

ABSTRACT

Title of Thesis: LINE PROBING IN VOIP NETWORKS TO FIND PERFORMANCE LIMIT OF ECHO CANCELLER

Jerker Taudien, M.S., 2007

Directed by: Associate Professor, Steven A. Tretter,
Department of Electrical and Computer
Engineering

Voice over Internet Protocol (VoIP) has become an increasingly popular way to provide phone services. At this moment there is a transition going on from delivering voice over the Plain Old Telephone System (POTS) to using VoIP technology. Line echo is created in the 2-wire to 4-wire hybrid circuit between the VoIP phone and the POTS phone and is much more apparent due to the large delay in the packet network; line echo cancellation is necessary to ensure satisfactory Quality of Service (QoS). It turns out that the performance limit of the amount of echo that can be cancelled is set by the non-linear portion of the signal. Line probing is a method of inserting a known signal at the far-end and recording the near-end signal. The two signals are then analyzed together for various impediments like, non-linearities, bad ERL, and noise. Line probing is used in this thesis to find the performance limit of the echo canceller as well as other useful metrics of performance.

LINE PROBING IN VOIP NETWORKS TO FIND
PERFORMANCE LIMIT OF ECHO CANCELLER

by

Jerker Taudien

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2007

Advisory Committee:

Associate Professor Steven A. Tretter, Chair/Advisor

Professor K.J. Ray Liu

Assistant Professor Richard J. La

© Copyright by
Jerker Taudien
2007

Acknowledgements

I would like to thank Texas Instruments for the opportunity to be one of the Texas Instruments scholars 2006-2007. You have provided me with a great opportunity to work on exciting and challenging projects. A special thanks goes to Dr. Bogdan Kosanovic, who has been guiding me in my research at Texas Instruments. Professor Steven Tretter has been of great importance to me, by introducing me to Texas Instruments and supporting me in my research. Thank you very much.

I would also like to thank my family for always giving me support when I need it. Without you I would never have made it this far. My friends also need to be thanked for being great friends and helping me out when I need you.

Table of Contents

List of Tables	v
List of Figures	vi
1 Introduction	1
1.1 Introduction to VoIP Networks	1
1.2 Potential QoS Problems and their causes in VoIP Networks	3
1.3 Line Probing	6
1.4 Software Platform	7
1.5 Thesis Outline	7
2 Echo Cancellation	9
2.1 Echo in VoIP Networks	9
2.1.1 Line Echo	10
2.1.2 Acoustic Echo	10
2.2 Line Echo Cancellation	11
2.3 Limit of Performance: maxACOM	13
3 Non-Linear Distortion Analysis Tool	16
3.1 Definition of dBm0	16
3.2 Probing Signals for Non-linear Distortion Analysis Tool	16
3.3 Objective of Non-Linear Distortion Analysis Tool	17
3.4 Tool Functionality	19
3.4.1 Tone Detection in Time	19
3.4.2 Spectral Estimation	21
3.4.3 Frequency and Power Detection	22
3.4.4 Linear and Total Response, Signal to Noise Ratio and ERL	24
3.4.5 ACOM Computation	25
3.4.6 Performance Metric	26
3.4.7 Running the Non-Linear Distortion Analysis Tool	28
4 Noise Analysis Tool	30
4.1 Probing Signals	30
4.2 Objective of Noise Analysis Tool	31
4.3 Tool Functionality	31
4.3.1 Detect Start of Probing Signal	31
4.3.2 Noise Power over Time	34
4.3.3 Power Spectral Density of Noise	34
4.3.4 Finding the Power in a Given Band	36
4.3.5 Running the Noise Analysis Tool	38

5	Summary and Future Work	40
5.1	Summary	40
5.2	Future Work	42
A	Demonstration of Line Probing Tools	44
A.1	Demonstration of Non-Linear Distortion Analysis Tool	45
A.1.1	Non-Linear Distortion Analysis Tool Plots	45
A.1.2	Non-linear Distortion Analysis Tool Text Files	47
A.2	Demonstration of Noise Analysis Tool	49
A.2.1	Near-End Noise Analysis Tool Plots	50
A.2.2	Near-End Noise Analysis Tool Text Files	51
A.3	Abnormal Examples	53
A.3.1	Saturation	53
A.3.2	Bad ERL	56
A.3.3	Noise	57
	Bibliography	60

List of Tables

1.1	Problems in VoIP Networks	3
1.2	ITU-T G.114 limits for one way delay	4
1.3	Transmission delay	4
3.1	Tone sweep power levels	17
3.2	Objective of non-linear distortion analysis tool	18
3.3	Performance metric based on maxACOM	27
3.4	μ -law compressed example for finding maxACOM	28
3.5	m-files used to implement the Non-linear distortion analysis tool	29
4.1	Objective of noise analysis tool	32
4.2	Power computation from PSD, example	37
4.3	m-files used to implement the Noise analysis tool	39
A.1	Hybrid circuit simulations	44
A.2	Raw data text file	47
A.3	Processed data text file	48
A.4	Non-linear summary file	49
A.5	Power spectral density text file	52
A.6	Noise summary file	53
A.7	Saturation: summary file	55
A.8	Bad ERL: summary file	57
A.9	Noise: summary file	59

List of Figures

2.1	2-wire to 4-wire hybrid circuit	10
2.2	Line echo canceller	12
2.3	Hybrid model	14
3.1	Power over time	20
4.1	Power Spectral Density	36
4.2	Power Spectral Density Integration	38
A.1	Time domain plot from non-linear distortion analysis tool	46
A.2	Frequency domain plot from non-linear distortion analysis tool	46
A.3	Time domain plot from noise analysis tool	50
A.4	Frequency domain plot from noise analysis tool	51
A.5	Saturation: power level	54
A.6	Saturation: ERL and SNR	55
A.7	Bad ERL: ERL and SNR	56
A.8	Noise: Time domain values	57
A.9	Noise: PSD	58

Chapter 1

Introduction

1.1 Introduction to VoIP Networks

In the very beginning of the telephone age, all calls were made over an analog pair of copper wires. The technology has moved to digital circuit switched networks over the past few decades. Today most of the phone traffic is handled by the Public Switched Telephone Network (PSTN), which provides end to end dedicated circuits for the duration of the call. During the past few years a move to packet-switched networks has been initiated to support voice traffic over Internet Protocol (IP).

The main reason for the move from circuit switched voice networks to packet switched networks is to enable convergence between data services and voice services. It is of economical interest to be able to use the same equipment for voice traffic and data traffic. Reduced cost of placing a phone call is expected, since the voice-packet is treated and routed much in the same way as any other data-packet. Long distance tariffs will completely be eliminated in Voice over IP (VoIP) networks.

There are additional benefits with VoIP networks like improved scalability. Packet-switching network equipment is designed to be readily scalable. An additional router can be connected to the existing ones when the capacity is reached of the existing system. Networking equipment is much more scalable than PSTN equipment, since the later consists of proprietary equipment. Most of the PSTN

switches and circuit-boards are produced by a specific vendor and are not standardized. This means that an upgrade has to be done by a specific vendor each time capacity is added or equipment is replaced. Network equipment, on the other hand, is standardized amongst all vendors and can be upgraded using equipment from any vendor.

So far we have seen many advantages of VoIP networks over current Plain Old Telephone Systems (POTS). However, there are many drawbacks and problems that have to be solved before VoIP can completely replace POTS [1]. Reliability is one of the main issues with VoIP networks which might delay a wide scale deployment. POTS systems are extremely reliable due to the proprietary nature of such systems. A dial tone is expected every time the phone is picked up. The reliability of POTS systems is about 99.999%, which cannot be matched by a packet-switched network [2].

Another prevalent problem with VoIP networks is poor Quality of Service (QoS). We are so used to the audio quality the PSTN network delivers that any degradation in quality will be unacceptable. The most common problems in VoIP networks are large delays and echo. The threshold of tolerable delay that is not audible to the human ear is considered to be around 150 ms [2]. A conversation over PSTN rarely has a delay larger than 150 ms. However, delays in the order of 400-500 ms is not uncommon in VoIP networks. Echo is a problem that becomes increasingly annoying as the delay is increased. Fortunately echo can be removed by adaptive echo cancellers which will be discussed in Chapter 2. Jitter is another problem that is common in VoIP networks, and is the variation in time of packet

Table 1.1: Problems in VoIP Networks

Delay
Echo
Jitter
Noise
<u>Non linear distortion</u>

arrivals. Jitter can be caused by packets taking different paths from source to destination. Even worse than delayed packets are packets that are completely lost. Packet loss is a reality in packet-switched networks and can cause severe reduction of QoS. A more in detail discussion of QoS issues in VoIP networks will follow in Section 1.2.

1.2 Potential QoS Problems and their causes in VoIP Networks

Various QoS problems in VoIP networks were discussed in the previous section. This section discusses the problems in more detail as well as some additional known problems and the cause of them. A summary of the problems can be seen in Table 1.1. The most common problem in VoIP networks is excessive delay. The ITU-T recommendation G.114 recommends that the one way delay should not exceed the following values in Table 1.2 [5]. There are two main components of the total delay in networks: transmission delay and processing delay. The transmission delay is due to the propagation delay in the copper wires or optical fiber and is a linear function of distance. See Table 1.3. Processing delay is incurred at each router and other

Table 1.2: ITU-T G.114 limits for one way delay

One-way transmission time	User acceptance
0–150 ms	Acceptable for most applications
150–400 ms	Acceptable for international connections
> 400 ms	Unacceptable for general network planning

Table 1.3: Transmission delay

Transmission facility	Delay per 100 miles
T1 carrier over copper wire	1 ms
Fiber optic cable	1 ms
Microwave radio	0.7 ms

network equipment between the source and destination. Processing delay can be reduced if the number of routing points is decreased, but transmission delay cannot be reduced and the fundamental limit is set at the speed of light.

A large delay can contribute to a possible echo being perceived as more disturbing in VoIP applications. Echo in a network with no delay cannot be perceived by the human ear at all. It is very important to keep echo to a minimum in VoIP networks due to the large delay, which will result in the echo being perceived as more disturbing. There are two main types of echo in VoIP networks: line echo and acoustic echo. Line echo is only present in situations where at least one of the communicating sites is connected through PSTN. The line echo is generated in the hybrid circuit that converts between the 2-wire the 4-wire circuits. Acoustic echo, on the other hand, stems from leakage from the loudspeaker into the microphone.

A more in detail discussion of echo generation and cancellation will be given in Chapter 2.

Jitter is another possible cause of reduced QoS. Jitter is defined as variation of packet arrival times. A large variation can distort the speech a considerable amount. Jitter can be reduced by using a playout buffer at the receiving side. A large playout buffer prevents large variations of packet arrivals which can be the result of packets taking different routes from the source to the destination. However, the playout buffer introduces additional delay which is undesirable. Hence, there is a tradeoff between jitter and delay.

Another contributor to reduced QoS is noise which can be caused by transmission errors or low bit rate codecs. User Datagram Protocol (UDP) is an unreliable best effort protocol commonly used in voice applications, which does not guarantee any type of reliability [12]. It is possible for unreliable UDP protocol to lose packages without attempting to retransmit, which will result in added noise to the reconstructed signal. Noise can also stem from low bit rate codecs. The bit rate of uncompressed narrow band voice in the ITU-T G.711 standard is 64 kbps. It is possible to reduce the bit rate to an average of 5.3 kbps using the ITU-T G.723, which of course will introduce quantization noise [5]. Hence, there is a tradeoff between speech quality and bit rate.

It was previously discussed that echo becomes more prevalent in networks with large delay. Fortunately the linear portion of the echo can be cancelled by linear adaptive filters. This leads us to the discussion of the last contributor to reduced QoS in VoIP networks; non-linear distortion. Non-linear distortion can itself be a

significant contributor to degraded voice quality. However, the biggest problem with non-linear distortion is the inability of echo cancellers to cancel nonlinear echo. It will later be seen that the amount of non-linearities sets the performance limit of the echo canceller.

1.3 Line Probing

It was seen in the previous section that there are many potential problems in VoIP networks that have to be solved before it can compete with a PSTN network. Many times it is desirable to gain information about a particular connection such as delay, echo, noise, and non-linear distortion. Information can be gathered either on-line or off-line, where on-line implies that the information is periodically updated as a conversation goes on. Off-line evaluation is done on a recorded conversation or other type of audio sent between two phones and is post processed. Another distinction that has to be made is that of active vs. passive probing, where active probing implies inserting some known signals in one end and recording what is reflected back. The process of gathering information off-line in an active sense is known as line probing. The advantages of off-line active probing is that one knows exactly what is inserted in one end what is expected at the other end. A very accurate assessment can then be made of the quality of the connection by comparing the known inserted signal with the signal reflected back. This principle will be discussed in detail in later chapters.

1.4 Software Platform

The code that implements the line probing tools in this thesis is all written in Matlab version 7. The word tools refers to a collection of functions and m-files that implement the line probing functionality discussed in Chapter 3 and 4. The tools are run by typing the function name followed by a number of arguments. An example is given by: `nonlinfn(path, testid, ftype, nfc, plot_enable)` which runs the non-linear distortion analysis tool. The tools produce text files and plots that will be discussed in more detail in Appendix A.

1.5 Thesis Outline

This thesis is broken up into five chapters and one appendix. This first chapter gives an introduction to VoIP networks, discusses potential QoS problems, and briefly defines what line probing is. Chapter 2 discusses the different types of echo that can be present in VoIP networks as well as methods to cancel unwanted echo. Chapter 3 describes the tool that can be used to find non-linear distortion, ERL, and a few other parameters by probing the far-end side with a tone sweep. Chapter 4 introduces the second tool that is used to find noise and characterize it if present. The far-end probing signal used with the noise analysis tool is absolute silence. Chapters 3 and 4 are describing the design of the two tools, but does not elaborate on how they are used. Chapter 5 gives a summary of the whole thesis as well as future work that can be done to compliment the work done in this thesis. The last portion of the thesis is Appendix A, where the functionality of the Non-linear

distortion analysis tool and the Noise analysis tool are demonstrated. The function prototypes including the input arguments are presented as well as the text files and plots that are generated.

Chapter 2

Echo Cancellation

2.1 Echo in VoIP Networks

It was discussed in Section 1.2 that the large delay in VoIP networks creates a scenario where echo becomes more prevalent. It is therefore paramount for the QoS to use good echo cancellation algorithms in VoIP networks. VoIP networks are very different from PSTN networks in the sense that echo cancellation is only needed for very long distance connections in PSTN. Short delay echoes (< 30 ms) are usually not perceived by the human ear unless the power of the echo signal is very large. For this reason echo cancellation is not needed for short distance PSTN connections. On the other hand, the round-trip delay in VoIP networks is rarely less than 30 ms [9].

If a VoIP system connects to a PSTN serviced site, echo cancellation is needed most of the time to remove hybrid reflections. Echo cancellation should still be enabled in a connection between two VoIP serviced sites to cancel acoustic echo. There are in general two types of echo: line echo and acoustic echo. These two types of echo are discussed in Sections 2.1.1 and 2.1.2.

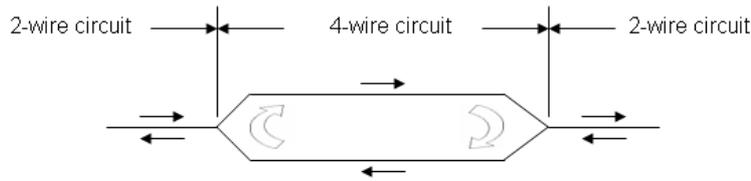


Figure 2.1: 2-wire to 4-wire hybrid circuit

2.1.1 Line Echo

In PSTN networks, line echo is generated from an impedance mismatch in the 2-wire to 4-wire hybrid circuit. The hybrid is used to reduce the number of wires that are used to carry the phone conversation between the telephone equipment and the central office. A single cable pair is used to carry both directions of transmission in the 2-wire circuit. Amplifiers can usually not pass transmission in both directions, so it is necessary to convert from 2-wire to 4-wire transmission at one point. The transmission (transmit and receive) are separate in a 4-wire circuit and flow in two different cable pairs as can be seen in Figure 2.1 [10]. It is not always possible to match the impedance of the 2-wire to 4-wire hybrid circuit, which results in unwanted feedback (echo), as illustrated by the large arrows.

Line echo can be handled by echo cancellers in VoIP networks, which are usually built into the voice codec. Line echo cancellation can work really well under controlled conditions, which will be discussed in 2.2.

2.1.2 Acoustic Echo

Acoustic echo is the second type of echo that is common in VoIP networks. It originates from poor isolation between the loudspeaker and microphone in a VoIP

handset; the voice from the speaker leaks back into the microphone and gets transmitted back to the source. Many handsets have a special acoustic echo canceller built in to deal with such situations. Severe acoustic echo is common when at least one site uses a microphone and loudspeakers connected to a computer, where the isolation between the microphone and loudspeaker is poor. Acoustic echo can easily be dealt with by acoustic echo cancellers in the handset and will not be discussed further.

2.2 Line Echo Cancellation

It was previously seen that echo can be generated either as line echo or acoustic echo. This section will discuss cancellation of line echo to enhance the QoS of the VoIP network. The notation used in this section is adopted from ITU-T recommendation G.168 [8].

A high level block diagram of a line echo canceller can be seen in Figure 2.2. In VoIP networks there is usually a line echo canceller on each of the communicating sites. In Figure 2.2 only one echo canceller is shown. The right hand side, where the echo canceller is present is referred to as the far-end and the left hand side is referred to as the near-end. The echo canceller has four ports, two that correspond to the far-end and two that correspond to the near-end side. The speech is inserted at the far-end side in R_{in} and transmitted to R_{out} at the near-end side. The echo is assumed to be inserted between R_{out} and S_{in} . The echo estimate by the echo canceller is then subtracted from the actual echo and then passed through the Non

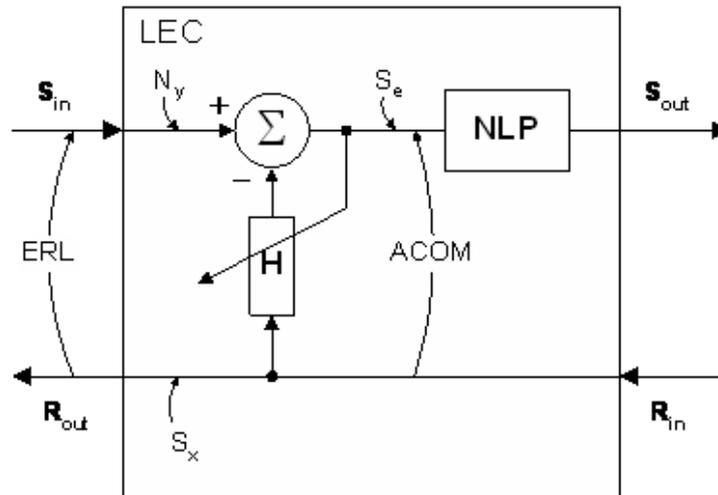


Figure 2.2: Line echo canceller

Linear Processor (NLP) to S_{out} , where it is received by the telephone loudspeaker.

- ERL – Echo Return Loss. The amount of attenuation of the echo in relation to the speech signal. ERL is measured in dB between S_{in} and R_{out} .
- ERLE – Echo Return Loss Enhancement. The amount of attenuation provided by the echo canceller. ERLE is measured in dB between S_e and S_{in} .
- ACOM – Combined Loss. The total amount of attenuation of the echo in relation to the speech signal. ACOM is measured in dB between S_{out} and R_{in} .

An echo canceller consists of three main modules: Adaptive filter, double talk detector, and NLP. The double talk detector measures the signal power at R_{in} and S_{in} to detect periods of time where there is speech activity at the far-end and near-end simultaneously. Double talk detection is necessary in order for the echo canceller to work properly; the adaptive filter uses the error signal S_e to estimate the tap weights. S_e will not be the true error signal when there is speech activity

at the near end. The adaptive filter will not converge in situations of double talk. The double talk detector therefore disables the adaptation algorithm in the echo canceller when it detects double talk to avoid such situations.

The adaptive filter is the most complex module of the line echo canceller. It is necessary to use an adaptive filter as opposed to a fixed filter, because the frequency response of the echo path is constantly changing over time. However, it is assumed that the echo path is changing slowly enough in order for the adaptive filter to converge. The adaptive filter (H) in Figure 2.2 is used to estimate the impulse response of the echo path (between R_{out} and S_{in}). The error signal S_e is then computed as the difference between S_{in} and R_{in} convolved with the tap weights of H. The error signal S_e and R_{in} are used to update the tap weights of the adaptive filter. There are many algorithms to estimate the optimal tap weights, a few examples are: the method of steepest descent and Least Mean Square (LMS) [4].

The non-linear processor is used to further attenuate the echo by using non-linear methods. The NLP will not be discussed further in this paper, but a complete discussion is presented in [8].

2.3 Limit of Performance: maxACOM

The use of adaptive linear filters in line echo cancellers was discussed in the previous section. The echo canceller is used to improve the attenuation of the unwanted echo. It would be of great interest to be able to find an upper bound on

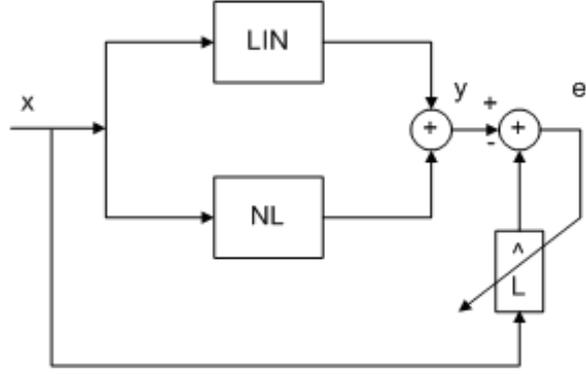


Figure 2.3: Hybrid model

the amount of achievable echo attenuation (maxACOM). It was seen in Section 2.2 that ACOM is the sum of ERL and ERLE. ERL is completely determined by the hybrid circuit. ERLE, on the other hand, is determined by the convergence of the adaptive filter in combination with the amount of non-linearities present in the echo path. The adaptive filter can only model the linear portion of the echo path and can therefore only cancel the linear portion. The maximum achievable combined loss (maxACOM) will be reached when the adaptive filter has converged completely. The hybrid can be modelled as a linear system in parallel with a non-linear system as can be seen in Figure 2.3. The parameter maxACOM can be derived from Equation 2.1 to 2.3. Equation 2.1 states that the total power is the sum of the linear and non-linear powers.

$$P_{tot} = P_L + P_{NL} \quad (2.1)$$

The power at the output of the non-linear system is found to be

$$\begin{aligned} P_{y,tot} &= P_{x,tot} \cdot g_{tot}^2 \\ &= P_x \cdot g_L^2 + P_x \cdot g_{NL}^2 \end{aligned}$$

$$\begin{aligned}
&= P_x \cdot g_L^2 + P_{y,NL} \\
&\Rightarrow \\
P_{y,NL} &= P_x \cdot g_T^2 - P_x \cdot g_L^2. \tag{2.2}
\end{aligned}$$

The value of $\max\text{ACOM}$ can be found from the ratio of the input power P_x and the output power $P_{y,NL}$ of the non-linear system.

$$\begin{aligned}
\max\text{ACOM} &= \sqrt{\frac{P_x}{P_{y,NL}}} \\
&= \sqrt{\frac{P_x}{P_x \cdot g_T^2 - P_x \cdot g_L^2}} \\
&= \sqrt{\frac{1}{g_T^2 - g_L^2}} \\
&= \frac{1}{\sqrt{g_T^2 - g_L^2}} \tag{2.3}
\end{aligned}$$

Equation 2.3 describes the upper limit of the maximum achievable combined loss. This equation will be used frequently in later sections when the performance of the line echo canceller is assessed. The dB quantity can be found from

$$\max\text{ACOM}_{dB} = 20 \cdot \log \max\text{ACOM}. \tag{2.4}$$

Chapter 3

Non-Linear Distortion Analysis Tool

3.1 Definition of dBm0

All power levels in this chapter will be measured with reference to 0 dBm0. The 0 dBm0 level corresponds to the digital milliwatt (DMW) and is defined as the absolute power level at a digital reference point of the same signal that would be measured as the absolute power level, in dBm, if the reference point was analog [14]. The absolute power level in dBm is defined as $\log \frac{\text{power in mW}}{1\text{mW}}$ when the test impedance is 600Ω . In the remainder of this document it will be assumed that all signals are 16 bit signals, where the maximum level is $2^{15} - 1 = 32767$. The signal power in dBm0 can then be found to be

$$P_{dBm0} = 10 \cdot \log \frac{P_{lin}}{2^{29}} + 3. \quad (3.1)$$

A full scale sine wave will map to 3 dBm0.

3.2 Probing Signals for Non-linear Distortion Analysis Tool

The general ideas of line probing were discussed in Section 1.3. It was discussed that line probing uses known signal to actively probe the line. Three tone sweeps of different power levels are used to probe the line in the non-linear distortion analysis tool. The tone sweeps consists of sinusoids of increasing frequency followed by

Table 3.1: Tone sweep power levels

Tonesweep	Power (dBm0)
t20	-20
t10	-10
t03	-3

silence. The silence is inserted to allow accurate detection of the start and stop of the tones. There are two different versions of the tone sweep depending on if the IP phone to be probed is a Narrow Band (NB) or Wide Band (WB) phone. The frequency of the first tone is always 100 Hz. The frequencies of the tones then increase linearly in increments of 100 Hz up to and including 3400 Hz for NB and 6800 Hz for WB. The tone duration is 1 s and the silence in between the tones is 0.5 s. The first tone (100 Hz) follows a 1 s period of silence. The tone sweeps are recorded in three different power levels for a few reason, where the main reason is to be able to detect clipping. The power levels can be seen in Table 3.1

3.3 Objective of Non-Linear Distortion Analysis Tool

This section states the objective of the non-linear distortion analysis tool. The first function of the tool is to find the power of each of the tones and to detect the frequency and power of the fundamentals in each tone sweep. Next the tool should find the frequencies and powers of a user defined number of harmonics corresponding to the fundamentals in each tone. Linear gain and total gain can then be computed from the power of the fundamental and the harmonics. Signal to Noise Ratio (SNR)

Table 3.2: Objective of non-linear distortion analysis tool

Full Name Metric	Abbreviation	Units
Tone Power	P_{tone}	dBm0
Fundamental Frequency	f_{fund}	Hz
Fundamental Power	P_{fund}	dBm0
Harmonic Frequency	f_{har}	Hz
Harmonic Power	P_{har}	dBm0
Linear Gain	g_L	dB
Total Gain	g_{tot}	dB
Signal to Noise Ratio	SNR	dB
Signal to Noise Difference	SND	dB
Maximum Achievable Combine Loss	maxACOM	dB
Total ERL	tERL	dB
Linear ERL	fERL	dB

and SND, can also be computed from the powers of the fundamental and harmonics. SNR is defined as the ratio between the powers of the fundamental and the largest harmonic. SND is defined as the ratio between the power of the fundamental and the tone power minus the power of the fundamental. Hence, SNR is always larger than SND. Linear and total ERL should also be found as well as maxACOM. Finally the tool should compute a performance metric that will tell the user about the echo performance. This performance metric should be based on maxACOM. The parameters to be computed are summarized in Table 3.2

3.4 Tool Functionality

The functionality of the non-linear distortion analysis tool will now be described in detail. The tool is broken up into six different modules, which will be described in the next few sections.

3.4.1 Tone Detection in Time

The non-linear distortion analysis tool has access to both the far-end and the near-end recordings. It is assumed that the delay in the IP network is within some reasonable limit, so that the far- and near-end recordings can be assumed to be synchronized. This assumption simplifies the matter of finding the start and end of each of the tones in the tone sweep. The far-end file is a lot cleaner and the tones have larger power than the tones in the near-end file; it is easier to use the far end file to detect the start and end of the tones.

The power of the signal in the far end file is computed and used to find the beginning and end of the tones. The power has to be computed and averaged over time to find a good estimate of the instantaneous power. An example of power over time can be seen in Figure 3.1. Eventually the power spectrum will be computed using a sliding Fast Fourier Transform (FFT) in order to find the frequency components of the tone sweep. The power spectrum is used to compute the instantaneous power over time, since it is needed for later modules anyway. The FFT coefficients can be summed and averaged by the number of FFT points to find the power. The power in the frequency domain is the same as the power in the time domain due to

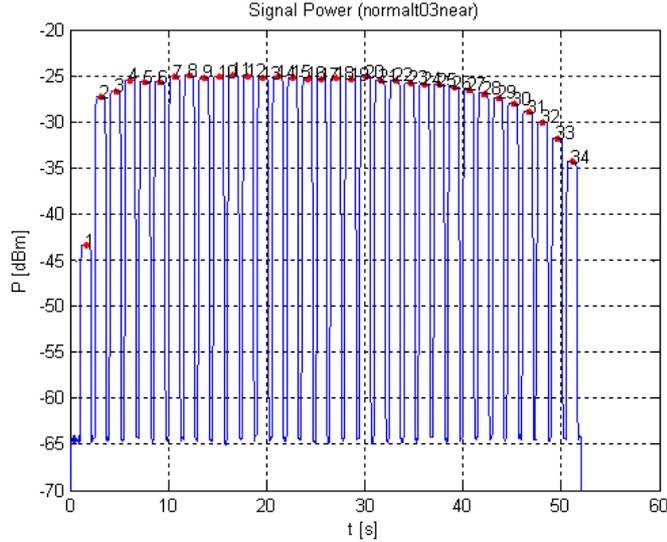


Figure 3.1: Power over time

the fact that the FFT is a unitary transform [11]. The instantaneous power estimate is then searched for points that have a power level above a user defined threshold. In order to pass for a tone, the power estimate must be above the threshold for at least 0.7 s (the actual tone duration is 1 s). The power estimate is then searched for power variations within the tone if the previous criterion is satisfied. The power is not allowed to vary by more than 0.1 dB to be classified as a detected tone.

Every time a tone is detected, the frequency of the fundamental must be found in order to compare it to the expected frequency. Detecting the frequency of the tone will be discussed in Section 3.4.3. The expected frequency depends on if the system is in NB or WB mode as discussed in Section 3.2. The first frequency to look for is always 100 Hz. A finite state machine (FSM) is used to find the start and end time of all the tones. The FMS advances to the next state if the detected frequency is within some tolerance of the expected frequency. The implicit assumption here is

that the fundamental always has larger power than any of the harmonics. The FSM is reset to state zero if the detected frequency is not equal to the expected frequency. When the FSM reaches its final state, all of the tones and their time locations are found. The algorithm for finding the location of the tones is summarized below.

1. Compute the power over time from the sliding FFT.
2. Find the time location of all power measurements that are larger than the user specified threshold.
3. Is the length of the current tone larger than 0.7 s?
4. Is the maximum power variation within the tone less than 0.1 dB?
5. Is the frequency of the detected tone within the tolerance of the expected tone? If yes, go to the next step. If no, reset FSM, advance to the next tone, and go to step 3.
6. Advance FSM and store the start and end time of tone.
7. Has the FSM reached its final state? If yes, done. If no, advance to the next tone, and go to step 3.

3.4.2 Spectral Estimation

The power spectrum is computed using methods of sliding FFTs. The length of the FFT is set to 2048 points and the slide forward factor is set to 1/8 of the FFT length. Let's assume that the sampling rate is 8 kHz for the moment. The number of slide forward points is $2048/8$. The time corresponding to one slide forward is $2048/(8 \cdot 8000) = 32$ ms. Hence the time resolution of this method is 32 ms. Windowing by the Blackman-Harris method is used before the FFT is computed

to reduce the side lobe effect. Each of the FFTs are stored in a matrix, where the rows correspond increasing frequency and the columns correspond to increasing time. This matrix is going to be used by the next module.

3.4.3 Frequency and Power Detection

Finding the location of the tones and computing the power spectrum was discussed in previous sections. The time location of the tone and the power spectrum matrix are going to be used when finding the powers and the frequencies of the fundamental and the harmonics. The power spectrum corresponding to a particular tone is computed as the median of the power spectrum that fit within the start and the end of that tone. Using the median approach ensures that the power spectrum that corresponds to the start and end of the tone does not get factored in.

Total power of each tone is easy to find by summing the squared magnitude of the FFT points and dividing by the length of the FFT. It is much trickier to find the power of the fundamental and harmonics.

The frequency response of a sinusoid is ideally a delta function whose location is determined by the frequency of the sinusoid. However, a finite window length will result in smearing of the spectrum; the power of each sinusoid in the tone sweep will be spread around its center frequency [3]. The estimated spectrum is the periodic convolution of the actual spectrum and the spectrum of the window given by

$$X_N(e^{j\omega}) = X(e^{j\omega}) \otimes W(e^{j\omega}), \quad (3.2)$$

where $X_N(e^{j\omega})$ is the estimated spectrum, $X(e^{j\omega})$, is the actual spectrum, and

$W(e^{j\omega})$ is the spectrum of the window. It is known that the main lobe width of the Blackman-Harris window is approximately $\frac{12\cdot\pi}{N-1}$, where N is the length of the FFT [13]. The spectrum is then spread over $\frac{12\cdot\pi}{N-1} \cdot \frac{F_s}{2\cdot\pi} \approx 24Hz$.

Power spectral density is defined as power per Hz. Integration over frequency has to be performed to find the total power of a given frequency component due to the spectral smearing. One can assume that the frequency component that corresponds to the largest power has the largest peak in the FFT vector. The power of the frequency component with the largest power can then be found by integrating around the center frequency. The resolution of the FFT vector is $8000/2048 \approx 4$ Hz. It was found above that the power of a tone spreads over roughly 24 Hz which corresponds to 6 FFT bins. It is desirable to integrate over frequency in a symmetric way around the center frequency. Therefore, the number of bins to integrate over has to be odd. The smallest odd number also larger than 6 is 7. Integration over frequency is therefore done over 7 FFT bins.

It was seen that each FFT bin corresponds to 4 Hz. The minimum separation of two sinusoids has to be at least 4 Hz to be able to individually resolve them. One might now think that the frequency location of the detected fundamental can only be estimated in increments of 4 Hz. However, there is a more clever approach that can be used in order to obtain better precision. The shape of the main lobe of the Blackman-Harris window can be approximated by a quadratic polynomial

$$y = a \cdot x^2 + b \cdot x + c. \quad (3.3)$$

Polynomial coefficients a , b , and c can be found by quadratic regression. The

maximum can be found by differentiating 3.3 and setting the derivative to zero

$$\begin{aligned}
 \frac{dy}{dx} &= 2 \cdot x + b \\
 \frac{dy}{dx} &= 0 \\
 \Rightarrow \\
 x &= \frac{-b}{2 \cdot a}.
 \end{aligned} \tag{3.4}$$

By using this method it is possible to increase the precision of the detected frequency a considerable amount.

The frequency and power of the fundamental has now been found. A user defined number of harmonics can be found in a similar way as described above. However, the bins corresponding to the fundamental have to be set to zero so that the largest harmonic will be found next instead of the fundamental when the FFT points are searched for the maximum value. After finding another harmonic the corresponding bins have to be set to zero again to enable for finding the next largest harmonic and so on.

3.4.4 Linear and Total Response, Signal to Noise Ratio and ERL

The majority of the computations are already performed as seen in the previous sections. The total power of each of the tones is found as well as the frequency and power of the fundamental and all requested harmonics. It is easy to compute the linear and total gains using

$$g_L = 20 \cdot \log \sqrt{\frac{P_{fund}}{P_0}} \tag{3.5}$$

$$g_{tot} = 20 \cdot \log \sqrt{\frac{P_{tone}}{P_0}}, \quad (3.6)$$

where P_0 is the power of the far end clean tone. It is assumed that all the powers are in linear values and not dBm0. The values of g_{tot} and g_L are in dB. Similarly, SNR and SND are computed using

$$SNR = 20 \cdot \log \sqrt{\frac{P_{fund}}{P_{har,1}}} \quad (3.7)$$

$$SND = 20 \cdot \log \sqrt{\frac{P_{fund}}{P_{tone} - P_{fund}}}, \quad (3.8)$$

where all powers are in linear values and not dBm0. $P_{har,1}$ is the power of the largest harmonic. SNR and SND are computed in dB quantities. Total and linear echo return loss, tERL and fERL, are computed for each tone from g_{tot} and g_L using

$$fERE = -20 \cdot \log g_L \quad (3.9)$$

$$tERL = -20 \cdot \log g_{tot}, \quad (3.10)$$

where g_{tot} and g_L are given in linear values and not dB. The quantities fERL and tERL are computed in dB.

3.4.5 ACOM Computation

The upper performance limit of the line echo canceller is set by maxACOM as discussed in Section 2.3. ACOM is the combined loss, which is the sum of the echo return loss ERL and the echo return loss enhancement ERLE. It is possible for ACOM to approach maxACOM as the echo canceller is approaching perfect convergence. The quantity maxACOM is computed from the linear and total gain

vectors g_L and g_{tot} respectively. Total gain always has to be larger than or equal to linear gain. The ACOM module first checks if this is true. If it is not true it checks if g_{tot} and g_L are within some error tolerance. ACOM is set to **Inf** if within tolerance and set to **NaN** if not. It is set to **NaN** in order to not include the points that do not satisfy the criteria in further calculations. The symbols **NaN** and **Inf** represent Not a Number and Infinity respectively in Matlab which is the software platform used throughout this thesis. The software platform Matlab is discussed in Section 1.4. ACOM is calculated for all points that satisfy the criteria using Equation 2.3. ACOM is calculated for all frequency points (detected tones). The value of maxACOM is calculated using

$$maxACOM = min(\overrightarrow{ACOM}). \quad (3.11)$$

The procedure is summarized below.

1. Initialized \overrightarrow{ACOM} to zero.
2. Find points where $-\varepsilon < g_{tot} - g_L < 0$, and set those points to inf
3. Find points where $g_{tot} - g_L < -\varepsilon$ and set those points to nan.
4. Calculate ACOM using Equation 2.3 for all points that do not satisfy 2 and 3.
5. Find maxACOM using Equation 3.11

3.4.6 Performance Metric

Non-linearities cannot be cancelled by the adaptive filter in the line echo canceller. The method of using maxACOM to assess the performance of a particular

Table 3.3: Performance metric based on maxACOM

Range [dB]	Distortion
$-\infty < \text{maxACOM} < 25$	Major
$25 \leq \text{maxACOM} < 36$	Moderate
$36 \leq \text{maxACOM} < \infty$	Minor

system is based on that fact. MaxACOM truly models the potential to reduce the amount of echo in a given system. It is therefore a good way to measure overall performance. A performance metric will be used to classify the echo in a particular system as Minor, Moderate, or Major, based on maxACOM. Table 3.3 shows the range of maxACOM for each of the classifications.

μ -law compression is the standard compression method in the United States. It is a non-uniform compression method used to keep the SNR constant for all possible amplitudes of the signal. μ -law compression uses a logarithmic function to compress the signal. Non-linearities are created due to the non-uniform way of quantizing the signal. The theoretical output SNR of a μ -law quantizer is

$$\frac{s_0}{N_0} = 10 \cdot \log \frac{3 \cdot L^2}{[\ln 1 + \mu]^2} \approx 38dB, \quad (3.12)$$

where μ is 255 and L is $2^8 - 1 = 255$ [10]. According to Table 3.3, the performance metric should be Minor. The non-linear distortion analysis tool was tested with a μ -law compressed near-end signal. In this case the minimum SNR and maxACOM is the same, because the linear gain is 0 dB. The results can be seen in Table 3.4.

Table 3.4: μ -law compressed example for finding maxACOM

Tone Power (dBm0)	SNR and maxACOM (dB)	Distortion Metric
-20	36.0	Minor
-10	37.2	Minor
-3	34	Moderate

3.4.7 Running the Non-Linear Distortion Analysis Tool

The Non-linear distortion analysis tool can be run by typing the following command at the Matlab prompt: `nonlinfn(path, testid, ftype, nfc, plot_enable)`. The arguments of the `nonlinfn` function will now be described. The argument `path` contains the path of the probing signals. The argument `testid` is the test id of the probing signals. The filename of the probing signals is the testid concatenated with one of the extensions in Table 3.1 followed by either an 'f' or an 'n' which refers to far-end or near-end respectively. The probing audio files can be recorded in any of the following formats: μ -law, A-law, PMC, or WAV, where `ftype` specifies the file type. The number of frequency components to search for, discussed in Section 3.4.3, is given by `nfc`. The argument `plot_enable` specifies if plots are to be generated.

A complete demonstration of the Non-linear distortion analysis tool is given in Appendix A. The plots can be seen in Figure A.1 and A.2. The output text files can be seen in Tables A.2 - A.4.

A number of m-files and Matlab functions were written to implement the functionality of the Non-linear distortion analysis tool. A list of the m-files, the functionality, and the creator of the files can be seen in Table 3.5. The creator is

Table 3.5: m-files used to implement the Non-linear distortion analysis tool

m-file name	Description	Creator
acom.m	Compute acom	TI
createsummary.m	Write parameters to summary file	JT
getala.m	Read A-law audio file	TI
getbin.m	Read PCM audio file	TI
getfreq.m	Get frequency components from power spectrum	TI
getloc.m	Find time location of tones	JT
getula.m	Read μ -law audio file	TI
getwav.m	Read WAV audio file	TI
initglobal.m	Initialize global variables	JT
nonlinfn	Main function	JT
stfft.m	Compute short time FFT	TI

either TI which stands for Texas Instruments or JT which stands for Jerker Taudien.

Chapter 4

Noise Analysis Tool

4.1 Probing Signals

The general ideas of line probing were discussed in Section 1.3. It was discussed that line probing uses known signals to actively probe the line. The goal of the noise analysis tool is to characterize the near-end noise. The noise is assumed to be of an additive nature. In order to only capture noise, the probing signal has to be complete silence. However, it is not only hard, but impossible, to find the start and end of the silence probing signal. Three tones of frequency 1004 Hz and duration of 1 s, separated by 0.5 s silence, were inserted at the beginning of the silence to simplify detection of the beginning of the actual silence. The end of the silence can be found by adding the duration of the silence to the time at which the last tone was detected. The choice to use a tone of frequency 1004 Hz was made based on recommendations by AT&T, on how to measure loss in telephone lines[6]. The frequency response of a telephone line has its peak around 1004 Hz; it is therefore desirable to use that frequency to detect the start of the silence signal. The larger the signal, the easier it is to detect it.

4.2 Objective of Noise Analysis Tool

This section states the objective of the noise analysis tool. It is desirable to find the noise power over time. From the noise power vector, the maximum, minimum, and average noise power will be found as well as the time at which they occur. The instantaneous DC component will be found over time. From the DC component vector, the maximum, minimum, and average DC component will be found as well as the time at which they occur. Power Spectral Density (PSD) can give insightful information about the nature of the present noise. White noise for example has flat PSD over all frequencies. Pink noise on the other hand has equal power in each octave [3]. The maximum, minimum, and average PSD can be found from the PSD vector as well as the frequencies at which they occur. The power of a user defined band will also be found by integrating the PSD between the user supplied lower and upper frequency. The parameters to be computed can be seen in Table 4.1.

4.3 Tool Functionality

The functionality of the noise analysis tool will now be described in detail. The tool is broken up into four different modules, which will be described in the next few sections.

4.3.1 Detect Start of Probing Signal

The noise analysis tool has access to both the far-end and the near-end recordings. It is assumed that the delay in the IP network is within some reasonable limit,

Table 4.1: Objective of noise analysis tool

Full Name Metric	Abbreviation	Units
Minimum noise power	$P_{n,min}$	dBm0
Maximum noise power	$P_{n,max}$	dBm0
Average noise power	$P_{n,avg}$	dBm0
Minimum DC component	DC_{min}	no unit
Maximum DC component	DC_{max}	no unit
Average DC component	DC_{avg}	no unit
Time of occurrence	t	s
Minimum Power Spectral Density	PSD_{min}	dBm0/Hz
Maximum Power Spectral Density	PSD_{max}	dBm0/Hz
Average Power Spectral Density	PSD_{avg}	dBm0/Hz
Frequency of occurrence	f	Hz
Power in user defined band	$P_{n,def}$	dBm0

so that the far-end and near-end recordings can be assumed to be synchronized. This assumption simplifies the matter of finding the start and end of the three preamble tones of frequency 1004 Hz. The far-end file is a lot cleaner and the tones have larger power than the tones in the near-end file. It is therefore easier to use the far end file to detect the preamble tones. The reason that the far-end file is cleaner than the near-end file is that the near-end file is the output of the echo path with the far-end signal being the input. The echo path usually attenuates, adds noise, and distorts the input in some way.

Finding tones in time was discussed in Section 3.4.1, where the start and end of the tones were found by computing the average instantaneous power over time of the signal. The same approach is used for finding the start and end of the preamble, which is inserted before the actual probing silence. The algorithm introduced in Section 3.4.1 is used, but with some minor modifications. The Finite State Machine (FSM) is implemented in a different way. It looks for three consecutive tones of frequency 1004 Hz, instead of the tone sweep. The algorithm for finding the preamble is summarized below.

1. Compute the power over time from the sliding FFT.
2. Find the time location of all power measurements that are larger than the user specified threshold.
3. Is the length of the current tone larger than 0.7 s?
4. Is the maximum power variation within the tone less than 0.1 dB?
5. Is the frequency of the detected tone within 10 Hz of 1004 Hz (994 - 1014)? If yes, go to the next step. If no, reset FMS, advance to the next tone, and go

to step 3.

6. Advance FSM and store the start and end time of tone.
7. Has the FSM reached state 3 (3 tones found)? If yes, done. If no, advance to the next tone, and go to step 3.

The probing signal consists of three tones of frequency 1004 Hz, separated by 0.5 s silence, followed by an additional second of silence before the actual 30 seconds of silence. The start of the actual silence is found by assuming that the silence starts one second after the last detected tone. This assumption is valid as long as the delay in the IP network is within a reasonable limit.

4.3.2 Noise Power over Time

The power over time of the noise signal is estimated using the IEC.651 standard [7]. Segments of the signal are analyzed separately, where the segment length is 5 ms or 40 samples; the time resolution of the power measurements is 5 ms. In the IEC.651 standard an option of frequency weighting is available, but is not used in the noise analysis tool. An additional parameter known as the time constant, tau, specifies the weight in time of the samples. The time constant is set to 35 ms and is used to perform exponential averaging over time.

4.3.3 Power Spectral Density of Noise

A common method for estimating the PSD of a data set is the periodogram method. The power spectral density is the Fourier transform of the auto-correlation

function of the data. The periodogram is given by

$$\hat{P}_{PER}(f) = \frac{1}{L} \left| \sum_{n=0}^{L-1} x[n] \cdot e^{-j2\pi fn} \right|^2, \quad (4.1)$$

where L is the number of PSD points. The periodogram is an asymptotically unbiased estimator, since the expected value of the PSD is the same as the actual PSD as the number of data points approaches infinity [3]. However, it is not consistent in the sense that the variance of the estimate does not go to zero as the number of data points approaches infinity. Using the average periodogram instead of the simple periodogram solves the problem of consistency, which is given by

$$\begin{aligned} \hat{P}_{AVPER}(f) &= \frac{1}{K} \sum_{m=0}^{K-1} \hat{P}_{PER}^{(m)}(f) \\ \hat{P}_{PER}^{(m)}(f) &= \frac{1}{L} \left| \sum_{n=0}^{L-1} x_m[n] \cdot e^{-j2\pi fn} \right|^2. \end{aligned} \quad (4.2)$$

The actual PSD estimate used for the noise analysis tool is based on Welch's method. The number of points used to compute the FFT is 512, and the window type is Hamming. Welch's method uses overlapping data, where the amount of overlap is to be specified. The overlap used with the noise analysis tool is set to 3/4 of the number of FFT points (384). It was previously seen that the variance of the PSD estimate decreases as the number of averaged estimates is increased. The number of PSD estimates that are averaged in this case is

$$N_{AVG} = \frac{Dur_{SIG} \cdot FS}{N_{FFT} - N_{OVERLAP}} = \frac{30 \cdot 8000}{512 - 512 \cdot \frac{3}{4}} = 1875 \quad (4.3)$$

The variance of the estimate should be very close to zero with this many averaged PSD estimates. An example of the PSD estimate can be seen in Figure 4.1. It can

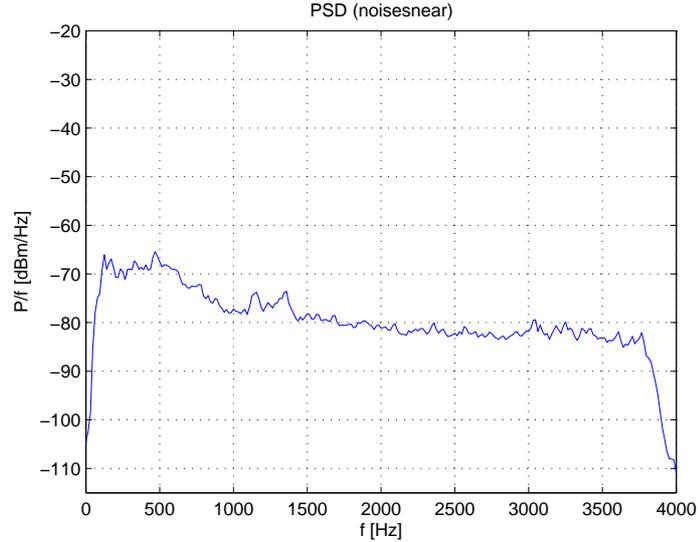


Figure 4.1: Power Spectral Density

be seen that the noise is approximately pink. The maximum, minimum, and average PSD can be found from the PSD vector, computed using the Welch's method. The frequency of occurrence for the minimum and maximum PSD are computed from the PSD vector as well.

4.3.4 Finding the Power in a Given Band

The PSD computed in the previous section is measured in dBm0/Hz and not in dBm0/bin. It should be noted that the bin width is usually not equal to 1 Hz. To find the total power in a given bin, the PSD value in that bin has to be multiplied by the bin width in Hz.

The lower and upper bound (f_1 and f_2) of power integration has to be specified by the user. The total power within the lower and upper limit is to be found from the PSD vector. The values of PSD have to be given in linear form, and not dB, to

Table 4.2: Power computation from PSD, example

Frequency band (Hz)	Power (dBm0)
200–400	-46.1
1000–2000	-47.8

be able to integrate over frequency.

Figure 4.2 illustrates how the total power is found within f_1 and f_2 . This example has 8 PSD points: p_0 – p_7 . The dashed lines are the mid-points between consecutive PSD points. The most accurate estimate of PSD at any frequency point is the nearest neighbor. Integration is performed over frequency to find the total power within the band $[f_1, f_2]$. At every point of integration the PSD of the nearest neighbor has to be used. The example shows that the PSD value, p_1 , is integrated over from f_1 to the frequency point in between p_1 and p_2 . Similarly, the PSD value, p_6 , is integrated over from f_2 down to the frequency point in between p_5 and p_6 . These two endpoints have to be treated differently compared to the PSD points, p_2 to p_5 . The power between the points between p_2 and p_5 is found by multiplying each of the PSD values by the bin width in Hz and summing. The total linear power in dBm0 quantities is computed using Equation 3.1.

It was seen in Figure 4.1 that the PSD is approximately pink. It is known that pink noise has equal power in each octave. The algorithm for finding the total power can be tested by finding the total power within two different octaves. The results can be seen in Table 4.2.

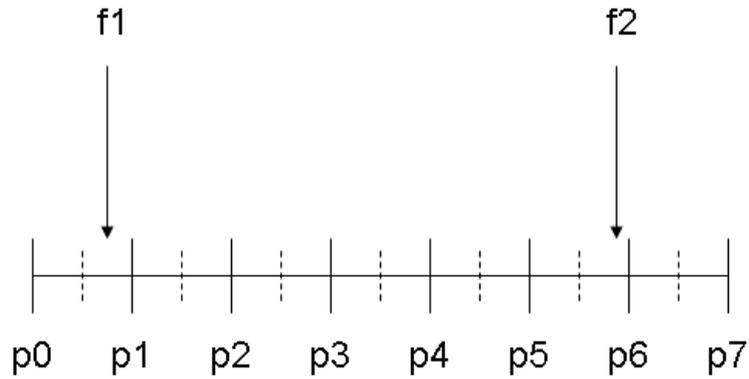


Figure 4.2: Power Spectral Density Integration

4.3.5 Running the Noise Analysis Tool

The Noise analysis tool can be run by typing the following command at the Matlab prompt: `noisefn(path, testid, ftype, tau, f1, f2, plot_enable)`. The arguments of the `noisefn` function will now be described. The argument `path` contains the path of the probing signals. The argument `testid` is the test id of the probing signals. The filename of the probing signals is the `testid` concatenated with 's' (stands for silence) followed by either an 'f' or an 'n' which refers to far-end or near-end respectively. The probing audio files can be recorded in any of the following formats: μ -law, A-law, PMC, or WAV, where `ftype` specifies the file type. The argument `tau` specifies the time constant for exponential power averaging. The lower and upper frequency of PSD integration discussed in Section 4.3.4 is given by `f1` and `f2`. The argument `plot_enable` specifies if plots are to be generated.

A complete demonstration of the Noise analysis tool is given in Appendix A. The plots can be seen in Figure A.3 and A.4. The output text files can be seen in Tables A.5 and A.6.

Table 4.3: m-files used to implement the Noise analysis tool

m-file name	Description	Creator
getala.m	Read A-law audio file	TI
getbin.m	Read PCM audio file	TI
getfreq.m	Get frequency components from power spectrum	TI
getloc.m	Find time location of tones	JT
getula.m	Read μ -law audio file	TI
getwav.m	Read WAV audio file	TI
ninitglobal.m	Initialize global variables	JT
noisefn	Main function	JT
scanp651.m	Compute exponentially averaged power	TI
stfft.m	Compute short time FFT	TI

A number of m-files and Matlab functions were written to implement the functionality of the Noise analysis tool. A list of the m-files, the functionality, and the creator of the files can be seen in Table 4.3. The creator is either TI which stands for Texas Instruments or JT which stands for Jerker Taudien.

Chapter 5

Summary and Future Work

5.1 Summary

Voice over Internet Protocol (VoIP) has become an increasingly popular way to provide phone services. At this moment there is a transition going on from delivering voice over the Plain Old Telephone System (POTS) to using VoIP technology. However, there are many potential problems with VoIP networks that have to be solved before it will replace the old technology. One of the fundamental problems with transmitting voice over a packet network is the large delay, which is a sum of transmission and processing delay. The large delay makes the possibly present echo appear much more disturbing to the user.

There are two main types of echo: acoustic and electrical (line) echo. Line echo is generated in the 2-wire to 4-wire hybrid circuit that is present in all POTS networks. Acoustic echo will not be discussed further in this thesis. Line echo can be cancelled using a Line Echo Canceller (LEC), which is an adaptive filter that estimates the frequency response of the echo path. The far-end receive signal is passed through the adaptive filter and the output is subtracted from the near-end send signal. A simple block diagram of a LEC can be seen in Figure 2.2. The LEC module is a linear estimator and can only cancel out the linear portion of the distortion. The non-linear portion cannot be cancelled and therefore sets the upper

limit of performance of the echo canceller. The combined loss (ACOM) is the sum of Echo Return Loss (ERL) and Echo Return Loss Enhancement (ERLE). The limit of combined loss (maxACOM) is determined by the amount of non-linearities.

Line probing is a method of inserting a known signal at the far-end and recording the near-end signal. The two signals are then analyzed together for various impairments like non-linearities, bad ERL, and noise. Line probing is used in this thesis to find the performance limit of the echo canceller as well as other useful metrics of performance.

Line probing is used in the Non-linear distortion analysis tool and the Noise analysis tool, which are the two tools that are discussed in this thesis. The objective of the Non-linear distortion analysis tool is to find ACOM, ERL, and SNR for each of the tones in the tone sweep of frequencies from 100 Hz to 3400 Hz (100Hz to 6800 Hz in the wideband case). maxACOM is then found by taking the min of ACOM over frequency. A performance metric is calculated based on maxACOM, which can be minor, moderate, or major.

The objective of the Noise analysis tool is to find properties of noise. The noise power is calculated over time and minimum, average, and maximum values are found over time. Power spectral Density (PSD) is also found from the noise signal. The minimum, average, and maximum values are found as well as the frequencies at which they occur. Power in a given frequency band is found by integrating PSD over frequency.

5.2 Future Work

Throughout this thesis a few assumptions were made about the far-end and near-end signals. It was assumed that the line probing tool has access to the far-end signal, which might not always be the case. Furthermore, it was assumed that the far-end and near-end signals are synchronized in time. For the synchronization to hold it is imperative that the network delay is small in relation to the duration of a single tone in the tone sweep.

It would be possible to solve the line probing problem even if there was a large delay between the far-end and near-end signals, by finding the delay and re-synchronizing the two signals in time. However, it would still be necessary to have access to the far-end signal. Another possibility would be to not use the far-end signal. Then there would be no issue of finding the network delay. The tone frequencies would then be found from the near-end signal directly. Using the latter methodology would introduce additional problems like incorrectly detected frequencies and missed tones due to the near-end signal being a lot less clean than the far-end signal. These ideas would be ways of improving the versatility of the line probing tool.

It would be possible to find the network delay by estimating the transfer function of the hybrid circuit. The delay could easily be found by visually inspecting the delay between the first filter coefficient and the first non-zero filter coefficient modelling the transfer function. Work has been done in this area, but it is not further discussed in this thesis. The delay could be used to synchronize the far-

end and near-end signals. The line probing tool would then work well even under conditions of large delay.

Work has also been done on detecting frequencies that are incorrectly detected by the line probing tool or even not detected at all. This work could be used with the line probing tool to circumvent the need for the far-end signal.

A few suggestions were made about how to improve the versatility of the line probing tool. There are many more ways to improve not only the versatility, but also the functionality of the line probing tool.

Chapter A

Demonstration of Line Probing Tools

Sections A.1 to A.2.2 are illustrating the functionality of the Non-linear distortion analysis tool and Noise analysis tool. The far-end and near-end signals are simulated using a G.168 problem free hybrid. Sections A.3 to A.3.3 show examples of a problematic hybrid circuit which will result in poor QoS. Three pairs of far-end and near-end signals are simulated from a G.168 hybrid circuit with the following problems: saturation, excessive noise, and bad ERL. The line probing tools are used to characterize the problems with the hybrid circuit.

The values of ERL, delay, near end White Gaussian Noise (WGN), Office Noise (ON), and forward path gain for the four simulations of the hybrid circuit can be seen in Table A.1. The first hybrid corresponds to a problem free hybrid, and the last three have some type of abnormalities.

Table A.1: Hybrid circuit simulations

Hybrid	ERL (dB)	Delay (ms)	P WGN (dBm0)	P ON (dBm0)	Gain (dB)
Normal	23	4 ms	-65	0	0
Saturation	6	4 ms	-65	0	15
Bad ERL	3	4 ms	-65	0	0
Noise	23	4 ms	-65	-40	0

A.1 Demonstration of Non-Linear Distortion Analysis Tool

The non-linear distortion analysis tool is used to find non-linear distortion in a pair of near-end and far-end signals. The signals used to demo this tool are simulated to resemble signals that would be obtained from a problem free hybrid circuit. The far-end probing signal is a tone sweep, which will enable finding non-linear distortion. Information about the syntax and running the tool will not be discussed here. The plots and output files produced by the tool will now be discussed.

A.1.1 Non-Linear Distortion Analysis Tool Plots

The non-linear distortion analysis tool produces 6 plots in total, 2 for each near and far end file pair. Figure A.1 shows the first plot, which contains average power over time. The intervals of relatively large power correspond to the tones and the intervals of relatively small power correspond to the silence between the tones of the tone sweep.

Figure A.2 shows the second plot produced by the non-linear distortion analysis tool, which contains frequency domain values. Subplot 1 shows total and fundamental hybrid response over frequency, which in this case are very close to each other. Subplot 2 shows simple SNR and SND, where simple SNR and SND are defined to be power of the fundamental over power of the largest harmonic and power of the fundamental over power of everything else respectively.

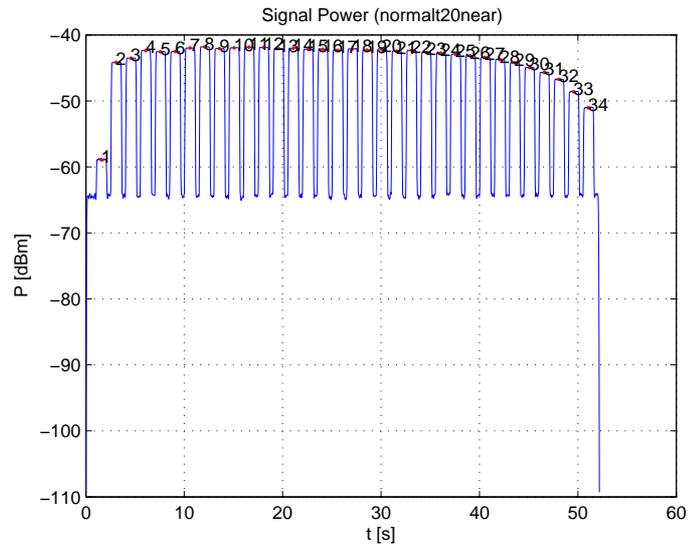


Figure A.1: Time domain plot from non-linear distortion analysis tool

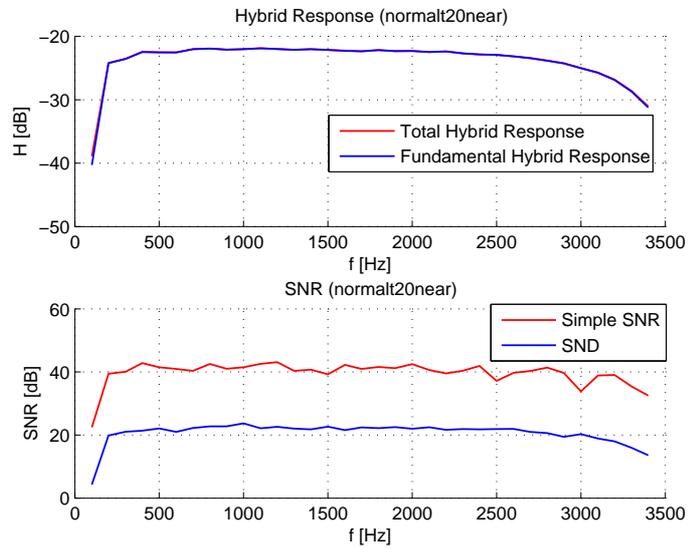


Figure A.2: Frequency domain plot from non-linear distortion analysis tool

Table A.2: Raw data text file

f_{fund}	f_{har}	f_{har}	P_{fund}	P_{har}	P_{har}
100.83	1300.26	1500.01	-60.28	-82.75	-84.10
199.57	1398.38	2600.40	-44.22	-83.66	-85.47
300.43	2300.47	1899.91	-43.56	-83.61	-85.49
399.17	2000.76	1597.22	-42.39	-85.41	-86.20
500.00	2500.29	1498.52	-42.59	-84.46	-84.07
⋮	⋮	⋮	⋮	⋮	⋮
3000.00	999.93	2160.23	-44.99	-78.82	-84.00
3100.84	301.15	2898.29	-45.75	-84.64	-84.74
3199.57	2137.41	1988.12	-46.79	-86.28	-86.71
3300.43	1299.46	2500.46	-48.72	-84.12	-84.22
3399.16	601.04	1800.70	-51.21	-83.70	-84.58

A.1.2 Non-linear Distortion Analysis Tool Text Files

The non-linear distortion analysis tool produces 7 text files in total, 2 for each near and far end file pair and one summary file. A few sample rows of the raw text file can be seen in Table A.2. The first four and last four columns contain the frequencies [Hz] and powers [dBm] of the fundamental and the largest harmonics respectively. For example the fifth row tells the user that the largest harmonics corresponding to the 500 Hz tone are 2500 Hz and 1499 Hz. The power of the fundamental and the two largest harmonics are -42.59 dBm, -84.48 dBm, and -84.07 dBm.

Table A.3: Processed data text file

f_{fund}	P_{tone}	P_{fund}	SNR	SND	maxACOM
100.83	-58.91	-60.28	22.47	4.30	44.60
199.57	-44.18	-44.22	39.44	19.83	44.08
300.43	-43.53	-43.56	40.05	21.05	44.63
399.17	-42.36	-42.39	43.02	21.39	43.88
500.00	-42.56	-42.59	41.49	22.07	44.61
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
3000.00	-44.95	-44.99	33.83	20.29	45.32
3100.84	-45.70	-45.75	38.89	18.90	44.66
3199.57	-46.73	-46.79	39.48	18.02	44.91
3300.43	-48.61	-48.72	35.41	15.96	44.70
3399.16	-51.02	-51.21	32.49	13.57	44.81

A few sample rows from the processed text file can be seen in Table A.3. The columns correspond to: frequency [Hz], total and linear power [dBm], SNR [dB], SND [dB], and maxACOM [dB]. For example the fifth row shows the values corresponding to the 500 Hz tone. The values of total power, linear power, SNR, SND, and maxACOM are -42.56 dBm, -42.59 dBm, 41.49 dB, 22.07 dB, and 44.61 dB.

The summary text file contains a summary of some of the most important parameters as well as the path of the input near-end and far-end end file pairs and the path of the output files. The summary parameters can be seen in Table A.4. The

Table A.4: Non-linear summary file

```

normalnonlin
This file was automatically generated by nonlinfn. It
contains a summary of the results of the nonlinear analysis
and directions where to find the generated output files.
*****
*****
Test inputs
Far end files:
      G:\TI\signals\hybsim\normal\normalt20f.ula
      G:\TI\signals\hybsim\normal\normalt10f.ula
      G:\TI\signals\hybsim\normal\normalt03f.ula
Near end files:
      G:\TI\signals\hybsim\normal\normalt20n.ula
      G:\TI\signals\hybsim\normal\normalt10n.ula
      G:\TI\signals\hybsim\normal\normalt03n.ula
*****
*****
Test output files
G:\TI\signals\hybsim\normal\normalt20nonlinfp.txt
G:\TI\signals\hybsim\normal\normalt10nonlinfp.txt
G:\TI\signals\hybsim\normal\normalt03nonlinfp.txt
G:\TI\signals\hybsim\normal\normalt20nonlinfsnr.txt
G:\TI\signals\hybsim\normal\normalt10nonlinfsnr.txt
G:\TI\signals\hybsim\normal\normalt03nonlinfsnr.txt

-Files with extension raw contain the frequencies and powers
of the fundamental and 4 largest harmonics
-Files with extension nonlinsnr contain the frequencies of the
fundamentals, total power of the tone, power of the fundamental,
simple SNR, SND, and maxACOM
*****
*****
Units of F0, Fmin, SNR, SND, ERL, and ACOM are in Hz, Hz, dB, dB, dB, and dB
repectively

File          Px [dBm] Min SNR( F0,Fmin) Max SNR( F0,Fmin) Min SND( F0)
normalt20     -20.0    22.5( 101,1300)    43.1(1200,2001)    4.3( 101)
normalt10     -10.0    25.3( 101,2700)    49.9(2000, 823)   12.6( 101)
normalt03      -3.0    24.7( 101, 700)    60.0(2000,1039)   17.0( 101)

File          Max SND( F0) fERL  tERL maxACOM Distortion
normalt20     23.7(1000) 23.4 23.4 43.5 Minor
normalt10     32.5(1000) 24.5 24.5 51.4 Minor
normalt03     40.2(2000) 23.4 23.4 53.4 Minor

```

top part of the summary file contains the path of the input and output files. The bottom part of the summary file contains a table with some of the most important parameters and the applicable units. The parameters F0 and Fmin are the frequency of the fundamental and largest harmonic respectively. The parameters fERL and tERL are the fundamental and total ERL respectively.

A.2 Demonstration of Noise Analysis Tool

The near and noise analysis tool is used to find noise properties of a pair of near-end and far-end end signals. The signals used to demonstrate this tool are

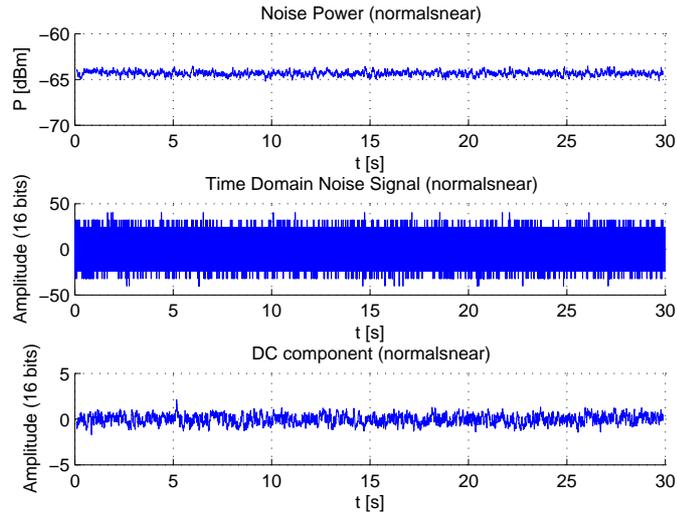


Figure A.3: Time domain plot from noise analysis tool

simulated to resemble signals that would be obtained from a problem free hybrid circuit. The far-end probing signal consists of silence, which will enable finding properties of the noise that might be mixed in at the near-end. Information about the syntax and running the tool will not be discussed here. The plots and output files produced by the tool will now be discussed.

A.2.1 Near-End Noise Analysis Tool Plots

The near-end noise analysis tool produces 2 plots in total. Figure A.3 shows the first plot which contains time domain values: Noise power, noise amplitude, and DC amplitude. It can be seen that the noise power is fairly constant around -64 dBm in subplot 1. In subplot 2 and 3 it can be seen that the DC offset is relatively small compared to the noise amplitude.

Figure A.4 shows the second plot produced by the near end noise analysis tool

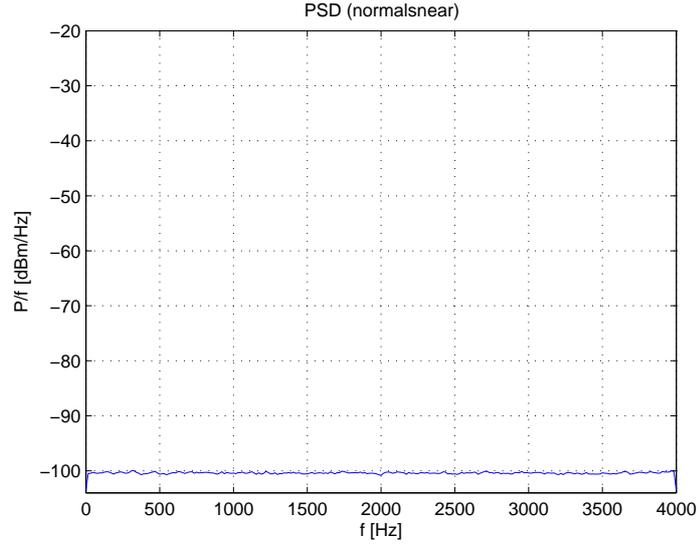


Figure A.4: Frequency domain plot from noise analysis tool

which contains Power Spectral Density (PSD) over frequency. It can be seen that the noise has a very flat frequency response which is consistent with that of WGN. The total power can be found by integration of PSD over a desired frequency band.

A.2.2 Near-End Noise Analysis Tool Text Files

The near-end noise analysis tool produces 2 text files in total, one file with PSD values and one summary file. A few sample rows of the PSD text file can be seen in Table A.5. The columns contain frequency [Hz] and PSD [dBm/Hz]. For example the fifth row shows that the power spectral density at 62.5 Hz is -100.38 dBm/Hz.

The summary text file, which can be seen in Figure A.6 contains a summary of some of the most important parameters as well as the path of the input near-end and far-end files and the path of the output files. The top part of the summary file

Table A.5: Power spectral density text file

f	PSD
0.00	-103.45
15.63	-100.46
31.25	-100.48
46.88	-100.28
62.50	-100.38
⋮	⋮
3937.50	-100.22
3953.13	-100.22
3968.75	-99.94
3984.38	-100.17
4000.00	-103.57

Table A.6: Noise summary file

```

normalnoise
This file was automatically generated by noisefn. It
contains a summary of the results of the noise analysis
and directions where to find the generated output files
*****
*****
Test inputs
Far end file:      G:\TI\signals\hybsim\normal\normalsf.ula
Near end file:    G:\TI\signals\hybsim\normal\normalsn.ula
*****
Test output files
G:\TI\signals\hybsim\normal\normalpsd.txt

The file with the extension psd contains the frequency and psd columns.
*****
*****
Average power in the band [ 0,4000] Hz is -64.4 dBm
*****
Units of time, frequency, Pn, and PSD are in s, Hz, dBm, and dBm/Hz respectively.
DC is a unitless quantity and is limited to that of 16 bit integers.

File              Min Pn( t) Max Pn( t) Avg Pn Min DC( t) Max DC( t) Avg
DC
normals          -65.2( 9.7) -63.5(26.1) -64.4      0(22.2)      2( 5.2)
-0

File              Min PSD( f) Max PSD( f) Avg PSD
normals          -103.6(4000) -99.9( 313) -100.4

```

contains the path of the input and output files. The bottom part of the summary file contains a table with some of the most important parameters and the applicable units. Another informative part of the summary file is the line above the bottom table, which provides information about the average power in a user defined frequency band. In this case the average power is -64.4 dBm in the band 0 - 4000 Hz.

A.3 Abnormal Examples

A.3.1 Saturation

Saturation will give rise to non-linear distortion. It is therefore useful to look at the saturation examples with the non-linear distortion analysis tool. The other tool will not be used to analyze the test signals in this section. The tone sweep

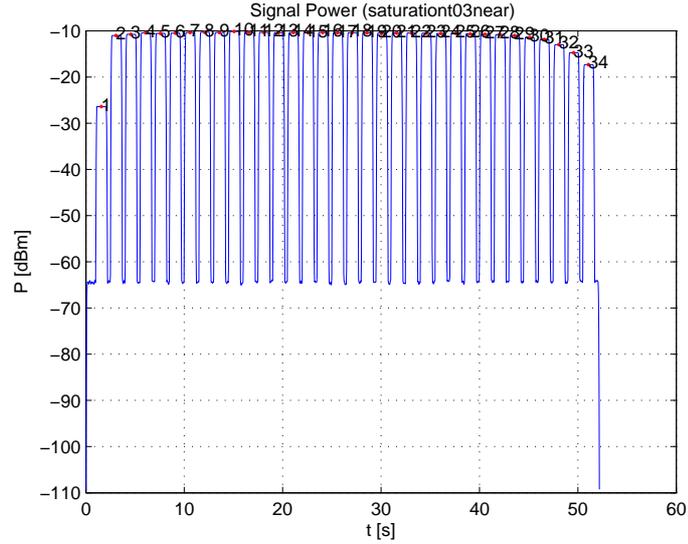


Figure A.5: Saturation: power level

probing signals used with the non-linear distortion analysis tool have different levels. Only the signal at the largest level, which is at -3dBm, will become saturated. The plots and text files produced by the non-linear distortion analysis tool will now be discussed. Figure A.5 shows the power level over time produced by the non-linear distortion analysis tool. It can be seen that the response is very flat between 5 and 45 seconds. The response can be compared to that of Figure A.1 which has a less flat response. The flatness is caused by the saturation, but in general nothing can be concluded about saturation from the shape of the power level.

The bottom plot in Figure A.6 shows SNR and SND vs. frequency. Low values of SNR and SND will in general indicate that there are large amounts of non-linearities present. It can be seen that the SND and SNR are much smaller in Figure A.6 compared to that in Figure A.2.

The plots give a hint of what is going on with the investigated hybrid. It

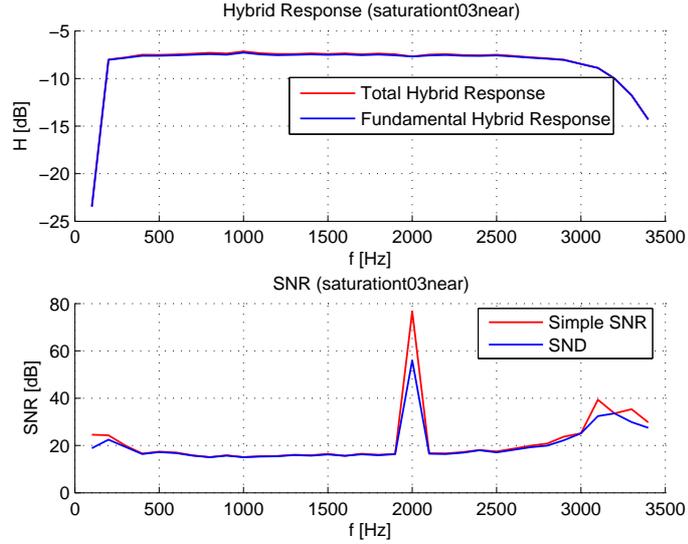


Figure A.6: Saturation: ERL and SNR

Table A.7: Saturation: summary file

File	Px [dBm]	Min SNR(F0,Fmin)	Max SNR(F0,Fmin)	Min SND(F0)
saturationt20	-20.0	25.1(101,1300)	Inf(2399, 0)	17.2(101)
saturationt10	-10.0	26.3(101,2700)	93.2(2000,1970)	20.5(101)
saturationt03	-3.0	15.1(1000,3000)	76.7(2000, 532)	15.0(800)

File	Max SND(F0)	fERL	tERL	maxACOM	Distortion
saturationt20	45.5(500)	6.4	6.4	35.6	Moderate
saturationt10	58.9(2000)	6.4	6.4	38.0	Minor
saturationt03	56.0(2000)	8.2	8.1	22.4	Major

is useful to look at the text files produced by the tool to get more insight in the problem. The summary file in Table A.7 confirms that the SNR and SND are low. More importantly the summary file provides information about maximum ACOM, which is much larger for the files at a smaller level than for the file at -3dBm. This together with the distortion metric, which is classified as major are good proof of saturation or non linear response in the hybrid.

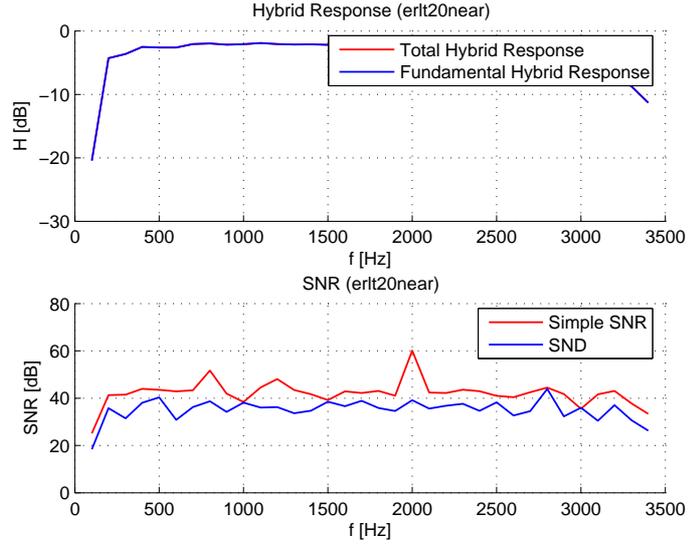


Figure A.7: Bad ERL: ERL and SNR

A.3.2 Bad ERL

ERL can be found using the Non-linear distortion analysis tool. The noise analysis tool will not be used to analyze the test signals in this section. The Non-linear distortion analysis tool computes hybrid response which is the reciprocal of ERL. The first plot in Figure A.7 shows the hybrid response as computed by the Non-linear distortion analysis tool. It can be seen that the largest response is somewhere around -3 dB, which translates to 3 dB ERL, which is a very poor value.

The summary text files produced by the Non-linear distortion analysis tool list the ERL. The summary file produced by the Non-linear distortion analysis tool is shown in Table A.8. Fundamental and total ERL are both found to be 3.4 dB.

Table A.8: Bad ERL: summary file

File	Px [dBm]	Min SNR(F0,Fmin)	Max SNR(F0,Fmin)	Min SND(F0)
erlt20	-20.0	25.1(101,1300)	60.1(2000,2072)	18.4(101)
erlt10	-10.0	26.4(101,2700)	Inf(800, 0)	20.7(101)
erlt03	-3.0	24.6(101, 700)	55.4(2000,2922)	19.1(101)

File	Max SND(F0)	fERL	tERL	maxACOM	Distortion
erlt20	43.9(2800)	3.4	3.4	33.4	Moderate
erlt10	45.5(1601)	3.4	3.4	33.9	Moderate
erlt03	42.4(1601)	3.4	3.4	34.1	Moderate

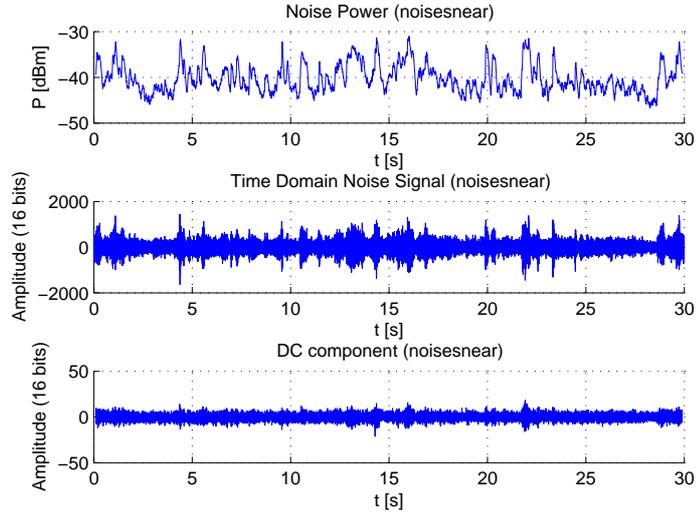


Figure A.8: Noise: Time domain values

A.3.3 Noise

It is self explanatory that the best choice of line probing tool to analyze noise is the noise analysis tool. The other two tools will not be used to analyze the test signals in this section. The plots and text files produced by the noise analysis tool will now be discussed. Figure A.8 shows time domain values of the noise signal. The top two plots show noise power and amplitude respectively. It can be seen that the power level is fluctuating around -40 dBm, which is large enough to be able to cause problems.

It has now been concluded that there exists noise at a fairly large level. More

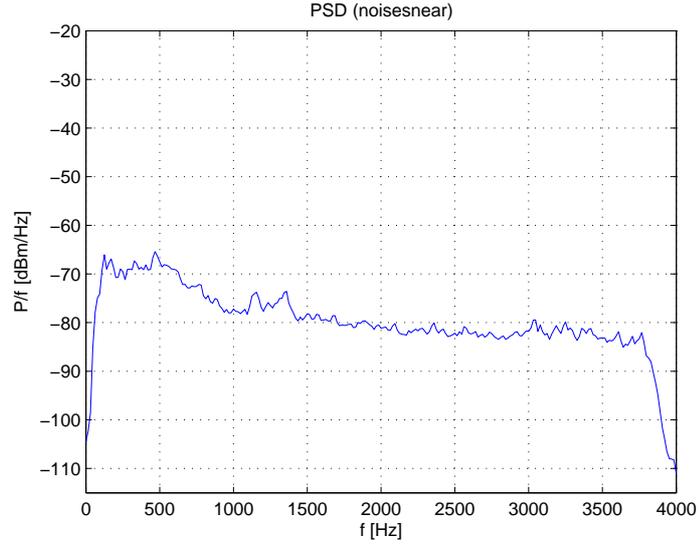


Figure A.9: Noise: PSD

information about the noise can be obtained from the second plot produced by the noise analysis tool. Figure A.9 shows the PSD of the noise. It can be seen that the spectrum of the noise has low pass characteristics. It can be concluded that the noise is not generated by a white process since the spectrum is not flat. It is not possible to conclude that the noise is office noise from the two plots. One needs to listen to the noise to be able to conclude anything else about it.

The plots give a hint of what is going on with the investigated hybrid. It is useful to look at the text files produced by the tool to get more insight in the problem. The summary file in Table A.9 confirms that the noise power is large. The information obtained from the plots and the text file is good evidence that there is a problem with noise in this particular hybrid circuit.

Table A.9: Noise: summary file

```

Average power in the band [ 100,3400] Hz is -39.3 dBm
*****
*****
Units of time, frequency, Pn, and PSD are in s, Hz, dBm, and dBm/Hz respectively.
DC is a unitless quantity and is limited to that of 16 bit integers.

File          Min Pn(  t) Max Pn(  t) Avg Pn Min DC(  t) Max DC(  t) Avg
DC
noises        -46.7(28.3) -31.0(16.0) -39.2      1(11.1)   -21(14.3)
0

File          Min PSD(  f) Max PSD(  f) Avg PSD
noises        -111.1(4000) -65.4( 469)  -75.2

```

Bibliography

- [1] Steven Cherry. Seven myths about voice over ip. *IEEE, Spectrum*, 42(3):52–57, March 2005.
- [2] Hui Min Chong and H. Scott Matthews. Comparative analysis of traditional telephone and voice-over-internet protocol (voip) systems. pages 106–111. *Electronics and the Environment Conference Record. IEEE International Symposium*, 10-13 May 2004.
- [3] Monson H. Hayes. *Statistical Digital Signal Processing and Modeling*. Wiley, 1st edition, 1996.
- [4] Simon Haykin. *Adaptive Filter Theory*. Prentice Hall, 4th edition, 2002.
- [5] IEEE. *Voice Over Internet Protocol (VoIP)*, volume 90. *Proceedings of the IEEE*, Sep 2002.
- [6] International Data Sciences INC. Model 91 analog test set. <http://www.idpdata.com/m91spec.htm>, Feb 2007.
- [7] International Electrotechnical Commission. *Sound level meters*, 1st edition, 1979.
- [8] International Telecommunication Union. *ITU-T Recommendation G.168 Digital network echo cancellers*, 1997.
- [9] Andre Neumann Kauffman. An algorithm to evaluate the echo signal and voice quality in voip networks. Master’s thesis, University of Maryland, 2006.
- [10] B.P. Lathi. *Modern Digital and Analog Communication Systems*. Oxford University Press, Inc., 3rd edition, 1998.
- [11] B.P. Lathi. *Signal Processing & Linear Systems*. Oxford University Press, 1st edition, 1998.
- [12] Albert Leon-Garcia. *Communication Networks*. McGraw-Hill, 2nd edition, 2004.
- [13] Sanjit K. Mitra. *Digital Signal Processing, A Computer-Based Approach*. McGraw-Hill, 2nd edition, 2001.
- [14] Telecommunications Industry Association. *TIA Standard 912 IP Telephony Equipment*, 2002.