

There are a lot of similarities between the precoding and DFE block diagrams. In both cases, precursors are filtered out using a standard FIR equalizer. Postcursors are removed, preemptively this time, in a nonlinear loop. The main difference between the DFE and precoding is the choice of nonlinear block. Without the modulus operator, the precoding equalizer would be unstable since it would be implementing the inverse of the postcursor part of the channel, $\frac{1}{H(z)}$. Unless $H(z)$ happens to be minimum phase, the inverse will be unstable. Even an almost unstable filter would introduce a lot of sharp peaks, which are highly undesirable for a transmitter with limited peak output power.

The modulus operator keeps the loop stable by limiting the output value from becoming to high. Essentially, the modulus block checks to see if the input is within a certain range, from $-M$ to $+M$. If it is, the signal passes through unchanged. If not, an integer multiple of $2M$ is added. The integer multiple, b, is unique because once the sum of the input and $b \cdot 2M$ is within the $-M$ to $+M$ range, adding or subtracting another $2M$ will by definition move the sum out of range.

The output of the precoder $y[n]$ consists of the original input $x[n]$, a subtracted $I[n]$ term to compensate for ISI, and an added $b \cdot 2M$ term to keep the output within range. If the channel is well approximated, then the ISI will be exactly as predicted, and $I[n]$ will be added to the signal by the time it reaches the receiver. To recover the original $x[n]$, we still need to remove the term added by the modulus block. Fortunately, another modulus block at the receiver does exactly that. Unless b was zero (in which case neither modulus blocks did anything), after the ISI is removed the signal will be out of the $-M$ to $+M$ range. The second modulus operation will subtract $b \cdot 2M$ to move the signal into range. The only thing left is our original $x[n]$. The entire process is illustrated in Figure 2-6.

Amazingly, the Tomlinson-Harashima precoder compensates for ISI with the help of a modulus block at the receiver input, with no possibility of making symbol decision errors and without any knowledge of the mode. However, when imperfect equalization and noise are considered, precoding does reveal one problem. Suppose that our $2M$ range were the same as the range for $x[n]$, which means that the transmitter would never have to handle signals larger than it did without precoding. If $x[n] - I[n]$ starts out right at the border of this range and the $I[n]$ term is extremely small, even a a little bit of noise or unanticipated ISI will move it out of range. Then the receiver modulus will shift it back into range by subtracting $2M$ and return almost the negative of the original $x[n]$. These kinds of errors can only be avoided by making the $-M$ to $+M$ range of the transmitter larger than the range of $x[n]$. This way, $y[n]$

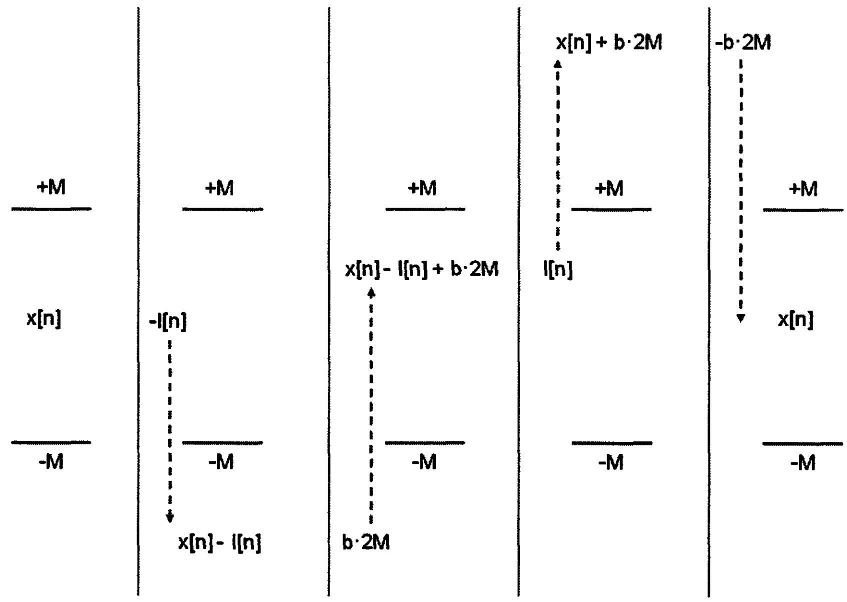


Figure 2-6: Detailed operation of a Tomlinson-Harashima Precoder.

will be close to the border of the range only if $I[n]$ term is somewhat large. In this case, it is unlikely that noise will make a difference since it would have to be larger than the already large $I[n]$. This fix comes at the cost of increasing the transmitter's peak power, which could result in efficiency losses. However, it is likely that only a small increase is necessary for useful precoding.

Chapter 3

System Design

3.1 Performance Metrics

The problems of spectral regrowth and interference with adjacent channels are outside the scope of what an equalizer can improve. Instead, we focus on bit error rate (BER), which is one of the most useful ways to measure the quality of a communication link. BER depends on the constellation and SNR. The closer the symbols are in the constellation, the more susceptible the system is to noise. SNR is not always the best indicator of the noise level, since it is a function of both noise density and symbol rate. It is also not clear how exactly the noise bandwidth is defined. The ratio of energy per symbol to noise density, written $\frac{E_s}{N_0}$, removes most of the ambiguities of SNR. $\frac{E_s}{N_0}$ uses energy rather than power ratios, so it is completely independent of symbol rate. Since we only require noise density, the noise bandwidth becomes irrelevant. A typical BER verses $\frac{E_s}{N_0}$ plot for the 12/4-QAM constellation is shown in Figure 3-1.

The BER plot provides a good way of comparing constellations and evaluating noise/symbol rate/error rate trade-offs. The problem with relying on BER by itself is that it is difficult to measure at low noise levels. A huge number of symbols need to be processed to accurately calculate the BER. For each symbol, let the probability of error be P_e. We assume that the constellation uses Gray coding and that the noise is small enough that a single symbol error corresponds to a single bit error. The number of bit errors in N independent symbols would have a binomial distribution.

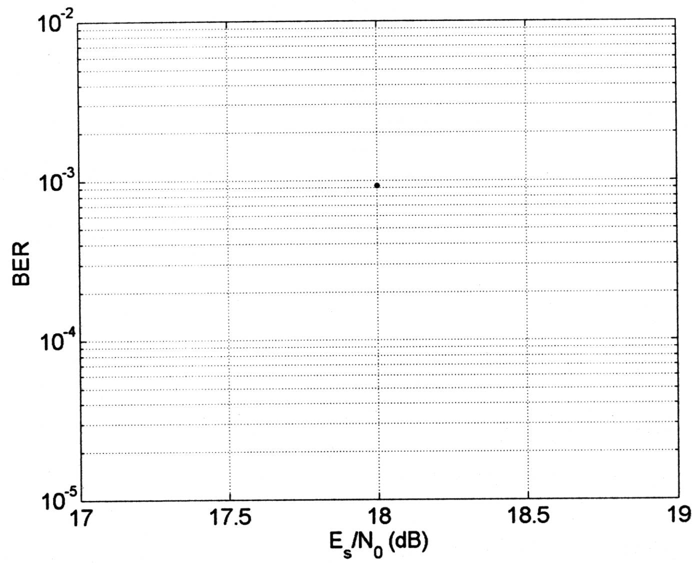


Figure 3-1: Plot of BER vs $\frac{E_s}{N_0}$ for 12/4-QAM.

Therefore the mean of the measured BER would be P_e and variance would be $\frac{P_e - P_e^2}{N}$. To determine how large N need to be for a given confidence in our measurement, we will approximate the distribution as Gaussian. To measure the BER within 15% with 68% confidence (one standard deviations) we require that

$$1 \cdot \frac{\sqrt{P_e - P_e^2}}{\sqrt{N}} = 0.15 P_e \tag{3.1}$$

Since P_e is very small we can ignore the P_e^2 term and find that

$$N \approx \left(\frac{1}{0.15}\right)^2 \frac{1}{P_e} \tag{3.2}$$

For a reasonably small P_e of 10^{-5}, we would need to test about $4.4 \cdot 10^6$ symbols. This kind of measurement is quite time consuming unless we use specialized machines that process symbols in real time and count the errors. However, we would not be able to easily use our own system of snapshot memories, which only store about 10,000 symbols at a time.

Despite the importance of BER, we need some other metric to use when developing equalizers. Error vector magnitude (EVM) is a popular choice because it is easy to compute and ultimately leads to good performance. EVM is defined as the average distance between the received symbol and the ideal symbol on the IQ plane, as shown in Figure 3-2. This average error is divided by the magnitude of the symbol in the constellation that is farthest from the center of the constellation. This scale factor makes it easy to compare the EVMs of systems with different peak energies. Of course, BER is still the final test of link quality, but we do not need to use it all the time.

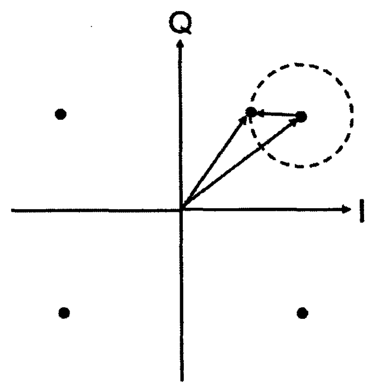


Figure 3-2: Definition of EVM.

It would be very convenient to be able to measure an improvement in EVM due to better equalization and quickly relate it to an improvement in BER. Fortunately, we can do this by noting that EVM can quantify errors from noise just as well as it quantifies errors from dispersion. There is no reason to treat these two components separately. So if we improve our equalization so that the EVM becomes smaller by 1%, then this will have roughly the same effect on BER as reducing the noise so that the EVM becomes smaller by 1%. Now the only piece missing is some way of translating from EVM to more familiar noise measurement units. A simulation was used to generate a plot for describing EVM as a function of $\frac{E_s}{N_0}$. Using this plot, a change in EVM can be related to a change in $\frac{E_s}{N_0}$, which can finally be related to a change in BER. The net result is the plot of BER verses EVM, shown in Figure 3-3. It should be noted that complete satellite communications systems use coding so that at 10 dB $\frac{E_s}{N_0}$, the packet error rate (PER) for a 12/4-QAM constellation can be close to 10^{-4} even though the BER is close 10^{-1}.

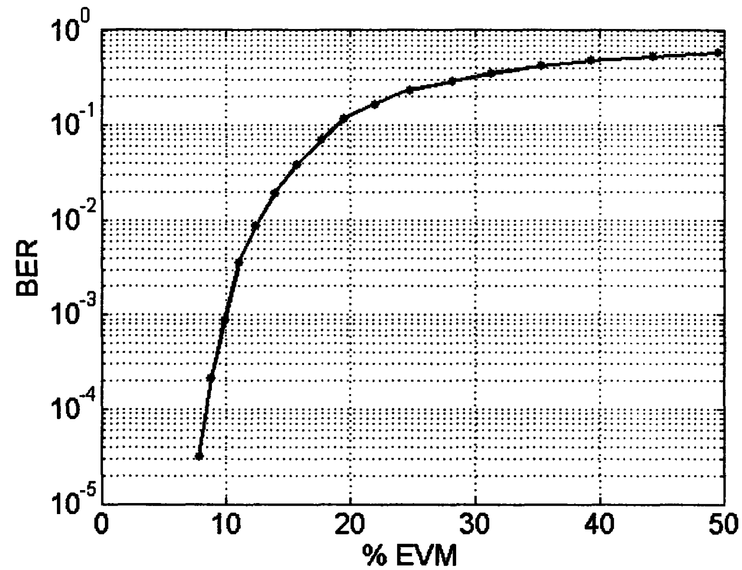


Figure 3-3: Plot of BER verses EVM for 12/4-QAM.

3.2 IF Calibration Procedure

The first method we developed for calibrating the equalizer was designed to work only for IF communication at 3.1 GHz. This method can never completely compensate the channel because it does not take into account the high frequency upconversion and downconversion stages. However, IF calibration still serves as an invaluable tool for learning about various equalizer trade-offs and design issues. The relatively low frequencies involved make IF calibration much more straightforward than RF calibration (see Chapter 4), and this simplicity makes the process extremely fast. With full automation, the entire process takes only a few minutes, making it easy to gather data.

The calibration procedure is split into two steps, followed by performance testing. The first step is to calibrate the transmit equalizer, and the second step is to use the equalized transmitter to calibrate the receive equalizer. The entire procedure is controlled by an external workstation, as shown in Figure 3-4. We use the workstation to setup all the required instrumentation through a General Purpose Interface Bus (GPIB) to Internet Protocol (IP) converter, control the snapshot memories, and process the data in order to synthesize an equalizer.

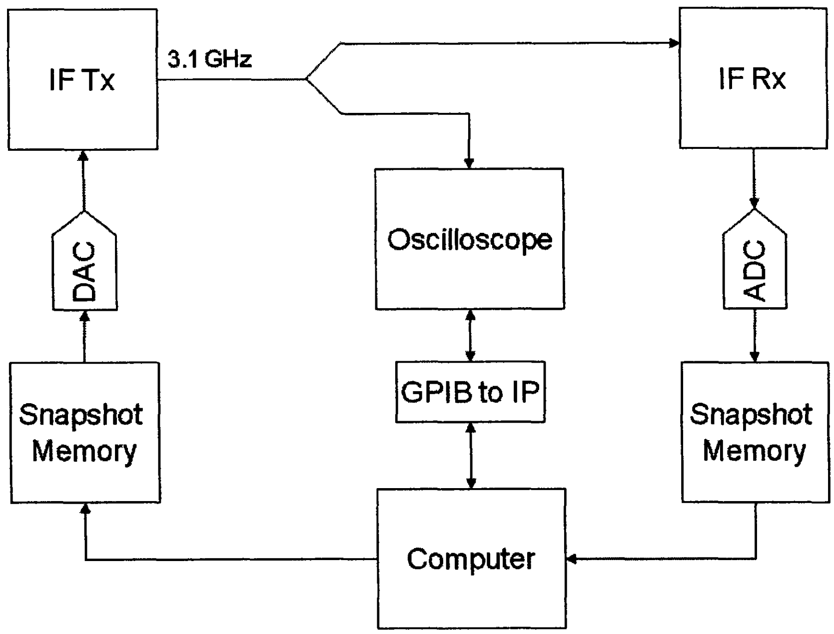


Figure 3-4: Setup for IF equalizer calibration.

To start, we connect the output of the transmitter to a high-speed sampling oscilloscope. This key step is only possible at IF, since even the fastest oscilloscope cannot sample the RF transmitter's output frequency. The transmitter's 10 MHz reference clock is fed into the oscilloscope's 10 MHz reference input so that the two clocks are synchronized. Otherwise, over the course of reading the data the clocks will drift apart, leading to gradually increasing phase errors. The oscilloscope is triggered from a signal that indicates the start of a hop. The communications waveform allows some settling time between when the hop starts and when symbols are sent. This guarantees that the oscilloscope always starts capturing before the first symbol. Once all the data is captured, it is sent to the workstation. Here we perform post-processing that removes the "dead time" before the first symbol, and changes the sampling rate to match the 360 MHz clock in the transmitter. Now the transmitted and captured data are ready for direct sample to sample comparison. These two sample streams can be fed into an adaptive algorithm to arrive at an optimal equalizer. Usually only a few thousand samples are needed for convergence.

In addition to outputting an equalizer at this stage, we can also estimate the

EVM of the equalized transmitter and see how close it is to ideal. Of course, that assumes that the oscilloscope used for measurement is nearly ideal itself. We verified this assumption by measuring the response of the oscilloscope to a range of pure sine waves from a calibrated signal generator. The amplitude response of the oscilloscope, shown in Figure 3-5 varies by about 0.1 dB. This is small enough to accurately capture the variations of approximately 1 dB in the channel.

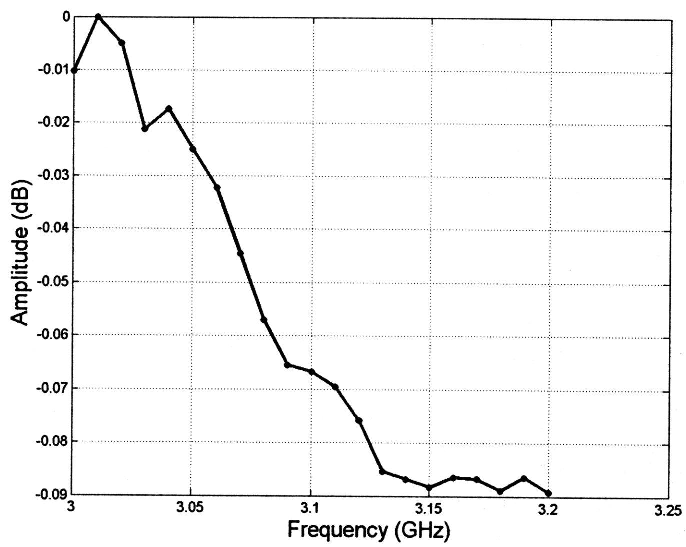


Figure 3-5: Oscilloscope's amplitude response.

The next step is to calibrate the receive equalizer. We do this by sending data from the equalized transmitter to the receiver, and then capturing the results from the snapshot memory. The sent and captured data are already at the same sampling rates and aligned in time since the clocks of the transmitter and receiver are synchronized by other means. Just as in the first step, two symbol streams are fed into an adaptive algorithm to arrive at an optimal equalizer. This process would work perfectly if the equalized transmitter were ideal. We will analyze what happens when imperfect transmitters are used in the next section.

We can use a procedure similar to receiver calibration to determine the frequency response of the channel. This is useful for simulation, allowing us to rapidly test equalizers in software before trying them on the real system. The only difference is that the transmitter is unequalized (to capture the entire channel) and the two symbol stream inputs of the adaptive algorithm are reversed. Now the adaptive algorithm will try to find a filter that recreates the same effects as the channel, as opposed to a filter that undoes them. An example of a channel frequency response is shown in Figure 3-6. Only the part of the channel used for communication is shown.

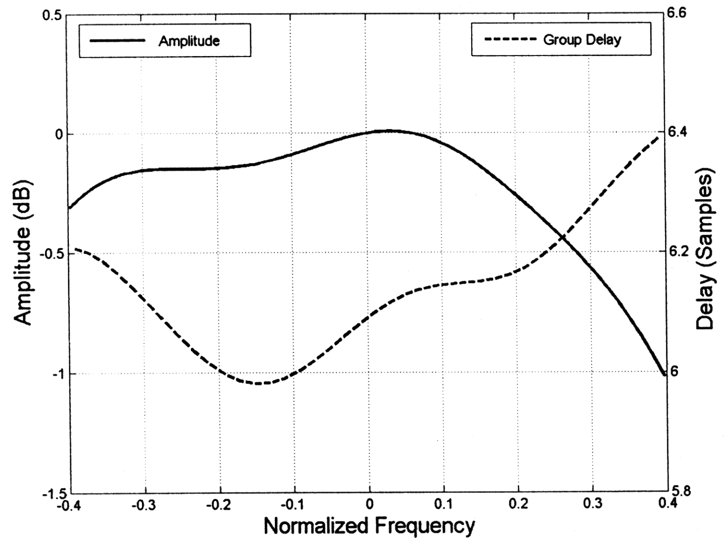


Figure 3-6: Communication channel frequency response.

Once we have both the transmit and receive equalizer we can evaluate the performance of the overall system. This involves sending various waveforms from transmitter to receiver, and measuring the EVM. These measurements can be repeated many times automatically to determine the measurement variance. In addition, the noise level for each measurement can be changed so that we can observe how EVM varies with noise.

3.3 Separation of Transmitter and Receiver

Both the transmitter and receiver require their own equalizers. Linear filtering is commutative, which means that there are many ways to split the burden of compensating the channel that would be equivalent from the point of view of reducing ISI. In practice, many other factors such as power efficiency or ease of use need to be considered to determine how to distribute equalization.

To optimize the power efficiency of the transmitter, the peak to average ratio (PAR) of the signal should be minimized. High PAR would force the PA to accommodate large spikes in the waveform that might push into its nonlinear regime. As a result the amplifier's bias would have to be increased, which would cause efficiency losses. PAR increases with higher ISI because ISI creates the possibility of precursors and postcursors from many symbols adding up constructively to create a large peak. The average power, however, does not change with ISI since destructive interference is just as likely. Therefore, we would like to feed the PA a signal with no ISI and in turn a low PAR. This requires that the transmit equalizer compensates for transmitter dispersion only. If this transmit equalizer also compensated for the receiver to any extent, the ISI of the PA's input signal would be unnecessarily high. Of course, this ISI would be canceled at the receiver, but at the cost a of power efficiency. Fortunately, practical consideration lead to the same conclusion as optimizing power efficiency. Satellite terminals and satellites will be manufactured separately and must perform equally well in any combination. Therefore, we do not want a calibration procedure that maximizes the performance of a particular transmitter/receiver pair. We again arrive at the same conclusion that the transmit equalizer should compensate for the transmitter only, which leaves the receiver equalizer to compensate for the receiver only.

To pursue the goal of separating equalization evenly between the transmitter and receiver, we would ideally calibrate many systems and then test them in many combinations. If on average all the system combinations worked equally well, then we could conclude that our calibration procedure correctly separated equalization. In our current lab setup is it difficult to test more than one terminal/satellite combination, let alone the many tests required for statistical averaging. Since we are unable to gather a lot of empirical evidence, we need some theoretical framework to evaluate how well the calibration procedure works in general.

To begin our analysis, we will take a closer look at EVM. We will ignore the details of demodulation since they complicate the mathematics and have a negligible impact on the EVM. By definition, for a transmitted signal $y[n]$, received signal $x[n]$, and difference $e[n] = x[n] - y[n]$, the EVM is proportional to the average of the absolute value of the difference, or more compactly

$$EVM \propto E[|e[n]|] \tag{3.3}$$

For small variations of $e[n]$, the mean of the absolute error will be close to the square root of the mean squared error, so we can write

$$EVM^2 \propto E[|e[n]|^2] \tag{3.4}$$

If we assume that the transmitted signal was filtered by the channel response $h[n]$, then we can further simplify to get

$$EVM^2 \propto E[|y[n] * h[n] - y[n]|^2] = E[|(h[n] - \delta[n]) * y[n]|^2] \tag{3.5}$$

Since $y[n]$ is a sequence of pseudorandom symbols, its autocorrelation is similar to white noise. This means that its power spectral density is approximately constant. To take advantage of this simplification, we move to the power spectral density domain. We can then use a form of Parseval's theorem that equates mean squared error to the area under the power spectral density curve.

$$\begin{aligned} EVM^2 \propto E[|e[n]|^2] &= \int_{-\pi}^{\pi} S_{ee}(z)dz \\ &= \int_{-\pi}^{\pi} S_{yy}(z) \cdot |H(z) - 1|^2 dz \\ &= \sigma_{yy}^2 \int_{-\pi}^{\pi} |H(z) - 1|^2 dz \end{aligned} \tag{3.6}$$

The only non-constant term left is $\int_{-\pi}^{\pi} |H(z) - 1|^2 dz$, which is simply the total squared difference between $H(z)$ and an ideal channel with unity gain. Using equation 3.6, we can easily predict the EVM given the channel transfer function. Now we can begin to see what happens to the overall EVM when the transmitter and receiver channels are combined together. There is an infinite range of possible interactions between the two transfer functions. They could destructively interfere so that the EVM of the total system is much less than the sum of the EVMs of each part. But it is also possible that they interfere constructively, so that the total EVM becomes much larger than the sum of the EVMs of each part. Any outcome in between these two extremes is possible, so it might seem that our theory cannot make any useful predictions without more information. However, if we restrict the type of channels we consider, we can come up with a useful heuristic.

From experimental evidence, we know that the channel dispersion is not severe enough to cause more than 15% EVM. This number is actually small enough that even with perfect constructive interference, the channel EVMs add almost linearly. This is because the constructive interference becomes large only when the channel gain is much more or less than unity, but our gain deviations are small. Destructive interference is far more noticeable. We can then place a practical upper bound on the EVM of two combined channels, summarized as

$$\begin{aligned} H_1(z) &\Rightarrow EVM_1 \\ H_2(z) &\Rightarrow EVM_2 \\ H_3(z) = H_1(z)H_2(z) &\Rightarrow EVM_3 \leq EVM_1 + EVM_2 \end{aligned} \tag{3.7}$$

We can now look at what happens when imperfect instruments are used during calibration. Let the transfer function of the equalized transmitter be $T(z)$, with a measured EVM of EVM_T. EVM_T is on the order of a few percent, partially due to quantization errors, phase noise, nonlinear distortion, and some uncompensated dispersion. In an ideal calibration process, the equalized receiver transfer function would be $R_I(z)$. Its EVM, denoted as EVM_{R_I}, would be similar in composition to EVM_T.

When the imperfect transmitter is used in the real calibration process, the equalized receiver will have a different transfer function, $R(z)$. $R(z)$ must compensate for the residual dispersion in the transmitter, so it will be

$$R(z) = R_I(z)\frac{1}{T(z)} \tag{3.8}$$

Using equation 3.6 and the fact that $T(z)$ only deviates slightly from unity, it can be shown that $\text{EVM}_{\frac{1}{T}}$ is approximately equal to EVM_T. We are now ready to apply equation 3.7. The EVM of the receiver is

$$EVM_R \leq EVM_{R_I} + EVM_T \tag{3.9}$$

This means that if we can measure EVM_T, we can estimate the drop in receiver performance due to an imperfect transmitter. We know that EVM_T is fairly small in practice, especially if only a part of it is due to residual dispersion. With a small EVM_T, equation 3.9 tells us that the receive equalizer will perform well.

3.4 Noise Effects

Initial tests of the equalizer involved signals with very little noise. These tests indicated how well a given filter removed ISI, but said nothing about the potential problem of noise amplification. A white noise generator was used to add noise during calibration and performance testing. During equalizer calibration, adding different noise levels results in different filter shapes. Each one of these filters is optimized to reduce ISI and noise in the right proportion to achieve a minimum EVM. The optimum filter at 2 dB $\frac{E_s}{N_0}$ would look different from the optimum filter at 10 dB $\frac{E_s}{N_0}$. Therefore, it is important to calibrate the equalizer in the presence of realistic noise levels.

During performance testing, adding noise allows us to measure the equalizer's noise handling ability. The high data rate modes are actually not the most useful indicators in this case, since the errors will almost always have comparable components of ISI and noise. Unless the equalizer has terrible noise amplification, the high data rate modes will always improve after equalization. However, narrow band modes have very little ISI, making it immediately obvious if noise is being amplified and degrading the

EVM.

When addressing noise issues it is also important to keep in mind where matched filtering occurs in the signal chain. Since our system has fractionally spaced taps, the equalizer could perform some matched filtering. This approach would present a major problem to our general goal of having a single equalizer optimize performance for all the modes. Each mode requires a matched filter with a different bandwidth, and ignoring these differences could cause noise amplification. In a communication satellite's channel, the high frequencies are usually boosted. For a low data rate mode, these frequencies are not even used, so the only result would be amplifying noise. Fortunately, matched filtering is combined with demodulation. Demodulation is mode dependent, so the matched filter shape is selected appropriately.

3.5 Software Filters

In the current version of the SSTVE, the equalizer is a 13 tap FIR filter with adjustable coefficients implemented on an FGPA. The FPGA could be reprogrammed to extend the filter beyond 13 taps or to try a completely different architecture altogether. However, this is not a simple task and may not even be possible in some cases because of the limited number of multipliers available in hardware. The system allows an easier alternative for equalizer experimentation that greatly reduces development time. This alternative comes from the digital back-end, which has the ability to fill the baseband with an arbitrary signal, not just a sequence of pre-specified symbols. This signal is read from a memory that holds about 20,000, 16-bit values that are fed into a 360 MHz DAC. So, instead of requiring that equalization be performed in real time, we can do arbitrarily complex processing in software, and then load the prefiltered signal into memory. The net effect is that any kind of transmit equalizer can be implemented in software, but the performance can be measured on the real system. Similarly, on the receive side, an ADC loads the signal into memory. Software post-processing can then simulate a receive equalizer.

One important consideration when implementing software equalizers is the precision of the mathematical operations. The currently used FPGA uses 12 to 16-bit fixed point arithmetic. Probably any practical means of implementing an equalizer at the required data rates will have similar precision, since floating point is much more computationally intensive. The software, however, does use 32-bit floating point arithmetic. Therefore an FIR equalizer implemented in software will always perform better than the same equalizer implemented in an FPGA due to the reduced quantization noise. Ignoring roundoff errors could lead to favoring more complex architectures that would have poor performance if actually built. The greater number of mathematical operations would lead to larger roundoff errors and could eliminate most of the perceived advantages. To avoid this problem, software filters can be forced to use fixed point precision.

The potential problem of too much precision and power in software filters can be used as an advantage in some situations. When we wish to create as high quality a transmitter as possible in order to calibrate the receiver, the use of an equalizer that cannot be practically realized is not a concern. On the contrary, the better the equalizer the more ideal the transmitter. A more nearly perfect transmitter will ultimately lead to superior calibration of the receiver.

Chapter 4

RF Calibration

4.1 Introduction

The IF calibration procedure is a very useful tool but is incapable of fully correcting dispersion in the system. Since the transmitter is calibrated at the 3.1 GHz output, the imperfection in the RF stage is completely ignored. This will result in an transmitter RF stage that is entirely uncorrected, or corrected by the receive equalizer. Although the latter option might seem to produce reasonable results, it violates the principle of separating transmitter and receiver equalization. The RF calibration procedure is designed to allow the transmitter to compensate its own RF stage.

The reason the IF calibration procedure is inadequate is because oscilloscopes are not fast enough for data collection at the highest RF carrier frequency of 44 GHz. Instead of directly connecting to an oscilloscope, the RF calibration procedure uses a spectrum analyzer as a tuned receiver to first downconvert the test signal to 321.4 MHz. The tuned receiver mode is enabled by setting the frequency span to zero. In this mode, the spectrum analyzer is fixed at one frequency at a time instead of sweeping across a range. The downconverted signal is also provided as an output. The main shortcoming of using the spectrum analyzer's downconverter is that it is narrow band, less than 40 MHz. However, we need to be able to equalize over 240 MHz of bandwidth.

The RF equalization procedure solves this problem by taking several measurements, looking at only a one small part of the total 240 MHz each time. The results from each measurement are combined together to synthesize an equalizer that corrects for the entire bandwidth.

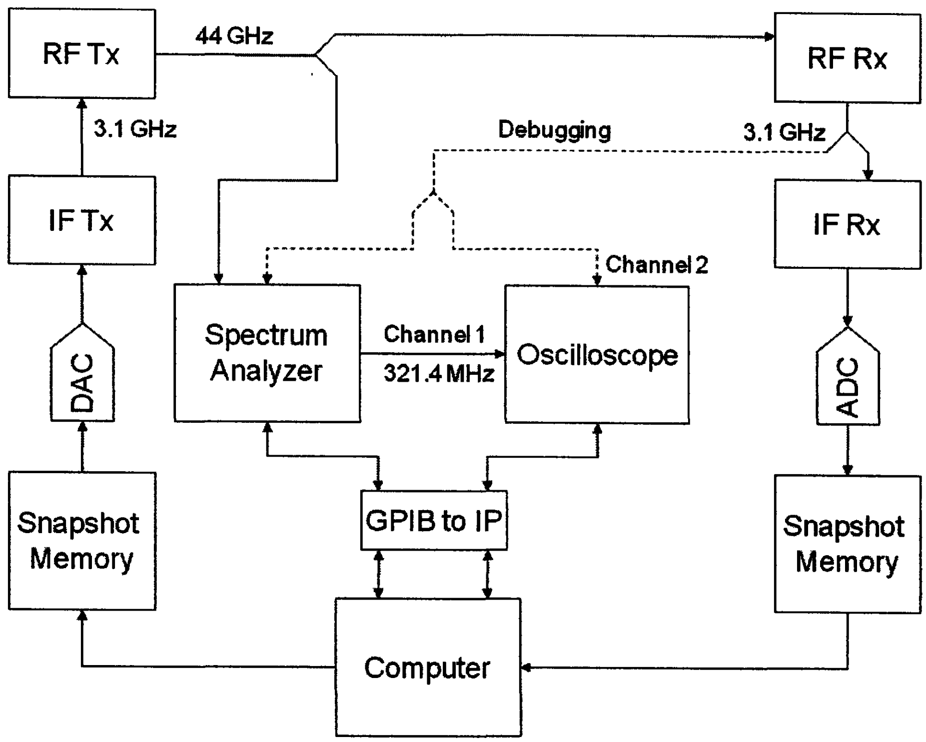


Figure 4-1: Setup for RF equalizer calibration.

The setup for developing the RF calibration procedure is shown in Figure 4-1. The final goal is for the spectrum analyzer's input to come from the RF transmitter output. However, it would be impossible to verify that the equalizer was working correctly at this stage. This capability is exactly what the RF calibration procedure is replacing. For development, the dotted connection is used instead. The signal is upconverted to 44 GHz and then downconverted back to 3.1 GHz, finally going to both the oscilloscope and the spectrum analyzer. With this setup, the spectrum analyzer can still be used for RF calibration. The key addition is that now the oscilloscope allows us to generate an equalizer using the much simpler IF procedure. Since IF calibration does not have to combine narrow band measurements, the equalizer it outputs can be considered the best possible equalizer. We can therefore establish an upper bound on performance for RF calibration and try to get as close to this bound as possible. Without this special debugging setup, it would difficult to measure how well the RF procedure works.

The lower bound on acceptable performance can be found by using only the IF calibration procedure for the RF channel. The IF stage of the transmitter will be equalized, but the RF stage will be ignored. If the RF procedure does not beat this lower bound, then we can conclude that equalizing the IF stage well is more important than poorly equalizing both the IF and RF stages.

4.2 Measuring the Frequency Response

The IF calibration procedure compares transmitted and received data to create an equalizer. This method is compatible with many existing algorithms, but is not the only approach. Another method might involve directly measuring the frequency response of the channel. Using this frequency response as a filter, an artificial set of sent and received data could be generated. Then, the same standard algorithms could be used to create an equalizer.

The frequency response method has more steps than the data approach, so it is more likely to accumulate unnecessary errors, but has one feature that makes it an excellent fit for RF calibration. The frequency response method can easily be spread out over many measurements. Each amplitude and phase measurement is independent of all the others, and the main task is keeping them consistent. The task of sending small bands of data and then stitching them together is much more challenging. The many measurements also make the frequency response method more redundant.

The frequency response is measured by sending sine wave "probes" at a one frequency at a time, and measuring how the sine wave's amplitude and phase change. To span the entire 240 MHz band of interest, the spectrum analyzer is tuned to downconvert one small part of the total at a time, called a segment. The bandwidth of a segment is less than or equal to the bandwidth of the downconverter. Many probes (usually between 10 and 20) are sent to find the response of this segment. Once we have all the necessary information, the spectrum analyzer is retuned to downconvert the next segment. The process is continued until we have gone through all the segments and can determine the total frequency response.

4.3 Measuring Amplitude

Measuring the amplitude of a probe is relatively simple since amplitude is invariant to random phase shifts and delays. Averaging can be used to improve accuracy. However, there are still several effects of the spectrum analyzer that need to be taken into account.

Every time the spectrum analyzer is retuned to a different segment, there is no guarantee that the downconversion has not changed. Any small change would make the amplitude relationship between segments inaccurate. To correct for this problem, there is a forced overlap between segments. Thus, the frequency of the last probe in a segment is the same as the frequency of the first probe of the next segment. A multiplying factor is introduced to make the amplitudes the same, compensating for any variations in the spectrum analyzer. Thus, amplitude continuity is guaranteed.

Another problem is dispersion introduced by the downconverter itself. The easiest way to minimize these errors is to use the downconverter over a narrow band. Of the full 40 MHz, we only use middle 10 or 20 MHz. The frequency response of the downconverter is much flatter over these regions. In addition we can improve the downconverter response by turning off the preselector, which is a narrow band filter at the front end that blocks images. The preselector is not essential in this application because we have control over the frequency content of the signal.

4.4 Measuring Phase

Measuring the phase involves all the same techniques as measuring amplitude, but also has many other difficulties to overcome. The main problem comes from the way the downconverted data is captured in the oscilloscope. The start of capturing is triggered in such a way that there is guaranteed to be some dead time before the symbols start. Removing this dead time is simple in IF calibration since it is only done once. In the worst case, a bulk phase shift is introduced into the signal, which does not matter since no particular phase shift is assumed during demodulation. When this same method is repeated many times for RF calibration, the result is that different phase shifts are introduced throughout the measured frequency response.

The usual method for performing dead time removal is to do a cross-correlation between the received and sent signals. A peak occurs in the cross-correlation when the signals are aligned. If we applied this method for each segment in RF calibration, we would be removing the dead time and any other delays caused by the channel response. Thus, we would be destroying part of the information we set out to gather.

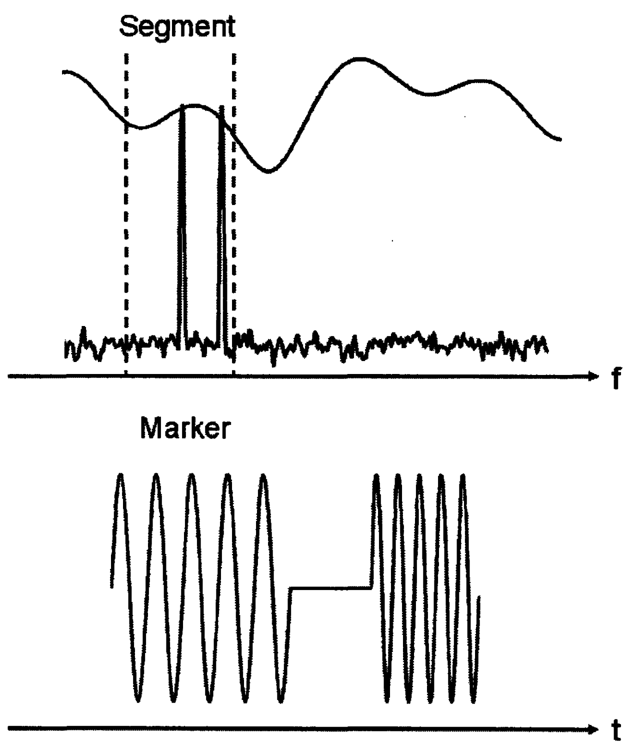


Figure 4-2: Measurement probe in frequency and time domain.

To solve the problem of removing dead time, we change the probe so that it contains two sine waves. These waves are separated in frequency and time, as shown in Figure 4-2. The first frequency is the same as before, gradually changing until the entire segment has been covered. The second frequency is called the "marker," and always has the same frequency as the center frequency of the downconverter. Dead time is still removed by finding the peak in the cross-correlation, but now we only use the markers of the sent and received signals. Since the marker is always at the same frequency for a segment, the channel delay of the marker will be constant. Therefore the amount of dead time removed will not depend on the channel frequency response. The marker makes phase measurements within a segment consistent by providing a constant reference.

We have made phase measurements consistent across a single segment by removing the dead time in a channel independent way, but we also have to make sure the phase is consistent between segments. Unless special precautions are taken, the phase would change drastically on the border between segments. This is because every time the spectrum analyzer is retuned, the phase between the spectrum analyzer and system clocks changes randomly. Synchronizing the clocks does not solve this problem, and only keeps the phase difference from drifting. In the same way that we forced amplitude continuity by overlapping segments, we can force phase continuity as well.

Chapter 5

Results

5.1 Mode Performance

One of the main questions initially posed at Lincoln Laboratory was whether or not equalization could degrade the EVM for some communication modes. Although this might seem counterintuitive, since equalizing a channel theoretically improves performance for any waveform, the data did suggest that some modes performed worse after equalization. The initial hypothesis was that since the widest band waveform was used for calibration of the equalizer, the EVM of the other waveforms was not considered directly and might even increase. More specifically, the equalizer could potentially improve a large part of the channel at the expense of a narrow section. This narrow section is mostly irrelevant for the wide band mode, but could have a serious cost for other modes. Automation and control of the calibration process allowed highly repeatable testing of this theory, ultimately leading to a different conclusion about the equalizer's effect on various modes.

The first observation found from repeated testing was that the EVM measurement had a very high variance. Of course, some variance due to noise was to be expected. But at high SNR (over 35 dB) and after averaging over 10,000 symbols, the standard deviation should have been close to 0.1%, confirmed by simulation. In fact, some modes had variance more than an order of magnitude greater.

The reason for this wider measurement distribution was due to the demodulation of the signal prior to computing the EVM. During demodulation, reference symbols at the beginning of a frequency hop are used to recover the phase of the waveform. These reference symbols are important since phase is lost after a frequency hop occurs and needs to be restored quickly to resume useful communication. The number of reference symbols varies among the different modes, depending on how precise the timing needs to be and how much overhead in the hop is acceptable. A signal with both reference and data symbols is shown in Figure 5-1.

Any errors in the reference symbols cause phase errors that are propagated through the rest of the hop. In this way, a small amount of noise can create large variance in the EVM measurement because the noise is only averaged out over a small number of reference symbols rather than the much more numerous data symbols. The observation that confirmed this theory is that the modes with fewer reference symbols (less averaging) had proportionally higher variance. It was also these modes that were suspected to have worse EVM after equalization. The initial sense that these modes degraded after equalization was mostly due to the high variance of the EVM measurement and the lack of data needed to accurately compute the average.

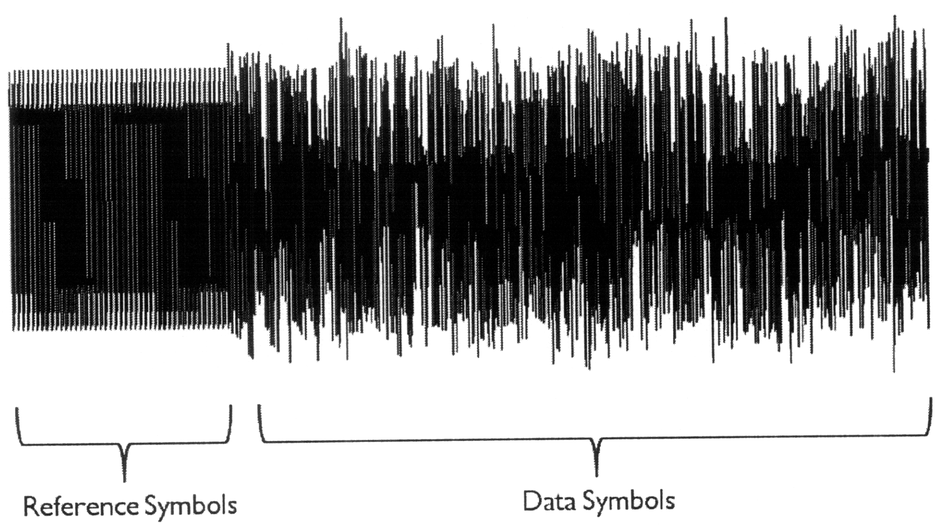


Figure 5-1: Reference and data symbols.

5.2 Optimal Filter Type

After experimenting with the FIR, DFE, and precoding architectures, it was determined that FIR equalizers were the best choice for a protected communications satellite channel. This is a surprising result, since from the standpoint of general equalization, nonlinear filters should be better than their linear counterparts. We found, however, that the advantages of DFEs and precoding depend largely on the nature of the channel. Furthermore, many of FIR filter's potential problems with noise amplification were drastically reduced through fine tuning and matched filtering. This conclusion is based on the assumption that a more complete system with nonlinear but memoryless PAs would not change the equalization requirements.

Nonlinear equalization can be extremely effective for channels with deep nulls and a lot of dispersion. These channels commonly occur due to destructive interference in cellular communication. A simulation was used to test various equalizers on a high dispersion channel, with the results shown in Figure 5-2. Precoding is not shown since its nonlinear loop has essentially the same properties as the DFE. One of the FIR equalizers is calibrated with no noise, while the other two equalizers are recalibrated for each SNR. Every equalizer had the same number of total taps. In the DFE, half the taps were used in the precursor equalizer and the other half were used in the nonlinear loop.

The most prominent feature of Figure 5-2 is the impressive performance of the DFE. At low SNR, the DFE does a better job removing ISI without amplifying noise, so it achieves a much lower EVM. As the SNR increases, all the EVMs decrease and asymptotically reach a minimum. The DFE EVM is the lowest, but not because of its ability to avoid noise amplification (the SNR is too high for this to be important). Rather, the nonlinear loop is simply better able to invert the channel using a limited number of taps. Comparing the two FIR equalizers, we see that the noise calibrated filter performs much better at low SNR. This is because it is recalibrated for each noise level, and optimally chooses between reducing ISI and amplifying noise.

The huge advantage of nonlinear equalizers in high dispersion channels almost

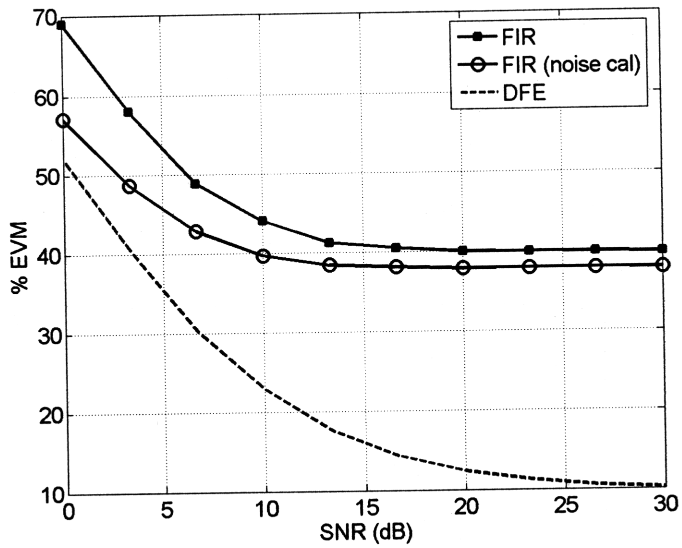


Figure 5-2: Comparison of equalizers for a high dispersion channel.

completely disappears in the low dispersion case. The results of the low dispersion channel simulation are shown in Figure 5-3. This time, the noise calibrated FIR equalizer and DFE have almost the exact same EVMs. Without deep nulls, noise amplification is negligible and not too many taps are required to approximate the channel inverse. The FIR equalizer without noise calibration had the worst performance, demonstrating the importance of noise calibration even for low dispersion channels.

The channel we expect in a communications satellite will always be closer to the low dispersion case (1 dB variations in gain) than the high dispersion case (5 dB variations). In addition, several aspects of the DFE and precoders make them an undesirable choice in general. We must consider the cost of clustering of errors and mode dependence for DFEs and the losses in PA efficiency for precoders. Experiments with the SSTVE matched the low dispersion simulation, with very little difference between FIR equalizers and DFEs. Given the implementation difficulties of nonlinear filters and the small likelihood that they would improve performance, we can conclude that the FIR architecture is best.

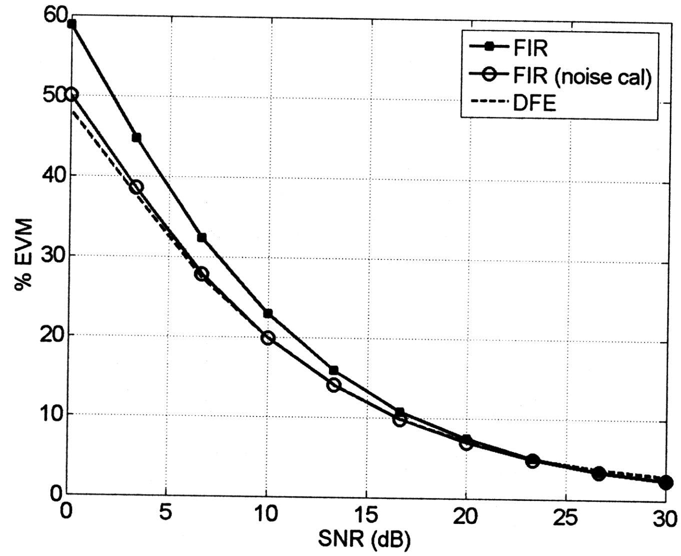


Figure 5-3: Comparison of equalizers for a low dispersion channel.

5.3 Optimal Filter Complexity

Given our choice of the FIR equalizer architecture, we can analyze the changes in performance that result from increasing or decreasing the number of taps. Increasing the number of taps could allow better equalization, but also increase the complexity and power consumption of the signal processing FPGA. More taps also results in more rounding errors, which can be non-negligible in a fixed point system.

We expect from the simulation of the low dispersion channel shown in Figure 5-3 that increasing the number of taps will not make a difference. If more taps were required for complete equalization, then we would see a substantial advantage in the DFE over the FIR equalizer. For high dispersion channels, the DFE was consistently able to do more with fewer taps.

Figure 5-4 confirms our expectation. It shows the EVM of the end-to-end IF channel as the number of taps are changed. At the 13 tap mark, which was the original number of taps in the system and the value used in simulation, the EVM has already leveled out to a minimum. Reducing to 9 or even 7 taps would result in almost the same performance. It is also interesting that the plot does not monotonically decrease, suggesting that rounding errors play an important role. Performing a similar test using software filters that do not include floating point effects produces a more smoothly decreasing plot. The EVM floor is near 2.9% EVM. About 1.5% of this can be attributed to quanitzation noise from the 8-bit receiver ADC, while the remainder is probably caused by small amounts of nonlinear distortion and timing errors.

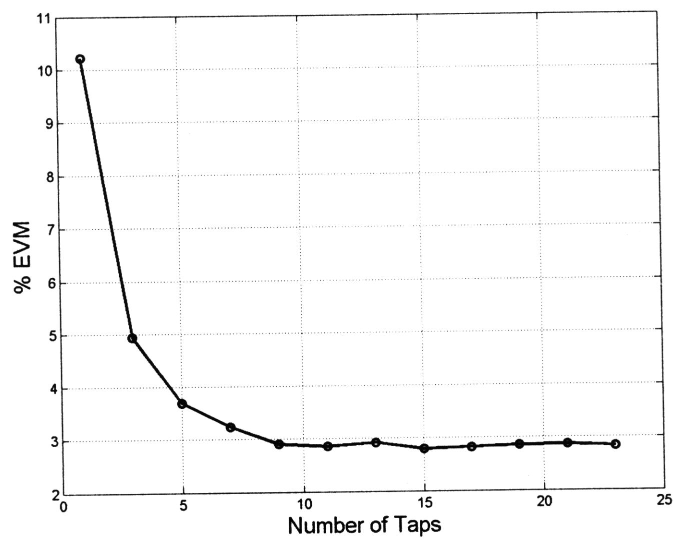


Figure 5-4: EVM verses FIR equalizer taps.

Chapter 6

Conclusion

6.1 Summary

Equalization for protected communications satellites is an interesting problem with several unique aspects. We developed an automated procedure for calibrating equalizers and experimenting with different designs. Our findings suggest that FIR equalizers are the best choice of architecture given their simplicity and effectiveness. The successful use of FIRs depends on the presence of the appropriate level of noise during calibration. We also determined that increasing the number of FIR taps beyond 13 would not improve performance, and that the number of taps can actually be reduced. Despite some of our initial observations, repeated testing showed that the equalizer does not degrade performance in the lower data rate modes.

To cope with the requirement of separating the transmit and receive equalizers as much as possible, a detailed theory of EVM and channel transfer functions was developed. The theory led us to conclude that the EVM of the equalized receiver in our procedure will likely be less than the sum of the EVM of the equalized transmitter and the EVM of an ideally equalized receiver. We can therefore maximize the separation of equalizers by using software filters of unrealizable complexity on the transmitter side while calibrating the receive equalizer.

After mastering the fundamentals of equalizer calibration, we extended our calibration procedure to work at the RF frequencies. This was accomplished by using a spectrum analyzer as a tuned receiver. The equalizers generated from the RF calibration procedure do not perform as well as those produced at IF, but are still better than the established lower bound of performance.

6.2 Future Work

The next steps in the development of a communications satellite equalizer system will focus on the areas involving the RF path. In the final system, the FIR taps will change at the start of each hop to compensate for the new channel. The first major task will be to characterize how the equalizer filter requirements change during frequency hopping. From these experiments, we would want to find out how many different filters are required to equalize the full range of possible frequency responses. Some way of compensating for aging by applying a blind adaptive algorithm on the receiver taps would also be useful.

The RF calibration procedure also has several aspects to improve. The limiting factor is the downconversion path in the spectrum analyzer, which could be made much better with a more specialized instrument. A general purpose spectrum analyzer has to be able to detect small signals and quickly sweep across a wide range of frequencies. An instrument that took advantage of the fact that we can use high SNR signals and change frequencies slowly might be a more ideal downconverter.

We are also in need of some method of verifying that the final version of the RF calibration procedure works. For IF calibration, we used our ability to measure only the transmitter's EVM to estimate how well the transmit equalizer worked. We then used this EVM to determine how well the transmit and receive equalizers were separated. During development of the RF calibration procedure, we performed both RF and IF calibration in parallel and compared the two. Although somewhat artificial, this process showed that RF calibration was theoretically viable. At 44 GHz, we can neither measure the transmitter's EVM directly, nor can we use IF calibration for comparison.

Bibliography

[1] J. G. Proakis, *Digital Communications*, 4th ed. New York: McGraw-Hill, 2001.

[2] C. B. Haskins, "Diode predistortion linearization for power amplifer RFICs in digital radios," Masters Thesis, Virginia Polytechnic Institute, Department of Electrical Engineering, Apr. 200.

[3] J. Kurzweil, *An Introduction to Digital Communications.* New York: John Wily & Sons, 2005.

[4] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication.* New York: Cambridge University Press, 2005.

[5] K. Yamauchi, "A novel series diode linearizer for mobile radio power amplifiers," *IEE MTT-S Digest*, pp. 831–834, Jun. 1996.

[6] B. Farhang-Boroujeny, *Adaptive Filters Theory and Applications.* Chichester, England: John Wily & Sons, 1998.

[7] B. Mulgrew and C. F. Cowan, *Adaptive Filters and Equalisers.* Boston: Kluwer Academic Publishers, 1988.

[8] Z. Ding and Y. Li, *Blind Equalization and Identification.* New York: Marcel Dekker Inc., 2001.

[9] E. A. Lee and D. G. Messerchmitt, *Digital Communications.* Boston: Kluwer Academic Publishers, 1988.

[10] M. Tomlinson, "New automatic equalizer employing modulo arithmetic," *Electron. Lett.*, vol. 7, p. 138139, Mar. 1971.

[11] M. Miyakawa and H. Harashima, "A method of code conversion for a digital communication channel with intersymbol interference," *Trans. Inst. Electron. Commun. Eng. Jpn.*, vol. 52-A, p. 272273, Jun. 1969.