Telogy Software

AER Component

Acoustic Echo Removal Developer Guide

Applies to Product Release: 15.1
Publication Number: IPP-001187/Revision: A
Publication Date: March 2010

Texas Instruments Incorporated
20450 Century Boulevard
Germantown, MD 20874 USA

Copyright and Contact Information

Document Copyright

Publication Title: Acoustic Echo Removal Developer Guide
Publication Number: IPP-001187
Revision: A

© 1998-2010 Texas Instruments Incorporated
All Rights Reserved. Reproduction, adaptation, or translation without prior written permission is prohibited, except as allowed under the copyright laws.

Software Copyright

Product Name: AER Component
Product Release: 15.1

© 1998-2010 Texas Instruments Incorporated
All Rights Reserved.

Contact Information

General

20450 Century Boulevard
Germantown, MD 20874

Voice: 301.515.8580
Fax: 301.515.7687

Web: www.ti.com (Broadband/Voice over IP)

Sales Information

E-mail: mktgsupport@list.ti.com

Applications Engineering

For Registered Customers Only:
E-mail: tech_support@ti.com

The Telogy Software Applications Engineering group is available to all customers who need technical assistance with a Telogy product, technology, or solution. Inquiries are categorized according to the urgency of the issue, as follows:

Priority Level 4 (P4) —You need information or assistance about Telogy product capabilities, product installation, or basic product configuration.

Priority Level 3 (P3) —Your network performance is degraded. Network functionality is noticeably impaired, but most business operations continue.

Priority Level 2 (P2) —Your production network is severely degraded, affecting significant aspects of business operations. No workaround is available.

Priority Level 1 (P1) —Your production network is down, and a critical impact to business operations will occur if service is not restored quickly. No workaround is available.

Notices and Trademarks

Important Notice

Texas Instruments Incorporated reserves the right to make changes to its products or discontinue any product or service without notice, and to advise customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied upon is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgement, including those pertaining to warranty, patent infringement, and limitation of liability.

Customers are responsible for their applications using Texas Instruments Software.

Notice of Proprietary Information

Information contained herein is subject to the terms of the Non-disclosure Agreement between Texas Instruments Incorporated and your company, and is of a highly sensitive nature and is confidential and proprietary to Texas Instruments Incorporated. It shall not be distributed, reproduced or disclosed orally or in written form, in whole or in part, to any party other than the direct recipients without the express written consent of Texas Instruments Incorporated.

Telogy Software, VLYNQ, PIQUA Software, wONE, PBCC, Uni-DSL, Dynamic Adaptive Equalization, Telinnovation, TurboDSL Packet Accelerator, interOps Test Labs, TurboDOX, and INCA are trademarks of Texas Instruments Incorporated.

All other brand names and trademarks mentioned in this document are the property of Texas Instruments Incorporated or their respective owners, as applicable.

Preface

About this Document

This document is in support of the Texas Instruments Acoustic Echo Removal Software Component release 15.1. It was originally written for IP Phone product release 12.0. For AER 15.1 release, all IP phone related information are stil kept in this document.

Document Conventions

This document uses the following conventions:

- Commands and keywords are in **boldface** font.
- Arguments for which you supply values are in *italic* font.
- Terminal sessions and information the system displays are in `screen font`.
- Information you must enter is in **`boldface screen font`**.
- Elements in square brackets ([ ]) are optional.

Notes use the following conventions:

NOTE—Means reader take note. Notes contain helpful suggestions or references to material not covered in the publication.

The information in a caution or a warning is provided for your protection. Please read each caution and warning carefully.

CAUTION—Indicates the possibility of service interruption if precautions are not taken.
WARNING—Indicates the possibility of damage to equipment if precautions are not taken.

Document Revision History

Release	Chapter	Description of Change
15.1		•
12.0	All	• Added Dynamic Range Compression (DRC) content • Added Frequency Domain Nonlinear Processing (FDNLP) and Adaptive Spectral Noise Reduction (ASNR) feature descriptions • Added new Automatic Gain Control (AGC) parameters • Updated block diagrams and configuration flag tables as necessary
11.3	Configuring and Optimizing Parameters for Acoustic Echo Removal	• Corrected the order of the bandsplit events (items 48 and 49) in the ec_debug table
11.2/B	Configuring and Optimizing Parameters for Acoustic Echo Removal	• Added AER/PIQUA Performance Statistics
11.2/A	Guidelines for Designing and Testing IP Phones	• Added summary table "Considerations for IP Phone Enclosure Design" • Compliance with TIA\EIA-810A updated to TIA\EIA-810B (approved 2006) • Added section "Testing a Speakerphone Enclosure"
	Configuring and Optimizing Parameters for Acoustic Echo Removal	• Added rx/tx signal limiters to AER block diagram • Added section "Four Linear Gain Switching Slew Rates" • Added section "Gain Splitting Time Constant" • Added section "Signal Limiter" • Added section “aer_nlp_ramp_scale Bit”
11.1.1	Guidelines for Designing and Testing IP Phones	• Added guidelines for reducing problems from wind noise • Added guidelines for maximizing full-duplex performance in handsfree mode
11.1	Guidelines for Designing and Testing IP Phones	• Clarified pass-band loss for “Response of Recommended Microphone Hookup”
	Configuring and Optimizing Parameters for Acoustic Echo Removal	• Added bandsplitting to AER components block diagram • Updated the AER Configurable Flags diagram • Updated the list of AEC/AER debug commands • Added the procedure AEC Adaptive Tail Model Convergence Test • Updated the Procedure 2-1 “Checking the Setup” • Updated the system response table for ec_debug_stat with bandsplit entries • Added parameters and values for bandsplitting to Control Bitfield 1 table • Added the AER control parameters [bs_himatchlo_on|hs_himatchlo_off]

Release	Chapter	Description of Change
11.0	Guidelines for Designing and Testing IP Phones	• Updated "Optimizing Handset Performance" with techniques for increasing EM shielding • Added information about shielding microphone wires to reduce noise in handsfree enclosures • Added guidelines for designing microphone traces on circuit boards
	Configuring and Optimizing Parameters for Acoustic Echo Removal	• Updated the block diagram and descriptions of AER components • Added comfort noise generator (CNG) parameters • Added NLP linear attenuation parameters • Updated graphic of NLP Center Clipper • Added descriptive section "Comfort Noise Generator (CNG)" • Added descriptive section "Configurable NLP Attenuation Limits" • Updated the section "Hardware Codec Requirement" • Added information throughout about the duplex stabilizer parameters and their effect on IP Phone full-duplex and partial-duplex performance • Updated the reference table of values for ec_debug_stat • Added the section Microcode Modifications with description of how to enable dimchan commands for AER configuration • Updated the Control Bitfield tables • Updated aerc options list
11.0 (Cont)	Configuring and Optimizing Parameters for Acoustic Echo Removal (Cont)	• Added diagrams of frequency response before and after equalization • Added High Level Compensation (HLC) feature description and parameters • Added a configuration procedure for HLC • Added command: aer_hlc_ctrl debug • Added new section Default Parameter Settings with tables listing default flags for AER/Noise Guard/EQ/HLC/AGC and optimized settings for 8 kHz and 16 kHz sampling frequencies
11.0	Configuring and Optimizing Parameters for Acoustic Echo Removal	• Added new command to AER Internal Parameters • Added description of new aer_cng_nlp_params command • Added command: aer_eq_ctrl • Added command: dsp <tcid> aer_gain_chg_params • Added description and procedure for AGC/AER gain change delay and interpolation count
	Signal Equalizer MATLAB Design Tool	• Added figure of the internal structure of the equalizer • Updated the list of Standards Masks delivered with the software • Updated the output examples of the equalizer tool to version 2.15
10.2	All	• First Issue in current format

Contents

Copyright and Contact Information ..... *ø-ii*
Notices and Trademarks ..... *ø-iv*
Preface ..... *ø-v*
About this Document ..... *ø-v*
Document Conventions ..... *ø-v*
Document Revision History ..... *ø-vii*
List of Figures ..... *ø-xii*
List of Tables ..... *ø-xiii*
List of Procedures ..... *ø-xiv*

Chapter 1

Guidelines for Designing and Testing IP Phones 1-1
1.1 Introduction to IP Phone Design and Testing 1-2
1.2 Enclosure Design Fundamentals 1-3
1.2.1 Key Design Elements of a Desktop Speakerphone 1-3
1.2.2 Enclosure Constraints on Echo-Free, Full-Duplex Quality 1-5
1.2.3 VoIP Increases the Need for Enclosure Quality 1-6
1.3 Component Guidelines 1-8
1.3.1 Hardware Codec Requirement 1-8
1.3.2 Speaker and Microphone Overall Loudness 1-8
1.3.3 Spectral Response Specifications 1-12
1.3.3.1 Primary Specifications 1-12
1.3.3.2 Additional Documents 1-12
1.4 Handset and Headset Applications 1-17
1.4.1 General Guidelines 1-17
1.4.2 Optimizing Handset Performance 1-17
1.4.3 Optimizing Headset Performance 1-19
1.5 Handsfree Application 1-20
1.5.1 Minimizing Enclosure Echo Magnitude 1-20
1.5.2 Minimizing Enclosure Echo Nonlinearity 1-20
1.5.2.1 Linearity and Signal Conditioning at Low Frequencies 1-20
1.5.2.2 Linearity Issues 1-22
1.5.3 Choosing and Housing the Speaker for a Speakerphone 1-25
1.5.4 Choosing, Wiring, and Housing the HF Microphone 1-29
1.5.4.1 General 1-29
1.5.4.2 Microphone Hookup Circuit 1-33
1.5.4.3 Third-order RC Filter 1-34
1.6 Design Guidelines for Phone Enclosures 1-35
1.7 Testing a Speakerphone Enclosure 1-36
1.7.1 Test Description and Printed Wiring Board Preparation 1-36
1.7.1.1 Board Modifications 1-36
1.7.1.2 Audio Interface Test Cables 1-37
1.7.2 Test Set Up 1-38
1.8 Diagnosing Problems External to the Near-End Phone 1-40
1.9 Preparing for Performance Assessment 1-41
1.9.1 Test Equipment Checklist 1-41
1.9.2 Testing Environment Checklist 1-41
1.10 Configuring the IP Phone for Performance Assessment Tests 1-43
1.11 Performance Assessment Tests 1-45
1.11.1 Determining the Sum of Send Path Gains 1-45
1.11.2 Testing the Speaker Volume and Determining Receive Path Gains 1-46
1.11.3 Testing the ERL 1-47
1.11.4 Measuring Harmonic Distortion and Reducing Echo Nonlinearity 1-47
1.11.5 Measuring Fundamental Harmonic Distortion 1-48

1.11.6 Measuring the Noise Level of the Handsfree Microphone....1-49
1.11.7 Testing the Handset for Low-Level Nasal Exhale Noise....1-49
1.12 Equipment for Measuring Acoustical Response....1-50
1.12.1 Required Equipment....1-50
1.12.2 Equipment Substitutions....1-51
1.13 Referenced Suppliers and Products....1-53
1.13.1 Acoustic Systems....1-53
1.13.2 Adobe Systems, Inc....1-53
1.13.3 Bruel & Kjaer North America Inc. HQ....1-53
1.13.4 Head Acoustics, Inc....1-53
1.13.5 Microtronix Systems Ltd....1-53
1.13.6 MWM Acoustics, LLC....1-53
1.13.7 The Loudspeaker Design Cookbook....1-53

Chapter 2

Configuring and Optimizing Parameters for Acoustic Echo Removal 2-1
2.1 Introduction to AER Components and Parameters....2-2
2.1.1 Purpose....2-2
2.1.2 Scope....2-2
2.1.3 Microcode Modifications....2-3
2.1.4 AER Components....2-3
2.1.5 Parameters....2-4
2.1.5.1 Configuration and Control....2-4
2.1.5.2 Gains and AER-AGC Interactions....2-5
2.1.5.3 AER and AGC Configurable Control Flags....2-5
2.1.5.4 AER and AGC Configurable Parameters....2-6
2.1.6 Control Commands....2-20
2.3 Send and Receive Path Gain Adjustments....2-22
2.3.1 Setup Checking....2-22
2.3.2 Gain Calibration Procedure....2-23
2.3.3 Equalization....2-26
2.3.4 High Level Compensation....2-29
2.4 Optimizing Parameters for Handsfree Operation....2-30
2.4.1 AEC Adaptive Tail Model Convergence....2-30
2.4.2 Optimizing for Handsfree Full-Duplex Operation....2-31
2.4.3 Optimizing for Handsfree Half-Duplex Operation....2-34
2.4.4 AER Noise Guard....2-36
2.4.5 AER High Level Compensation (HLC)....2-36
2.4.6 Automatic Gain Control (AGC)....2-38
2.5 Optimizing Handset and Headset Parameters....2-40
2.6 NMM Commands Relevant to AER Performance....2-41
2.6.1 AER....2-41
2.6.1.1 AER Debug Statistics....2-41
2.6.1.2 AER Frequency Domain Adaptive Tail Model Coefficients....2-44
2.6.1.3 AER Control Using *dsp aerc*....*2-44*
2.6.1.4 AER Control Using *dsp aert*....*2-49*
2.6.1.5 AER Internal Parameters....2-50
2.6.2 NMM Commands External to AER....2-57
2.6.2.1 Gains and Signal Levels....2-58
2.6.2.2 AER HLC....2-59
2.6.2.3 AGC....2-59
2.6.2.4 AER Noise Guard....2-61
2.7 Default Parameter Settings....2-63
2.7.1 Default Settings of Configurable Flags for AER/ AER Noise Guard/ AER EQ/ AER HLC/ AGC....2-63
2.7.2 AER Optimized Parameter Settings for 8 kHz and 16 kHz Sampling Rates....2-64

Chapter 3

Signal Equalizer MATLAB Design Tool 3-1
3.1 Introduction to Equalization Using Software 3-2
3.2 Design of the Equalization Filter 3-3
3.3 Parameter Optimization 3-5
3.3.1 Measured Frequency Response 3-5
3.3.2 Target Frequency Response 3-6
3.3.3 Margin from Target Frequency Response 3-6
3.3.4 Sampling Frequency 3-6
3.3.5 IIR and FIR Requirements 3-7
3.3.5.1 Standard Name 3-7
3.3.5.2 Phone Mode 3-8
3.3.5.3 Direction 3-8
3.3.6 Roll-Off Specification 3-10
3.3.7 Extension of Lower Frequency Mask 3-11
3.3.8 Safety Margin from the Frequency Masks 3-11
3.3.9 Margin from 3 dBm0 Reference for Overflow Analysis 3-11
3.3.10 Random Number Generator State for Initial Conditions 3-12
3.3.11 Maximum Number of Iterations 3-12
3.4 Output 3-13
3.4.0.1 Notes 3-19

Chapter 4

Dynamic Range Compression (DRC) 4-1
4.1 Introduction 4-2
4.2 DRC Functionality 4-5
4.3 Configuring a Specific Set of DRC Parameters 4-7
4.4 Choosing Values for the DRC Configurable Parameters 4-9
4.5 DRC Time Constants 4-12
4.6 Multiband Compression Considerations 4-16
4.7 DSP DRC Configurable Parameter API Nomenclature 4-17

Appendix A

AER Module API Calls A-1

Glossary GL-1

Index IX-1

List of Figures

Figure 1-1 Spectral Weighting Used to Determine Overall Loudness. 1-10
Figure 1-2 Send Path Spectral Response Specifications for 8 kHz Sampling Rate 1-14
Figure 1-3 Send Path Spectral Response Specifications for 8 and 16 kHz Sampling Rates 1-15
Figure 1-4 Receive Path Spectral Response Specifications for 8 kHz Sampling. 1-15
Figure 1-5 Receive Path Spectral Response Specifications for 8 and 16 kHz Sampling 1-16
Figure 1-6 Handset with Angled Microphone Grille to Resist Wind Noise 1-18
Figure 1-7 Microphone Capsule and Puff Filter 1-18
Figure 1-8 Inside View of a Speakerphone Back Volume Chamber 1-26
Figure 1-9 Gasket Seals Random Air Leaks from the Back Volume Chamber. 1-27
Figure 1-10 View of an Installed Back Volume Chamber 1-28
Figure 1-13 Foam-Encased Microphone Isolates from Non-Airborne Vibrations 1-32
Figure 1-14 Microphone Hookup Circuitry with Passive Analog Band Pass Filtering 1-33
Figure 1-15 Response of Recommended Microphone Hookup 1-34
Figure 1-18 Audio Interface Test Cable 1-38
Figure 1-19 Microphone Port and Speaker Ports on the SDB. 1-39
Figure 2-1 AER Performance-Related Components. 2-3
Figure 2-2 AER and AGC Configurable Control Flags 2-5
Figure 2-3 AER NLP Center Clipper Action 2-9
Figure 2-4 Example of FDCC Center Clipper Rail Values in Different Frequency Ranges 2-17
Figure 2-5 Frequency Response Before Equalization 2-28
Figure 2-6 Frequency Response After Equalization. 2-29
Figure 2-7 Linear Attenuation Switching Slew Rates Reduction (k=10). 2-48
Figure 2-8 Calibration Method for Gain Change Parameters. 2-55
Figure 2-9 Gain Change Delay Too Short (AER Tx digital changes first). 2-56
Figure 2-10 Gain Change Delay Too Long (ADC PGA changes first) 2-56
Figure 2-11 Distorted Samples During a 1.5 dB Gain ADC PGA Change. 2-57
Figure 3-1 Block Diagram of the Equalization Filter 3-3
Figure 3-2 Internal Structure of the Equalization Filter 3-4
Figure 3-3 Plot 1—Responses at Measurement Frequencies Before Equalization. 3-16
Figure 3-4 Plot 3—Magnitude of the Equalizer Transfer Function at Measurement Frequencies 3-17
Figure 3-5 Plot 4—Pole Zero Plot of the Equalizer IIR (if order>0) 3-17
Figure 3-6 Plot 5—Zero Plot of the Equalizer FIR (if length >1). 3-18
Figure 3-7 Plot 6—Overall Group Delay of the Equalizer 3-18
Figure 4-1 Location of the DRC Module in the IP Phone Rx Path 4-2
Figure 4-2 DRC signal processing schematic. 4-3
Figure 4-3 Compressor Structure. 4-3
Figure 4-4 Example of DRC Multiband Low (Red), Mid (Green), and High (Blue) Filters. 4-5
Figure 4-5 Compressor Input Versus Output and Gain Target Gt = Gt(X).. 4-5
Figure 4-6 Exemplary Rx Equalizer Gain 4-11
Figure 4-7 Multiband DRC Peak Output Versus Low-Band Compressor Time Constants 4-14

List of Tables

Table 1-1 Key Design Elements of a Desktop Speakerphone 1-4
Table 1-2 Spectral Response Specifications 1-13
Table 2-1 Definitions of AER System Components 2-4
Table 2-2 AER and AGC Configurable Parameters 2-6
Table 2-3 Gain Transition Half Life 2-12
Table 2-4 Signal Limiter Parameter Values and Corresponding Saturation Levels 2-15
Table 2-5 Frequency Domain Nonlinear Processor Configurable Parameters 2-17
Table 2-6 Configurable AGC Parameters 2-19
Table 2-7 System Response to ec_debug_stat 2-42
Table 2-8 Parameters and Values in the Control Bitfield 0 2-43
Table 2-9 Parameters and Values in the Control Bitfield 1 2-43
Table 2-10 AER Performance Statistics 2-43
Table 2-11 Phone Modes 2-45
Table 2-12 Recommended Tail Length for Each Phone Mode 2-51
Table 2-13 System Response to agc_debug_stat 2-59
Table 2-14 AER Noise Guard Parameters 2-62
Table 2-15 Default Flags: AER/AER Noise Guard/ AER EQ/ AER HLC/ AGC Parameters 2-63
Table 2-16 AER Optimized Parameters for 8-kHz Sampling Frequency 2-64
Table 2-17 AER Optimized Parameters for 16-kHz Sampling Frequency 2-65
Table 3-1 Definitions of Equalization Filter Elements 3-3
Table 3-2 Standard Masks in the optfilter/stdmasks Directory 3-8
Table 4-1 Spreadsheet Example to Help Determine DRC Gain Table Parameters 4-10
Table 4-2 DSP DRC Configurable Parameter API Nomenclature 4-17
Table 4-3 DRC Limiter Configurable Parameter Nomenclature 4-18
Table 4-4 Configurable Parameters for Full and Multiband Compressor 4-20
Table 4-5 Configurable Parameters Determining Overall DRC Setup 4-23
Table A-1 DSP Commands Used in the *AER Developer Guide* and Their Corresponding AER API Calls A-1

List of Procedures

Procedure 1-1	Interconnecting the Phone Enclosure and SDB for Testing	1-38
Procedure 1-2	Setting Up for an IP Phone Performance Assessment	1-44
Procedure 1-3	Determining the Sum of Send Gains	1-45
Procedure 1-4	Testing the Speaker Volume and Determining Receive Path Gains	1-46
Procedure 1-5	Testing the ERL	1-47
Procedure 1-6	Measuring Harmonic Distortion and Reducing Echo Nonlinearity	1-47
Procedure 1-7	Comparing Harmonic Distortion with the Texas Instruments EVAL Unit	1-48
Procedure 1-8	Measuring the Noise Level of the Handsfree Microphone	1-49
Procedure 1-9	Testing for Low-Level Nasal Exhale Noise	1-49
Procedure 2-1	Checking the Setup	2-22
Procedure 2-2	Calibrating the Gain	2-24
Procedure 2-3	Getting Tx and Rx Equalizer Parameters	2-27
Procedure 2-4	AEC Adaptive Tail Model Convergence Test	2-30
Procedure 2-5	Optimizing a Handsfree, Full-Duplex Phone	2-32
Procedure 2-6	Optimizing a Handsfree Half-Duplex Phone	2-35
Procedure 2-7	Configuring AER High Level Compensation (HLC)	2-37
Procedure 2-8	Inducing a Gain Change by AGC	2-54

Chapter 1

Guidelines for Designing and Testing IP Phones

This document contains the following sections:

1.1 "Introduction to IP Phone Design and Testing" on page 1-2
1.2 "Enclosure Design Fundamentals" on page 1-3
1.3 "Component Guidelines" on page 1-8
1.4 "Handset and Headset Applications" on page 1-17
1.5 "Handsfree Application" on page 1-20
1.6 "Design Guidelines for Phone Enclosures" on page 1-35
1.7 "Testing a Speakerphone Enclosure" on page 1-36
1.8 "Diagnosing Problems External to the Near-End Phone" on page 1-40
1.9 "Preparing for Performance Assessment" on page 1-41
1.10 "Configuring the IP Phone for Performance Assessment Tests" on page 1-43
1.11 "Performance Assessment Tests" on page 1-45
1.12 "Equipment for Measuring Acoustical Response" on page 1-50
1.13 "Referenced Suppliers and Products" on page 1-53

1.1 Introduction to IP Phone Design and Testing

This chapter contains

- Guidelines for designing an IP Phone speakerphone enclosure for optimal acoustic echo reduction and duplex performance
- A method for modifying your printed wiring board (PWB) so you can test enclosure designs early in the product development cycle
- Procedures for tests to measure the quality of an IP Phone enclosure

This chapter describes recommended hardware design for achieving the best performance of an IP Phone using the Texas Instruments Acoustic Echo Remover (AER) software. The AER module can be configured for use as an Acoustic Echo Canceller (AEC). In this document, AER and AEC are used interchangeably.

This document suggests hardware implementations that minimize internal coupling between the speaker and microphone within the phone enclosure. This document does not address the problem of line echo that originates from a source external to the IP Phone. When an undesirable acoustic echo is heard on a phone, the phone causing the problem is at the other end of the line. Speaker-to-microphone coupling at the near-end (NE) phone transmits the voice of the far-end (FE) user back to the far-end, giving the acoustic echo a round trip transmission delay.

1.2 Enclosure Design Fundamentals

- "Key Design Elements of a Desktop Speakerphone"
- "Enclosure Constraints on Echo-Free, Full-Duplex Quality"
- "VoIP Increases the Need for Enclosure Quality"

1.2.1 Key Design Elements of a Desktop Speakerphone

The goals for good speakerphone performance are

- Good full-duplex performance during double-talk
- No audible echo under any conditions
- Minimal variations in the speech level during double-talk
- Minimal artifacts—such as crackling, modulation, howling, echo, or choppiness—during double-talk
- Consistent, high-quality sound from the speakerphone at all volume settings

Table 1-1 lists the key design elements and how each element affects the performance of a desktop speakerphone when it is implemented correctly or incorrectly.

Table 1-1 Key Design Elements of a Desktop Speakerphone (Part 1 of 2)

	Effects:	
Design Element	**When implemented correctly**	**When implemented incorrectly or omitted**
Speakerphone back volume chamber (See "Choosing and Housing the Speaker for a Speakerphone" on page 1-25)	• Minimizes speaker-to -microphone coupling which improves stability and depth of AEC convergence • Improves full-duplex performance • Improves speaker linearity • Improves loudness and frequency response of the speaker • Reduces sensitivity to the manufacturing process by minimizing unit-to-unit variability of acoustic performance parameters of the phone enclosure when operating in handsfree mode • Simplifies tuning of the AER parameters Sometimes it may be beneficial to use an external amplifier and a larger diameter speaker in addition to improving the back volume. Most published acoustic characteristics of speakers are measured with a back volume in place.	• Reduces voice quality • Increases potential for echo • Increases potential for nonlinear echo and distortion at low frequencies • Reduces full-duplex range considerably • Causes high variability in performance among manufactured units • Reduces the ability to achieve the published frequency characteristics of the speaker
Handsfree speaker external amplifier (See "Choosing and Housing the Speaker for a Speakerphone" on page 1-25)	• Improves speaker linearity at higher volume settings • Expands the full-duplex range to higher speaker volumes Sometimes it may be necessary to use a larger diameter speaker and back volume to experience the full benefits of an external amplifier.	At higher volumes: • Reduces full-duplex range • Increases potential for nonlinear echo • Increases potential for echo • Reduces voice quality • Fails to provide adequate loudness
Large diameter speaker (See "Choosing and Housing the Speaker for a Speakerphone" on page 1-25)	• Improves speaker efficiency and linearity • May expand the full-duplex range toward the higher speaker volumes To get the full benefits, consider using an external amplifier and back volume.	At higher volumes when external amplifier and/or back volume are not present: • Reduces full-duplex range • Increases potential for nonlinear echo • Increases potential for echo • Reduces voice quality • Fails to provide loudness If an external amplifier is used, be careful not to overload the speaker; smaller speakers may not be large enough to handle the power.
Speaker gasket (See "Choosing and Housing the Speaker for a Speakerphone" on page 1-25)	• Improves speaker efficiency by minimizing leakage • Reduces speaker nonlinear distortion • Improves full-duplex performance	• Increases potential for echo • Increases potential for nonlinear echo • Reduces full-duplex range considerably • Causes high variability in performance among manufactured units • Reduces the ability to achieve the published frequency characteristics of the speaker
Large microphone-to-speaker distance in a speakerphone (See "Choosing and Housing the Speaker for a Speakerphone" on page 1-25 and "Choosing, Wiring, and Housing the HF Microphone" on page 1-29)	• Reduces speaker-to-microphone coupling • Improves full-duplex performance as long as linearity of the design is good	• Introduces strong echo • Reduces full-duplex range considerably

Table 1-1 Key Design Elements of a Desktop Speakerphone (Part 2 of 2)

Design Element	Effects: When implemented correctly	Effects: When implemented incorrectly or omitted
Handsfree microphone casing (See "Choosing, Wiring, and Housing the HF Microphone" on page 1-29)	• Reduces the coupling between the speaker and microphone by damping mechanical vibrations from the speaker • Reduces ambient noise and improves send loudness of the phone • Improves full-duplex performance	• Increases potential for echo • Reduces full-duplex range • Increases potential for nonlinear echo • Decreases the signal-to-noise ratio • Can decrease the send loudness
Handset microphone puff filter (See "Optimizing Handset Performance" on page 1-17)	• Reduces wind noise and peak saturation during speech • Improves full-duplex performance when the remote end is in Handsfree mode • Improves voice quality and frequency response	• Reduces voice quality • Degrades full-duplex performance when the remote end operates in Handsfree mode
Handset microphone grille (See "Optimizing Handset Performance" on page 1-17)	• Reduces wind noise • Prevents excess humidity in the microphone chamber • Improves durability of the puff filter	• Reduces voice quality • Degrades full-duplex performance when the remote end operates in Handsfree mode
End of Table 1-1		

When designing a new enclosure, it is important to make the correct design choices to meet performance goals. If you are using an existing enclosure, it is important to review the design elements to help you determine if the desired performance goals can be achieved. Knowing the shortcomings of an existing design can help predict the physical limitations that might come into play when tuning the IP Phone parameters. Even the highest quality acoustic echo removal (AER) software cannot reliably mask or overcome the limitations of a flawed enclosure design.

For information about how to test a prototype speakerphone enclosure, see "Testing a Speakerphone Enclosure" on page 1-36.

1.2.2 Enclosure Constraints on Echo-Free, Full-Duplex Quality

During double talk—when people on a phone connection talk at the same time—only a phone that can transmit near-end speech and receive far-end speech at the same time *without attenuating either signal* is considered full duplex. Hardware design affects the degree to which a handsfree phone can be echo-free and full duplex. Even with the best AEC software, sound quality will suffer on a poorly-designed phone.

Texas Instruments cannot assure customers that AER software will overcome the deficiencies in an inferior hardware enclosure and make it sound better than a competitive handsfree phone with a superior enclosure. It is a fallacy that any half-duplex handsfree phone can be upgraded into a full-duplex phone by merely introducing good AEC software. The enclosure places limitations on duplex performance when the echo reduction is acceptable.

A microphone on a telephone enclosure picks up both near-end speech and echo. The echo is typically far-end speech that is reproduced by the speaker in the near-end enclosure, then picked up by the near-end microphone. To reduce this echo, AEC software processes the received far-end signal through an adaptive linear filter to predict and subtract echo from the near-end handsfree microphone signal.

Subtraction alone reduces the echo by an adequate amount only if the initial echo is small and the echo prediction is accurate. If the filter does not remove enough echo, the AEC must attenuate the far-end and near-end signals. However, as attenuation decreases the echo it also decreases the quality of duplex performance.

The Terminal Coupling Loss (TCL) is the difference in the receive and transmit echo signal power in dB. Higher TCL values indicate less echo. The TCL is the sum of the hardware enclosure Echo Return Loss (ERL) and the software AEC Echo Return Loss Enhancement (ERLE). The ERLE is the sum of the loss attained by AEC selectively cancelling the echo, ERLEcanc, and ERLEnlp: the sum of the signal attenuation in the receive and transmit signals that the AEC NLP (nonlinear processor) applies. Therefore,

TCL=ERL + ERLE = ERL + (ERLEcanc + ERLEnlp)

Solving for the amount of signal attenuation that indicates a deviation away from full duplex gives:

ERLEnlp = TCL – ERL – ERLEcanc

The minimal TCL adequate for telephony echo reduction varies with circumstances, but is typically 45 dB. The ERL is the amount of echo loss with AEC disabled, which is entirely enclosure-dependent. For a handsfree phone configured to meet TIA-810B specified nominal loudness ratings, the ERL values vary widely for different enclosure designs, ranging from –25 dB to 10 dB.

ERLEcanc fluctuates in time during AEC operation, normally between 0 and 42 dB. With a few exceptions, the most limiting factor restricting ERLEcanc is enclosure low-frequency nonlinearities. During double talk, ITU-T P.340 requires that ERLEnlp is 6 dB or less for a full-duplex, handsfree phone. The ERLEnlp equation indicates that for a fixed TCL value, every dB increase in enclosure echo (lower ERL) and enclosure nonlinear breakdown (lower ERLEcanc) translate into another dB deviation away from full duplex (higher ERLEnlp). AEC software cannot produce a value of ERLEnlp that is adequately large for echo reduction and adequately small for full duplex on an arbitrary enclosure. Handsfree performance is the result of a combination of the hardware enclosure and software AEC.

1.2.3 VoIP Increases the Need for Enclosure Quality

To achieve full-duplex performance in a VoIP network, the need to reduce the magnitude and nonlinearity of enclosure echo is even greater. Due to longer transmission delays in VoIP, echo becomes more audible and objectionable. This effect is counteracted by decreasing the echo which increases the TCL. As the ERLEnlp equation indicates, this results in higher ERLEnlp attenuation and lower duplex quality.

For example, a full-duplex handsfree enclosure with ERLEnlp=0 and TCL=45 dB will meet a TCL=52 dB requirement after increasing to ERLEnlp=7 dB, and full duplex is lost. TIA/EIA-810B requires that PSTN/ISDN Handsets (HS) and Headsets (HeS) have TCL≥45 dB for the PSTN/ISDN network, but TCL≥52 dB for VoIP. The requirement

for stricter echo reduction is due to the additional delays inherent to IP Phone operation. This crudely indicates that converting a phone with minimally adequate TCL from PSTN/ISDN to VoIP results in a 7 dB loss of duplex performance. A common fallacy is that echo free/full-duplex performance in a PSTN/ISDN network predicts an equal performance after the same phone is converted to operate in a voice over IP (VoIP) network.

The longer VoIP delay implications are not limited to handsfree mode. For good phones, the handset ERL is 45 dB and headset ERL is 40 dB. This means that AER is required for handset and headset operation. While the AEC cancellation (ERLEcanc) needed for handset or headset is normally much less than for handsfree, the handset and headset modes are still challenging. A smaller echo is more difficult to identify due to masking from near-end speech and near-end background noise. The echo dependence on receive path data that the AEC adaptively models changes rapidly as near-end users move their heads. Performance must be satisfactory when the handset speaker volume is at maximum and the handset is placed faced down on a hard surface.

AEC software will not eliminate handset and headset echo in all circumstances without problematic side effects. Therefore, it is warranted to give attention to handset and headset hardware for reducing echo magnitude and nonlinearity. Movement of the handset, headset, and user's head can make AEC lose convergence, causing the enclosure echo (ERL) to be exposed or a loss of duplex performance if AEC is configured to do so.

Generally, handsfree mode is the most challenging. Texas Instruments AEC software is designed to support both higher cost IP Phones that have full-duplex handsfree performance and IP Phones that can only have half-duplex handsfree performance due to enclosure limitations. A half-duplex phone will not transmit and receive audio signals simultaneously. However, if it becomes necessary to design a half-duplex handsfree phone to reduce the cost of the enclosure, the design recommendations in this document should still be pursued as fully as possible.

A half-duplex handsfree phone differs from a full-duplex handsfree phone only by the amount of attenuation applied to the subdominant path. The subdominant path is the path with lower speech levels, which is either the receive or the send path, after neglecting the echo contribution. For example, just after an intermittent reduction in receive path speech on a handsfree phone with a negative ERL and nonlinear echo, identifying a subdominant path change is difficult. Identifying the subdominant path during double talk or near double talk conditions will be done more accurately by the handsfree phone that can best discriminate far-end echo and near-end speech in the near-end microphone signal. This will be done best by phones that follow the recommendations given in this document.

1.3 Component Guidelines

- "Hardware Codec Requirement"
- "Speaker and Microphone Overall Loudness"
- "Spectral Response Specifications"

1.3.1 Hardware Codec Requirement

Texas Instruments recommends the TNETV1050 PG 3.1 internal hardware codec for TNETV105x-based IP Phones and the AIC22C for IP Phones that use TNETV1001. Among other issues, earlier versions of the TNETV1050 have a noisier ADC and earlier versions of the AIC22 have a noisier DAC.

1.3.2 Speaker and Microphone Overall Loudness

To ensure that phones have at least adequate audio fidelity, the speaker and microphone responses are subject to various standards, such as

- TIA\EIA-810B (recently updated from TIA\EIA-810A)
- TIA\EIA-920
- TIA\EIA-540
- IEEE.1329.1999
- IEEE.269-2002
- ITU-T P.310-313, 340-342
- IETS-300-245.3

For more information, see "Spectral Response Specifications" on page 1-12.

The standards prescribe the spectral response and overall loudness of the send (microphone) and receive (speaker) paths of the phone. Adhering to overall loudness requirements enhances interoperability with different types of phones and different phone modes. For example, the far-end user should not require a significant speaker volume adjustment when the near-end changes from handset mode to handsfree mode or calls from a different type of phone.

If phones are developed in isolation without regard for standards, a comfortable overall loudness may result from abnormally low gains in the transmit path and abnormally high gains in the receive path. Later, when the phones interoperate unsatisfactorily with other phones that better adhere to standards, it will be apparent that much time was wasted testing with badly adjusted gains.

This document does not contain all the information in the referenced standards, but extracts some essential concepts. The Receive Loudness Rating (RLR) gives the loss or attenuation in output acoustical power from the phone's speaker relative to input digital signal power in the receive path. The Send Loudness Rating (SLR) gives the loss in output digital signal power relative to input acoustical signal power in the send path.

Phone measurements can be made at the test facilities of Texas Instruments or others, such as MWM Acoustics. Making the measurements independently requires equipment that can be purchased from companies such as Microtronix, Head Acoustics, and Bruel & Kjaer.

For handsfree mode, RLR=16 dB and SLR=13 dB are required. For handset mode, RLR=2 dB and SLR=8 dB are required. Roughly speaking, this means for a handset to handset call, you hear near-end speaker sound 10 dB lower (8 dB+2 dB) than the sound at the far-end handset microphone. Meeting the targeted SLR and RLR values allows the proper selection of gains. For example, if a handsfree SLR=13 is targeted, and SLR=23 dB is measured, you must increase the sum of the gains in the transmit path by 10 dB. For connecting to the PSTN, PSTN analog transmission loss, and compensating loss plan gains may be required in either the IP Phone or the intervening network. See TIA/EIA-TSB122, IETS.ES.202.020, and ITU-T G.101 for loss plan information.

The handset sidetone should yield an STMR=18 dB acoustical loss from the handset microphone to handset speaker. The sidetone should be muted for handsfree operation. The sidetone signal goes to the speaker but is not represented in the receive path data AER uses to linearly predict echo. The sidetone reduces ERLEcanc, and is required only for handset and headset operation. The AIC2x analog sidetone is preferable, so the AIC2x digital sidetone should be muted. Texas Instruments gain control software automatically adjusts the analog sidetone gain to maintain a fixed STMR value when the microphone gain of the ADC PGA is changed.

One method to reduce the echo magnitude and nonlinearity is to cheat on the target SLR and RLR specifications by reducing the nominal default gains in the send or receive path. In general, the echo magnitude and nonlinearity increase as the loudness increases. While a limited reduction in gains may increase customer satisfaction, caution is advised. The customer may get a negative first impression due to weak audio loudness, or simply increase the speaker volume, negating any advantage of starting with low nominal default gains. Sometimes a phone manufacturer does not sell phones directly to the public, and the phones reach the market through an intermediate company that insists on adhering to standards that include loudness ratings. In this case, lowering gains is very restricted, and some of the ± 4 dB allowance for variations in SLR and RLR should be allocated for phone-to-phone variations. Given the large sensitivity tolerances of many handsfree microphones, one cost-sensitive but viable option for reducing the worst case echo is to calibrate the send path gain independently in every phone.

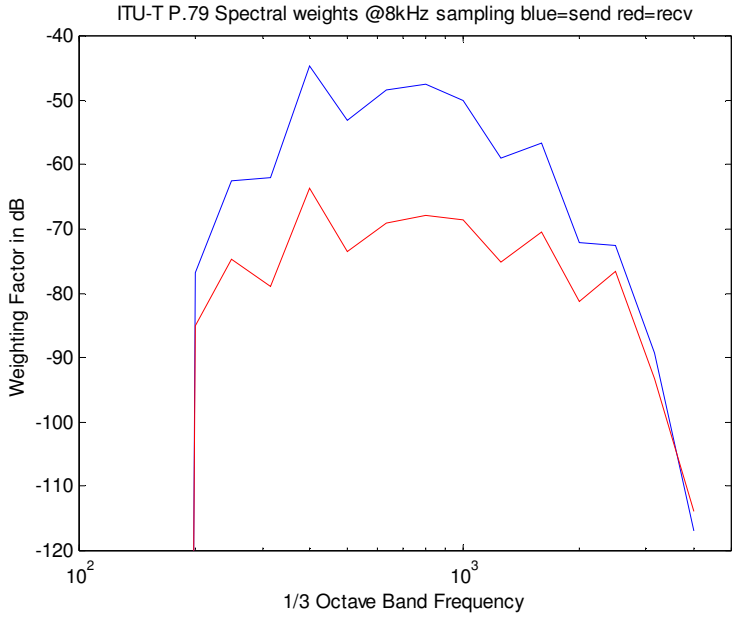
When restrictions on the standardized overall SLR and RLR loudness are not adhered to, lowering default gains can dramatically improve full-duplex performance. Perhaps raising the SLR and RLR equally above the recommended SLR and RLR would be a good approach, to maximize interoperability.

Other than getting better duplex performance (by lowering the ERL and nonlinearity), there are two additional reasons why this approach is worth investigating:

1. Texas Instruments and others in the field often consider the standardized SLR ratings to be too hot; normal conversation too often results in digital signal saturation. Even hotter signals result when a tired handset user leans against a handset. Contact between the mouth and handset can increase the send signal by about 10 dB.
2. The psychoacoustical loudness of a phone is hard to quantify and the ITU-T approach has shortcomings.

When restrictions on the standardized overall SLR and RLR loudness are adhered to, there is a significant motive to exploit the standards' spectral response tolerance. The idea is to design an enclosure that deviates from a flat spectral response in the speaker and microphone, and better matches the ITU-T P.79 weighting spectrum. This will result in meeting the specified RLR and SLR with lower gains, and reduce enclosure echo and nonlinearities. A phone's speaker and microphone spectral responses can have roughly a ± 5 dB tolerance relative to an ideal response. Tolerances given by specifications on spectral response are in "Spectral Response Specifications" on page 1-12. The ITU-T P.79 one-third-octave-band weightings, on which TIA and IETS also rely, vary a lot with frequency. For example, the 400 Hz one-third-octave band is weighted (to dominate the spectral average) much more than its neighboring one-third-octave bands. In the send path, 400 Hz is weighted 17.3 dB more than 315 Hz and 8.4 dB more than 500 Hz. In the receive path, the 400 Hz one-third-octave band also "counts" 15.7 dB more than 315 Hz and 9.8 dB more than 500 Hz. To met loudness standards, a phone with a speaker or microphone response that is relatively weak at 400 Hz may have to use significantly higher nominal default gains relative to a phone with a 400 Hz spectral response peak. Therefore, an enclosure spectral response deficiency at 400 Hz can indirectly lead to more echo magnitude, nonlinearities, and a loss of duplex performance. There is often a strong peak (or resonance) in the handsfree speaker spectral response at some frequency between 300-500 Hz. A drawback of enhancing the 400 Hz response of the handsfree speaker too much is that at this low a frequency there is vulnerability to nonlinearity (Figure 1-1).

Figure 1-1 Spectral Weighting Used to Determine Overall Loudness



Amplitude compression is an essential tool for limiting echo magnitude and nonlinearity on many enclosures; amplitude compression can be achieved in Telogy/TI Germantown software by hard limiting and soft limiting. This is significant regardless of whether adherence to standards is important. Additional information is provided regarding relevant specifications. Compression is a mixed topic that does not cleanly fit into either hardware enclosure or AER parameter optimization. Suppose a hardware

enclosure design can be made that is extremely linear all the way up to the final 3 dB of the digital dynamic range in the receive path input, but experiences a harsh nonlinear breakdown for signals above this level. Thus, a breakdown in ERLEcanc occurs most often when the far-end speech is abnormally loud. Should the entire enclosure be redesigned for better linearity? Perhaps it should, but it is likely that the best performance/cost trade off will be obtained by limiting the digital dynamic range in the receive path. For this example, the hard-limiting solution is to increase the AER receive path digital gain by 3 dB and decrease the receive path handsfree speaker DAC PGA analog gain by 3 dB. This eliminates the top 3 dB of digital dynamic range, but does not affect the overall RLR loudness. The TIA-810B handsfree RLR test uses a –22.8 dBm0 test signal, which a 3 dB digital gain will not saturate. In practice, the sound quality often seems adequate even after additional flat-top saturation is introduced. We are accustomed to hearing distortion from both saturation and nonlinearity, and hard limiting increases the former while reducing the latter.

Another option to eliminate the top 3 dB of the Rx path digital dynamic range is to configure the AER HLC to target a level of 0 dBm0 or less (3 dBm0 corresponds to a full-scale digital sine wave.) With HLC, you can achieve soft limiting and avoid flat-top saturation distortion when attenuation is required.

The drawback to soft limiting with HLC is that after someone yells, it may take time to recover normal gains during subsequent weak speech. Also, the volume button seems less responsive when HLC is counteracting higher Rx digital gains with increased Rx path attenuation (to prevent Rx path flat-top saturation.) In practice, the best solution may be to combine some hard and soft limiting.

The TIA-810B standard includes a handsfree speaker distortion test at nominal default gains and a receive path digital signal that is 6 dB down from full scale. To avoid failing this test due to flat-top saturation distortion introduced in the Rx path, this test requires hard limiting to eliminate no more than 6 dB of the high end of the input Rx digital dynamic range.Therefore, at the nominal default gain level, limit the AER receive path digital gain to 6 dB if meeting this standard is important. At handsfree speaker volume levels above the nominal default, TIA-810B currently does not test distortion, but does test the overall RLR loudness test at maximum volume, again using a –22.8 dBm0 signal. In some circumstances, compression is desirable. For example, the near-end handsfree speaker volume is turned high due to a soft spoken far-end person. When intermittent room noise, yelling, or a louder participant suddenly comes on, compression reduces the severity of the jolt. A good preliminary, if not final approach, is to optimize the amount of compression at only one handsfree speaker volume level. Given that an AER digital gain above 0 dB is used (rendering some compression), all handsfree speaker volume levels above that are reached by only increasing the AER digital gain. At a handsfree speaker gain X dB higher, there would be X dB additional compression. For lower handsfree speaker volume levels, lower the AER digital gain to 0, and then only use the DAC PGA analog gain to further reduce the handsfree speaker volume. An AER receive path digital gain below 0 dB degrades the receive path response SNR.

This incremental hard limiting compression approach (increasing volume by increasing Rx digital gain only) saves time during testing because all the higher volume levels share the worst case echo magnitude. Although compression is a cost-effective tool to eliminate nonlinear breakdowns, keep in mind that the audio quality is improved by making louder output accessible, designing out nonlinear breakdowns, and adding as little compression as possible. For customers that are hearing impaired or using large conference rooms, reaching higher sound levels is important. Other than adding distortion in the direct voice path, another compression drawback for hard limiting is that ERLEcanc can be reduced. Given that additional nonlinear breakdown is not a competing issue, just when the speaker has maximum displacement and the echo is maximum, sporadic flat-top saturation can cause a sudden changes the far-end spectrum that the echo canceller has optimally converged on. Therefore, using HLC to achieve soft limiting has distinct advantages.

1.3.3 Spectral Response Specifications

- "Primary Specifications"
- "Additional Documents"

1.3.3.1 Primary Specifications

The primary specifications include the following:

- TIA-EIA.IS-810-B, TIA-EIA.504 (North American Standards)
- ITU-T P.340, P.341, P.342 (International Standards)
- I-ETS.300.245-3 (European Standard)
- IEEE.1329-1999 (Measurement Standard)
- IEEE.269-2002 (Measurement Standard)



NOTE—In 2006, TIA-810A was replaced by TIA-810B. TIA-810B does not permit Type I IEC-318 sealed ear test head measurements. A HATS with Type 3.3 ear and pinna with Shore hardness 35 is strongly recommended by TIA-810B.

A "leaky ear" Type 3.2 and 3.4 HATS is required for handset receive path testing. This type of testing favors handset receivers with a low acoustic impedance driver. Most commercial handsets have a bass response that will be measured as much weaker by a HATS leaky ear.

1.3.3.2 Additional Documents

The primary specifications often rely on additional documents for definitions. Important peripheral documents include the following:

- ITU-T G.113
- ITU-T P.51, P.56, P.57, P.58, P.64, P.79, P.310, P.360, P.561

1.3.3.2.1 SLR and RLR

Among other things, these documents specify

- Send Loudness Rating (SLR)
- Receive Loudness Rating (RLR)

The SLR specification determines the total gain needed in the send path that transmits the microphone signal. The overall SLR quantifies how acoustical sound pressure levels at a mouth reference point are converted into a (frequency domain weighted) transmitted digital signal level.

The RLR specification determines a nominal default RLR value, and the range above and below that value, that should be accessible using the speaker volume control. The Texas Instruments IP-phone microcode sets gain levels for the Texas Instruments Reference Design Board (RDB) platform IP Phone enclosure to pass TIA/EIA-810B and TIA/EIA-920.

Table 1-2 lists the spectral response specifications by the respective standards organizations. The figures in this section include only ITU-T wideband standards; other important wideband standards are TIA-920, IETS-300-245-5 (HS), and IETS-300-245-6 (HF).

Table 1-2 Spectral Response Specifications

Sampling Rate/Phone Mode	ITU-T	TIA	IETS
8 kHz handset	P.310, P.313	TIA-810B	IETS-300-245-2
8 kHz handsfree	P.342		IETS-300-245-3
16 kHz handset	P.311	TIA-920A	IETS-300-245-5
16 kHz handsfree	P.341		IETS-300-245-6

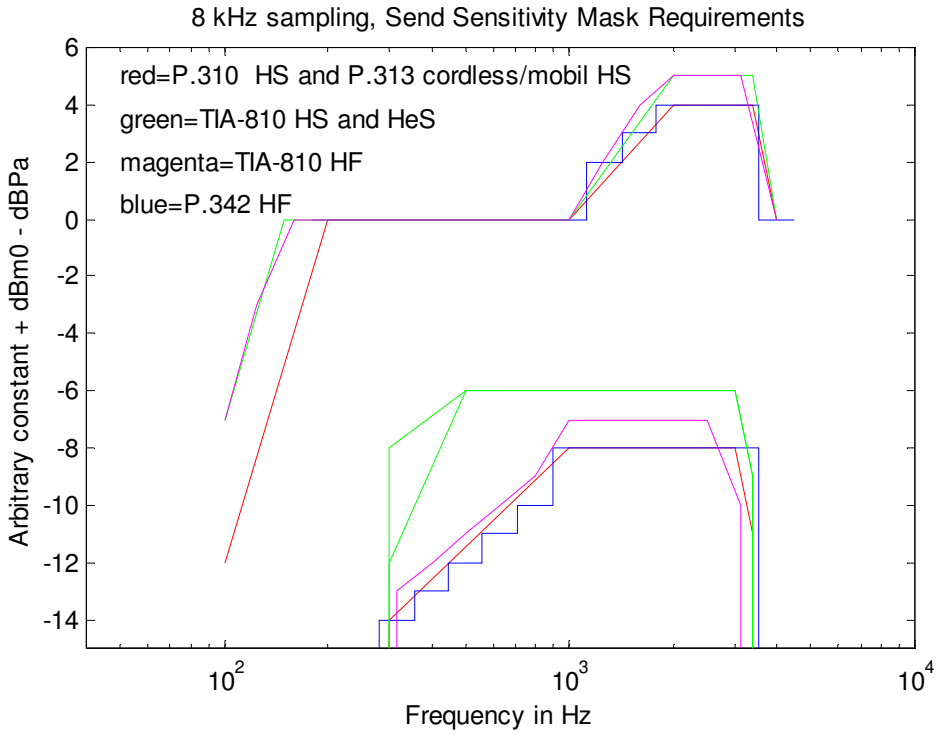
An important consideration for phone enclosure design is whether to support voice-coding software based on 8 kHz or 16 kHz sampling. A drawback of meeting the 16 kHz sampling P.311 (HS) and P.341 (HF) specifications is that the additional lower frequency near-end transmissions can be problematic for the far-end handsfree phone, which may be limited in performance by low frequency nonlinearities. The following paragraph presents an enclosure design issue that should be considered even if the phone manufacturer has no interest in adhering to spectral response specifications. After this, standardized response requirements are given.

The Texas Instruments RDB and SDB (software development board) support voice-coding schemes based on vocoders with 8 kHz sampling, such as G.711, and 16 kHz sampling, such as G.722. For the normal handsets (unlike the RDB handset) wideband G.722 audio quality relative to G.711 is not audible, to a large degree. This is because typical handset speaker and microphone spectral responses roll off sharply above 3.5 kHz. An appropriate wideband compatible handset is needed to evaluate G.722 in handset mode. Texas Instruments recommends sampling the MWM Acoustics wideband handset. If the enclosure spectral responses are deficient from 150-300 Hz and 3400-7000 Hz, the 16 kHz sampling enhancement may not be noticeable.

The ITU-T G.722 (16 kHz sampling) and G.711-712 (8 kHz sampling) specifications cover the hardware codec and software voice codec (encoder/decoder), but exclude audio components. For handsfree mode, ITU-T P.341 specifies the speaker/microphone responses for 16 kHz sampling, and P.342 covers similar material for 8 kHz sampling. P.341 requires a 150-7000 Hz response, and P.342 requires a 300-3400 Hz response. A microphone and ADC together convert sound in units

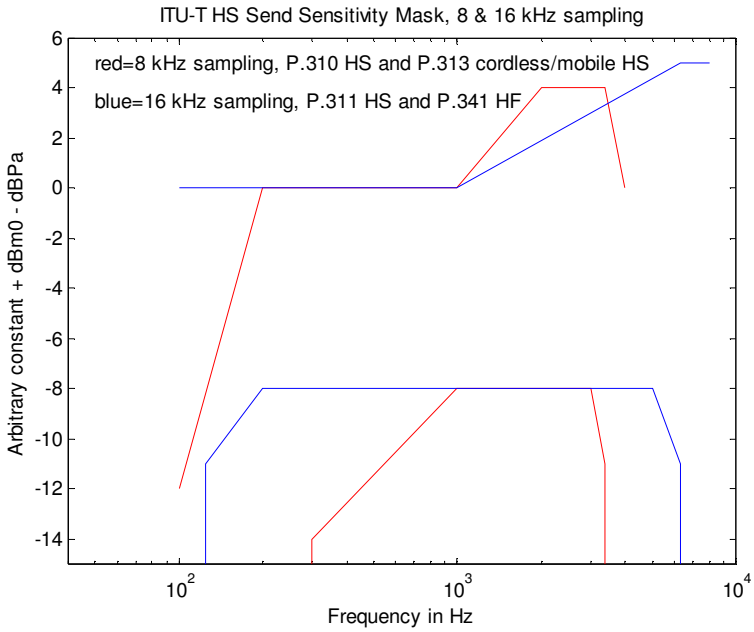
of dBPA (Paschal) to dBm0, a digital signal power unit. After adding an arbitrary constant to the response magnitude (y-axis) the frequency dependent response must lie between an upper and lower limit. Together, these limits are referred to as a mask. See Figure 1-2 and Figure 1-3 on page 1-15.

Figure 1-2 Send Path Spectral Response Specifications for 8 kHz Sampling Rate



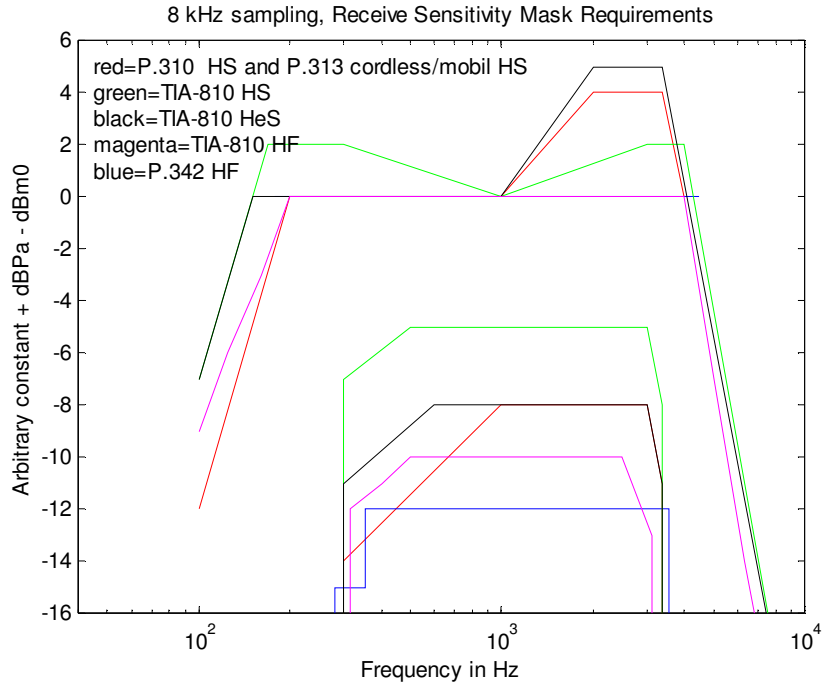
NOTE—The upper blue P.342 8 kHz handsfree mask was overwritten, and it has no low frequency roll off requirement.

Figure 1-3 Send Path Spectral Response Specifications for 8 and 16 kHz Sampling Rates



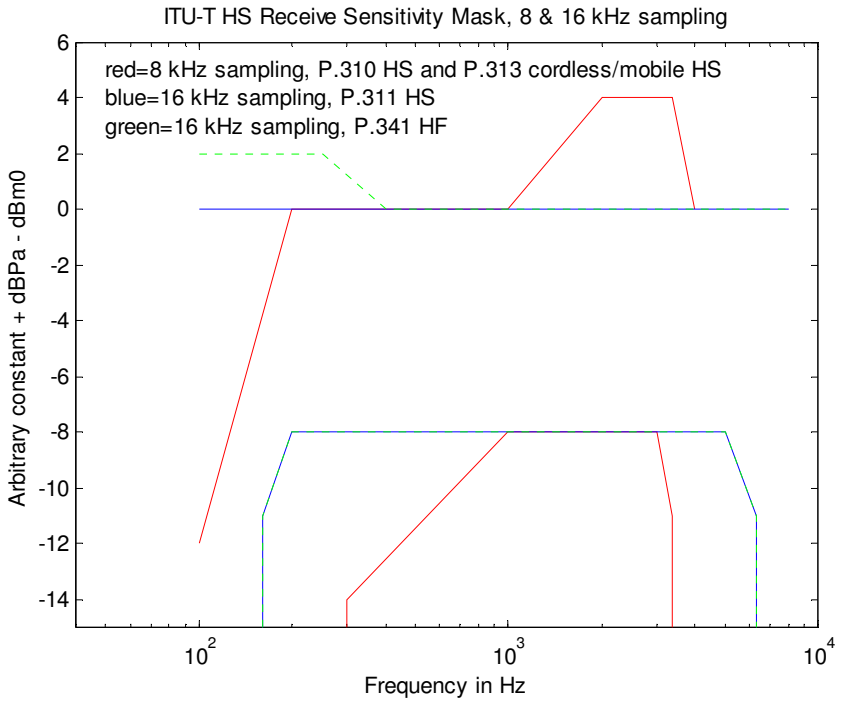
Perhaps a manufacturer could design one handset microphone with a response in the intersection of the 8 and 16 kHz sampling masks in Figure 1-3, but there is insufficient tolerance for phone-to-phone variation at 125 Hz. One solution is to use the AIC2x ADC optional HPF only for 8 kHz sampling. Simultaneously meeting handset speaker response specs for 8 and 16 kHz is less difficult. See Figure 1-4 "Receive Path Spectral Response Specifications for 8 kHz Sampling" on page 1-15 and Figure 1-5 "Receive Path Spectral Response Specifications for 8 and 16 kHz Sampling" on page 1-16.

Figure 1-4 Receive Path Spectral Response Specifications for 8 kHz Sampling



NOTE—The upper blue P.342 8 kHz handsfree mask was overwritten, and it has no low frequency roll off requirement.

Figure 1-5 Receive Path Spectral Response Specifications for 8 and 16 kHz Sampling



1.4 Handset and Headset Applications

For the optimal microphone hookup circuitry, see "Microphone Hookup Circuit" on page 1-33. Although the quality of microphone hookup circuitry is most critical for handsfree mode, Texas Instruments recommends the same microphone hookup for handset and headset modes. Headset mode also benefits from the removal of low-frequency noise from the signal.

1.4.1 General Guidelines

For handset and headset applications, observe the following guidelines.

- Use a four-wire cord.

  The IP Phone cord to the handset or headset should include two dedicated electrical connections for the speaker and two for the microphone. Introducing a 4-to-2 wire hybrid junction in the phone will cause line echo. Using a three-wire cord with the speaker and microphone sharing a common ground will also cause problems. The common ground may increase crosstalk echo, and will decrease power of the AIC2x differential speaker driver by 6 dB.

- Consider upgrading the cord design.

  Texas Instruments has tested phones using standard phone cords for which most of the dominant handset echo is not due to speaker-microphone acoustical coupling, but electrical coupling in a standard four-wire handset cord. Twisted pair, four-wire cords remove cord echo. For a standard cord, if the enclosure analog output port and the handset or headset wiring can be customized, map the cord wires so the differential speaker connections and hot microphone wire are on opposite sides of the microphone ground to significantly reduce cord echo.

- Minimize the acoustical speaker-to-microphone coupling and maximize the linearity of any coupling that does exist. Back up intuitive guessing on design improvements for these goals with real data.
- Check for DC current across the speaker; differential output can cause heating and nonlinear effects.

1.4.2 Optimizing Handset Performance

To optimize handset performance, observe the following guidelines.

- Increase the distance between the handset speaker and microphone to reduce echo.
- Make sure there is material in place to block any internal path between the speaker and microphone, and dampen plastic vibrations.
- Spray the inside of the chamber that houses the microphone with metallic spray to create a "Faraday cage" to help reduce EM noise pickup.
- Choose a microphone that has low sensitivity to wind, then optimize the handset to detect near-end speech as fully as possible. Make sure the mouthpiece has enough holes that are large enough in diameter. A small parabolic cavity can help focus sound to the microphone.

Figure 1-6 Handset with Angled Microphone Grille to Resist Wind Noise



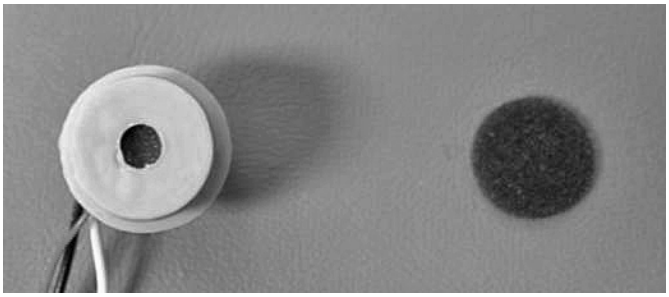
- Use a handset that resists detecting wind noise, and especially nasal wind noise.

NOTE—To minimize potential problems from wind noise, do the following:

  - Put a "puff filter" over the microphone of the handset to
    - » Reduce wind noise
    - » Increase full-duplex performance between the handset and a handsfree phone
  - Make sure the holes or slots in the grille of the mouthpiece of the handset are oriented in such a way that they do not expose the microphone directly to the opening.

  The holes in the handset in front of the microphone should point toward the mouth to reduce nasal exhale wind. A thin foam or fabric "puff filter" between the holes in the handset and the microphone can further reduce nasal wind from reaching the microphone. The microphone may need this surrounding "wind screen" foam to reduce wind noise.

Figure 1-7 Microphone Capsule and Puff Filter



07-00048

Some handsets exceed the specified high frequency response. A ceramic or film capacitor wired in parallel with the microphone element can add RC filtering to flatten the high frequency response. The puff filter can also dampen high frequency resonance due to the chamber housing of the microphone.

1.4.3 Optimizing Headset Performance

To optimize headset performance, observe the following guidelines.

- Use a binaural headset; the user will require less sound in each ear and thus generate less overall echo.
- Use a directional or pressure gradient “noise cancelling” microphone. This normally reduces unwanted transmission of external noise, yields less echo, and helps AEC converge better.
- Design an earmuff around the headset speaker to reduce echo and reduce masking background noise that otherwise encourages use of a higher gain.
- Allocate a specific hardware port for the headset. The headset speaker and microphone paths require different gains than the handset. Switches for rerouting the handset and headset to the same hardware codec analog I/O port can add unjustified costs to the product. The AIC2x and TNETV105x provide distinct ports for handset, headset, and handsfree operation.

1.5 Handsfree Application

- "Minimizing Enclosure Echo Magnitude"
- "Minimizing Enclosure Echo Nonlinearity"
- "Choosing and Housing the Speaker for a Speakerphone" on page 1-25
- "Choosing, Wiring, and Housing the HF Microphone" on page 1-29

1.5.1 Minimizing Enclosure Echo Magnitude

An enclosure is improved by a modification that increases the ERL and decreases ERLEcanc by equal amounts. For example, if the ERL is increased by 6 dB, there is less echo left to cancel; this can decrease the ERLEcanc values by 6 dB. Because the ERL increase is permanent, the handsfree performance will improve.

There will always be circumstances for which AEC loses convergence and temporarily yields a 0 dB ERLEcanc. When this happens, AEC typically does one of the following:

- Increases ERLEnlp and degrades duplex performance
- Transmits too much echo

1.5.2 Minimizing Enclosure Echo Nonlinearity

After optimally reducing the enclosure echo magnitude, the next priority is to examine what echo remains and reduce its most detrimental component— nonlinear echo. A requirement for good AEC performance (high ERLEcanc) is linearity in the acoustic echo path that AER models linearly. The critical path starts at the handsfree speaker DAC, goes through the DAC PGA, handsfree speaker, handsfree microphone, ADC PGA, the handsfree microphone ADC, and ends after the Tx path equalizer, if it was enabled. This critical path should be as linear as possible.

Choosing a handsfree speaker and microphone that have low distortion as isolated audio components does not guarantee optimal results. When the send and receive path are tested separately, distortion levels that are considered tolerable during single talk subjective listening tests cannot be tolerated in a full-duplex handsfree phone. The echo path accumulates both send and receive path distortion, and adds additional coupling distortion. The echo path will include distortion due to the handsfree housing, speaker-to-microphone vibration coupling, electronic coupling, etc. Turning AER off, setting the AER send path digital gain to zero, and recording handsfree transmissions for a later subjective *echo* listening test will be more productive.

1.5.2.1 Linearity and Signal Conditioning at Low Frequencies

After reducing echo nonlinearity as much as possible, the next priority is to examine what nonlinearity remains and to reduce the most detrimental type of echo nonlinearity—in-band harmonic distortion echo that comes from an out-of-band, low-frequency fundamental driving the handsfree speaker. Consider the response of the handsfree speaker alone, from its input voltage to output acoustic sound (ignore any high-pass filtering done upstream of the speaker in the Rx path.) A handsfree speaker with a flat response well below 300 Hz and no out-of-band low-frequency nonlinear breakdowns will yield better duplex performance. For an unoptimized handsfree enclosure, it is common for the handsfree enclosure to exhibit low-frequency echo harmonic distortion power in excess of the echo fundamental power. The worst design fallacy is that a 300 Hz to 3400 Hz pass band targeted for the handsfree speaker

response indicates that there is no problem using a small, inexpensive speaker with a weak bass response and a dramatic nonlinear breakdown below 300 Hz. The uninitiated will question why anyone would design a handsfree speaker with good bass response, then put it in a system that attenuates the low frequencies out of the handsfree speaker driving signal. Ironically, the overall handsfree performance is often severely limited by the handsfree speaker quality at out-of-band low frequencies. This section describes why this limitation exists. For this discussion, frequencies from 300 to 3400 Hz are considered in-band and frequencies outside this range are out-of-band and subject to attenuation by band pass filtering.

Conventional telephony band pass filtering of the near-end receive path handsfree speaker driving signal can increase linearity. This is because the handsfree speaker linearity normally breaks down first at low frequencies, so reducing the handsfree speaker driving signal to below 300 Hz will reduce echo nonlinearity. Although the far-end send path preferably does some band pass filtering, it is best not to rely on an external far-end phone for good near-end handsfree performance.

Because band-pass signal conditioning occurs in the near-end receive path or far-end send path, it is also good to apply similar band pass signal conditioning to the *send* path near-end handsfree microphone signal. Send path high-pass filtering

- increases coherence of the speaker and microphone
- band-limits the echo
- reduces extraneous low-frequency near-end sound that masks the echo that AER is trying to model
- attenuates nonlinearity at the fundamental frequency for low frequency out-of-band echo

If no harmonic distortion is introduced by the enclosure, the factors listed above increase ERLEcanc. The send path high-pass filter *increases* the percentage of echo that is nonlinear when the harmonic distortion is above the high-pass-filter cutoff and the fundamental is below the cutoff. Removing the send path high pass filter to deal with this issue forfeits the opportunity to exploit the four favorable factors listed above and provokes more low-frequency nonlinearities in the far-end handsfree phone.

For 8 kHz sampling, it is common for lots of signal power in the input receive path to be well below 300 Hz. Even if the far-end send path adheres to standards, only a limited amount of roll-off is required well below 300 Hz, and none is required for handsfree mode (see Figure 1-2 on page 1-14 and Figure 1-4 on page 1-15). In the near-end send path, a high pass roll-off may come from digital filtering, analog filtering, or the rest of the handsfree microphone enclosure. Regardless of its origin, the send response drop below 300 Hz can degrade near-end handsfree AEC performance when the enclosure allows low frequency echo harmonic distortion.

For example, assume loud voiced male speech is at the far-end with a 130 Hz fundamental format. (The 120- to 170-Hz range is typical for voiced male speech.) Even after some high-pass filtering, there is a 130 Hz signal that reaches the handsfree speaker. Assume this low frequency signal drives the handsfree speaker cone to contact its positive displacement limitation. Although the 130 Hz fundamental dominates the speaker cone displacement, the handsfree speaker bass response is weak but the third

harmonic motion generates sound much more efficiently. Because the send path has a high frequency roll off, the echo fundamental will be further reduced relative to the harmonic distortion. Thus, at the near-end send path's AIC2x 170 Hz cutoff *high pass* filter, the acoustic echo's higher frequency harmonics are *amplified* relative to the fundamental. Assume the echo harmonic distortion is so bad at this point, that the harmonic distortion dominates the fundamental. When this occurs, what AER might perceive as far-end echo (the echo fundamental) is subdominant to what AER perceives as near-end speech (echo harmonic distortion). Then AER will be tasked with switching the AER NLP attenuation away from the send path and into the subdominant receive path. Therefore, AER may turn off the NLP attenuation in the send path. Then the nonlinear echo harmonics are not cancelled or attenuated, producing a burst of echo.

Fixing this echo break-in problem requires some combination of the following:

- adding high-pass filtering and compression in the receive path signal before the handsfree speaker
- redesigning the enclosure to reduce the low-frequency nonlinearity of the handsfree speaker
- redesigning the enclosure to boost the fundamental relative to the harmonic distortion

Using filtering to reduce the problems due to nonlinearity in the handsfree speaker at out-of-band low frequencies should be approached before the low frequency sound reaches the handsfree speaker, so the speaker cone displacement stays small and in a more linear regime. A drawback is that if the handsfree speaker already has a weak low frequency acoustical response, excessive high-pass filtering of its electronic driving signal can make the handsfree speaker sound worse and possibly fail standards.

1.5.2.2 Linearity Issues

Additional circumstances when linearity is violated include the following:

1. Signal saturation or compression can occur in the critical echo path.

   Digital saturation can occur in DSP software when a digital gain above 0 dB is applied. Because they exist outside the critical echo path, the loss plan and AER digital gains are not suspect here for Texas Instruments AER software.

2. Digital or analog saturation occurs in the hardware codec.

   The AIC2x DAC and ADC do digital filtering. Analog saturation can occur at the

   - DAC output
   - DAC PGA
   - external speaker audio amplifier
   - microphone
   - microphone ADC PGA
   - microphone ADC

   The AIC2x DAC PGA gain should not be set above 0 dB because there will be inadequate output voltage to linearly process a full-scale digital input signal. With the exception of external audio amplifiers, for the AIC2x controlled by Texas Instruments gain control software, this problem is addressed. There is

vulnerability if the dsp 0 acregw… command is used to poke in AIC2x DAC PGA gains above 0 dB. When connected to the 8-ohm (handsfree) analog output port, a maximum gain of –1.5 dB is recommended to provide a safety factor against flattop saturation occurring in the echo path.

3. The handsfree near-end microphone ADC can introduce nonlinearities two ways:
   - If the input signal is too high, saturation clipping or flat-topping occurs and prevents AEC from converging correctly.
   - If the input signal is too low, truncation effects become too dominant because of the limited resolution between discrete ADC digital outputs.

   Therefore, a good ADC/DAC digital dynamic range helps AEC. An appropriate, handsfree speaker volume dependent, ADC PGA gain can be initialized, and AGC can be relied on to ensure a healthy send path digital dynamic range. The Texas Instruments AGC automatically changes the AIC2x ADC PGA handsfree microphone analog gain to optimize the send path digital dynamic range that is input to AER. AGC accomplishes this without causing an SLR change or audible clicks. This makes the phone more robust for different environments. The AER HLC optimizes receive path digital dynamic range by adding Rx path attenuation when Rx input levels exceed a configurable threshold level.
4. The speaker, microphone, and amplifiers are occasionally pushed beyond their linear regime. A pre-AEC design may be too tolerant of nonlinear distortion.
5. Time-dependent variations are caused by inexpensive, nonlinear, and temperature-dependent electronic components that were fully adequate in a pre-AEC design.
6. A microphone muting option is activated or deactivated and AEC is not informed. AEC will deconverge.
7. There is a separate, unintegrated analog/digital AGC system that changes gains in the aforementioned path without informing AEC.
8. A near-end user adjusts some external variable resistor to change the near-end mechanical speaker volume level; this changes the analog signal level driving the near-end mechanical speaker but has no effect on the receive path data that is input to AEC. For the Texas Instruments IP Phone reference design, divergence is avoided because speaker volume changes are done through a PGA and AEC that is properly informed by the AGC.
9. The handsfree phone introduces a signal—for example, sidetone, comfort noise, or DTMF tones—that drives the near-end handsfree speaker but has no effect on the AEC speaker input data. Handsfree operation should not include a handsfree sidetone. A side-tone deteriorates AEC performance even if it is represented correctly in the AEC receive path data (generally it is not). Ignoring AEC optimization, the best quality sidetone is an analog sidetone.
10. A handset, headset, or other application has an active sidetone during handsfree operation. For example, assume the handset analog sidetone is on and the handset is in its cradle on top of the handsfree speaker location. The handsfree speaker acoustical output saturates the handset microphone, and this generates a nonlinear flat-topped sidetone in the handset speaker, which acoustically couples into the handsfree microphone. The microprocessor software should mute the

sidetone for all inactive applications. Not doing this is a common mistake due to the misconception that a digital phone handset sidetone dies if the handset is cutoff from the digital paths. The AIC2x sidetone does not require ADC or DAC digital data flow to be active.

11. The phone circuitry uses the electronic audio path for additional functions. For example, a low-frequency signal is added into the critical path to control some type of switch located elsewhere and connected to the same electronic path.
12. The entire phone system, in the absence of AEC, is not completely full duplex. The phone has a non-AEC means for some degree of muting to be applied to the near-end or far-end signal, according to which is subdominant. A legacy analog suppressor may have been left in the speaker or microphone circuitry. Even if such devices are outside the critical path for linearity, placing AEC and other unintegrated, independent mechanisms that are also capable of attenuating the signal paths often results in poor audio quality. Time-dependent attenuation mechanisms may be in the near-end phone, far-end phone, or intervening network. Examples include line echo suppressors or cancellers, noise filters, and voice activity detectors that stop transmitting during silence to save bandwidth.
13. The handsfree speaker amplifier and handsfree microphone DC bias are connected to a common voltage source and are not sufficiently decoupled.

    For example, the input DC voltage supplying power to the entire IP Phone may be low-pass filtered. (The low-pass filter may use ferrite beads or RC filtering with a tantulum capacitor to keep high-frequency switching noise out.) When the receive path signal is large and the handsfree speaker amplifier draws maximum current, the voltage across this low-pass filter may increase and lower the voltage on the handsfree microphone DC bias. For a full-scale square wave in the receive path, the AIC2x 8 ohm driver output power can go up to 780 mW, and the input current to this driver will be even higher.

    For example, if this type of coupling is occurring for a 0 dB ERL enclosure with 40 dB of gain in the send path and handsfree microphone bias voltage second harmonic distortion that is 70 dB down from the receive path voltage levels, ERLEcanc cannot exceed 30 dB.
14. All EM pickup or ground loops should be avoided. When a relatively small EM pickup is introduced into the near-end signal, AEC can break down. For example, suppose the near-end and far-end have EM noise at 60, 180, and 300 Hz. When no far-end speech is present, the ratio of the transmit to receive signal is dominated by the ratio of near-end EM pickup to far-end EM pickup. But during higher level far-end speech, the same transfer function is dominated by near-end acoustical echo coupling. Whenever the transfer function varies as a function of the far-end signal level, linearity is broken.
15. The enclosure vibrates with the handsfree speaker and may lack reinforcements and an optimal design to prevent buckling of entire surfaces, resulting in harmonic distortion. A conceptual mechanism for surface buckling is as follows.

    As the handsfree speaker drives the attached flat horizontal surface, the surface vibrates vertically at frequency f, but also stretches horizontally at frequency 2f, toward the point of maximum vertical vibration displacement magnitude. Whether the flat surface is at maximum or minimum vertical displacement (180 degrees out of phase), horizontal inward stretching occurs in the same direction (in phase). While horizontal vibrations of a horizontal surface do not efficiently

couple to airborne sound, a second surface attached at a right angle can efficiently generate sound at frequency 2f. It is typical for enclosure buckling to dominate the handsfree speaker distortion at lower frequencies, and thus be the dominant impediment to good handsfree performance. Typically, harmonic distortion due to enclosure buckling greatly exceeds the nonlinearity due to the handsfree speaker itself. Attaching a speaker to a concave or convex enclosure surface and reinforcing the enclosure around and behind the handsfree speaker can reduce harmonic distortion due to buckling. Thicker plastic in the area where the handsfree speaker attaches also helps. Adding stiffeners to the back volume chamber (Figure 1-8 on page 1-26) also reduces buckling. If the speaker attaches directly to the front panel, a gasket (that normally comes attached to the speaker frame) is advised; if the speaker frame is metal without an integrated gasket, the rigid surfaces of the speaker frame and enclosure panel can vibrate or rattle at various frequencies.

If the speaker grille surface is concave, additional clearance is added to generate sound more efficiently when a handset is in a cradle just over the handsfree speaker.

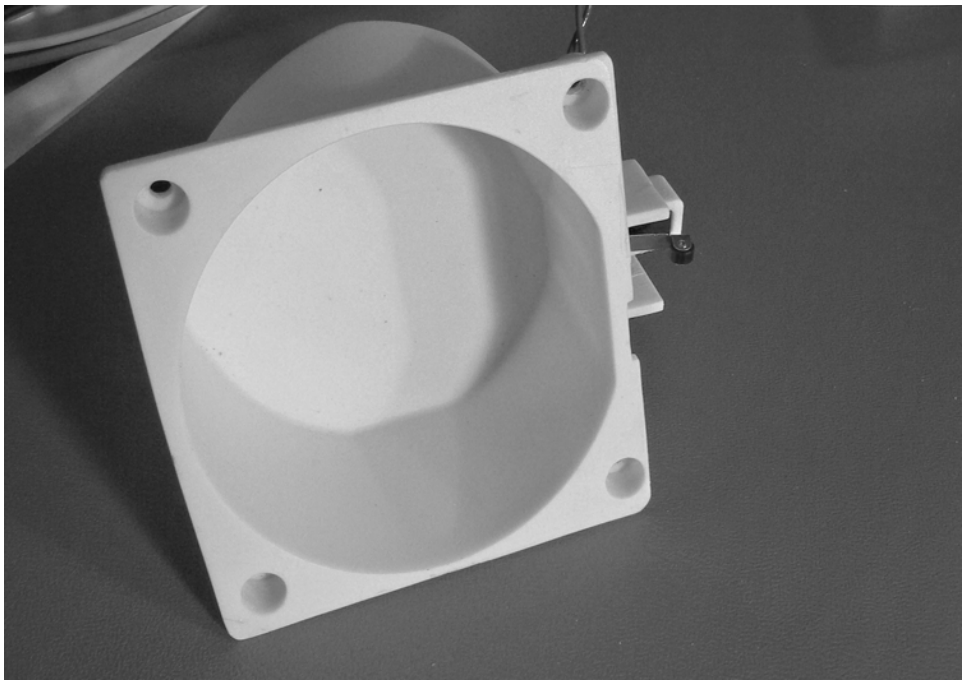
16. The phone has loose components that rattle, shake, shimmy, and add audible distortion when the handsfree speaker is driven hard. Often, driving a handsfree speaker at maximum volume at 150, 200, and 300 Hz will result in clearly audible and undesirable distortion. Normally, the AEC software only eliminates echo at the driving frequency cleanly; at all other frequencies the sound is detected by the near-end handsfree microphone and interpreted as near-end speech. Adding sound absorbing foam can help dampen unwanted plastic casing vibrations. For a handsfree speaker built into the handset cradle, pads to cushion the handset can reduce enclosure rattling. This also reduces the transmission of loud handsfree microphone noise when the handset is picked up or put down.

1.5.3 Choosing and Housing the Speaker for a Speakerphone

NOTE—To maximize full-duplex performance in handsfree mode, design the following into your IP Phone enclosure:

- The largest possible back volume of the chamber behind the handsfree speaker
- An external amplifier at the output of the DAC of the TNETV105x
- A high-efficiency speaker with the largest diameter you can fit in the enclosure

Figure 1-8 Inside View of a Speakerphone Back Volume Chamber



07-00045

Generally, speakers that are larger and rated for higher power are more linear. To ensure there is less need for digital path amplification (saturating high level input receive signals) or external audio amplification (that adds costs), the speaker should have an impedance of 8 ohms, a power rating of 2W or higher, and a sensitivity of 90 dBSPL/W/m or higher. This power rating better ensures the maximum AIC2x output of 390 mW is within the linear regime of the speaker. For a saturated input square wave, the AIC2x power may go up to 780 mW.

A full-scale digital sine wave going to the AIC2x DAC with the maximum DAC PGA gain (of 0 dB) produces a 3 dBm0 digital level, equivalent to a "virtual" voltage level of 0.8 dBV for TIA-810B units. The 90 dBSPL/W/m sensitivity approximately results in 96 dBSPL/W/0.5m. For an 8 ohm load and 96 dB/W/0.5m sensitivity, *in theory* 390 mW produces 91.9 dBSPL at 0.5 m or –2.1 dBPa and an RLR=0.8-(-2.1)=2.9 dB. This is 5.1 dB louder than the TIA-810B maximum handsfree speaker gain loudness loss recommendation, RLR=8 dB. It should be emphasized that in practice—especially when the handsfree speaker has a small, sealed, back volume chamber—the handsfree speaker sound can turn out to be 12 dB or more weaker than this rough RLR computation suggests.

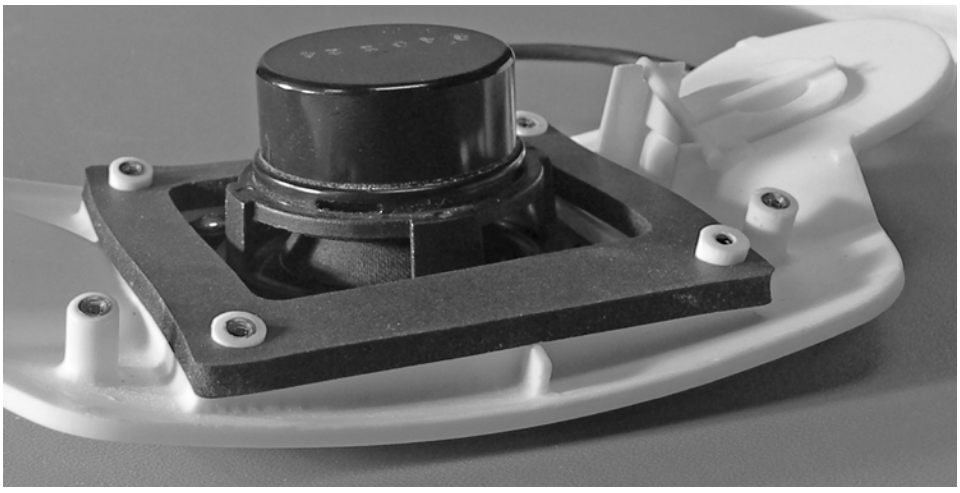
Continuing with another crude computation, assume the only source of echo is the acoustical speaker-to-microphone coupling that is airborne and external to the enclosure. Assume the handsfree speaker and microphone are 10 cm apart. A handsfree speaker and microphone located 10 cm from target/source (instead of 50 cm, as in the handsfree near-end user) will both become about 14 dB hotter. Then, at nominal default volume the calculation is as follows:

ERL=SLR-14+RLR-14 dB=13-14+16-14=1 dB

This is not overly optimistic, as there is no assumption that the handsfree microphone is in the acoustical shadow, rather than the direct path, of the handsfree speaker sound. At a maximum handsfree speaker volume that is 8 dB higher than the nominal default, the echo loss would be –7 dB. In practice, at maximum gains for the worst possible frequency, it is not unusual to have a 20 dB echo gain. Thus, the theory allows for significant improvement over what is often seen in practice. Some causes for this undesirable 13 dB discrepancy are as follows:

- Reducing echo at the worst-case frequency may involve "flattening out" a resonant frequency. You can use the AER Rx equalizer to flatten a speaker response; however, it is not recommended that the equalizer gain vary more than ±6 dB. To redesign the phone hardware to flatten the response or otherwise shape the frequency-dependent response of the speaker, you may need to consult an acoustics expert. This document describes some basic handsfree speaker design issues. For in-depth treatment of the subject, a recommended resource is "The Loudspeaker Design Cookbook" on page 1-53.
  - Adding a back volume chamber behind the speaker gives better control of the volume of space behind the speaker; the back volume can be chosen to flatten the handsfree speaker spectral response. Often the same narrow frequency band resonance that violates a standard's RLR or SLR mask also causes the poor AEC performance, when the receive path signal happens to have sufficient input in the resonant band.
  - If the air behind the speaker is free to move throughout the phone enclosure, this may result in an undesirable acoustic resonance. The back volume chamber also provides an additional barrier to block acoustical paths for speaker-to-microphone coupling inside the phone. To increase the effectiveness of the barrier, the surface on which the back volume chamber attaches to the enclosure should have a gasket (Figure 1-9) to stop random air leakage (sound leakage) from the chamber into connected spaces of the enclosure.

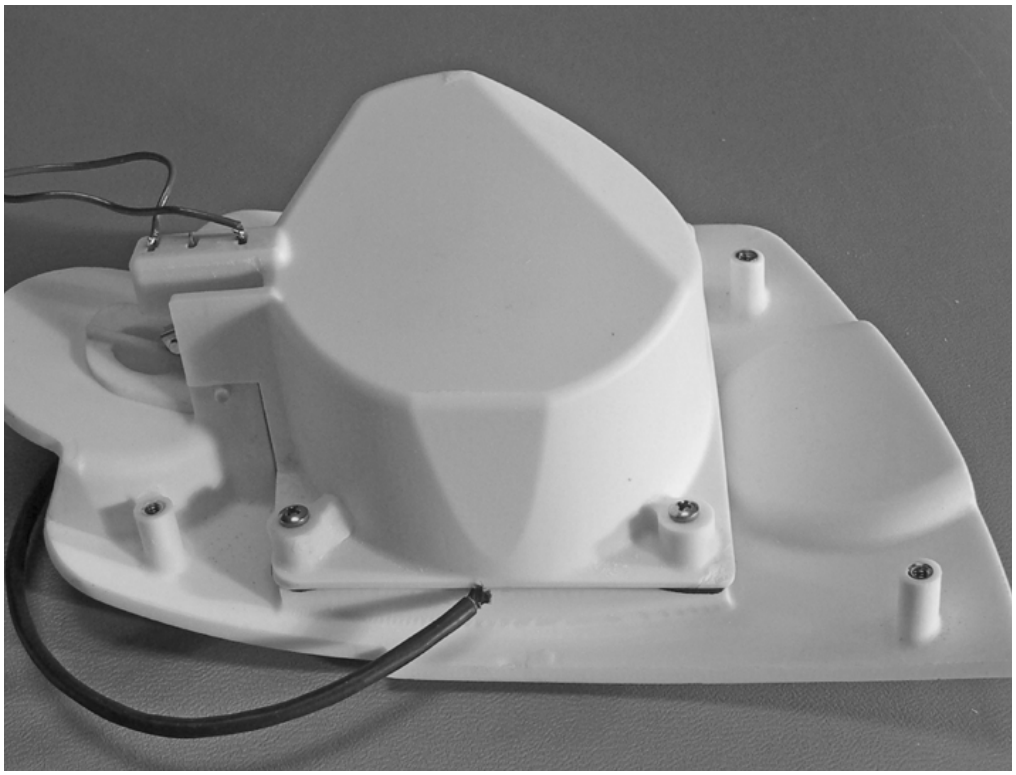
Figure 1-9 Gasket Seals Random Air Leaks from the Back Volume Chamber



07-00046

- Using a back volume chamber with an irregular shape—or at least a back wall not parallel to the front surface the speaker is attached to—decreases resonances by encouraging phase mixing of sound bouncing off the walls of the back volume chamber (Figure 1-10).
- Adding sound absorbing material behind the speaker can also help damp out a resonance.

Figure 1-10 View of an Installed Back Volume Chamber



- Making the handsfree speaker more efficient in channeling acoustic energy to the near-end user rather than the handsfree microphone will reduce the ERL. Greater efficiency yields handsfree speaker DAC gains, and this reduces distortion and vibration.

  Adding an acoustic exhaust (a small opening for air to escape from the back volume chamber) can make the speaker more efficient at generating sound. Direct the acoustic exhaust away from the handsfree microphone location. Instead of an exhaust, you can increase the amount of air trapped behind the handsfree speaker to increase its acoustical output.
- Front-surface enclosure vibrations are often encouraged to increase the sound output, especially when the speaker is small and inefficient. A drawback to this approach is that it generally makes consistently meeting a standard's frequency dependent RLR mask more challenging. Various spatial modes of plastic vibration have resonances and generate complex interference patterns that may vary from phone to phone. Another drawback is that the intentional vibrations of the enclosure may unintentionally couple to the handsfree microphone.

- Increasing the handsfree microphone efficiency at focusing on one sound from the near-end user relative to the handsfree speaker enhances the ERL. A general technique to decrease the ERL is tilting the entire enclosure to direct the handsfree speaker toward a target, given in standards at 40 cm out and 30 cm up. This flattens the response at the target, especially at high frequencies. Without tilting, sound from the handsfree speaker may be directed straight vertically. Especially at high frequencies, sound passes over the target external microphone, so a larger receive path gain is required to meet a target RLR. After tilting, a lower gain recovers the RLR=16 dB default and the echo is reduced. For a tilted phone, the handsfree microphone can be directed at right angles to the handsfree speaker, to focus on sound originating from the same target that bounces off a hard surface just in front of the phone.
- The coupling between the handsfree speaker and microphone may not be primarily due to external airborne sound, but due to vibration and electronic coupling. If the design is flawed, handsfree speaker generated sound inside the phone enclosure may couple to the handsfree microphone much better than the external sound.
- Making the pieces of the phone structure that are assembled together fit tightly reduces spurious rattling. The circuit board should be firmly attached.
- Adding mass (weight) reduces vibrations. A heavier speaker magnet is good because the speaker paper cone diaphragm moves, but the magnet and frame stay relatively fixed.
- At least four soft rubber feet should be on the four bottom corners of the phone. Soft, hemispherical rubber feet have proven to have the best traction and absorb vibrations better. Soft rubber supports under the bottom of the phone may help dissipate speaker-induced vibrations and shield the microphone from table vibrations.

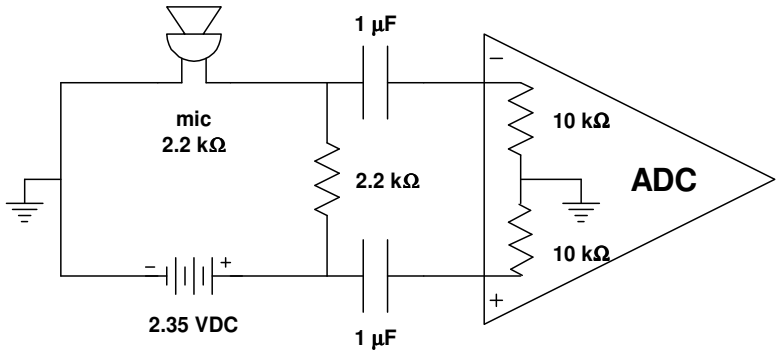
1.5.4 Choosing, Wiring, and Housing the HF Microphone

1.5.4.1 General

- The AIC2x provides a 2.35 V DC microphone bias. Choosing a compatible handsfree microphone reduces complexity, simplifies the design, and lowers board space and costs relative to a design based on a different DC bias voltage. Texas Instruments is accustomed to working with microphones with a sensitivity in the range of –40 to –45 dB at 1 kHz (where 0 dB is equal to 1 V/Pa). If the microphone and speaker sensitivity are too low, the AIC2x handset and headset maximum sidetone may be inadequate.
- The simplest (but not the best) handsfree microphone hookup connects the following in series:
  - Ground
  - 2.35 V source
  - 2.2 kohm (matching) external resistor
  - Microphone with 2.2 kohm internal impedance
  - Ground

The AIC2x handsfree microphone differential input measures the voltage across the 2.2 kohm resistor. Capacitors that block DC and act as a short at audio frequencies are placed on each side of the AIC2x differential input. This hookup allows the handsfree microphone chassis to be grounded. However, exploiting the AIC2x differential ADC capability with a fully differential microphone hookup has advantages, which are covered in "Microphone Hookup Circuit" on page 1-33.

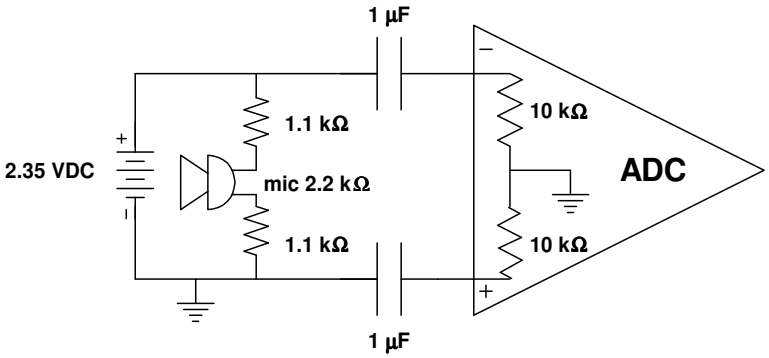
Figure 1-11 Handsfree Microphone Circuit—6 dB/Octave Filter



- For handsfree, it is even more important that the microphone connecting wires be twisted and, preferably, shielded. If one wire is grounded, the ground can be connected to the outer shielding conductor. If the external microphone chassis (the metallic external case of the microphone) is electronically connected to one input wire, it can be advantageous to connect the input wire to the ground so the chassis cannot short with any nearby housing. If an untwisted, unshielded pair of wires is used to connect the handsfree microphone, the Texas Instruments RDB and SDB IP Phones can transmit excessive pickup noise when these wires are placed near some components on the internal circuit board of the phone.
- To further reduce noise, collectively shield the entire length of the wires (chassis and hot) that connect the microphone to the circuit board.
- Design the microphone wire traces on the circuit board as short as possible and wide enough to not add impedance.
- Make sure that microphone traces on the circuit board are between ground planes and as far as possible from the 8-ohm driver traces and other noise sources.
- If an unshielded, untwisted microphone hookup is preferred, the evaluation platform handsfree microphone hookup (for example) can be reconfigured to attempt common mode rejection. This circuit would connect, the following in series:
  - Ground
  - 2.5 V source
  - 1.1 kohm external resistor
  - Microphone with 2.2 kohm internal impedance
  - 1.1 kohm external resistor
  - Ground

Now, the voltage across the microphone would be measured and the same capacitors would be used to block DC on each side of the AIC2x differential input.

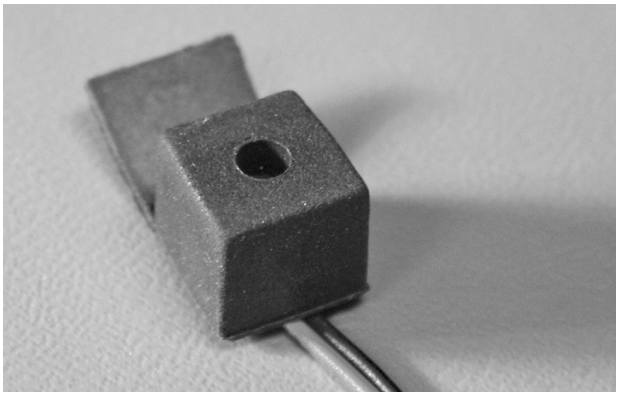
Figure 1-12 Microphone Circuit with Common Mode Rejection



- A directional or hypercardioid microphone directs the near-end mechanical speaker at a right angle to directional microphones. If the near-end participants' mouths are directed toward the microphones they are preferentially detected. The drawback with a directional microphone comes if the handsfree phone is placed on a conference table with participants surrounding it who are not all facing the directional microphone. Summing an array of horizontally aligned directional microphones is a possibility. Another drawback is that the directional microphone frequency dependent response is generally less flat.
- Obtain data for the frequency response of the handsfree microphone. Also, experiment to obtain the standardized handsfree microphone response inside the proposed handsfree microphone encasing. Confirm this is satisfactory before finalizing the choice of handsfree microphone and the enclosure housing it. Avoid a situation where the wrong choice of microphone causes the SLR frequency response to violate a relevant specification, such as TIA-810. Changing the design of the plastic enclosing the handsfree microphone can alter the send spectral response. Consult an acoustical expert if shaping this response is needed.
- The microphone should have low sensitivity to acceleration of the microphone caused by non-airborne structural vibrations. Low sensitivity to wind is also desirable.
- The microphone connection to the circuit board should be shielded. Using 30-gauge wire coiled at the end greatly damps the transmission of vibrations to the handsfree microphone.
- To reduce vibrations, configure the microphone diaphragm to face at right angles to the speaker-induced sound and vibrations. Usually this means placing the microphone diaphragm and speaker cone at a right angle.
- Try to attenuate non-airborne vibration coupling between the near-end speaker and near-end microphone. Often microphones are encased in foam or rubber grommets to absorb vibrations (Figure 1-13 on page 1-32). The microphones and speakers can be better isolated from table vibrations by means of adhesive rubber

foam padding. It is helpful to mount the speaker and microphone on different plastic pieces, joined together with some type of vibration damping interface. Avoiding vibrations and maintaining a precisely fixed speaker-to-microphone distance and geometry are important for good AEC convergence.

Figure 1-13 Foam-Encased Microphone Isolates from Non-Airborne Vibrations



- The foam placed around microphones also acts as a windscreen, reducing the sensitivity of the microphone to wind velocity.
- Design the phone to optimize distance between the speaker and microphone.
- Consider placing the handsfree microphone in a separate unit that rests on the table and has a long cable to reach the active near-end participant.
- When the speaker and microphone are inside the same enclosure, there should be internal obstructions to block airborne sound from propagating from behind the speaker to the microphone. Even a small air leak will transmit or allow sound to pass through. It is useful to encase the space behind the speaker with a back volume chamber (Figure 1-10 on page 1-28) and to encase the microphone. There should be acoustical barriers between the speaker and microphone inside the phone. As the speaker vibrates the larger plastic piece it adheres to, and this displaces air volume inside the phone, the microphone should be isolated from the effects of this changing air volume. If the speaker cone rear-radiation output and the microphone are connected by a direct acoustical path inside the phone, the inner acoustical path can degrade the ERL much more than the external acoustical path from the front of the speaker to the microphone.
- If the handsfree microphone is placed near the plane that the speaker cone lies on, it may encourage destructive interference of low-frequency back volume leakage, with sound originating from the front of the speaker. It is useful to experiment by moving the handsfree microphone around its general desired location to find a local minimum in the enclosure echo at low frequencies.
- Some speakerphone designs direct the handsfree microphone toward the surface just in front of the phone. The intention is for near-end sound to bounce off the surface into the handsfree microphone. The drawback here is that such a phone transmits too weakly on a soft surface or cluttered desk. The benefit is that the handsfree microphone can be placed deeper into the acoustical shadow of the handsfree speaker.

If you imagine the handsfree speaker as a shining light, the handsfree microphone should be located in as dark a spot as possible. Under the belly of the phone it is darker. However, some phones apply this principle badly. Some phones have a relatively small volume of air between the enclosure's bottom flat surface and the parallel table surface. Sound can be efficiently driven into this small volume of partially trapped air by the enclosure's bottom surface vibrations. A handsfree microphone in a position under such a phone can result in excessive coupling with handsfree speaker driven vibrations of the bottom surface.

- Modifying the external plastic to form a parabolic chamber or horn around the handsfree microphone can increase the near-end speech acoustical intensity reaching the microphone. This enables lower gain on the ADCPGA, which reduces electronic amplifier noise, EM pick up, and handsfree speaker coupling (echo).

1.5.4.2 Microphone Hookup Circuit

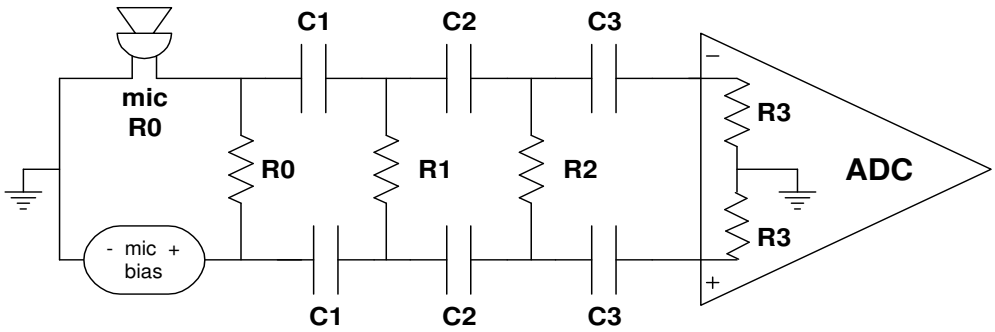
Texas Instruments recommends placing a passive analog high-pass filter between the microphone output and the input of the ADC, especially for wideband compatible IP Phones.

Analog high-pass filtering can have many advantages:

- At low frequencies, the typical spectrum of room and microphone noise increases inversely with frequency. This corresponds to an increase in low frequency noise of 6 dB per octave, so Texas Instruments recommends an external high-pass filter that attenuates at least 12 dB per octave and preferably 18 dB per octave.
- At 16 kHz sampling, the AIC2x band-pass filter, high-pass cutoff doubles in frequency. This cuts into the voice band so the high-pass filter must be turned off.

Figure 1-14 shows an example of microphone hookup circuitry with passive analog band pass filtering. This circuit is currently used on the TNETV105x software development board (SDB) and Texas Instruments IP Phones.

Figure 1-14 Microphone Hookup Circuitry with Passive Analog Band Pass Filtering



The internal impedance of the microphone is 2.2 kohms. The following table lists the values of the other circuit components.

Component	Value
R0[1]	2.2 kohm
R1, R2, R3[2]	10 kohm
C1	470 nF
C2, C3	270 nF

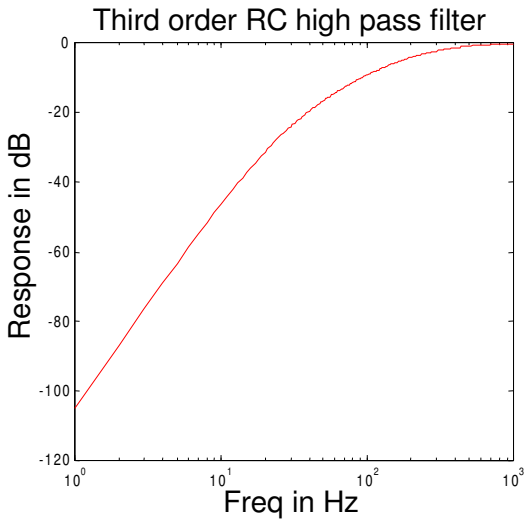
1. The internal impedance of the microphone is also 2.2 kohms; mic bias is supplied by the TNETV105x.
2. The R3 resistance is internal to the TNETV105x hardware codec ADC.

For these values, the pass band is attenuated 1.57 dB relative to a microphone hookup with all capacitors shorted and resistors removed except R0 and R3. Therefore, the high pass filter adds 1.57 dB attenuation plus the transfer function T in Figure 1-15 "Response of Recommended Microphone Hookup" on page 1-34. The total pass-band loss is 8.5 dB. This 8.5 dB resistive loss is normalized out of Figure 1-15, where the pass-band loss is shown as 0 dB.

This produces the following results:

- T(125 Hz) = –7.43 dB
- T(300 Hz) = –2.35 dB

Figure 1-15 Response of Recommended Microphone Hookup



1.5.4.3 Third-order RC Filter

The ADC of the TNETV105x and AIC2x hardware codec has no high pass filter in the microphone transmit path that is appropriate for 16-kHz sampling. It is important to avoid TNETV1050 releases prior to PG3.1 to avoid excessive ADC squiggle noise. Using the third order RC filter in versions prior to PG3.1 reduces squiggle noise by 6 dB, but does not eliminate audible handsfree microphone squiggle noise.

1.6 Design Guidelines for Phone Enclosures

Successful implementation of a handsfree speakerphone depends on a careful design. Observe the following guidelines to minimize potential for problems related to the physical design and fit of the phone enclosure:

- Attach four rubber pads to the bottom of the phone enclosure to reduce speaker-to-microphone vibration coupling and table-to-microphone coupling. This also reduces the possibility of unintended movement of the phone which helps AEC stay converged.
- Encase and house the microphone in foam. Make sure the foam pinches the wires leading to the microphone. This will increase the ERL by reducing vibration coupling.
- Encase the microphone and surrounding foam in the enclosure to shield the microphone from sound originating from the back of the speaker and the inner front panel of the phone.
- Line up the microphone with a hole in the phone enclosure to couple directly with external sound. Users normally face the phone, so the hole should be on the front of the phone.
- If consistent with objectives, analog or digital telephony high-pass filtering of both the microphone and signals is recommended.
- When testing handsfree AEC, connect to a far-end handset. Test with the handset cord disconnected, with the cord only, and with the cord with the handset against someone's face. Snap fingers at the handsfree microphone to see if echo returns. Possible problems:
  - The handsfree phone should not have sidetone. (near-end microphone to near-end speaker coupling.)
  - The near-end and far-end phone should be four wire and not have a local hybrid causing line echo.
  - The line and far-end phone should be free of line echoes. AEC cannot reduce far-end line echo.
- When evaluating handsfree AEC during a real phone call, place the interacting phones in different rooms far apart. AEC reduces handsfree near-end speaker to near-end microphone coupling, but not near-end handsfree speaker to far-end microphone coupling.

1.7 Testing a Speakerphone Enclosure

The acoustical performance of an IP Phone speakerphone is determined by the operation and interaction of the following elements:

- DSP acoustic echo cancellation (AEC) software
- Design of the phone enclosure including gaskets and back volume chamber
- Characteristics of the speaker and microphone
- Internal wiring

Experience has shown that the design of the hardware elements (the last three items in the list above) can have a significant effect on the performance of AEC.

To best evaluate the design of the enclosure, the hardware elements should be fabricated and available for testing at the same time that the printed wiring board (PWB) assembly is first tested and integrated with the Telogy software. To this end, Texas Instruments strongly suggests that the following hardware testability features be designed into your product prototypes; these testability features enable you to do total design evaluation and software parameter verification early in the development process so you can identify design changes to help you meet or improve product performance. Testing, verifying, and finalizing the enclosure design early in the process also enables you and Texas Instruments to tune the AEC parameters quickly and reliably, reduces the probability of costly rework, and potentially shortens overall product development time.

1.7.1 Test Description and Printed Wiring Board Preparation

The technique used for AEC tuning and testing is to interconnect a Texas Instruments IP Phone software development board (SDB) with the customer-developed enclosure prototype and evaluate the performance. To do this 'hybrid' testing, the printed wiring board (PWB) of the enclosure prototype must be fitted with connectors at two test points—the microphone input and the speaker output. These connectors allow the PWB to be bypassed so the SDB can

- Receive input signal from the microphone of the prototype enclosure
- Send output signal to the speaker in the prototype enclosure



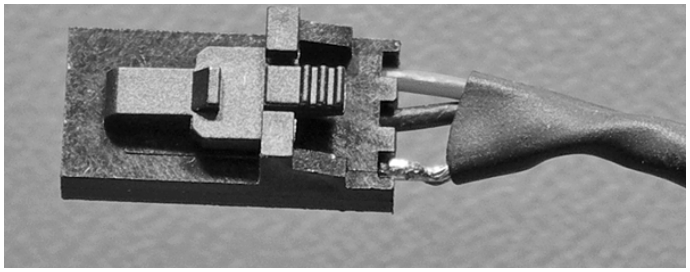
NOTE—The following modifications should be done to at least two—and preferably three—prototype units that can be made available for testing.

1.7.1.1 Board Modifications

To facilitate testing, a prototype board is modified as follows:

- The microphone is connected to the PWB through a shielded, twisted-pair cable that is soldered to the microphone terminals on one end and soldered to a single row, three-socket connector with 0.1-inch spacing (Molex 50-57-9403) on the other end (Figure 1-16). One pin is used to connect the shield to the ground of the PWB. Earth ground (if available) is preferable to logic ground. The ground is unconnected at the microphone end.

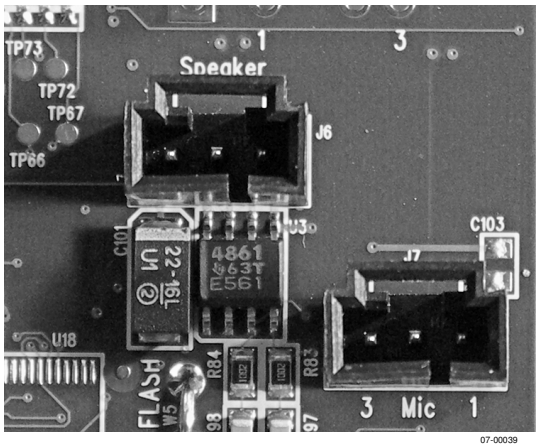
Figure 1-16 Three-socket Header for Microphone and Speaker Connections



07-00040

- The speaker of the speakerphone is connected to the PWB through a shielded, twisted-pair cable that is soldered to the speaker terminals on one end and soldered to a single row, three-socket connector with 0.1-inch spacing (Molex 50-57-9403) on the other end (Figure 1-16). One pin is used to connect the shield to the ground of the PWB. Earth ground (if available) is preferable to logic ground. The ground is unconnected at the speaker end.
- The PWB is fitted with two single row, three-pin cable receptors (Molex 70543-0002) with which the three-socket connectors mate to connect the microphone and speaker (Figure 1-17). These connections must be easily accessible when the phone enclosure is opened.

Figure 1-17 Speaker and Mic Receptors on the Printed Wiring Board (PWB)



07-00039

- The twisted pairs should be tightly twisted and colored red and black, with the red wire connected to the positive or driving signal, if applicable. A good example of cable is part number C1228.18.01 made by General Cable.
- The documentation must clearly state which color wire is connected to which lead for both the speaker and microphone.
- A means of access must be available to route two such twisted-pair cables (each 0.125-inches in diameter) into the closed assembly without the need to drill holes in the phone enclosure.
- Opening and closing the phone enclosure must be accomplished with ordinary tools.

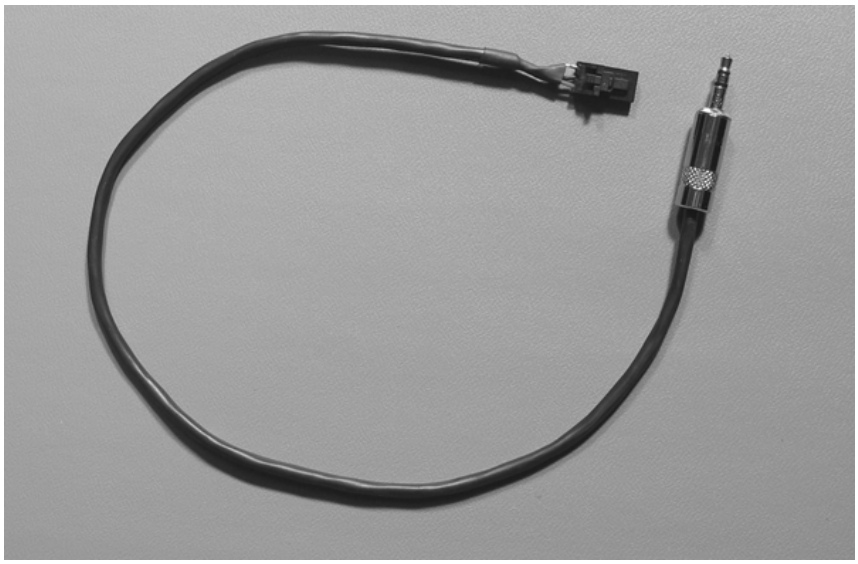
1.7.1.2 Audio Interface Test Cables

Two audio interface test cables are required to interconnect the prototype phone enclosure and the SDB.

The Texas Instruments/Telogy test facility has created audio interface test cables (Figure 1-18) as follows:

- 18-inch length of General Cable C1228.18.01 shielded, twisted-pair cable
- A 3.5 mm audio plug (Neutrik NYS231) at one end
- A three-socket connector (Molex 50-57-9403) at the other end for the three wires (red, black, and white (shield)) that mate with 0.025-inch posts in the three-pin cable receptor on the PWB

Figure 1-18 Audio Interface Test Cable



07-00041

Label one audio interface cable MIC and the other audio interface cable SPKR.

1.7.2 Test Set Up

After making the modifications described above to the PWB, use the following procedure to interconnect the prototype phone enclosure and the IP Phone SDB.

Procedure 1-1 Interconnecting the Phone Enclosure and SDB for Testing

Step	Action
1	Open the IP Phone enclosure to expose the printed wiring board (PWB).
2	Disconnect the internal speaker cable from the Speaker cable receptor on the PWB (Figure 1-17).
3	Connect the audio interface cable labeled SPKR to the Speaker cable receptor.
4	Disconnect the internal microphone cable from the Mic cable receptor on the PWB (Figure 1-17).
5	Connect the audio interface cable labeled MIC to the Mic cable receptor.
6	Route the free ends of both audio interface cables out of the phone enclosure through convenient access points, then close the enclosure.
7	Plug the microphone (MIC) audio interface cable into the MIC port of the SDB. See Figure 1-19.
8	Plug the speaker (SPKR) audio interface cable into the correct speaker port on the SDB according to the hardware design of the phone enclosure: amplified signal (Speaker w/ Amp) or unamplified signal (Speaker). See Figure 1-19.

9 The phone enclosure and the SDB are now interconnected for testing.

End of Procedure 1-1

Figure 1-19 Microphone Port and Speaker Ports on the SDB



1.8 Diagnosing Problems External to the Near-End Phone

Use caution when trying to solve far-end phone quality problems at the near-end handsfree phone. For example, improved full-duplex performance can expose problems in distant line echo cancellers that may chop (intermittently attenuate) the signals when both paths are active. You cannot automatically assume the near-end handsfree phone is faulty if a spurious chopping problem is fixed by using more half-duplex handsfree operation. In fact, this result may be observed because a network line echo canceller behaves much better when it does not encounter speech signals in both directions at the same time (double talk). Therefore, overall system quality may initially be better in handsfree half-duplex mode relative to a more full-duplex phone, but later—after the intervening network is improved—the half-duplex phone may be evaluated as having inferior duplex quality. Such improvements are common when the line echo suppressor is upgraded or removed after a digital carrier obsoletes the analog hybrid circuitry.

Diagnosing such a problem can involve sniffing packets and using the IP Phone PCM trace feature to see where the chopping attenuation occurs. Many external causes for chopping may exist in addition to line echo cancellers, such as lost packets, an active noise guard, a vocoder saving bandwidth by limiting transmissions when its VAD (voice activity detector) detects noise or silence, noise filtering, automatic gain control, high level compression, etc.

Another important example of an overall system problem is a handsfree half-duplex phone connected to a far-end handset microphone with abnormally high sensitivity to nasal exhale wind noise from the far-end user. This results in choppiness of the near-end transmitted speech due to spurious far-end nasal exhale noise break-ins. It is better to fix the far-end handset instead of fixing the problem by increasing the near-end AEC receive path noise level parameter that governs the break-in power threshold in the half-duplex handsfree receive path. This near-end fix has an undesirable effect for a different far-end phone that transmits weakly. Thus, after fixing the far-end nasal wind noise problem at the near-end handsfree half-duplex phone, the side effect is that desirable low level far-end signals do not adequately break-in through the near-end handsfree receive path (i.e., avoid the attenuation AER Rx NLP applies to the subdominant path).

1.9 Preparing for Performance Assessment

The following sections list test equipment and test environment guidelines:

- "Test Equipment Checklist"
- "Testing Environment Checklist"

1.9.1 Test Equipment Checklist

The following list contains suggested facilities and resources for doing the "Performance Assessment Tests" on page 1-45. Not all the tests require all the equipment. See the Equipment List before each procedure.

- A "quiet room" in which to test the IP Phone.

  Companies such as Acoustic Systems sell and build acoustically customized inner rooms that fit in normal offices.
- IP Phone Test Set—Handset testing fixture and handsfree testing fixtures with ear coupler and artificial mouth.

  Companies such as Head Acoustics and Microtronix sell testing fixtures and acoustic measurement and analysis systems for IP telephones.
- A good quality external microphone and peripheral amplifier, power supply, and cables for recording handsfree speaker sound.

  A top-of-the-line microphone for telephony acoustical measurements would be a half-inch 4192 B&K microphone with calibration unit. A lower cost microphone will suffice for the testing given in "Performance Assessment Tests" on page 1-45.
- A good quality PC sound card for recording analog signals.
- Network Protocol Analyzer PC software (Ethereal) for decoding recorded packets and converting them into audio wave files.
- Audio wave file analysis PC software—such as *Adobe Audition*—for playing audio wave files to an external speaker, viewing the files, and performing spectral data analysis of data.
- (Suggested) Packet transmission PC software for transmitting audio wave files to an IP Phone with a specified IP address and UDP port.

Texas Instruments Technical Support can help customers set up an in-house system to send and record packets.

1.9.2 Testing Environment Checklist

The following guidelines describe how to maximize testing efforts by operating the handsfree phone in a typical work setting.

- Use the phone in a room with sound absorbing materials such as sound absorbing tiles, curtains, carpet, etc.

  Place the phone on a hard surface to reflect sound into the handsfree microphone. Instruct users to speak close to and direct their voice toward the handsfree microphone.
- Seat participants roughly the same distance from the phone.

For fewer participants, if possible, turn the phone to face them. Speech signal levels may be closer, resulting in better AEC performance. Soft-spoken people may need to move closer to the phone or move the phone closer to them.

- Keep stationary objects away from the phone.

  Objects can reflect additional handsfree speaker sound into the handsfree microphone and block near-end speech from reaching the handsfree microphone.

- Keep moving objects and noise sources as far from the phone as possible.

  Plan ahead (have extension cords, etc.) so that a noisy projector fan does not end up right beside the handsfree microphone. Keep the phone stationary. If convenient, users should remain relatively stationary and avoid placing hands or objects close to the phone. Avoid unnecessarily vibrating the table the phone is on.

1.10 Configuring the IP Phone for Performance Assessment Tests

When testing two or more IP-phones at the same time, make sure that none of the test phones are in the same room or a nearby room. For handsfree-to- handsfree testing at maximum speaker volumes, phones may need additional separation, depending on how easily airborne sound can travel between the two phones.

All tests are done with AER disabled. This allows for handsfree enclosure optimization without the need to adjust AER parameters with every change.

During loudness calibration tests and other acoustical tests of a phone that does not explicitly require AER/AER Noise Guard/AER EQ/AGC/VAD, disable these features. Otherwise, these DSP modules can give misleading results for tests that send a signal through the send path. The AER HLC should also be disabled to better characterize the Rx path. Due to the AER Rx NLP, AER can also create misleading results for tests that involve the receive path. For example, assume the send path noise and distortion is tested for relatively low near-end sound test signal. Under some conditions, AER may not switch off attenuation in the Tx NLP. AGC might increase tx_ag and lower aer_tx_dg. After AGC changes the gains from their nominal default levels, the send path noise and distortion would also change and yield a false indication of conditions that exist during normal operation when intermittent speech occurs in both paths. AER Noise Guard may dramatically reduce near-end noise during testing, while a standard's specification for maximal noise is violated just after near-end speech and just before the AER Noise Guard attenuation engages.

Texas Instruments IP Phone Acoustic Testing Service can do all the tests described plus additional TIA-810B tests, or describe how to do some of the tests at the customer location. For more information, see the document "Preparing for AEC Tuning" available from the Applications Engineering group in Germantown, MD.

When initially measuring the handsfree send and receive spectral response and overall loudness ratings the following steps should be taken.



NOTE—To generate a build that supports the <dsp 0 ...> commands, you must set the flag for `GG_INC_DIM_CHAN_IF` to 1 to enable DIMCHAN API support in IP Phone, then recompile the microcode.

Some AEC/AER debug commands used for calibration/debug purposes use the dimchan interface, such as:

- dsp <tcid> aerc
- dsp <tcid> aerp

For calibration purposes these commands may be required. After calibration is done, you can set the flag back to the default value (0) in **ggtune.h**.

Procedure 1-2 Setting Up for an IP Phone Performance Assessment

Step – Action

1. Set up a call between two IP Phones and place the near-end phone to be tested in handsfree mode.

   Make sure that the two IP Phones and PC Ethernet cable are connected into an otherwise isolated private HUB.

2. Disable the echo canceller, using the following command:

   ```
   dsp 0 aerc off
   ```

3. Disable noise guard using the following command:

   ```
   dsp 0 aer_nguard_ctrl off na na na
   ```

4. Disable the adaptive ADC PGA function of AGC using the following command:

   ```
   dsp 0 agc on 0 na na na
   ```

5. Before using an external source to send in test packets, disconnect the Ethernet connection of the far-end IP Phone that was used to set up the call.

   Because many IP Phones time out after a long delay, a convenient alternative is to put the far-end IP Phone in loop mode, using

   ```
   dsp 0 loop snd on
   ```

   This eliminates transmissions from the far-end IP Phone used to set up the call.

End of Procedure 1-2

1.11 Performance Assessment Tests

Before doing the performance assessment tests, read or review the following:

- "Test Equipment Checklist" on page 1-41
- "Design Guidelines for Phone Enclosures" on page 1-35
- "Configuring the IP Phone for Performance Assessment Tests" on page 1-43

After you set up and configure the IP Phone, do these tests in order:

- Procedure 1-3 "Determining the Sum of Send Gains" on page 1-45
- Procedure 1-4 "Testing the Speaker Volume and Determining Receive Path Gains" on page 1-46
- Procedure 1-5 "Testing the ERL" on page 1-47
- Procedure 1-6 "Measuring Harmonic Distortion and Reducing Echo Nonlinearity" on page 1-47
- Procedure 1-7 "Comparing Harmonic Distortion with the Texas Instruments EVAL Unit" on page 1-48
- Procedure 1-8 "Measuring the Noise Level of the Handsfree Microphone" on page 1-49
- Procedure 1-9 "Testing for Low-Level Nasal Exhale Noise" on page 1-49

1.11.1 Determining the Sum of Send Path Gains

To do Procedure 1-3, you need the following equipment:

Equipment	Comments
Quiet Room	
Handsfree Testing Fixture	
IP Phone Test Set	

Procedure 1-3 Determining the Sum of Send Gains

Step – Action

1. Obtain the handsfree send spectral response and overall SLR loudness from a qualified source.
2. Check if the response is within a specified mask of interest.

   If the response is not within the mask, you can apply equalization filtering. See "Equalization" on page 2-26.
3. Determine the sum of send path gains (aer_tx_dg + tx_ag) that will be used for subsequent computations.

   This will be the sum of send path gains that yield a nominal SLR value of 13 ±4 dB for the phone.
4. Repeat the measurements in step 1 to verify that you achieve a nominal SLR value in the range 13 ±4 dB for handsfree.

   If there are problems:

   - Make sure that the microphone wires are connected with the correct polarity.
   - Make sure that the microphone was not damaged by heat during soldering.

- Make sure that the DC bias voltage for the microphone is the correct value and that the bias voltage is noise-free.

End of Procedure 1-3

1.11.2 Testing the Speaker Volume and Determining Receive Path Gains

To do Procedure 1-4, you need the following equipment:

Equipment	Comments
Quiet Room	
Handsfree Testing Fixture	
IP Phone Test Set	

Procedure 1-4 Testing the Speaker Volume and Determining Receive Path Gains

Step – Action

1. Obtain the receive a spectral response and overall RLR from a qualified source. If facilities permit, make the measurement in-house.
2. Check if the spectral response is within a specified mask of interest.

   If the response is not within the mask, you can apply equalization filtering. See "Equalization" on page 2-26.
3. Check if the speaker is sufficiently loud to meet TIA-810B standards.

   If the speaker is not loud enough, you must adjust the receive-path gain. See "Gain Calibration Procedure" on page 2-23.
4. Determine the precise receive path nominal default gains (aer_rx_dg + rx_ag) that will be used for subsequent handsfree mode testing. Initially this will probably be the gains that result in an RLR value of 16 ±4 dB for the test phone.

   For TIA-810B compliance, the targeted RLR is reached with a maximum gain of –1.5 dB or less on the AIC2x DAC PGA and a digital gain of 6 dB or less on the AER receive path.

   The digital gain of the AER receive path can be used to get RLR=16 dB to the nearest 0.5 dB. Otherwise, set digital gains to zero.
5. Repeat the test with appropriate gains and verify that you achieve a nominal RLR value in the range 16 ±4 dB.

   If the speakers are not loud enough when DAC PGA = –1.5 dB and aer_rx_dg = 6 dB:

   - Try a larger, more sensitive speaker
   - Seal the spaced behind the speaker with a back volume chamber to stop excessive destructive interference from speaker back-volume pressure
   - Add some acoustic exhaust to the back volume chamber to decrease the "spring stiffness" of trapped air behind the handsfree speaker
   - Make sure that the front panel of the phone is attached well to the speaker
   - Make sure that the speaker is not overly damped; over-damping keeps the front panel from vibrating enough with the speaker to extend the radiating area of phone.
   - Add an external audio amplifier

End of Procedure 1-4

1.11.3 Testing the ERL

To do Procedure 1-5, you need the following equipment:

Equipment	Comments
Ethereal Network Protocol Analyzer	
Adobe Audition	
Packet Transmission Software	

Procedure 1-5 Testing the ERL

Step – Action

1. Generate a file of a linear tone sweep from 100 to 3400 Hz in 100 Hz steps in PCM μ-law format.
2. Set up a call between two phones in PCM μ-law format.
3. Stop the far-end phone from transmitting packets by unplugging the network cable from the far-end phone.
4. Disable AER for the near-end test phone.

   Note—AER continues to implement aer_rx_dg, aer_tx_dg, HLC, and all equalizers even when disabled.
5. Inject the tone-sweep file and transmit toward the test phone using packet transmission software.
6. Sniff the packets coming out of the phone under study with the network protocol analyzer (Ethereal).
7. Extract the payload from the packets and use Adobe Audition to measure the level of each tone.

   If there is saturation, decrease the handsfree microphone ADC PGA until saturation is just avoided, then set the gain of the AER digital send path to 0 dB for simplicity. Resume testing with step 5 above.
8. Record the difference in level between each tone in the original file and the echo file.
9. Average the differences of the tones from 300 to 3400 Hz to get an unweighted ERL.
10. After data collection, linearly interpolate the ERL result to the amount expected if gains were adjusted to give the nominal RLR and SLR.

End of Procedure 1-5

1.11.4 Measuring Harmonic Distortion and Reducing Echo Nonlinearity

To do Procedure 1-6, you need the following equipment:

Equipment	Comments
Ethereal Network Protocol Analyzer	
Adobe Audition	
(Optional) Packet Transmission Software	
External microphone to record signal from the speaker of the phone enclosure	

Procedure 1-6 Measuring Harmonic Distortion and Reducing Echo Nonlinearity

Step – Action

1. From the PC, transmit test packets that represent 0dBm sine waves for the following one-third-octave center frequencies:

- 100 Hz
- 125 Hz
- 150 Hz
- 200 Hz
- 315 Hz
- 400 Hz
- 630 Hz
- 800 Hz
- 1000 Hz

2 Record with maximum ADC PGA gain, that does not allow saturation.

3 Using PC spectral analysis of the final audio wave data results, record how many dB attenuation the worst-case harmonic distortion has relative to the fundamental.

4 Verify sufficiently low harmonic distortion. A good quality handsfree phone will have 20 dB or more attenuation of all higher harmonics for a fundamental of 200 Hz or higher.

5 For the same gains, repeat this test while recording analog audio signal from an external microphone placed just over (but not touching) the handsfree speaker. The close placement yields a better SNR.

6 For the same gains, repeat the test recording of packets after replacing the handsfree microphone with an equivalent impedance magnitude resistor. Harmonic coupling should be attenuated by at least 20 dB.

If the distortion is excessive, take described measures to reduce it, such as compression, analog high-pass filtering of DAC output, enclosure improvements, etc. Reduce the test signal amplitude and DAC PGA gain accordingly if the decision is made to use compression and a lower maximum DAC PGA value. Use as large a receive path signal as possible without inducing flat-top saturation, so harmonic distortion reflects only the enclosure limitations, not the software flat-top saturation.

End of Procedure 1-6

1.11.5 Measuring Fundamental Harmonic Distortion

To do Procedure 1-7, you need the following equipment:

Equipment
Ethereal Network Protocol Analyzer
Adobe Audition
(Optional) Packet Transmission Software
External microphone to record signal from the speaker of the phone enclosure

Procedure 1-7 Comparing Harmonic Distortion with the Texas Instruments EVAL Unit

Step – Action

1 Measure fundamental nonlinear distortion.

2 Repeat Procedure 1-6 after reducing the AER receive path digital gain by –3 dB and –6 dB.

3 Get the echo drop in dB at all frequencies, and check how close the reduction in response is to –3 dB and –6 dB.

As the fundamental drops in power, the deviation from linearity in this test poses less of a problem.

4 Repeat this test with a different IP Phone and after incremental upgrades of phone under development.

End of Procedure 1-7

1.11.6 Measuring the Noise Level of the Handsfree Microphone

To do Procedure 1-8, you need the following equipment:

Equipment
Ethereal Network Protocol Analyzer
Adobe Audition

Procedure 1-8 Measuring the Noise Level of the Handsfree Microphone

Step – Action

1 With correct nominal default gains (RLR=16 dB, SLR=13 dB) measure the handsfree microphone noise level when the handsfree phone is in a quiet room.

2 Put the far-end phone in loopback mode:

```
dsp 0 loop snd on
```

3 Compare with Texas Instruments RDB handsfree send path noise floor, which is equal to or less than –62 dBm0.

Do not use low send path gains and attempt to linearly extrapolate to a SLR=13 dB. DAC echo noise and ADC amplifier noise scales nonlinearly, and mu-law G.711 packets are all zeros below –72 dBm0.

End of Procedure 1-8

1.11.7 Testing the Handset for Low-Level Nasal Exhale Noise

To do Procedure 1-9, you need the following equipment:

Equipment
Ethereal Network Protocol Analyzer
Adobe Audition

Procedure 1-9 Testing for Low-Level Nasal Exhale Noise

Step – Action

1 Get the gains calibrated in the send path for handset operation.

2 Calibrate gains yielding a handset SLR=8 dB, and use gains.

3 Have people normally exhale through nose while holding a Texas Instruments handset and test handset.

4 Record transmitted packets to compare average nasal exhale noise.

5 Ask subjects to breath similarly on the two phones as if during a call, and take care to hold the handset microphone the same distance and orientation from mouth. Preferably, subjects do not know which handset is Telogy's.

End of Procedure 1-9

1.12 Equipment for Measuring Acoustical Response

- "Required Equipment"
- "Equipment Substitutions"

1.12.1 Required Equipment

The following conditions impact decisions on the equipment that will be used to perform the measurements described in the standards listed in "Spectral Response Specifications" on page 1-12:

A conventional phone test head (IEC-318 coupler) is currently used to position a handset microphone at a realistic and consistent position with respect to a mouth simulator. The test head determines the handset orientation and distance from the mouth simulator. Thus, the distinctive spatial variations of the mouth simulator's near field sound and the test head geometry have a big effect on all handsets' send characteristics. A shorter handset's send response drops off because the handset's speaker is aligned with the test head ear, and the handset microphone lands further away from the mouth simulator. Unfortunately, buying a test head (including mouth simulator and artificial ear) does not ensure a permanent measurement solution. The current TIA-920 and proposed TIA-810B standards require HATS (Head and Torso Simulator) for handset testing.



NOTE—Telephony test heads and head and torso test sets are available from Bruel and Kjaer (B&K). Head Acoustics and Microtronix sell competitive IP Telephone test systems. (For more information, see "Referenced Suppliers and Products" on page 1-53.)

It may be helpful to buy handsets from a manufacturer that provides specification details such as the one-third-octave band spectral response of the send and receive paths. However, without handset calibration equipment, quality assurance cannot be checked independently of the external manufacturer providing the handset.

Texas Instruments has software and equipment from B&K (various calibrated microphones), Head Acoustics (head and torso simulator), and Microtronix (conventional test head and head). The Texas Instruments IP Phone Acoustic Testing Service has experience in measuring telephony spectral responses, overall loudness, and most TIA-810B tests. For more information, see the document "Preparing for AEC Tuning" available from the Applications Engineering group in Germantown, MD.

Third-party companies can also test telephony responses and provide certification of compliance with various standards. When arranging for preliminary externally-implemented telephony testing work, it is advisable to have a person on site who is well-trained for resetting all the telephone gain parameters. For example, suppose during TIA-810B testing that the acoustical to digital *loss* in the handsfree send path, given by the handsfree overall SLR rating, is found to be 17.5 dB, failing a 13+/–4 dB spec. You should immediately stop subsequent tests, increase the AER Tx digital gain (aer_tx_dg) by 1.5 dB, burn the new gain setting into flash memory, and begin the handsfree tests again, starting with the verification of a 16 dB SLR. Without such intervention, more time and money will be lost iterating the full suite of tests until full compatibility with a standard is attained. Changing any gain setting for a particular

phone mode (handset, headset, handsfree) will invalidate all standardized tests of that phone mode using different gains. A detailed description of all the gains (including the AER Tx digital gain mentioned above) is given later in this document. Assuming the phone manufacturer does not want to calibrate every phone, eventually the phones may need to pass the overall loudness ratings with the tolerance given by a standard. Still, it is useful to use intervention to more quickly achieve the milestone of having a "Golden Reference" phone that passes all desired tests.

If a commercial system for measuring acoustical response is not available, the following equipment or suitable substitutes (see below) is the minimum required to measure the response of a handsfree phone:

- B&K model 4227 Mouth Simulator
- B&K 2804 Power Supply, 2669 Amplifier, 4192 ½-inch Microphone
- B&K 4231 Microphone Calibration Device
- BNC cable, B&K JP-0144 Coaxial Adapter
- A system for audio one-third-octave band spectral analysis functionally equivalent to the Hewlett Packard HP35665A
- A system for generating and recording test signals. Preferably digital test signals can encode test signals into packets (off-line) that are externally transmitted to the phone, while the phone's transmitted packets can be recorded and decoded. Analog signals can be sent to the Mouth Simulator and recorded from the output of the speaker on the phone.

1.12.2 Equipment Substitutions

The following equipment can be substituted for the equipment listed in "Required Equipment" on page 1-50:

- A small speaker can be used for a commercially calibrated mouth simulator to obtain rough handsfree measurements. The replacement mouth simulator must be calibrated, using the calibrated microphone. Unfortunately, this can involve a lot of work to ultimately get a crude result. The small speaker is put in an anechoic room, or as quiet a room as possible. The speaker and microphone face each other and are held in position by as little material as possible, to make a "free space" measurement. The speaker's response is measured at a distance of 50 cm away in each one-third-octave band. (When the same mouth simulator later faces a handsfree microphone 50 cm away on a hard table, the hard table will reflect sound and increase the sound stimulus level at the handsfree microphone.) The response at 2.5 cm from the speaker does not need to be measured (for handsfree purposes) as one simply assumes that it is 24 dB higher than the free space result. Thus the free space result at 50 cm in dBPa plus 24 dB provides the sound stimulus input used to calculate the handsfree SLR, as described in ITU-T P.340 4.5.1.1. It is also possible to calculate the handsfree SLR by actually using the sound pressure level 2.5 cm from the mouth simulator, but this complicates the computation. For this approach, the deviation from a 24 dB drop in free space from 2.5 cm to 50 cm will also have to be measured for each one-third octave and properly taken into account. Standards define the handsfree SLR in terms of the level 2.5 cm away from the mouth simulator so this reference point, and thus overall SLR loudness level, is more consistent with that of the handset.

- Parts with equivalent performance are acceptable for the external microphone and microphone calibration device.
- Originally, Texas Instruments used both analog (far-end IP Phone with THAT-2 box handset I/O) and digital (rtpsend and capture/sniff software) to do TIA-810B and TIA-920 testing needed for phone calibration. Texas Instruments concluded that unless there was a significant effort to streamline and cross-check in-house developed procedures, the use of commercial automated measuring systems was advantageous. After purchasing automated systems, in house developed testing procedures are no longer used at Texas Instruments for the RLR or SLR measurements.

Rudimentary loudness ratings could be obtained using the numerical output of a relatively inexpensive sound level meter. However, it is preferable for the microphone output of a sound level meter to be input to a spectrum analyzer, so that DC offsets, EM pickup, and room noise outside of the one-third-octave band of interest can be filtered.

1.13 Referenced Suppliers and Products

The suppliers and products listed here are referenced in this document.

1.13.1 Acoustic Systems

www.acousticsystems.com
415 East Saint Elmo Road
Austin, TX 78745-1221
800-749-1460, 512-444-1961

1.13.2 Adobe Systems, Inc.

Adobe Audition (formerly *Cool Edit Pro* by Syntrillium Software)

www.adobe.com
345 Park Avenue
San Jose, California 95110-2704
408-536-6000

1.13.3 Bruel & Kjaer North America Inc. HQ

www.bksv.com
2815-A Colonnades Court
Norcross, GA 30071-1588
770-209-6907

1.13.4 Head Acoustics, Inc.

www.headacoustics.com
6964 Kensington Road
Brighton MI 48116
248-486-0099

1.13.5 Microtronix Systems Ltd.

www.microtronix.ca
120 Bessemer Road
London, Ontario, Canada, N6E 1R2
519-649-4900

1.13.6 MWM Acoustics, LLC

www.mwmacoustics.com
6602 East 75th Street Site 520
Indianapolis, IN 46250
317-849-8177

1.13.7 The Loudspeaker Design Cookbook

by Vance Dickason
Sixth Edition, 2000
ISBN-1-882580-33-8

Old Colony Sound Lab
P.O. Box 876
Peterborough, NH 03458

Chapter 2

Configuring and Optimizing Parameters for Acoustic Echo Removal

This document contains the following sections:

2.1 "Introduction to AER Components and Parameters" on page 2-2
2.2 "Parameter Optimization" on page 2-21
2.3 "Send and Receive Path Gain Adjustments" on page 2-22
2.4 "Optimizing Parameters for Handsfree Operation" on page 2-30
2.5 "Optimizing Handset and Headset Parameters" on page 2-40
2.6 "NMM Commands Relevant to AER Performance" on page 2-41
2.7 "Default Parameter Settings" on page 2-63

2.1 Introduction to AER Components and Parameters

This section contains the following topics:

- "Purpose" on page 2-2
- "Scope" on page 2-2
- "AER Components" on page 2-3
- "Parameters" on page 2-4
- "Control Commands" on page 2-20

2.1.1 Purpose

This chapter describes how to optimize the performance of the Acoustic Echo Removal (AER) feature of the Texas Instruments IP Phone. The procedures give guidelines for attenuating echo without attenuating desirable speech signals. Optimal parameter values for AER depend on the hardware enclosure and mode of operation of the IP Phone (handsfree, handset, or headset).

To speed product development, Texas Instruments IP Phone software requires little speaker-volume dependence in parameters optimized for AER performance. However, in some cases fine tuning parameters other than gains to make them speaker-gain dependent will improve the results. After determining optimal parameters through testing, the phone is configured to apply the correct optimized parameters when the phone mode or speaker volume changes.



NOTE—Parameters that are optimal for the Texas Instruments evaluation platform (EVAL) IP-phone may not be optimal for other phones.

2.1.2 Scope

Acoustical qualities of the hardware enclosure place strict limitations on the quality of AER performance and determine how to configure the AER. Material in this document is complementary with Chapter 1 "Guidelines for Designing and Testing IP Phones" on page 1-1. The AER can be configured with the duplex stabilizer on or off. Enclosures with better acoustics are typically configured with the duplex stabilizer off, which will enhance full-duplex performance. However, enclosures with relatively bad acoustics can yield excessive echo with the duplex stabilizer off.

All parameter optimization procedures in this chapter are done with the duplex stabilizer off. For phones with hardware enclosure problems such as low ERL and higher nonlinear echo, AER is best configured with the duplex stabilizer on. This can result in half-duplex performance. For half-duplex phones, ITU-T P.340 requires a noise guard, which can be enabled in the AER Tx NLP.

The following sections contain procedures for optimizing AER internal parameters and give guidance for setting external parameters relevant to AER performance. These external parameters include the AIC (Analog Interface Codec) send and receive path analog gains. There are also procedures for configuring the AGC (Automatic Gain Control) DSP module to optimally complement AER performance.

This chapter also contains syntax for specific NMM commands that set configurable parameters that influence AER performance. For more information about using NMM commands, see the *Application Services Command Reference Manual*.

CAUTION—Before you optimize AER performance, you must calibrate the phones correctly to adjust the overall gains in the send (microphone) and receive (speaker) paths. The optimal AER parameters can vary according to these gains, so it can be counter-productive to adjust AER parameters before setting the gains.

This chapter does not describe the calibration process of measuring loudness ratings; instead, it lists the equipment and documents you need to configure the IP phone for acoustical testing. This document facilitates optimizing AER performance by explaining how to distribute a given overall gain at different points along the send and receive paths.

2.1.3 Microcode Modifications

The IP Phone software for "production" builds might not include support for some NMM commands of the form <dsp 0 ...>. These commands are useful during IP Phone development and this document references these commands frequently.

To generate a build that supports all the <dsp 0 ...> commands, you must set the flag for `GG_INC_DIM_CHAN_IF` to 1 to enable DIMCHAN API support in IP Phone, then recompile the microcode.

2.1.4 AER Components

Figure 2-1 shows the performance-related components of the AER architecture.

Figure 2-1 AER Performance-Related Components

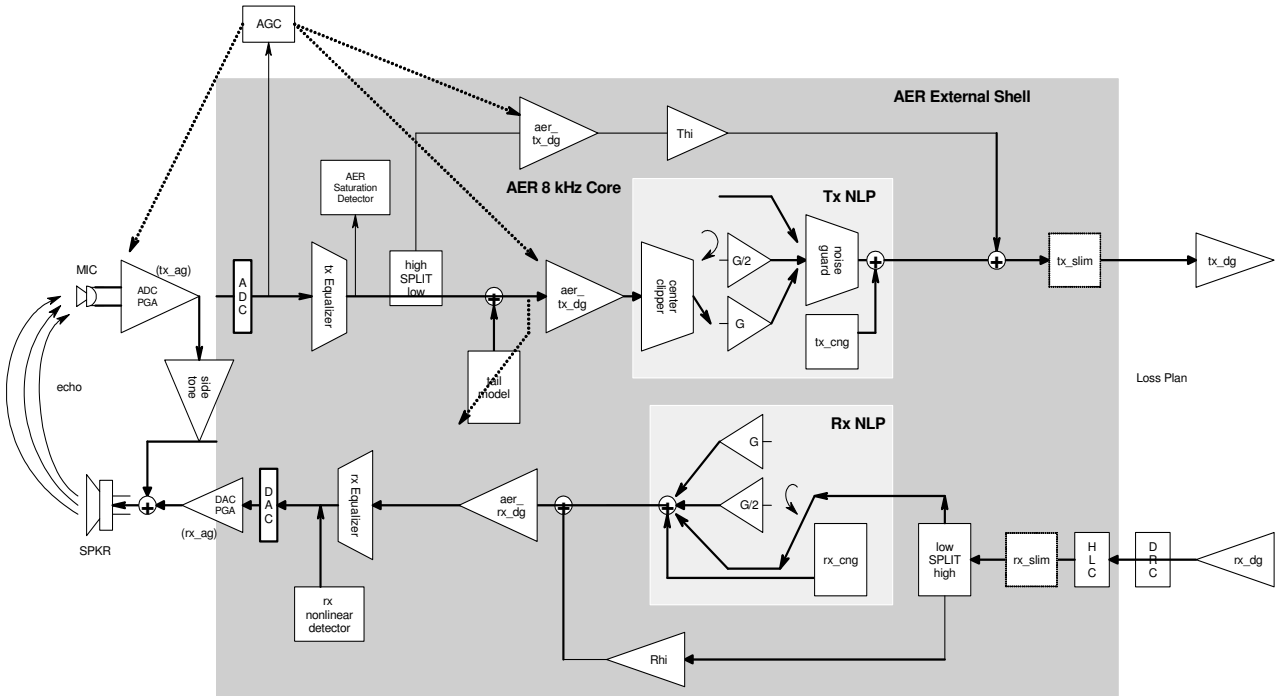


Table 2-1 Definitions of AER System Components

Component	Description
ADC	Analog-to-Digital Converter
aer_rx_dg	AER receive path internal digital gain, may vary with volume settings of the speaker
aer_tx_dg	AER transmit path internal digital gain, may be affected by AGC (automatic gain control)
DAC	Digital-to-Analog Converter
DRC	Dynamic Range Compression
G	AER NLP linear attenuation during single talk (applied to the subdominant path)
G/2	AER NLP linear attenuation during double talk
HLC	High Level Compensation
Rhi	Receive path hi band gain
rx_ag	DAC PGA receive path programmable analog gain for the DAC
rx_cng	Receive path comfort noise generator
rx_dg	Receive path digital gain for loss plan
rx_slim	Receive path signal Limiter (narrowband only)
Thi	Transmit path hi band gain
tx_ag	ADC PGA transmit path programmable analog gain for the ADC
tx_cng	Transmit path comfort noise generator
tx_dg	Transmit path digital gain for loss plan
tx_slim	Transmit path signal Limiter (narrowband only)
End of Table 2-1	

2.1.5 Parameters

This section lists parameters that affect the operation of IP Phone.

2.1.5.1 Configuration and Control

The following functions need configuration and control:

- Gains
  - AIC Analog gains for the ADC PGA and DAC PGA (programmable gain amplifier)
  - AER digital gains
- Operational Mode
  - Sampling rate (8 or 16 kHz)
  - Phone mode (handsfree, handset, headset, etc.)
  - Duplex stabilizer
- AER internal parameters
  - NLP parameters
  - CNG parameters
  - Thresholds/parameters for various state machines
  - Configurable control flags
  - Mute indication
  - ADC Tx input saturation indication (through AGC)
- Equalization
- High Level Compensation (AER HLC)

- Dynamic Range Compression (DRC)

2.1.5.2 Gains and AER-AGC Interactions

Good AER performance depends heavily on the correct gain information. Appropriate default gain settings can be obtained from the TIA 810B/920 (or other comparable standards) loudness requirements.

- SLR: determined by tx_ag and aer_tx_dg
- RLR: determined by rx_ag and aer_rx_dg

The AGC relays analog gain change information to the AER.

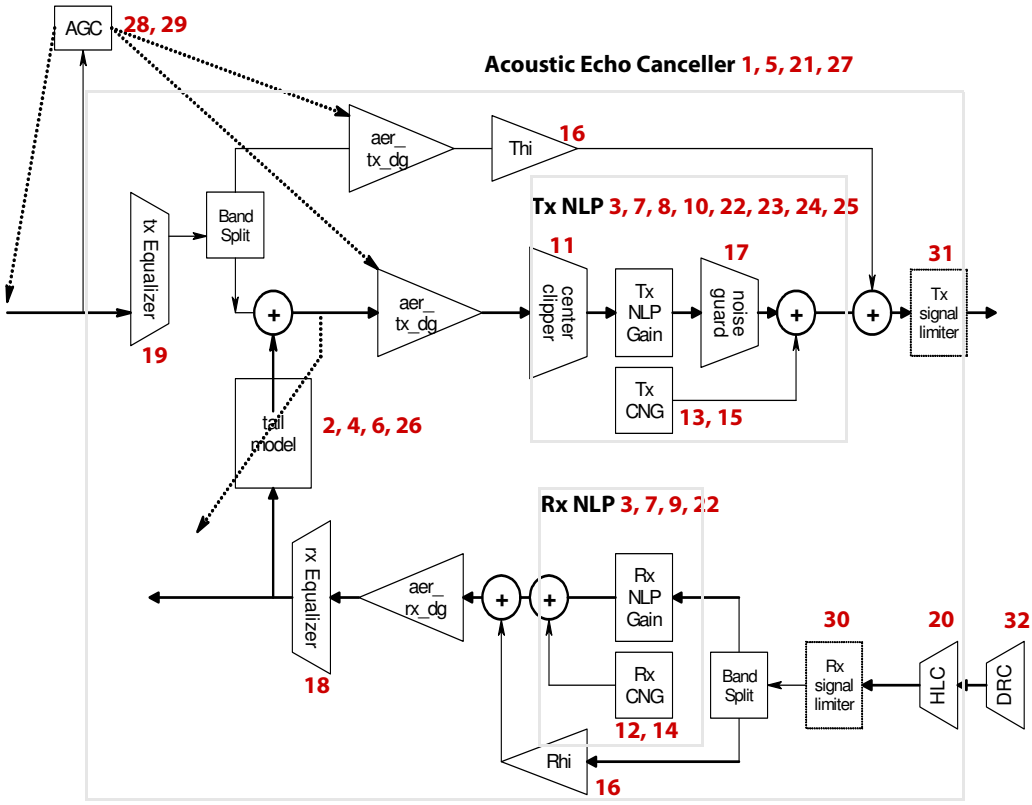
Rescaling of the adaptive tail model is required during analog (tx_ag and rx_ag) gain changes.

The AGC can control the tx_ag and aer_tx_dg adaptively, but does not impact the receive path gains. Saturation nonlinearity is affected by gains; the AGC signals the AER if saturation occurs in the ADC output. See "Gain Calibration Procedure" on page 2-23 for more information.

2.1.5.3 AER and AGC Configurable Control Flags

Figure 2-2 shows where configuration actions take place.

Figure 2-2 AER and AGC Configurable Control Flags



AER

1. AER enable/disable
2. AER Tail Model Filter and state machine updates enable/disable

3. NLP enable/disable
4. AER Tail Model Filter and state machine updates during mute enable/disable
5. Reset AER tail model
6. Reset AER all
7. Duplex stabilizer on/off
8. Tx NLP relaxation during mutual silence enable/disable
9. Rx NLP enable/disable
10. Tx NLP enable/disable
11. Tx center clipper enable/disable
12. Rx CNG enable/disable
13. Tx CNG enable/disable
14. Rx forced CNG enable/disable
15. Tx forced CNG enable/disable
16. Reduce high band attenuation of subdominant path

AER Noise Guard

17. Noise Guard enable/disable

AER Equalizer

18. Tx Path Equalizer enable/disable
19. Rx Path Equalizer enable/disable

AER HLC

20. HLC enable/disable
21. Phone mute on/off
22. NLP gain ramp scaling enable/disable
23. FDNLP/TDNLP selected
24. ASNR enable/disable
25. Fixed/adaptive CNG
26. Tail model filter freeze/update
27. Partial reset

AGC

28. AGC enable/disable
29. AGC adaptive mode enable/disable
30. AER Rx signal limiter
31. AER Tx signal limiter
32. Dynamic Range Compression (DRC)
    See "Dynamic Range Compression (DRC)" on page 4-1

For more information, see "AER Control Using dsp aerc" on page 2-44 and Table 2-15.

2.1.5.4 AER and AGC Configurable Parameters

The table below lists the parameters that are described in the following sections.

Table 2-2 AER and AGC Configurable Parameters (Part 1 of 3)

Component	Configurable Parameters
AER	AER filter length or tail length
	NLP send path center clipper aggression
	NLP linear combined loss target
	AER send path digital gain

Table 2-2 AER and AGC Configurable Parameters (Part 2 of 3)

Component	Configurable Parameters
	AER receive path digital gain
	Minimum noise power for receive path (based on user estimate)
	Minimum noise power for send path (based on user estimate)
	Hangover to switch away from receive path
	Hangover to switch away from transmit path
	Linear gain switching slew rates (Rx: in and out; Tx: in and out)
	Gain splitting time constant
	Threshold for receive path non linearity detection
	Tx analog gain change synchronization delay
	Rx analog gain change synchronization delay
	Tx analog gain change settling period
AER Noise Guard	Noise Guard hangover period
	Noise Guard desired send noise level
	Noise Guard ramping in period
AER Equalizer	AER equalizer filter parameters
AER HLC	HLC target level
	HLC ramp up time constant
	HLC power calculation time constant
AER Signal Limiter	Rx signal limiter
	Tx signal limiter
AER CNG	CNG Rx Level
	CNG Tx Level • Fixed level for fixed CNG • Minimum level for adaptive CNG
	Tx FDCNG maximum level
AER NLP Limits	NLP linear attenuation maximum ERLE
	NLP center clipper maximum ERLE
	NLP total linear attenuation minimum
	NLP Rx linear attenuation minimum
	NLP Rx linear attenuation maximum
	NLP Tx linear attenuation minimum
	NLP Tx linear attenuation maximum
AER FDNLP	FDNLP delay
	FDCC low-frequency boundary 1
	FDCC low-frequency boundary 2
	FDCC high-frequency boundary 1
	FDCC high-frequency boundary 2

Table 2-2 AER and AGC Configurable Parameters (Part 3 of 3)

Component	Configurable Parameters
AER ASNR	ASNR frequency boundary 1
	ASNR frequency boundary 2
	ASNR maximum attenuation if frequency band 1
	ASNR maximum attenuation if frequency band 2
	ASNR maximum attenuation if frequency band 3
	ASNR signal update rate lower bound
	ASNR signal update rate upper bound
	ASNR noise threshold
	ASNR hangover
AGC	AGC saturation detection threshold
	AGC saturation detection hangover
	AGC maximum gain swing
	AGC maximum mic gain
	AGC minimum mic gain
End of Table 2-2	

For more information, see "AER Internal Parameters" on page 2-50 and "Default Parameter Settings" on page 2-63.

2.1.5.4.1 AER Filter Length or Tail Length

This corresponds to the maximum duration in ms of the AER echo tail model. AER adaptively determines a linear correlation between the Rx and Tx signals based on the relative delay between these two signals. The tail length is the maximum range of this delay in the linear correlation that AER models. The echo tail model is the set of delay-dependent coefficients that collectively model this correlation. A distinct optimal tail length should be determined for different phone modes.

Due to room reverberation, the microphone/ADC may detect echo long after a receiver/DAC pulse. AER can continue to cancel echo caused by a DAC pulse for a time limited by the AER tail length parameter. AER NLP switched attenuation is required to reduce a DAC pulse echo that is longer than the tail length. Longer tails are needed for lower ERLs (that is, louder echoes) and rooms with longer reverberation times. The room reverberation time measures how long acoustical sound decays by 60 dB. Specifying a shorter maximum tail length can result in faster convergence in some cases. However, AER normally uses an "active" tail length shorter than the configurable maximum value when this is optimal. Optimizing a distinct tail length for each receiver volume level is not recommended.

The tail length available depends on the phone mode AER is configured for:

- Handsfree: Range from 20 to 200 ms, in increments of 20 ms
- Handset: Range from 4 to 20 ms, in increments of 4 ms
- Headset: Range from 20 to 60 ms, in increments of 20 ms

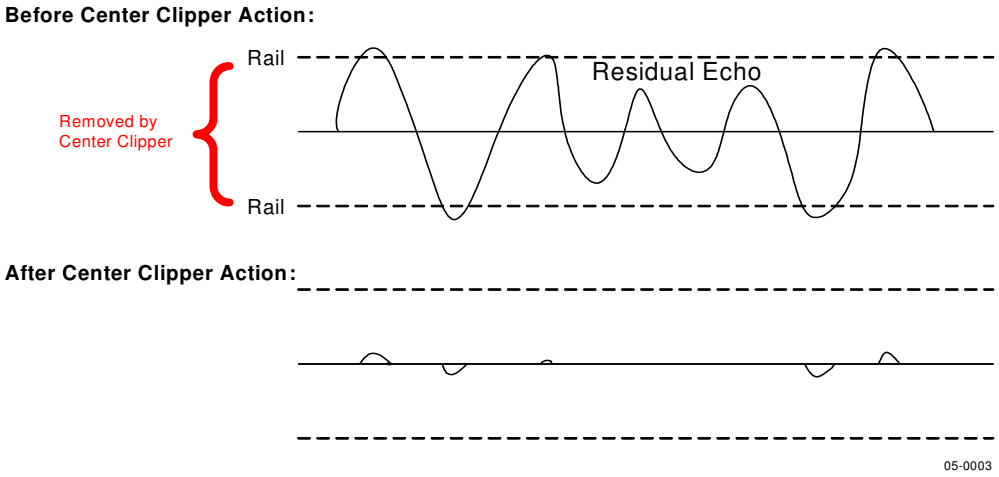
Default values are 200 ms for handsfree, 8 ms for handset, and 60 ms for headset. Increasing the values affects reconvergence times. For example, a longer value for handset can cause problems because the handset is subject to frequent movement during conversation; this movement causes frequent echo path changes, so faster convergence is desirable.

2.1.5.4.2 NLP Send Path Center Clipper Aggression

This parameter specifies the AER Tx path NLP center clipper rail amplitude scaling. The clipper rail level is computed based on the estimates of the residual echo signal, which depends on an estimate of the current ERLE. Clipper rail-level computations are not affected by the configurable parameter setting of the NLP linear combined loss target. Center clipping is applied to the signal before the NLP linear attenuation is applied. Thus the Tx NLP linear attenuation and Tx NLP center clipper attenuation are independent.

The clipper aggression is specified in 3 dB steps in the range [–30, 30]. Increasing the clipper aggression results in a larger clipper rail and more clipper attenuation. For best performance, clipper aggression is sometimes set to higher values at higher speaker volume settings.

Figure 2-3 AER NLP Center Clipper Action



2.1.5.4.3 NLP Linear Combined Loss Target

This parameter determines maximal NLP attenuation of the subdominant path, *not including the contribution of the clipper*. This number includes the total attenuation from ERL, ERLE, and NLP linear attenuation. Based on the estimates of ERL and ERLE, the NLP linear attenuation can be computed. The computation of NLP linear attenuation does not depend on the clipper settings.

The following formula shows how the configurable NLP combined loss target determines the linear attenuation, G (see Figure 2-1 on page 2-3):

$$-20*\log_{10}(\text{nlp_clt}/2^{15}) = \text{ERL_est} + \text{ERLE_canc} - 20*\log_{10}(G/2^{15})$$

For example, assume nlp_clt = 325 is configured corresponding to a $-20*\log_{10}(nlp_clt/2^{15}) = 40$ dB combined loss target. Assume the AER estimate of ERL = 6 dB (that is, the echo in the Tx path output is 6 dB less than the Rx path input when AER is disabled. If subtracting the linearly predicted echo attenuates echo by 28 dB, then $40 = 6 + 28 - 20*\log_{10}(G/2^{15})$. Therefore, the AER applies 6 dB of linear attenuation to the subdominant path by multiplying the signal by one half, or $G = 2^{14}$ in Q15 format. The AER ERLE_canc value can be seen from P[10] in Table 2-7 "System Response to ec_debug_stat" on page 2-42.

2.1.5.4.4 AER Send Path Digital Gain (aer_tx_dg)

The AER Tx digital gain is applied just before the AER Tx path NLP. This gain (aer_tx_dg) allows the ADC PGA gain (tx_ag) to be set independently to optimize the digital dynamic range of AER input in the transmit path. Requirements for the overall loudness ratings of the send path determine the sum of aer_tx_dg and tx_ag.

For example, assume the phone has a good overall loudness, a negative ERL, and the ADC PGA gain is high enough that persistent nonlinear flat top saturation of the ADC output inhibits convergence of the linear AER echo tail model. This problem can be fixed without changing the overall Tx path loudness by lowering tx_ag and increasing the aer_tx_dg. Thus, before applying aer_tx_dg, the AER input digital dynamic range is optimized for AER tail model convergence by adjusting tx_ag. After the aer_tx_dg is applied, the AER Tx NLP input has the correct adjustment for comparing Rx and Tx speech powers. When AGC adaptation is enabled, it automatically adjusts tx_ag for current conditions without changing the sum of tx_ag and aer_tx_dg; this maintains the overall loudness.

The value of aer_tx_dg is specified in the range [–200, 200] and measured in 0.5 dB steps. Default value is 0.

2.1.5.4.5 AER Receive Path Digital Gain (aer_rx_dg)

This gain has two purposes:

1. To provide louder receiver volume levels when the DAC PGA Rx analog gain (rx_ag) is already at maximum.

   To increase receiver volume levels, it is best to limit rx_ag to –1.5 dB for the TNETV105x and AIC2x hardware codecs to avoid DAC induced flat-top saturation distortion. To obtain various loudness levels, set aer_rx_dg=0 dB and increase rx_ag from a minimal value up to –1.5 dB. Then fix rx_ag=–1.5 dB and increase aer_rx_dg to further increase the Rx path loudness.

2. To avoid the breakdown of echo removal due to hardware enclosure nonlinearities, specifically when the far end is very loud.

   To eliminate the top (loudest) part of the Rx path dynamic range, increase the AER Rx digital gain while lowering DAC Rx analog gain, making sure that the sum of the two gains stays fixed to obtain a desired Rx path loudness. For example, reduce the maximum DAC PGA rx_ag from –1.5 dB to –6.0 dB and

increase aer_rx_dg by 4.5 dB. This reduces the maximum possible echo by 4.5 dB. Even at increased overall volume settings, the analog part of the gain is hard limited to a maximum "safe" limit with respect to receiver distortion and echo, but the higher volume level settings still amplify weak Rx input signals.



NOTE—Limiting the signal by adjusting fixed gains as described is called "hard limiting". Limiting the signal using HLC is called "soft limiting."

HLC allows the maximum echo to be very loud for a brief period before HLC attenuates a large Rx input signal.

The value of aer_rx_dg is specified in the range [–200, 200] and is measured in 0.5 dB steps. Default value is 0.

2.1.5.4.6 Minimum Noise Power Estimates for Tx / Rx Paths

This is used to make decisions regarding NLP break-in by constraining the minimum total Tx /Rx path signal power. Excessive noise picked up by the near end microphone may need a higher Tx noise power estimate to limit break-in of near end noise. For receive break-ins to occur with more or less Rx signal power, raise or lower the Rx noise power threshold respectively.

To convert a configurable minimum noise power estimate into dBm0 units:

$$f(x) = 10*\log10\ [x/(80 \cdot 2^{22})]$$

The default Rx/Tx noise minimum estimate is 5380, which corresponds to –48 dBm0 = f (5380).

2.1.5.4.7 Hangover to Switch Away from the Tx or Rx Paths

If the AER state machines first indicate that the Tx path is dominant, then Rx path speech becomes dominant, the switch to remove Rx NLP attenuation and permit Tx NLP attenuation should not happen immediately. If switching occurs too quickly, speech in the Tx path can be attenuated between syllables, which is perceptually annoying. Therefore, a hangover period is associated with the switchover. Only after the hangover period expires can the switching away from the Tx path begin. The hangover is in 5 ms units. The default value of the Tx hangover period is 10, which gives a duration of 50 ms.

After Rx path dominance ends, the lingering echo may not have decayed enough after a hangover period that is too short.The default value of the Rx hangover period 30, which gives a duration of 150 ms.

2.1.5.4.8 Four Linear Gain Switching Slew Rates

Four configurable parameters are available to control the linear attenuation slew rate, a time constant that determines the speed at which the linear attenuation gain can change. Table 2-3 lists the gain transition half life.

Table 2-3 Gain Transition Half Life

k value	Gain transition half life in milliseconds
1	5
2	10
3	24.1
4	51.9
5	107.4
6	218.3
7	440.1
8	883.8
9	1771.0
10	3545.4
End of Table 2-3	

The k value is configurable for four distinct transitions. The Tx NLP linear attenuation slew rate time constant, which is applied when the Tx path becomes subdominant or dominant, is given by `aer_nlp_tx_in_tc` and `aer_nlp_tx_out_tc`.

The Rx NLP parameters for Rx becoming subdominant or dominant are `aer_nlp_rx_in_tc` and `aer_nlp_rx_out_tc`.

Often during conversations with only single talk, pauses occur between transitions from far-end speech to near-end speech or vice versa. During these pauses, AER goes to an idle state where none of the four linear attenuation slew rate parameters have any effect. When this happens, the AER NLP linear attenuation is removed completely from the dominant path in 10 ms and applied completely to the subdominant path in 10 ms.

During fast transitions of the dominant speech direction (Rx or Tx) and during double talk, these sharp 10 ms transitions can be perceived as annoying artifacts for phones with relatively low linear attenuation. What is otherwise nearly perfect full-duplex performance can be marred by *sudden* signal level after a transition of the dominant path.

The default values of all four slew rate time constants are one, which lets NLP finish transitions within one frame.

2.1.5.4.9 Gain Splitting Time Constant

When AER is configured to disable the “duplex stabilizer” and AER detects double talk, gain splitting can occur. Even during double talk, one of the two paths (Tx or Rx) is evaluated by AER to be subdominant.

For example, assume there is 6 dB of total linear attenuation, the Rx path is considered to be subdominant, and AER determines that double talk is taking place. The linear attenuation in the Rx path changes gradually from 6 dB to 3 dB and the linear attenuation in the Tx path changes gradually from 0 dB to 3 dB. The speed at which the total linear attenuation is split between the two paths during double talk is controlled by the parameter `aer_gain_split_tc`.

Gain splitting requires AER to gradually let the subdominant and dominant gains to the square root of the total linear attenuation. (Division by 2 for a gain in logarithmic form (dB) requires the square root for the linear gain.) The algorithm for gradually converging to the square root is not linear, so a table of half lives is not applicable here. The half life required for converging to the square root increases by roughly a factor of two for each time the gain splitting time constant is increased by one.

The default of the gain-split time constant is 5.

2.1.5.4.10 Threshold for Rx Path Non Linearity Detection

This parameter signals AER that nonlinear breakdown may occur when the digital signal level driving the handsfree speaker exceeds a specified threshold. Whether the threshold is exceeded is determined downstream of the AER Rx digital gain and Rx equalizer (Figure 2-1 on page 2-3). This parameter helps attain the best degree of full-duplex performance when the handsfree speaker is driven below a fixed threshold and makes sure that there are no echo leaks due to nonlinear breakdowns that occur when the handsfree speaker is driven by an abnormally loud signal. This parameter is specified in the absolute signal amplitude level. The default value of this parameter is 32000. This corresponds to a sine wave peak value at $20*\log_{10}(32000/2^{15})+3=$ 2.8 dBm0.

2.1.5.4.11 Tx/Rx Gain Change Synchronization Delay and Interpolation Count

An important function of the AGC is to control the ADC PGA Tx analog and AER Tx digital gains, if the AGC is configured to be in adaptive mode. AGC maintains a healthy digital dynamic range for AER, so the AER send path echo input is safely below saturation but large enough to not suffer excessively from ADC digital quantization errors. When adaptive AGC invokes a microphone ADC PGA Tx analog gain transition, no audible click should be produced.

A parameter enables AER “click removal” by specifying the Tx path sample delay required for the gain transition to be complete. In general, the optimal value of this parameter can change with the Tx path equalization filter, sampling frequency, hardware codec, and software implementation of the gain interface. During ADC PGA analog gain changes, some Tx samples collected during the transition to a new gain may get garbled or distorted. Such artifacts should be removed by interpolating these

samples from the send path data before and after these garbled samples. Therefore, for a sample delay of J samples and interpolation count of K samples, after a gain change signal samples between the $(J)^{th}$ and $(J\text{-}K)^{th}$ samples should be discarded and replaced using interpolation.

NOTE—Initially, you should not try to change these gain-change parameters. These are provided by Texas Instruments. Use of both the Tx path Equalizer and adaptive AGC may motivate changing these parameters.

2.1.5.4.12 AER Noise Guard Hangover Period

If Noise Guard is enabled, before starting to attenuate the noise signal, Noise Guard lets the hangover specified by this parameter expire to make sure that what seems like near-end noise is not just a pause in near-end speech. The hangover period has a default value of 1000 ms, and a range of [1-2500] ms.

2.1.5.4.13 AER Noise Guard Send Noise Level

This parameter specifies the level of noise to send when near-end speech is not present. If Noise Guard detects near-end speech, it should pass the signal unaltered. However, if it detects noise at the near end, Noise Guard should attenuate the signal locally to get as close as possible to the selected noise level. If the estimated noise level is at or below the specified noise level, Noise Guard should not do anything. The noise level parameter is specified in dBm0. The default is (–65) dBm0. The recommended range of send_noise_level values is (–45) to (–75) dBm0.

2.1.5.4.14 AER Noise Guard Ramping Period

This parameter controls how fast Noise Guard ramps in to the send_noise_level from the measured near-end noise level. Along with the Noise Guard hangover period, it determines the total duration, in ms, for the Noise Guard action to take full effect. When near-end speech is detected, the noise–speech transition is instantaneous. The default value of the ramp-in period is 800 ms with a range of [1-2000] ms.

2.1.5.4.15 AER Equalizer Filter Parameters

There is support for signal equalization in the Rx and Tx paths of the AER (Figure 2-2 on page 2-5). AER equalization filters are intended to help the speaker and microphone output's spectral response to meet specified masks as closely as possible, and to attenuate frequencies that may induce nonlinearity on the microphone input and output. Equalization is done using a combination of an infinite impulse response (IIR) filter and a finite impulse response (FIR) filter, which can be used together to shape the frequency response (Figure 3-1 on page 3-3). The Rx and Tx equalizers are independent; each one is described by a set of 42 parameters.

2.1.5.4.16 AER HLC Parameters

High Level Compensation (HLC) attenuates peaks in the receive (Rx) signal to ensure that the incoming signal from the packet network does not cause saturation or nonlinear distortion at the speaker output (Figure 2-2 on page 2-5).

When HLC is enabled, you can configure the target level (threshold). The threshold parameter is defined in 0.5 dB steps between –48 dBm0 and +3dBm0. If the HLC output signal exceeds the HLC threshold, the HLC output ramps down as the HLC attenuation increases. If the HLC output signal is below the HLC threshold, the HLC output ramps up as the HLC attenuation decreases. To avoid abrupt audible changes during the attenuation action, HLC has parameters for ramp_down and ramp_up time constants. The ramp_down time is fast and non-configurable (fixed at 10 ms per dB) and the ramp_up time should be slower.

Another configurable constant governs the rate at which the current Rx signal power estimate is updated. Customers are normally advised to change the target level only.



NOTE—The Dynamic Range Compressor (DRC) full-band compressor and limiter are upgraded versions of the AER HLC and AER Rx SLIM, respectively. When DRC is activated, make sure that AER HLC and AER Rx SLIM are disabled or inactive. For more information, see Chapter 4 "Dynamic Range Compression (DRC)" on page 4-1.

2.1.5.4.17 Signal Limiter

The Rx signal limiter for AER is just after HLC in the Rx path, and the Tx signal limiter is at the end of the AER Tx path. The DSP uses default values of zero which disable the signal limiters. The output of the signal limiter saturates at a level corresponding to the peak level of sine waves with the following signal powers. Table 2-4 lists the parameter values for signal limiting and the corresponding saturation level for each value.

Table 2-4 Signal Limiter Parameter Values and Corresponding Saturation Levels

aer_tx_slim_mode or aer_rx_slim_mode parameter	Saturation level
0	Signal limiter disabled
1	-10.5 dBm0
2	-8.2 dBm0
3	-6.6 dBm0
4	-4.4 dBm0
5	-2.2 dBm0
End of Table 2-4	

The signal limiter starts distorting speech at 7 dB below the threshold level. A hard limiter will flat-top saturate at a fixed level, but the signal limiter must attenuate the signal at lower levels to avoid a sharp discontinuity in the slope when the input signal reaches the signal limiter output saturation level. Therefore, the signal limiter output signal will be attenuated well before the input signal reaches the saturation level, and this level-dependent attenuation will cause distortion.

The digital and analog gains in both the Rx and Tx paths can be altered to create a hard limiter. A hard limiter will have relatively less distortion for input below the saturation level. The soft limiter will have relatively less distortion for input above the saturation level.

NOTE—The Dynamic Range Compressor (DRC) full-band compressor and limiter are upgraded versions of the AER HLC and AER Rx SLIM, respectively. When DRC is activated, make sure that AER HLC and AER Rx SLIM are disabled or inactive. For more information, see Chapter 4 "Dynamic Range Compression (DRC)" on page 4-1.

2.1.5.4.18 Frequency Domain NonLinear Processor (FDNLP)

The Frequency Domain NonLinear Processor (FDNLP) has three capabilities:

- Frequency Domain Center Clipper (FDCC)
- Spectrally-matched Frequency Domain Comfort Noise Generator (FDCNG)
- Adaptive Spectral Noise Reduction (ASNR)

NOTE—FDNLP cannot be activated when AER is configured to use time domain NLP.

The Frequency Domain Center Clipper (FDCC) customizes the clipper level independently according to the FE speech spectrum. During double talk, FE and NE speech spectrums typically do not match well; this provides the opportunity to selectively increase center clipper spectral attenuation where the residual echo is large and decrease attenuation where NE speech is large. Spectral focusing of NLP attenuation on echo results in better full-duplex performance. In addition, the FDCC does not introduce sample-to-sample discontinuities; this normally produces less distortion relative to the pre-existing AER time-domain NLP center clipper.

The Frequency Domain Comfort Noise Generator (FDCNG) produces comfort noise that matches the background noise in both level and spectrum, and maintains the continuity of the background noise after echo cancellation and nonlinear processing. A maximum and minimum FDCNG level is configurable for each spectral bin. Thus if the FDNLP adaptively tracks a noise estimate that is too high or too low relative to the configured constraints, white noise at the high or low CNG noise limit will result.

Adaptive Spectral Noise Reduction (ASNR) attenuates NE Tx path stationary noise, which may include contributions from the following noise sources, among others:

- External NE acoustic noise
- Microphone vibrations
- Microphone EM pickup
- ADCPGA amplifier noise
- Noise from the mic bias
- Limited ADC SNR

Noise Guard cannot be activated if ASNR is enabled. Thus ASNR takes precedence over Noise Guard. ASNR attenuation limits can be configured for three distinct spectral bands. This can be used to limit the aggression of ANSR because excessive attenuation can produce more speech distortion artifacts. These limits also can enable the Tx path to pass an adequate level of DTMF or signaling tones when ASNR is enabled.

ASNR can be configured to apply another type of attenuation limit. Attenuation can be limited to not reduce the ASNR output below a configurable noise threshold. If ASNR input goes below this noise threshold, ASNR is effectively disabled. This is advantageous because noise reduction is not needed when the noise is sufficiently low, and the speech distortion artifacts due to ASNR are unjustified when noise is sufficiently low. In some cases, the noise threshold needs to be well below the Tx CNG level. For example, this type of configuration can help mask low residual echo and tonal EM pickup.

Figure 2-4 Example of FDCC Center Clipper Rail Values in Different Frequency Ranges

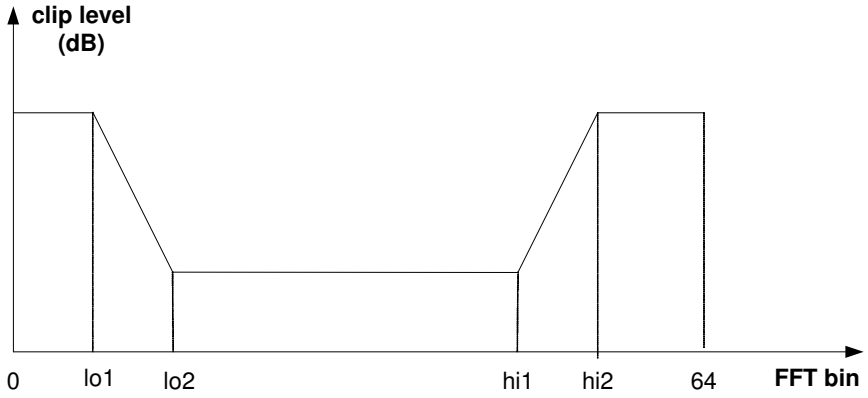


Table 2-5 Frequency Domain Nonlinear Processor Configurable Parameters (Part 1 of 2)

Configurable Parameter	Description
Tx CNG level	For a Tx TDNLP, this number is the fixed level of CNG white noise. For a Tx FDNLP, this number is the minimum level of total adaptive CNG noise, which means that FDCNG noise will always be above this level even though background noise is lower than this level.
FDNLP maximum CNG level	FDCNG adaptively matches Tx NE stationary noise in level and spectrum, so FDCNG output may become very high at times. This parameter will limit the Tx FDCNG signal power (which is heard during Tx NLP attenuation). This parameter sets a limit on maximum power in dBm0 units for the adaptive FDCNG noise estimate & FDCNG output. Range is [-80, 0] dBm0. Default is -55 dBm0.
FDNLP delay	FDNLP delay in msecs. It may be set to 3, 4, 5, or 6. Default is 5 msecs. FDNLP FDCC and ASNR may produce less unwanted noise and speech artifacts for a longer delay.
FDCC low-frequency boundary 1	High FDCC attenuation is applied for frequency bins less than txfdnlp_bin_lo1*62.5 Hz, because no echo subtraction ERLE is anticipated for these bins. First of four Tx FDNLP parameters affecting FDCC. Must be greater than or equal to 0. Default 1. See Figure 2-4.
FDCC low-frequency boundary 2	Low FDCC attenuation is applied for frequency bins greater than or equal to txfdnlp_bin_lo2*62.5 Hz because good echo subtraction ERLE is anticipated for these bins. Must be greater than or equal to tx_fdnlp_bin_lo1. Default 6. See Figure 2-4.
FDCC high-frequency boundary 1	Low FDCC attenuation at frequency bins less than or equal to txfdnlp_bin_hi1*62.5 Hz. Must be greater than tx_fdnlp_bin_lo2. Default 56. See Figure 2-4.
FDCC high-frequency boundary 2	High FDCC attenuation at frequency bins greater than tx_fdnlp_bin_hi2*62.5 Hz. Must be greater than or equal to tx_fdnlp_bin_hi1. Default 60. See Figure 2-4.
ASNR frequency boundary 1	Boundary 1 for ASNR attenuation limit: corresponding to fbin1*62.5Hz. Default is 10 (625Hz). Range 1 to 62.

Table 2-5 Frequency Domain Nonlinear Processor Configurable Parameters (Part 2 of 2)

Configurable Parameter	Description
ASNR frequency boundary 2	Boundary 2 for ASNR attenuation limit: corresponding to fbin1*62.5Hz. Default is 32 (2000Hz). Range 2 to 63, 0<asnr_fbin1_lim<asnr_fbin2<64
ASNR maximum attenuation if frequency band 1	Maximum attenuation (dB) for frequency band [0, asnr_fbin1_lim-1]*62.5 Hz. Default is 12. Range is [0, 90]. Expected range [6, 18].
ASNR maximum attenuation if frequency band 2	Maximum attenuation (dB) for frequency band [asnr_fbin1_lim, asnr_fbin2_lim]*62.5 Hz. Default 12. Range is [0, 90]. Expected range [6, 18].
ASNR maximum attenuation if frequency band 3	Maximum attenuation (dB) for frequency band [asnr_fbin2_lim+1, 64]*62.5 Hz. Default is 12. Range is [0, 90]. Expected range [6, 18]. The average ASNR attenuation applied in the third band, subject to this constraint, is also applied uniformly to the Tx high band. This enables ASNR to uniformly attenuate white noise.
ASNR signal update rate lower bound	ASNR adaptive speech signal estimate inverse update rate lower bound in Q15. The actual inverse update rate is varied internally by AER between the configured minimum (this parameter) and maximum constraints. A higher inverse update rate reduces frame to frame changes yielding greater temporal smoothing and less time varying ASNR distortion artifacts during speech or noise. A lower inverse update rate yields quicker speech onset "breaking in" to remove ASNR attenuation. Default is 29818 (0.91). Range is 1 to 32766. Expected range is 32767*[0.75, 0.95].
ASNR signal update rate upper bound	ASNR adaptive speech signal estimate inverse update rate upper bound in Q15. Default is 31129 (0.95). Range is 2 to 32767. Expected range is 32767*[0.8, 0.98], 0<asnr_sig_upd_rate_min<asnr_sig_upd_rate_max
ASNR noise threshold	This is the lowest signal level input to ASNR for which ASNR would be effective. ASNR will not reduce output noise below this level. When noise is already below this level, ASNR will not apply any attenuation. ASNR spectral noise estimate will continue to update even if Tx signal is below this noise threshold. Default is -75dBm0 and the valid range is [-80, -40] dBm0. If set to any value less than -80, this parameter has no effect. Expected range is [-75, -45].
ASNR hangover	After noise goes above asnr_noise_thresh, ASNR waits this amount of time, in 10 msec units, before applying attenuation again. Default is 150 (1.5 seconds). This avoids attenuating the onset of weak speech when the noise is very low. Range is [0, 32767], expected range is [0, 200].
End of Table 2-5	

2.1.5.4.19 Comfort Noise Generator (CNG)

The AER contains a comfort noise generator (CNG) in the Rx and Tx paths of the AER NLP. The CNG adds either fixed-level random white noise or spectrally-matched noise to the voice signal so that listeners at both ends of the call do not perceive the connection as 'dead' due to the AER NLP attenuation of background noise. The Rx path has only fixed-level CNG and the Tx path has either fixed-level or spectrally-matched CNG depending on user configuration.

The CNG signal level is configured separately for the Rx and Tx paths. The amount of CNG signal that the AER adds to the voice signal is scaled by the amount of attenuation that the AER NLP is currently applying. When the AER NLP applies no attenuation, no CNG signal is output. However, when the AER NLP attenuates the voice signal, this can damage the natural background noise level; the AER CNG replaces the background noise.

The AER CNG

- Does not degrade the SNR of weak speech during single talk because CNG output is zero for the dominant path transmitting speech during single talk
- Does not clip the onset of subtle weak speech because CNG signal is mixed in and does not overwrite the voice signal

- Reduces the severity of abrupt NLP switching transitions without adding abrupt on/off transitions because the amount of added CNG signal varies continuously with the amount of AER NLP attenuation. AER calculates how much signal power would have been lost due to NLP attenuation if only comfort noise at the configured CNG level were input to the NLP. The same amount of comfort noise is then added to the NLP output.
- Cannot introduce saturation into the signal no matter how it is configured

An exception to the preceding "normal" CNG description is possible when AER is configured to generate "forced" CNG. In this case, the voice signal input to AER is deleted and overwritten with comfort noise at the full configured CNG level at all times. Forced CNG is available in both the Rx and Tx paths. Some exemplary uses of forced CNG include the implementation of mute in the Tx path, and "self calibration" of the echo path using forced CNG in the Rx path to avoid startup convergence after a power cycle.



NOTE—Tx forced CNG is not available when FDNLP is used.

For fixed CNG, the Tx or Rx CNG level parameter determines the overall white noise power level. However, for spectrally-matched (or adaptive) CNG, the Tx or Rx CNG level parameter refers to the minimum power level. Adaptive CNG has another parameter, the maximum CNG power, which sets a limit on the maximum power of Tx CNG.

2.1.5.4.20 Configurable NLP Attenuation Limits

The AER NLP Rx linear attenuation, Tx linear attenuation, and Tx center clipper rail take on values that are normalized with respect to an internal estimate of the ERL (echo return loss) which includes the effect of ERLE obtained from subtracting the linearly predicted echo. As the tail model converges better, the ERLE estimate increases and the attenuation from these three NLP attenuation mechanisms compensates by decreasing together. Conversely, when AER detects that cancellation performance is degrading, the ERLE estimate decreases and the three NLP mechanisms increase attenuation together. While some degree of adaptation in the NLP attenuation is desirable, under some circumstances it is desirable to limit the variation in NLP attenuation. Therefore there are configurable parameters that place limitations on the range over which NLP attenuation varies with estimated ERLE.

2.1.5.4.21 Configurable AGC Parameters

Table 2-6 Configurable AGC Parameters

Parameter	Description
AGC Saturation Detection Threshold	This is the signal level of the Tx-path ADC output sample, above which saturation is declared. The default value is 32500.
AGC Saturation Detection Hangover	This is the period, in milliseconds, before AER is informed
AGC Maximum Gain Swing	In case of max gain settings on HF on both ends, adaptive AGC tries to control saturation by lowering tx_ag and increasing aer_tx_dg. Excessive use of aer_tx_dg can cause high amount of digital/quantization noise. This parameter sets the maximum amount of gain swing the adaptive AGC can cause.
AGC max mic gain	Adaptive AGC will not request gains that are above this parameter.
AGC min mic gain	Adaptive AGC will not request gains that are below this parameter.

2.1.6 Control Commands

Control commands belong to two classifications:

1. There are "`set ipp` ..." commands that (after activate/commit/reboot) automatically restore specified configurable parameters under various circumstances. These commands are useful for storing final optimized parameters.

   For information about using these commands, see the *Application Services Command Reference Manual*. For a list of available commands online, type "`show ipp`", "`show ipp_prof 2`", "`show ipp_gains 2`" into a Texas Instruments IP-phone serial port interface and see how the exemplary Texas Instruments IP-phone operates.
2. There are "`dsp 0` ..." on-the-fly commands that alter a configurable parameter immediately, but generally must be reissued after subsequent control messages. Initially, these commands are useful for calibration and optimization.



NOTE—To generate a build that supports the <dsp 0 ...> commands, you must set the flag for `GG_INC_DIM_CHAN_IF` to 1 to enable DIMCHAN API support in IP Phone, then recompile the microcode.

Some "dsp" AEC/AER debug commands used for calibration/debug purposes use the dimchan interface, such as:

- dsp <tcid> aerc
- dsp <tcid> aerp
- dsp <tcid> aer_asnr_config
- dsp <tcid> aer_fdnlp_config
- dsp <tcid> aer_hlc_ctrl
- dsp <tcid> aer_gain_chg_params
- dsp <tcid> aer_cng_nlp_params
- dsp <tcid> aert
- dsp <tcid> agc
- dsp <tcid> aer_nguard_ctrl
- dsp <tcid> aer_eq_ctrl
- dsp <tcid> gains

These commands may be required to calibrate and debug during IP Phone product development. When these commands are no longer needed and you want to reduce microcode memory requirements, you can set the flag back to the default value (0) in **ggtune.h**.



CAUTION—For older Texas Instruments IP Phone software (based on the TNETV1001), the hardware codec driver is not compatible with the PG1 (revision 1) AIC22. The software works correctly only when the AIC22 Register 13 bit 1 is set to 1 to disable the valid data flag bit; the PG1 AIC22 cannot operate with this value.

2.2 Parameter Optimization

The following list gives the general tasks for parameter optimization.

1. Choose two phones that support all phone modes of interest (handsfree, handset, headset). Test other phones later to check for variations and tolerances.
2. For each phone and phone mode, choose the total send and receive path gains with knowledge of the resulting Send Loudness Rating (SLR) and Receive Loudness Rating (RLR) values. Determine the send path gain distribution. See "Send and Receive Path Gain Adjustments" on page 2-22.
3. Determine if Tx and Rx equalization is required. If so, determine the parameters of the filter using the "Signal Equalizer MATLAB Design Tool" on page 3-1 (for more information, see "Equalization" on page 2-26). Repeat steps 2 and 3 until you achieve the best results.
4. Select one phone of the pair and use it in handset mode to calibrate the handset and handsfree parameters of the other phone. Compare with the default parameters.

   For each phone mode, optimize internal AER parameters for acceptable echo reduction and application-specific goals. Determine the receive path gain distribution for nominal speaker volume.

   Some details of the optimization process will vary according to the application for which the AER is used. To help sort out these nuances, this document contains the following sections:

   a. "Optimizing for Handsfree Full-Duplex Operation" on page 2-31
   b. "Optimizing for Handsfree Half-Duplex Operation" on page 2-34
   c. "Optimizing Handset and Headset Parameters" on page 2-40
5. Repeat step 4 for the range of available receive path gains. Configure the phone to automatically use the optimal parameters found in step 4.

   Typically, initial testing to optimize AER NLP clipper and combined loss target parameters is done at nominal and maximum volume. These parameters can be unchanged below nominal and interpolated for volume levels between nominal and maximum. Further refinements are made as needed during subsequent testing.
6. Optimize AGC and AER Noise Guard to best complement AER performance. See "Automatic Gain Control (AGC)" on page 2-38 and"AER Noise Guard" on page 2-36.

2.3 Send and Receive Path Gain Adjustments

This section contains the following topics:

- "Setup Checking" on page 2-22
- "Gain Calibration Procedure" on page 2-23
- "Equalization" on page 2-26
- "High Level Compensation" on page 2-29

2.3.1 Setup Checking

With multiple AER instances, multiple DSP telephony channel IDs (TCIDs), multiple timeslots, and two hardware codec channels, there is potential for misconfiguration. Before you change gains to achieve a nominal target loudness (send, receive, sidetone), it is useful to do a sanity check procedure. This is also a useful exercise to become familiar and efficient in changing gains.

See "NMM Commands Relevant to AER Performance" on page 2-41 for more information about NMM commands.

Procedure 2-1 Checking the Setup

Step – Action

1. If you are not already in debug mode, enter the following command:
   ```
   dbgcmd
   ```
2. For acoustical testing, see "Configuring the IP Phone for Performance Assessment Tests" on page 1-43 to disable AER, AER Noise Guard and adaptive AGC.
3. At the MXP> prompt, enter the following command:
   ```
   show ec_debug_stat <tcid>
   ```
   The ec_debug_stat table appears. (See Table 2-7 "System Response to ec_debug_stat" on page 2-42).
4. Record the values of P[43] and P[44] from the ec_debug_stat response.
5. Enter the following command, which sets tx_ag (0.5 dB steps):
   ```
   dsp <tcid> gains na na 0 na na na na
   ```
   Listen to the output to verify that this affects far-end speaker loudness.
6. Enter the following command:
   ```
   show ec_debug_stat <tcid>
   ```
   Look at the table to verify that the value of P[44]=tx_ag_cnt incremented.
7. Enter the following command, which sets rx_ag (0.5 dB steps):
   ```
   dsp <tcid> gains na na na –100 na na na
   ```
   Listen to the output to verify that this affects near-end speaker loudness.
8. Enter the following command:
   ```
   show ec_debug_stat <tcid>
   ```
   Look at the table to verify that the value of P[43]=rx_ag_cnt incremented.
9. Reset the phone to recover default gains, then enter the following command, which sets aer_tx_dg (0.5 dB steps):
   ```
   dsp <tcid> aerp na na na 5 na na na na na na
   ```
   Listen to the output to verify that this affects far-end speaker loudness.

10 Enter the following command:

```
show ec_debug_stat <tcid>
```

Look at the table to verify that P[15] = aer_tx_dg = 40 (1/16 dB steps).

11 Enter the following command:

```
show gains <tcid>
```

Look at the table to verify that aer_tx_dg=5 (0.5 dB steps).

12 Enter the following command, which sets aer_rx_dg (0.5 dB steps):

```
dsp <tcid> aerp na na na na 6 na na na na na
```

Listen to the output to verify that this affects near-end speaker loudness.

13 Enter the following command:

```
show ec_debug_stat <tcid>
```

Look at the table to verify that P[16] = aer_rx_dg = 48 (1/16 dB steps).

14 Enter the following command:

show gains <tcid>

Look at the table to verify that aer_tx_dg=5 and aer_rx_dg=6 (0.5 dB steps).

End of Procedure 2-1

2.3.2 Gain Calibration Procedure

A non-zero aer_tx_dg>0 and aer_rx_dg>0 are needed the most in handsfree mode, but handset and headset mode may also require this at higher speaker volume levels. For example, some manufacturers support handset speaker volumes that produce the loudest recommended loudness ratings for the hearing impaired (TIA.EIA-504 recommends 12 to 18 dB gain above nominal). At maximum volume, often a handset can be laid on its side and crudely used as a half-duplex handsfree speaker, although its send path is too weak. The handset at maximum volume should remain stable against AEC instability when placed face down on a hard surface, as tested in TIA.EIA-810B. Under these circumstances, the handset ERL can become negative and require aer_tx_dg>0. Similarly, many headsets will lack sufficient efficiency to keep aer_rx_dg=0 and attain the maximum desired loudness at the maximum DAC PGA gain. Small negative levels of these digital gains can be used for loudness calibration fine tuning if there is need for a gain level between available analog DAC PGA levels., but this is normally not recommended.

```
MXP>dsp 0 gains
dsp [tcid] gains [tx_dg] [rx_dg] [tx_ag] [rx_ag] [rx_sec_ag] [lcd_g] [Side Tone]
```

Procedure 2-2 Calibrating the Gain

Step – Action

1 Set the loss plan channel tx and rx digital gains to 0, as follows:

```
dsp <tcid> gains 0 0 na na na na na
```

The rx_dg and tx_dg gains (see Figure 2-1 on page 2-3) are normally not used for a loss plan in an IP-phone. Normally loss plans are in gateways that connect digital and analog networks to compensate analog signal attenuation over a long transmission line. As the speaker volume level increases, Texas Instruments recommends increasing rx_ag to 0, then increasing aer_rx_dg (but not rx_dg).

A potential problem occurs for a half-duplex phone exposed to high near-end room noise and a weak far-end speech signal that prompts the user to increase the handsfree speaker volume to maximum. Increasing aer_rx_dg to obtain the highest signal levels would not help far-end break-ins, which require Rx path signal power to exceed the Tx path signal power at the Rx and Tx NLP, respectively. Rather than increasing rx_dg instead of aer_rx_dg, Texas Instruments recommends decreasing "rx_noise_min" (see "AER Internal Parameters" on page 2-50, Line 3) with increasing speaker gain if better Rx path break-in is desirable.

2 Set aer_tx_dg=aer_rx_dg=0, as follows:

```
dsp <tcid> aerp na na na 0 0 na na na na na
```

3 While measuring the SLR, vary tx_ag only to find a reference value for the sum of (tx_ag+aer_tx_dg) gains that most precisely achieves the relevant specification's center target for the overall SLR loudness rating.

```
dsp <tcid> gains na na tx_ag na na na na
dsp <tcid> aerp na na na aer_tx_ag na na na na na na
```

This will determine a constant C where, SLR = C –(tx_ag+aer_tx_dg)/2. (Dividing by two converts the 0.5 dB units into 1.0 dB units.)

a Use the "show gains" command to get the actual tx_ag, as the tx_ag request is rounded to the nearest available PGA gain, in generally 1.5 dB steps.

b At this point, you can optionally set aer_tx_dg gain to 0.5 dB or 1.0 dB to fine tune the SLR calibration to a standard.

c Record the SLR and (tx_ag+aer_tx_dg) values.

Note—If the primary goal is full-duplex performance, decrease tx_ag for all phone modes to adjust the SLR to a higher value within specified tolerances. (Even for a handset, this may enhance duplex performance at the highest handset speaker volume.)

For example, if the target value for handsfree SLR is 13±4 dB, decrease the gain to get SLR=16 dB. If phone-to phone microphone sensitivity tolerances are large, individual phone calibration too burdensome, and specification adherence more important than full-duplex enhancements, this may not be acceptable. However, if it is acceptable, this lowers echo levels of both the near-end *and* far-end phones, and increases full-duplex quality at both ends. Therefore, there is a motive to raise even the handset SLR above the central target, even though the near-end handset has excellent full-duplex performance at nominal default speaker volume.

4 Determine the total receive path gain (rx_ag+aer_rx_dg) needed for nominal default.

```
dsp <tcid> gains na na na rx_ag na na na
dsp <tcid> aerp na na na na aer_rx_dg na na na na na
```

As with the gain in the send path, this value is determined by measuring the overall RLR loudness rating, and then raising it to a higher RLR to increase full-duplex performance if desired. A typical requirement for the handsfree RLR is 16 dB ± 4 dB. For some constant D, RLR=D-(rx_ag+aer_rx_dg)/2. If more loudness or a lower RLR is needed, begin by using the maximum value rx_ag=−1.5 dB before raising aer_rx_dg above zero.

5 Using the commands from step 3, determine the optimal distribution of gains in the send path while keeping (tx_ag+aer_tx_dg) fixed. A non-zero positive value of aer_tx_dg is needed only when microphone ADC saturation is excessive during normal far end conversation (not shouting). The aer_tx_dg>0 gain is used to defer some of the send path gain downstream of AER input, so AER will not see nonlinear flat-top saturation distortion (due to the ADC input voltage rail limit) in its critical acoustic echo path.

6 With AER enabled during typical conversation from a far-end handset (calibrated in the microphone path), observe the **`show ec_debug_stat`** value P[42]=Sat_cnt to determine whether the near-end microphone ADC is saturating too much.

The goal is to optimize P[10]=log2_attn, and excessive saturation will inhibit this. Typically, only handsfree mode will need a decrease in tx_ag (and thus increase aer_tx_dg>0) to stop excessive saturation. The goal is to lower tx_ag by only enough to stop excessive saturation; lowering tx_ag increases ADC digital quantization noise for weak near-end speech.

There are three ways to do this:

a Observe log2_attn and Sat_cnt from **`show ec_debug_stat`** during normal far-end speech.

b For handsfree mode, enable AGC in addition to AER; after 10 minutes of normal two-way conversation, use the **`show gains`** command to see tx_ag and aer_tx_dg.

c Using the software on the far-end phone, generate 300, 500, and 1000 Hz test tones at a digital level of 0 dBm0. For example,

```
dsp <tcid> tone net 300 0
```

Observe the near-end Sat_cnt result.

7 If the near microphone is saturating frequently at nominal speaker volume, decrease tx_ag and increase aer_tx_dg by equivalent amounts. This preserves the SLR and reduces microphone ADC saturation due to acoustic echo.

Using different values of aer_tx_dg for different speaker volume levels is rarely advisable, but may be needed if very high speaker volume levels are to be available and the adaptive AGC feature is not used. A good rule for scaling aer_tx_dg across different speaker volume settings is

```
aer_tx_dg = max(0, C+rx_ag)
```

for some constant C while adjusting tx_ag to keep (tx_ag+aer_tx_dg) and the SLR loudness rating fixed. The loudest possible echo should vary linearly with the rx_ag gain. It is improper to assume this optimization step can always be harmlessly omitted for handsfree mode because of the AGC adaptive tx_ag/aer_tx_dg mode. Optimal AER speed of convergence is desired right after a power cycle, and it will take time for AGC to change a sub-optimal default tx_ag gain.

During initial development, set aer_tx_dg = 0 and rely on AGC adaptive mode to lower tx_ag and increase aer_tx_dg as needed.

Note—Testing to determine the optimal distribution of the send path gain between tx_ag and aer_tx_dg should only be pursued to obtain a level of resolution of 3 dB. Texas Instruments does not recommend more testing to statistically determine a finer resolution than 3 dB. The goal is that immediately after a power cycle, AER send path input will normally have a good digital dynamic range, with infrequent clipping, and not consistently delay optimal convergence until AGC (in adaptive mode) adjusts the tx_ag gain. Therefore, the phone should start with the best compromise between using the full dynamic range of the handsfree microphone ADC and maintaining a safety factor against excessive saturation of the ADC. After running awhile, AGC gain adaptation can customize the gains for the signal levels encountered.

8 See sections "2.4 Optimizing Parameters for Handsfree Operation" on page 2-30 and "2.5 Optimizing Handset and Headset Parameters" on page 2-40 to optimize AER internal parameters for performance by testing at normal conversational levels.

The optimal selection of gains is not necessarily completed after this point, especially for handsfree mode. You may find that the performance is good during normal conversational levels, but during far-end shouting, the echo reduction breaks down—due to hardware enclosure nonlinearities—and some degree of compression is needed. To completely eliminate the top (loudest) part of the receive path dynamic range, you can increase aer_rx_dg and lower rx_ag to keep the sum of aer_rx_dg + rx_ag fixed. However, this adjustment introduces flat-top saturation distortion. To eliminate this type of distortion but still prevent high signal levels from reaching the speaker, engaging the AER HLC is recommended. HLC will also have some unwanted artifacts, such as temporarily attenuating weak Rx input speech that follows high level Rx input, such as intermittent yelling.

End of Procedure 2-2

2.3.3 Equalization

As part of the standard SLR and RLR calibration procedure, you may also get the frequency-dependent transfer characteristics of the phone along the Tx and Rx paths with the intent of making sure that the measured response in either direction fits inside predefined masks for that phone mode, such as those defined by TIA-810B or 920 for handsfree, handset, and headset modes. The receive path equalizer is associated with characteristics of the speaker and the transmit path equalizer with characteristics of the microphone.

Group listen modes use the internal handsfree speaker with another speaker (handset/headset). In the current DSP architecture of Texas Instruments IP Phone builds, two instances of AER cannot run in parallel so the same signal is split and guided to the two transducers downstream of AER in the receive path. An equalization filter used with this instance would affect both the handsfree and the handset/headset speaker, which may not be acceptable. Therefore, the user may decide to disable Rx path equalization for group listen modes. Tx path equalization could still be done.

The following procedure describes how to get the equalizer parameters.

Procedure 2-3 Getting Tx and Rx Equalizer Parameters

Step – Action

1. Adjust the gains as described in the "Gain Calibration Procedure" on page 2-23.

   Keep the same gains for the rest of this procedure.

2. Obtain the Tx and Rx frequency responses using the measurements of transducer outputs.

   Make sure AER, AER Equalizers, VAD, and adaptive AGC are disabled when you take these measurements.

3. Determine whether the Rx and Tx responses meet the specifications of the masks defined by you or one of the standards in "Spectral Response Specifications" on page 1-12.

   If the response meets the specifications, equalization may not be required. However, if the response barely fits the masks, or if you want to meet some other spectral characteristics, you can continue with equalization.

4. If equalization is not needed in either or both directions, you can disable the corresponding equalizer(s).

   If equalization is needed, you must get the filter parameters using an independent MATLAB tool, which is described in "Signal Equalizer MATLAB Design Tool" on page 3-1. The tool provides the complete set of 42 parameters (18 for IIR, 19 for FIR, and 5 for gains).

5. When the equalization filter parameters are available, you must load them into the DSP using the available APIs, then enable the corresponding equalizer. You must select the correct sampling rate and direction (Tx or Rx) to look at the coefficients. Use the following set commands to set this information.

   - To set the equalizer parameters, use

     ```
     set ipp_prof <prof> aer_eq_params <tx|rx> <param1>..<paramn>
     ```

   - To set the status of the equalizer for that profile, use the following:

     ```
     set ipp_prof <prof> aer_eq <rx|tx|rxtx> <enable|disable>
     ```

     For example, to set the handsfree equalizer information for an 8-kHz transmit path, use the following:

     ```
     set ipp_prof 2 aer_eq_params tx 32767 –12332 1014 –14301... (all parameters
     are not shown here)

     set ipp_prof 2 aer_eq tx enable
     ```

   Note—You can use the following dsp command to change these parameters on the fly or enable/disable the equalizer during the call. However, it is recommended that the equalizer be disabled before you load new equalizer parameters.

```
dsp <tcid> aer_eq_ctrl <rx_enable/rx_disable> <tx_enable/tx_disable>
```

For example, to change the Rx path equalizer parameters when the equalizer is already enabled, do the following:

```
dsp <tcid> aer_eq_ctrl  rx_disable tx_disable

dsp <tcid> aer_eq_ctrl params rx 16535 –10332 –8775 1934 … (all parameters
are not shown here)

dsp <tcid> aer_eq_ctrl  rx_enable tx_disable
```

6 Verify the settings of the equalizer using the following show commands:

```
show ipp_prof <prof_id>
```

To verify the current state of the Tx and Rx equalizers, use the following command (see details in Table 2-7 and Table 2-9)

```
show ec_debug_stats <tcid>
```

7 Repeat the SLR and RLR measurements and note the equalizer can introduce some loss or gain that might change the overall loudness. You may need to make only minor modifications to the previous gains since the equalizer-induced loss or gain is not expected to be substantial.

End of Procedure 2-3

Figure 2-5 shows the frequency response relative to the masks before applying equalization.

Figure 2-5 Frequency Response Before Equalization

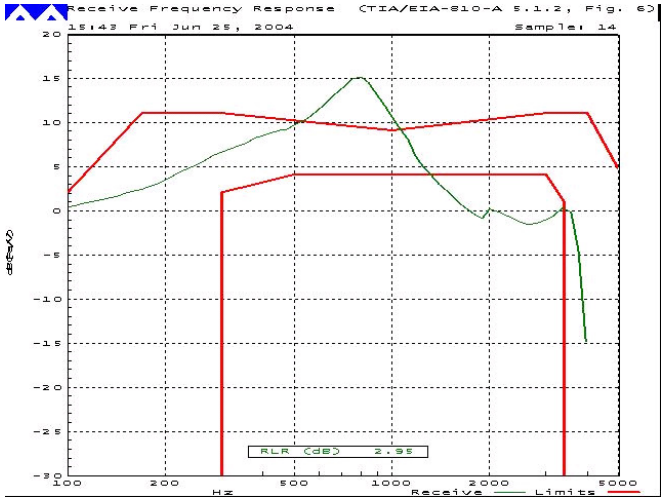
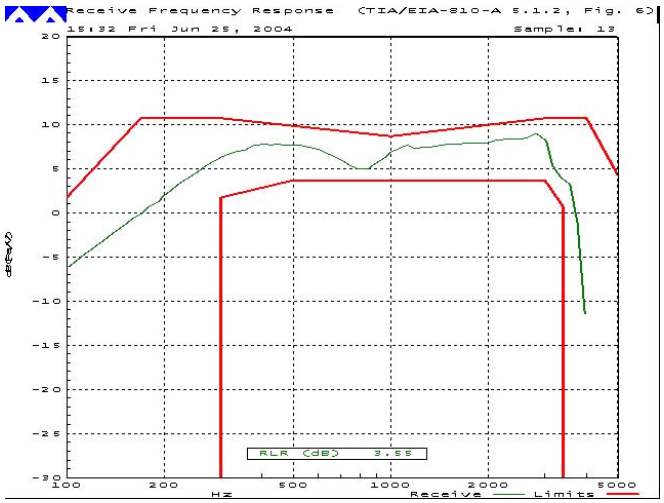


Figure 2-6 shows frequency response relative to the masks after applying equalization.

Figure 2-6 Frequency Response After Equalization



2.3.4 High Level Compensation

High Level Compensation (HLC) can be enabled in the receive path of the AER module (Figure 2-2 on page 2-5) to attenuate peaks in the input signal automatically. HLC is applied to the incoming (Rx) signal from the packet network to avoid saturation or nonlinear distortion at the speaker output. By keeping signal peaks in the Rx path below a user-specified threshold level, HLC can improve the performance of the AEC linear adaptive tail model filter by reducing echo nonlinearity and providing better full-duplex operation.

If the signal level goes below the HLC target level threshold, HLC attenuation disengages slowly to avoid sudden audible changes in the signal level, especially during short pauses in speech. The ramp_up time is configurable in 10 ms steps from 10 ms to 1000 ms per dB reduction in attenuation. Ramping up is triggered only for powers below the HLC threshold level. During this period, attenuation reduces gradually according to the ramp_up setting.

In general, selecting a small ramp_down time and large ramp_up time ensures that flat-top saturation distortion is eliminated quickly and that the required attenuation is not removed too quickly, which can cause abrupt and annoying audible level changes. The ramp-down time is not configurable and is fixed at 10 ms per dB. HLC can selectively attenuate but not amplify Rx path signals.

HLC cannot do the following to signals that it receives from the packet network:

- Restore an incoming signal that has flat-top distortion
- Amplify the level of a weak signal

NOTE—The Dynamic Range Compressor (DRC) full-band compressor and limiter are upgraded versions of the AER HLC and AER Rx SLIM, respectively. When DRC is activated, make sure that AER HLC and AER Rx SLIM are disabled or inactive. For more information, see Chapter 4 "Dynamic Range Compression (DRC)" on page 4-1.

2.4 Optimizing Parameters for Handsfree Operation

This section contains the following topics:

- "AEC Adaptive Tail Model Convergence" on page 2-30
- "Optimizing for Handsfree Full-Duplex Operation" on page 2-31
- "Optimizing for Handsfree Half-Duplex Operation" on page 2-34
- "AER Noise Guard" on page 2-36
- "AER High Level Compensation (HLC)" on page 2-36
- "Automatic Gain Control (AGC)" on page 2-38

AEC full-duplex performance is intended for circumstances when duplex quality is a high priority, but half-duplex performance is sometimes required when duplex quality is too severely limited by the hardware enclosure, including the specific relatively large gain level settings. The duplex quality of a handsfree phone refers to the degree to which the phone can avoid attenuating the send and receive paths during double talk. Disabling the AEC duplex stabilizer is appropriate for handsfree phone enclosures that can support full and partial duplex, as defined by ITU-T P.340. Roughly speaking, for P.340 "full-duplex" limits the combined attenuation during double talk in the send and receive paths to 6 dB while the "partial duplex" limit is 22 dB, and "no duplex" has no attenuation limit.

Because a full-duplex handsfree phone can be very desirable, correctly choosing whether to optimize AEC with the duplex stabilizer disabled or enabled can be confusing. It is not essential to understand this distinction; you can easily configure AEC both ways and choose an audible preference. Assume a superior phone applies 9 dB of combined attenuation during double talk. Reducing this attenuation to 6 dB yields a noticeable improvement in duplex performance, so risking additional echo leakage may be acceptable. Assume an inferior phone must apply 40 dB of combined Rx-path and Tx-path attenuation during double talk for an acceptable amount of echo reduction. For this inferior phone, after sacrificing quality with additional echo leakage after a 3 dB attenuation reduction (from 40 dB to 37 dB attenuation), most people would only notice the extra echo leakage rather than the duplex improvement. Thus after a modest relative improvement in duplex performance, it still sounds like any half-duplex phone but now has the added drawback of causing echo. Under these circumstances, you should enable the AEC duplex stabilizer. Even a full-duplex phone at nominal default speaker volume may benefit from making a transition to enabling the duplex stabilizer for higher speaker volume levels. Enabling the duplex stabilizer is also recommended when the IP Phone is transmitting to more than one far-end phone in conference mode. (Enabling noise guard is also helpful during conferencing.)

2.4.1 AEC Adaptive Tail Model Convergence

Before optimizing for full-duplex or partial-duplex performance, it is useful to test and measure AER adaptive tail model convergence (Procedure 2-4). For example, you might want to verify that an incremental enclosure modification that improves echo linearity also results in better echo cancellation.

Procedure 2-4 AEC Adaptive Tail Model Convergence Test

Step	Action
1	Connect a call between two phones.

2 Select a method (analog or digital) to send an Rx input test file. The file can contain speech, white noise, or an ITU-T CSS signal.

For analog test input (for example, generated by a PC sound card) route the signal to the far-end handset transmit path using a THAT-2 box.

For digital test input, type the following command in the far end IP-phone:

```
dsp <tcid> loop snd on
```

This will keep the call connected but stop the far end IP-phone from sending packets to the phone under test. Far-end packets will otherwise interfere with the test signal. Then, from a PC, use rtp_send software to send the test file to the phone under test.

3 After you set the characteristics of the (near-end) IP phone under test— for example, set mode to handsfree, set the volume level, etc.—issue the following command:

```
show ec_debug_stat <tcid>
```

4 Enter the following commands:

```
dsp 0 aerc rst_all nLoff txhi_bs_off noise_guard_off
```

5 Activate the application software you will use to capture transmitted packets (such as Ethereal or MS-Network Monitor), then send the Rx input file.

6 After Rx input is finished, do the following steps:

a enter the command

dsp 0 aerc off

b Resend the Rx input file.

This yields a reference level of the uncancelled echo with AER disabled.

Note—The AER digital gains (aer_tx_dg and aer_rx_dg), the AER Rx and Tx Equalizers, and AER HLC continue to operate normally even when AER is disabled.

7 Stop capturing packets. Echo cancellation is determined by the attenuation obtained when AER is enabled relative to disabled.

If you also want to eliminate the high band in AER Rx output during convergence testing, add `rxhi_bs_off` to the `dsp 0 aerc` command. During normal operation, the NLP is enabled, and the subdominant path high band is eliminated by the NLP to prevent high band echo. Because AER is half duplex in the [3400 Hz, 8000 Hz] frequency band, no convergence can be expected to occur in this region, so mixing in (uncancelled) high-band output with low band can obscure how well the adaptive tail model is cancelling low-band echo.

End of Procedure 2-4

2.4.2 Optimizing for Handsfree Full-Duplex Operation

For this mode, an important goal is to achieve the best duplex performance during double talk. See Figure 2-1 "AER Performance-Related Components" on page 2-3. The AEC should be configured with the duplex stabilizer disabled. In handsfree mode with the duplex stabilizer disabled, many changes—two of which are mentioned here—occur during double talk (relative to the actions when the duplex stabilizer is enabled.) First, the AER Tx NLP center clipper is aggressively reduced. The second change is referred to as "gain splitting". The sum of the linear attenuation in dB for the Rx NLP and Tx NLP is given by G=P[19] as reported by "show ec_debug_stat" in

"NMM Commands Relevant to AER Performance" on page 2-41. During single talk all of this attenuation is applied to the subdominant path, and during double talk the attenuation (in dB) may be equally split into both paths, as shown in Figure 2-1 on page 2-3.

For AEC configured with the duplex stabilizer disabled, the goal is to decrease the aer_nlp_clip and increase aer_nlp_clt until echo reduction is just satisfactory and no more. Testing for full-duplex performance involves taking turns generating single talk in both directions, and then having periods of simultaneous conversation. Beginning double talk immediately after a power cycle will never result in full duplex performance, because AEC will require time to converge. Test with double talk at moderate conversational levels until the tradeoff between echo reduction and duplex performance is evaluated to be optimal.

Ideally, when P[10]=log2_attn reported by ec_debug_stat reaches its typical converged value (around 10 or more, depending on the enclosure) then P[19]=G is in the 32,767 to 16,000 range, respectively indicating 0 dB to 6 dB total linear attenuation occurs. This is compatible with full-duplex operation as defined by most specifications. The configured NLP linear attenuation combined target loss, aer_nlp_clt, is given by ec_debug_stat P[20]. This parameter normalizes the actively applied linear attenuation that is adaptively defined, P[19]=G. For AEC, repeating an experiment with P[20]=aer_nlp_clt a factor of two lower yields P[19]=G a factor of two lower, or 6 dB more linear attenuation and less full-duplex performance.

The important parameters that affect echo reduction and duplex quality tradeoff are aer_nlp_clip and aer_nlp_clt.

Procedure 2-5 Optimizing a Handsfree, Full-Duplex Phone

Step – Action

1. Disable the duplex stabilizer. See "AER Control Using dsp aerc" on page 2-44 for more information about how to use this command.

   dsp <tcid> aerc on nlon update aon_mute_off duplex_stabilizer_off tx_idle_on

2. Do testing with a far end handset and near end handsfree phone, both at nominal default gains.

3. The tail length is set to 200 ms. Lower values may result in faster convergence under some circumstances but less cancellation in rooms with long reverberation times. The rx_linear_thresh is initially set to 30,000. This value effectively enables a receive path flat top saturation detector. Even when outside of the critical path AER models, receive path flat top distortion can increase nonlinearity because a discontinuity may reach the handsfree speaker when it is at maximum displacement. This 30,000 threshold is lower than 32,767 maximum positive 16-bit sample to take into account the effect of compression from far end 8-bit software vocoders. The clipper aggression parameter (aer_nlp_clip) is initially set to –10 to reduce the effect of the transmit (or send) path NLP center clipper.

   dsp <tcid> aerp 200 –10 <aer_nlp_clt> na na na na na na 30000 na na na na na

4 The aer_nlp_clt parameter is varied first. After extensive far-end speech, use the ec_debug_stat command to verify P[10]=log2_attn has attained its maximal value, and observe the current P[19]=G value, also given in "System Response to ec_debug_stat" on page 2-42. For AEC full duplex, the goal is for G to typically converge to roughly 16,000. AER applies an attenuation of -20*log10(G/2^15) to either the receive or transmit path, according to which path AER determines is subdominant.

During several trials, periodically observe the "show ec_debug_stat <tcid>" result for P[19]=G when P[10]=log2_attn is near at a relatively high level, about 10, indicating convergence is complete. If G is typically below/above 16,000 then increase/decrease aer_nlp_clt and repeat tests.

To increase G by a factor of two (after repeating identical conditions), increase aer_nlp_clt by a factor of two. In general, aer_nlp_clt is not optimized to more than 40 percent of what appears to be a good target value.

dsp <tcid> aerp na na <aer_nlp_clt> na na na na na na na na na na na

show ec_debug_stat <tcid>

5 Increase the aer_nlp_clip parameter, initially set to –10, until acoustic echo heard at the far end handset speaker is subjectively normally inaudible for normal conversations, excluding shouting.

dsp <tcid> aerp na <aer_nlp_clip> na na na na na na na na na na na na

6 If echo leaks when the far end shouts, capture packets to estimate the maximum sample threshold required for an echo problem. The <rx_linear_thresh>, initially set to 30,000 should be lowered below this threshold. This will probably not immediately solve the echo problem during far end shouting, but will help AER avoid losing convergence during the shouting; that is, perform better after shouting. Either add "hard" compression (increase aer_rx_dg/decrease rx_ag) or "soft" compression (lower HLC target level threshold) or further increase aer_nlp_clip to solve the echo break-down during shouting.

dsp <tcid> aerp na na na na na na na na <rx_linear_thresh>

7 The above procedure is repeated for the maximum speaker volume setting. If any differences in optimized parameters result, some interpolation is needed to make transitions at specific speaker volume levels.

End of Procedure 2-5

During subjective handsfree testing, the following can happen:

- With log2_attn = 6, the performance is excellent
- If log2_attn increases to 8, an echo is audible shortly thereafter
- Later log2_attn = 6 and performance is excellent

If, after testing, one can often predict echo after such a temporary log2_attn increase, this indicates the adaptive tail starts performing better momentarily and AEC responds by automatically reducing the NLP attenuation. Then the additional convergence is lost, but before this is understood, the reduced NLP attenuation causes echo. To prevent the AER NLP from continuing to turn itself off gradually when the log2_attn rises above 6 (which corresponds to 18 dB of cancellation), configure nlp_linattn_max_erle and nlp_clipper_max_erle=18 using the following command:

dsp 0 aer_cng_nlp_params na na 18 18 na na na na na na na na na na

The NLP linear attenuation and Tx NLP center clipper are not reduced when log2_attn exceeds 6.

For other useful parameters that set boundaries on the NLP attenuation, refer to "dsp <tcid> aer_cng_nlp_params <cng_rx_level [0,-90]|na> <cng_tx_level [0,-90]|na>" on page 2-53. Set additional "aerp" parameters to na (not applicable). The additional parameters are only for special circumstances as described in "AER Internal Parameters" on page 2-50.

You can do additional fine tuning to reduce intermittent echo by configuring minimum total linear attenuation or by configuring the Rx NLP and Tx NLP independently for a specific subdominant path. However, when the duplex destabilizer is disabled, normally only the total linear attenuation is restricted. This allows better full-duplex performance during gain splitting without compromising the total amount of echo attenuation. For example, to test with a total linear attenuation of 6 dB, enter the following command:

dsp 0 aer_cng_nlp_params na na na na 6 na na na na na na na na na

This parameter can be useful for reducing intermittent echo at higher speaker volumes.

NOTE—Currently, the Telogy microcode software sets this parameter to the same value for all speaker volume levels.

2.4.3 Optimizing for Handsfree Half-Duplex Operation

For phones with hardware enclosure problems that make AER full-duplex performance unattainable, configure AEC with the duplex stabilizer enabled. An NMM command for setting this bit is given in "AER Control Using dsp aerc" on page 2-44.

Enabling the duplex stabilizer is good for hardware enclosures with excessive echo and nonlinearity, where you want to save costs rather than improve this problem. Among other things, enabling the duplex stabilizer inhibits features that relax the Tx NLP attenuation during double-talk conditions. Simultaneous send and receive path transmission during double talk is not a goal as half-duplex performance is expected when the duplex stabilizer is enabled.

However, other important goals are

1. the ability to break in or interrupt
2. an adequately fast NLP switching time

Normally, the slowest switching time is for the near end participant's speech to break-in just after a burst of far-end speech is projected by the handsfree speaker. For this case, the switching time is governed by how quickly the NLP attenuation in the send path can be eliminated.

To address the problem of excessive echo, you can relinquish the goal of full duplex and allow an increase in aer_nlp_clip and a decrease in aer_nlp_clt. However, extending aer_nlp_clip to its maximum (20) and aer_nlp_clt to its minimum (0) for the most aggressive NLP attenuation possible may not optimally balance the goals of acceptable echo reduction and with these two goals listed above. The best test for setting these values is to have a far-end participant talk constantly and a near-end participant trying to break in.

Also testing "alternative counting" is recommended. For this test, the far-end and near-end participants exchange single words quickly. For example, one side speaks a random number, then the other side adds one to the number and speaks the response as quickly as possible. Note that this test yields a much faster turn-around on a conventional analog or ISDN phone. The NLP switching delays caused by the phone are added to the round trip IP-phone network delay that may exceed an ISDN phone by over 200 ms.

In summary, the change in optimizing AER parameters for half-duplex mode is primarily by enabling the duplex stabilizer and choosing best values for aer_nlp_clt and aer_nlp_clip.

Procedure 2-6 Optimizing a Handsfree Half-Duplex Phone

Step – Action

1. See "AER Control Using dsp aerc" on page 2-44 for more details about how to use this command.

   dsp <tcid> duplex_stabilizer tx_idle_off

2. Do testing with a far end handset and near end handsfree phone, both at nominal default gains.

   dsp <tcid> aerp 200 <aer_nlp_clip> <aer_nlp_clt> na na na na na na 30000 na na na na na

   A lower **aer_nlp_clt** value due to a lower converged value of P[19]=G is acceptable. A converged value of G=10,400 is acceptable for ITU-T P.340 partial duplex. However, a lower target is optimal for a half-duplex phone.

   A higher **aer_nlp_clip** value is generally optimal, because the echo reduction is not compromised by the competing goal for duplex quality during double talk. Some tradeoff may still exist, by taking into consideration the magnitude and delay required for near end speech to break in.

End of Procedure 2-6

There are additional AER parameters for optimizing performance in AEC half-duplex mode (see "NMM Commands Relevant to AER Performance" on page 2-41). These parameters may be useful in special circumstances, but are not generally recommended. The important parameters are aer_nlp_clip and aer_nlp_clt, as is the case for AEC full-duplex mode. Relative to a full-duplex phone, the main difference for AER parameter optimization is a change in the performance goals.

2.4.4 AER Noise Guard

NOTE—If ASNR is enabled, noise guard is automatically disabled.

Noise guard is intended for handsfree operation and has been tested by Texas Instruments for use in handsfree mode only. One purpose of noise guard is to enhance the half-duplex performance by letting the far end (receive path) break in more easily during high near-end (transmit path) background noise. Noise guard attenuates the near-end background noise in the transmit path when there is no near-end speech.

For handsfree operation, configure AER Noise Guard with the following NMM command.

```
dsp 0 aer_nguard_ctrl on 1000 -68 800
```

These settings may be different from the default settings of these parameters. For more information, see "AER Noise Guard" on page 2-61.

For a full-duplex phone, the drawback is that during the onset of speech the sudden withdrawal of Noise Guard attenuation yields an audible effect often associated with half-duplex performance. Technically, however, standards such as ITU-T P.340 define duplex quality by attenuation during double talk and not by artifacts during an abrupt near-end transition from attenuated background noise to background noise plus speech. The need for Noise Guard is relatively higher for handsfree mode, because the output of the handsfree microphone is typically noisier. The need for Noise Guard is increased as the duplex quality is reduced at the near end *or* the far end. Even high quality full-duplex phones may temporarily lose duplex quality after tail model divergence, or permanently lose duplex quality at high speaker volume and speech signal magnitude.

The AER Tx-path noise guard can correct problems in the following situations:

- During a conference call, many handsfree phones without Noise Guard are connected. The transmitted noise of many phones combines to increase the noise level. The noise is sufficiently dominant to cause a far-end half-duplex phone to "chop up" speech that it should be transmitting.
- During a handsfree-to-handsfree call, far-end room noise is heard for an *indefinitely* long period during mutual speech silence if the far-end participant spoke last. The one way direction of room noise transmissions is toggled by speech from the person currently hearing far-end room noise. (This "ping-pong" effect is removed if both phones have perfect full-duplex performance.)
- A handsfree phone in a very noisy environment transmits –40 dBm0 background room noise. At the other end, a half-duplex handsfree phone chops off the beginning of near-end speech, which must ramp up and persist to be dominant in relation to the noise in the opposite path.

2.4.5 AER High Level Compensation (HLC)

HLC is a feature of the Rx path that ensures that the signal received from the packet network does not cause saturation or nonlinear distortion at the speaker output. HLC cannot remove saturation distortion that already exists in the Rx path signal, but can prevent any further flat top saturation.

Because it is first in the signal path, HLC protects the signal from saturating in subsequent gain stages in the Rx path, including the DAC programmable gain. HLC only helps to reduce the signal; it does not apply any positive gain to boost a weak signal to a louder level. HLC is primarily for use with the handsfree mode, although it is available for all modes supported by Telogy software.

HLC attenuates the Rx path signal temporarily after the Rx input exceeds the configurable HLC target level threshold. HLC enables the phone to be used at higher Rx gain levels that usefully amplify weak input but Rx would otherwise (without HLC) sound bad when the Rx path input level is high.

The gains that impact HLC parameters include:

- Overall Rx Equalizer gain/loss (if any)
- AER Rx digital gain
- DAC programmable gain amplifier analog gain and digital gain (if any)

The DAC PGA gain amplifier settings should take into account any nonlinearity that the DAC may have near its maximum allowable gain. Gain tables should make sure that this gain is always restricted to its linear operating range. Any additional gain needed to meet RLR requirements by increasing volume level should be delegated to the AER Rx digital gain. This ensures that the analog part of the Rx path is linear. Therefore, HLC just needs to make sure that the Rx equalizer and AER Rx digital gain do not cause any saturation.

Procedure 2-7 gives the steps for configuring HLC.

Procedure 2-7 Configuring AER High Level Compensation (HLC)

Step – Action	
1	If Rx equalization is required, obtain the Rx path equalizer as described in the previous section. To prevent most AER Rx EQ-induced flat-top saturation distortion in the Rx signal path, follow the steps below. Examine the equalizer function, find its peak magnitude, and call it rxEq_peak_db. Convert to 0.5 dB units and eliminate negative results. rx Eq_peak = max (0, 2*rxEq_peak_db)
2	Determine the maximum level of Rx signal acceptable in the DSP after Rx equalization and AER Rx digital gains have been applied. The full scale signal is +3dBm0. To maintain a safety margin, you can limit the level to 0-2dBm0 instead. This level (max_siglevel) is selectable in 0.5dBm0 steps.
3	To configure HLC, compute the following: hlc_max_siglevel (0.5 dBm0) = max_siglevel - rxEQ_peak Given the hlc_max_siglevel, the microcode computes the HLC target level using the following formula at each volume setting: hlc_target_level = min(6, hlc_max_siglevel - AER Rx digital) **Note—**AER Rx digital level is measured in 0.5dB steps; “6” is the value that represents the 3dBm0 peak digital signal.
4	To configure other parameters for HLC:

a To enable/disable HLC for a specific phone mode, use

set ipp_prof <prof> aer_hlc <disable|enable>

b To configure hlc_max_siglevel, use

set ipp_prof <prof> aer_hlc_max_siglevel <in 0.5 dBm0 steps, MAX:6>

c To configure the HLC ramping up time constant, use

set ipp_prof <prof> aer_hlc_ramp_up_tc <in 10 ms/dB steps>

(Default value is 700msec/dB)

d To update the power calculation time constant, three settings could be used, as follows:

set ipp_prof <prof> aer_hlc_power_tc <1 (4ms), 2 (8ms), 3 (16ms)>

(Default value setting is 1)

e HLC ramp_down_tc is not configurable through the scripts, although there is a non-functional `set` and `dsp` command to do so. The DSP uses a hard-coded internal value (10ms/dB).

End of Procedure 2-7

NOTE—You can use the following dsp command to change these parameters on the fly or enable/disable HLC during the call.

```
dsp <tcid> aer_hlc_ctrl [enable|disable]
    [target_level((-96,6), default:0x8000)] [ramp_down_tc(0,20)]
    [ramp_up_tc(10,1000)] [power_tc(1,2,3)]
```

NOTE—Because this diagnostic command calls the DIM directly, you must specify the HLC target_level. That is, the value is not calculated dynamically based on the Rx digital gain.

2.4.6 Automatic Gain Control (AGC)

AGC is described in this handsfree section because the optional "adaptive" AGC mode is only recommended for—and has only been tested for—handsfree operation. However, it is a mistake to identify AGC too strongly with the adaptive mode, which is only one of its three functions.

For example, if you disable AGC only because you do not need its adaptive function, you also disable the saturation detector, which is needed for all phone modes that enable AER. AGC should be enabled for all phone modes, but the adaptive feature should be enabled only for handsfree mode.

For handsfree operation, you can configure AGC with the following NMM command:

```
dsp <tcid> agc on enable na na na na na
```

For more information about the command parameters, see "AGC" on page 2-59.

AGC does the following:

- Informs AER of all gain changes in the critical echo path that AER models. This allows AER to rescale adaptive coefficients and avoid temporarily losing convergence.
- Informs AER if there is nonlinear flat-top saturation of the ADC microphone signal. This allows AER to stop adapting temporarily and avoid losing convergence.

  The saturation threshold is configurable (see "AGC" on page 2-59). Lowering this configurable threshold may improve performance during nonlinear breakdowns.
- Adaptively defines tx_ag and aer_tx_dg to optimize the ADC microphone input dynamic range, see Figure 2-1 on page 2-3.

  In adaptive mode, AGC maintains a healthy digital dynamic range for AER that keeps the input level of the AER send path safely below saturation and is large enough to avoid ADC digital quantization errors.

  When adaptive AGC invokes a microphone ADC PGA tx_ag gain transition, no audible click is produced. In addition, because AGC keeps the sum (tx_ag+aer_tx_dg) fixed, it cannot result in the overall SLR loudness drifting out of specification. Thus, adaptive AGC gain changes will not cause background "noise pumping" or other problems often associated with other types of AGC systems.

2.5 Optimizing Handset and Headset Parameters

Highest quality duplex performance is normally an objective for these modes of operation. The AER aer_nlp_clt should be set to 32767; this effectively turns off the linear attenuation mechanism. Therefore, in Figure 2-1 on page 2-3, a G value corresponding to 0 dB results from this parameter setting. The clipper aggression parameter should typically be set at a low value compared to handsfree mode to ensure full-duplex operation at the nominal default speaker volume and above. Compromises in full duplex would be expected only for large receive signals and maximum speaker volume levels, appropriate for hearing impaired. There are specific goals for operation at maximum volume, such as adequate echo reduction when the handset or headset is placed face down on a hard table.

```
dsp <tcid> agc on disable na na na na na

dsp <tcid> aer_nguard_ctrl off na na na
```

For handset mode,

1. dsp <tcid> aerp 8 –10 32767 na na na na na na –32768 `na na na na na`

   is used to set the tail length to 8 ms, turn off the (imperfectly full duplex) NLP linear attenuation mechanism by setting aer_nlp_clt=32767, and disable the nonlinear threshold detector by setting the threshold to a special value reserved for this purpose, rx_linear_thresh=–32768. In practice there may be acoustic echo due to speaker-to-microphone coupling beyond an 8 ms delay, but a shorter tail is appropriate because changes in the echo path are common as the handset moves, and less echo reduction is normally needed for a handset.
2. The aer_nlp_clip value, initiated at –10, is increased until echo reduction is acceptable. Here one is conservative against using a high value for aer_nlp_clip, as customers expect full-duplex performance from a handset.

For headset mode,

1. dsp <tcid> aerp 60 –10 32767 na na na na na na –32768 `na na na na na`

   is used to set the tail length to 60 ms, turn off the (imperfectly full duplex) NLP linear attenuation mechanism by setting aer_nlp_clt = 32767, and disable the nonlinear threshold detector by setting the threshold to a special value reserved for this purpose, rx_linear_thresh = –32768.
2. The aer_nlp_clip value, initiated at –10 is increased until echo reduction is acceptable.

2.6 NMM Commands Relevant to AER Performance

This section contains the following topics:

- "AER"
- "NMM Commands External to AER" on page 2-57

Texas Instruments IP-Phone customers normally customize their own microprocessor code. However, it may be useful to copy parts of the pretested Telogy-provided microprocessor code.

Some microprocessor commands are low-level NMM commands (dsp <tcid> …) that go directly to the DSPMIS and have an immediate effect. Other commands are high level NMM commands (set …) that may require saving to flash using the `activate` and `commit` commands, recycling power or resetting, then pushing the phone buttons to enter a phone mode.

The information presented here is complementary to the information in the *Application Services Command Reference Manual*. This additional documentation is needed to use the Texas Instruments AER, AGC, and AER Noise Guard software described in this document.



NOTE—In command documentation, angle brackets (< >) indicate a mandatory field and square brackets ([ ]) indicate an optional field. An "OR" symbol is indicated by a vertical line (|) that separates the available options for a field.

2.6.1 AER

2.6.1.1 AER Debug Statistics

The information listed in Table 2-7 "System Response to ec_debug_stat" on page 2-42 is returned by the AER in response to the ec_debug_stat command, which has the following syntax:

```
show ec_debug_stat <tcid> [clear]
```

Using the clear option resets the debug statistic counters, which are the variables 39 ~ 50 in Table 2-7 on page 2-42. The ec_debug_stat reports the last values for these counters just before resetting them. The DSPMIS APIs, req_ec_dbgstat, and get_ec_dbgstat implement this NMM command.

The ec_debug_stat information is presented on eight lines with eight numbers per line.

Table 2-7 lists the values of the debug statistic counters.

Table 2-7 System Response to ec_debug_stat

	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6	Word 7	Word 8
P[1-8]	Control Bitfield 0 (See Table 2-8 on page 2-43)	Control Bitfield1 (See Table 2-9 on page 2-43)	private	private	private	private	private	filter_length in samples
P[9-16]	private	Best recent pre-NLP cancellation in 3 dB units log2_attn	Exponentially averaged cancellation in 3 dB units log2_residual	Current Echo Loss in 3 dB units	Maximum absolute error sample in last frame	Maximum absolute sample in predicted echo in last frame	Tx digital gain in 0.5 dB steps aer_tx_dg	Rx digital gain in 0.5 dB steps aer_rx_dg
P[17-24]	ADC PGA gain in 0.0625 dB units, includes correction tx_ag	DAC PGA gain in 0.0625 dB units, includes correction rx_ag	NLP linear attenuation $=-20^{*}\log_{10}(G/2^{15})$ G	NLP echo combined loss target for linear attn aer_nlp_clt	Center Clipper Aggression aer_nlp_clip	Clip level	NLP hangover Rx-to-Tx in 5 ms units rx_to_tx_hangover	NLP hangover Tx-to-Rx in 5 ms units tx_to_rx_hangover
P[25-32]	Noise Guard attenuation	HLC attenuation	private	private	private	private	private	private
P[33-40]	private	private	private	private	private	private	how many times reset/ startup activated SU_cnt	How many divergences (howling) detected HO_cnt
P[41-48]	How many times coherence detected CO_cnt	How many saturation events detected Sat_cnt	AER informed of how many rx_ag gain changes? rx_ag_cnt	AER informed of how many tx_ag gain changes? tx_ag_cnt	private	Number of saturation events in Tx equalizer	Number of saturation events in Rx equalizer	Tx Bandsplit Filter Saturation Events
P[49-56]	Rx Out Bandsplit Saturation Events	Rx In Bandsplit Filter Saturation Events	Far-end signal RMS power	Near-end signal RMS power	Post-echo removal signal RMS power	Near-end noise level	Echo return loss (ERL)	Echo return loss enhancement (ERLE)
P[57-64]	Tx idle counter	ASNR signal update rate	ASNR high band gain	ASNR gain limit	ASNR hangover counter	private	AEC version	AEC revision
P[65-72]	AEC branch	AEC features						

End of Table 2-7

A specific element can be referenced as P[k], where k gives the index of the element. For example, P[19]=G, P[10]=log2_attn, and so on.

2.6.1.1.1 Control Bitfield 0

Control bitfield 0 P[1] is defined as follows:

Table 2-8 Parameters and Values in the Control Bitfield 0

Control Bitfield 0								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Parameter	Hi Band Atten	Forced Tx CNG	Forced Rx CNG	Normal Tx CNG	Normal Rx CNG	Tx NLP CLP Bypass	Tx NLP Bypass	Rx NLP Bypass
Values	enabled/ disabled	enabled/ disabled	enabled/ disabled	enabled/ disabled	enabled/ disabled	enabled/ disabled	update/ freeze	enabled/ disabled
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Parameter	Tx_idle	Duplex Stabilizer	Clear All	Clear Filter	Adapt in Mute	NLP	State and Tail Adapt	AER
Values	enabled/ disabled	enabled/ disabled	enabled/ disabled	enabled/ disabled	enabled/ disabled	enabled/ disabled	update/ freeze	enabled/ disabled

2.6.1.1.2 Control Bitfield 1

Control bitfield 1 P[2] is defined as follows:

Table 2-9 Parameters and Values in the Control Bitfield 1

Control Bitfield 1																
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Parameter	Drop Rx Hi Band	Drop Tx Hi Band	Increase min tail	Test bits (private)		Partial reset	Adapt tail model	Spectral matching CNG	ASNR	Frequency domain NLP	Ramp Scale	Mute Indicate	HLC	Rx EQ	Tx EQ	Noise Guard
Values	enabled/ disabled	enabled/ disabled	0	0		0	0	0	0	0	enabled/ disabled	enabled/ disabled	enabled/ disabled	enabled/ disabled	enabled/ disabled	enabled/ disabled

2.6.1.1.3 AER Performance Statistics

The AER performance statistics listed in table elements [P51] through [P56] are also reported as part of the PIQUA voice quality statistics.

Table 2-10 AER Performance Statistics (Part 1 of 2)

Element	Statistic	Description
[P51]	**Far-end signal RMS power (Px)**	Estimated power level in 1/16 dBm of active portions of a speech signal that arrives from the packet network in the receive direction of the AEC.
[P52]	**Near-end signal RMS power (Py)**	Estimated power level in 1/16 dBm of active portions of a speech signal that arrives from the PCM network in the transmit direction of the AEC.
[P53]	**Post-echo removal signal RMS power (Pe)**	Estimated power level in 1/16 dBm of the signal after echo cancellation. In the absence of a near-end signal, this measures the power of residual error.

Table 2-10 AER Performance Statistics (Part 2 of 2)

Element	Statistic	Description
[P54]	**Near-end noise level (Ny)**	Estimated power in 1/16 dBm of background noise present within the near-end signal that arrives from the PCM network in the transmit direction of the AEC. Low levels provide the best conditions for good AEC operation, while high levels could result in harsh conditions for the AEC.
[P55]	**Echo Return Loss (ERL)**	Estimated attenuation in 1/16 dB increments of the echo level along the echo path with or without the use of an echo canceller. If an Rx voice signal enters the echo path from the packet network at a level of *x* dB, the echo coming from the tail circuit back into the echo canceller is *x*–ERL. Low ERL values indicate an increased risk of echo problems; high ERL values indicate a reduced risk of echo problems.
[P56]	**Echo Return Loss Enhancement (ERLE)**	Measured in dB, ERLE is the ratio of the send-in power and the power of the residual error signal after cancellation has been applied. In effect, ERLE measures the cancellation introduced by the adaptive filter. Low ERLE values indicate increased risk of echo problems; high ERLE values indicate reduced risk of echo problems.

2.6.1.2 AER Frequency Domain Adaptive Tail Model Coefficients

The following microprocessor command can be used to look at frequency domain adaptive tail model coefficients. (See the *Application Services Command Reference Manual* for more information.)

```
show ecpath_coeff <tcid> <start index> <number of coefficients>
```

For example, the following command with a start index of 0 and the number of coefficients set to 10 gives the first 10 coefficients.

```
show ecpath_coeff 0 0 10
```

The DIMIS commands req_ecpath_coeff and get_ecpath_coeff implement this command. The first parameter returned is the power of 2 exponent used to scale all the subsequent coefficients. The first 258 numbers represent 129 real and imaginary pairs modeling the first 20 ms of the telephone's speaker-to-microphone coupling.

By turning off adaptation and changing an analog gain in the acoustic echo path (rx_ag or tx_ag) you can test if AER is being informed of and correctly rescaling the tail model for gain changes. Also, P[43] and P[44] of the ec_debug_stat matrix give the respective number of rx_ag and tx_ag gain changes reported to AER.

2.6.1.3 AER Control Using *dsp aerc*

The following example shows the parameters and options that you can use with the AER control (aerc) command:

```
dsp <tcid> aerc [-u <usage: hs|hes|hf|gl_hs|gl_hes> <instance>]
     [on|off] [nlon|nloff] [update|freeze]
     [aon_mute_off|aon_mute] [rst_tail_off|rst_tail]
     [rst_all_off|rst_all] [duplex_stabilizer_off|duplex_stabilizer]
     [tx_idle_on|tx_idle_off]
     [rx_nlp_on|rx_nlp_off] [tx_nlp_on|tx_nlp_off]
     [tx_clip_on|tx_clip_off]
     [rx_cng_on|rx_cng_off]  [tx_cng_on|tx_cng_off]
     [rx_forced_cng_on|rx_forced_cng_off]
     [tx_forced_cng_on|tx_forced_cng_off]
     [bs_himatchlo_on|bs_himatchlo_off]
     [noise_guard_on|noise_guard_off]
     [tx_eq_on|tx_eq_off] [rx_eq_on|rx_eq_off]
     [hlc_on|hlc_off] [mute_on|mute_off]
     [nlp_ramp_scale_on|nlp_ramp_scale_off]
     [fdnlp_on|fdnlp_off] [asnr_on|asnr_off]
     [cng_adapt_on|cng_adapt_off]
     [tail_coeff_update|tail_coeff_freeze]
```

```
[partial_reset_off|partial_reset_on]
[increase_min_hvec_on|increase_min_hvec_off]
[rxhi_bs_on|rxhi_bs_off] [txhi_bs_on|txhi_bs_off]
Send AER control messages.
```

If the options are omitted, their values are not changed.

2.6.1.3.1 dsp <tcid> aerc <usage: hs|hes|hf|gl_hs|gl_hes> <instance>

The **aerc** command controls the mode of operation of the AER. When you issue this command, the phone mode is updated as specified. Table 2-11 lists the valid values for the phone mode.

Table 2-11 Phone Modes

Value	Description
hs	handset
hes	headset
hf	handsfree
gl_hs	group listen with handset
gl_hes	group listen with headset

Providing AER with information about the current mode is essential for AER to work correctly for that phone mode.

Type **show dsp_version 0** to view a description of the maximum tail length and phone mode capabilities of each AER instance. This command displays information about the number of AER instances available and the maximum duration of the tail model of each instance in ms.

A 60 ms tail is adequate for applications of AER to all phone modes except handsfree mode. A 200 ms tail is adequate for handsfree mode. The duration of the tail length can be configured to be shorter than this maximum. The phone mode capabilities for an AER instance are organized in the most significant bits of a phone mode bitfield in the following order: handset, headset, handsfree, group listen handset, and group listen headset. Because the 200 ms tail is long enough for all phone modes, its usage code is 0xF800. For a 60 ms tail, the usage code returned by **show dsp_version 0** *should* be 0xD800.

2.6.1.3.2 [on|off] [nlon|nloff] [freeze|update]

For the **off** option, AER is disabled and input is passed as output unchanged, with the exceptions of applying the aer_tx_dg gain in the send path, the aer_rx_dg gain and HLC in the receive path, and any equalization filtering. This facilitates testing what the phone would sound like while not altering the loudness but otherwise removing AER.

For the **freeze** option, the adaptive tail model and AER state become fixed and stop adapting.

For the **nloff** option, the AER NLP is disabled and no attenuation is applied in the send path or receive path. You can verify the status of AER enabled, update enabled, and NLP enabled, respectively, from the ec_debug_stat report for parameter P[1].

2.6.1.3.3 [aon_mute|aon_mute_off] [rst_tail|rst_tail_off]

For the **aon_mute** option, AER tail model adaptation continues to update during mute if **update** is enabled. Texas Instruments prefers to implement mute on/off using two micro-to-DSP messages:

- Channel_state (2058) controlling the mute bit in tele-to-packet direction
- AER_mute_indicate (19)

Some customers prefer to implement mute by setting the ADC PGA gain to mute. In this case, AER no longer has an echo path to model and, therefore, AER should be configured to not adapt during mute.

Regardless of how the phone is set up to operate during mute, AER must be informed just before mute is enabled and just after mute is disabled, using the `aer_mute_indication` command. This allows AER to operate correctly.

For example, if AER is correctly informed, it will do both of the following actions:

- disable/enable the receive path NLP during mute on/off
- turn off/on adaptation during mute on/off (when adapt in mute is configured to be false)

The adaptive tail model can be reset to zero by the **rst_tail** command.

2.6.1.3.4 [rst_all, rst_all_off] [duplex_stabilizer | duplex_stabilizer_off] [tx_idle_on | tx_idle_off]

The **rst_all** command

- resets the tail model
- resets all other AER adaptively-tracked parameters
- goes into a more aggressively converging startup state

When using one AER instance to process multiple phone modes, issue the **rst_all** command when the phone mode is initiated if the same AER instance was last used to process a different phone mode. This will avoid trying to cancel with an incorrect tail model that models a different echo path. Do not issue the **rst_all** command when a phone mode is initialized using an AER instance that was last used for the same phone mode. Avoiding this will improve the AER performance at the start of the call.

For Texas Instruments microcode, **rst_all** is issued automatically after a power cycle or phone reset, so a redundant AER control **rst_all** should not be issued at the start of the first call or it may delay the start of AER convergence by a small amount.

The **duplex_stabilizer** option enables AER to operate in a mode that is more compatible with inferior hardware enclosures. For this mode of operation, all other AER parameters will have a similar effect, but AER will greatly compromise any attempts to relax NLP attenuation during double-talk conditions. For phones with very loud nonlinear acoustic echo, the duplex stabilizer can optimize performance by eliminating acoustic echo that would be too problematic otherwise.

The **tx_idle** option forces the AER NLP to consider the Tx path as dominant after a period of mutual Rx/Tx path silence. This can be advantageous because the Rx-to-Tx hangover is typically longer.

2.6.1.3.5 [rx_nlp_on|rx_nlp_off] [tx_nlp_on|tx_nlp_off] [tx_clip_on|tx_clip_off]

The **rx_nlp_off** option can be used to turn off the Rx NLP linear attenuation and Rx CNG (comfort noise). The **tx_nlp_off** option can be used to bypass the AER Tx NLP, including the Tx NLP linear attenuation, center clipper, noise guard, and CNG. To bypass the Tx NLP center clipper only, use the command **tx_clip_off**.

2.6.1.3.6 [rx_cng_on|rx_cng_off] [tx_cng_on|tx_cng_off]

These commands can activate or bypass normal CNG (comfort noise generation) in each respective path. CNG is output at the level for which each path is currently configured. CNG prevents the phone from sounding “dead” due to attenuation of background noise by the AER NLP.

2.6.1.3.7 [rx_forced_cng_on|rx_forced_cng_off] [tx_forced_cng_on|tx_forced_cng_off]

Whereas normal CNG is added to the Rx or Tx path only when NLP attenuation is present, forced CNG always overwrites any Rx or Tx path input with CNG at the configured level. Therefore, voice cannot be passed during forced CNG. Forced CNG can be useful during mute, when a call is on hold, or for converging the echo path after a power cycle.



NOTE—Forced Tx CNG is not available when FDNLP is selected.

2.6.1.3.8 [bs_himatchlo_on|hs_himatchlo_off]

During 16-kHz wideband operation, activating this bit by using **bs_himatchlo_on** significantly reduces the attenuation of the high band of the subdominant path.

The Hi Band Atten bit (control bitfield 0, bit 15) has an effect if both of the following conditions are met:

- A wideband-compatible DSP build is being executed, so the hardware codec sampling rate is 16 kHz
- The received or transmitted packets use a vocoder with a 16-kHz sampling rate

The low band is defined from [0 to 3400 Hz] and the high band from [3400 to 8000 Hz].

For the dominant path, the high-band contribution is attenuated with NLP linear attenuation matching that of the low band, regardless of this bit.

For the subdominant path the high-band contribution is

- Removed if control bitfield 0, bit 15 = 0 (False) (Default)
- Attenuated if control bitfield 0, bit 15 = 1 (True)

The amount of attenuation in high_band is given by the low band NLP linear attenuation plus the amount of low band echo attenuation expected from subtracting the predicted echo.

2.6.1.3.9 [noise_guard_on|noise_guard_off] [tx_eq_on|tx_eq_off] [rx_eq_on|rx_eq_off] [hlc_on|hlc_off]

These options turn the following on or off: noise guard, Tx/Rx equalizers, and high level compensation. These features are described in "AER Noise Guard" on page 2-36, "Equalization" on page 2-26, and "AER High Level Compensation (HLC)" on page 2-36, respectively.

2.6.1.3.10 [mute_on|mute_off]

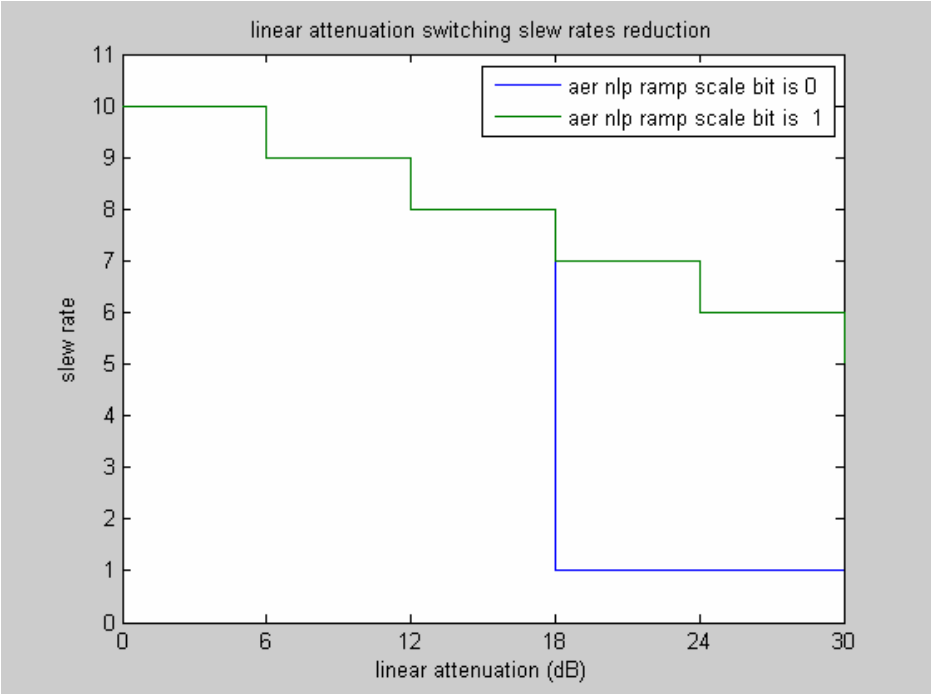
The mute_on|mute_off parameter informs AER of the status of mute. Generally, this option should not be used by the AER control command.

2.6.1.3.11 [nlp_ramp_scale_on|nlp_ramp_scale_off]

When this bit is set, all four of the values of the "k" linear attenuation switching slew rate (defined in "Four Linear Gain Switching Slew Rates" on page 2-12) are decreased by 1 for every 6 dB that the linear attenuation goes above 6 dB until the slew rate reaches a minimum of 1.

If the bit is not set, all k slew rate values are decreased the same way as when the bit is set, but are replaced by 1 when the linear attenuation goes above 18 dB resulting in the fastest gain swing transitions. Figure 2-7 shows this logic assuming the initial value of a slew rate is 10.

Figure 2-7 Linear Attenuation Switching Slew Rates Reduction (k=10)



2.6.1.3.12 [fdnlp_on|fdnlp_off] [asnr_on|asnr_off] [cng_adapt_on|cng_adapt_off]

Option fdnlp_on selects FDNLP and fdnlp_off selects TDNLP, asnr_on/off turns ASNR on or off, and cng_adapt_on/off selects spectrally-matched CNG (on) or fixed CNG (off).



NOTE—ASNR and spectrally-matched CNG are available only when FDNLP is selected.

2.6.1.3.13 [tail_coeff_update|tail_coeff_freeze]

The option tail_coeff_freeze freezes the update of tail model coefficients. In contrast to option freeze (described in "[on|off] [nlon|nloff] [freeze|update]" on page 2-45), this option freezes only the coefficients update, not AER states.

2.6.1.3.14 [partial_reset_off|partial_reset_on]

The option partial_reset_on issues a partial reset to AER, which clears AER states and delay lines, but does not reset tail model coefficients.

2.6.1.3.15 [increase_min_hvec_on|increase_min_hvec_off]

The option increase_min_hvec_on sets minimum tail length—with which converging starts—to total tail length for non-handsfree phone mode. This may be useful if the IP phone has a wireless headset or handset, which may result in long echo path delay. If the delay is longer than the minimum tail length, there will never be convergence. Setting the minimum tail length to be equal to the total tail length avoids this problem, assuming total tail length is longer than the delay, but this may cause slower initial convergence.

2.6.1.3.16 [rxhi_bs_on|rxhi_bs_off] [txhi_bs_on|txhi_bs_off]

The options rxhi_bs_off and txhi_bs_off can be used to drop the high band of the Rx and Tx path signal. Dropping high band is helpful when doing convergence testing since echo adaptation happens in low band only.

2.6.1.4 AER Control Using *dsp aert*

The **`aert`** command specifies the value of the AER Control Bitfields. These Control Bitfields are respectively given by P[1] and P[2] output resulting from the `show ec_debug_stat <tcid>` command.

```
dsp <tcid> aert <value|na> <value_mask|na> <value2|na> <value2_mask|na>
```



NOTE—Setting a mask value to "na" yields a mask of 0x0000

This command maps values directly into AER Control Bitfield 0 and AER Control Bitfield 1. A pair of values, value and value_mask, must be specified to change each bitfield, where value_mask tells which bits are to be changed and value gives the values of the bits.

For example, dsp 0 aert 0x3 0x7 0xf 0xe sets the following options:

- AER enabled (bit 0 of bitfield 0)
- AER update enabled (bit 1 bitfield 0)

- NLP disabled (bit 2 of bitfield 0)
- Noise guard disabled (bit 0 of bitfield 1)
- Tx equalizer enabled (bit 1 bitfield 1)
- Rx equalizer enabled (bit 2 bitfield 1)
- HLC enabled (bit 3 bitfield 1)

All other bits are not affected by this command.

2.6.1.5 AER Internal Parameters

Figure 2-1 "AER Performance-Related Components" on page 2-3 enables you to see where each AER, AER Noise Guard, AER EQ, AER HLC, and AGC parameter has its effect within the IP phone.

The following on-the-fly dsp commands set all parameters for internal tuning of the AER.

After determining an optimal parameter, the microcode can configure AER to this value permanently using a `set` command. Texas Instruments provides an exemplary script that uses the `set` command. Entering `dsp` or `set` alone on the command line displays online documentation that gives the syntax of available commands.

```
dsp <tcid> aerp <tail(msec)> <aer_nlp_clip(-10,20)>
   <aer_nlp_clt(0,32767)> <aer_tx_dg (0,60)> <aer_rx_dg>
   <rx_noise_min> <tx_noise_min> <rx_to_tx_hangover>
   <tx_to_rx_hangover> <rx_linear_threshold>
   <tx_slim_mode [0, 5]> <rx_slim_mode [0, 5]>
  <coh_hangover [1, 32767]> <coh_ratio_thresh [0, 65535]> <coh_cnt_thresh [0, 100]>
   Update AER parameters, Use "na" to ignore any param.

dsp <tcid> aer_fdnlp_config <txfdnlp_msec_delay|na> <txfdnlp_bin_lo1|na>
     <txfdnlp_bin_lo2|na> <txfdnlp_bin_hi1|na>
     <txfdnlp_bin_hi2|na> <txfdnlp_cng_max|na>

dsp <tcid> aer_asnr_config <nr_fbin1_lim|na> <nr_fbin2_lim|na>
     <nr_fband1_max_atten|na> <nr_fband2_max_atten|na>
     <nr_fband3_max_atten|na> <nr_sig_upd_rate_max|na>
     <nr_sig_upd_rate_min|na> <nr_noise_thresh|na>

dsp <tcid> aerp <tail(msec)|na> <aer_nlp_clip|na>
     <aer_nlp_clt|na> <aer_tx_dg|na> <aer_rx_dg|na>
     <rx_noise_min|na> <tx_noise_min|na> <rx_to_tx_hangover|na>
     <tx_to_rx_hangover|na> <rx_linear_threshold|na> <tx_slim_mode|na>
     <rx_slim_mode|na>

dsp <tcid> aer_cng_nlp_params <cng_rx_level [0,-90]|na>
     <cng_tx_level [0,-90]|na> <nlp_linattn_max_erle [0,45]|na>
     <nlp_clipper_max_erle [0,45]|na> <nlp_total_linattn_min [0,70|na>
     <nlp_rx_linattn_min [0,70]|na> <nlp_rx_linattn_max [0,70]|na>
     <nlp_tx_linattn_min [0,70]|na> <nlp_tx_linattn_max [0,70]|na>
     <gain_split_tc [1, 10]> <nlp_tx_in_tc [1, 10]> <nlp_tx_out_tc [1, 10]>
     <nlp_rx_in_tc [1, 10]> <nlp_rx_out_tc [1, 10]>

dsp [tcid] aer_eq_ctrl> <rx_enable|rx_disable> <tx_enable|tx_disable>

dsp [tcid] aer_eq_ctrl params <rx|tx> <paramf1> <param2> .. <param42>

dsp <tcid> aer_hlc_ctrl [enable|disable]
     [target_level|na] [ramp_down_tc|na]
     [ramp_up_tc|na] [power_tc|na]

dsp <tcid> aer_gain_chg_params [tx_ag_chg_synch_delay]
     [rx_ag_chg_synch_delay] [tx_ag_chg_settling_period]

dsp [tcid] aer_nguard_ctrl <on|off> <hangover_period|na>
     <send_noise_level|na> <ramping_in_period|na>
```

```
dsp [tcid] agc <on|off> <adaptive:0=disabled|1=enabled> <sat_thresh|na>
     <sat_hangover|na>
```

The following sections describe the actions in the command line by line.

2.6.1.5.1 dsp <tcid> aerp <tail(msec)|na> <aer_nlp_clip|na>

Table 2-12 lists the recommended tail length for each phone mode.

Table 2-12 Recommended Tail Length for Each Phone Mode

Phone Mode	Tail Length
handset	8 ms
handsfree	200 ms
headset	20, 40, or 60 ms

If similar far-end stimulation results in higher “show ec_debug_stat <tcid>” P[10]=log2_attn when the tail length duration is longer, increase the tail length. In general, adjusting the tail length to optimize performance for every gain level is not advised.

The center clipper level aggression is defined in 3 dB steps. As AGC adaptively changes tx_ag, the sum tx_ag+aer_tx_dg remains constant so that the effect of the center clipper remains constant during adaptive AGC operation. If the experiment is repeated and the only thing that changes is an increase by one of the center clipper aggression, the center clipper will “bite” 3 dB more out of any echo.

2.6.1.5.2 <aer_nlp_clt|na> <aer_tx_dg|na> <aer_rx_dg|na>

The **`aer_nlp_clt`** parameter refers to the NLP linear attenuation combined loss target. The combined loss target in dB is -20*log10(aer_nlp_clt/2^15).

If an experiment is repeated with a value of **`aer_nlp_clt`** that is a factor of (1/sqrt(2)) smaller, the linear attenuation will be greater and reduce the echo by an additional 3 dB. As AER converges, the ERLE increases; therefore, the amount of linear attenuation applied to maintain a fixed combined loss (ERL+ERLE) is reduced. When AER is well-converged on a good phone enclosure with a relatively low ERL, the linear attenuation will approach 0 dB; that is, the NLP linear attenuation automatically turns itself off. Ideally, when P[10]=log2_attn reported by ec_debug_stat reaches its maximum converged value—around 10 or more, depending on the enclosure—P[19]=G is in the 16,000 to 32,000 range, indicating 6 dB or less total linear attenuation occurs. This is compatible with full-duplex operation as defined by most specifications. The fixed t=0 NLP linear attenuation combined target loss, aer_nlp_clt, is given by ec_debug_stat P[20]=aer_nlp_clt, and the actively applied linear attenuation that this fixed parameter governs, P[19]. For AEC, repeating an experiment with P[20]=aer_nlp_clt a factor of two lower yields P[19] a factor of two lower, or more linear attenuation and less full-duplex performance.

The aer_tx_dg and aer_rx_dg gains are shown in Figure 2-2 on page 2-5. The active values of aer_tx_dg and aer_rx_dg and all the other path gains (except the AER NLP linear attenuation) can be verified simultaneously using the **`show gains <tcid>`** command.

2.6.1.5.3 <rx_noise_min|na> <tx_noise_min|na> <rx_to_tx_hangover|na> <tx_to_rx_hangover|na>



CAUTION—The four parameters in this command line are normally used with their default values. However, some customers have solved some phone problems and improved performance by using non-default values. To make best use of your testing efforts, make sure you understand these parameters completely before testing with non-default values.

The **rx_noise_min** and **tx_noise_min** parameters can be increased from their defaults (5380) to counter the problems caused by noise-induced false break-ins of the NLP. Half-duplex handsfree operation will preferably occur at times, even for full-duplex phones that are temporarily not well-converged. For adequate echo reduction, it is sometimes desirable for speech in one path to cause high NLP attenuation in the other path. A false break-in occurs when the NLP attenuation chops into the desired speech signal because the AER NLP determines that the speech is subdominant to noise in the opposite path.

You can increase the tolerance to break-ins of unwanted noise by increasing a minimum level that noise has to be above before it will be recognized as speech. For example, increasing **rx_noise_min** for AER in handsfree mode decreases false break-ins due to handset nasal exhale noise coming from the far end. The drawback is that soft-spoken speech from the far end may not break in immediately. Similarly, after increasing **tx_noise_min**, handsfree mode may better avoid applying the Rx NLP linear attenuation to far-end speech because of intermittent room noise. In this case, the drawback is that a soft-spoken person far from the handsfree unit in a large conference room will be less likely to break in.

The **rx_to_tx_hangover** and **tx_to_rx_hangover** parameters govern how fast NLP attenuation switching can occur after a change in dominant path occurs in the Rx/Tx paths. These values are specified in units of 5 milliseconds.

2.6.1.5.4 <rx_linear_threshold|na>

The **rx_linear_thresh** parameter defines a receive path nonlinearity threshold. It warns the AER that nonlinear breakdown may occur when the digital signal level driving the handsfree speaker exceeds a specified fixed level. This parameter can be effective when a handsfree phone is otherwise working well, but leaks echo when the far end shouts.

2.6.1.5.5 dsp <tcid> aer_cng_nlp_params <cng_rx_level [0,-90]|na> <cng_tx_level [0,-90]|na>

The **cng_rx_level** parameter sets the AER Rx NLP CNG (Comfort Noise Generator) level, given in dBm0 units. The **cng_tx_level** parameter sets the AER Tx NLP CNG level, for fixed CNG, and minimum level for adaptive CNG.

All the optimal parameters for this command that constrain NLP attenuation may need to be made volume dependent. For example, a minimum amount of attenuation is often desirable at maximum volume (to prevent echo), but not at nominal volume for handsfree operation (for better full-duplex quality). The current microprocessor code does not support making these parameters volume dependent by using the **set** command.

2.6.1.5.6 <nlp_linattn_max_erle [0,45]|na> <nlp_clipper_max_erle [0,45]|na>

The AER response to **show ec_debug_stat** includes a report of P[10]. Multiplying this value by 3dB yields ERLE_canc, the amount of echo cancellation obtained by subtracting the predicted echo. As the value of 3*P[10] *increases*, the NLP linear and (Tx) clipper attenuation *decreases*. You can limit the range over which these adaptive NLP attenuation changes take place by using the **nlp_linattn_max_erle** and **nlp_clipper_max_erle** parameters. For example, if you set nlp_clipper_max_erle=21, the AER NLP linear attenuation normally decreases as P[10] changes from 0 up to 7. But for P[10]≥7, no further reduction in either the linear attenuation (as indicated by $-20*\log_{10}P[19]$) or the clipper rail (P[22]) will take place.

2.6.1.5.7 <nlp_total_linattn_min [0,70|na>

You may determine during testing, particularly at high volume levels in handsfree mode, that a minimum amount of linear attenuation is needed at all times to adequately ensure that echo is not transmitted. To configure the phone to restrict the total linear attenuation to some minimum, the **nlp_linattn_min** parameter is used. If, for example, nlp_linattn_min=6, during double talk (with the duplex_stabilizer disabled) gain splitting may occur, resulting in a minimum of 3 dB linear attenuation in both the Rx and Tx NLP.

2.6.1.5.8 <nlp_rx_linattn_min [0,70]|na> <nlp_rx_linattn_max [0,70]|na>

These parameters constrain the minimum and maximum Rx NLP linear attenuation.

2.6.1.5.9 <nlp_tx_linattn_min [0,70]|na> <nlp_tx_linattn_max [0,70]|na>

These parameters constrain the minimum and maximum Tx NLP linear attenuation.

2.6.1.5.10 dsp <tcid> aer_eq_ctrl [<params>]

The **aer_eq_ctrl** command enables or disables the AER equalizer on the receive path or the transmit path or both. It also configures the AER equalizer parameters. If the 'params' keyword is given, all the <rx | tx> <param1>…<param42> coefficients that define the equalization filter are set.

2.6.1.5.11 dsp <tcid> aer_hlc_ctrl

The **aer_hlc_ctrl** command sets the parameters for high level compensation associated with the AER HLC. You can set the target level threshold and set time constants for ramp up, power, and estimate.

2.6.1.5.12 dsp <tcid> aer_gain_chg_params

The **aer_gain_chg_params** command sets the parameters for synchronizing gain change delays in the receive and transmit paths, and the settling period of the ADC PGA after a gain change in the transmit path.

Procedure 2-8 and the analysis that follows describe a method for obtaining an adaptive AGC/AER gain change delay and interpolation count. This method is recommended to determine the two gain-change-related AGC/AER parameters. However, if subjective performance is *acceptable* and you do not hear frequent clicks or pops due to AGC action, it may not be necessary to determine these parameters too accurately. These parameters would depend on the implementation of the gain change APIs and the characteristics of the hardware codec and the Tx equalization filter.

NOTE—The settings of these two parameters are not configurable through AER external APIs before IP Phone Release 11.0.

This process estimates the correct values of the parameters by trial and error rather than by exact measurement. Start with a gain change delay of 0 samples and an interpolation count of 0.

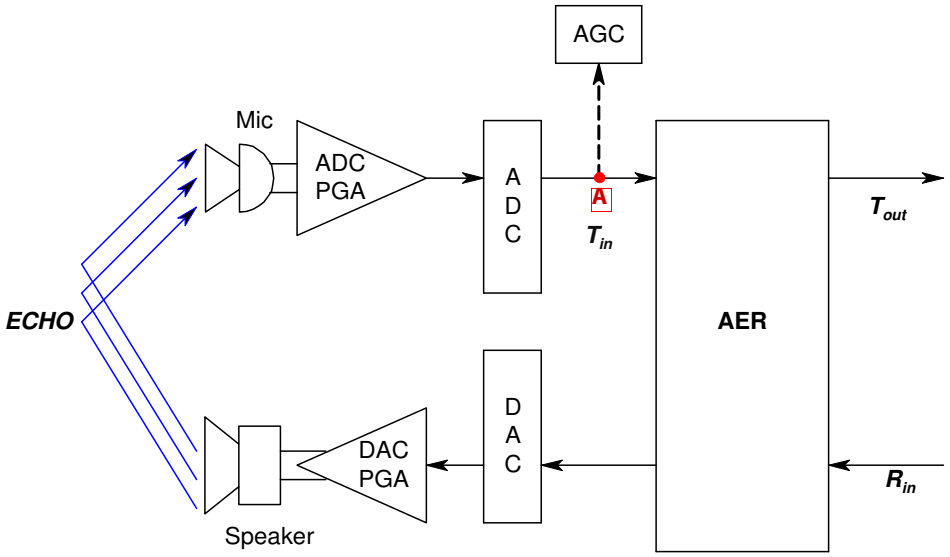
Procedure 2-8 Inducing a Gain Change by AGC

Step	Action
1	Disable the AGC and AER.
2	Set the speaker to its maximum volume setting. This will cause a louder echo.
3	Send a 0dBm0 300-Hz tone to the AER receive path at R_{in} (see Figure 2-8 on page 2-55). The tone will be played out at the handsfree speaker.
4	Set the AER Tx path digital gain to 0.
5	Set the AGC saturation detection threshold to 16000.
6	Adjust the ADC PGA gain so that the signal at point A (in Figure 2-8 on page 2-55) exceeds a linear value of 16000, but is below 32767.
7	Clear the AER debug stats using the reset option for the aerGetPerformance() API: `show ec_debug_stat <tcid> clear` This resets the stats for the number of saturations and gain changes introduced in the Tx path to 0.
8	Enable the AER, but disable update and turn off NLP.
9	Start recording the signal at T_{out} using realtime PCM tracing. If tracing is not available, you can use network captures; however, captures are not recommended because voice codecs may affect the signal.
10	Enable the AGC and make sure that the adaptive mode is enabled. This should cause the AGC to trigger saturation detection on echo and adaptive measures for gain control. In such a situation, AGC will try to decrease the ADC PGA gain by 1.5dB and increase the AER Tx digital gain by 1.5dB.

11 Using the AER debug stats, verify that a Tx gain change occurred through the use of the `aerGetPerformance()` API. See the User's manual for a description of the debug stats.

End of Procedure 2-8

Figure 2-8 Calibration Method for Gain Change Parameters



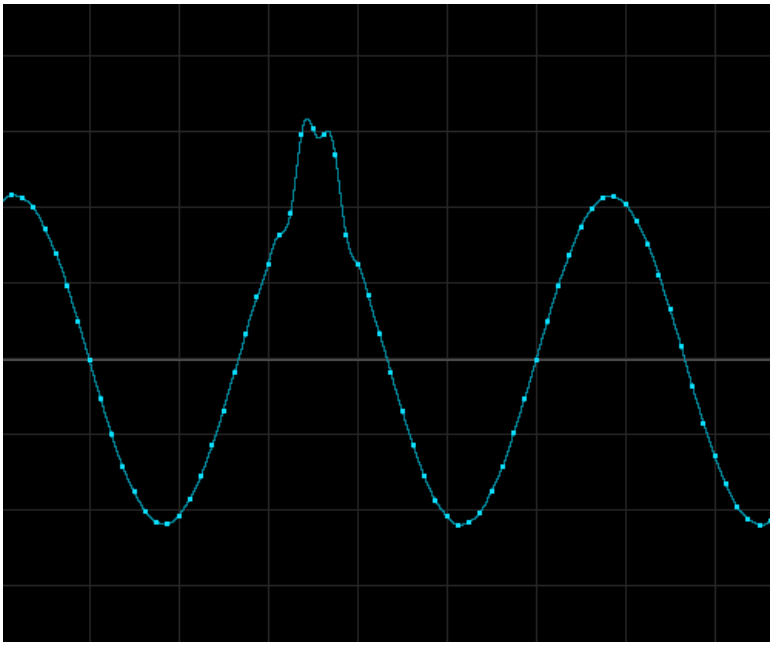
After you verify the occurrence of a gain-change event, look at the trace of T_{out} that was created in step 9. To determine where the gain change happened, listen for any artifacts, using a tool like Adobe Audition (formerly Cool Edit Pro). You can use a low-frequency signal like 300 Hz for this test to help locate clicks and pops more easily. After you determine the location of the artifact, you can make some decisions about the value of the gain change parameters by looking at the signal waveform more closely near the artifact.

Clicks and pops may appear if the ADC PGA gain and AER Tx digital gain are not well synchronized.

NOTE—Instead of sending the tone to the receive path, you may also play out the tone into the microphone input of the phone in test.

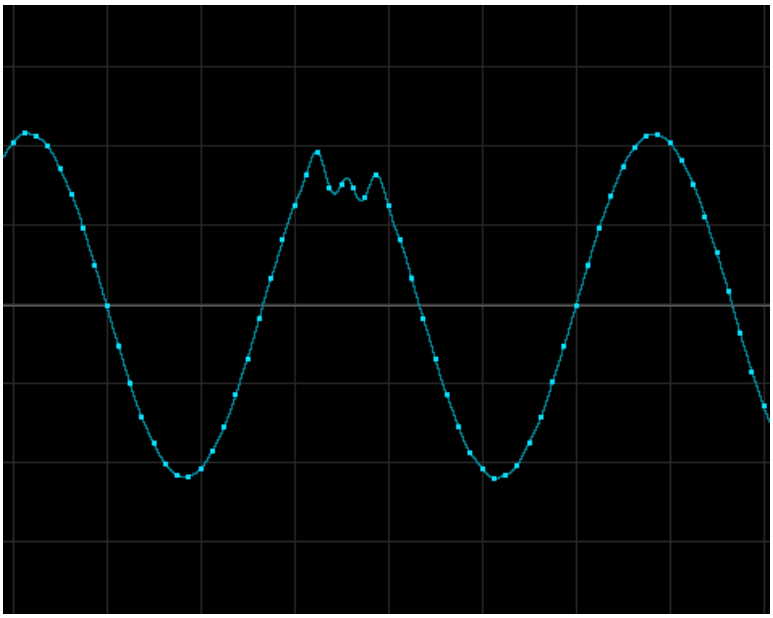
If the delay is set too short, the increased AER Tx digital gain appears first and the reduced analog gain change takes affect later.

Figure 2-9 Gain Change Delay Too Short (AER Tx digital changes first)



If the delay is set too long, the analog gain change appears first.

Figure 2-10 Gain Change Delay Too Long (ADC PGA changes first)



In both cases, some transients occur at the point that the analog gain changes.

The sample delay should be adjusted so that ADC PGA gain change and AER Tx digital gain change happen at the same time.The interpolation count is determined by measuring how many samples get distorted due to transients introduced by the settling time of the ADC PGA gain.

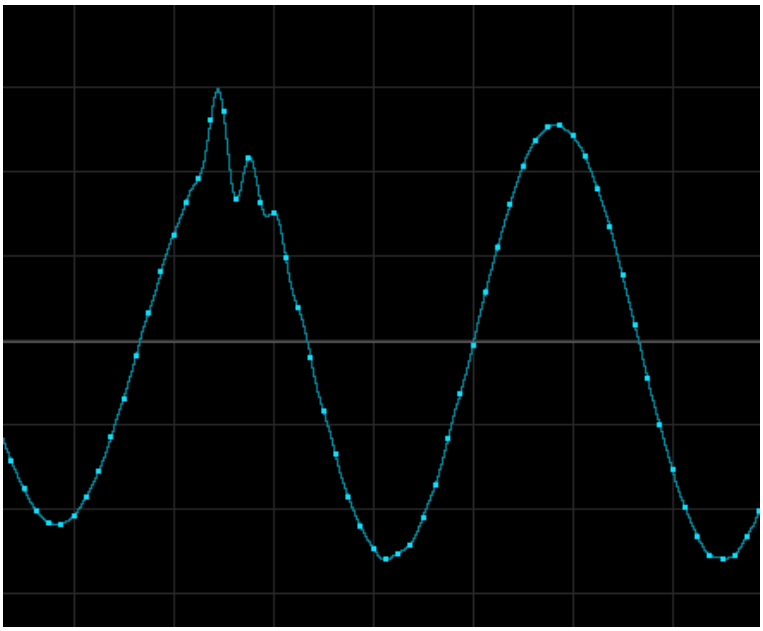
Figure 2-9 and Figure 2-10 show this concept. In these figures, the AGC introduced a +1.5 dB change in the AER Tx digital gain and a –1.5 dB change in the ADC PGA gain.

NOTE—The resolution for changes in AGC-induced gain is 1.5 dB. For each induced gain change, the value of the ADC PGA gain changes by ±1.5 dB and the value of the AER digital gain changes by the same amount in the opposite direction.

Figure 2-11 shows a hypothetical situation of ADC PGA gain increased by 1.5 dB (assuming that AGC adaptive is turned off and there were no AER Tx digital gain changes). The figure shows that about five samples are distorted due to the PGA gain settling time. This may be done before determining the gain change delay to make sure that the transients do not corrupt the estimate of the delay. Even if no ringing occurs, a value of 2 is recommended for the interpolation count.

NOTE—To determine the interpolation count, you can use any ADC PGA gain change recording. The gain change does not need to be AGC-induced.

Figure 2-11 Distorted Samples During a 1.5 dB Gain ADC PGA Change



2.6.2 NMM Commands External to AER

- "Gains and Signal Levels"
- "AER HLC" on page 2-59
- "AGC" on page 2-59
- "AER Noise Guard" on page 2-61
- "Equalization" on page 2-26

2.6.2.1 Gains and Signal Levels

The following command can set all the relevant gains that are external to AER:

```
dsp <tcid> gains <na|tx_dg (-14 to 14 dB)> <na|rx_dg (-14 to14 dB)>
<na|tx ag(in 0.5 dB)>

<na|rx_ag (in 0.5 dB)> <na|rx_sec_ag (in 0.5 dB)> <na|lcd_g (0-15)>
<na|sidetone gains (in 0.5 dB)>
```

An active call is assigned a DSP channel number <tcid> and a hardware codec channel. For the Texas Instruments RDB or SDB platform, pressing the speaker volume button changes the DAC PGA gain (rx_ag) on the primary hardware codec channel. Thus rx_ag_sec indicates the other "secondary" DAC PGA gain. For example, in group listen mode with handset, rx_ag is the handsfree speaker DAC PGA gain, and rx_ag_sec is the handset speaker DAC PGA gain.

To change the handset speaker volume in group listen mode with handset on the Texas Instruments RDB or SDB platform, toggle into handset mode, push the volume button, then toggle back. You can use a different button/microcode protocol and set rx_ag_sec directly to not require toggling phone modes.

The following NMM command reports current gain values:

```
show gains <tcid>
```

For handsfree mode, the sidetone gain should be muted, as indicated by the value –32768 from **`show gains <tcid>`**. The sidetone gain reported by `show gains <tcid>` is exceptional because it is the configured *target* value—not the current value—of the *sum* of gains in the sidetone path.

It is **strongly recommended** that microcode software in an IP-phone product use other Texas Instruments NMM gain commands and not AIC control commands `dsp 0 acregr` and `dsp 0 acregw`. Because these primitive commands are not reported to the AIC, AGC, and AER, they can cause problems. For example, the "poke" command can result in bad hardware codec gains that are automatically avoided when the `dsp <tcid> gains` API is used instead. The case of an ADC PGA gain with a net negative digital gain component (see Figure 2-1 on page 2-3) is covered in the "sat_thresh" parameter description of "AGC" on page 2-59. A worse case occurs when a "poke" is used to set a DAC PGA gain with a net positive digital gain component (see Figure 2-1 on page 2-3). Using a value of rx_ag>0 places a positive digital gain in the critical path that AER models, enabling nonlinear "flat-top" saturation and a breakdown in AER-predicted echo cancellation (see Figure 2-1 on page 2-3). This problem is unnecessary, as the positive digital gain should properly be implemented by aer_rx_dg so nonlinear saturation is kept outside the critical path that AER models linearly.

To find the digital signal levels for the transmit path, use the following microprocessor command:

```
show tlevels <tcid>
```

Unfortunately, the syntax for the output of this command uses labels that *reverse* the normal rx/tx convention. Here "rx_level" refers to the power in 0.1 dBm0 units of the transmit (=send) path signal just after the microphone ADC, which will include the effect of the ADC PGA tx_ag gain but exclude the aer_tx_dg, Tx equalizer, and tx_dg digital gains. Here "tx_level" refers to the power in dBm0 of the receive path signal just before the DAC, which will *exclude* the effect of the DAC PGA rx_ag gain and include the aer_rx_dg, Rx equalizer, and rx_dg digital gains.

2.6.2.2 AER HLC

The NMM has built-in profiles for different phone modes, such as handsfree, handset, and headset. You can modify the HLC values with the following functions:

```
set ipp_prof <prof> aer_hlc <disable|enable>
set ipp_prof <prof> aer_hlc_ramp_down_tc (currently fixed at 10 ms/dB steps)
set ipp_prof <prof> aer_hlc_ramp_up_tc <value in 10 ms/dB steps>
set ipp_prof <prof> aer_hlc_power_tc <1 (4ms), 2 (8ms), 3 (16ms)>
set ipp_prof <prof> aer_hlc_max_siglevel <value in 0.5 dBm0 steps, MAX: 6>
```



NOTE—The IPP profile must be set before bringing up a call.

For the Texas Instruments IP Phone microcode, the HLC threshold is calculated dynamically based on the current Rx digital gain and max_siglevel. Each time you press the up or down volume key, new values are read from the gain table and the calculated HLC threshold is sent to the DSP.

You can also set the HLC threshold using the set ipp_gains command:

```
set ipp_gains <prof> aer_ hlc_target_level <value in 0.5 dB steps from -96 to +6>
```

2.6.2.3 AGC

The "**show agc_debug_stat <tcid>**" NMM command reports the data shown in Table 2-13.

Table 2-13 System Response to agc_debug_stat

	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6	Word 7	Word 8
P[1 - 8]	Control bitfield	private	ADC saturation level sat_thresh	ADC saturation hangover sat_hangover	private	private	private	private
P[9-16]	private	private	private	private	private	private	private	private
P[17-23]	private	private	AGC version	AGC revision	AGC branch	AGC feature	private	

Control Bitfield 0								
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Parameter	private							
Values								
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Parameter	private						Adaptive control of tx_ag & aer_tx_dg enabled?	AGC enabled? TRUE=1 FALSE=0 Enable
Values							enabled/ disabled	enabled/ disabled

See the agc_control documentation in the *Telogy Software DSP Microprocessor Interface Specification* with the additional information below.

```
dsp <tcid> agc <on|off> <adaptive:0=disabled|1=enabled> <sat_thresh|na>
    <sat_hangover|na> <max_micgain|na> <min_micgain|na> <max_micgain_swing|na>
        [clear]
    Note: Use [clear] option to clear everything and restart
```

It is a mistake to disable AGC ("off") as a means of disabling the AGC adaptive tx_ag/aer_tx_dg function because this is only one of many functions that AGC carries out. If enabled, AGC detects ADC saturation and informs AER not to adapt when the microphone signal is too high, in a regime considered to be nonlinear. During microphone ADC flat-top saturation AER should not adapt its linear tail model of the echo path. AGC also intercepts both ADC and DAC PGA gain change messages, so it can report these gain changes to AER. This allows AER to rescale its adaptive tail model coefficients during gain changes and, therefore, stay converged.

The adaptive function of AGC to define tx_ag and aer_tx_dg is an important feature of the software. The sum of tx_ag + aer_tx_dg remains constant, so the overall SLR (send path loudness rating) is not altered. This type of adaptive AGC never poses a risk for pumping up near-end background noise during mutual silence or letting the overall SLR loudness drift into violation of a standardized specification.

The objective of AGC is to optimize the send path digital dynamic range to enhance AER tail model convergence. If the ADC signal is flat-top-saturating too frequently, AGC lowers tx_ag automatically. This removes a source of nonlinearity in the echo path and allows AER to better converge. If the ADC signal from the microphone is too small, this also limits the resolution of the signal used to adapt the AER tail model. The ADC quantization noise is itself a source of nonlinearity. If conditions of low ADC signal level persist after a longer period of time during which AER has perceived a sufficient amount of speech in both the send and receive directions, the AGC (configured to adaptive mode) increases tx_ag, the microphone gain. These automated transitions occur without inducing discontinuities or audible clicks in the send path signal. This adaptation makes the IP-phone more robust for converging AER in a wider range of signal levels. Most importantly, it reduces the burden on the manufacturer to do a good job choosing an optimal tx_ag value for every level of handsfree speaker volume on a phone with negative ERL values. When the ERL is negative, every increase in the speaker volume may need a compensating decrease in tx_ag so the acoustic echo is not typically saturated. Choosing optimal rx_ag-dependent values for tx_ag may still be advantageous after power cycle or reboot; however, it is no longer essential for getting AER to converge.

The sat_thresh variable is intended only for customers who disregard Texas Instruments recommendations and either use a hardware codec that is not supported by Telogy software, or bypass recommended Texas Instruments gain control software and use the "poke" NMM command, dsp <tcid> acregw. Although the "ag" in tx_ag refers to "analog" the hardware codec may actually implement part of the tx_ag gain digitally.

The ADC saturation hangover (sat_hangover) determines the amount of time after an ADC flat-top-saturation event occurs, before AGC reports to AER that the ADC saturation status is over. The default value is 20 ms: 10 ms to ensure that the saturated sample exits the active AER error signal and 10 ms to ensure that the ADC digital filters are cleared. The hangover may be extended if it is felt that the fact that the signal saturated at all indicates the handsfree speaker is in a nonlinear state. However, Texas Instruments recommends reducing the rx_linear_thresh only for this purpose and manipulating sat_thresh and sat_hangover only as a last resort. The configurable saturation hangover time should be increased if the phone tester observes that after the internal amplifier of the microphone saturates it sometimes and requires a much longer period to recover to a linear state.

The maximum and minimum microphone gains (max_micgain and min_micgain) set the range of microphone gains that adaptive AGC can adjust, i.e. AGC will not request gain that is out of this range.

The maximum gain swing (max_micgain_swing) sets a limit on how much the adaptive AGC can change the microphone gain from the reference gain, which is the microphone gain configured by the user.

2.6.2.4 AER Noise Guard

The AER Noise Guard control command follows:

```
dsp [tcid] aer_nguard_ctrl <on|off> <hangover_period|na>
<send_noise_level|na> <ramping_in_period|na>
```

For normal Noise Guard functionality, if AER detects speech, the transmit signal is unaltered. If AER detects NOISE, the transmit signal is attenuated linearly so that output power approximates the configurable send_noise_level parameter of Noise Guard.

Assume the AER Noise Guard is enabled, send path speech ends, and only background noise is being transmitted that is greater than send_noise_level. After a hangover_period, Noise Guard begins to reduce the transmitted signal. This reduction continues over the ramping-in period until the output signal power reaches send_noise_level.

Therefore, after speech the Noise Guard output reaches send_noise_level over a period of time given by the sum of the hangover period and the ramping-in period. No random noise is introduced. AER Noise Guard only determines how much attenuation to apply in the send path so that the send path output level approximately matches the configured send_noise_level.

Table 2-14 lists the AER Noise Guard parameters.

Table 2-14 AER Noise Guard Parameters

Parameter	Description
Hangover _Period	• Specified in 1 ms units from 0-2500 ms • Default: 1000 ms
send_noise_level	• Specified in dB units from –35 dBm0 to –70 dBm0 • Default: –70 dBm0.
Ramping-In Period	• Specified in 1 ms units from 0-2000 ms • Default: 800 ms
End of Table 2-14	

The following command sets all AER Noise Guard parameters to their default values:

```
dsp 0 aer_nguard_ctrl on na na na
```

NOTE—Many software codecs save network bandwidth when VAD is enabled by not transmitting packets when the VAD determines that voice is not present. In some circumstances, this may have undesirable effects, such as chunks of speech that fail to be transmitted or audible variations in the background noise. If these problems are falsely attributed to the AER NLP, you might direct unnecessary efforts to further tune AER instead of the VAD. To help determine whether it is AER or the VAD that needs tuning, disable the VAD temporarily (or permanently) for handsfree operation, as follows:

```
dsp [tcid] vad [on|off]
```

2.7 Default Parameter Settings

This section contains the following topics:

- "Default Settings of Configurable Flags for AER/ AER Noise Guard/ AER EQ/ AER HLC/ AGC"
- "AER Optimized Parameter Settings for 8 kHz and 16 kHz Sampling Rates" on page 2-64

2.7.1 Default Settings of Configurable Flags for AER/ AER Noise Guard/ AER EQ/ AER HLC/ AGC

Table 2-15 lists the default settings of AER/ AER Noise Guard/ AER EQ/ AER HLC /AGC configurable flags for the Texas Instruments RDB IP Phone, enclosure, and transducers.

For more information, see Figure 2-2 on page 2-5.

Table 2-15 Default Flags: AER/AER Noise Guard/ AER EQ/ AER HLC/ AGC Parameters

	Parameter (options)	Default
1	AER (enabled /disabled)	enabled
2	Duplex stabilizer (enabled / disabled)	disabled
3	Phone mode (Handsfree /Handset/Headset/Group Listen)	All phone modes are supported; settings vary according to the phone mode in use.
4	NLP (enabled /disabled)	enabled
5	Filter updates (enabled /disabled)	enabled
6	Filter updates during mute (enabled /disabled)	enabled
7	Tx NLP relaxation during mutual silence (enabled /disabled)	enabled
8	AER Noise Guard (enable/disable)	disabled
9	Tx Path Equalizer (enable/disable)	depends on phone mode
10	Rx Path Equalizer (enable/disable)	depends on phone mode
11	AER HLC (enable/disable)	enabled
12	AGC (enable/disable)	enabled
13	AGC adaptive mode (enable/disable)	enabled
End of Table 2-15		

2.7.2 AER Optimized Parameter Settings for 8 kHz and 16 kHz Sampling Rates

Table 2-16 lists optimized parameter settings for each of the phone modes using a sampling frequency of 8 kHz for a Texas Instruments IP Phone.

Table 2-16 AER Optimized Parameters for 8-kHz Sampling Frequency

8 kHz case (nominal volume settings only)	HandsFree	HandSet	HeadSet	GL-Handset	GL-Headset
ADC PGA (tx_ag) analog gain (dB)	39	21	15	21	15
DAC PGA (rx_ag) analog gain (dB)	–3	–15	–4.5	–3	–3
Sidetone Gain (dB)	n/a	–7.5	–12	–7.5	–12
AER filter length or tail length (ms)	200	8	60	60	60
AER send path (aer_tx_dg) digital gain (dB)	7	0	0	0	0
AER receive path (aer_rx_dg) digital gain (dB)	0	0	0	0	0
Hangover to switch away from transmit path	10	10	10	10	10
Hangover to switch away from receive path	30	30	30	30	30
NLP linear combined loss target	700	32767	32767	100	100
NLP send path center clipper aggression	7	3	3	9	9
Threshold for receive path non linearity detection	32000	–32768	–32768	32000	32000
Noise power thresholds for send path	5380	5380	5380	5380	5380
Noise power thresholds for receive path	5380	5380	5380	5380	5380
AER Noise Guard desired send noise level	disabled	n/a	n/a	n/a	n/a
AER Noise Guard hangover period	1000	n/a	n/a	n/a	n/a
AER Noise Guard ramping in period	disabled	n/a	n/a	n/a	n/a
AER HLC max signal level (dBm)	–12	–6.5	2		
AER HLC ramp down time constant (ms/dB)	10	10	10		
AER HLC ramp up time constant (ms/dB)	300	700	700		
AER HLC power time constant	1	1	1		
AGC saturation detection threshold	32000	32000	32500	32500	32500
AGC saturation detection hangover	20	20	20	20	20
End of Table 2-16					

Table 2-17 lists optimized parameter settings for each of the phone modes using a sampling frequency of 16 kHz for a Texas Instruments reference IP Phone.

Table 2-17 AER Optimized Parameters for 16-kHz Sampling Frequency

16 kHz case (nominal volume settings only)	HandsFree	HandSet	HeadSet	GL-Handset	GL-Headset
ADC PGA (tx_ag) analog gain (dB)	34.5	15	15	15	15
DAC PGA (rx_ag) analog gain (dB)	–7.5	–18	–4.5	–4.5	–3
Sidetone Gain (dB)	n/a	0	–12	0	–12
AER filter length or tail length (ms)	200	8	60	60	60
AER send path (aer_tx_dg) digital gain (dB)	9	0	0	0	0
AER receive path (aer_rx_dg) digital gain (dB)	0	0	0	0	0
Hangover to switch away from transmit path	10	10	10	10	10
Hangover to switch away from receive path	30	30	30	30	30
NLP linear combined loss target	700	32767	32767	100	100
NLP send path center clipper aggression	7	3	3	9	9
Threshold for receive path non linearity detection	32000	–32768	–32768	32000	32000
Noise power thresholds for send path	5380	5380	5380	5380	5380
Noise power thresholds for receive path	5380	5380	5380	5380	5380
AER Noise Guard desired send noise level	disabled	n/a	n/a	n/a	n/a
AER Noise Guard hangover period	1500	n/a	n/a	n/a	n/a
AER Noise Guard ramping in period	disabled	n/a	n/a	n/a	n/a
AER HLC max signal level (dBm)	–10.5	–2	2		
AER HLC ramp down time constant (ms/dB)	10	10	10		
AER HLC ramp up time constant (ms/dB)	300	700	700		
AER HLC power time constant	1	1	1		
AGC saturation detection threshold	32000	32500	32500	32500	32500
AGC saturation detection hangover	20	20	20	20	20
End of Table 2-17					

Chapter 3

Signal Equalizer MATLAB Design Tool

This chapter contains the following topics:

3.1 "Introduction to Equalization Using Software" on page 3-2

3.2 "Design of the Equalization Filter" on page 3-3

3.3 "Parameter Optimization" on page 3-5

3.4 "Output" on page 3-13

3.1 Introduction to Equalization Using Software

Most IP Phone manufacturers and service providers require some standard specifications to be met for the spectral response of their transducers in transmit and receive directions. The choice of standard depends on the requirements of country where their IP Phones will be deployed. Acoustical reproduction of speech spectrum is best accomplished with audio components that have a flat spectral response; therefore, most standards require that spectral response in the voice pass band be flat within some allowable tolerance. However, the transducers used in the IP Phone may not meet these requirements; this makes it necessary to use software equalization where hardware modification is not viable.

This document contains guidelines for customers who need to design an equalization filter for IP Phones that they will be using with the Texas Instruments IP Phone software with Acoustic Echo Canceller (AEC). Note that AER 15.1 supports two types of equalizers, and only one of them is described in this chapter. The other one, based on Bi-quad implementation, is described in the design tool, which can be obtained separately. In future AER releases, the documentation for Bi-quad equalizer will be added to this document.



WARNING—This tool shows how equalization filter parameters can be obtained. Texas Instruments does not claim these to be the best or most optimal set of parameters. It is the responsibility of the customer to make sure that the parameters obtained from this tool are suitable for their application. Texas Instruments does not provide customer support for this tool.

3.2 Design of the Equalization Filter

Figure 3-1 shows a block diagram of the equalization filter. The equalization filter has two primary components:

1. IIR (Infinite Impulse Response) filter or an ARMA (Auto Regressive Moving Average) filter
2. FIR (Finite Impulse Response) filter

You can use one or both of these components in the filter design based on the requirements of phase distortion and group delay. The equalizer design also has five gain units. The gain units ensure that the filter coefficients are represented as 16-bit fixed-point values and that the signal output at the end of MA (moving-average), AR (auto-regressive), and FIR filter does not overflow or underflow. Each of these gains is specified in 6 dB steps. All filter coefficients are in the range of [-1, 1] as a result of normalization in the design procedure. The effect of normalization is included in the gains.

Figure 3-1 Block Diagram of the Equalization Filter

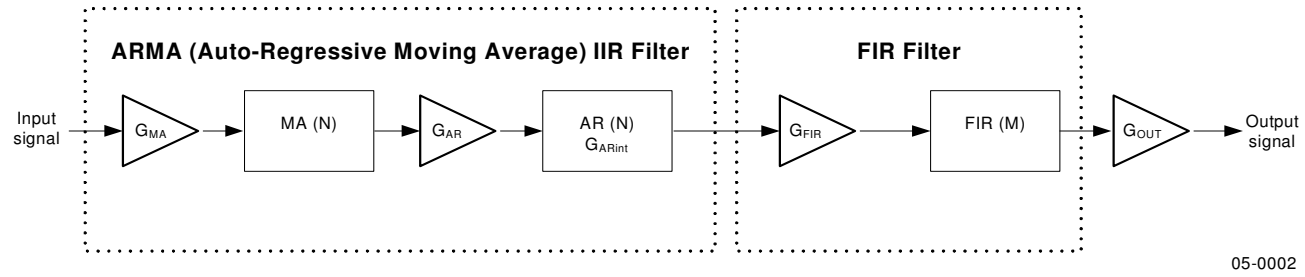
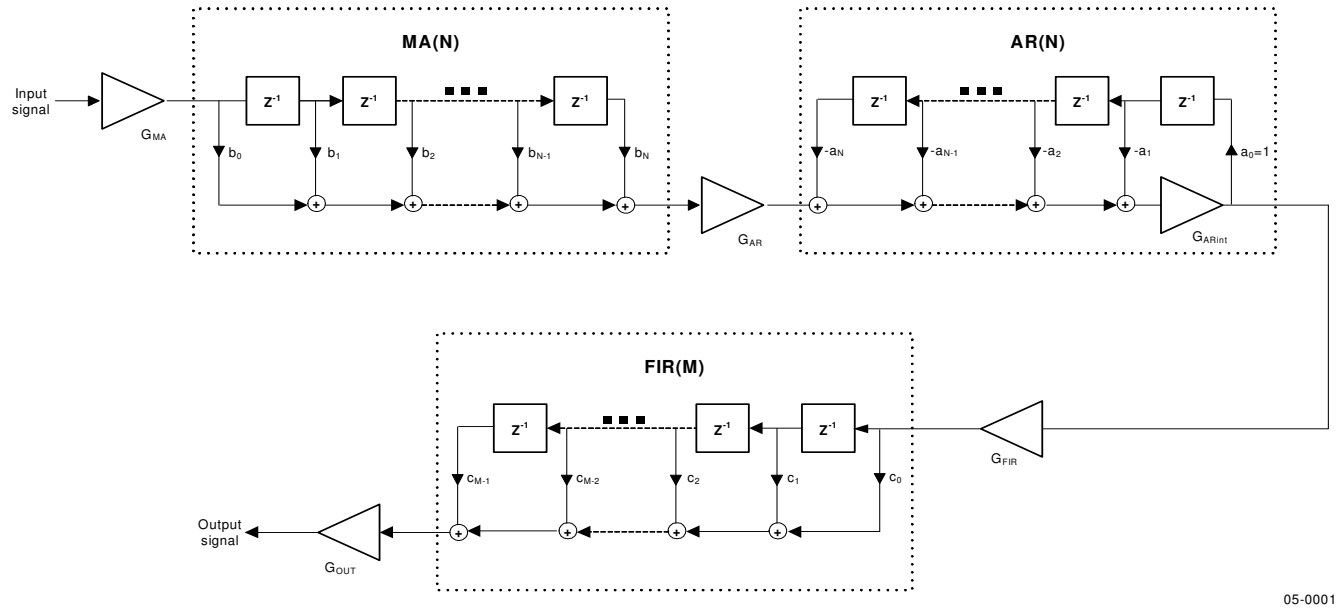


Table 3-1 Definitions of Equalization Filter Elements

Item	Description
N	Order of the IIR filter
M	Length of the FIR filter
G_{MA}	Input gain for the MA (moving average) stage of IIR filter, including gain to compensate for preventing overflows/underflows.
G_{ARint}	Internal gain of the AR (auto-regressive) stage of IIR filters, used only for coefficient normalization.
G_{AR}	Input gain for the AR (auto-regressive) stage of IIR filter, including gain to compensate for preventing overflows/underflows.
G_{FIR}	Input gain for the FIR filter, including gain to compensate for preventing overflows/underflows.
G_{OUT}	Output gain to compensate for the portions of G_{MA}, G_{AR} and G_{FIR} that were used to prevent overflows/underflows.

Figure 3-2 shows the internal structure of the equalization filter.

Figure 3-2 Internal Structure of the Equalization Filter



The purpose of the equalizer is to help fit the frequency response between the masks when the response varies from the acceptable behavior by about ±6dB. If you try to correct larger impairments with the equalizer, it may not always be feasible to balance the added requirements of keeping distortion or quantization noise at a minimal level.

In general, do not use the equalizer to compensate for deficiencies in enclosure design or poor transducer quality. Such problems are better solved by modifying the hardware.

If the equalization requirement becomes too demanding, it is better to study the cause of the problem to determine if the enclosure or transducer response can be improved. Equalization cannot usually compensate effectively for

- enclosures with leaks and large resonances
- attempts to get wideband performance from a narrow band transducer
- enclosure-induced distortion

3.3 Parameter Optimization

The equalization filter consists of an IIR filter, an FIR filter, and five gain units. The order of the IIR filter is N and the FIR filter has M taps. In general, you can omit the IIR or the FIR components in the equalizer design.

1. **IIR order**: N is the total number of zeros (or poles) in the IIR filter. For a real filter, the zeros (and poles) are in conjugate symmetric pairs. As a result of normalization, the first coefficient for the MA or AR stages may not be equal to 1. Therefore, there are (N+1) coefficients each for the MA and AR stages of the IIR filter.
2. **FIR length**: M is the total number of taps in the FIR filter. As a result of normalization, the first coefficient of the FIR may not be equal to 1. Therefore, all the M coefficients of the FIR filter are required.

All filter coefficients are in the range of [-1, 1] due to normalization in the design procedure.

The following command executes the script that carries out the optimal equalization filter computations:

```
>> aereq
```

The script requires several user inputs. The tool has default values for most of the input parameters.

When the script runs for the first time, the defaults appear in square brackets. If you press RETURN without specifying an input value, the default value is used. If you enter a new value, the new value is used. In subsequent runs of the script, the tool API displays the value used by the previous run of the script. Just press RETURN to use the same value as the last run.

The following sections describe the user inputs:

- Measured Frequency Response
- Target Frequency Response
- Margin from Target Response
- Sampling Frequency
- IIR and FIR Requirements
- Mask Description
- Roll-Off Specification
- Extension of Lower Frequency Mask
- Safety Margin from the Frequency Masks
- Safety Margin from 3 dBm0 Reference for Overflow Analysis
- Maximum Number of Iterations

3.3.1 Measured Frequency Response

This is the response of the transducer (Tx or Rx path) without equalization; it is obtained from user measurements from some reliable method. This input should be provided in an input response file (.txt) that has the frequency response(s) specified at all $1/12^{th}$ octave frequencies, as used by most standards.

The file *myresp.txt* should contain the response information.

```
Enter the filename with actual and desired (optional) response []:  myresp.txt
```

The same file is used for the parameter Target Frequency Response.

3.3.2 Target Frequency Response

This is the desired frequency response after equalization. Enter "1" if the input file contains the target response. It is not necessary to specify the target response. Usually, the equalizer targets a response based on the specified masks. This input, if provided, should be provided in an input response file (.txt). It is required that target frequency response be specified at the same frequencies as the measured response and at the same overall gain settings.

Format of the response file:

f_resp = file with actual and required loudness rating frequency response.

The file should be arranged in 2 (or 3) columns, as follows:

Column 1 = frequencies at which response is specified

Column 2 = measured frequency response

Column 3 = desired frequency response (if you choose to specify it)

```
Enter the filename with measured and target(optional) response []: myresp.txt
Target response specified in response file above? 0=No, 1=Yes [0]: 0
```

3.3.3 Margin from Target Frequency Response

With this parameter, you can specify how much deviation above or below the target frequency response is acceptable. Normally, you can allow small deviations from the target frequency response. The objective is usually to have the response fit within the masks rather than try to achieve the exact midpoint between the two masks. Therefore, you do not need to compensate for 'acceptable' deviations that fall within the margin you specify.

```
Specify acceptable margin from target response (dB) [1]:
```

3.3.4 Sampling Frequency

The equalizer design is supported for both narrow band (8 kHz) and wideband (16 kHz). This parameter is used in the optimization process as well as in selecting the appropriate masks.

Enter a value in the following format:

8 kHz (narrow band) = 8000

16 kHz (wideband)= 16000

```
Specify Sampling Frequency - 8000 or 16000 Hz [8000]: 8000
```

3.3.5 IIR and FIR Requirements

You can specify whether to include the IIR/FIR components of the equalizer in the design process. If you choose the FIR filter, you can also specify whether the FIR is symmetric.

- IIR order (see "Parameter Optimization" on page 3-5)—The maximum value allowed is 8. If you do not want to use the IIR filter in the equalizer, use a value of 0.
- FIR length (see "Parameter Optimization" on page 3-5)—The maximum value allowed is 19. If you do not want to include the FIR filter in the equalizer, use a value of 1.
  - N should be an even non-negative integer (otherwise the nearest smaller even number is used).
  - M must be an odd non-negative integer (otherwise the nearest smaller odd number is used).



CAUTION—You must specify these values carefully because they are the most critical design parameters. Over-specifying or under-specifying the filter may result in less than optimal performance. As a guideline, begin by choosing the lowest orders or lengths of the IIR and FIR filters as possible without compromising performance.

```
Specify Order (N) of the IIR filter (must be even,0<=N<=8) [4]: 2
Specify number of taps (M) for the FIR filter (must be odd,0<=M<=19) [5]: 9
Constrain FIR to be symmetric? 0=No, 1=Yes [0]: 0
Mask Description
```

The requirements to shape the spectral responses in the Tx and Rx paths are usually guided by requirements of one of the three standard committees—TIA, ITU, and IETS. There are different requirements for different phone modes.

You must specify three parameters to determine exactly the masks to use—the name of the standard, the phone mode, and the direction; these parameters are in addition to the value specified in "Sampling Frequency" on page 3-6.

3.3.5.1 Standard Name

The name code is specified at the command line:

- TIA = 0
- ITU = 1
- IETS = 2
- TBR8 = 3
- TBR10 = 4
- user-defined = 5

If you specify your own masks in an input file (.txt), the other two mask descriptors (phone mode, direction) are not used.

The file *mymasks.txt* should contain the mask information in the following format:

f_mask = file with lower and upper frequency masks specified. The file should be arranged in 3 columns, as follows:

Column 1 = frequencies at which mask specified. Specify the mask at 1/3rd or 1/12th octave frequencies where most of the standard masks are specified.

Column 2 = lower frequency mask

Column 3 = upper frequency mask

3.3.5.2 Phone Mode

To select the correct mask, you must specify if the response for the transducer is in handsfree, handset, or headset mode. If a selected phone mode does not have the masks specified, TIA masks for that mode are used.

3.3.5.3 Direction

You must specify if the transmit or receive direction response must be equalized, so that the appropriate mask can be selected.

Inputs for standards-based masks:

```
Select the standard to use: 0=TIA, 1=ITU, 2=IETS, 5:User Defined [0]:  1
Select usage mode: 0=Handsfree, 1=Handset, 2=Headset [0]:  2
Select whether receive/transmit masks to use: 0=Receive, 1=Transmit [0]:  1
```

Inputs for user-designed masks:

```
Select the standard to use: 0=TIA, 1=ITU, 2=IETS, 5:User defined [0]:  3
Enter the filename containing the lower and upper masks []: mymasks.txt
```

There are four standard specifications that the tool allows you to select:

- TIA
- ITU
- IETS
- TBR

These standards specify the upper and lower frequency masks at specified frequencies. However, the specified frequencies may not be consistent across these standards and may change with the phone mode or direction (Rx/Tx) within the same standard.

The masks for these standards are specified in files in the directory `optfilter/stdmasks`.

The MATLAB file **`getmaskfile.m`** associates each standard with an input mask file according to the phone mode, direction, and sampling rate in use. Table 3-2 lists the filenames in getmaskfile.m. You can edit these files if you need to change a value in these masks due to a discrepancy or deviation from the specifications of a standard.

Table 3-2 Standard Masks in the optfilter/stdmasks Directory (Part 1 of 2)

S.No.	Filename	Description[1]
1	TIA810_HFRx.txt	TIA810B, handsfree, 8kHz, Receive
2	TIA810_HFTx.txt	TIA810B, handsfree, 8kHz, Send
3	TIA810_HSRx.txt	TIA810B, handset, 8kHz, Receive

Table 3-2 Standard Masks in the optfilter/stdmasks Directory (Part 2 of 2)

S.No.	Filename	Description[1]
4	TIA810_HSTx.txt	TIA810B, handset, 8kHz, Send
5	TIA810_HeSRx.txt	TIA810B, headset, 8kHz, Receive
6	TIA810_HeSTx.txt	TIA810B, headset, 8kHz, Send
7	TIA920_HFRx.txt	TIA920, handsfree, 16kHz, Receive
8	TIA920_HFTx.txt	TIA920, handsfree, 16kHz, Send
9	TIA920_HSRx.txt	TIA920, handset, 16kHz, Receive
10	TIA920_HSTx.txt	TIA920, handset, 16kHz, Send
11	TIA920_HeSRx.txt	TIA920, headset, 16kHz, Receive
12	TIA920_HeSTx.txt	TIA920, headset, 16kHz, Send
13	ITU342_HFRx.txt	ITU P342, handsfree, 8kHz, Receive
14	ITU342_HFTx.txt	ITU P342, handsfree, 8kHz, Send
15	ITU310_HSRx.txt	ITU P310 (or ITU P313), handset (or cordless), 8kHz, Receive
16	ITU310_HSTx.txt	ITU P310 (or ITU P313), handset (or cordless), 8kHz, Send
17	ITU341_HFRx.txt	ITU P341, handsfree, 16kHz, Receive
18	ITU341_HFTx.txt	ITU P341, handsfree, 16kHz, Send
19	ITU311_HSRx.txt	ITU P311, handset, 16kHz, Receive
20	ITU311_HSTx.txt	ITU P311, handset, 16kHz, Send
21	IETS24503_HFRx.txt	ETSI 300.245.03, handsfree, 8kHz, Receive
22	IETS24503_HFTx.txt	ETSI 300.245.03, handsfree, 8kHz, Send
23	IETS24502_HSRx.txt	ETSI 300.245.02, handset, 8kHz, Receive
24	IETS24502_HSRx.txt	ETSI 300.245.02, handset, 8kHz, Send
25	IETS24506_HFRx.txt	ETSI 300.245.06, handsfree, 16kHz, Receive
26	IETS24506_HFRx.txt	ETSI 300.245.06, handsfree, 16kHz, Send
27	IETS24505_HSRx.txt	ETSI 300.245.06, handset, 16kHz, Receive
28	IETS24505_HSRx.txt	ETSI 300.245.06, handset, 16kHz, Send
29	TBR008_HSRx.txt	TBR 008, handset, 8Hz, Receive
30	TBR008_HSTx.txt	TBR 008, handset, 8kHz, Send
31	TBR010_HSRx.txt	TBR 010, handset, 8Hz, Receive
32	TBR010_HSTx.txt	TBR 010, handset, 8kHz, Send
End of Table 3-2		

1. For cordless phones or for WLAN IP Phone handsets, it is recommended that you use ITU P313 masks, which are the same as ITU P310 masks for IP Phone handsets.

For simplicity, the Texas Instruments tool applies mask constraints at all one-third-octave frequencies starting at 100Hz, regardless of whether the standard specifies the mask at that frequency. The value of the mask at each one-third-octave frequency is obtained directly from the standard or linearly interpolated if the standard does not provide a value at that frequency.

Therefore, the constraints are stricter than required, but it helps to make sure that the equalized response does not deviate too much between the mask frequencies (which are sometimes widely-spaced). If you do not want to use the one-third-octave frequency masks, you can specify your own mask through the file `mymasks.txt`.

It is also important to note that the ITU and IETS standards do not define the masks for all phone modes for both Rx and Tx directions. In such cases, the following rules apply:

- If the handset response is not available for a specific configuration, the corresponding TIA mask is used.
- If the headset response is not available for a specific configuration, the handset mask of the same standard for the same configuration is used.

To use a different mask, specify a mask in the file `mymasks.txt`.

If you selected TIA, ITU, or IETS masks, the masks are displayed in the results figures so that the transitions in the mask values appear at the midpoint of two one-third-octave frequencies (computed as the square root of the two adjacent one-third-octave frequencies). For user-specified masks, the masks are displayed exactly as specified.

3.3.6 Roll-Off Specification

The masks usually require the frequency response to lie between the upper and lower mask specifications. If you do not specify a target response, the design tool uses the midpoint of the masks as the target. The lower mask specifications are set to a value of -∞ outside the voice band—for example, under 315 Hz or over 3150 Hz for TIA-810B. In that case, targeting the midpoint of the two masks is not ideal. Instead, you must specify the minimum roll off in the frequency response outside the voice band where the lower mask is not defined. The response of the equalized signal should meet or exceed this requirement.

Sometimes, it is desirable to ensure that the roll-off continues until it reaches the lowest and highest frequencies of interest (example: 1Hz at the lower end and 4kHz at the higher end for 8kHz sampling), even though the measured response or masks are not provided at these frequencies (example: measured response provided between 100 and 3400 Hz only). In that case, you can choose to apply the roll-off specifications even to these frequencies.

The equalizer tool is designed to handle such requirements. It attempts to optimize the filter parameters so that the equalized response meets the specified roll-off criterion. You can specify different roll-off requirements at the lower and higher frequency ends. The measured response and masks are both assumed to be flat outside of the specified range. This may not be a realistic assumption (especially for the measured response), but if you want to optimize for these frequencies, it is desirable for the measured response (or masks) to be available at these frequencies.

Input depends on whether the target response is specified in `myresp.txt`:

- If the target response is not specified in the input file *myresp.txt*, you are prompted to answer the following:

```
Specify low frequency roll-off/octave? (dB units, <=0)[-6]: -9
Specify high frequency roll-off/octave? (dB units, <=0)[-3]: -4
Continue to apply roll-off at frequencies outside the mask frequencies?
                                                    0=No, 1=Yes [1]: 1
```

A “0” specifies that the roll-off is applied only within the range of the frequencies where the lower mask is -∞. A “1” specifies that the roll-off specification should continue to be applied outside these mask frequencies. At the lower end, the roll off continues till 1 Hz and at the upper end, roll off is applied till Fs/2 (Fs = sampling frequency). Note that the roll-off is always applied at all frequencies where the lower mask is -∞.

- If the target response is provided in the file *myrest.txt*, you are prompted to provide the following information:

```
Apply roll-off at frequencies outside measurement frequencies? 0=No, 1=Yes [1]: 1
Specify low frequency roll-off/octave? (dB units, <=0)[-6]: -6
Specify high frequency roll-off/octave? (dB units, <=0)[-3]: -6
```

The target response is assumed to implicitly contain all roll-off requirements at the measurement frequencies. You are prompted to choose only if there are additional roll-off requirements outside the specified frequency range. If “1” is specified, the additional roll-off is applied. You can also specify the roll-off requirements at either end. At the lower end, the roll off continues till 1 Hz and at the upper end, roll off is applied till Fs/2 (Fs = sampling frequency).

3.3.7 Extension of Lower Frequency Mask

Sometimes, at the end of optimization—especially when strict roll-offs are specified—it is possible to have the equalized response which barely misses the lower mask at the lower and higher ends just before the lower mask becomes -∞. Because this is not always desirable, the tool lets you select if the lower mask should be extended outwards somewhat before constraints are applied. By extending the mask, the roll off is deferred to the new rims of the new lower mask. Therefore, the result of optimization solves an over-specified problem, and helps make sure that the response lies comfortably above the ends of the lower mask.

```
Extend the specified lower mask at mask boundaries? 0=No, 1=Yes [1]:
Specify how much to extend mask at lower end?(percentage>=0) [20]: 25
Specify how much to extend mask at higher end?(percentage>=0) [10]: 15
```

3.3.8 Safety Margin from the Frequency Masks

This input specifies an extra safety margin so that the equalized frequency response does not exceed the upper and lower limits of the mask. These margins are defined uniformly for all frequencies where the mask is specified. These margins should not be too high; otherwise the optimizer may find it hard to converge. A value of 0.5 dB is recommended.

```
Specify safety margin (dB) from the upper mask [0.5]: 0.2
Specify safety margin (dB) from the lower mask [0.5]: 0.8
```

3.3.9 Margin from 3 dBm0 Reference for Overflow Analysis

This input specifies a margin that the user wants to specify for the full scale signal (3dBm0 reference). This reference is used to make sure that there are no overflows/underflows for the MA, AR, or FIR stage outputs.

The selection of this margin may be standards-based. For example, the TIA810B handset specifications would allow this margin to be +3dB for all handset loudness and distortion testing. Using a higher margin enables higher gains for equalization, making the design less susceptible to quantization noise.

```
Specify safety margin (dB) from 3dBm0 reference for overflow analysis [0.5]: 1
```

In some cases, the value of the upper mask may be above the full scale reference. In such cases, you can update the upper mask value at that frequency by the 3 dBm0 reference value instead; you are prompted to choose whether you want to update the upper mask. This would determine the constraints applied for optimization and the final result.

```
Use 3 dBm0 reference to update upper mask? 0=No, 1=Yes [1]:
```

3.3.10 Random Number Generator State for Initial Conditions

The initial conditions of the equalizer parameters are generated using the random number generator from MATLAB. The state of the random number generator determines the initial conditions. In some cases, depending on where the initial conditions occur, the optimization function may get stuck at some local minima and would not be able to converge to a better solution. In such cases, it may help to try a different set of initial conditions and arrive at the solution. The input for random number generator state (any integer value) helps to handle such cases. Specifying a certain fixed random number generator state always results in the same set of initial conditions.

```
Specify the random number generator state for initial conditions [0]: 2
```

3.3.11 Maximum Number of Iterations

This input enables you to specify the maximum number of optimization iterations. In general, a value around 500 works very well; however, more iterations may be needed when the measured response has exceptional variations from the desired response.

```
Specify the maximum iterations for filter optimization [1000]: 400
```

3.4 Output

```
>> aereq

--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--

      Texas Instruments Equalizer Design Tool for Telogy AEC Software

    Tool Version Number:  2.15    (C)Copyright 2004, Texas Instruments

--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--


Please provide the following inputs to the equalizer tool -->

NOTE: Use forward slash(/) to specify path, not back slash(\) (example c:/x/y.txt)
Enter filename with measured and target(optional) response [ ]:
examples/hsnbrlr.txt
Target response specified in response file above? 0=No, 1=Yes [0]:
Specify acceptable margin from target response (dB) [1]:  2
Specify Sampling Frequency - 8000 or 16000 Hz [8000]:
Specify Order (N) of the IIR filter (must be even,0<=N<=8) [4]:
Specify number of taps (M) for the FIR filter (must be odd,0<=M<=19) [5]:  3
Constrain FIR to be symmetric? 0=No, 1=Yes [0]:
Select the standard to use: 0=TIA, 1=ITU, 2=IETS, 3=TBR8, 4=TBR10, 5=User Defined
[0]:
Select the usage mode: 0=Handsfree, 1=Handset, 2=Headset [0]:  1
Select whether receive/transmit masks to use: 0=Receive, 1=Transmit [0]:
Specify margin (dB) from 3dBm0 reference for overflow analysis [-0.5]:  3
Specify low frequency roll-off/octave? (dB units, <=0)[-6]:
Specify high frequency roll-off/octave? (dB units, <=0)[-3]:  0
Continue to apply roll-off at frequencies outside measurement frequencies? 0=No,
1=Yes [1]:
Extend the specified lower mask at mask boundaries? 0=No, 1=Yes [1]:
Specify how much to extend mask at lower end?(percentage>=0) [20]:  25
Specify how much to extend mask at higher end?(percentage>=0) [5]:
Specify safety margin (dB) from the upper mask [0.5]:  1
Specify safety margin (dB) from the lower mask [0.5]:  1
Use 3dBm0 reference to update upper mask? 0=No, 1=Yes [1]:
Specify the random number generator state for initial conditions [0]:  1
Specify the maximum iterations for filter optimization [1000]:  200
This starts the filter design using the MATLAB Optimization Toolbox and the
diagnostics from the optimizer appear on the screen as follows.
                        max                          Directional   First-order
Iter F-count  f(x)    constraint  Step-size   derivative   optimality Procedure
0   12      623.604     15.43                           Infeasible  start point
1   26      444.993     9.02        0.5     2.54e+003   1.01e+003   infeasible
2   39      72.6984     4.855       1       -235        1.02e+003   infeasible
3   53      102.609     2.908       0.5     85.4        6.46e+004
4   69      61.4432     2.678       0.125   1.22e+003    912    Hessian modified
5   84      294.714     5.512       0.25    1.23e+003    5.18e+003
6   98      108.241     5.153       0.5     1.98e+003   4.05e+003  Hessian modified

… and so on. When optimization finishes, a message similar to the following appears:

Optimization terminated successfully:
 Magnitude of directional derivative in search direction
  less than 2*options.TolFun and maximum constraint violation
  is less than options.TolCon
Active Constraints:
    18
   173
The final result of executing the script is the following parameters as the output
of optimization.

--------------------------------------------------------------------------------
                    OPTIMAL EQUALIZATION FILTER PARAMETERS

<--- iir_params = [(8+1) MA Coefficients, (8+1) AR Coefficients] --->

N =
     2

MA_params =
       32767
      -25747
        7322
           0
```

```
           0
           0
           0
           0
           0

AR_params =
       32767
      -25654
       10041
           0
           0
           0
           0
           0
           0

<--- fir_params = 19 FIR Coefficients --->

M =
     7

fir_params =
       17804
        -429
       -5752
      -32753
       26812
       -9369
        3683
           0
           0
           0
           0
           0
           0
           0
           0
           0
           0

<--- gain_params = MA input gain, AR input gain, AR internal gain, FIR input gain
                   Output gain (6dB steps) --->

gain_params =
    -1
    -2
     1
     0
     3

<--- Overall system gain (dB) adjustment required (not accomplished by equalizer)

gain_offset =
   -4.5388

--------------------------------------------------------------------------------
Optimal equalizer parameters are also written to an output file (output.dat)
--------------------------------------------------------------------------------
```

The output includes the (8+1) MA coefficients in **MA_params**, (8+1) AR coefficients in **AR_params** (unused MA and AR coefficients are set to 0 if an order lesser than 8 is used), M FIR coefficients in **fir_params** (unused FIR coefficients are set to 0 in case fewer than 19 taps are used) and four gain values in **gain_params**.

When you run the script from the MATLAB command line, the following output appears in the MATLAB window at the end of the run. In this example, IIR and FIR were used and the values of N and M were 2 and 7, respectively. The IIR/FIR coefficients are all output as 16-bit, fixed-point numbers and the gains are displayed here as the number of 6 dB steps. The overall **gain_offset** is also displayed.

NOTE—The first coefficient in AR_params is always 32767. Although it is output by the tool, this parameter is ignored by the DSP.

gain_params: A gain of –1 indicates 6 dB attenuation and can be accomplished by a right shift in the Texas Instruments DSP implementation. A gain of +1 indicates a 6 dB increase and can be implemented using a left shift in the Texas Instruments DSP implementation. G_{MA}, G_{AR}, and G_{FIR} are always <=0 by design. G_{ARint} may be positive or negative. The last of the five gains—G_{OUT}— is a positive gain that accounts for the negative gains G_{MA}, G_{AR}, G_{FIR} introduced to avoid overflows in the MA, AR, and FIR stages of the filter.

The equalizer design tool allows the overall equalizer transfer function to have positive gains at some frequencies. This may not be acceptable in some cases. For example, in an Rx equalizer designed for the handset for TIA810B, the maximum gain that the equalizer should apply (to avoid flat-top saturation under any Rx path TIA810 testing) is +3 dB at the frequencies of interest. But the tool may yield a design that has the equalizer transfer function with a maximum gain >3 dB at the frequencies of interest. To ensure that there is no flat-top saturation due to the equalizer, you may want to adjust G_{OUT} manually to a lower value so the overall equalizer transfer function stays at or below the acceptable level. However, doing so would impact the loudness, and require adjustment to the analog gain of DAC PGA to compensate for the lost loudness.

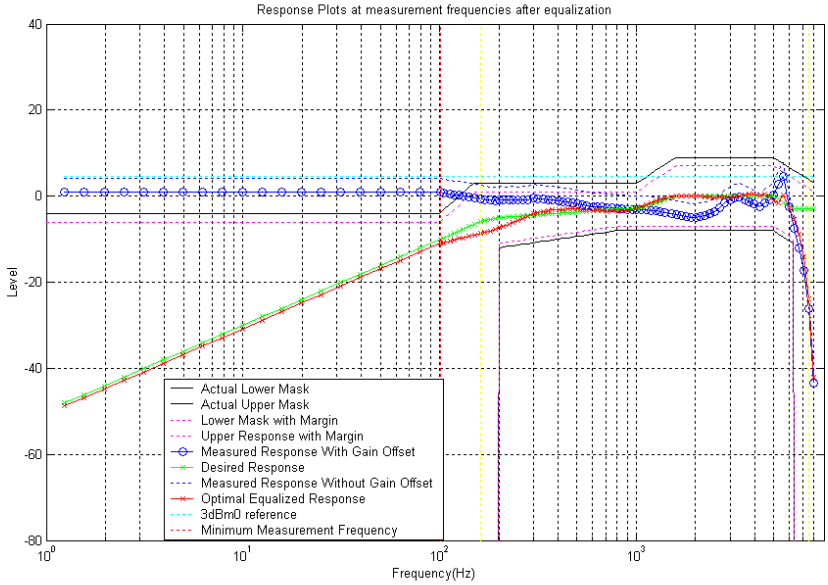
Before starting optimization, the tool compensates for any gross mismatch in levels by translating the measured response so that it starts with minimum deviation from the masks. **gain_offset** is the parameter that specifies how much the response needed to be translated up or down to make the best fit within the masks. The equalizer does not provide gain or loss to compensate for this offset. You must adjust external gains to make sure that the offset is accounted for.

In addition to the parameters, the tool also produces a series of plots that you can use to help interpret the results. The plots produced from the example above are shown and described below.

Plot 1—*Plot of the responses at measurement frequencies before equalization* (Figure 3-3). This plot also shows the effect of gain offset on the measured response (blue curves). In this example, the target response was not specified; therefore, the midpoint of the masks with appropriate roll off is being used. The solid green line on the plot reflects the target response, specified within the margins indicated by the dotted green lines. The cyan line shows the 3dBm0 reference. The magenta lines on the plot reflect the upper and lower masks after taking into account the user-specified margins from the actual upper and lower masks in black. The red line is the equalized response.

Measured response is assumed flat outside the range of measurement frequencies. The roll off (if applicable) for the target response applies in this region, too. The yellow vertical lines indicate the frequencies to which the lower mask has been extended (if so requested) on the lower and higher frequency ends.

Figure 3-3 Plot 1—Responses at Measurement Frequencies Before Equalization



Plot 2—Plots of the responses at mask frequencies after equalization: In general, the measurement and mask frequencies are different. However, in the case of this example, they may be the same. If the mask and measurement frequencies are identical, Plot 1 and Plot 2 will also be identical, so the tool only creates Plot 1.

Figure 3-4 Plot 3—Magnitude of the Equalizer Transfer Function at Measurement Frequencies

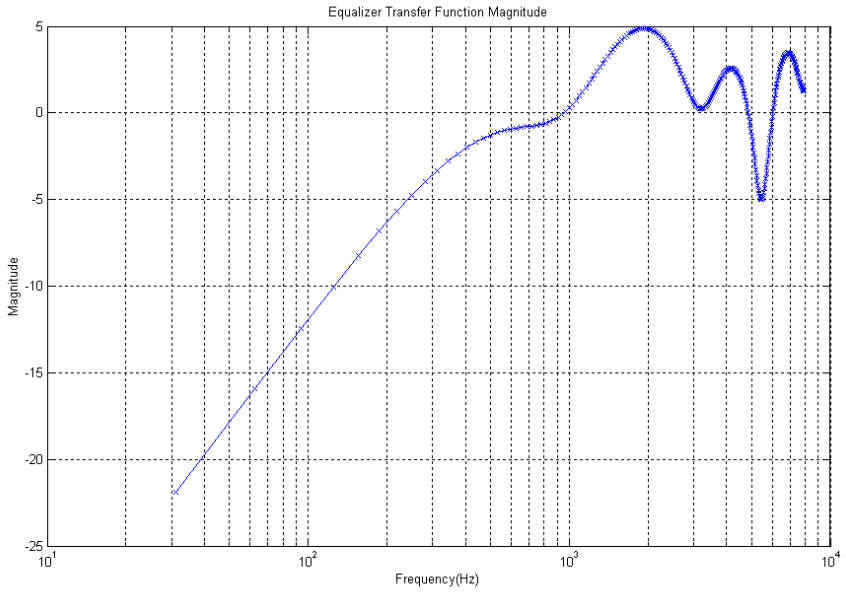


Figure 3-5 Plot 4—Pole Zero Plot of the Equalizer IIR (if order>0)

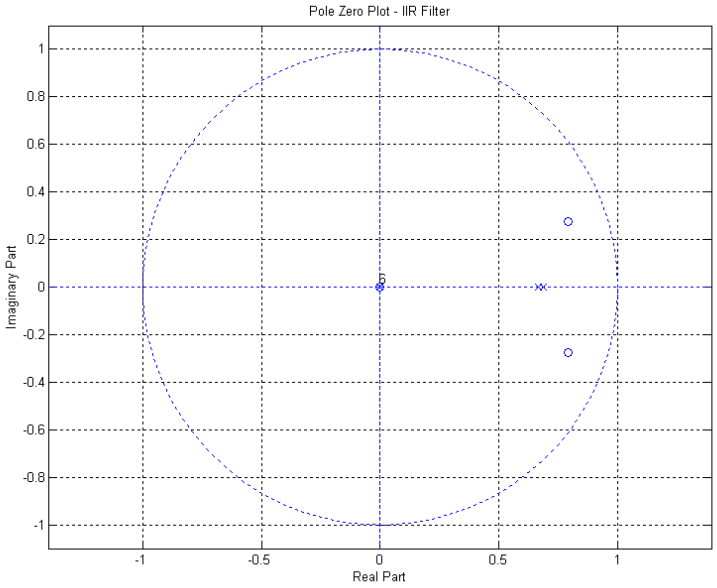


Figure 3-6 Plot 5—Zero Plot of the Equalizer FIR (if length >1)

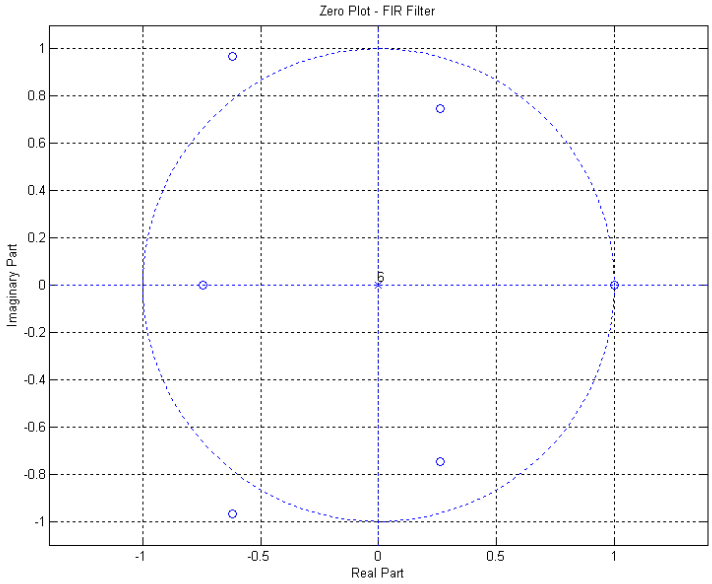
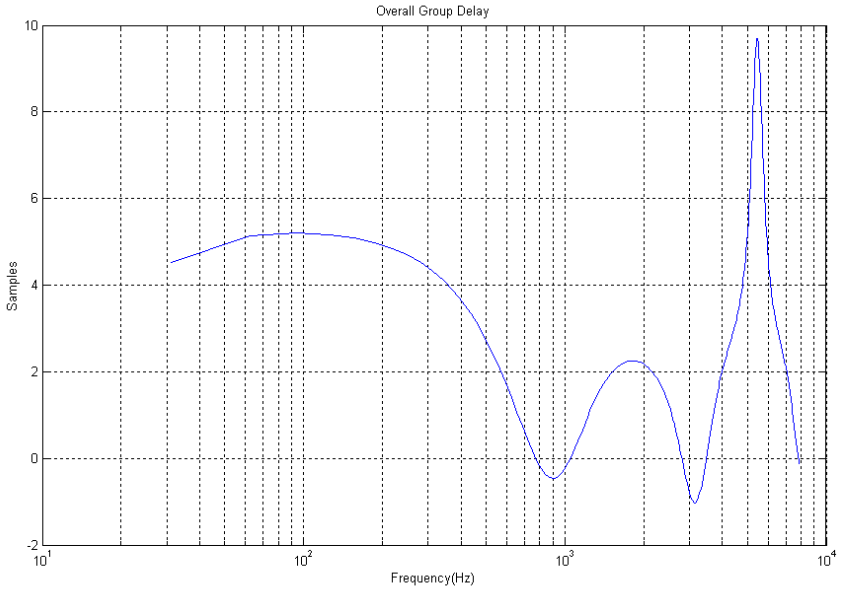


Figure 3-7 Plot 6—Overall Group Delay of the Equalizer



3.4.0.1 Notes

- You must try to change the input parameters and not just rely on the defaults to make sure that you get the best results. Among the parameters that would have the most impact on how the equalized response looks are
  - Decision whether to use IIR or FIR
  - Order of the IIR filter, if used
  - Length of the FIR filter, if used
  - Roll-off specifications
  - Margins from the upper and lower masks
  - Whether to extend the lower mask, and by how much

  The plots above are a good visual indication of how well the design criteria have been met.
- Texas Instruments DSP software uses a fixed order (8) for the IIR and a fixed length (19) for the FIR. Therefore, although you may specify a value of N less than the maximum of 8 and a value of N less than the maximum of 19, the output of the tool provides the complete set of coefficients (9 each for MA and AR stages, 19 for FIR stage). These are the values you must use to configure the Texas Instruments DSP software.
  - If you select a smaller value for the IIR filter, the other poles and zeros will be added at origin, and will not impact the magnitude or group delay for the IIR filter. If you do not use the IIR filter, all poles and zeros are at the origin (only the first coefficient for the MA and AR parts would be set to 32764 (1 in 16-bit representation).
  - If you select a smaller value for the FIR filter, the other coefficients are all set to 0 and will not impact the magnitude or group delay for the FIR filter. If you do not use the FIR filter, all coefficients are set to 0, except the first one, which is set to 32764 (1 in 16-bit representation).
- If the gain or loss required for overflow or underflow compensation in the MA, AR, or FIR stages is too high (>18 dB, equivalent to 3 bit shifts), a warning message appears with the output, as follows:

```
**** WARNING: This may not be the optimal set of parameters ****


Too much overflow/underflow adjustment gains used! Please try again with a
different set of parameters
```

These results are also printed into an output file (`output.dat`) that is created in the local directory from where the script is executed.

The output file looks something like the following for the current example.

```
------------------------------------------------------------------------

                    TELOGY EQUALIZER PARAMETERS

                      Tool Version Number:2.15

------------------------------------------------------------------------


Measured Response obtained from file: examples/hsnbrlr.txt

Sampling Frequency (Hz): 8000

Mask Specification
------------------
Standard (0=TIA, 1=ITU, 2=IETS, 3=TBR8, 4=TBR10, 5=User Defined): 0
Usage Mode (0=Handsfree, 1=Handset, 2=Headset): 1
Direction (0=Receive, 1=Transmit): 0

Roll-Off and Mask Extension Options
-----------------------------------
Low Frequency Roll-Off(dB/octave): -6
High Frequency Roll-Off(dB/octave): 0
Roll-off requirement also applied outside the measurement frequencies
Lower mask extension on the low frequency end (percent): 25
Lower mask extension on the high frequency end (percent): 5

Margins
-------
Acceptable margin from target response (dB): 2
Margin from upper mask (dB): 1
Margin from lower mask (dB): 1
Margin from full scale (3dBm0) reference for overflow analysis (dB): 3
Upper mask updated with the 3dBm0 reference:
New upper mask = min(original upper mask, 3dBm0 ref) at each mask frequency.

Optimization Settings
---------------------
Maximum number of iterations: 200
Random Number Generator State: 1


MA Coefficients (Order=4):
--------------------------
16384
-21620
4796
1578
86
0
0
0
0

AR Coefficients (Order=4):
--------------------------
32767
-28415
12665
0
0
0
0
0
0

FIR Coefficients (Length=3, Non-symmetric):
-------------------------------------------
18043
9673
-27706
0
0
0
0
0
0
0
```

```
0
0
0
0
0
0
0
0
0

Gains (6dB steps):
------------------
G_MA =  0
G_AR =  0
G_ARint =  1
G_FIR =  0
G_out =  1

Gain offset(dB) = -10.5509

Average Equalizer Delay (samples) = 3.81345


Example DIM messages to be sent to the DSP
------------------------------------------
To load equalizer parameters, use:
dim msg 0 0 MSG_ID 0 4 0 0 DIR 16384 -21620 4796 1578 86 0 0 0 0 32767 -28415 12665
0 0 0 0 0 0 18043 9673 -27706 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1

where DIR is the choice of transmit/receive equalizer, MSG_ID is
the message ID for equalizer configuration message.

Similarly, messages can be sent to enable/disable the equalizer
For more information, see the DIM API manuals!
```

NOTE—At the end of the file, there is an example DIM message that could be used to send the equalizer parameters to the DSP. Telogy software should also provide other means and APIs to send these parameters and settings for the transmit/receive equalizers. Those details are outside the scope of this document. ***It is the responsibility of the user to verify that the message format and contents are correct***.

Chapter 4

Dynamic Range Compression (DRC)



NOTE—The DRC full-band compressor and limiter are upgraded versions of the AER HLC and AER Rx SLIM, respectively. When DRC is activated, make sure that AER HLC and AER Rx SLIM are disabled or inactive.

This chapter contains the following sections:

4.1 "Introduction" on page 4-2
4.2 "DRC Functionality" on page 4-5
4.3 "Configuring a Specific Set of DRC Parameters" on page 4-7
4.4 "Choosing Values for the DRC Configurable Parameters" on page 4-9
4.5 "DRC Time Constants" on page 4-12
4.6 "Multiband Compression Considerations" on page 4-16
4.7 "DSP DRC Configurable Parameter API Nomenclature" on page 4-17

4.1 Introduction

The DRC module enables Dynamic Range Compression of the receive (Rx) path signal. DRC provides refined speaker volume level control by defining a configurable signal gain that can depend on the Rx input level and spectrum. DRC is intended primarily for speakerphone mode and has multiple purposes:

- Attenuates low-level noise and high-level speech
- Amplifies mid-level weak speech
- Decreases distortion in the acoustic output of the speaker
- Increases full-duplex performance by limiting the magnitude of acoustic echo and nonlinear distortion in the echo path
- Increases the acoustic power of the speaker. Unacceptable speaker distortion often occurs at high levels for a specific spectral band. DRC can preferentially limit a spectral band and increase loudness using the rest of the spectrum.

DRC consists of a compressor and limiter that are in series. The DRC full-band compressor and limiter are upgraded versions of the AER HLC and AER Rx SLIM, respectively. When DRC is activated, HLC and Rx SLIM should be disabled or inactive.

Figure 4-1 shows the location of the DRC module in the Rx path.

Figure 4-1 Location of the DRC Module in the IP Phone Rx Path

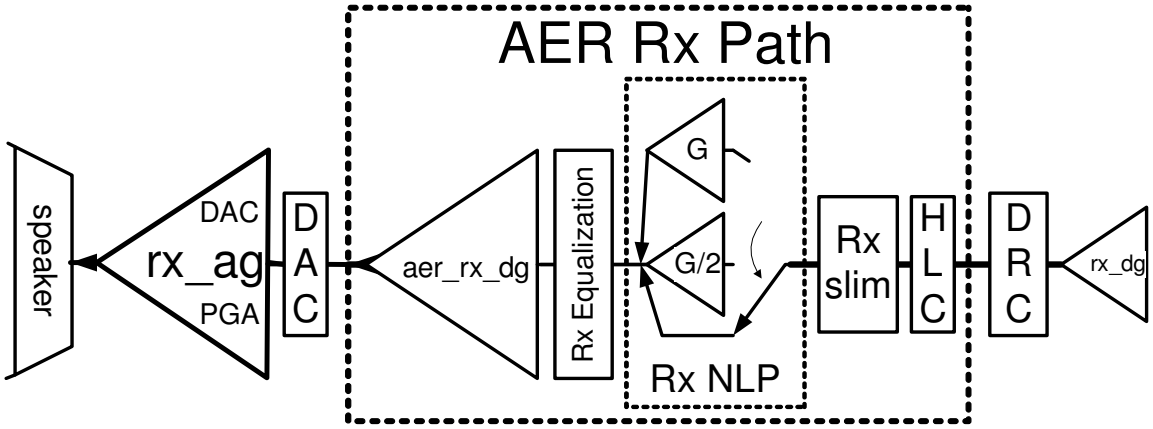
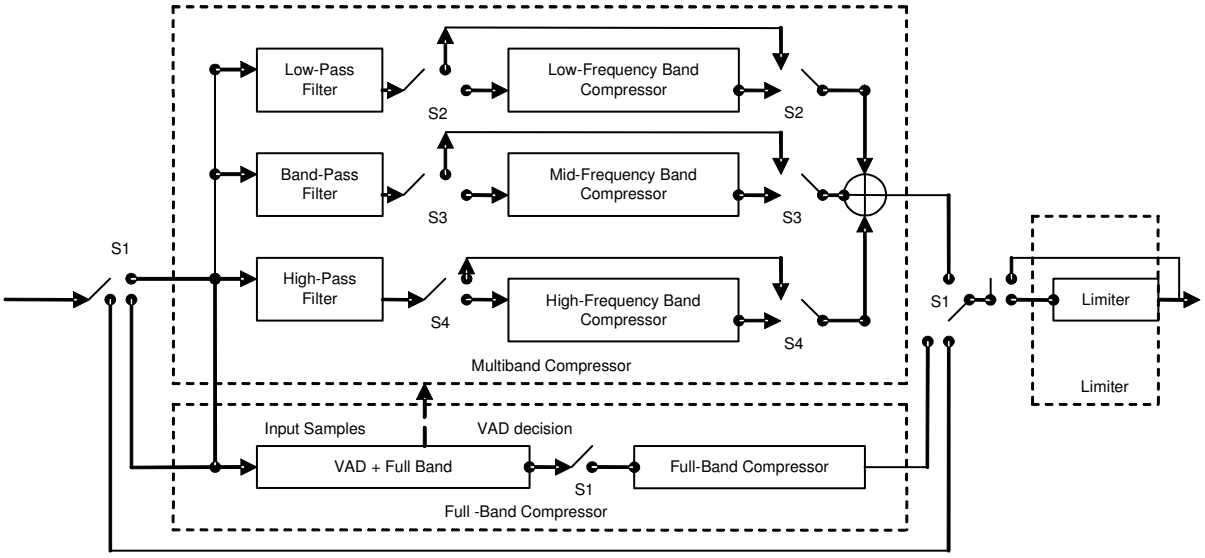


Figure 4-2 shows the DRC implementation.

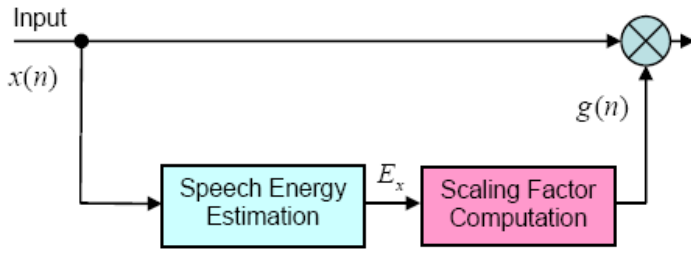
Figure 4-2 DRC signal processing schematic



Each compressor and the limiter determine and apply a time-dependent linear gain. The DRC compressor can operate in full or multiband mode. Full band is simpler to configure and recommended if there is no compelling need for multiband. Multiband might limit low-frequency echo nonlinearity and thus enhance full-duplex performance. For multiband, configurable filters separate the input signal into three spectral bands. The compressor for each spectral band operates independently, with the exception of relying on the full-band VAD. If the VAD is enabled and determines speech is not present, the gain of each compressor remains constant. This avoids gain changes during pauses in speech. The limiter is not affected by the VAD.

Figure 4-3 shows how a compressor or limiter gain, g(n), is updated for every sample n.

Figure 4-3 Compressor Structure



The limiter output has a configurable delay of *d* milliseconds. For a compressor, d=0. The output may be delayed, but the most recent input sample, x(n), affects both the power estimation and gain processor's computation. Thus the limiter's g(n) computation "looks ahead" *d* milliseconds to alter g(n) in the direction that anticipates and accommodates future input samples. The look-ahead delay can compensate for the delay required for g(n) to respond to a change in input signal power. This g(n) adaptation delay is determined by the configurable time constants of the power estimation and gain processor.

A difference between the compressors and limiter is the DRC configurable time constant nomenclature. The compressor can increase signal to greater than unity and is thought of as applying *gain*, whereas the limiter restrains large signals by applying *attenuation*. For any DRC power estimate and *compressor* gain processor computation, an attack time refers to the time required for this exponentially time-averaged estimate to *increase* by a factor of e=2.718. For the *limiter* gain processor computation, g(n), an attack time refers to the time for this gain to *decrease* by 1/e, as the attenuation increases. This DRC limiter convention is the same for the AER HLC, which is also thought of as applying attenuation. The release time governs the rate of magnitude changes in the opposite direction with respect to the attack time.

4.2 DRC Functionality

A Matlab DRC tool is available to help determine and configure parameter values. (See the *Main Menu* of the document set and click *Supporting Documents* > *Miscellaneous Documents*.) The Matlab DRC tool provides gain plots, multiband digital filter coefficients, parameter ranges, an output file with all DRC configurable parameters, and generates DRC output audio files for test input files for any set of DRC parameters. Figure 4-4 and Figure 4-5 were generated by the Matlab DRC tool.

Figure 4-4 gives the transfer function for an exemplary multiband case.

Figure 4-4 Example of DRC Multiband Low (Red), Mid (Green), and High (Blue) Filters

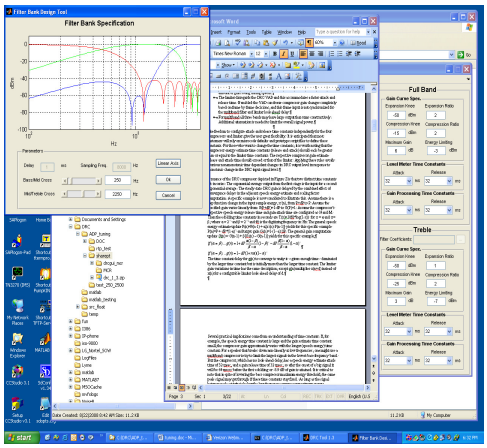
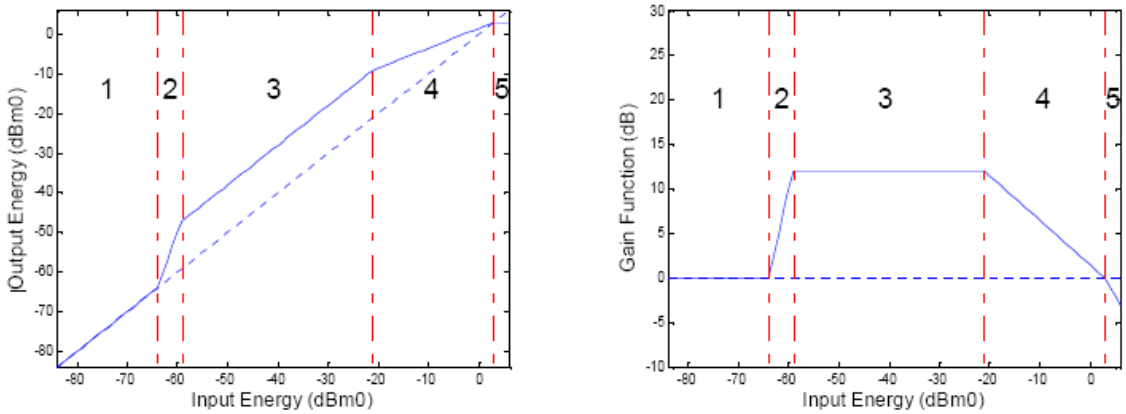


Figure 4-5 shows a specific choice for a compressor gain target function.

Figure 4-5 Compressor Input Versus Output and Gain Target $G_t = G_t(X)$.



The limiter gain target can only have a unity gain region and energy limit, analogous to regions 1 and 5. Region 1 is unity gain, region 2 is expansion, region 3 is a maximum gain, region 4 is compression, and region 5 is a limiting output level. The configurable input power level (x-axis) of the expansion and compression “knee points” define the left boundary of the respective regions 2 and 4.

All DRC gain changes yield speech distortion, but this disadvantage is often outweighed by DRC benefits. The expansion region (2) can function as a noise block. DRC can use expansion to raise low-level far-end speech and make it easier to hear, but remove this amplification for lower level background noise. When a call is connected between two speakerphones at high volume, sometimes low-level "whistling" is heard, due to feedback instability. A receive-path noise block can remove this whistling problem. Changes in the maximum gain (3) can vary speech loudness and facilitate different receiver volume levels. The compression region (4) can limit output, but still allow some audible indication of level changes in loud speech. Compression and limiting (5) can avoid a regime where the acoustic echo (with AER disabled) is unacceptably loud and/or nonlinear. Limiting can also avoid flat-top saturation in the Rx path, given knowledge of the AER Rx digital gain and maximum AER Rx Equalizer gain.

4.3 Configuring a Specific Set of DRC Parameters

The DRC configurable parameters which normally do not vary with volume level are

- 20 time constants. Attack and release times exist for both the power estimation and gain computation. This yields four parameters that are independently defined for five DRC entities:
  - Limiter
  - Full-band compressor
  - Three multiband compressors
- Look-ahead delay in milliseconds for the limiter
- Three symmetric FIR filter coefficients. These filters define the multiband's low, mid, and high bands. These filters have a one-millisecond delay.

The DRC configurable parameters which normally vary with volume level are

- DRC configuration bitfield
- Bit 1 = full band on
- Bit 2 = multiband low band on
- Bit 3 = multiband mid band on
- Bit 4 = multiband high band on
- Bit 5 = VAD enabled
- Limiter energy limit
- 24 parameters defining the target gain function. There are six parameters for each of four compressors. The six parameters include an energy limit, maximum gain, and ratios and knee points for both expansion and compression.

The DRC Matlab tool approximates time constants by $T(x) = -1/(fd \cdot \log(1-2^{-N}))\ 2^N/fd$. DRC is configured using time constants in logarithmic form, $N = \log_2[fd \cdot T(x)]$ where N must be an integer. The Matlab DRC tool's "generate" button writes all the Matlab DRC configurable parameters to an output file, which includes logarithmic conversions for time constants. The output file also converts expansion and compression ratios into the correct integer form for configuring DRC, by multiplying these ratios by 16. To obtain syntax for commands to configure DRC parameters given by the Matlab tool output file, it is advisable to rely on online syntax for TI IPP software, script files, or a DRC module header file. Table 4-3 on page 4-18 through Table 4-5 on page 4-23 are intended for reference only, if the DRC configurable parameter syntax needs clarification.

NMM commands to configure collective sets of DRC/AER parameters are listed below.

```
set ipp_gains <prof>  num_rows <num_rows>
set ipp_gains <prof>  default_row <row>
set ipp_gains <prof>  values <row> <rx_ag > <aer_rx_dg> <aer_nlp_clip>
<aer_nlp_clit> <aer_tx_dg> <tx_ag> [aer_hlc_target_level ([-96,6] or 0x7fff
(unspecified))] [CNG RX level ([0,-90] or 0x7fff (unspecified))dBm0] [CNG TX level
([0,-90] or 0x7fff (unspecified)) dBm0]Row elements
set ipp_drc_gains <prof> values <row> <cfgbits> <thresh_dBm>
<exp_knee_l> <com_ratio_l> <exp_ratio_l> <max_amp_l> <com_knee_l> <energy_lim_l>

    (Full/Low band parameters)
<exp_knee_m> <com_ratio_m> <exp_ratio_m> <max_amp_m> <com_knee_m> <energy_lim_m>
```

```
    (Mid band parameters)
<exp_knee_h> <com_ratio_h> <exp_ratio_h> <max_amp_h> <com_knee_h> <energy_lim_h>
    (High band parameters)
Note: If cfgbits specifies FULL BAND, only the first set of parameters is required
```

These NMM commands can define gain tables in which some DRC and AER parameters are customized for each speaker volume level. This creates two values for one parameter, with one value stored in a gain table as a function of speakerphone volume level and another value stored in an ipp_profile. Two commands remove this ambiguity:

```
set ipp_prof <prof> drc_gain_src <ipp_prof|gain_table>
set ipp_prof <prof> aer_gain_src <ipp_prof|gain_table>
```

If drc_gain_src is set to gain_table, values given by `show ipp_drc_gains <prof>` are used. If aer_gain_src is set to gain_table, values given by `show ipp_gains <prof>` are used. If these commands are set to ipp_prof, the corresponding gain table's parameter values are ignored and the ipp_prof values are used.

NMM commands control whether additional sets of DRC parameters stored in ipp_prof are ignored or used. For example, the user may decide *not* to use multiband mode under any circumstances. In this case, the microcode need not be bogged down with the task of sending multiband parameters to the DSP DRC module. The following NMM command sequence will send all the parameters in Table 4-2 on page 4-17 to all compressors:

```
set ipp_prof <prof> drc_band none
set ipp_prof <prof> drc_band lband
set ipp_prof <prof> drc_band mband
set ipp_prof <prof> drc_band hband
set ipp_prof <prof> drc_band_fband
```

Omitting one NMM command line above for a specific compressor results in Table 4-2 data for that compressor NOT being sent to the DSP. This omission is only certain because of the leading "none" option, which removes whatever value was previously assigned to drc_band. Similarly the following commands determine whether multiband FIR filter coefficients are ignored or sent to the DSP:

```
set ipp_prof <prof> drc_fir_band none
set ipp_prof <prof> drc_fir_band lband
set ipp_prof <prof> drc_fir_band mband
set ipp_prof <prof> drc_fir_band hband
```

4.4 Choosing Values for the DRC Configurable Parameters

The following material gives suggestions for determining a preliminary set of DRC configurable parameters. DRC users are encouraged to further customize and optimize for any particular product.

Step 1—Use the Matlab tool to define DRC parameters for the case of nominal volume. Use the full-band compressor and default values for all time constants.

Step 2—Place volume-dependent parameters in a spreadsheet, similar to Table 4-1 on page 4-10.

- Understand how the exemplary spreadsheet reveals consistent application of a
  - Total Rx path maximum output level threshold constraint
  - Total Rx path gain that sequentially increases at each volume level for signal levels just below the range where compression is applied
- Set the limiter signal output energy limit to match the highest energy limit among active compressors.

Determining many DRC parameters for every available speakerphone volume level can be difficult. Use an exemplary spreadsheet as given in Table 4-1. Note that the order of the columns facilitates cutting and pasting into a script file with the NMM commands:

```
set ipp_gains <prof> values <row> …
set ipp_drc_gains <prof> values <row> …
```

Step 3—Above the nominal volume (see Table 4-1, row 6) increase the DRC max gain by up to 9 dB to define volume levels up to 9 dB above nominal. To define higher volume levels, increase the analog DACPGA gain. Use an exemplary script file of NMM commands for DRC configuration provided by Texas Instruments.

Step 4—Determine the maximum targeted value for the “thresh sum” column. Every volume level should stay consistently below the “thresh sum” limitation. Because the echo can limit full-duplex performance, it is advisable to measure the speakerphone echo magnitude and nonlinearity as a function of Rx input signal levels and frequency with AER disabled. Regions of unacceptably loud echo or unacceptably nonlinear echo are eliminated by choosing the correct “thresh sum” limitation. Ideal targets are ERL 6 dB and ERL+THD 30 dB. For a high-quality speakerphone, acceptable Rx loudness can be achieved for even higher echo performance targets, but for a lower quality enclosure these targets (and full-duplex performance) may need to be sacrificed to achieve adequate loudness.

Step 5—Experiment with multiband for high volume cases. The material in “DRC Time Constants” on page 4-12 and “Multiband Compression Considerations” on page 4-16 is relevant to using multiband to reduce DRC low-frequency output. Frequently, speakerphone echo (with AER disabled) is more nonlinear for low frequencies.

Table 4-1 shows an example spreadsheet for determining DRC gain table parameters.

Table 4-1 Spreadsheet Example to Help Determine DRC Gain Table Parameters

	AIC	AER	max	Thresh		DRC	DRC	DRC	DRC	DRC	DRC	DRC	DRC	DRC	DRC	HLC	HLC_max_sig
	rx_ag	rx_dg	gsum	sum		cfg	lim	exp_k	com_r	exp_r	maxg	com_k	lim	lim	com_k	thresh	equivalent
	.5dB	.5 dB				bits	thresh		*16	*16		fb/lo	fb/lo	mid/hi	mid/hi	equivalent	in dB
0	-51	0	-25.5	-25.5	0	0x31	0	-50	64	32	0	-21	0			0	-22.5
1	-45	0	-22.5	-22.5	1	0x31	0	-50	64	32	0	-21	0			0	-19.5
2	-39	0	-19.5	-19.5	2	0x31	0	-50	64	32	0	-21	0			0	-16.5
3	-33	0	-16.5	-16.5	3	0x31	0	-50	64	32	0	-21	0			0	-13.5
4	-27	0	-13.5	-13.5	4	0x31	0	-50	64	32	0	-21	0			0	-10.5
5	-21	0	-10.5	-11.5	5	0x31	-1	-50	64	32	0	-21	-1			-2	-8.5
6	-15	0	-7.5	-11.5	6	0x31	-4	-50	20	32	0	-7	-4			-8	-8.5
7	-15	0	-4.5	-11.5	7	0x31	-4	-50	20	32	3	-10	-4			-8	-8.5
8	-15	0	-1.5	-11.5	8	0x31	-4	-50	29	32	6	-13	-4			-8	-8.5
9	-15	0	1.5	-11.5	9	0x31	-4	-50	32	32	9	-16	-4			-8	-8.5
10	-9	0	4.5	-11.5	10	0x31	-7	-50	25	32	9	-19	-7			-14	-8.5
11	-21	0	7.5	-11.5	11	0x3E	-1	-78	64	32	18	-28	-10	-1	-21	-2	-8.5
12	-3	0	7.5	-11.5	12	0x31	-10	-50	20	32	9	-22	-10			-20	-8.5

The second and third columns give rx_ag and aer_rx_dg. The next column sums the gains (rx_ag+aer_rx_dg)/2 + “maxg”. The resulting “max gsum” column verifies that the overall Rx gain uniformly increases for higher volume levels. The next column measures the maximum possible steady state analog sine wave signal power reaching the speaker in “dBm0” (when the Rx EQ is disabled) defined relative to the case where rx_ag = 0 dB and Rx input is 0 dBm0. This computation requires the most restrictive (minimum) energy limit for the compressor and limiter. The compressor energy limit is either the full-band energy limit or maximal multiband energy limit, depending on which is active. Note the resulting “thresh sum” column shows that the maximum possible Rx path analog output signal does not change for volume levels 5 through 12. This reflects the intention to consistently restrict the maximum echo and avoid a nonlinear region. The “cfgbit” column indicates the full-band compressor is used for all volume levels except row 11, which uses multiband.

A goal of this spreadsheet was to have the limiter reassert its energy limit at the maximum possible steady state level that any active compressor could output. The “lim thresh” column values were calculated to achieve this goal. At each gain level, the inequality com_knee energy_lim – max_gain is enforced by lowering com_knee, if needed. Violating this inequality yields a com_knee value that lies within gain region 5 of Figure 4-5 on page 4-5 and is thus misleading. The last two columns derive HLC values that would be used if DRC was disabled and HLC enabled with the goal of recovering the “thresh sum” column.

Gains and DRC parameters from Table 4-1 were pasted into the script file excerpt below.

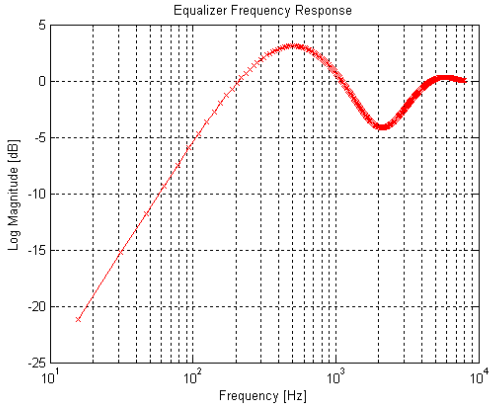
```
set ipp_gains 7 num_rows    13
set ipp_gains 7 default_row  6
set ipp_gains 7 values  0 -51 0 7 700 18 69
set ipp_gains 7 values  1 -45 0 7 700 18 69
```

```
set ipp_gains 7 values  2 -39 0 7 700 18 69
set ipp_gains 7 values  3 -33 0 7 700 18 69
set ipp_gains 7 values  4 -27 0 7 700 18 69
set ipp_gains 7 values  5 -21 0 7 700 18 69
set ipp_gains 7 values  6 -15 0 7 700 18 69
set ipp_gains 7 values  7 -15 0 7 700 18 69
set ipp_gains 7 values  8 -15 0 7 700 18 69
set ipp_gains 7 values  9 -15 0 7 700 18 69
set ipp_gains 7 values 10  -9 0 8 700 18 69
set ipp_gains 7 values 11 -21 0 9 500 18 69
set ipp_gains 7 values 12  -3 0 9 500 18 69
set ipp_drc_gains 7 values  0 0x31   0 -50 64 32  0 -21   0
set ipp_drc_gains 7 values  1 0x31   0 -50 64 32  0 -21   0
set ipp_drc_gains 7 values  2 0x31   0 -50 64 32  0 -21   0
set ipp_drc_gains 7 values  3 0x31   0 -50 64 32  0 -21   0
set ipp_drc_gains 7 values  4 0x31   0 -50 64 32  0 -21   0
set ipp_drc_gains 7 values  5 0x31  -1 -50 64 32  0 -21  -1
set ipp_drc_gains 7 values  6 0x31  -4 -50 20 32  0  -7  -4
set ipp_drc_gains 7 values  7 0x31  -4 -50 20 32  3 -10  -4
set ipp_drc_gains 7 values  8 0x31  -4 -50 29 32  6 -13  -4
set ipp_drc_gains 7 values  9 0x31  -4 -50 32 32  9 -16  -4
set ipp_drc_gains 7 values 10 0x31  -7 -50 25 32  9 -19  -7
set ipp_drc_gains 7 values 11 0x3E  -1 -78 64 32 18 -28 -10 -78 64 32 18 -28 -1 -78
64 32 18 -28 -1
set ipp_drc_gains 7 values 12 0x31 -10 -50 20 32  9 -22 -10
```

Note that in lieu of the spreadsheet, merely looking at the script file excerpt above does *not* make it clear that the each row's loudness increases monotonically, or that the maximum Rx analog output is fixed in the volume levels defined by rows 5 through 12.

The "thresh sum" column indicates the largest limiter energy limit is 0 dBm0, and not 3 dBm0 (full-scale sine wave). At the lowest volume levels, removal of the top of the Rx input digital dynamic range is not needed to reduce echo or echo nonlinearity. For the exemplary case shown here, this decision was made to prevent steady-state flat-top saturation resulting from a maximal 3 dB Rx Equalizer gain, show in Figure 4-6.

Figure 4-6 Exemplary Rx Equalizer Gain



4.5 DRC Time Constants

Most DRC users should rely on the TI-provided Matlab DRC tool defaults, microcode and DSP DRC initialization defaults, or prototype script files to define these time constants.

Note that a fast limiter attack time (one sample) does not ensure attenuation of isolated input peaks when the limiter release time is also fast. A small limiter release time may allow the limiter attenuation to be removed before the large sample arrives *d* samples later. Because the limiter has the final DRC gain, the limiter's power estimate and gain processor attack times should be less than or equal to that of the compressor. The limiter needs enough speed to cope with upstream DRC compressor gain changes to limit the DRC output level more rigorously.

The multiband compressor achieves spectral separation with symmetric FIR filters that add one millisecond of signal delay. The full-band compressor adds no signal delay. The limiter's configurable look-ahead delay is 0 to 5 milliseconds. The limiter has the sole function of limiting the final DRC output to a configured maximum level. This capability is one of the many functions that each (upstream) compressor can do.

There are several reasons why the limiter function is not redundant or superfluous:

- For the compressor, the configurable attack and release time constants apply equally over five distinct ranges of the input power estimate. Speech distortion resulting from fast attack and decay times is often only justified in the "fifth" region, which limits total output power for *high* input signals. The limiter is designed only for limiting high input signals. Thus by configuring and relying on the limiter to quickly attenuate transiently large input signals, one can slow down DRC compressor gain variations and consequently reduces speech distortion for *low* input signals.
- The look-ahead delay of the limiter compensates the limiter's gain processor adaptation delay, facilitating attenuation of a transiently large signal more quickly with less voice distortion.
- The limiter disregards the DRC VAD and this accommodates a faster attack and release time. If enabled, the VAD can freeze compressor gain changes completely based on frame-by-frame decisions, and this frame input is not synchronized for the multiband filter delay and limiter look-ahead delay.
- For multiband, all three bands may have large output that will sometimes sum constructively. Additional attenuation is needed to limit the overall signal power.

The attack and release time constants are independently configurable for the four compressors and limiter. Note that the full-band power estimate not only provides input to the full-band gain processor, but also provides input for the full-band VAD. Thus full-band power estimate time constants affect multiband DRC output when the VAD is enabled.

A nuance of the DRC compressor (Figure 4-3 on page 4-3) is that two exponential averages over time occur in series. The steady state DRC gain, g(n), is delayed by the combined effect of convergence delays in the adjacent power estimation and gain processor computation.

The following example shows how the distinct configurable time constants have a combined affect. Assume the following:

- The input sample energy changes from $x^2(n) = P + dP$ for n0 to $x^2(n) = P$ for n>0 where n gives the sample number.
- The specified gain target varies linearly from $G_t(P+dP) = 1 - dP$ to $G_t(P) = 1$.
- The compressor's respective power estimation release time and gain processor attack time are configured as N and M.
- The e-folding time constants in seconds are $T(x) = 1/(fd \cdot \log(1-x))$ for x = and x =, where = 2^{-N} and = 2^{-M} and fd is the digitizing frequency in Hz.

The general power estimate and gain processor updates:
$X(n) = X(n-1) + [x^2(n) - X(n-1)]$ $G(n) = G(n-1) + [G_t(n) - G(n-1)]$
for this example yield $X(n)=P + (1-)^n dP$, target gain $G_t(n) = 1 - (1-)^n dP$ and gain:

$$if\ (\alpha \neq \beta) \ldots g(n) = 1 + dP\left[\alpha(1-\beta)^{n+1} - \beta(1-\alpha)^{n+1}\right]/(\beta - \alpha)$$
$$if\ (\alpha = \beta) \ldots g(n) = 1 - dP(1 + n\alpha)(1-\alpha)^n$$

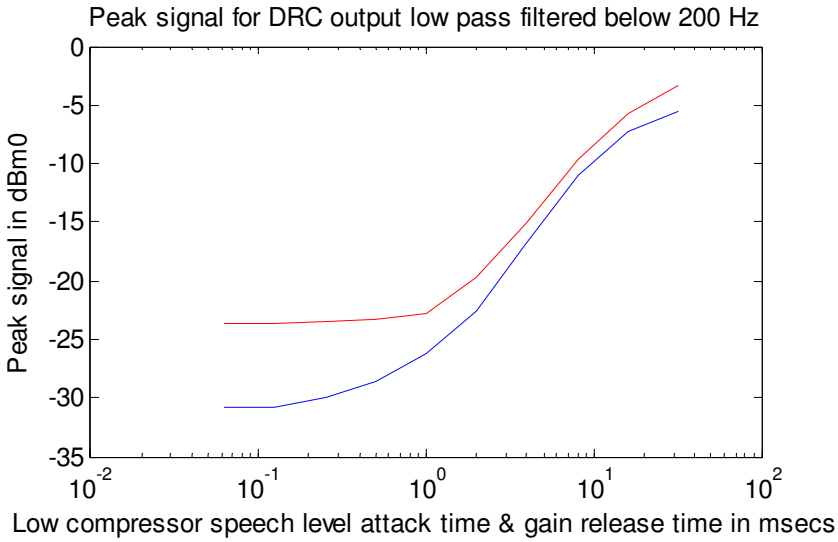
The time constant delay for g(n) to converge to unity is—given enough time—approximately the larger time constant (of the power estimate and gain processor) but is initially more than the larger time constant. The limiter's gain variations in time have the same behavior, but occur *d* samples sooner.

Several practical implications come from an understanding of time constants. As shown mathematically in the example above, if the power estimate and gain processor time constants vary, the gain approximately varies with the longer time constant. For a speaker that breaks down nonlinearly at low frequencies, a multiband compressor could be used to try to limit the largest signals in the lowest bass frequency band. Assume the compressor has a power estimate attack time of 32 milliseconds and a gain processor release time of 32 milliseconds, so after the onset of a big signal it will be 64 milliseconds before achieving the first e-folding or 8.9 dB of bass attenuation. A further reduction of the bass compressor maximum energy limit typically has a **negligible** effect on the signal for the first 64 milliseconds. An initial input signal power surge will have the same compressor output power, *regardless* of how much the compressor's maximum energy limit parameter is reduced. Lowering this parameter primarily extends the time period over which the compressor gain can continue to decrease, but does not affect compressor output for the initial transient. To reduce the maximum bass spectral output, it is advisable to reduce the time constants of the bass compressor's power estimate attack time and gain processor computation release time. Reducing only the bass compressor's maximum energy limit can be *inadequate*.

For an exemplary high signal level speech input test file, the multiband cutoffs were set to 250 and 2250 Hz. The limiter energy threshold was set to +3 dBm0 to effectively disable the DRC limiter. All time constants were set to 32 milliseconds, except the low compressor's signal level attack time constant and gain release time constant. These were set to the same value which varied from 32 milliseconds to 0.125 seconds. Multiband low, mid, and high energy limits were respectively set to -30, -24, -24 dBm0. After processing the output file through a sharp low-pass filter with a 200 Hz cutoff, the peak signal that resulted is shown by the upper curve in Figure 4-7. For low bass time

constants, a -24 dBm0 peak signal limit is enforced rather than -30 dBm0. This is due to leakage through the middle band. The lower curve resulted from repeating these tests but lowering the middle band energy limit to -35 dBm0 and setting the middle band power estimate attack and gain computation release time constants to the minimum allowed values. This removed leakage into the middle band and the anticipated -30 dBm0 bass limit was recovered for sufficiently small low-band time constants.

Figure 4-7 Multiband DRC Peak Output Versus Low-Band Compressor Time Constants



This result shows that using DRC to preferentially limit low frequency peak signals requires considering the effects of both DRC time constants and leakage. “Multiband Compression Considerations” on page 4-16 addresses leakage further. Leakage can be reduced by scaling the DRC multiband compressor energy thresholds and maximum gains as high as possible, and keeping the desired total Rx path loudness and power threshold by lowering other Rx path gains downstream of DRC. This was done in Table 4-1, row 11, where the Rx analog gain was reduced to –21/2 dB and the DRC maximum gain was increased to 18 dB. Row 11’s mid and high energy thresholds were scaled to –1 dBm0, which is considerably higher than the full-band energy thresholds for volume levels of adjacent rows (–7 and –10 dBm0).

Thus Table 4-1, Row 11 illustrates how to achieve additional bass compression (relative to higher frequencies) by scaling up the multiband compressor energy limits. To obtain performance reasonably “down” the blue curve in Figure 4-7, low-band power estimate attack and gain processor release time constants of roughly 0.5 milliseconds are recommended. Explicit script file commands that customize these low-band time constants for this purpose are given below. A 16 kHz sampling rate is assumed, for which all time compressor defaults are 9. The low-band compressor’s signal power estimate attack time and gain processor release time are set to 3, corresponding to a time constant of $2^3/fd = 8/16000 = 0.5$ millisecond. At 8 kHz sampling, 8 is the default and 2 is analogously recommended for the power estimate attack and gain processor release to preferentially suppress multiband compressor output for one spectral band.

```
set ipp_prof 7 drc_com_tc_log2_power fband 9 9
set ipp_prof 7 drc_com_tc_log2_power lband 9 3
set ipp_prof 7 drc_com_tc_log2_power mband 9 9
set ipp_prof 7 drc_com_tc_log2_power hband 9 9
set ipp_prof 7 drc_com_tc_log2_gain    fband 9 9
set ipp_prof 7 drc_com_tc_log2_gain     lband 3 9
set ipp_prof 7 drc_com_tc_log2_gain   mband 9 9
set ipp_prof 7 drc_com_tc_log2_gain    hband 9 9
```

Suppose the goal is to slow down full-band compressor gain changes. One might think that it is arbitrary to either increase the power estimate time constant and/or the gain processor time constant. The power estimate time constant should be changed only when this results in a more accurate measurement of signal power. The speed and accuracy of DRC VAD decisions depends on the DRC full-band power estimate. If the target gain (y-axis) changes a lot for a small change in the power estimate (x-axis), increasing the power estimate time constant does not ensure slower gain changes (see Figure 4-5 on page 4-5). So the goal of slowing down gain changes is best achieved by increasing the gain processor time constants.

Note that for a DRC power estimation, if the attack time exceeds the release time there is a positive bias that yields a larger power estimate. Conversely, a release time that exceeds the attack time yields a smaller power estimate. These biases affect the DRC gains because the gain computation requires a power estimate for input.

4.6 Multiband Compression Considerations

The effect of multiband compression can create objectionable artifacts. This is inherent in all multiband compressors. Additionally, the multiband filters are imperfect and have leakage that must be taken into account. Full-band compression is advised unless there is a clear need for multiband compression. Multiband should be used exclusively if there is a pressing need for compression that includes spectral discrimination.

If multiband is configured for expansion for low-level speech signals, the amount of expansion gain will be different for all multiband compressors, which independently compute an in-band power estimate and gain. A different gain in the three bands for *high* level input signals may be desired. However, one may wish to eliminate compressor expansion to avoid spectral alteration for *low* level input. Low-level speech distortion is typically not desired or justified, as receiver nonlinear distortion generally occurs at high signal levels. Often it is desirable to enable the VAD so the DRC gain applied to speech does not change much during pauses in speech. But when multiband's three spectral gains change only during speech, there is a perceptible altered background noise spectrum between speech that persists indefinitely until speech resumes. The typical multiband manifestation of this problem is DRC high-band "drop-outs" that intermittently result after far-end speech. The background noise spectrum often varies like 1/f, so typically the high-band compressor's energy estimate falls to near the expansions knee point and the high-band gain is reduced. Thus lowering the multiband high-band compression knee point may alleviate this issue. Lowering every band's compression knee point also alleviates this issue, but this removes DRC expansion capability.

Another issue for multiband is referred to as "leakage". If DRC is configured to limit bass (*i.e.*, low band) output to –3 dBm0 and + 3 dBm0 above bass, DRC performs as expected. Then assume DRC is configured to keep this relative 6 dB reduction in bass, but also lower the overall maximum output. Suppose the energy limits are set to -20 dBm0 for bass, and -14 dBm0 for the mid and high DRC compressor bands. Assume the DRC multiband bass filter has a 250 Hz bass/mid cutoff frequency. A +3 dBm0 200 Hz tone input will appear in the bass compressor output at -20 dBm0 as expected. Assume the DRC mid-band filter is -19 dB at 200 Hz, as shown in Figure 4-4. An unwanted 3 – 19 = –16 dBm0 200 Hz tone will "leak" into the mid-band compressor. This leakage is below the mid energy limit of -14 dBm0, so the mid-band compressor applies no attenuation. The mid-band outputs a –16 dBm0 200 Hz tone that is in phase with the low-band output. Constructive interference yields a **–11.75 dBm0** 200 Hz tone in the multiband output. For this configuration, a tone input to the mid filter's band pass at 1250 Hz and 3 dBm0 has negligible leakage into other bands and yields **–14 dBm0** multiband compressor output, as expected. So an input full-scale bass tone yields DRC output that is 2.25 dB *larger* than a mid-band tone, in spite of the intention to reduce the bass by 6 dB. This example illustrates that the multiband spectral filters often have inadequate resolution for low energy limits. Note that Table 4-1 row 11 (multiband) lowers other gains to scale the multiband energy limits higher to alleviate this leakage issue. But this adjustment involves increasing the maximum DRC gain and this makes the aforementioned high-band "drop-outs" issue more prominent.

4.7 DSP DRC Configurable Parameter API Nomenclature

The DRC configurable parameters are assigned clearly understood names by the DRC Matlab tool. However, in other contexts their names may be more cryptic. This section cross-references the names given to a DRC configurable parameter in seven different contexts. The table header gives abbreviations for these seven different contexts.

Table 4-2 DSP DRC Configurable Parameter API Nomenclature

Context	API/Command Syntax	Description
C	drcControl(…)	API for the DRC component module which appears in the drc.h header file.
D	dsp <tcid> …	An NMM command of this form can change DRC parameter values in the middle of a call.
Se	set ipp_prof <prof>…	An NMM command of this form can set the parameter value in permanent flash memory for a specific IP-phone profile.
Seg	set ipp_drc_gains <prof>…	An NMM command of this form can set the parameter value in permanent flash memory for a specific IP-phone profile and volume level.
Sh	show ipp_prof <prof>…	NMM Show command that displays some DRC parameter values
Shg	show ipp_drc_gains <prof>…	NMM, displays some DRC parameter values
Shp	show ipp_drc_params <prof>…	NMM, displays some DRC parameter values

Table 4-3 DRC Limiter Configurable Parameter Nomenclature (Part 1 of 2)

C = drcControl() Sh = show ipp_prof <prof> D = dsp <tcid> Shg = show ipp_drc_gains <prof> Se = set ipp_prof <prof> Shp = show ipp_drc_params <prof> Seg = set ipp_drc_gains <prof> <param> <row> and set ipp_drc_gains <prof> values <row>		Description
C	ctl_code = drc_CTL_SET_LIM limiter.valid_bitfield = 2^(drc_LIM_CFG_VALID_LEV_TC_BIT) limiter.tc_log2 bits 0-3	This parameter is always zero for DRC and other requested values ignored. Sets Limiter Power Estimation Attack Time Constant to $1/(fd \cdot \log(1-2^{-N}))$
D	drc_limiter_cfg tc_log2_lev_attack	
Se	drc_lim_tc_log2_power Limiter Power attack TC in log2	
Sh	drc_lim_tc_log2_power bits 0-3	
Shp	lim Power Attack log2	
C	ctl_code = drc_CTL_SET_LIM limiter.valid_bitfield = 2^(drc_LIM_CFG_VALID_LEV_TC_BIT) limiter.tc_log2 bits 4-7	Sets Limiter Power Estimation Release Time Constant to $1/(fd \cdot \log(1-2^{-N}))$ Default N=8 @fd=8 kHz N=9 @fd=16 kHz Range = [0,15]
D	drc_limiter_cfg tc_log2_lev_release	
Se	drc_lim_tc_log2_power Limiter Power release TC in log2	
Sh	drc_lim_tc_log2_power bits 4-7	
Shp	lim Power Release log2	
C	ctl_code = drc_CTL_SET_LIM limiter.valid_bitfield = 2^(drc_LIM_CFG_VALID_GAIN_TC_BIT) limiter.tc_log2 bits 8-11	Sets Limiter Gain Processor Attack Time Constant to $1/(fd \cdot \log(1-2^{-N}))$ Default N=1 @fd=8 kHz N=2 @fd=16 kHz Range = [0,15]
D	drc_limiter_cfg tc_log2_gain_attack	
Se	drc_lim_tc_log2_gain Limiter Gain attack TC log2	
Sh	drc_lim_tc_log2_gain bits 0-3	
Shp	lim Gain Attack log2	
C	ctl_code = drc_CTL_SET_LIM limiter.valid_bitfield = 2^(drc_LIM_CFG_VALID_GAIN_TC_BIT) limiter.tc_log2 bits 12-15	Sets Limiter Gain Processor Release Time Constant to $1/(fd \cdot \log(1-2^{-N}))$ Default N=9 @fd=8 kHz N=10 @fd=16 kHz Range = [0,15]
D	drc_limiter_cfg tc_log2_gain_release	
Se	drc_lim_tc_log2_gain Limiter Gain release TC log2	
Sh	drc_lim_tc_log2_gain bits 4-7	
Shp	lim Gain Release log2	
C	ctl_code = drc_CTL_SET_LIM limiter.valid_bitfield = 2^(drc_LIM_CFG_VALID_THRESH_BIT) limiter.thresh_dBm	Sets Limiter Threshold Energy Level in dBm0 Used to limit output level Default = -50 Range = [-78,+3]
D	drc_limiter_cfg thresh_dBm	
Se	drc_lim_thresh	
Seg	thresh_dBm	
Sh	drc_lim_thresh	
Shg	lim_thresh	
Shp	lim Limiter Threshold (dBm0)	

Table 4-3 DRC Limiter Configurable Parameter Nomenclature (Part 2 of 2)

C = drcControl() **Sh = show ipp_prof <prof>** **D = dsp <tcid>** **Shg = show ipp_drc_gains <prof>** **Se = set ipp_prof <prof>** **Shp = show ipp_drc_params <prof>** **Seg = set ipp_drc_gains <prof> <param> <row> and set ipp_drc_gains <prof> values <row>**		**Description**
C	ctl_code= drc_CTL_SET_LIM limiter.valid_bitfield = 2^(drc_LIM_CFG_VALID_DELAY_BIT) limiter.delay_len (in samples)	Sets Limiter "Look Ahead" Delay Time Default = 1 ms Range = [0,5] ms Convert to samples for drcControl()
D	drc_limiter_cfg delay_ms (in ms)	
Se	drc_delay_ms (in ms)	
Sh	drc_delay_ms (in ms)	
Shp	lim Current Delay (ms) (in ms)	
End of Table 4-3		

Table 4-4 Configurable Parameters for Full and Multiband Compressor (Part 1 of 3)

C = drcControl() Sh = show ipp_prof <prof> D = dsp <tcid> Shg = show ipp_drc_gains <prof> Se = set ipp_prof <prof> Shp = show ipp_drc_params <prof> Seg = set ipp_drc_gains <prof> <param> <row> and set ipp_drc_gains <prof> values <row>		Description
C	ctl_code = drc_CTL_SET_XBAND for X = F, L, M, H band.valid_bitfield = 2^(drc_BAND_CFG_VALID_LEV_TC_BIT) band.tc_log2 bits 0-3	Sets Compressor Power Estimation Attack Time Constant to $1/(fd \cdot \log(1-2^{-N}))$. Default 8 @fd=8kHz 9 @fd=16 kHz Range=[0,15]
D	drc_band_cfg tc_log2_lev_attack	
Se	drc_com_tc_log2_power Compressor Power attack TC in log2	
Sh	drc_com_tc_log2_power bits 0-3	
Shp	xband Power Attack log2 for x = f, l, m, h	
C	ctl_code = drc_CTL_SET_ XBAND for X = F, L, M, H band.valid_bitfield = 2^(drc_BAND_CFG_VALID_LEV_TC_BIT) band.tc_log2 bits 4-7	Sets Compressor Power Estimation Release Time Constant to $1/(fd \cdot \log(1-2^{-N}))$. Default 8 @fd=8kHz 9 @fd=16 kHz Range=[0,15]
D	drc_band_cfg tc_log2_lev_release	
Se	drc_com_tc_log2_power Compressor Power release TC in log2	
Sh	drc_com_tc_log2_power bits 4-7	
Shp	xband Power Release log2 for x = f, l, m, h	
C	ctl_code = drc_CTL_SET_ XBAND for X = F, L, M, H band.valid_bitfield = 2^(drc_BAND_CFG_VALID_GAIN_TC_BIT) band.tc_log2 bits 8-11	Sets Compressor Gain Processor Attack Time Constant to $1/(fd \cdot \log(1-2^{-N}))$. Default 8 @fd=8kHz 9 @fd=16 kHz Range=[0,15]
D	drc_band_cfg tc_log2_gain_attack	
Se	drc_com_tc_log2_gain Compressor Gain attack TC log2	
Sh	drc_com_tc_log2_gain bits 0-3	
Shp	xband Gain Attack log2 for x = f, l, m, h	
C	ctl_code =drc_CTL_SET_ XBAND for X = F, L, M, H band.valid_bitfield = 2^(drc_BAND_CFG_VALID_GAIN_TC_BIT) band.tc_log2 bits 12-15	Sets Compressor Gain Processor Release Time Constant to $1/(fd \cdot \log(1-2^{-N}))$. Default 8 @fd=8kHz 9 @fd=16 kHz Range=[0,15]
D	drc_band_cfg tc_log2_gain_release	
Se	drc_com_tc_log2_gain Limiter Gain release TC log2	
Sh	drc_com_tc_log2_gain bits 4-7	
Shp	xband Gain Release log2 for x = f, l, m, h	
C	ctl_code = drc_CTL_SET_ XBAND for X = F, L, M, H band.valid_bitfield = 2^(drc_BAND_CFG_VALID_CURVE_PARAM_BIT) band.curve.exp_knee	The compressor's target gain expansion region begins above this Power Estimate Level, given in terms of dBm0. Default -50 dBm0. Range=[-78,3] dBm0
D	drc_band_cfg exp_knee	
Se	drc_exp_knee	
Sh	drc_exp_knee	
Shg	exp_knee_l, exp_knee_m, exp_knee_h the full-band mode value is stored in exp_knee_l	
Shp	xband Expansion Knee (dBm0) for x = f, l, m, h	

Table 4-4 Configurable Parameters for Full and Multiband Compressor (Part 2 of 3)

C = drcControl() Sh = show ipp_prof <prof> **D = dsp <tcid> Shg = show ipp_drc_gains <prof>** **Se = set ipp_prof <prof> Shp = show ipp_drc_params <prof>** **Seg = set ipp_drc_gains <prof> <param> <row> and set ipp_drc_gains <prof> values <row>**		**Description**
C	ctl_code = drc_CTL_SET_XBAND for X = F, L, M, H band.valid_bitfield = 2^(drc_BAND_CFG_VALID_CURVE_PARAM_BIT) band.curve.ratio bits 0-7	The compressor target gain curve expansion region's ratio for a change in output level divided by a change in input level, times 16. Default l/m/h = 32,32,16 Range = [16,255]
D	drc_band_cfg exp_ratio	
Se	drc_ratios Expansion ratio	
Sh	drc_ratios bits 0-7	
Shg	exp_ratio_l, exp_ratio_m, exp_ratio_h the full-band mode value is stored in exp_ratio_l	
Shp	xband Expansion Ratio for x = f, l, m, h	
C	ctl_code = drc_CTL_SET_ XBAND for X = F, L, M, H band.valid_bitfield = 2^(drc_BAND_CFG_VALID_CURVE_PARAM_BIT) band.curve.ratio bits 8-15	The compressor's target gain curve ratio in the compression region for a change in input level divided by a change in output level, times 16. Default l/m/h = 32,32,32 Range = [16,255]
D	drc_band_cfg com_ratio	
Se	drc_ratios Compression ratio	
Sh	drc_ratios bits 8-15	
Shg	com_ratio_l, com_ratio_m, com_ratio_h the full-band mode value is stored in com_ratio_l	
Shp	xband Compression Ratio for x = f, l, m, h	
C	ctl_code = drc_CTL_SET_XBAND for X = F, L, M, H band.valid_bitfield = 2^(drc_BAND_CFG_VALID_CURVE_PARAM_BIT) band.curve.max_amp	The compressor's target gain curve maximum amplification gain in the region between expansion and compression. Default l/m/h = 6,3,3 dB Range = [0,18] dB.
D	drc_band_cfg max_amp	
Se	drc_max_amp Maximum amplification	
Sh	drc_max_amp	
Shg	max_amp_l, max_amp_m, max_amp_h the full-band mode value is stored in max_amp_l	
Shp	xband Maximum Amplification (dB) for x = f, l, m, h	
C	ctl_code = drc_CTL_SET_XBAND for X = F, L, M, H band.valid_bitfield = 2^(drc_BAND_CFG_VALID_CURVE_PARAM_BIT) band.curve.com_knee	The compressor's target gain curve compression region begins above this Power Estimate Level given in terms of dBm0. Default l/m/h= -15,-20,-25 dBm0 Range=[-78,3] dBm0
D	drc_band_cfg com_knee	
Se	drc_com_knee	
Sh	drc_com_knee	
Shg	com_knee_l, com_knee_m, com_knee_h the full-band mode value is stored in com_knee_l	
Shp	xband Compression Knee (dBm0) for x = f, l, m, h	

Table 4-4 Configurable Parameters for Full and Multiband Compressor (Part 3 of 3)

C = drcControl() Sh = show ipp_prof <prof> D = dsp <tcid> Shg = show ipp_drc_gains <prof> Se = set ipp_prof <prof> Shp = show ipp_drc_params <prof> Seg = set ipp_drc_gains <prof> <param> <row> and set ipp_drc_gains <prof> values <row>		**Description**
C	ctl_code = drc_CTL_SET_ XBAND for X = F, L, M, H band.valid_bitfield = 2^(drc_BAND_CFG_VALID_CURVE_PARAM_BIT) band.curve.energy_lim	The compressor's target gain curve energy limit in terms of Power Estimate Level given in terms of dBm0. Default l/m/h = -3,-6,-7 dBm0 Range=[-78,3] dBm0
D	drc_band_cfg energy_lim	
Se	drc_energy_lim	
Sh	drc_energy_lim	
Shg	energy_lim_l, energy_lim_m, energy_lim_h the full-band mode value is stored in energy_lim_l	
Shp	xband Energy Limiting Threshold (dBm0) for x = f, l, m, h	
End of Table 4-4		

Table 4-5 Configurable Parameters Determining Overall DRC Setup

C = drcControl() **Sh = show ipp_prof <prof>** **D = dsp <tcid>** **Shg = show ipp_drc_gains <prof>** **Se = set ipp_prof <prof>** **Shp = show ipp_drc_params <prof>** **Seg = set ipp_drc_gains <prof> <param> <row> and set ipp_drc_gains <prof> values <row>**		**Description**
C	ctl_code = drc_CTL_DRC_ON	Enables DRC
D	drc_enable on	
Se	drc_enable on	
Sh	drc_enable ENABLE	
C	ctl_code = drc_CTL_DRC_OFF	Disables DRC
D	drc_enable off	
Se	drc_enable off	
Sh	drc_enable DISABLE	
C	ctl_code = drc_CTL_SET_BITS cfgbits = ZZZ	If bit set to one: Bit 0 Full Band Enabled Bit 1 Low Band Enabled Bit 2 Mid Band Enabled Bit 3 High Band Enabled Bit 4 Limiter Enabled Bit 5 VAD Enabled
D	drc_cfgbits cfgbits (bit-mapped)	
Se	drc_cfgbits DRC Configuration bitfield	
Sh	drc_cfgbits	
Shg	cfgbits	
Shp	cfgbits CFGBITS	
C	ctl_code = drc_CTL_SET_FIR_ XBAND for X = F, L, M, H coef[0], coef[1], … coef[drc_FIR_HALF_COEF_8KHZ] for fd=8 kHz coef[0], coef[1],…. coef[drc_FIR_HALF_COEF_16KHZ] for fd=16 kHz	Sets a multiband FIR filter that is symmetric about the last coefficient and has one ms delay. Defining half of coefficients requires N=9 coefs @fd=8 kHz N=17 coefs @fd=16 kHz
D	drc_coef <coef_1>, <coef_2>, …<coef_N>	
Se	drc_coef <coef_1>, <coef_2>, …<coef_N>	
Sh	fir_band: LOW/MID/HIGH BAND FIR	
Shp	xband_coef LOW/MID/HIGH BAND FIR FILTER COEFFICIENTS for x = f, l, m, h	
End of Table 4-5		

Appendix A

AER Module API Calls

The AER Developer Guide contains commands that enable you to change AER parameters for IP Phone tuning. However, those commands are valid only for IP Phone customers.

Table A-1 lists the corresponding AER API calls for the commands listed in "NMM Commands Relevant to AER Performance" on page 2-41. In the table, <tcid> of the IP Phone command is the telephone channel ID, and `aerID` and `agcID` are the corresponding AER/AGC instance IDs.

Table A-1 DSP Commands Used in the *AER Developer Guide* and Their Corresponding AER API Calls (Part 1 of 3)

DSP Command	AER API
Command: show ec_debug_stat <tcid> [clear] **Purpose:** display AER statistics	`aerGetPerformance(aer_inst[aerid], aer_stat[aerid], aer_version[aerid], 0)` assuming `aer_inst[]` is the array of AER instance pointers, `aer_stat[]` is the array of pointers of AER statistics of all instances, `aer_version[]` is the array of pointers of AER version information. The Boolean value "0" decides not to reset AER statistics.
Command: show ecpath_coeff <tcid> <start index> <number of coefficients> **Purpose:** obtain coefficients of the adaptive tail model (frequency domain)	`aerGetFilter(aer_inst[aerid], &tail_model, &coeff[0], start, length)` where tail_model is a structure that holds all necessary information of the tail model in addition to raw coefficients, coeff[] is the buffer to store the coefficients, start is the displacement from the start of the coefficients, length is the number of coefficients to obtain, tail_model.hscale_l2 is the power of 2 exponent of the coefficients.

Command: dsp <tcid> aerc <function enable or disable>
Purpose: enable or disable a specific AER function or feature
Example: `dsp <tcid> aerc nlpon` enables NLP
	Enabling NLP:
	`aerCtl.u.ctl_modes.mask = aer_CTL0_ENABLE_NLP;`
	`aerCtl.u.ctl_modes.value = aer_CTL0_ENABLE_NLP;`
	`aerCtl.ctl_code = aer_CTL_MODES_0;`
	`aerControl(aer_inst[aerid], (aerControl_t *)&aerCtl);`
	Disabling NLP:
	`aerCtl.u.ctl_modes.mask = aer_CTL0_ENABLE_NLP;`
	`aerCtl.u.ctl_modes.value = 0;`
	`aerCtl.ctl_code = aer_CTL_MODES_0;`
	`aerControl(aer_inst[aerid], (aerControl_t *)&aerCtl);`
	where `new_control_bits` will replace all old control bits of an AER instance; it must be the result of a logical OR operation of all bit values that need to be set.

Table A-1 DSP Commands Used in the *AER Developer Guide* and Their Corresponding AER API Calls (Part 2 of 3)

DSP Command	AER API

Command: dsp <tcid> aert <value|na> <value_mask|na> <value2|na> <value2_mask|na>
Purpose: set control bit field 0 and control bit field 1
	`aer_ctl.ctl_code = aer_CTL_MODES_0;`
	`aer_ctl.u.ctl_modes.mask = value_mask;`
	`aer_ctl.u.ctl_modes.value = value;`
	`aerControl(aer_inst[aerid], &aer_ctl);`
	
	`aer_ctl.ctl_code = aer_CTL_MODES_1;`
	`aer_ctl.u.ctl_modes.mask = value2_mask;`
	`aer_ctl.u.ctl_modes.value = value2;`
	`aerControl(aer_inst[aerid], &aer_ctl);`

Command: dsp <tcid> aerp <parameters>
Purpose: change AER parameters
Example: change tail length
	`aer_ctl.ctl_code = aer_CTL_TAIL_LENGTH;`
	`aer_ctl.u.tail_length = 1600; /* 200msec tail */`
	`aerControl(aer_inst[aerid], &aer_ctl);`
	
	Where `tail_length` is in number of filter taps at 8kHz

Command: dsp <tcid> aer_cng_nlp_params <parameters>
Purpose: change AER NLP and CNG parameters
Example: change Rx CNG level
	`aer_ctl.ctl_code = aer_CTL_CNG_RX_LEVEL;`
	`aer_ctl.u. cng_rx_level = -60; /* -60dBm0 */`
	`aerControl(aer_inst[aerid], &aer_ctl);`

Command: dsp [tcid] aer_eq_ctrl params <rx|tx> <parameters>
Purpose: change equalizer parameters
Example: set Tx Equalizer parameters
	`aer_ctl.ctl_code = aer_CTL_TXRX_EQ_PARAMS;`
	`aer_ctl.u.eq_params.tx_or_rx = aer_EQ_CTLM_TX_PARAMS;`
	`aer_ctl.u.eq_params.num_params = AER_NUM_EQ_PARS;`
	`aer_ctl.u.eq_params.eq_filter = (Fract *)&eqParams[0];`
	`aerControl(aer_inst[aerid], &aer_ctl);`
	
	where `eqParams` is the buffer to store equalizer coefficients, and `AER_NUM_EQ_PARS` is the number of equalizer parameters, which is 42 for this release. Refer to AER developer's guide about equalizer parameters.

Command: dsp <tcid> aer_eq_ctrl <rx_enable|rx_disable> <tx_enable|tx_disable>
Purpose: enable/disable Tx/Rx equalizer
	`aer_ctl.ctl_code = aer_CTL_TX_EQ_ONOFF;`
	`aer_ctl.u.txeq_enable = 1;  /* enable Tx EQ */`
	`aerControl(aer_inst[aerid], &aer_ctl);`

Command: dsp <tcid> aer_hlc_ctrl [enable|disable] [target_level|na] [ramp_down_tc|na] [ramp_up_tc|na] [power_tc|na]
Purpose: enable/disable HLC and set HLC parameters
	`aer_ctl.ctl_code = aer_CTL_HLC_THRESHOLD;`
	`aer_ctl.u.thresh_hlc = -12; /* -6dBm0 */`
	`aerControl(aer_inst[aerid], &aer_ctl);`
	
	`aer_ctl.ctl_code = aer_CTL_HLC_RAMP_OUT_TC;`
	`aer_ctl.u.ramp_out_tc_hlc = 700; /* 700msec/dB */`
	`aerControl(aer_inst[aerid], &aer_ctl);`
	
	`aer_ctl.ctl_code = aer_CTL_HLC_POWER_TC;`
	`aer_ctl.u.power_tc_hlc = 1; /* 4msec time constant */`
	`aerControl(aer_inst[aerid], &aer_ctl);`
	
	`aer_ctl.ctl_code = aer_CTL_HLC_ONOFF;`
	`aer_ctl.u.hlc_enable = 1;  /* enable HLC */`
	`aerControl(aer_inst[aerid], &aer_ctl);`

Command: dsp <tcid> aer_gain_chg_params [tx_ag_chg_synch_delay] [rx_ag_chg_synch_delay] [tx_ag_chg_settling_period]
Purpose: set AER gain change related parameters.
	`aer_ctl.ctl_code = aer_CTL_TX_AG_CHG_DELAY;`
	`aer_ctl.u.delay_tx_ag_chg = x; /* measured by user */`
	`aerControl(aer_inst[aerid], &aer_ctl);`
	
	`aer_ctl.ctl_code = aer_CTL_RX_AG_CHG_DELAY;`
	`aer_ctl.u.delay_rx_ag_chg = y; /* measured by user */`
	`aerControl(aer_inst[aerid], &aer_ctl);`
	
	`aer_ctl.ctl_code = aer_CTL_TX_AG_CHG_INTERP;`
	`aer_ctl.u.num_samp_interp = z; /* measured by user */`
	`aerControl(aer_inst[aerid], &aer_ctl);`

Table A-1 DSP Commands Used in the *AER Developer Guide* and Their Corresponding AER API Calls (Part 3 of 3)

DSP Command	AER API

Command: dsp [tcid] aer_nguard_ctrl <on|off> <hangover_period|na> <send_noise_level|na> <ramping_in_period|na>
 	
Purpose: enable/disable noise guard and set its parameters
	`aer_ctl.ctl_code = aer_CTL_NG_HANGOVER`
	`aer_ctl.u.hangover_ng = 500; /* 500msec */`
	`aerControl(aer_inst[aerid], &aer_ctl);`
	
	`aer_ctl.ctl_code = aer_CTL_NG_NOISE_LEVEL;`
	`aer_ctl.u.noise_level_ng = -70; /* -70dBm0 */`
	`aerControl(aer_inst[aerid], &aer_ctl);`

Command: dsp <tcid> agc <on|off> <adaptive:0=disabled|1=enabled> <sat_thresh|na> <sat_hangover|na> <max_micgain|na> <min_micgain|na> <max_micgain_swing|na> [clear]
 	
Note: Use [clear] option to clear everything and restart
 	
Purpose: enable/disable AGC and change AGC parameters
	`agc_ctl.ctl_code = agc_CTL_SET_SAT_THRESHOLD;`
	`agc_ctl.u.sat_threshold = 32500; /* 32767 is maximum */`
	`agcControl (agc_inst[agcid], &agc_ctl);`
	
	`agc_ctl.ctl_code = agc_CTL_SET_SAT_HANGOVER;`
	`agc_ctl.u.sat_threshold = 20; /* 20msec */`
	`agcControl (agc_inst[agcid], &agc_ctl);`
	
	`agc_ctl.ctl_code = agc_CTL_MODES;`
	`agc_ctl.u.ctl_modes.mask = agc_CTL_MODES_ENABLE_AGC |`
	`agc_CTL_MODES_ENABLE_ADAPT;`
	`agc_ctl.u.ctl_modes.value = agc_CTL_MODES_ENABLE_AGC;`
	`/* AGC on but not adaptive */`
	`agcControl (agc_inst[agcid], &agc_ctl);`
	
	where `agc_inst[]` is the array of AGC instance pointers.

End of Table A-1	

Glossary

Term	Definition
ADC	Analog-to-Digital Converter
AEC	Acoustic Echo Canceller (one type of acoustic echo removal (AER)—not an echo suppressor)
AGC	(Automatic Gain Control) For Texas Instruments software, AGC alters the microphone ADC PGA analog gain, but preserves the SLR and RLR. The sum of the ADC PGA analog gain and the AER transmit digital gain remain fixed. Performs multiple functions that affect the AER. AGC detects handsfree microphone input clipping and informs other DSP modules. AGC also provides true automatic gain control for the handsfree microphone input PGA gain. The handsfree microphone PGA gain changes can be generated internally by AGC. For example, after excessive microphone input ADC clipping, the PGA input gain is lowered automatically. Similarly, if the signal is persistently and excessively small while transmitting speech in either direction, the gain is increased; this improves audio quality by increasing the dynamic range. Under some circumstances, automatic gain changes may be compensated for with digital gain changes, downstream, that are opposite in magnitude. This preserves the transmitted signal level. The AER unit is informed about the magnitude of an external analog gain change to prevent de-converging. Also, AER needs to be informed about the sample delay until the AGC microphone gain change is activated is given to AER to eliminate any pops during a gain transition.
AIC	(Analog Interface Codec) A Texas Instruments two-channel hardware codec, such as the AIC22C and AIC20
ALC	Automatic Level Control. Typically, ALC alters digital gains to alter transmitted and/or received speech levels into some desirable range. ALC may cause problematic “noise pumping”, i.e. undesirable variations in background noise.
ASNR	ASNR (Adaptive Spectral Noise Reduction)
CNG	Comfort Noise Generation
Combined Loss	Ratio of the receive-out signal to the send-out signal power. Measures the total loss of the receive-out signal due to near-end reverberations, adaptive filter, and the NLP.
DAC	Digital to Analog Converter
dBPa	Paschal unit of sound pressure level (SPL). 1 dBPa=94 dBSPL.
dBV, dBm0	For IP Phones dBV is a virtual voltage unit and dBm0 is a digital signal power. In standards, 0 dBV=2.2 dBm0. A full scale digital sine wave has 3.0 dBm0.
Double Talk	The condition of having considerable activity in both the receive and transmit signals. This condition can also occur when the far end signal is propagated to the transmit signal due to a large echo without having explicit near-end signal activity.
DRC	Dynamic Range Compression
DSP	Digital Signal Processor
ERL	(Echo Return Loss) Measured in dB, ERL is the ratio of the receive-out and send-in powers. It measures the signal loss when it is transmitted back as echo (due to reflections at the NE) with the send-in signal.
ERLE	(Echo Return Loss Enhancement) Measured in dB, ERLE is the ratio of the send-in power and the power of the residual error signal after cancellation has been done. In effect, ERLE measures the cancellation introduced by the adaptive filter.
FDCC	FDCC (Frequency Domain Center Clipper)
FDCNG	FDCNG (Frequency Domain Comfort Noise Generation)
FDNLP	FDNLP (Frequency Domain Nonlinear Processor)

Term	Definition
FE	(Far End) The remote end of a two-node connection.
FFT	FFT (Fast Fourier Transform)
Filter Length	Specified in the number of filter taps. If N_t is the number of real-time coefficients used to model the impulse response, the time duration that this model covers in ms is given as $t = N_t/(S*8)$.
Frame Length	Length—specified in number of samples—of external signal frames that are supplied to the echo canceller for real-time processing. This parameter is configurable. The number of samples per frame depends on the sampling frequency. AER supports 8 kHz and 16 kHz sampling frequencies (F_s).
Full Duplex	Both send and receive channels are active and open
Group Listen	One person uses the handset while the group listens over the speaker.
HeS	(Headset) A telephone device that contains a microphone and speaker that is normally fitted over the ears in a headphones arrangement (with speaker elements for one or both ears), or small 'bud(s)' that are fitted in the ear canal.
HF	(Handsfree) A telephone unit with a microphone and speaker built into the phone base; typically used for group listening or group conversations in a small office or conference room.
HLC	(High Level Compensation) attenuates peaks in the input signal so that incoming signal from the packet network does not cause saturation or nonlinear distortion at the speaker output. HLC can be enabled or disabled and the threshold level set by the user. Also commonly known as *High Level Compressor*.
HS	(Handset) A conventional, hand-held telephone device that contains a microphone and speaker.
IFFT	IFFT (Inverse Fast Fourier Transform)
ISDN	(Integrated Services Digital Network)
mic	(Microphone)
NE	(Near End)
NLP	(Nonlinear Processor) Logic to provide additional echo suppression using additional attenuation and center clipper logic. The center clipper is used only for the transmit path, but an NLP gain is present in both the receive and transmit paths.
Noise Guard	This is a feature that attenuates the near-end noise (if required) to a comfortable level so that the far end can easily break in. When Noise Guard is active, it waits for a specified hangover to expire, making sure that it does not ramp in on short pauses in near-end speech. After the hangover expires, Noise Guard begins to ramp in the near-end noise.
Partial Duplex	One channel is active and the other channel is suppressed
PGA	Programmable Gain Amplifier
Phone Mode	There are several modes in which an IP Phone can be used • handset (HS) mode • handsfree (HF) mode • headset (HeS) mode • group listen (GL) mode—(can be used with handset (HSGL) or headset (HeSGL) Based on the usage, there are several permitted modes in which the echo canceller software can be used.
PSTN	Public Switched Telephone Network.
RDB	Reference Design Board
RLR	Receive Loudness Rating, the loss in levels from the digital handsfree speaker input in dBV to the sound output at 0.5 m away in dBPa.
Rx Path	Receive signal path from the network to the IP Phone speaker. The receive signal is also referred to as the far end (FE) signal.
Sampling Rate	Defined as $S = F_s/8000$. AER supports 8 kHz and 16 kHz sampling frequencies. S=1 for 8 kHz sampling frequency and S=2 for 16 kHz sampling frequency.
SDB	Software Development Board
SLR	Send Loudness Rating, the loss in levels from sound in dBPa 0.5 m from the handsfree microphone to transmitted digital levels in dBV.
STMR	Sidetone Masking Rating, the loss in sound level from NE microphone to NE speaker during NE speech. Hearing their own voice reassures the NE user that the phone isn't "dead". To isolate the handset sidetone, compare the handset speaker sound when blowing on the handset microphone with the handset cord disconnected and connected.
Tail Length	Represents the largest echo path impulse response that the AER can cancel successfully. This parameter (expressed in ms) is configurable with some restrictions. For AEC in handsfree phone mode, tail length is given in 20 ms steps within the range 20-200 [20, 200]. For AEC in handset mode, the range is [4, 20] in steps of 4 ms.
TCL	Terminal Coupling Loss

Term	Definition
TDCC	TDCC (Time Domain Center Clipper)
TDCNG	TDCNG (Time Domain Comfort Noise Generation)
TDNLP	TDNLP (Time Domain Nonlinear Processor)
Tx Path	Transmit signal path from the microphone of the IP Phone to the network. The transmit signal is also referred to as the near end (NE) signal.
VoIP	Voice over Internet Protocol
VoP	Voice over Packet

Index

A

adaptive mode, 2-6, 2-13, 2-26, 2-38 to 2-39, 2-54, 2-60, 2-63
AER (acoustic echo removal), 1-2, 1-4 to 1-5, 1-7, 1-9 to 1-11, 1-20 to 1-23, 1-27, 1-40, 1-43 to 1-48, 1-50 to 1-51, 2-2 to 2-6, 2-8 to 2-16, 2-18 to 2-65, 4-1 to 4-2, 4-4, 4-6 to 4-10, A-1 to A-3, GL-1 to GL-2
analog interface codec (AIC), 2-2, 2-4, 2-58, 4-10, GL-1
ASNR (Adaptive Spectral Noise Reduction), 2-6, 2-16 to 2-18, 2-36, 2-42 to 2-44, 2-49, GL-1
automatic gain control (AGC), 1-23, 1-43 to 1-44, 2-2, 2-4 to 2-6, 2-10, 2-13 to 2-14, 2-19 to 2-22, 2-25 to 2-27, 2-38 to 2-41, 2-50 to 2-51, 2-54, 2-57 to 2-61, 2-63 to 2-65, A-1, A-3, GL-1

B

back volume (chamber), speaker, 1-4, 1-25 to 1-28, 1-32, 1-36, 1-46
bandsplitting, 2-42

C

comfort noise generator, 1-23, 2-4, 2-6, 2-16 to 2-19, 2-43 to 2-44, 2-47, 2-49 to 2-50, 2-53, 4-7, A-2, GL-1, GL-3
conferencing, 2-30

D

dbgcmd (debug command), 2-22
debug
  statistics, 2-41
debug mode, 2-22
diagnostics, 3-13
DIM (DSP Interface Module), 2-38, 3-21
DRC (Dynamic Range Compression), 2-4 to 2-6, 4-1 to 4-10, 4-12 to 4-23, GL-1
DTMF, 1-23, 2-17

E

earth ground, 1-36 to 1-37
echo. See also ECU, EQI., 1-2 to 1-7, 1-9 to 1-10, 1-12, 1-17, 1-19 to 1-27, 1-29, 1-32 to 1-33, 1-35 to 1-36, 1-40, 1-44, 1-47, 1-49, 2-1 to 2-2, 2-8 to 2-11, 2-13, 2-16 to 2-17, 2-19, 2-21, 2-24 to 2-26, 2-29 to 2-35, 2-39 to 2-40, 2-42 to 2-44, 2-46 to 2-47, 2-49, 2-51 to 2-54, 2-58, 2-60, 3-2, 4-2 to 4-3, 4-6, 4-9 to 4-11, GL-1 to GL-2
equalizer, 1-20, 1-27, 2-6, 2-13 to 2-14, 2-26 to 2-28, 2-37, 2-42, 2-50, 2-53, 2-59, 2-63, 3-1, 3-3 to 3-7, 3-10, 3-12 to 3-15, 3-17 to 3-18, 3-20 to 3-21, 4-6, 4-11, A-2
error reporting and messages, 2-42 to 2-44, 2-54, 2-61, GL-1
ETSI (European Telecommunications Standards Institute), 3-9

F

FDCC (Frequency Domain Center Clipper), 2-16 to 2-17, GL-1
FDCNG (Frequency Domain Comfort Noise Generation), 2-16 to 2-17, GL-1
FDNLP (Frequency Domain Nonlinear Processor), 2-6, 2-16 to 2-17, 2-19, 2-44, 2-47, 2-49, GL-1
FFT (Fast Fourier Transform), GL-2
flash memory, 1-50, 2-41, 4-17

G

gasket, 1-4, 1-25, 1-27
grille, handset microphone, 1-5, 1-18, 1-25

H

handset, 1-5, 1-7 to 1-9, 1-12 to 1-13, 1-15, 1-17 to 1-19, 1-23 to 1-25, 1-29, 1-35, 1-40 to 1-41, 1-49 to 1-52, 2-2, 2-4, 2-8 to 2-9, 2-21, 2-23 to 2-27, 2-31 to 2-33, 2-35, 2-40, 2-45, 2-49, 2-51 to 2-52, 2-58 to 2-59, 2-63 to 2-65, 3-8 to 3-10, 3-12 to 3-13, 3-15, 3-20, GL-2
handset mode, 1-8 to 1-9, 1-13, 2-21, 2-40, 2-58, GL-2
handsfree, 1-4 to 1-11, 1-13 to 1-14, 1-16 to 1-35, 1-40 to 1-51, 2-2, 2-4, 2-8 to 2-9, 2-13, 2-21, 2-23 to 2-27, 2-30 to 2-38, 2-40, 2-45, 2-49, 2-51 to 2-54, 2-58 to 2-65, 3-8 to 3-9, 3-13, 3-20, GL-1 to GL-2

handsfree mode, 1-4 to 1-5, 1-7 to 1-9, 1-13, 1-17, 1-21, 1-25, 1-44, 1-46, 2-23, 2-25 to 2-26, 2-31, 2-36 to 2-38, 2-40, 2-45, 2-52 to 2-53, 2-58
headset, 1-7, 1-9, 1-17, 1-19, 1-23, 1-29, 1-51, 2-2, 2-4, 2-8 to 2-9, 2-21, 2-23, 2-26 to 2-27, 2-40, 2-45, 2-49, 2-51, 2-59, 2-63 to 2-65, 3-8 to 3-10, 3-13, 3-20, GL-2
headset mode, 1-17, 2-23, 2-40, 3-8
HLC (High Level Compensation), 1-11 to 1-12, 1-23, 1-43, 1-47, 2-4, 2-6, 2-11, 2-14 to 2-15, 2-26, 2-29, 2-31, 2-33, 2-36 to 2-38, 2-42 to 2-45, 2-48, 2-50, 2-53, 2-59, 2-63 to 2-65, 4-1 to 4-2, 4-4, 4-10, A-2, GL-2
hold, 1-49, 2-47

I

IFFT (Inverse Fast Fourier Transform), GL-2

L

LCD, 2-23, 2-58
LEC (line echo canceller), 1-40
loopback, 1-49

M

mode
  adaptive, 2-6, 2-13, 2-26, 2-38 to 2-39, 2-54, 2-60, 2-63
  debug, 2-22
  handset, 1-8 to 1-9, 1-13, 2-21, 2-40, 2-58, GL-2
  handsfree, 1-4 to 1-5, 1-7 to 1-9, 1-13, 1-17, 1-21, 1-25, 1-44, 1-46, 2-23, 2-25 to 2-26, 2-31, 2-36 to 2-38, 2-40, 2-45, 2-52 to 2-53, 2-58
  headset, 1-17, 2-23, 2-40, 3-8
  phone, 1-13, 1-51, 2-2, 2-4, 2-8, 2-21, 2-26, 2-38, 2-41, 2-45 to 2-46, 2-49, 2-51, 2-63, 3-7 to 3-8, GL-2
  signal limit, 2-15, 2-50
mute, 1-23, 2-4, 2-6, 2-19, 2-43 to 2-44, 2-46 to 2-48, 2-63

N

NLP (nonlinear processor), 1-6, 1-22, 1-40, 1-43, 2-2, 2-4, 2-6, 2-8 to 2-12, 2-16 to 2-19, 2-21, 2-24, 2-31 to 2-35, 2-40, 2-42 to 2-48, 2-50 to 2-54, 2-62 to 2-65, A-1 to A-2, GL-1 to GL-2
NMM (Network Management Module), 2-3, 2-22, 2-34, 2-36, 2-38, 2-41, 2-57 to 2-59, 2-61, 4-7 to 4-9, 4-17

P

packet
  buffer(s), A-1 to A-2
PCM, 1-40, 1-47, 2-43 to 2-44, 2-54
phone mode, 1-13, 1-51, 2-2, 2-4, 2-8, 2-21, 2-26, 2-38, 2-41, 2-45 to 2-46, 2-49, 2-51, 2-63, 3-7 to 3-8, GL-2
ping command, 2-36
PIQUA, 2-43
puff filter, 1-5, 1-18
PWB (printed wiring board), 1-2, 1-36 to 1-38

R

ramp, 2-6, 2-14 to 2-15, 2-29, 2-36, 2-38, 2-43, 2-50, 2-53, 2-64 to 2-65, A-2, GL-2
ramp_out, A-2
RTP (Realtime Transport Protocol), 2-31

S

sampling, 1-13 to 1-16, 1-21, 1-33 to 1-34, 2-4, 2-13, 2-27, 2-47, 2-64 to 2-65, 3-5 to 3-6, 3-8, 3-10 to 3-11, 3-13, 3-20, 4-14, GL-2
signal limit mode (AER), 2-15, 2-50
speakerphone, 1-2 to 1-5, 1-25 to 1-26, 1-32, 1-35 to 1-37, 4-2, 4-8 to 4-9
  speaker back volume chamber, 1-4, 1-25 to 1-28, 1-32, 1-36, 1-46
statistics
  collecting and reporting, 2-41, 2-43, A-1
  debug, 2-41
switchover, 2-11

T

tail length, 2-5 to 2-6, 2-8, 2-10, 2-19, 2-29 to 2-33, 2-36, 2-40, 2-43 to 2-46, 2-49 to 2-51, 2-60, 2-64 to 2-65, A-1 to A-2, GL-2
TDCC (Time Domain Center Clipper), GL-3
TDCNG (Time Domain Comfort Noise Generation), GL-3
TDNLP (Time Domain Nonlinear Processor), 2-6, 2-17, 2-49, GL-3
tlevels (transmit and receive levels), 2-58

U

UDP (User Datagram Protocol), 1-41

V

VAD (Voice Activity Detector), 1-40, 1-43, 2-27, 2-62, 4-3, 4-7, 4-12, 4-15 to 4-16, 4-23
version, 2-42, 2-59, 3-13, 3-20, A-1
voice codec support
  G.711, 1-13, 1-49
VOIP (Voice Over IP), 1-6 to 1-7, GL-3

W

WLAN, 3-9