NOTE TO USERS

Page(s) not included in the original manuscript and are unavailable from the author or university. The manuscript was scanned as received.

22, 88, 92, 94, 96, 100, 103, 105

This reproduction is the best copy available.

UMI[®]

HANDS-FREE TERMINALS FOR VOICE OVER IP

by

Trevor Yensen, B. Eng., M.Eng

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

Ottawa-Carleton Institute for Electrical and Computer Engineering
Faculty of Engineering
Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario, Canada
K1S 5B6

January 13, 2003

© Copyright 2003, Trevor Yensen

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-79469-5

Canada

The undersigned recommend to
the Faculty of Graduate Studies and Research
acceptance of the thesis:

“HANDS-FREE TERMINAL FOR VOICE OVER IP”

submitted by Trevor Yensen, , B.Eng., M.Eng.
in partial fulfillment of the requirements for
the degree of Doctor of Philosophy

Department Chair (Dr. Rafik Goubran)

External Examiner (Dr. Benoit Champagne)

Thesis Co-Supervisor, (Dr. Rafik Goubran)

Thesis Co-Supervisor, (Dr. Ioannis Lambadaris)

Abstract

Designing a hands-free terminal for voice over Internet protocol (VoIP) presents many challenges but also offers designers the opportunity to offer new features not currently available or even feasible on plain old telephone system (POTS) networks. This thesis focuses on techniques for offering hands-free spatial audio to the end-user and the problems associated with offering this enhanced feature set.

Multimedia-ready VoIP terminals are generally capable of offering stereo audio and some advanced terminals are even capable of offering surround sound audio to the end-user. Techniques for providing spatial audio to the near-end user are reviewed in this thesis and a recommended spatialization technique involving multiple speakers is made with reference to a common application programmer interface (API) called Microsoft DirectX.

VoIP systems typically suffer from long delays, which necessitate the addition of acoustic echo suppression techniques such as adaptive echo cancellers and optionally microphone array beamformers to hands-free terminals. This thesis proposes novel structures for synthetic stereo and synthetic surround sound audio presentation combined with acoustic echo cancellation. To further enhance echo suppression and provide increased near-end gain, this thesis proposes a combination of near-field microphone array beamforming and the synthetic stereo acoustic echo cancellation structure.

Long delays in a VoIP conference contribute to listener confusion and also require very large levels of echo suppression. It is therefore desirable to reduce the end-to-end delay of VoIP conferences to a minimum. A technique that compromises between the end-to-end delay and packet loss rate is proposed in this thesis that uses hidden Markov model (HMM) techniques. The proposed technique differs from other delay computation techniques since generic model parameters are precomputed in advance and are then applied to predict suitable VoIP conference delays.

The HMM technique or another existing playout delay computation technique can be used to compute the end-to-end delay as perceived by the user, called the acoustic round trip delay (ARTD). ARTD can be used to compute appropriate acoustic echo suppression targets using talker echo loudness rating curves, which indicate the appropriate amount of echo suppression required for a given end-to-end delay. This thesis proposes an echo suppression system for VoIP terminals that uses attenuators that are precisely controlled by echo suppression target levels to reduce half-duplex attenuation.

This thesis proposes several new technologies and shows unique ways of applying various existing technologies to provide enhanced hands-free full-duplex VoIP conferencing to the user. By following the recommendations made in this thesis, VoIP terminals will be able to provide users with an enhanced conferencing experience with new features previously infeasible on a POTS network.

Acknowledgements

I would like to thank my supervisors, Dr. Rafik Goubran and Dr. John Lambadaris for their assistance in helping direct my studies and determine an appropriate topic for my PhD thesis. During the course of the research for this thesis I obtained assistance from many parties. In particular, I would like to thank Dr. Jeff Lariviere, Dr. Neil Birkett and Dr. Jim Ryan for participating in many brain-storming sessions and providing helpful suggestions. It is important to note that Dr. Jim Ryan provided the near-field beamformer configuration used in this thesis.

The chapter on playout delay prediction using a hidden Markov model could not have been simulated without the aid of Dr. Cormac Sreenan and Dr. Henning Schulzrinne who allowed me to access their Internet delay trace data. I offer my thanks to you and your fellow researchers who were involved in collecting this data.

I would like to thank my family and friends for their encouragement, which was crucial to motivating me to complete this thesis. In particular, I would like to thank Tamara Trudeau for her support and countless hours spent proofreading this document.

Finally, I would like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC), Communications and Information Technology Ontario (CITO) and Carleton University for their financial support.

Table of Contents

ABSTRACT	III
TABLE OF CONTENTS.....	VI
LIST OF FIGURES	IX
LIST OF TABLES	X
LIST OF TABLES	X
LIST OF SYMBOLS	XI
1 INTRODUCTION.....	1
1.1 SPATIAL AUDIO	2
1.2 ACOUSTIC ECHO CANCELLATION	2
1.3 MICROPHONE ARRAY BEAMFORMING	3
1.4 PACKET DELAY PREDICTION.....	4
1.5 ECHO SUPPRESSION TARGET DETERMINATION.....	4
1.6 SYNOPSIS.....	5
2 BACKGROUND INFORMATION	7
2.1 ACOUSTIC ECHO CANCELLATION.....	7
2.1.1 <i>Acoustic Echo Canceller Limitations</i>	9
2.2 ADAPTIVE FILTERING ALGORITHMS.....	10
2.2.1 <i>The Least Mean Square Algorithm (LMS)</i>	10
2.2.2 <i>The Recursive Least Square Algorithm (RLS)</i>	11
2.2.3 <i>The Affine Projection Algorithm (APA)</i>	12
2.3 NONLINEAR PROCESSOR	14
2.4 MICROPHONE ARRAY BEAMFORMING	19
2.5 TCP/IP PROTOCOL SUITE	23
2.5.1 <i>User Datagram Protocol</i>	24
2.5.2 <i>Transmission Control Protocol</i>	24
2.5.3 <i>Internet Protocol</i>	25
2.6 ITU-T H.323 AND IETF SESSION INITIATION PROTOCOL.....	25
2.6.1 <i>ITU-T H.323</i>	25
2.6.2 <i>IETF Session Initiation Protocol</i>	27
2.6.3 <i>Real-Time Transport Protocol/Control Protocol</i>	27
3 REVIEW OF THE STATE OF THE ART.....	29
3.1 ACOUSTIC SPATIALIZATION TECHNIQUES.....	29
3.2 STEREOPHONIC ACOUSTIC ECHO CANCELLATION.....	32
3.3 SYNTHETIC STEREO ACOUSTIC ECHO CANCELLATION	35
3.3.1 <i>Synthetic Stereo with a Single Canceller Per Channel</i>	37
3.4 VOICE PACKET PLAYOUT DELAY ESTIMATION	38
3.4.1 <i>Voice Packet Timing Notation</i>	41
3.4.2 <i>Optimal Playout Delay</i>	42
3.4.3 <i>Playout Delay Scheduling Techniques</i>	44
3.4.3.1 Off-line Playout Delay Computation	45
3.4.3.1.1 Optimal Algorithm for an Isolated Talkspurt.....	45
3.4.3.1.2 Minimum Average Playout Delay	46

3.4.3.1.3	Fixed Playout Delay	46
3.4.3.2	Predictive Playout Delay Algorithms	47
3.4.3.2.1	Causal Version of Optimal Algorithm	48
3.4.3.2.2	Delay Statistics Algorithm	48
3.4.3.2.3	Packet Delay Distribution Algorithm	49
4	SPATIAL AUDIO AND ECHO CANCELLATION	52
4.1	SYNTHETIC STEREO ACOUSTIC ECHO CANCELLATION	52
4.1.1	Overview	52
4.1.2	Synthetic Stereo Acoustic Echo Cancellation Structure	52
4.1.3	Multiple Active Spatial Regions	57
4.1.4	Comparison to Other Stereo Structures	59
4.1.5	Simulation	61
4.2	SYNTHETIC SURROUND SOUND ACOUSTIC ECHO CANCELLATION	65
4.2.1	Overview	65
4.2.2	Synthetic Surround Sound Acoustic Image	66
4.2.3	Synthetic Surround Sound Acoustic Echo Cancellation Structure	68
4.2.4	Analysis	71
4.2.5	Comparing Single Channel to Synthetic Multi-Channel Acoustic Echo Cancellation	74
4.2.6	Simulation	75
5	NEAR-FIELD MICROPHONE ARRAY AND SYNTHETIC STEREO ACOUSTIC ECHO CANCELLATION	79
5.1	OVERVIEW	79
5.2	NEAR-FIELD OPTIMIZED MICROPHONE ARRAY	79
5.3	MICROPHONE ARRAY BEAMFORMER	81
5.4	ARRAY RESPONSE TO NEAR-END TALKER	84
5.5	SYNTHETIC STEREO	86
5.6	ACOUSTIC ECHO CANCELLATION STRUCTURE	88
5.7	SIMULATION	91
6	HIDDEN MARKOV MODEL PLAYOUT DELAY PREDICTION	94
6.1	OVERVIEW	94
6.2	HMM TECHNIQUE OVERVIEW	94
6.3	GENERIC PLAYOUT DELAY MODEL	97
6.4	HIDDEN MARKOV MODEL NOTATION	99
6.5	HIDDEN MARKOV MODEL STRUCTURE	103
6.6	TRAINING VS. PLAYOUT DELAY PREDICTION	104
6.7	TRAINING THE HMM	104
6.7.1	Compute Playout Delay Off-Line	106
6.7.2	Initial State and State Transition Probabilities	107
6.7.3	Excess Delay Probabilities	109
6.8	PLAYOUT DELAY PREDICTION USING THE HMM	112
6.8.1	Compute the Mean Delay Estimate	114
6.8.2	Viterbi Algorithm	114
6.8.3	Compute the Probability of the Observation Sequence	115
6.9	SIMULATION	117
6.9.1	Network Delay Traces	117
6.9.2	Generating Talkspurt Information	117
6.9.3	Simulated Delay Prediction Techniques	118
6.9.4	Testing Results	119
7	ACOUSTIC ROUND TRIP DELAY AND ECHO SUPPRESSION TARGET DETERMINATION	123
7.1	ECHO TOLERANCE CURVES	123
7.1.1	Asymmetric Internet Transmit/Receive Paths	124

7.1.2	<i>TELR Curves for VoIP</i>	125
7.2	TIME SYNCHRONIZATION	129
7.3	ACOUSTIC ROUND TRIP DELAY MEASUREMENT	131
7.4	ECHO SUPPRESSION TARGET DETERMINATION.....	137
8	CONCLUSIONS.....	139
8.1	SYNTHETIC STEREO ACOUSTIC ECHO CANCELLATION	141
8.2	SYNTHETIC SURROUND SOUND ACOUSTIC ECHO CANCELLATION.....	142
8.3	NEAR-FIELD BEAMFORMING AND SYNTHETIC STEREO ACOUSTIC ECHO CANCELLATION	143
8.4	HMM PLAYOUT DELAY PREDICTION	144
8.5	ACOUSTIC ROUND TRIP DELAY AND ECHO SUPPRESSION TARGET DETERMINATION	145
8.6	SUMMARY OF CONTRIBUTIONS.....	146
8.6.1	<i>Synthetic Stereo Acoustic Echo Cancellation</i>	146
8.6.2	<i>Synthetic Surround Sound Acoustic Echo Cancellation</i>	146
8.6.3	<i>Near-Field Beamforming and Synthetic Stereo Acoustic Echo Cancellation</i>	147
8.6.4	<i>HMM Playout Delay Prediction</i>	147
8.6.5	<i>Echo Suppression Target Determination</i>	148
8.7	FUTURE WORK	148
9	REFERENCES	151

List of Figures

FIG. 1. ACOUSTIC ECHO CANCELLATION STRUCTURE.....	8
FIG. 2. IDEAL CENTRE CLIPPER TRANSFER FUNCTION – OPTION 1	16
FIG. 3. IDEAL CENTRE CLIPPER TRANSFER FUNCTION – OPTION 2	16
FIG. 4. ECHO CANCELLATION STRUCTURE WITH SWITCHED-LOSS	18
FIG. 5. TYPICAL NARROWBAND BEAMFORMER USED FOR RF.....	21
FIG. 6. TYPICAL BROADBAND BEAMFORMER USED FOR VOIP	22
FIG. 7. TCP/IP PROTOCOL STACK.....	24
FIG. 8. H.323 TERMINAL EQUIPMENT	26
FIG. 9. NEAR-END USER LOUDSPEAKER CONFIGURATION	30
FIG. 10. VIRTUAL SOURCE POSITIONED IN NEAR-END USER'S ACOUSTIC SPACE	31
FIG. 11. TRUE STEREOPHONIC ACOUSTIC ECHO CANCELLATION STRUCTURE FOR LEFT NEAR-END MICROPHONE ...	33
FIG. 12. NEAR-END TALKER SPATIALIZATION WITH FOUR SPATIAL REGIONS ($M=4$)	36
FIG. 13. SYNTHETIC STEREO STRUCTURE WITH A SINGLE CANCELER PER-CHANNEL	38
FIG. 14. HIGH-LEVEL VIEW OF PACKET VOICE APPLICATION	39
FIG. 15. VOICE PACKET TIMING DIAGRAM	41
FIG. 16. PACKET DELAY DISTRIBUTION (PDD).....	50
FIG. 17. SYNTHETIC STEREO STRUCTURE WITH A SINGLE CANCELER PER-SPATIAL-REGION	53
FIG. 18. MEASURED LOUDSPEAKER TO MICROPHONE IMPULSE RESPONSE.....	62
FIG. 19. SYNTHETIC STEREO PERFORMANCE WITH A SINGLE CANCELER PER-SPATIAL-REGION	63
FIG. 20. SYNTHETIC STEREO PERFORMANCE WITH A SINGLE CANCELER PER-SPATIAL-REGION AND SPEECH REFERENCE SIGNALS.....	64
FIG. 21. NEAR-END TALKER SPATIALIZATION.....	67
FIG. 22. SYNTHETIC SURROUND ACOUSTIC ECHO CANCELLATION STRUCTURE	69
FIG. 23. FRONT-RIGHT POSITION LOUDSPEAKER SPATIALIZATION IMPULSE RESPONSES	71
FIG. 24. RECEPTION ROOM IMPULSE RESPONSES	76
FIG. 25. SYNTHETIC STEREO VERSUS SYNTHETIC SURROUND	77
FIG. 26. MULTIPLE ACTIVE SPATIAL REGIONS.....	78
FIG. 27. PHYSICAL ARRANGEMENT OF MICROPHONE ARRAY AND MULTIMEDIA LOUDSPEAKERS.....	80
FIG. 28. NEAR-FIELD GAIN OF BEAMFORMER.....	84
FIG. 29. ENERGY DECAY FOR NEAR-END TALKER.....	86
FIG. 30. NEAR-END TALKER SPATIALIZATION WITH BEAMFORMING	87
FIG. 31. SYNTHETIC STEREO ACOUSTIC ECHO CANCELLATION STRUCTURE WITH BEAMFORMER	89
FIG. 32. ACOUSTIC ECHO CANCELER PERFORMANCE WITH BEAMFORMER.....	93
FIG. 33. LONG-TERM AND SHORT-TERM PDDs	96
FIG. 34. PLAYOUT DELAY COMPONENTS.....	98
FIG. 35. ERGODIC HMM.....	103
FIG. 36. TRAINING PROCEDURE	105
FIG. 37. INITIAL STATE PROBABILITY	107
FIG. 38. STATE TRANSITION PROBABILITIES.....	109
FIG. 39. EXCESS DELAY MEAN AND VARIANCE.....	112
FIG. 40. PLAYOUT DELAY PREDICTION PROCEDURE	113
FIG. 41. ARTIFICIAL TALKSPURT GENERATION	118
FIG. 42. TALKER ECHO LOUDNESS RATING TOLERANCE CURVES.....	126
FIG. 43. VOIP PACKET TIMING	132
FIG. 44. ROUND-TRIP TIME COMPUTATION USING RTCP	136
FIG. 45. ECHO SUPPRESSION TARGET DETERMINATION.....	138

List of Tables

TABLE I. HMM DELAY PREDICTION RESULTS FOR SREENAN DATA	120
TABLE II. HMM DELAY PREDICTION RESULTS FOR SCHULZRINNE DATA	122

List of Symbols

3D	Three Dimensional
AEC	Acoustic Echo Canceller
alp	Average Lost Packet Rate
APA	Affine Projection Algorithm
ARTD	Acoustic Round Trip Delay
Codec	Coder/Decoder
CNG	Comfort Noise Generation
DLSR	Delay Since Last Sender Report
DOA	Direction of Arrival
DOPT	Delayed/Causal Version of the Optimal Algorithm for Voice Packet Delay Prediction
DSP	Digital Signal Processing
ECAN	Echo Cancellation
EL	Echo Loudness Loss
ERLE	Echo Return Loss Enhancement
ETSI	European Telecommunications Standards Institute
FAP	Fast Affine Projection Algorithm
FEC	Forward Error Correction
FIR	Finite Impulse Response

GPS	Global Positioning System
HMM	Hidden Markov Model
HRTF	Head Related Transfer Function
IETF	Internet Engineering Task Force
IIR	Infinite Impulse Response
ILD	Interaural Level Difference
IP	Internet Protocol
ITD	Interaural Time Difference
ITU-T	International Telecommunications Union – Telephony Division
LAN	Local Area Network
LMS	Least Mean Square
NLMS	Normalized Least Mean Square
NLP	Nonlinear Processor
NTP	Network Time Protocol
OPT	Optimal Off-Line Algorithm for Voice Packet Delay Prediction
PC	Personal Computer
PDD	Packet Delay Distribution
POTS	Plain Old Telephone System
PSTN	Public Switched Telephone Network
QoS	Quality-of-Service

RF	Radio Frequency
RLR	Receiver Loudness Rating
RLS	Recursive Least Square
RTCP	Real-Time Transport Protocol Control Protocol
RTP	Real-Time Transport Protocol
RTT	Round-Trip Time
SIP	Session Initiation Protocol
SLR	Sender Loudness Rating
STAT	Statistics Based Algorithm for Voice Packet Delay Prediction
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
ted	Total End-to-End Delay
TELR	Talker Echo Loudness Rating
TIP	Total Impulse Power
TP	Tail Power
UDP	User Datagram Protocol
VAD	Voice Activity Detector
VoIP	Voice Over Internet Protocol
WAN	Wide Area Network

1 Introduction

Voice over Internet protocol (VoIP) is a relatively new communications technology that uses Internet protocol (IP) based networks to deliver multimedia information such as speech over a data network. A single medium for voice and data reduces costs from duplication of services, but new complexities arise due to the combination of these services. The Internet was not originally designed to replace the circuit switched networks that traditionally carry voice traffic over the public switched telephone network (PSTN). Several technologies have evolved to carry voice and video successfully over the Internet; however, since the Internet was not originally designed to carry this sort of traffic, issues such as long delays continue to be a problem. This thesis will propose new technologies important to designing a hands-free VoIP terminal and address how the Internet's shortcomings affect the design.

A full-duplex hands-free VoIP system requires a network interface, monaural, stereo or surround sound loudspeakers, a single microphone transducer or microphone array and VoIP software that is compliant with an appropriate standard such as the International Telecommunications Union – Telephony Division (ITU-T) recommendation H.323, or the Internet Engineering Task Force (IETF) proposed standard 2543 called the Session Initiation Protocol (SIP). A VoIP system can be coupled with video, whiteboard, text messaging and a variety of other services; however, this thesis will focus on the hands-free audio terminal portion of a VoIP system.

1.1 Spatial Audio

A new feature that is currently unimplemented in plain old telephone system (POTS) networks and that can be easily implemented in a voice over Internet protocol (VoIP) terminal is the addition of surround sound or 3D spatial audio generated by two or more loudspeakers for multiple participant conferences. Many terminals designated for VoIP are already or can easily be equipped with multiple loudspeakers. Surround sound spatial audio can be used in multiple participant voice conferencing to enhance the user's ability to distinguish between remote participants or to provide spatial cues for virtual reality. A multiple participant conference with surround sound audio would be well complemented by a 3D video display of avatars. The division between VoIP and virtual reality becomes less apparent as multimedia technologies are successfully combined. These techniques will be referred to throughout this thesis as audio spatialization techniques since they translate monaural signals into multi-channel equivalents.

1.2 Acoustic Echo Cancellation

A full-duplex hands-free terminal is desirable for VoIP since it offers the user the ultimate flexibility of not having to wear a headset (wired or wireless), carry a handset or be in constant close proximity to a microphone. A pure full-duplex system does not perform any switching between transmit and receive directions and it does not add any attenuation to the transmit or receive path to reduce echo. Switching between transmit and receive directions can cause the user's speech to be partially cut off and can make it difficult to carry on a normal conversation. Adding attenuation

to the transmit or receive path reduces acoustic echo, but also reduces the system gain. The reduction in system gain may make it difficult for the near-end listener to understand the far-end talker due to an attenuated receive signal level or may make it difficult for the far-end listener to understand the near-end talker due to an attenuated transmit signal level. Ideally VoIP systems should be entirely full-duplex, but in reality they are often a combination of the two. It is desirable to maintain as much full-duplex functionality as possible through a full-duplex preserving technique such as acoustic echo cancellation. This thesis will present structures for performing acoustic echo cancellation with one, two or more loudspeakers.

1.3 Microphone Array Beamforming

Microphone array beamforming is another technique that preserves the full-duplex nature of the audio terminal and can be used to provide acoustic echo suppression to the system. Microphone array beamforming also reduces background noise and reverberation, and can provide extra transmit path gain. This thesis will show how near-end microphone array beamforming can be successfully combined with the aforementioned acoustic echo cancellation structures and audio spatialization techniques to achieve higher levels of echo suppression without resorting to transmit-receive switching or audio path attenuation.

1.4 Packet Delay Prediction

VoIP systems use the Internet protocol, which is a packet switched networking technology originally designed for non-real-time data communication applications such as file transfers and electronic mail. The latencies for packet switched networks can vary from a few milliseconds to hundreds of milliseconds or more and may vary over time, depending on the topology of the network and the various components involved. This thesis proposes a new method, for any number of participants, based on hidden Markov models (HMM) used to predict appropriate packet playout delays that compromise between packet loss and end-to-end delay. Human speech has a large amount of built in redundancy that can be exploited to reduce the end-to-end delay of a VoIP system. Predicting appropriate end-to-end delay levels allows the system to minimize the amount of signal attenuation used to provide adequate levels of echo suppression because the amount of echo suppression required for a specific end-to-end propagation delay increases as the delay increases [1][2]. If the end-to-end delay can be minimized then the amount of echo suppression provided by non full-duplex preserving techniques can be minimized so that the overall system gain is improved and the user's perception of the conference quality is increased.

1.5 Echo Suppression Target Determination

This thesis proposes a method for determining the end-to-end delay as perceived by a hands-free VoIP conference participant, called the acoustic round trip delay (ARTD). This metric can be used to compute an appropriate acoustic echo suppression target. Two methods for computing the ARTD and their advantages and

disadvantages are presented. The ARTD time can be computed through coordination with playout delay scheduling algorithms, such as the HMM delay prediction technique proposed by this thesis, which require clock synchronization by the network time protocol (NTP). The ARTD time can also be computed by using network round trip time information, which does not require clock synchronization. Acoustic echo suppression targets are determined by computing the ARTD for a VoIP conference and combining this information with psycho-acoustic echo suppression data obtained from ITU-T standards. Accurate determination of echo targets is required to minimize signal path attenuation and switching delay effects from the nonlinear processor in full-duplex hands-free conferences. The nonlinear processor can fine-tune signal attenuation levels if ARTD information is used to accurately compute the total echo suppression required by the system.

1.6 Synopsis

The remainder of this thesis is structured in the following manner. Chapter 2 introduces the reader to the central concepts that are the focus of this thesis, in particular acoustic echo cancellation, microphone array beamforming, the non-linear processor, TCP/IP and VoIP standards. Chapter 3 is a review of the current state of the art in acoustic spatialization, stereophonic acoustic echo cancellation with two microphone transducers and two loudspeakers, synthetic stereo per-channel acoustic echo cancellation with a single microphone and two loudspeakers and voice packet playout delay estimation techniques. Chapter 4 proposes a structure for synthetic stereo per-spatial-region acoustic echo cancellation with a single

microphone and two loudspeakers and an extension of this technique from two loudspeakers to three or more loudspeakers, called synthetic surround per-spatial region acoustic echo cancellation. Chapter 5 proposes combining near-field beamforming with the synthetic stereo per-spatial-region acoustic echo cancellation structure for increased echo suppression and reverberation reduction. Chapter 6 discusses estimation of voice packet playout delay using hidden Markov models that are trained a priori. Chapter 7 links the acoustic echo cancellation structures to the proposed playout delay prediction technique by proposing a method for computing echo suppression targets using playout delay information. Finally, chapter 8 summarizes the thesis and references are presented in chapter 9.

2 Background Information

2.1 Acoustic Echo Cancellation

The acoustic echo canceller (AEC) is a major part of the echo suppression system for hands-free VoIP conferencing. The acoustic echo canceller eliminates the coupling between the speaker and microphone by adapting to the room impulse response [3].

Fig. 1 shows a typical acoustic echo cancellation structure with reference signal, $s(n)$, primary signal, $d(n)$, and error signal, $e(n)$. The far-end talker's voice, $s(n)$, is played over the near-end talker's speakers and propagates from the loudspeaker to the microphone, $h(n)$, where it is recorded as echo, $d(n)$, by the near-end talker's microphone. The output from the adaptive filter, $s(n) * \hat{\mathbf{h}}(n)$, is used to cancel the echo, where $*$ denotes convolution. The adaptive filter, $\hat{\mathbf{h}}(n)$, is defined at time n with N coefficients, $\hat{\mathbf{h}} = [\hat{h}_1, \hat{h}_2, \dots, \hat{h}_N]$.

Fig. 1. Acoustic echo cancellation structure

The usual convergence criteria for an AEC is to minimize the error signal, $e(n)$, by adapting its model of the room, $\hat{\mathbf{h}}(n)$, to equal the actual room impulse response, $\mathbf{h}(n)$ [3][4]. If perfect cancellation occurs, in the absence of noise, then $e(n) = 0$ and the adaptive filter impulse response is exactly equal to the room impulse response, $\mathbf{h}(n) = \hat{\mathbf{h}}(n)$, provided that the covariance of the far-end signal driving the adaptive filter is full rank (positive definite).

$$e(n) = d(n) - \mathbf{s}(n) * \hat{\mathbf{h}}(n) = \mathbf{s}(n) * [\mathbf{h}(n) - \hat{\mathbf{h}}(n)] \quad (1)$$

The filter is only allowed to adapt when near-end speech is not detected. When near-end speech is detected the adaptation is frozen and the near-end speech signal propagates through the system. Acoustic echo from the far-end is still cancelled but

the filter coefficients used, referred to as the frozen coefficients, are the values for $\hat{\mathbf{h}}(n)$ before near-end speech was detected.

The filter should also not be allowed to adapt during periods of double-talk, where both near-end and far-end speech are present. The filter is still active, but adaptation is not possible due to the existence of near-end speech.

2.1.1 Acoustic Echo Canceller Limitations

An AEC is unable to achieve perfect cancellation due to acoustic noise, quantization noise, truncation noise, undermodelling of the acoustic transfer function, enclosure vibration effects, transducer nonlinearities and algorithm tracking and convergence rates [5][6]. Due to these limitations, the maximum echo return loss enhancement (ERLE) for an acoustic echo canceller is often insufficient for the requirements of the conference.

Echo return loss enhancement is a common measure of AEC performance. ERLE is a measure of the amount of echo power that is cancelled by the AEC [5], as shown in Fig. 1.

$$ERLE(dB) = 10 \log_{10} \frac{E[d^2(n)]}{E[e^2(n)]} \quad (2)$$

The maximum theoretical amount of ERLE that can be obtained for a specified number of adaptive filter taps is computed by the total impulse power (TIP) to tail power (TP) ratio [5].

$$\frac{TIP}{TP}(n, M)(dB) = 10 \log_{10} \left[\frac{\sum_{i=0}^{M-1} h_i^2}{\sum_{i=n}^{M-1} h_i^2} \right] \quad (3)$$

2.2 Adaptive Filtering Algorithms

This thesis focuses on acoustic echo cancellation structures but does not propose any new adaptive filtering algorithms. The acoustic echo cancellation structures proposed by this thesis will function with any acceptable adaptive filtering algorithm. A brief discussion of three well-known time domain adaptive filtering algorithms is presented in this section. The following adaptive algorithms are discussed:

- The Least Mean Squares Algorithm (LMS)
- The Recursive Least Squares Algorithm (RLS)
- The Affine Projection Algorithm (APA)

A more detailed description of these algorithms can be found in [3][7][8][9][10][11].

2.2.1 The Least Mean Square Algorithm (LMS)

The least mean square algorithm (LMS) is perhaps the best-known adaptive filtering algorithm since it is a very versatile algorithm with low computational complexity. The LMS algorithm is simple because it does not require any off-line gradient estimations or data repetition [11]. The LMS algorithm is stated below:

$$\hat{\mathbf{h}}(n+1) = \hat{\mathbf{h}}(n) + \mu e(n) \mathbf{x}(n)$$

$$e(n) = d(n) - \hat{\mathbf{h}}^T(n) \mathbf{x}(n)$$

where $\hat{\mathbf{h}} = [\hat{h}_1, \hat{h}_2, \dots, \hat{h}_N]^T$ is the adaptive filter vector of length N , $\mathbf{x}(n) = [x_1, x_2, \dots, x_N]^T$ is the reference vector of length N , $d(n)$ is the desired or primary signal, $e(n)$ is the error signal and μ is the step-size.

The NLMS algorithm is the most commonly used variant of the LMS algorithm since its step-size is normalized such that the convergence rate and steady state performance of the adaptive filtering algorithm are invariant to the input signal characteristics. The step-size, μ , is normalized to the input power of the system. The standard LMS algorithm uses a fixed step-size. The step-size for the NLMS algorithm is computed at each iteration as follows:

$$\mu = \frac{\alpha}{\varepsilon + N \cdot E[x^2(n)]}$$

where $E[x^2(n)]$ is the expected value of the squared input signal, α is the NLMS step-size and ε is a small positive constant added to the denominator to avoid numerical instability.

2.2.2 The Recursive Least Square Algorithm (RLS)

The recursive least-square (RLS) algorithm has the fastest convergence rate of the three time domain adaptive algorithms reviewed in this thesis but has the highest computational complexity [12][13]. The convergence rate of the RLS algorithm is fast because the input and desired signals are used in an optimal manner to compute the gradient. The RLS algorithm involves inversion of the autocorrelation matrix, $\mathbf{R}_n$,

which, if performed using the classical Gaussian elimination technique is computationally wasteful [10]. Fortunately, a matrix inversion lemma can be applied to substantially reduce the computational complexity surrounding inversion of the $\mathbf{R}_n$ matrix. The resultant computational complexity of the RLS algorithm is $2N^2+4N$ multiplications. The RLS algorithm is as follows [10]:

1. Accept new samples $x(n)$ and $d(n)$ to form new reference vector $\mathbf{X}(n)$.
2. Compute a priori output $y(n) = \hat{\mathbf{h}}_n^T \mathbf{X}(n)$
3. Compute a priori error $e(n) = d(n) - y(n)$
4. Compute filtered information vector $\mathbf{Z}_n = \mathbf{R}_n^{-1} \mathbf{X}(n)$
5. Compute normalized error power $q = \mathbf{X}^T(n) \mathbf{Z}_n$
6. Compute gain constant $\nu = \frac{1}{1+q}$
7. Compute normalized filtered information vector $\hat{\mathbf{Z}}_n = \nu \cdot \mathbf{Z}_n$
8. Update optimal weight vector $\hat{\mathbf{h}}_{n+1} = \hat{\mathbf{h}}_n + e(n) \hat{\mathbf{Z}}_n$
9. Update inverse correlation matrix $\mathbf{R}_{n+1}^{-1} = \mathbf{R}_n^{-1} - \hat{\mathbf{Z}}_n \mathbf{Z}_n^T$

The inverse autocorrelation matrix is set to $\mathbf{R}^{-1} = \eta \mathbf{I}_N$ where η is a large positive constant and $\mathbf{I}_N$ is the identity matrix, upon algorithm initialization.

2.2.3 The Affine Projection Algorithm (APA)

The affine projection algorithm (APA) offers improved behavior over the NLMS algorithm by making better use of input signal data [12]. The affine projection algorithm uses a multi-dimensional projection for each adaptive filter tap [12]. The

NLMS algorithm is equivalent to an APA with a projection order of 1 since each adaptive filter tap is one-dimensional.

The affine projection algorithm is capable of canceling P a posteriori errors such that $1 \leq P \leq N$ where P is the projection order and N is the length of the filter vector. If the projection order is 1, $P=1$, then the algorithm performs identically to the NLMS, but if the projection order is N , $P=N$, then the algorithm performs identically to the block based recursive least square (BRLS) algorithm [12]. The filter coefficients are adjusted in the direction of the plane produced by $x(k)$, $x(k-1)$, ..., $x(k-P+1)$, where k is the time index [7]. The input is whitened according to the projection order, P [14].

Increased APA convergence and tracking performance can be obtained by increasing the projection order, which in turn increases the computational complexity [8]. It has been shown, [9], that APA tracking ability can outperform the LMS and RLS algorithms.

The affine projection algorithm is as follows:

$$\begin{aligned} \mathbf{e}_n &= \mathbf{s}_n - \mathbf{X}_n^T \mathbf{h}_{n-1} \\ \boldsymbol{\epsilon}_n &= [\mathbf{X}_n^T \mathbf{X}_n + \sigma \mathbf{I}] \mathbf{e}_n \\ \mathbf{h}_n &= \mathbf{h}_{n-1} + \mu \mathbf{X}_n \boldsymbol{\epsilon}_n \end{aligned}$$

where $\mathbf{X}_n = [\mathbf{x}_n, \mathbf{x}_{n-1}, \dots, \mathbf{x}_{n-(P-1)}]$ is the excitation signal matrix, with $\mathbf{x}_n = [x_n, \dots, x_{n-P+1}]^T$, $\mathbf{h}_n = [h_{0,n}, \dots, h_{N-1,n}]^T$ is the adaptive filter coefficients, $\mathbf{e}_n = [e_n, e_{n-1}, \dots, e_{n-P+1}]$ is the error vector and σ is the regularization parameter.

The computational complexity of the affine projection algorithm is $2PN + K_{\text{inv}}N^2$ multiplies per sample period, where K_{inv} is a constant depending on the inversion method used. (K_{inv} for Levinson Durbin is about 7.)

The fast affine projection algorithm (FAP) is considered the most important advancement for the APA. The FAP algorithm has LMS like complexity and memory requirements and RLS like convergence for speech signals. The computational complexity of FAP is roughly $2N + 20P$ multiplications per sample period [8]. FAP does not delay the input or output signals [8]. If the relaxation parameter is set to one, or smaller, the computational complexity becomes $2P + 14N$ multiplications per sample period. A more detailed description of the FAP algorithm can be found in [8].

2.3 Nonlinear Processor

Practical implementations of echo suppression systems require additional signal attenuation levels determined by the nonlinear processor (NLP) to overcome the acoustic echo canceller's limitations [15]. The nonlinear processor adds additional signal attenuation to reduce the effect of the nonlinear elements of the acoustic echo that cannot be modeled by the adaptive AEC. These nonlinear elements include nonlinear echoes and quantization noise caused by mathematical inaccuracies in the

modulation method. Nonlinear echoes can be caused by a number of factors including clipped speech, nonlinear microphone elements or loudspeakers, and speech coders. The NLP also compensates for the residual linear echo from the adaptive filter's finite impulse response length. Signal attenuation and centre-clipping are typically used to compensate for nonlinearities in the acoustic echo.

A centre-clipper is defined as a device that conceptually blocks low-level signals and passes high-level signals [15]. The NLP uses signal receive and transmit levels to determine an appropriate suppression threshold for the centre-clipper. A typical centre-clipper implementation sets all signal samples with amplitude less than a specified threshold value, T_{SUP} , to zero. Two possible centre-clipper implementations are shown in Fig. 2 and Fig. 3. A disadvantage of these centre-clipper implementations is that they typically introduce noise when the signal is gated by T_{SUP} .

Fig. 2. Ideal Centre Clipper Transfer Function – Option 1

Fig. 3. Ideal Centre Clipper Transfer Function – Option 2

Currently, the NLP does not use information about the delays in the conference to determine appropriate levels of attenuation and centre-clipping for the system. This thesis proposes a more accurate method for determining the amount of echo suppression to be provided by the NLP based on determination of the transmission time for a VoIP conference, see chapter 7.

Fig. 4 shows an acoustic echo cancellation structure with a nonlinear processor that controls the addition of switched-loss attenuation. In Fig. 4 the attenuators at the near-end, Participant A's terminal, are shown as inactive since the near-end user is speaking and the near-end center clipper is disabled or its attenuation level is reduced when near-end speech is detected. The echo loss, EL, is shown for both the acoustic echo cancellation, EL_{AEC} , and the nonlinear processor, EL_{NLP} .

The NLP is normally located in the send path between the output of the subtractor and send-out port of the echo canceller, L_1 , [15]. In hands-free systems, however, the NLP is often implemented as a combination of an attenuator in the receive direction and a centre-clipper in the send direction, called a bi-directional NLP implementation (see L_1 and L_2 in Fig. 4 assuming that Participant A is speaking). Bi-directional NLP implementations can be used to gracefully degrade performance towards half-duplex. The centre-clipper, L_1 , suppression threshold, T_{SUP} , is lowered by increasing the level of attenuation at L_2 . Lowering the centre-clipper suppression threshold, T_{SUP} , results in less chance of near-end speech being cutoff when near-end speech resumes or during double-talk.

Fig. 4. Echo cancellation structure with switched-loss

Echo suppression by the AEC is desirable since cancellation of the echo signal by this means does not require attenuation or centre-clipping of the signal path. Echo suppression through attenuation and centre-clipping, controlled by the NLP, causes attenuation of the signal path and adds noise, which reduces the overall system gain and adds unwanted half-duplex effects to the conference. The half-duplex effects are a result of the NLP switching the signal attenuation on and off and adjusting the suppression threshold, T_{SUP} . When Participant A is speaking, the coefficients for Participant B's acoustic echo canceller adapt and the NLP provides echo suppression through signal attenuation and centre-clipping. When Participant A stops speaking and Participant B begins speaking the coefficients for Participant B's AEC are frozen but continue to filter using the last known reception room impulse response estimate and the coefficients for Participant A's AEC begin to adapt and the NLP enables signal suppression and centre-clipping. The time delay required for

switching AEC adaptation on and off does not create any switching artifacts since the AEC filter is active at all times and normally the reception room impulse response does not change dramatically between talkers. The time delay required for the NLP to switch signal path attenuation on and off, however, can create detrimental effects such as cutting off part of a user's speech that can lead to talker confusion. Minimizing the amount of NLP signal path suppression and centre clipping is, therefore, desirable.

This thesis proposes new methods for accurately estimating the required amount of signal path attenuation to be provided by the NLP through measurement of the acoustic round trip delay (ARTD) for a VoIP conference. This technique is intended to be used by existing NLP algorithms. Current NLP techniques rely primarily on signal power information as the metric for determining their required echo suppression targets.

2.4 Microphone Array Beamforming

Multiple microphone sensor arrays can be used for a wide range of applications, such as detection, identification, localization and tracking of signal sources. For hands-free VoIP applications, microphone array beamformers can be used to reduce environmental noise, reduce reverberation, improve intelligibility, steer video cameras towards the active talker and to reduce acoustic echo. Microphone array beamformers combine spatially distributed microphone sensor signals with filters, which are typically linear, to achieve spatial filtering. Spatial filtering is an effective

means for enhancing a desired signal and attenuating any interfering signals because this filtering technique takes advantage of the desired signal's spatial location relative to the microphone array.

VoIP applications deal with audible waveforms that have spectral content in the range of 30Hz to 15kHz [16]. The ratio of these frequency limits is approximately 500, which makes this signal a broadband signal. In contrast, most radio frequency (RF), waveforms have a ratio between their highest frequency and lowest frequency near unity. These waveforms are classified as narrowband. A narrowband beamformer typically samples the propagating wave in the spatial domain only, see Fig. 5. The narrowband beamformer output, $y(n)$, is defined as follows, [17]:

$$y(n) = \sum_{l=1}^J b_l^* x_l(n)$$

where b_l^* is the complex conjugate weight for sensor l and $x_l(n)$ is the antenna input signal for sensor l .

Fig. 5. Typical narrowband beamformer used for RF

Broadband signals, such as speech, are not characterized by a nominal wavelength and a broadband beamformer typically samples the propagating wave in both the spatial and time domains, see Fig. 6. The broadband beamformer output, $y(n)$, is defined as follows, [17]:

$$y(n) = \sum_{l=1}^J \sum_{p=0}^{N-1} b_{l,p}^* x_l(n-p)$$

where $b_{l,p}^*$ is the complex conjugate weight for sensor l at time p , $x_l(k-p)$ is the microphone signal for sensor l at time p .

determine the source direction, also called direction of arrival (DOA) since the wavefront curvature cannot be explored to determine distance.

The propagation speed for acoustic sound in air at room temperature is constant at approximately 345 m/s, therefore acoustic wavefronts arrive at each microphone in the array at the same velocity. This phenomenon greatly simplifies processing at the microphone array. VoIP terminals are typically placed in environments that suffer from varying degrees of reverberation, as opposed to free-space environments, which are not reverberant. Reverberation can make DOA and distance estimation much more difficult depending on the level of the reflected energy. However, proper beamforming can help to reduce the detrimental effects of reverberation for applications such as VoIP.

2.5 TCP/IP Protocol Suite

The transmission control protocol/Internet protocol (TCP/IP) suite is the most widely used suite of protocols for non-guaranteed quality-of-service networks. The TCP/IP suite provides reliable packet transmission through the transmission control protocol (TCP) and unreliable best-try service through the user datagram protocol (UDP). The function of the IP layer is to route TCP packets or UDP datagrams. Fig. 7 shows a hierarchical view of the TCP/IP protocol stack [20].

OS/Application Process	
FTP, TELNET, SMTP, NSP, SNMP	
TCP	UDP
IP	
IEEE 802.X/X.25	

Fig. 7. TCP/IP protocol stack

2.5.1 User Datagram Protocol

The user datagram protocol (UDP) is a best-try, connectionless service. No prior connection needs to be established and packet overhead is lower than for connection-oriented service [20]. UDP is used extensively by H.323 and the session initiation protocol (SIP) to send packets via the real-time transport protocol (RTP), also called H.225.0, since there is not usually enough time to retransmit packets and lower packet overhead is desirable for real-time traffic [21].

2.5.2 Transmission Control Protocol

The transmission control protocol (TCP) is a connection-oriented, reliable service [20]. A TCP connection is established between two endpoints using a three-way handshaking protocol. The handshaking protocol exchanges sequence numbers and other information before a stream is transmitted. The connection established by TCP is full duplex and uses sliding window flow control with go-back-N error

correction. TCP is not normally used for real-time traffic since the packet overhead is much larger than with UDP. TCP is used when errors cannot be tolerated and preventative measures such as three-way handshaking and error correction are required.

2.5.3 Internet Protocol

The Internet protocol (IP) is responsible for fragmentation and reassembly of packets, routing and for error reporting [20]. The IP protocol is a best-try connectionless protocol. Version 4 of IP, IPv4, is the most commonly used on the Internet today; however, version 6, IPv6, is making inroads in many academic and research environments. Eventually IPv6 will replace IPv4 due to the ever increasing demand for Internet addresses and number of real-time applications in use. IPv6 expands the IP address field from 32-bits to 128-bits and provides a means of relieving traffic congestion through its inherent native multicasting capability.

2.6 ITU-T H.323 and IETF Session Initiation Protocol

2.6.1 ITU-T H.323

H.323, [22][23], is a toolkit standard which details the transmission of audio, video and data services over networks which do not provide a guaranteed quality-of-service. H.323 is a toolkit standard since it is a combination of several other ITU-T standards (see Fig. 8). The H.323 standard covers system and component descriptions, called model descriptions and control method formats. H.245 specifically focuses on messages and procedures used for opening and closing

logical channels for audio, video and data [23][24]. H.245 also covers capability exchange, mode requests, control and indications. H.225.0 focuses on audio and video stream packetization, media stream synchronization, control stream packetization and control message formats [23][25].

The audio codecs specified by H.323 include G.711 [26], which is mandatory, and the optional G.722 [27], G.723.1 [28], G.728 [29], and G.729 [30]. The mandatory video codec (video conferencing support is optional), is H.261 [31] and the optional video codec is H.263 [32]. Data applications, such as whiteboards, are covered by T.120 [33].

Fig. 8. H.323 terminal equipment

2.6.2 IETF Session Initiation Protocol

The H.323 standard is closely paralleled by work being performed by the Internet Engineering Task Force (IETF) on its own multimedia communications protocol suite called the Session Initiation Protocol (SIP) [34]. H.323 and SIP both use the real-time transport protocol (RTP) [35][36] to transport media streams [37]. SIP is a client-server protocol responsible for translating application-layer addresses, name translation, feature negotiation and establishing and managing calls [38]. The major difference between H.323 and SIP is how call signaling and control are handled. SIP uses a text-based format for signaling, which mandates a common structure for decoding by generic parsers. Text-based signaling is more readily extendible and the bandwidth penalty is minimal. Signaling information uses a very small percentage of the VoIP conference bandwidth, therefore, efficiency is not a concern. Signaling protocols such as those employed by H.323 and Q.931 [39] require a rigidly formatted signaling payload.

2.6.3 Real-Time Transport Protocol/Control Protocol

Perhaps the most important component for SIP and H.323 is the real-time transport protocol/control protocol (RTP/RTCP) that is used to carry all real-time audio and video streams and to provide feedback concerning the current network conditions and to predict future network congestion. RTP is responsible for sequencing, intra-media synchronization, inter-media synchronization, payload identification and frame indication [38]. Sequencing is the process of re-ordering packets that arrive out of order at the receiver. Intra-media synchronization is the process of communicating timing information necessary to the proper playout of the

audio or video stream, such as audio speech and silence periods. Inter-media synchronization is the process of synchronizing different media streams, such as lip-syncing for audio and video. The payload identifier is used to determine the current encoding scheme, which can vary depending on the available system bandwidth, such as adjusting an audio codec to a lower bitrate codec when network congestion occurs. Frame indication is a service provided to higher layers to indicate the logical beginning or end of a frame for audio and video streams.

RTP and RTCP have been designed to be multicast-friendly, media independent, interoperate with mixers and translators, provide QoS feedback via RTCP, allow for loose session control and provide encryption [38]. RTCP is the control protocol companion to RTP, which is responsible for periodically sending RTCP packets between the various hosts in a VoIP conference. These RTCP packets are used to provide session control, synchronization between different media and compute the network packet loss rate, jitter and end-to-end network delay, called round-trip time (RTT).

3 Review of the State of the Art

3.1 Acoustic Spatialization Techniques

Stereo representation of multiple talkers is possible through a variety of methods. A lot of recent attention has been focused on head related transfer functions (HRTF), which are well suited to binaural applications, such as those involving headphones [40]. HRTFs can be used in transaural monitoring, i.e. listening over loudspeakers, by convolving the HRTF with an appropriate loudspeaker decorrelation function [41]. The loudspeaker decorrelation functions compensate for the coupling between the left loudspeaker to right ear and right loudspeaker to left ear. These transaural decorrelation functions are only applicable to a known and static loudspeaker configuration and listener head position. Any deviation of the physical setup or movement of the listener's head will cause listener confusion.

For hands-free VoIP conferencing it is impractical to assume that the listener's head will remain in a constant position, and there is no practical listener head-tracking solution to date that does not involve extra hardware and increased computational complexity. As suggested in [4][42][43][44] the solution for desktop conferencing is to employ stereo techniques that use interaural level differences (ILD) and interaural time differences (ITD) [40][45].

Interaural level differences are produced by varying the amplitude of the signal on one loudspeaker with respect to the other. The stereophonic law of sines computes the apparent angle due to level differences in the two loudspeakers [45] as

$$\sin \Theta = \frac{L - R}{L + R} \sin \Theta_0 \quad (4)$$

where $2\Theta_0$ is the angle subtended between the two loudspeakers at the listener's head, Θ is the apparent off-centre angle produced by the level differences, $-\Theta_0 \leq \Theta \leq \Theta_0$, L is the left loudspeaker level and R is the right loudspeaker level. The typical near-end user's loudspeaker configuration is shown in Fig. 9. Fig. 10 shows how a virtual source is positioned in the near-end user's acoustic space using the stereophonic law of sines.

Fig. 9. Near-end user loudspeaker configuration

Fig. 10. Virtual source positioned in near-end user's acoustic space

Interaural time differences can be used to further enhance the listener's sound stage. Delaying the signal at one loudspeaker with respect to the other creates interaural time differences. ITDs, however, tend to introduce "phasiness" effects in transaural monitoring due to loudspeaker cross-coupling and may be undesirable in talker localization [45]. ITD should not exceed ± 3 ms for continuous image movement and for the avoidance of excessive talker confusion due to a limited stereo sweet spot [45].

3.2 Stereophonic Acoustic Echo Cancellation

Acoustic echo cancelers are essential in any full-duplex, hands-free communications system. The purpose of the acoustic echo canceler is to eliminate the echoes caused by the coupling between the loudspeaker and microphone. The true stereophonic structure, [46], requires a stereo microphone configuration at the far-end that experiences two transmission room impulse responses, which are depicted in Fig. 11 as $g_L(n)$ and $g_R(n)$. Four sources of feedback can be identified at the near end, for stereophonic acoustic echo cancellation. The feedback is caused by the coupling between the left loudspeaker and left microphone, the left loudspeaker and right microphone, the right loudspeaker and right microphone, and the right loudspeaker and left microphone.

Fig. 11 shows the structure of a stereophonic acoustic echo canceler for the left near-end microphone. The reception room impulse responses, $h_{LL}(n)$ and $h_{LR}(n)$, corresponding adaptive echo path models, $\hat{h}_{LL}(n)$ and $\hat{h}_{LR}(n)$ and the echo signal or primary signal, $d_L(n)$, are shown for the left near-end microphone. The right near-end microphone structure, not shown in Fig. 11, has an identical echo canceler structure with different reception room impulse responses, $h_{RL}(n)$ and $h_{RR}(n)$, corresponding to adaptive echo path models, $\hat{h}_{RL}(n)$ and $\hat{h}_{RR}(n)$ and primary signal $d_R(n)$.

Fig. 11. True stereophonic acoustic echo cancellation structure for left near-end microphone

A true stereophonic structure has two far-end microphones and two far-end loudspeakers. When two microphones are used in a true stereophonic system the reference signals are cross-correlated [46], which leads to echo canceller misconvergence. This phenomenon is called the fundamental problem of stereophonic acoustic echo cancellation. The stereophonic acoustic echo cancellation structure suffers from misconvergence due to cross-correlation between the reference signals, $s(n)*g_L(n)$ and $s(n)*g_R(n)$ [46]. The problem of echo canceller misconvergence can be explained by an analysis of the error signal at either the left or right microphone. The error signal for the left microphone is

$$e_L(n) = s(n) * \mathbf{g}_L(n) * [\mathbf{h}_{LL}(n) - \hat{\mathbf{h}}_{LL}(n)] + s(n) * \mathbf{g}_R(n) * [\mathbf{h}_{LR}(n) - \hat{\mathbf{h}}_{LR}(n)] \quad (5)$$

If the error signal is minimized in the absence of noise or interferers, $e_L(n) = 0$, then (5) reduces to the following relationship

$$\mathbf{g}_L(n) * [\mathbf{h}_{LL}(n) - \hat{\mathbf{h}}_{LL}(n)] = -\mathbf{g}_R(n) * [\mathbf{h}_{LR}(n) - \hat{\mathbf{h}}_{LR}(n)] \quad (6)$$

From (6) it can be seen that there are an infinite number of solutions to the problem. When the error signal is minimized it does not imply that $\mathbf{h}_{LL}(n) = \hat{\mathbf{h}}_{LL}(n)$ and $\mathbf{h}_{LR}(n) = \hat{\mathbf{h}}_{LR}(n)$. The canceler converges to any one of multiple instantaneous solutions to the problem, which do not identify the correct echo path [46].

Several structural and algorithmic solutions to deal with the misconvergence problem have been proposed in recent papers. The structural solutions include using a single canceler [47], a subband structure [48], a non-linear transformation [49], or adding noise to the reference signal [46][50]. The algorithmic solutions include the use of the least mean square (LMS) algorithm and its variants [46], the recursive least squares (RLS) algorithm [13], and the affine projection algorithm (APA) [8][51].

Modern stereophonic acoustic echo cancelers use front-end preprocessing techniques that modify the left and right channel acoustic signals in a manner that is deemed inaudible to the listener [49][50]. These structural modifications are combined with an algorithm that converges rapidly such as the RLS or APA algorithm. The preprocessor decorrelates the reference signals, allowing the

stereophonic acoustic echo canceler to converge to the proper left and right channel impulse responses.

3.3 Synthetic Stereo Acoustic Echo Cancellation

True stereophony is desirable for conferences between two physical locations since it gives the near-end listeners localization cues concerning the location and movement of talkers at the far-end. In multiple-participant VoIP conferences, where there are more than two physical locations with typically one participant at each location, the far-end spatial information for each participant is not as important to the near-end user as the ability to distinguish between participants [4][42][43][44]. Synthetic stereo conferencing gives the near-end user static localization cues for distinguishing between far-end talkers in the conference. Synthetic stereo does not provide information about the far-end talker's movement. These localization cues become increasingly important for distinguishing between talkers as the number of participants in a conference grows.

The synthetic stereo conferencing configuration uses only one microphone, a monaural source, for each far-end user and stereo loudspeakers at the near end [4][42][43][44]. The dynamic transmission room transfer functions that provide movement and position information in true stereophonic conferencing are replaced by static spatialization functions, unique to each participant, that provide user identification cues. Fig. 12 shows how six far-end talkers are presented to the near-end user on a video display terminal and in acoustic space. The video display and

acoustic space are divided into spatial regions and each far-end talker's image location on the video display corresponds to their location in acoustic space. With transaural monitoring, the user is only able to clearly distinguish between a limited number of regions in acoustic space due to horizontal localization blur [40]. The physical horizontal dimension of the video display terminal also limits the number of spatial regions that can be seen.

Fig. 12. Near-end talker spatialization with four spatial regions (M=4)

3.3.1 Synthetic Stereo with a Single Canceler Per Channel

A synthetic stereo acoustic echo cancellation structure was proposed in [43][44]. This novel structure performs input spatialization before taking the acoustic echo canceler reference, see Fig. 13, and therefore requires a separate echo canceler for each of the two channels in the stereo path. If the length of the adaptive echo path models $\hat{\mathbf{h}}_L(n)$ and $\hat{\mathbf{h}}_R(n)$ is T_F filter taps, then the per-channel structure will require $2T_F$ taps for the two stereo channels. By taking the reference after the spatialization functions, the same reference signal cross-correlation problem as in the traditional stereophonic structure is present [43][44]. The authors of [43][44] propose adding a small nonlinearity to the left and right channels and using the two-channel recursive least squares (RLS) or fast affine projection (FAP) algorithm to achieve convergence, similar to the true stereophonic structure proposed in [49]. Energy must be present on both loudspeakers for the echo canceler to maintain the correct impulse responses [44] due to canceler allocation on a per-channel basis. To maintain energy on both loudspeakers in the extreme left or right lateralization cases, some amount of interaural time difference (ITD) must be added to the interaural level difference (ILD), which can add “phasiness” effects [45]. Echo canceler coefficients can converge when multiple talkers are active since cancelers are assigned on a per-channel basis.

Fig. 13. Synthetic stereo structure with a single canceler per-channel

3.4 Voice Packet Playout Delay Estimation

Fig. 14 shows a high-level view of the various blocks required to construct a typical packet voice application for IP networks adapted from [22][52]. The sending user's speech must be converted from the analog to the digital domain by an A/D converter. The speech signal bandwidth is then reduced by the vocoder. The vocoder output is used to form a real-time transport protocol (RTP) packet, which is then encapsulated by the user datagram protocol/Internet protocol (UDP/IP) and sent over the network. The reverse process occurs at the receiving user's end, except that the packet spacing must be restored by introducing a suitable delay in the receive path.

Fig. 14. High-level view of packet voice application

Packet voice applications generate audio packets at a fixed interval, when the sender is speaking. The interval is dependent on the vocoder employed by the system (typically every 20ms). The voice packets are then transmitted over the Internet from the sender to the receiver and experience queuing delays, which vary from packet to packet depending on congestion in the network. These variable network queuing delays cause packet interarrival jitter at the receiver. During silence periods, when the far-end talker is not speaking, it is assumed that no packets are transmitted to reduce the bandwidth consumed by the packet voice application.

The voice packets are held in a playout buffer, also known as a jitter buffer, when they arrive at the receiver until their scheduled playout time at which point they are

passed to the voice decoder algorithm. The playout buffer is used at the receiving end to reconstruct the deterministic timing required by the vocoder to play the packets at a constant rate. The vocoder will introduce a fixed decoding delay separate from the network delay, which is not modeled by the playout delay prediction algorithm. If deterministic timing is not restored at the input to the vocoder then there will be audible gaps in the signal, which will greatly decrease the quality of the voice signal.

Fig. 15 shows the timing for voice packets for the following sequence of events. At the beginning of the figure the sender is speaking, this is called talkspurt k . During talkspurt k , packets are produced in a deterministic manner with constant packet spacing. These packets arrive at the receiver with different delays due to queuing in the Internet. The receiver delays the packets until a scheduled playout time, thereby reconstructing the deterministic timing. The end of talkspurt k occurs when the sender stops speaking. This is called the silence interval, during which the sender transmits no packets. After the silence interval, during talkspurt $k+1$, the sender resumes speaking.

Fig. 15. Voice packet timing diagram

3.4.1 Voice Packet Timing Notation

The following notation will be used for playout delay prediction algorithms to describe the packet-audio stream. This notation is adapted from the paper by Moon et al. [53]

- M : the number of talkspurts in a given Internet trace
- n_k : the length of talkspurt k in audio packets
- t_k^i : the time when the sender generates packet i in the k -th talkspurt
- a_k^i : the time when the receiver receives packet i in the k -th talkspurt

- p_k^i : the time when the receiver plays out packet i in the k -th talkspurt
- d_k^i : the normalized network delay experienced by packet i in the k -th talkspurt. Network delays are normalized by the smallest delay difference to generate positive relative delay values, where

$$d_k^i = a_k^i - t_k^i - \min_{1 \leq k \leq M, 1 \leq i \leq n_k} \{a_k^i - t_k^i\} \quad (7)$$

- $d_k^{(i)}$: the delay to play i packets in the k th talkspurt (i -th highest delay)
- $\hat{p}_k$: the predicted playout delay for the k -th talkspurt

The delay algorithms presented in this thesis do not assume that the sender and receiver clocks are synchronized and clock synchronization is not necessary for the algorithms presented in this thesis to function for playout delay prediction, since they predict relative delays. The predicted playout delays are based on the normalized one-way delay for the audio conference. The normalized delay, d_k^i , is introduced such that all normalized delays that the playout delay prediction technique deals with are positive and relative to the smallest delay of 0ms.

3.4.2 Optimal Playout Delay

Minoli et al. characterize total end-to-end delay as the summation of delay due to analog to digital conversion, the delay due to the packetization, the network propagation and queuing delay and the jitter buffer waiting time at the receiver end

[52]. The maximum total end-to-end delay that should not be exceeded by a packet voice application is 200 ms and typically total end-to-end delays between 100 and 200 ms are targeted [52]. The total end-to-end delay targets can often only be obtained if the system accepts a small number of lost packets due to late arrival. Long total end-to-end delays are unsuitable for interactive conversations since they impair normal conversation between participants and may intensify the effect of acoustic or network echoes [1][42].

The optimal playout delay is a compromise between three criteria, the end-to-end delay experienced by the user, also called total end-to-end delay (ted), the fluctuation of the end-to-end delay and the number of lost packets due to late arrival, called average lost packets (alp).

A compromise must be made between lowering the average end-to-end delay and average lost packet rate, since lowering one, generally results in an increase in the other. A low average end-to-end delay for a voice audio stream is desirable to the end user, as described above. Similarly, a low average number of lost packets is also desirable to the end user since the conference quality of service may be adversely affected if the number of lost packets exceeds the capabilities of the vocoder lost packet compensation scheme. A large amount of fluctuation in the end-to-end delay is also problematic to the user and results in lower quality of service.

There is a large amount of redundancy in a voice audio stream due to the nature of human voice production. A voice packet application can tolerate a small average number of lost packets since the vocoder can compensate for these lost packets. A

reasonable number of lost packets can, therefore, be tolerated in exchange for a decrease in the average end-to-end delay and delay fluctuation. Ramjee et al. state that a loss rate target between 1% and 10% can be tolerable, depending on the vocoder and masking techniques in use [55]. The joint optimization criteria for any playout delay prediction algorithm is, therefore, to minimize the total packet end-to-end delay, the end-to-end delay standard deviation and limit packet losses to an acceptable target level.

There is an important distinction between the number of packets lost due to late arrival and the average number of packets lost due to being dropped by the network. Packets lost due to being dropped by the network never arrive, whereas packets lost due to late arrival arrive too late to be used for playout. This thesis typically ignores packets dropped by the network due to congestion at routers and assumes that they are compensated for by the receiver, typically through some sort of comfort noise generation (CNG) technique. Packets lost due to late arrival, which is a function of the scheduled packet playout delay, fail to arrive at the receiver before their scheduled playout time and are unusable at that point. The alp statistic used by this thesis will only deal with packets lost due to late arrival.

3.4.3 Playout Delay Scheduling Techniques

Several existing playout delay scheduling techniques will be reviewed in this section. Off-line playout delay computation algorithms will be discussed for their usage in computing the optimal playout delay for an isolated talkspurt in a non-causal fashion, which can be used as a benchmark for other algorithms. The advantages

and disadvantages of a fixed playout delay technique will be discussed. Finally, the family of adaptive or predictive playout delay techniques will be discussed since they are the most practical for real-time implementation in network environments such as the Internet.

3.4.3.1 Off-line Playout Delay Computation

Off-line playout delay computation algorithms are useful in providing baseline reference measurements for comparison with the fixed and predictive playout delay algorithms. The first algorithm provides the optimal talkspurt delay for an isolated talkspurt and the second provides the minimum average delay over several talkspurts.

3.4.3.1.1 Optimal Algorithm for an Isolated Talkspurt

It is important to first present a technique that provides the optimal playout delay for any isolated talkspurt, but is impractical since it is non-causal in nature [53][54]. This algorithm is not optimal when the talkspurt is considered with respect to the whole conference since playout delay variation is not considered. The optimal algorithm for an isolated talkspurt uses advance knowledge of all the delays in the upcoming talkspurt to predict a playout delay that will give a certain desired average packet loss rate

$$\hat{p}_k = d_k^{(i)} \quad (8)$$

The playout delay for talkspurt k , $\hat{p}_k$, is computed by determining the number of packets to be played, i , based on the desired average lost packet rate, alp

$$i = n_k(1 - alp) \quad (9)$$

3.4.3.1.2 Minimum Average Playout Delay

Moon et al. present a recursive algorithm for off-line determination of a lower bound for the optimum average playout delay without collision detection, given a particular packet loss rate, [53]. This algorithm predicts playout delays with less fluctuation that are optimal for a number of talkspurts, not just one. This algorithm will be used to train our HMM predictor in a later section.

The minimum average playout delay is computed by choosing i packets from the k -th to M -th talkspurts in a manner that minimizes the average playout delay.

$$D(k,i) = \begin{cases} 0, & \text{if } i = 0 \\ d_k^{(i)}, & \text{if } k = M \text{ and } i \leq n_M \\ \infty, & \text{if } k = M \text{ and } i > n_M \\ \min_{0 \leq j \leq i} \left((i-j)D(k+1, i-j) + jd_k^{(j)} \right) / i, & \text{otherwise} \end{cases} \quad (10)$$

where $D(k,i)$ is the minimum average playout delay when choosing i packets from the k -th to M -th talkspurts.

Moon et al. present another algorithm that computes the upper bound on the optimum average playout delay off-line with collision detection [53]. This algorithm, however, introduces much greater computational complexity.

3.4.3.1.3 Fixed Playout Delay

The fixed playout delay technique assumes that the range of network delays is predictable and a suitable static playout delay can be chosen. This technique is only valid for simple network topologies with measurable delay properties that follow a static delay distribution. Due to the restriction on its usage this technique is not

globally applicable, nor is it useful on the Internet where delays are highly variable. The advantage of this technique is that it is the simplest playout delay scheduling technique available.

The fixed playout delay technique states that voice packets arriving after the playout buffer delay are lost and cannot be recovered. The fixed playout delay algorithm fixes the time that an audio packet is played out relative to its generation time,

$$\hat{p}_k = C, \quad 1 \leq k \leq M \quad (11)$$

where C is a constant playout delay.

3.4.3.2 Predictive Playout Delay Algorithms

Predictive playout delay algorithms are more practical for playout delay determination in the Internet and most non-trivial corporate intranets than the fixed delay technique since they attempt to predict a playout delay that is an ideal compromise between the average end-to-end delay and the average number of lost packets. The predicted playout delay is adapted during silence periods based on an analysis of the delay data obtained in the talkspurt periods.

Adapting the playout delay during silence periods is the recommended technique for adjusting the playout delay since a small amount of silence compression or expansion is unnoticeable to the listener [53]. Adapting the playout delay during a talkspurt introduces gaps inside the talkspurt, which negatively affects the user's perception of the conference quality [53].

3.4.3.2.1 Causal Version of Optimal Algorithm

The optimal off-line algorithm, see section 3.4.3.1.1, can be made causal by simply delaying usage of delay information for the $k-1$ -th talkspurt to the next talkspurt, k -th talkspurt [54]. This transformation ensures causality, but the technique no longer predicts optimal playout delays.

$$\hat{p}_k = d_{k-1}^{(i)}, \quad 1 \leq k \leq M \quad (12)$$

The optimal playout delay observations can be filtered to prevent over-reacting to short lived phenomena in the last talkspurt [54]. A simple infinite impulse response (IIR) filter can be applied to smooth the playout delay estimate over several past talkspurts:

$$\hat{p}_k = \rho \hat{p}_{k-1} + (1 - \rho) d_{k-1}^{(i)}, \quad 1 \leq k \leq M \quad (13)$$

where $\hat{p}_k$ is the filtered playout delay estimate for the k -th talkspurt, and $0 < \rho < 1$.

3.4.3.2.2 Delay Statistics Algorithm

A well-known algorithm from Ramjee et al. [53][55] predicts playout delay based on network delay mean and standard deviation estimates. The original paper depicts the algorithm as talkspurt-unaware [55], however in [53] talkspurt awareness has been added. This thesis deals with algorithms appropriate for vocoders with silence suppression; therefore the latter will be presented.

$$\hat{d}_k^i = \alpha \hat{d}_k^{i-1} + (1 - \alpha) d_k^i, \quad 2 \leq i \leq n_k, 1 \leq k \leq M \quad (14)$$

$$\hat{d}_k^1 = \alpha \hat{d}_{k-1}^{n_{k-1}} + (1 - \alpha) d_k^1, \quad 1 \leq k \leq M \quad (15)$$

$$\hat{v}_k^i = \alpha \hat{v}_k^{i-1} + (1 - \alpha) |\hat{d}_k^i - d_k^i|, \quad 1 \leq i \leq n_k, 1 \leq k \leq M \quad (16)$$

where $\hat{d}_k^i$ is the delay mean estimate calculated from the i-th packet in the k-th talkspurt, $\hat{v}_k^i$ is the delay variation estimate calculated from the i-th packet in the k-th talkspurt and $\alpha = 0.998002$ [53][55]. The predicted playout delay for the k-th talkspurt is computed by:

$$\hat{p}_k = \hat{d}_k^{n_k} + \beta \hat{v}_k^{n_k}, \quad 1 \leq k \leq M \quad (17)$$

where β is chosen to be sufficiently large to produce few lost packets due to late arrival (typically $\beta = 4$ is used)

The aforementioned algorithm functions well if the delay statistics in the system change slowly over time. In reality short-lived spike phenomena exist which cause a sudden change in the packet delay distribution. [53][55] introduce a spike-detection algorithm that detects spikes and rapidly adjusts the delay mean computation, $\hat{d}_k^i$, during the spikes.

Pinto et al. [56] propose a technique similar to the one in this section except that delay statistics are collected based on the computed gap between the packet's playout time and arrival time. Matic et al. [57] propose another interesting statistical technique based on refining the method for predicting the interarrival time, network jitter, between several points on the Internet.

3.4.3.2.3 Packet Delay Distribution Algorithm

An interesting technique by Sreenan et al.[58] involves generating a histogram of delay values combined with an aging technique for past delay observations to create a packet delay distribution (PDD). A suitable playout delay

can then be selected by identifying the desired packet loss rate and finding an appropriate playout delay on the PDD curve, see Fig. 16.

Fig. 16. Packet delay distribution (PDD)

The histogram technique increases the statistical accuracy over the technique from section 3.4.3.2.2 by not assuming that the packet delay distribution will assume a Gaussian form. The aging parameter must be varied to react appropriately to network conditions [58]. The histogram technique is designed to not react too quickly to short-lived network jitter [58], unlike the spike-detection algorithm. The authors claim that the packet burst losses experienced by the histogram algorithm were not sufficiently large to mandate over-compensation when network delay spikes occur.

A difference in design philosophy is evident from the work performed by the authors of [58] and the authors of [53][55]. One philosophy is to react quickly to short-lived network delay spikes and minimize packet loss. The other philosophy is to react less quickly to short-lived network delay spikes and incur some extra packet loss during periods of network delay spikes.

Reacting quickly to spikes causes a large variation in the playout delays, which can cause listener discomfort due to the large adjustments required during silence periods. Reacting slowly to spikes can cause speech to breakup, but if the spikes are infrequent enough then the effect will not make listening too uncomfortable. If the network delay spike spans only one talkspurt, then the data cannot be recovered by either technique and the spike-detection technique runs the risk of over-reacting in the next talkspurt. Formal listening tests using both philosophies under a wide variety of network conditions would be required to determine which technique is optimal.

4 Spatial Audio and Echo Cancellation

4.1 Synthetic Stereo Acoustic Echo Cancellation

4.1.1 Overview

This thesis proposes a novel acoustic echo cancellation structure intended for multiple participant, full-duplex, hands-free stereo voice over Internet protocol (VoIP) conferencing. A synthetic stereo image is generated through the use of spatialization functions, which are used to assist the near-end user in distinguishing between far-end talkers. The proposed synthetic stereo structure, which uses a single echo canceler per spatial region, is compared to structures with a single echo canceler per channel and to stereophonic echo cancelers. The proposed structure does not suffer from correlation of the reference signals, which can cause system misconvergence in structures that use a single canceler per channel. Correlation of the reference signals is eliminated since monaural signals are transmitted at the far-end and static spatialization functions are used at the near-end to allocate each participant to a spatial region.

4.1.2 Synthetic Stereo Acoustic Echo Cancellation Structure

The synthetic stereo structure with a single echo canceler per spatial region takes the acoustic echo canceler reference before spatialization of the input signals, see Fig. 17. This differs from the per-channel approach, see section 3.3.1, which takes the reference signal after spatialization. Each of the M spatial regions in the per-spatial region structure has its own spatialization coefficients and echo canceler. The echo cancelers for each of the M spatial regions, $\hat{\mathbf{h}}_1(n)$, $\hat{\mathbf{h}}_2(n)$, $\dots$ $\hat{\mathbf{h}}_M(n)$, each have T_F taps assigned to their echo path models. The total number of echo canceler

coefficients and taps in the system is, therefore, MT_F . For example, if there are four spatial regions in the conference, $M=4$, then there will be four acoustic echo cancelers each with its own set of acoustic echo canceler coefficients, meaning that there is a total of $4T_F$ taps in the system.

Fig. 17. Synthetic stereo structure with a single canceler per-spatial-region

The length of the adaptive echo path impulse responses, in the following equations, are denoted by T_F , the length of reception room impulse responses are denoted by T_R , and the length of the spatialization function impulse responses are denoted by T_S , where $T_S < T_F$.

There are $N+1$ participants in the VoIP conference with N far-end signals,

$$s_j(n), \quad j = 1, 2, \dots, N \quad (18)$$

Each of the N far-end signals is mapped into one of M spatial regions to form the far-end spatial signals, $v_i(n)$, as shown in Fig. 12:

$$v_i(n) = [v_i(n) \quad v_i(n-1) \quad \dots \quad v_i(n-T_F+1)]^T, \quad i = 1, 2, \dots, M \quad (19)$$

where $v_i(n)$ is the sum of all the far-end signals mapped to that spatial region.

The adaptive echo path model for a single spatial region in the per-spatial-region structure is denoted by, $\hat{\mathbf{h}}_i(n)$:

$$\hat{\mathbf{h}}_i(n) = [\hat{h}_{i,0}(n) \quad \hat{h}_{i,1}(n) \quad \dots \quad \hat{h}_{i,T_F-1}(n)]^T, \quad i = 1, 2, \dots, M. \quad (20)$$

The echo path model, $\hat{\mathbf{h}}_i(n)$, must adapt when any of the far-end spatial signals, $v_i(n)$, in the spatial region is active.

The number of far-end participants in the conference, N , does not determine the number of spatial regions, M . Each of the N participants in the conference is assigned one of the M spatial regions. Spatial region i is realized through its left and right channel spatialization functions, $\mathbf{g}_{Li}$ and $\mathbf{g}_{Ri}$.

$$\mathbf{g}_{Li} = [g_{Li,0} \quad g_{Li,1} \quad \dots \quad g_{Li,T_S-1} \quad \mathbf{0}_{T_F-T_S}]^T, \quad i = 1, 2, \dots, M \quad (21)$$

$$\mathbf{g}_{Ri} = [g_{Ri,0} \quad g_{Ri,1} \quad \dots \quad g_{Ri,T_S-1} \quad \mathbf{0}_{T_F-T_S}]^T, \quad i = 1, 2, \dots, M \quad (22)$$

where T_S is the length of the spatialization functions and $\mathbf{0}_{T_F-T_S}$ is a zero vector of length $T_F - T_S$. The system designer must determine the appropriate spatial mapping but it is recommended that it correspond to the video image. To ensure scalability of the system, when the number of participants becomes greater than M , any additional participants are assigned to one of the existing spatial regions. It is acceptable to have more than one participant in the same spatial region. Fig. 12 gives an example of participant spatialization for $N = 6$ participants with $M = 4$ spatial regions.

The left loudspeaker output, $\mathbf{x}_L(n)$, is formed by convolving the far-end spatial signals, $\mathbf{v}_i(n)$, by the appropriate left channel spatialization function for the spatial region, $\mathbf{g}_{Li}$

$$\mathbf{x}_L(n) = [x_L(n) \quad x_L(n-1) \quad \cdots \quad x_L(n-T_R+1)]^T \quad (23)$$

$$x_L(n) = \sum_{i=1}^M \mathbf{v}_i^T(n) \mathbf{g}_{Li} \quad (24)$$

where T_R is the length of the reception room impulse response. The right loudspeaker output, $\mathbf{x}_R(n)$, is formed by convolving the far-end talker signals, $\mathbf{v}_i(n)$, by the appropriate right channel spatialization function for the spatial region, $\mathbf{g}_{Ri}$

$$\mathbf{x}_R(n) = [x_R(n) \quad x_R(n-1) \quad \cdots \quad x_R(n-T_R+1)]^T \quad (25)$$

$$x_R(n) = \sum_{i=1}^M \mathbf{v}_i^T(n) \mathbf{g}_{Ri} \quad (26)$$

The primary signal, $d(n)$, received by the microphone is

$$d(n) = \mathbf{x}_L^T(n) \mathbf{h}_L(n) + \mathbf{x}_R^T(n) \mathbf{h}_R(n) \quad (27)$$

where $\mathbf{h}_L(n)$ is the left channel reception room impulse response and $\mathbf{h}_R(n)$ is the right channel reception room impulse response:

$$\mathbf{h}_L(n) = [h_L(n) \quad h_L(n-1) \quad \cdots \quad h_L(n-T_R+1)]^T \quad (28)$$

$$\mathbf{h}_R(n) = [h_R(n) \quad h_R(n-1) \quad \cdots \quad h_R(n-T_R+1)]^T. \quad (29)$$

If only one far-end participant is active, multiple active participants will be discussed in 4.1.3, in the conference with signal, $s_1(n)$, and is mapped into spatial region 1 to produce far-end spatial signal, $v_1(n)$, then the only active acoustic echo canceler is the one for that spatial region, $\hat{\mathbf{h}}_1(n)$. The acoustic echo canceler will converge to echo path model, $\hat{\mathbf{h}}_1(n)$

$$\hat{\mathbf{h}}_1(n) = \mathbf{g}_{L1} * \mathbf{h}_L(n) + \mathbf{g}_{R1} * \mathbf{h}_R(n) \quad (30)$$

where $*$ is the symbol used for convolution. The simulation, detailed in Section 4.1.5, shows convergence for a single active spatial region as in (30) and multiple active spatial regions.

It can be seen from (30) that the problem of convergence to multiple instantaneous solutions without identifying the correct echo path has been eliminated. There is no longer any cross-correlation in the reference signals to the

acoustic echo canceler. The system will converge to the correct combination synthetic stereo and echo path solution, $\hat{\mathbf{h}}_i(n)$, and will have the standard tracking performance of the simple and straightforward NLMS algorithm. There is no need for a complex algorithm such as the two-channel RLS or FAP, unless faster tracking is required for reasons other than the synthetic stereo acoustic echo cancellation structure. Also, no nonlinearities or uncorrelated noise sources need to be added to the channels to assure convergence.

This structure does not require that there is energy in both channels because the echo cancelers are allocated based on spatial regions. It is acceptable to perform spatialization using only ILD techniques even in the extreme lateralization case.

4.1.3 Multiple Active Spatial Regions

If it is desirable to have simultaneous echo canceler adaptation for multiple near-end spatial regions then the multi-reference LMS or NLMS algorithm must be employed. The multi-reference NLMS is given by the following equations [59]:

$$e(n) = d(n) - \sum_{i=1}^M \mathbf{v}_i^T(n) \hat{\mathbf{h}}_i(n) \quad (31)$$

$$\hat{\mathbf{h}}_i(n+1) = \hat{\mathbf{h}}_i(n) + \mu(n) \mathbf{v}_i(n) e(n) \quad (32)$$

where the step-size parameter, $\mu(n)$, for NLMS is defined as:

$$\mu(n) = \frac{\alpha}{\epsilon + T_F \left(\sum_{i=1}^M \mathbf{v}_i^2(n) \right)} \quad (33)$$

where $0 < \alpha < 2$ and $\epsilon \ll T_F \left(\sum_{i=1}^M v_i^2(n) \right)$.

The simulation in Section 4.1.5 shows convergence for a single active spatial region, two active regions and four active regions. It can be seen that when more than one spatial region is active, the system converges slower and the maximum achievable ERLE is lower. The system converges slower because the step-size is reduced as shown in (33). The maximum ERLE is lower because more energy is present in the tail of the impulse responses that are not modeled by the truncated adaptive filters. The maximum theoretical steady-state ERLE for an adaptive echo path model's impulse response truncated at tap n can be determined by computing the ratio of the total impulse power to tail power [5].

$$\frac{\text{TIP}}{\text{TP}}(n, T_R) = 10 \log \frac{\sum_{i=0}^{T_R-1} \mathbf{h}^2(i)}{\sum_{i=n}^{T_R-1} \mathbf{h}^2(i)} \text{ (dB)} \quad (34)$$

Echo canceler convergence is possible for multiple active spatial regions, but it may be desirable to decrease computational complexity by allowing only one echo canceler to be active at a time. If the system is designed to only adapt for one spatial region at a time then the chosen region should be the one with the largest signal power. Coefficients are frozen in the echo cancelers for active spatial regions with lower power, but convolution must still be performed using the frozen coefficients to reduce echo.

The system can determine which participants are active at any time by using information from the voice activity detector (VAD) which can be implemented locally or may already be present in the speech coder being used to code the far-end participant's voice. If the far-end participant is transmitting a silence frame, as determined by the VAD, then the participant will be marked inactive by the near-end per-spatial-region structure. If more than one participant is active then the multi-reference NLMS with multiple active participants or a coefficient-freezing scheme with a single active participant must be used.

On detection of conventional double talk, between the far-end and near-end users, all the echo canceler coefficients must be frozen as in a standard echo canceler implementation.

4.1.4 Comparison to Other Stereo Structures

The algorithm required for the per-spatial-region structure is of lower computational complexity than for the per-channel structure or true stereophonic structure when only one spatial region is active at a time. The problem of cross-correlation of the reference signals is eliminated, without requiring the addition of a nonlinear distortion to the channels, and faster tracking is possible. No convergence difficulties result from using ILD techniques for spatialization and there is no need for ITD, which can produce "phasiness".

The per-spatial-region structure also requires fewer echo cancelers than the per-channel structure if the number of active echo cancelers in the conference is limited to one. This is a reasonable assumption since there is normally only one active

spatial region unless a talker from a different spatial region interrupts. If the per-spatial-region structure is allowed to converge for multiple active talkers (see Section 4.1.3) then the number of echo cancelers required is greater than that for the per-channel structure when there are more than two active spatial regions. The maximum number of spatial regions is limited by the horizontal localization blur and video display space as discussed in Section 3.3.

Some additional overhead is required to switch active spatial regions when only one echo canceler is allowed to adapt at a time, but this overhead is very low. A further reduction in computational complexity is achieved by using the NLMS algorithm instead of the more complex recursive least squares (RLS) or fast affine projection (FAP) algorithms to achieve convergence.

The per-spatial-region structure reconverges when the room response changes only for the currently active spatial region. If a near-end spatial region is inactive and the room response changes, then when activity in that spatial region resumes there will be some echo in that region until the canceler for that region reconverges. The tracking and convergence ability of the per-spatial-region structure is very good, however, which quickly reduces annoying echo from room response changes.

The per-channel structure, however, does not require echo canceler switching and it allows all spatial regions to reconverge with multiple active spatial regions. If the per-spatial-region structure is used in the multiple active spatial regions mode (see Section 4.1.3) then all spatial regions reconverge when the room response changes, at the expense of having more active echo cancelers. There is a tradeoff

between the continuous tracking ability of multiple active echo cancelers versus the more rapid convergence and tracking, lower computational complexity, and higher maximum steady-state ERLE of a single active canceler.

True stereophonic structures must track changes in both the transmission room impulse responses, $\mathbf{g}_L(n)$ and $\mathbf{g}_R(n)$, and reception room transfer functions, $\mathbf{h}_L(n)$ and $\mathbf{h}_R(n)$. Synthetic stereo structures need only track changes in the reception room, since the spatialization functions are static. There is no need to add a front-end preprocessor to decorrelate the reference signals.

4.1.5 Simulation

Real room impulse responses, $\mathbf{h}_L(n)$ and $\mathbf{h}_R(n)$, were measured through our standard VoIP setup that consisted of multimedia loudspeakers and a microphone. The responses, $\mathbf{h}_L(n)$ and $\mathbf{h}_R(n)$, shown in Fig. 18, were measured for 1000 taps, at 16000Hz, and a sound pressure level of 100 dB.

Fig. 18. Measured loudspeaker to microphone impulse response

The spatialization functions used in the simulation, $\mathbf{g}_{Li}$ and $\mathbf{g}_{Ri}$ for $i = 1, 2, \dots, M$, are based on intensity stereo techniques, i.e., interaural level differences (ILDs), to localize the talkers with some additional amounts of ITD to further enhance the effect. The functions were 24-tap FIR filters, which correspond to a maximum ITD of ± 1.5 ms at a 16 kHz sampling rate.

Simulations of synthetic stereo acoustic echo cancellation were performed using independent Gaussian white noise signals, $v_i(n)$, for $i = 1, 2, \dots, M$, to represent the active spatial regions. Fig. 19 shows synthetic stereo echo canceler convergence for a 500-tap ($T_F = 500$) adaptive normalized least mean square (NLMS) filter for a

single active spatial region ($M = 1$), two active regions ($M = 2$) and four active regions ($M = 4$). The step-size for each of the scenarios remained constant.

Fig. 19. Synthetic stereo performance with a single canceler per-spatial-region

The maximum theoretical steady-state ERLE, (34), for this 500-tap adaptive filter with a small step-size, $\alpha \ll 2$, is 25.5 dB for one active spatial region, 23.3dB for two active regions and 20.4dB for four active regions, all of which were confirmed by simulation. To increase the convergence rates, a step-size of $\alpha=1$ was used to produce Fig. 19. As can be seen, this step-size results in increased misadjustment and therefore a reduction in the steady-state ERLE.

A simulation was run to determine the convergence of the synthetic stereo acoustic echo cancellation structure with the NLMS algorithm using speech as the reference signal. The simulation was run with a single active male talker, ($M = 1$), and two active male talkers, ($M = 2$), see Fig. 20. Performance is as expected for an NLMS algorithm [3]. Convergence is slower with speech than with white noise since short-term speech segments are much more narrowband than the white noise signal used in Fig. 19. The same step-size, $\alpha = 1$, was used in Fig. 20 and the same maximum ERLEs are expected, 25.5 dB for one active spatial region and 23.3dB for two active regions. The graph points were averaged over a duration of 200ms, since only one set of source files was used in the simulations.

Fig. 20. Synthetic stereo performance with a single canceler per-spatial-region and speech reference signals

4.2 Synthetic Surround Sound Acoustic Echo Cancellation

4.2.1 Overview

This thesis proposes an acoustic echo cancellation structure for hands-free synthetic surround sound applications, such as multiple participant conferencing, and virtual reality applications that extends synthetic stereo acoustic echo cancellation to surround sound. Voice over Internet protocol (VoIP) and other virtual reality applications can benefit from the addition of 3D spatial audio generated by more than two loudspeakers. When full-duplex audio is present in a system, acoustic echo cancellation is required to eliminate the feedback echo path. The acoustic echo cancellation structure proposed in this chapter is based on the acoustic echo canceller per spatial region allocation scheme previously introduced by the authors for two channel synthetic stereo, section 4.1. This section will show that the spatial region allocation scheme is extensible to any number of channels, which makes it extremely versatile and flexible, especially for surround sound audio. Microsoft Direct X 7 [60], a commonly used application programmer interface (API), was used in our simulations to generate the 3D spatial audio on a PC.

When the single canceler per channel structure [43][44] is extended to more than two channels there is an increasing amount of cross-correlation in the reference signals, which makes convergence increasingly difficult as the number of channels increases. The number of acoustic echo cancellers required for the per-channel structure is equal to the number of channels in the system. This does not apply to the proposed per spatial region structure which has an acoustic echo canceller for

each spatial region and does not suffer from cross-correlation of the reference signals.

4.2.2 Synthetic Surround Sound Acoustic Image

Surround sound spatial imaging is important in virtual reality applications, such as multiple participant voice conferencing over Internet protocol (IP) networks, to enhance the user's ability to distinguish between remote participants or to provide spatial cues for virtual reality. In VoIP applications, it is not always easy to distinguish between participants based on their voice alone, especially if the listener is not familiar with the talker's voice. Surround sound for virtual reality enhances the experience by adding spatial cues, which complement the graphical user environment.

Stereo loudspeakers are able to reproduce limited spatial imaging for multiple participant conferencing. Virtual sources, far-end talkers, can only be convincingly positioned in 3D acoustic space over two loudspeakers through the use of carefully designed head-related transfer functions, HRTF, and loudspeaker decorrelation functions. 3D positioning using this methodology requires complicated head-tracking mechanisms to ensure that the effect does not deteriorate with listener head movement. The addition of surround channels, loudspeakers behind the listener, however, can be used to create realistic spatial audio with a larger sweet spot than two channel methods and is the subject of this chapter. A surround sound acoustic echo canceler must deal with the addition of these extra surround channels and the single canceler per spatial region acoustic echo cancellation structure, proposed in 4.1, can be adapted to this task.

To facilitate distinguishing among participants in a multiple participant conference, the chosen spatialization method must distribute the participant's audio signals in a manner designed to minimize localization blur and must not sound awkward to the listener. (Localization blur is minimized by ensuring that virtual sources are well separated from one another.)

Fig. 21 shows five virtual source positions, which can be easily generated by the Microsoft DirectX API.

Fig. 21. Near-end talker spatialization

This section will explain how the acoustic echo canceller for a surround sound system with three or more loudspeakers operates under a spatial region echo canceller allocation.

4.2.3 Synthetic Surround Sound Acoustic Echo Cancellation Structure

This thesis proposes performing synthetic surround sound acoustic echo cancellation using the single canceler per spatial region structure. This structure first takes the far-end participant's signal, $s_j(n)$, and maps it into the appropriate spot in the near-end listener's acoustic space, $v_i(n)$, see Fig. 22. The position of the virtual source will likely correspond to far-end participant's location on the video display terminal for VoIP conferencing or their relative 3D position in a virtual reality system. If more than one far-end participant is mapped into the same spatial region, then the signal $v_i(n)$, is a combination of all the far-end participants allocated to the spatial region. The number of far-end participants, N , therefore, does not determine the number of spatial regions, M , since there may be more than one participant assigned to the same spatial region.

The acoustic echo canceller then takes the mapped signal, $v_i(n)$, as its reference before spatialization. This eliminates the fundamental problem of acoustic echo canceller misconvergence [4][46].

The mapped signals are then steered into the appropriate spot in the near-end participant's 3D acoustic space by convolving the mapped signal with the appropriate set of spatialization functions, $\mathbf{h}_{FL}(n)$, $\mathbf{h}_{FR}(n)$, $\mathbf{h}_{BL}(n)$ and $\mathbf{h}_{BR}(n)$. There are

unique spatialization functions associated with each spatial region for each loudspeaker. When four loudspeakers are in use the spatialization function set consists of four impulse responses per spatial region. For example, if two spatial regions are in use then a total of eight impulse responses are required.

Fig. 22. Synthetic surround acoustic echo cancellation structure

When the synthetic surround sound acoustic echo cancellation structure is being used for VoIP conferencing the position of the 3D virtual source does not change throughout the duration of the conference. Not allowing a virtual source to change positions during a conference makes it easier for the listener to distinguish between

participants. When the synthetic surround sound structure is used for virtual reality systems, the virtual sources can move in the environment, changing the loudspeaker spatialization function set. The acoustic echo canceller is able to track the changes in the spatialization functions and the impulse response of the reception room.

Each spatial region is generated by implementing a set of spatialization functions for each loudspeaker. The virtual sources used in this chapter were created using the Microsoft DirectX 7 API. Fig. 23 shows the loudspeaker spatialization impulse responses used to create the impression that the talker appears 45 degrees to the front-right, spatial region 3 in Fig. 21. The DirectX API allows the user to choose several options for positioning a virtual source in acoustic space. The spatial audio is generated in 3D buffers with parameters relative to a 3D listener.

Fig. 23. Front-right position loudspeaker spatialization impulse responses

4.2.4 Analysis

The synthetic surround acoustic echo cancellation structure relies on a separate acoustic echo canceller for each spatial region. If there are M spatial regions present then there must be M acoustic echo cancellers, $\hat{\mathbf{h}}_1(n)$, $\hat{\mathbf{h}}_2(n)$, ... $\hat{\mathbf{h}}_M(n)$. This assignment is scalable since there are only a limited number of spatial positions that the listener can distinguish between due to localization blur. This establishes a practical limit of the number of acoustic echo cancellers present in the system.

The number of echo canceller coefficients and taps in the system is dependent only on the number of spatial regions being used. The number of coefficients and taps is not dependent on the number of channels being used in the surround sound system, unlike the per channel structure. If five spatial regions are being used, $M=5$, then a total of $5T_F$ coefficients and taps will be required regardless of the number of surround sound channels, where T_F is the length of the adaptive echo path impulse response.

The N far-end talkers in an $N + 1$ participant conference or virtual reality system are denoted by $s_j(n)$:

$$s_j(n), \quad j = 1, 2, \dots, N \quad (35)$$

These far-end participants are mapped into one of M near-end spatial regions, to form the acoustic echo canceller reference for spatial region i ,

$$\mathbf{v}_i(n) = [v_i(n) \quad v_i(n-1) \quad \dots \quad v_i(n-T_F+1)]^T, \quad i = 1, \dots, M \quad (36)$$

where $v_i(n)$ is the sum of all the far-end signals mapped to that spatial region.

The adaptive echo path model for spatial region, i , is denoted by, $\hat{\mathbf{h}}_i(n)$:

$$\hat{\mathbf{h}}_i(n) = [\hat{h}_{i,0}(n) \quad \hat{h}_{i,1}(n) \quad \dots \quad \hat{h}_{i,T_F-1}(n)]^T, \quad i = 1, \dots, M \quad (37)$$

If four surround sound channels are being used in the system then the spatial regions are realized through the front left, front right, back left and back right

spatialization functions, $\mathbf{g}_{FLi}$, $\mathbf{g}_{FRi}$, $\mathbf{g}_{BLi}$ and $\mathbf{g}_{BRi}$. For example, the spatialization function for the front left channel is

$$\mathbf{g}_{FLi} = [\mathbf{g}_{FLi,0} \quad \mathbf{g}_{FLi,1} \quad \cdots \quad \mathbf{g}_{FLi,T_S-1} \quad \mathbf{0}_{T_F-T_S}]^T, \quad i = 1, \dots, M \quad (38)$$

where T_S is the length of the spatialization functions and $\mathbf{0}_{T_F-T_S}$ is a zero vector of length $T_F - T_S$.

The loudspeaker outputs, $\mathbf{x}_{FL}(n)$, $\mathbf{x}_{FR}(n)$, $\mathbf{x}_{BL}(n)$ and $\mathbf{x}_{BR}(n)$ are formed by convolving the mapped far-end signals, $v_i(n)$, by the appropriate spatialization function for the channel and spatial region. For example the front left loudspeaker output, $\mathbf{x}_{FL}(n)$, is formed by convolving it with the front left spatialization function, $\mathbf{g}_{FLi}$, for the appropriate spatial region i

$$\mathbf{x}_{FL}(n) = [x_{FL}(n) \quad x_{FL}(n-1) \quad \cdots \quad x_{FL}(n-T_R+1)]^T \quad (39)$$

$$x_{FL}(n) = \sum_{i=1}^M v_i^T(n) \mathbf{g}_{FLi} \quad (40)$$

where T_R is the length of the reception room impulse response. The loudspeaker outputs for the remaining channels are formed in a similar fashion.

For four channel surround sound the primary signal, $d(n)$, received by the microphone is

$$\begin{aligned} d(n) = & \mathbf{x}_{FL}^T(n) \mathbf{h}_{FL}(n) + \mathbf{x}_{FR}^T(n) \mathbf{h}_{FR}(n) + \mathbf{x}_{BL}^T(n) \mathbf{h}_{BL}(n) \\ & + \mathbf{x}_{BR}^T(n) \mathbf{h}_{BR}(n) \end{aligned} \quad (41)$$

where the reception room impulse responses are of length T_R and denoted by $\mathbf{h}_{FL}(n)$, $\mathbf{h}_{FR}(n)$, $\mathbf{h}_{BL}(n)$ and $\mathbf{h}_{BR}(n)$. For example, the front left reception room impulse response is given by:

$$\mathbf{h}_{FL}(n) = [h_{FL}(n) \ h_{FL}(n-1) \ \dots \ h_{FL}(n-T_R+1)]^T \quad (42)$$

The adaptive echo path model for spatial region, i , is therefore a sum of the convolutions of the loudspeaker spatialization function for the given spatial region and the respective reception room impulse response:

$$\begin{aligned} \hat{\mathbf{h}}_i(n) = & \mathbf{g}_{FLi} * \mathbf{h}_{FL}(n) + \mathbf{g}_{FRi} * \mathbf{h}_{FR}(n) + \mathbf{g}_{BLi} * \mathbf{h}_{BL}(n) \\ & + \mathbf{g}_{BRi} * \mathbf{h}_{BR}(n) \end{aligned} \quad (43)$$

where $*$ is the symbol used for convolution.

4.2.5 Comparing Single Channel to Synthetic Multi-Channel Acoustic Echo Cancellation

This section shows that when the single canceler per spatial region structure is used in a monaural, stereo or surround sound scenario with only one active spatial region at a time, the acoustic echo canceller performs identically with only a change in the echo path model, $\hat{\mathbf{h}}(n)$, see section 4.1.2. This indicates that acoustic echo cancellation convergence and tracking performance is independent of the number of loudspeakers. When the proposed structure is used with multiple channels the echo path model changes from including only the reception room impulse response, in the monaural case with no spatialization, to having spatialization information included.

For single channel applications the acoustic echo canceller simply converges to the reception room impulse response:

$$\hat{\mathbf{h}}(n) = \mathbf{h}(n) \quad (44)$$

For synthetic stereo applications with the single canceler per spatial region structure the acoustic echo canceller converges to the sum of the left loudspeaker spatialization function for region i convolved with the left loudspeaker reception room impulse response and the right loudspeaker spatialization function for region i convolved with the right loudspeaker reception room impulse response:

$$\hat{\mathbf{h}}_i(n) = \mathbf{g}_{L_i}(n) * \mathbf{h}_L(n) + \mathbf{g}_{R_i}(n) * \mathbf{h}_R(n) \quad (45)$$

For synthetic surround applications with the single canceler per spatial region structure and four loudspeakers, the acoustic echo canceller converges to the sum of all four loudspeaker spatialization functions for region i convolved with their respective reception room impulse responses (43).

It is easy to see that acoustic echo canceler allocation by spatial region works well for any number of loudspeakers and the convergence rate and tracking performance is equivalent as long as only one spatial region is allowed to be active at one time.

4.2.6 Simulation

The simulations performed for this chapter required measuring the reception room impulse responses for a synthetic stereo configuration with two loudspeakers,

$\mathbf{h}_L(n)$ and $\mathbf{h}_R(n)$, and a synthetic surround configuration with four loudspeakers, $\mathbf{h}_{FL}(n)$, $\mathbf{h}_{FR}(n)$, $\mathbf{h}_{BL}(n)$ and $\mathbf{h}_{BR}(n)$, see Fig. 24. Spatialization functions for two channel stereo, $\mathbf{g}_{Li}$ and $\mathbf{g}_{Ri}$, and four channel surround sound, $\mathbf{g}_{FLi}$, $\mathbf{g}_{FRi}$, $\mathbf{g}_{BLi}$ and $\mathbf{g}_{BRi}$, were generated using the Microsoft DirectX API [60]. Fig. 23 shows the spatialization impulse responses for the front right spatial region generated by the DirectX API.

Fig. 24. Reception room impulse responses

Fig. 25 compares the convergence rate for a synthetic stereo and a four channel synthetic surround sound acoustic echo canceller configuration with only one active

spatial region at a time. The input signal to the single spatial region, $\nu_1(n)$, was independent Gaussian white noise. It can be seen that both the convergence rate and the maximum echo return loss enhancement (ERLE) [5] are equivalent for each configuration. This simulation can be extended to the single channel case with similar results, see section 3.3. The simulation results for Fig. 26 are valid for 25000 iterations at a sampling rate of 16kHz using the normalized least mean square (NLMS) algorithm with a step-size of 0.5.

Fig. 25. Synthetic stereo versus synthetic surround

If more than one spatial region is allowed to be active at a time then convergence is still possible but the rate of convergence decreases. Fig. 26 shows the synthetic

surround sound acoustic echo canceller configuration with one active spatial region, two active spatial regions and four active spatial regions. This simulation shows that the rate of convergence decreases when more than echo canceller is allowed to be active at any one time, but the maximum ERLE value is still attainable. To maximize convergence rate and tracking it may be desirable to only allow one acoustic echo canceler to converge at a time based on the spatial region with the largest amount of energy although convergence is possible for multiple active spatial regions. The simulation results for Fig. 26 are shown for 60000 iterations at a sampling rate of 16kHz using the normalized least mean square (NLMS) algorithm with a step-size of 0.5 with independent Gaussian white noise input signal, $v_1(n)$.

Fig. 26. Multiple active spatial regions

5 Near-Field Microphone Array and Synthetic Stereo Acoustic Echo Cancellation

5.1 Overview

This thesis proposes a synthetic stereo acoustic echo cancellation structure with microphone array beamforming for multiple participant voice over Internet protocol (VoIP) conferencing. The proposed acoustic front-end combines a synthetic stereo structure, acoustic echo canceler and microphone array beamformer. Synthetic stereo is used to add spatial information to signals from multiple participants that facilitates distinguishing between participants. Microphone array beamforming is then combined with the single canceler per spatial region echo cancellation structure to reduce noise and reverberation for near-end speech pickup and to cancel echo. This chapter assumes that a fixed beamformer is used, however, it is possible to use a steerable beamformer that could automatically track the user's position and provide noise and reverberation reduction anywhere in the room.

5.2 Near-Field Optimized Microphone Array

This chapter deals with synthetic stereo VoIP using a microphone array beamformer to provide additional echo suppression and end-to-end conference gain. The VoIP hardware setup requires multiple microphone transducers in addition to the standard stereo loudspeaker configuration, see Fig. 27.

As will be discussed in more detail in Section 7.1, the amount of echo cancellation required for a specific one-way propagation delay increases as the delay increases [1][2]. The maximum amount of acoustic echo cancellation provided by a

typical acoustic echo canceler is not sufficient for long VoIP delays [1]. If the echo canceler is not able to achieve the echo cancellation target then the remaining echo must be either attenuated using switched-loss [1] or by combining the acoustic echo canceler with a microphone array beamformer. Adding attenuation in the signal path causes reduced signal gain, which is detrimental to the conference. This chapter describes the later approach, which involves using a microphone array beamformer to increase the amount of echo suppression through echo cancellation and microphone array beamforming, which are full-duplex techniques that do not reduce the signal path gain.

Fig. 27. Physical arrangement of microphone array and multimedia loudspeakers

The acoustic front-end described in this chapter uses an optimized microphone beamformer array to focus on the location of the near-end talker. The microphone array reduces acoustic echo, and rejects background noise and reverberation.

Synthetic stereo is used to present multiple far-end participants to the near-end user in a manner that permits perceptual localization of the different participants [4][42][43][44], see section 4.1. The synthetic stereo structure is combined with the single canceler per spatial region echo cancellation structure to achieve good echo canceler performance with minimal computational complexity [4][42]. The combination of the microphone beamformer and the echo cancellation structure provides an increase in attainable echo return loss enhancement (ERLE) for the system and improved near-end speech pickup. The physical configuration for a VoIP system that employs a computer terminal with multimedia speakers and a microphone array is shown in Fig. 27.

5.3 Microphone Array Beamformer

This thesis proposes a fixed beamformer design that provides optimum performance using apriori knowledge of the signal and noise fields where the array will be deployed. The beamformer coefficients do not vary with noise conditions, therefore, a fixed beamformer can be designed to provide distortion-free reception of the desired signal with no attenuation. The consequence of a fixed beamformer is that the user must remain within the beam-pattern of the microphone array or their

speech will be attenuated. However, the beam-pattern is chosen in such a manner that the position, with the user in front of the monitor, is ideal for VoIP conferencing.

A fixed beamformer designed for VoIP applications can be designed using the near-field optimization described in [61]. This near-field optimization provides additional array gain at low frequencies where the array size is a smaller fraction of a wavelength by adjusting the beamformer filter coefficients, $\mathbf{b}_k$, where $k = 1, 2, \dots, K$ and K is the number of microphones in the array. The optimization is performed using simplified models of the desired signal and noise. The desired signal, near-end user speech, is represented as an ideal point source. The noise field is modeled as spherically isotropic noise.

Array optimization can produce an array that is extremely sensitive to sensor noise and other types of uncorrelated errors [61]. To limit this sensitivity, the beamformer was designed using a constrained optimization that limits the white noise gain to a pre-determined level. This procedure was used to limit the white noise gain to 0 dB. The resultant beamformer is no more sensitive to sensor noise and errors than a single microphone.

Optimum beamformer weights are, in general, only valid for a single frequency but speech signals are not confined to a single frequency. A speech bandwidth array must, therefore, be optimized at several frequencies within the range of interest. A practical implementation can then be realized by designing an FIR filter that provides the correct amplitude and phase at the selected frequencies.

The optimum beamformer weights were calculated for frequencies in the range of 300 Hz to 3000 Hz. The collection of the complex weights within the desired frequency range provides the necessary transfer function for each sensor. The required filters are not linear phase or minimum phase, and conventional filter design methods are not directly applicable. A more general complex approximation problem must be solved to design these FIR filters [62]. A simple and effective technique based on least-squared criteria is the method chosen for this thesis.

The number of taps chosen for the sensor filters is an important parameter in determining the accuracy with which the optimum beamformer weights can be implemented. The filter design problem is similar to that encountered when implementing fractional time delays [63], since the FIR filters are used to approximate both an arbitrary phase and amplitude. This chapter uses 60-tap FIR filters for the beamformer weights that provide both the desired array gain and a target frequency response, which is flat to within ± 0.5 dB. The optimized array gain for a spherically isotropic noise field is shown in Fig. 28.

Fig. 28. Near-field gain of beamformer

5.4 Array Response to Near-End Talker

Array performance was simulated by using the impulse response measurements for a multimedia computer monitor affixed with a microphone array as shown in Fig. 27. A head and torso simulator, equipped with an artificial mouth, was used to represent the near-end talker. The near-end speech simulator approximates the acoustic properties of the human voice during speech for this near-field application.

The beamformer response to a given acoustic source was simulated by filtering the individual impulse responses, $\mathbf{h}_{Lk}(n)$ and $\mathbf{h}_{Rk}(n)$, from each stereo loudspeaker to each microphone transducer in the array. The impulse responses are filtered by

the appropriate sensor FIR filter, $\mathbf{b}_1$, $\mathbf{b}_2$, ..., $\mathbf{b}_K$ and summed to give the beamformer response. The beamformer array responses were normalized to provide the same target signal level as a single microphone to permit direct comparison between the two.

Fig. 29 shows the array responses to the near-end talker. The energy decay curve shows the change in reverberant energy in the impulse response with time. The array impulse response, Fig. 29, shows a more rapid decay than the single microphone response. The array provides a reduction in the reverberation for the near-end talker resulting in a higher perceived audio quality.

The array provided 8.4 dB of ERLE for far-end echo, which can be used to increase the overall acoustic echo cancellation. The ERLE was computed by determining the total impulse power in the array response to the sum of the left and right simulated stereo signals. The total impulse power was then compared to that obtained for a single microphone to get the ERLE for the array.

Fig. 29. Energy decay for near-end talker

5.5 Synthetic Stereo

In sections 3.3 and 4.1, this thesis introduced the notion of the synthetic stereo acoustic echo cancellation structure. This structure is particularly important for multiple participant VoIP conferences since it assists the near-end user in distinguishing between participants in the conference. N far-end talkers are positioned in M static spatial regions as described in section 3.3. In section 3.3 the far-end talker's speech propagated from the near-end user's loudspeakers to a single microphone placed in front of the near-end user. In this chapter, however, the far-end talker's speech propagates from the near-end user's loudspeakers to an array of microphones placed above the near-end user's video display terminal. Fig. 30

illustrates far-end talker spatialization and the near-end user's microphone array for six participants ($N = 6$) in that are assigned one of four spatial regions ($M = 4$). Fig. 30 is similar to Fig. 12 except for replacement of the single near-end user's microphone by the microphone array beamformer.

Fig. 30. Near-end talker spatialization with beamforming

As previously stated in section 4.1.2, the synthetic stereo assignment scheme is scalable since there can be a large number of participants, N , that are assigned into

NOTE TO USERS

Page(s) not included in the original manuscript and are unavailable from the author or university. The manuscript was scanned as received.

This reproduction is the best copy available.

UMI[®]

Fig. 31. Synthetic stereo acoustic echo cancellation structure with beamformer

The single canceler per-spatial region structure with microphone array beamformer does not suffer from the fundamental problem of cross-correlation of the reference signals as described in [46], for stereophonic acoustic echo cancellation. The problem of cross-correlation of the reference signals is explained in more detail in section 3.2 and reasons why the per-spatial region structure does not suffer from this problem are detailed in section 4.1.

The single canceler per-spatial region structure with microphone array beamformer can be used in two different modes, similar to the structure introduced in

section 4.1.3 without the beamformer. The structure can allow the echo cancelers for all spatial regions to be active all of the time or only the echo canceler for one spatial region can be allowed to be active at a time.

The equations introduced in the remainder of this section are similar to those for the per-spatial region acoustic echo cancellation structure without beamformer introduced in section 4.1, except for the inclusion of the beamformer. The equations refer to the structure shown in Fig. 31.

The far-end participant signals in the single canceler per-spatial region structure with beamforming are identified as, $s_j(n)$, where $j = 1, 2, \dots, N$.

The N far-end participant signals that are mapped into one of the M near-end spatial regions, to form the far-end spatial signals, $v_i(n)$, where $i = 1, 2, \dots, M$.

The adaptive echo path model for a single spatial region in the single canceler per spatial region structure is denoted by, $\hat{h}_i(n)$, where $i = 1, 2, \dots, M$.

A spatial region i is defined by its left and right channel spatialization functions, g_{Li} and g_{Ri} , where $i = 1, 2, \dots, M$. The system designer must determine the appropriate spatial mapping but it is recommended that it correspond to the video image.

The left loudspeaker output, $x_L(n)$, is formed by convolving the far-end spatial signals by the appropriate left-channel spatialization function. (right channel not shown)

$$\mathbf{x}_L(n) = \sum_{i=1}^M \nu_i(n) * \mathbf{g}_{Li} \quad (46)$$

The left and right channel reception room impulse responses to each of the K microphone transducers are denoted by $\mathbf{h}_{Lk}(n)$ and $\mathbf{h}_{Rk}(n)$, where $k = 1, 2, \dots, K$.

The beamformer weight functions are defined by the FIR filters $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_K$, where $k = 1, 2, \dots, K$.

The primary signal, $d(n)$, received by the microphone is, therefore,

$$\begin{aligned} d(n) = & \mathbf{x}_L(n) * [\mathbf{h}_{L1}(n) * \mathbf{b}_1 + \mathbf{h}_{L2}(n) * \mathbf{b}_2 + \dots + \mathbf{h}_{LK}(n) * \mathbf{b}_K] \\ & + \mathbf{x}_R(n) * [\mathbf{h}_{R1}(n) * \mathbf{b}_1 + \mathbf{h}_{R2}(n) * \mathbf{b}_2 + \dots + \mathbf{h}_{RK}(n) * \mathbf{b}_K] \end{aligned} \quad (47)$$

If only one spatial region echo canceler is allowed to be active at a time then the echo canceler converges to the convolution of the spatialization functions, reception room transfer functions and beamformer weight functions. (48) shows the adaptive echo path model for spatial region 1.

$$\begin{aligned} \hat{\mathbf{h}}_1(n) = & \mathbf{g}_{L1} * [\mathbf{h}_{L1}(n) * \mathbf{b}_1 + \mathbf{h}_{L2}(n) * \mathbf{b}_2 + \dots + \mathbf{h}_{LK}(n) * \mathbf{b}_K] \\ & + \mathbf{g}_{R1} * [\mathbf{h}_{R1}(n) * \mathbf{b}_1 + \mathbf{h}_{R2}(n) * \mathbf{b}_2 + \dots + \mathbf{h}_{RK}(n) * \mathbf{b}_K] \end{aligned} \quad (48)$$

5.7 Simulation

A simulation for one active spatial region shows that when the microphone array beamformer is combined with the single canceler per spatial region echo cancellation structure a performance increase is realized. The simulation, Fig. 32, shows that an

NOTE TO USERS

Page(s) not included in the original manuscript and are unavailable from the author or university. The manuscript was scanned as received.

This reproduction is the best copy available.

Fig. 32. Acoustic echo canceler performance with beamformer

NOTE TO USERS

Page(s) not included in the original manuscript and are unavailable from the author or university. The manuscript was scanned as received.

This reproduction is the best copy available.

UMI[®]

available. This chapter proposes a method where several network delay models are trained before initiation of the voice packet conversation and the algorithm adapts by intelligently switching between the pre-trained models as required [64]. The major difference is that other existing algorithms propose methods by which the network delay model is continuously adapted in response to the changing network. This chapter proposes a method that does not change the models. Adaptation is comprised of selecting the appropriate model in response to changing network conditions.

The proposed method preserves network delay models for all possible patterns. Network delay observations show that the network delay patterns change slowly during the course of the conference, but short-term phenomena occur, such as spikes [55], followed by a return to a pattern similar to the preceding one. It is desirable for the modeling technique to be able to switch fairly quickly between different models without losing the old model, since short-term phenomena usually die out quickly and it is beneficial to have the prior network delay pattern available to revert back to. The previously reviewed predictive algorithms for determining optimal playout buffer time suffer from their inability to react appropriately to changing network conditions due to their use of limited data and restrictive data aging techniques.

Fig. 33 shows two possible models for network delay. This first model represents the network delay during normal network operation. The second model represents the network delay during a network period where the mean delay is greatly increased and the packet inter-arrival jitter, shown as variation, is much larger. If the network

NOTE TO USERS

Page(s) not included in the original manuscript and are unavailable from the author or university. The manuscript was scanned as received.

This reproduction is the best copy available.

UMI[®]

6.3 Generic Playout Delay Model

This thesis proposes a model that breaks playout delay computation up into two discrete components to make the model generic enough for a wide range of network conditions. The first component of the proposed technique is the mean network delay estimate for the last packet of talkspurt k , $\hat{d}_k^{n_k}$. This component handles network propagation delay, overall long-term mean routing delay and compensates for clock skew between sender and receiver. The mean network delay estimate for packet i of talkspurt k is computed as in (49)(50)

$$\hat{d}_k^i = \alpha \hat{d}_k^{i-1} + (1 - \alpha) d_k^i, \quad 1 \leq i \leq n_k, 1 \leq k \leq M \quad (49)$$

$$\hat{d}_k^1 = \alpha \hat{d}_{k-1}^{n_{k-1}} + (1 - \alpha) d_k^1, \quad 1 \leq k \leq M \quad (50)$$

where d_k^i is the normalized network delay experienced by packet i in the k -th talkspurt.

The second component of the proposed technique is the required delay in addition to the mean network delay estimate to achieve a desired average lost packet rate, α_{lp} . This component handles the variable portion of the router queuing delays. The amount of delay in addition to the mean network delay required to achieve the desired α_{lp} is called the most probable quantized excess delay for talkspurt k , $Q[\hat{e}_k]$, and is computed by the hidden Markov model.

The appropriate playout delay for talkspurt k is predicted during the silence period after talkspurt $k-1$ using the mean delay estimate from the last packet of talkspurt $k-1$ and the most probable quantized excess delay from talkspurt $k-1$.

$$\hat{p}_k = \hat{d}_{k-1}^{n_{k-1}} + Q[\hat{e}_{k-1}] \quad (51)$$

Fig. 34 shows the two components of the playout delay for talkspurt k, $\hat{p}_k$, in a network delay trace. The playout delay is predicted based on the mean delay estimate, $\hat{d}_{k-1}^{n_{k-1}}$, and most likely quantized excess delay for talkspurt k-1, $Q[\hat{e}_{k-1}]$.

Fig. 34. Playout delay components

A brief example explaining why the proposed model is considered generic enough for a wide range of network conditions follows. Two different voice packet conferences may be proceeding at the same time. There is a substantial difference in the propagation delay for the two conferences, this fact will be handled by the

mean network delay estimation component. However, the routers exhibit similar variability in the queuing delays, which means both conferences require the same amount of quantized excess delay in addition to the mean network delay estimate for the appropriate playout delay.

Ideally the mean delay estimate, $\hat{d}_{k-1}^{n_{k-1}}$, will accurately represent the current long-term delay average for the conference. If the mean delay estimate is not accurate then the quantized excess delay variable will automatically provide compensation. If the mean delay estimate is too high then a small quantized excess delay will be predicted. If the mean delay estimate is too low then a high quantized excess delay will be predicted.

This model is extensible to multiple participants since each receiver will keep track of their own mean delay estimate, $\hat{d}_{k-1}^{n_{k-1}}$, and most likely quantized excess delay for talkspurt $k-1$, $Q[\hat{e}_{k-1}]$. These two quantities are not necessarily the same at each receiver, and may be quite different depending on the various network topologies involved.

6.4 Hidden Markov Model Notation

The following notation will be used throughout the remainder of this chapter to describe the concepts specific to quantized excess delay computation using an HMM. This notation is in addition to the notation presented in section 3.4.1. The HMM notation and theory is adapted from [65][66].

NOTE TO USERS

Page(s) not included in the original manuscript and are unavailable from the author or university. The manuscript was scanned as received.

100

This reproduction is the best copy available.

UMI[®]

Each state in the HMM corresponds to a particular quantized excess delay value. Excess delay was previously defined as the delay required in addition to the mean network delay estimate to achieve a desired alp. The excess delay values are quantized into N levels since it is difficult to work with an HMM with a continuous state space. The quantized excess delay values are chosen in a manner, which ensures that the range of values is applicable to a wide range of network conditions with highly variable network topologies. The quantized excess delay values can be determined by analyzing a large number of Internet delay traces that are representative of the range of network conditions off-line using the algorithm from section 3.4.3.1.1 - Optimal Algorithm for an Isolated Talkspurt. If the amount of required excess delay in the traces ranges from 5ms and 65ms after removing the mean network delay, then a reasonable set of quantized excess delay values would be {5ms, 10ms, 15ms, ..., 60ms, 65ms} for a quantization step-size of 5ms. This HMM would require $N = 13$ states to implement with one state for each quantized excess delay value. The quantization step-size can be determined by repeatedly training and testing the model until the desired accuracy is achieved.

The HMM predicts the most likely quantized excess delay value component, $Q[\hat{e}_{k-1}]$, of the playout delay by combining information from the initial state probability distribution, π , the excess delay state transition probability matrix, A , and the continuous observation density excess delay probability distribution, B .

The initial state probability for quantized excess delay state i , π_i , denotes the probability of the first quantized excess delay value predicted by the HMM.

The excess delay state transition probability, a_{ij} , denotes the probability of a transition from the most likely quantized excess delay for talkspurt k-2, $S_{k-2} = i$, to a quantized excess delay for talkspurt k-1, $S_{k-1} = j$.

The continuous observation density excess delay probability matrix, $b_j(\mathbf{O}_{k-1})$, denotes the probability that the observed excess network delays in talkspurt k-1, $\mathbf{O}_{k-1}$, match the pre-trained network excess delay mean and variance statistics for a particular quantized excess delay state, $S_{k-1} = j$.

The manner in which the HMM playout delay prediction technique responds to short-lived network delay spikes, a change in the packet routing or another short-lived network phenomena is important. The technique proposed by this thesis not only considers the probability of the observed network delays, but it also considers the likelihood of a transition from one quantized excess delay model to the another. For example, for a transition from normal network operation to a different routing with increased delay and jitter, the HMM technique would not only compute the probability of seeing these network delay values but also the probability of the transition to a new state in the HMM.

The HMM technique, therefore, combines desirable properties from the playout delay prediction techniques reviewed in section 3.4.3.2. The HMM technique collects network delay mean and variance data, through symbol probability, similar to the techniques described in 3.4.3.2.2, the collect delay statistics algorithm, and 3.4.3.2.3, the packet delay distribution algorithm. The HMM technique also provides filtering of

NOTE TO USERS

Page(s) not included in the original manuscript and are unavailable from the author or university. The manuscript was scanned as received.

103

This reproduction is the best copy available.

UMI[®]

6.6 Training vs. Playout Delay Prediction

Training the HMM requires computing the optimal quantized excess delay state sequence, $S_1 S_2 \dots S_M$, for a given set of observed excess delays, $O_1 O_2 \dots O_M$. The optimal quantized excess delay value for talkspurt k , $Q[\hat{e}_k]$, can be solved for by computing the appropriate playout delay for talkspurt k , $\hat{p}_k$, and the mean network delay estimate for the last packet of talkspurt k , $\hat{d}_k^{n_k}$, off-line. The hidden Markov model parameters, $\lambda = (A, B, \pi)$, can then be solved for from the optimal quantized excess delay state sequence.

Playout delay prediction using the HMM requires predicting the most likely playout delay for talkspurt k , $\hat{p}_k$, from the partial excess delay observation sequence from talkspurt 1 to $k-1$, $O_1 O_2 \dots O_{k-1}$. The mean network delay estimate for talkspurt $k-1$, $\hat{d}_{k-1}^{n_{k-1}}$, and the most likely quantized excess delay for talkspurt $k-1$, $Q[\hat{e}_{k-1}]$, are computed in real-time to form the playout delay estimate for talkspurt k , $\hat{p}_k$. The most likely quantized excess delay for talkspurt $k-1$, $Q[\hat{e}_{k-1}]$, is predicted using the hidden Markov model parameters, $\lambda = (A, B, \pi)$, computed during the training stage.

6.7 Training the HMM

The model is trained by submitting enough Internet trace examples to tune the parameters of the system to predict playout buffer delays for a wide range of network delay conditions. If the model is not trained with enough data for a given number of

NOTE TO USERS

Page(s) not included in the original manuscript and are unavailable from the author or university. The manuscript was scanned as received.

105

This reproduction is the best copy available.

UMI[®]

6.7.1 Compute Playout Delay Off-Line

The model is trained by providing network delay trace examples that are marked during the training phase with the optimal playout delay computed off-line. The optimal playout delay can be computed for the current talkspurt k based on the desired loss rate, see section 3.4.3.1.1 - Optimal Algorithm for an Isolated Talkspurt. Large variations in the optimal playout delay can occur using this method largely due to short talkspurts that do not fully capture the signal statistics accurately. In this chapter, we chose to smooth the optimal playout delay estimate over several talkspurts by applying the minimum average playout delay technique, as shown by equation (10). For example, the simulations in section 6.9 where smoothed over three talkspurts, $M - k = 3$.

The mean network delay, $\hat{d}_k^i$, was also computed for each talkspurt off-line by using (14)(15). During a silence period the mean network delay estimate is frozen.

The quantized excess delay, $\mathcal{Q}[\hat{e}_k]$, for the k -th talkspurt is then computed by subtracting the mean delay estimate for the last packet of talkspurt k from the desired playout delay and quantizing the results:

$$\mathcal{Q}[\hat{e}_k] = \mathcal{Q}[\hat{p}_k - \hat{d}_k^{n_k}] \quad (56)$$

The quantized excess delay is associated with a particular state in the HMM. The excess delay is quantized, as previously discussed, to limit the number of HMM states to a finite value.

6.7.2 Initial State and State Transition Probabilities

The initial state probability matrix, π , is computed by counting the number of occurrences of one quantized excess delay, or state i , with respect to the occurrences of all other or states.

$$\pi_i = \frac{\text{number of times in state } i}{\sum_{j=1}^N \text{number of times in state } j} \quad (57)$$

The initial state probability matrix for all the quantized excess delays in the range of 5ms to 65ms (states 1 to 13) are shown in Fig. 37.

Fig. 37. Initial state probability

The state transition probability matrix, A , is computed by counting the transitions from one quantized target excess delay value, or state i , to the next quantized target excess delay value, or state j .

$$a_{ij} = \frac{\text{number of transitions from state } i \text{ to } j}{\text{number of transitions from state } i} \quad (58)$$

The state transition probability matrix for all the quantized excess delays in the range of 5ms to 65ms (states 1 to 13), for the model shown in Fig. 35, are shown in Fig. 38. State transitions along the diagonal are most probable since the minimum average playout delay technique does not, typically, change optimal delay states in a rapid fashion. The diagonal shape of the state transition matrix is important in reducing end-to-end delay variation.

Fig. 38. State transition probabilities

6.7.3 Excess Delay Probabilities

The continuous observation density excess delay probability distribution matrix uses a single Gaussian probability distribution to model the probability of seeing the observation sequence [65][66]. The excess delay probability matrix is composed of the average excess delay mean parameter, μ_j , and the average excess delay variance parameter, U_j for each state j . The average excess delay mean and variance parameters are computed by first compiling excess delay statistics for all the

talkspurts in the entire network delay trace set, used for training the HMM, and then grouping all the statistics for a given state together and taking the average.

The excess delay, e_k^i , statistics for a talkspurt are determined by computing the mean and variance for delays in excess of the mean delay estimate for that talkspurt.

The excess delay mean for talkspurt k, $\mu(e_k)$, is computed by

$$\mu(e_k) = \sum_{i=1}^{n_k} e_k^i / \sum_{i=1}^{n_k} I(e_k^i) \quad (59)$$

The excess delay variance for talkspurt k, $\sigma^2(e_k)$, is computed by

$$\sigma^2(e_k) = \sum_{i=1}^{n_k} (e_k^i - \mu(e_k))^2 / \sum_{i=1}^{n_k} I(e_k^i) - 1 \quad (60)$$

The indicator function for the excess delay, $I(e_k^i)$, is used because delays below the mean delay estimate have a negative excess delay value and should not be factored into the excess delay mean and variance calculation. The indicator function is defined as

$$I(e_k^i) = \begin{cases} 1, & \text{if } e_k^i \geq 0 \\ 0, & \text{if } e_k^i < 0 \end{cases} \quad (61)$$

If there are no delays in excess of the mean delay estimate for talkspurt k, $\sum_{i=1}^{n_k} I(e_k^i) = 0$, then the excess delay mean and variance observation data should be discarded.

All the excess delay mean and excess delay variance values are then sorted according to the quantized excess delay for the talkspurt. Let L_j be the number of occurrences of a quantized excess delay state j in the training network delay data.

The average excess delay mean for quantized excess delay state j , μ_j , is computed from all L_j excess delay mean values by

$$\mu_j = \frac{1}{L_j} \sum_{k=1}^{L_j} \mu(e_k) \quad (62)$$

The average excess delay variance, U_j , for state j is computed from all L_j excess delay variance values by

$$U_j = \frac{1}{L_j} \sum_{k=1}^{L_j} \sigma^2(e_k) \quad (63)$$

The average excess delay mean and variance for all the quantized excess delays in the range of 5ms to 65ms (states 1 to 13) are shown in Fig. 39.

Fig. 39. Excess delay mean and variance

6.8 Playout Delay Prediction Using the HMM

The HMM technique predicts playout delay values in real-time by analyzing network delays during talkspurts and calculating the probability of the observed network delays matching a state sequence. The playout delay is only adjusted during the silence periods between talkspurts. The HMM technique assumes that enough network delay examples were submitted during the training procedure to sufficiently train the HMM for a wide range of network conditions. The current real-time network delay data is assumed to be typical of the examples submitted during

the training procedure and that the HMM parameters can accurately predict playout delays for this data.

The HMM playout delay prediction procedure for talkspurt k , illustrated in Fig. 40, is to compute the playout delay as the sum of two components (51). The first component is the mean network delay estimate for the last packet of talkspurt $k-1$. The second component is the most likely quantized excess delay for talkspurt $k-1$. This procedure is repeated during each silence period throughout the entire voice conference.

Fig. 40. Playout delay prediction procedure

6.8.1 Compute the Mean Delay Estimate

The mean network delay, $\hat{d}_{k-1}^i$, must be computed in real-time for all network delays inside talkspurt k-1 by using (14)(15). The mean network delay estimate for the last packet of the k-1-th talkspurt, $\hat{d}_{k-1}^{n_{k-1}}$, is used to predict the playout delay for the k-th talkspurt and for computing excess delays.

6.8.2 Viterbi Algorithm

The Viterbi algorithm, an important technique in HMM theory, is used to predict the most likely quantized excess delay state sequence for the observed network delays. The Viterbi algorithm computes the most likely quantized excess delay state using the initial excess delay state probability matrix, π , the excess delay state transition matrix, A , and the continuous observation density excess delay probability distribution, B . The Viterbi best score variable, $\delta_k(j)$, is defined as the best score along a single quantized excess delay state path from talkspurt 1 until talkspurt k ending in state j , $S_1 S_2 \dots S_{k-1}$, $S_k = j$, which accounts for the observation sequence $\mathbf{O}_1 \mathbf{O}_2 \dots \mathbf{O}_k$, given the model λ [66].

$$\delta_k(j) = P(S_1 S_2 \dots S_{k-1}, S_k = j, \mathbf{O}_1 \mathbf{O}_2 \dots \mathbf{O}_k | \lambda) \quad (64)$$

The Viterbi best score variable, $\delta_{k-1}(j)$, is used to compute the most likely quantized excess delay for talkspurt k-1, $\mathcal{Q}[\hat{e}_{k-1}]$. The best score variable, $\delta_{k-1}(j)$, is computed recursively [65][66] by the following algorithm.

For the first talkspurt, $k = 1$, the algorithm is initialized as follows

$$\delta_1(j) = \pi_j b_j(\mathbf{O}_1), \quad 1 \leq j \leq N \quad (65)$$

The algorithm proceeds recursively for talkspurts $2 \leq k \leq M$ as follows

$$\delta_k(j) = \max_{1 \leq i \leq N} (\delta_{k-1}(i) a_{ij}) b_j(\mathbf{O}_k), \quad 1 \leq j \leq N \quad (66)$$

The most likely quantized excess delay for talkspurt k-1, $Q[\hat{e}_{k-1}]$, is computed by determining the maximum $\delta_{k-1}(j)$ over all j and multiplying by the quantization increment (e.g. 5ms).

$$Q[\hat{e}_{k-1}] = \left(\arg \max_j \delta_k(j) \right) \times 5ms, \quad 1 \leq k \leq M \quad (67)$$

The playout delay estimate for talkspurt k is then computed by (previously defined)

$$\hat{p}_k = \hat{d}_{k-1}^{n_{k-1}} + Q[\hat{e}_{k-1}] \quad (51)$$

The Viterbi algorithm can, alternately, be implemented in the logarithmic domain [65][66]. Implementation in the logarithmic domain eliminates the risk of underflows and converts multiplication operations into additions, which reduces computational complexity.

6.8.3 Compute the Probability of the Observation Sequence

The probability of the observed network delays for talkspurt k-1 in quantized excess delay state j , $b_j(\mathbf{O}_{k-1})$, is required in the computation of the Viterbi best score variable (65)(66). The probability of the observed network delays is computed by first forming the observation vector for talkspurt k-1, $\mathbf{O}_{k-1}$, and then determining the

probability of this observation vector for each quantized excess delay state j , $b_j(\mathbf{O}_{k-1})$.

The observation vector for talkspurt $k-1$, $\mathbf{O}_{k-1}$, is formed from the sequence of excess delays for talkspurt, $k-1$, which are computed from (53) using the mean network delay for the last packet of talkspurt $k-1$, $\hat{d}_{k-1}^{n_{k-1}}$.

$$\mathbf{O}_{k-1} = [e_{k-1}^1 \quad e_{k-1}^2 \quad \cdots \quad e_{k-1}^{n_{k-1}}] \quad (68)$$

The probability of an excess delay value for the i -th packet of the k -th talkspurt, e_k^i , occurring in excess delay state j is computed using the Gaussian pdf equation.

$$f_e(e_k^i)_j = \begin{cases} \frac{1}{\sqrt{2\pi\mathbf{U}_j}} e^{-\frac{(e_k^i - \mu_j)^2}{2\mathbf{U}_j}}, & \text{if } e_k^i \text{ is defined} \\ 1, & \text{if } e_k^i \text{ is undefined} \end{cases}, \quad 1 \leq j \leq N \quad (69)$$

where μ_j is the average excess delay mean and $\mathbf{U}_j$ is the average excess delay variance which were computed during the training procedure.

The probability of the excess delay observation vector for talkspurt $k-1$, $\mathbf{O}_{k-1}$, occurring in quantized excess delay state j , $b_j(\mathbf{O}_{k-1})$, is then computed by

$$b_j(\mathbf{O}_{k-1}) = f_e(e_{k-1}^1)_j \times f_e(e_{k-1}^2)_j \times \cdots \times f_e(e_{k-1}^{n_{k-1}})_j \quad (70)$$

6.9 Simulation

6.9.1 Network Delay Traces

The widely accepted method of simulating playout delay prediction algorithms is to collect empirical Internet delay trace data from actual voice packet applications run over a wide range of network conditions, also called trace-driven simulation, [53][55][58]. Twelve of the Internet delay data traces, used in this chapter, were obtained from Sreenan et al. [58] and forty one delay traces were obtained from Dr. Henning Schulzrinne [55].

6.9.2 Generating Talkspurt Information

Artificial talkspurt information was generated for the Internet delay trace data since the Internet delay traces did not contain talkspurt information. The Internet delay traces were generated by a continuous 64 Kb/s audio stream that produced a packet every 20 ms even during silence periods [58]. Talkspurt information was artificially generated by using a simple on-off technique described by Brady [67][68]. The average talkspurt length was set to 1.5 s, $(1/\beta) = 1.5s$, and the average silence period was set to 2.25 s, $(1/\alpha) = 2.25s$, based on values from Chan and Leung [69]. The model used to artificially generate the talkspurt information is shown in Fig. 41. In Fig. 41 T denotes a talkspurt state and S denotes a silence state.

Fig. 41. Artificial talkspurt generation

6.9.3 Simulated Delay Prediction Techniques

This chapter compares several techniques for playout delay prediction against the HMM technique proposed by this thesis. The playout delay prediction performance is evaluated by analyzing the average total end-to-end delay, average packet loss rate and packet burst loss length numbers for each of the simulated techniques. It is desirable for the playout delay prediction technique to produce a good compromise between low end-to-end delay, end-to-end delay standard deviation, low average packet loss rate and low average packet burst loss length. Packet burst loss length statistics are collected since this data could be used in selecting a packet forward error correction (FEC) technique. The mean burst loss length is particularly important since the chosen FEC technique should be chosen for its ability to correct for the mean packet burst loss length [54].

The first algorithm simulated, referred to as OPT in this thesis, is the offline playout delay prediction algorithm (8)(9), see 3.4.3.1.1 - Optimal Algorithm for an Isolated Talkspurt. This algorithm computes the best playout delay on a per talkspurt basis, which is not optimal for the entire conference due to its high end-to-end delay

standard deviation. The technique, however, can be used as a benchmark measurement since it will inherently offer the best end-to-end delay for a given packet loss rate. This algorithm is non-causal and cannot be implemented in real-time voice packet applications.

The first algorithm can be made causal by delaying usage of the delay information from the $k-1$ -th talkspurt to the k -th talkspurt (12), see 3.4.3.2.1 - Causal Version of Optimal Algorithm. This algorithm is referred to as DOPT.

The classic delay statistics algorithm (14)(15)(16)(17), see 3.4.3.2.2 – Delay Statistics Algorithm, is called STAT in this thesis. Network delay mean and standard deviation information are collected and the playout delay is estimated with $\beta = 4$.

The final algorithm simulated is the HMM technique introduced by this thesis, referred to as HMM. The results of training the algorithm based on the technique described in sections 6.7 and 6.8 are shown in Fig. 37, Fig. 38, and Fig. 39. The overall complexity of this technique is greater than the other simulated algorithms. However, the test mode complexity is comparable to other algorithms proposed in recent literature [58].

6.9.4 Testing Results

The HMM, OPT and DOPT playout delay prediction algorithms were compared by targeting a specific average packet loss rate of 5.0% for the Sreenan data. The STAT algorithm was tested with parameter β set to 4, its commonly used value [55], which causes the STAT algorithm to generate very few lost packets. None of the

candidate algorithms achieve the 5.0% alp target exactly even though they are configured to do so. Many of the talkspurts contain very few packets, which causes widely varying instantaneous packet loss rate values resulting in the alp discrepancy. Fig. 37, Fig. 38, and Fig. 39 represent the resulting parameters from training the HMM model for the target alp.

The results from the four algorithms for the Sreenan data are shown in Table I. The OPT algorithm shows that if the optimum delay were computed off-line on a per-talkspurt basis due to rounding the best-case alp would be 3.0% even though it is supposed to drop on average 5.0% of the packets. The mean end-to-end delay is 829.8 ms for this benchmark algorithm. The end-to-end delay standard deviation is fairly high for this algorithm.

Table I. HMM delay prediction results for Sreenan data

Method	ted (ms)				alp (%)	Burst Loss Lengths		
	Min	Mean	Max	Std		Min	Mean	Max
OPT	94	829.8	2656	10.4	3.0	1	1.2	8
DOPT	94	829.8	2656	10.3	11.2	1	2.34	72
STAT	100	861.6	2685	10.7	0.5	1	1.58	17
HMM	101	834.5	2659	7.4	6.2	1	1.92	72

The DOPT algorithm does not perform well for predicting playout delay since it bases its playout delay prediction solely on the optimal playout delay for the last talkspurt. This prediction technique results in a high alp, has the highest mean packet burst loss length and has a high end-to-end delay standard deviation. The mean end-to-end delay is identical to OPT as would be expected.

The STAT algorithm uses delay mean and standard deviation estimates to predict the appropriate playout delay. The STAT algorithm obtains the lowest alp since it predicts the playout delay with a very conservative variation estimate, $\beta = 4$ standard deviations from the mean. This algorithm has the highest mean end-to-end delay since it sacrifices end-to-end delay in order to obtain a low alp. A low alp by itself, however, is not a good measure of the playout delay algorithm performance since minimizing the delay is equally as important. The end-to-end delay standard deviation is the highest for the STAT algorithm, which indicates that the playout delay fluctuates greatly between talkspurts. The large end-to-end delay standard deviation results in too much playout delay variation for good speech quality.

The HMM algorithm produces a good compromise between the mean end-to-end delay, the end-to-end delay standard deviation, alp and mean burst loss length. The HMM technique has the lowest end-to-end delay standard deviation, beating the STAT algorithm by over 30%, and has a mean end-to-end delay similar to the benchmark OPT algorithm. The 6.2% packet loss rate is considered acceptable for good quality speech using today's sophisticated coding techniques. The alp rate with the HMM technique is very close to the desired 5.0% alp and the mean packet burst loss length could easily be compensated for by a FEC technique [54].

The maximum packet burst loss length shows that even the best playout delay prediction techniques will sometimes rely heavily on error concealment techniques when the network undergoes a rapid change in its congestion status. The low mean

packet burst loss length statistics, however, show that the majority of lost packets could easily be handled by a FEC scheme.

A simulation was also run for several Schulzrinne Internet data traces. The results from these simulations are shown in Table II. These results were obtained for HMM parameters trained to produce an average packet loss rate of 4.7%. The Schulzrinne results also show that the HMM algorithm compromises well between mean delay end-to-end delay, end-to-end delay standard deviation and the average packet loss rate. Overall the HMM algorithm provides superior performance in our simulations.

Table II. HMM delay prediction results for Schulzrinne data

Method	ted (ms)				alp (%)	Burst Loss Lengths		
	Min	Mean	Max	Std		Min	Mean	Max
OPT	0	73.54	7686	53.6	3.83	1	1.23	18
DOPT	0	73.54	7686	53.6	10.9	1	2.03	198
STAT	1	90.51	4471	51	2.5	1	1.89	198
HMM	6	78.94	1129	38.9	4.95	1	2.06	216

7 Acoustic Round Trip Delay and Echo Suppression Target Determination

VoIP systems must provide better hands-free, full-duplex conferencing support for users, to compete with PSTN service. Speakerphones are in wide use by business customers for conference calling and by home users for the convenience offered from having their hands free to perform activities other than holding the handset.

Full-duplex capabilities are essential since it has been determined that normal human conversation is easily disrupted by half-duplex systems. A pure half-duplex system only allows one user to talk at a time by switching between the active talkers in the conference. The delay in half-duplex switching can cause a talker's speech to be cut-off and create switching noise. Full-duplex systems, in contrast, allow all users to talk simultaneously. Half-duplex systems can also be adversely affected by environmental noise causing the system to switch permanently to the loudest end.

This chapter proposes a metric called the acoustic round trip delay (ARTD), which can be used to measure the end-to-end delay as perceived by a conference participant, for hands-free VoIP conferences. This metric can, in turn, be used in combination with a subjective echo tolerance curve to predict the appropriate amount of echo suppression required by the conference.

7.1 Echo Tolerance Curves

Subjective tests of the psychological impact of echo show that the level of speech quality impairment due to echo is related to the power of the received echo and the

echo path delay. Results of subjective tests for a user's tolerance towards echo for a PSTN have been studied extensively and are outlined in [2][70]. These studies have generated talker echo loudness rating (TELR) curves, which are typically used by network planners to determine echo suppression targets for network echo cancellers. A loudness rating is defined as the weighted electro-acoustic loss between interfaces in the telephone network [70]. The TELR curves were designed primarily for use in analog PSTN networks for dealing with network echo caused by the 2-4 wire hybrid [70]. In hands-free digital VoIP networks, the TELR curves can be applied to acoustic echo to determine the required amount of echo suppression to ensure that acoustic echo is tolerable to the user [71].

TELR curves are not the best metric for hands-free VoIP, but they are the most suitable metric available to date. TELR curves do not include factors that are present in hands-free scenarios such as room reverberation and background noise. A preferable solution would be to measure TELR curves for hands-free scenarios in various rooms with various types of background noise. For example, if a room had a large amount of background noise then the listener may not require as much echo suppression since the acoustic echo would be dominated by background noise. This section will proceed with presentation of the acoustic round trip delay and use the TELR metric since it is the best metric available at present.

7.1.1 Asymmetric Internet Transmit/Receive Paths

The TELR curves found in [2][70] are based on the mean one-way propagation delay. Mean one-way propagation delay, for TELR purposes, is defined as half the sum of the total transmission time in both transmission directions [70]. This is a

convenient measure in PSTN networks since the transmit and receive path propagation delays are symmetric. In the Internet, however, transmit and receive path delays are not symmetric, since voice packets transmitted over the Internet from the source to the destination may follow a different path than the path taken for packets transmitted from the destination back to the source [72]. This phenomenon occurs because a different sequence of routers may be used for the forward and reverse paths. A round-trip time (RTT) measurement, as is commonly performed by the real-time transport protocol/RTP control protocol (RTP/RTCP) [35], measures the performance of two distinct paths in a combined metric. In addition, if the source-to-destination and destination-to-source paths are symmetric, their one-way propagation delays are not necessarily equivalent since the packets may incur asymmetric direction-dependent queuing delays at the various routers along the two paths. A quality-of-service provisioning technique, such as the differential services architecture, commonly called diffserv [73], can complicate matters further by providing an additional source of asymmetric delays if one path is provisioned differently from the other path.

7.1.2 TELR Curves for VoIP

For VoIP conferences, the horizontal axis for the TELR curves is converted into ARTD which is defined as the time taken for the near-end user's speech (Participant A in Fig. 4) to echo at the far-end (Participant B) in a full-duplex hands-free system and return to the near-end user's ears [1]. The ARTD metric includes the asymmetric transmit and received paths of the Internet for ease of computation. The resultant

TEL_R vs ARTD curves, shown in Fig. 42, are approximated by the following equation [70]

$$TEL_R = A_0 - 6e^{-0.075T^2} + 40 \log \frac{1 + T/20}{1 + T/300} \quad (71)$$

where $A_0 = 14\text{dB}$ for 1% objection or 7dB for 10% objection and T is the ARTD in ms.

Fig. 42. Talker echo loudness rating tolerance curves

The 1% objection curve is considered the acceptable case for network planning [70]. The 10% objection curve is only used in exceptional circumstances. The 1% objection curve will be used for discussion purposes in this thesis.

The factors that contribute to TELR are send loudness rating (SLR), receive loudness rating (RLR) and echo loudness loss (EL) [70]. EL is interpreted as the echo loudness loss of the electric echo path and SLR and RLR are interpreted as the losses in the interfaces to the electric path. Equation (72) is intended for use in PSTN telephony with handsets and network echo cancellers.

$$TEL R_{NETWORK} = EL + SLR + RLR \quad (72)$$

In PSTN networks SLR is defined as the loudness rating between the speaking subscriber's mouth and an electric interface in the network and RLR is defined as the loudness rating between an electric interface in the network at the listening subscriber's ear [70]. In hands-free VoIP telephony SLR and RLR are solely used to characterize the microphone and loudspeaker loss levels, which are the electric interfaces to the VoIP system, see Fig. 4.

The echo loudness loss, for hands-free VoIP systems, is a combination of echo suppression provided by the acoustic echo canceller, nonlinear processor signal path attenuation and the transmission and reception room signal attenuation, as shown in Fig. 4. In a purely Internet hosted VoIP conference there are no network echo cancellers required since network echo is created by 2-4 wire hybrids which are not present in this case.

The total amount of TELR for a VoIP conference is, therefore, a combination of the acoustic echo canceller echo suppression, EL_{AEC} , attenuation and centre-clipping controlled by the NLP, EL_{NLP} , loudspeaker attenuation, SLR , microphone attenuation, RLR , signal loss in the reception room between the loudspeaker and microphone, EL_{ROOM} and the two components of the signal loss in the transmission room, EL_S and EL_R .

$$TEL_{R_{VoIP}} = EL_{AEC} + EL_{NLP} + EL_{ROOM} + EL_S + EL_R + \sum SLR + \sum RLR \quad (73)$$

where $\sum SLR = SLR_A + SLR_B$, $\sum RLR = RLR_A + RLR_B$, EL_{AEC} applies to the currently active acoustic echo canceller and EL_{NLP} applies to signal attenuation provided by the currently active NLP.

Equation (2) shows that the amount of echo suppression being currently provided by the AEC can be measured, EL_{AEC} . The far-end loudspeaker attenuation, microphone attenuation and reception room attenuation, $SLR_B + RLR_B + EL_{ROOM}$ for Participant B with Participant A speaking, can be computed by a standard signal processing technique such as a cross-correlation. The near-end loudspeaker attenuation, microphone attenuation and transmission room attenuations, $SLR_A + RLR_A + EL_S + EL_R$ for Participant A speaking, can also be computed by standard signal processing techniques. The TELR, $TEL_{R_{VoIP}}$, can be determined by accurately computing the ARTD. The signal path attenuation controlled by the NLP, therefore, can be accurately determined by the VoIP system. Two methods for

measuring the ARTD of a VoIP conference are proposed in this thesis, the round-trip time method and the playout delay prediction method. The round-trip time method does not require clock synchronization, whilst the playout delay prediction technique does.

7.2 Time Synchronization

Two VoIP conference hosts can be synchronized in time over the Internet using the network time protocol (NTP) [74]. The NTP model is comprised of two types of time servers, primary reference servers, which are directly connected by wire or radio to a national time standard broadcast source, and secondary servers, which derive synchronization from primary servers or other secondary servers and provide time synchronization to local subnets.

The following nomenclature is used to represent the various facets of clock synchronization. Clock stability refers to a clock's ability to maintain a constant frequency, clock accuracy refers to how closely its time and frequency compare with national standards and clock precision refers to how precisely stability and accuracy can be maintained in the system. Clock synchronization is defined in terms of a clock's offset, skew and drift. The time difference between two clocks is referred to as clock offset. The first derivative of offset is called skew, which refers to the difference in frequency between two clocks. The second derivative of offset is called drift, which is assumed to be zero for NTP synchronization purposes.

The amount of elapsed time required to obtain clock synchronization using NTP depends on the level of accuracy required. NTP can be used to obtain coarse time synchronization to within a second or so with only a few measurements. Time synchronization on the order of a millisecond, however, requires many hours of measurements to resolve offset and to compensate for oscillator skew [74].

A VoIP terminal, called a host in NTP, synchronizes its clock with one or more time servers, called peers in NTP, over the Internet. This process requires sending NTP messages, which contain local timestamp information concerning when the message is sent, previously received timestamp information and other information necessary for managing the NTP hierarchy and associations. The rate at which these messages are transmitted is determined based on the desired clock accuracy. When an NTP message is received, the host, and optionally the peer, computes the round trip delay between the host and peer, then computes the clock offset and dispersion between the host and peer and finally incorporates this information into a data base. Filtering algorithms are used to discard inferior data and clock selection algorithms are used to select the most accurate and reliable subnets. Dispersion is used as a metric for measuring the amount of error in the VoIP host clock with respect to the national standard reference clock.

Transmit and receive paths between a host and peer on the Internet can often exhibit asymmetric one-way delays. To reduce the effect of the asymmetric path errors, it has been found that combining the apparent clock offsets of a number of peer time servers produces a more accurate offset [74].

The level of accuracy required for echo suppression target determination depends on which portion of the TELR curve, see Fig. 42, the system is operating on. For small ARTD values the TELR curve has the greatest slope and it is important to resolve time accuracy to a finer granularity than with larger ARTD values. A primary reference time server that employs a GPS radio clock can achieve synchronization to within several tens of a microsecond [72], this option affords the best accuracy in portions of the TELR curve with a large slope and clock synchronization occurs almost instantaneously, but this method is also the most costly to implement. NTP synchronization with multiple secondary time servers over the Internet allows synchronization to within several milliseconds [72], depending on the stability and transmit and receive path symmetry of the NTP agents used. Synchronization with multiple secondary time servers provides good accuracy for echo suppression target estimation at a lower cost, but achieving good clock synchronization takes time.

7.3 Acoustic Round Trip Delay Measurement

Public service telephone networks and current VoIP systems are unable to measure the system round trip delay in the acoustic domain. This delay is required to determine accurate acoustic echo cancellation targets for hands-free full-duplex conferencing. Voice over Internet protocol networks can be enhanced to compute the acoustic round trip delay [1][75] and use it to determine accurate acoustic echo suppression targets through the methodology proposed by this thesis [71].

The amount of talker confusion and difficulty in maintaining normal conversation flow are affected by the ARTD. A conference system with long delays causes users to engage in double talk and mutual silence more often. The likelihood of a participant interrupting the conversation increases with the delay. As the delay increases, it is less likely that the participant will terminate the interruption and respond to questions or pauses in the conversation [76].

This thesis proposes computing the ARTD through the use of playout delay scheduling algorithms [64] with system clock synchronization or with the RTP control protocol (RTCP) round trip time (RTT) measurement mechanism. Both methods must also take into account acoustic delays, vocoder delays and jitter buffer delays, see Fig. 43.

Fig. 43. VoIP Packet Timing

The following notation is introduced to describe timing variables for a packet-audio stream. This notation is adapted from Moon et al. [53].

- t_k^i : the time when the sender generates packet i in the k -th talkspurt
- a_k^i : the time when the receiver receives packet i in the k -th talkspurt
- p_k^i : the time when the receiver plays out packet i in the k -th talkspurt
- d_k^i : the network delay experienced by packet i in the k -th talkspurt, where

$$d_k^i = a_k^i - t_k^i$$

- $\hat{p}_k^i$: the predicted playout delay for packet i of the k -th talkspurt
- f_A, f_B : the time taken to buffer a frame of speech and encode it using the vocoder for Participant A and Participant B, respectively
- c_B : the delay for the direct acoustic path between the reception room loudspeaker and microphone
- $c_{1,A}$: the delay for the direct acoustic path between the speaker's mouth and the transmission room microphone
- $c_{2,A}$: the delay for the direct acoustic path between the transmission room loudspeaker and the speaker's ear
- $RTT^{j,l}$: the total network round trip time determined from the j -th RTCP control packet from Participant A to Participant B and the l -th RTCP control packet from Participant B to Participant A

No clock synchronization between VoIP terminals is assumed for ARTD determination through the RTT method. However, for the playout delay prediction method, clock synchronization is required. (For most applications a suggested nominal value for clock synchronization is to within several tens of milliseconds).

The ARTD can be computed through the RTCP RTT mechanism without clock synchronization by the following equation:

$$ARTD_{RTT} = c_{1,A} + f_A + RTT^{j,l} + \hat{p}_{k,B}^i - a_{k,B}^i + c_B + f_B + \hat{p}_{k,A}^i - a_{k,A}^i + c_{2,A} \quad (74)$$

The ARTD can be computed with a playout delay prediction algorithm [64] assuming NTP clock synchronization has occurred at the two VoIP terminals by the following equation:

$$ARTD_{PD} = c_{1,A} + f_A + \hat{p}_{k,B}^i - t_{k,A}^i + c_B + f_B + \hat{p}_{k,A}^i - t_{k,B}^i + c_{2,A} \quad (75)$$

The main difference between the two different ARTD computation methods, other than the clock synchronization issue, is how the delay from the time the audio packet is transmitted from a participant to the time it is played back by the other participant is measured. The RTCP RTT method measures the network RTT in special RTP control packets that are separate from the audio packet stream and requires the addition of the jitter buffer delays, $RTT^{j,l} + \hat{p}_{k,B}^i - a_{k,B}^i + \hat{p}_{k,A}^i - a_{k,A}^i$, [35]. The playout delay prediction method computes the delay from transmit to playback directly using the RTP packet audio stream, $\hat{p}_{k,B}^i - t_{k,A}^i + \hat{p}_{k,A}^i - t_{k,B}^i$.

The c_B , $c_{1,A}$ and $c_{2,A}$ values can be measured using standard signal processing techniques such as a cross-correlation during a system training phase or a nominal value may be used since they are very small compared to other terms. Vocoder delays, f_A and f_B , include algorithmic delay and look-ahead delay [52]. The algorithmic delay is the delay to buffer a frame's worth of speech samples before the speech can be analyzed. The look-ahead delay is present in some vocoder's that require samples beyond the frame boundary for speech analysis. In this chapter, it is assumed that the same vocoder is used by both participants in the conference, therefore delays f_A and f_B are equal.

RTCP RTT is defined in [35] as the arrival time of the sender report (SR) from participant B minus the departure time of the sender report (SR) from participant A, see Fig. 44. The delay between participant B receiving participant A's j -th SR and participant B sending a return i -th SR to participant A is called the delay since the last sender report and is denoted by $DLSR^{j,i}$. A sender report is a control packet used to collect information about the current status of the network and to predict future network congestion. These packets typically occupy 5% of the total VoIP network bandwidth. The RTT is computed by the following equation:

$$RTT^{j,i} = a_A^i - t_A^j - DLSR^{j,i} \quad (76)$$

where $DLSR^{j,i} = t_B^i - a_B^j$, therefore

$$RTT^{j,i} = a_A^i - t_B^i + a_B^j - t_A^j \quad (77)$$

Fig. 44. Round-Trip Time Computation using RTCP

For both ARTD computation methods there are delays that are computed at Participant A that Participant B does not have direct access to and vice-versa. For example, if Participant A is using the RTT method then the jitter buffer delay for Participant B, $\hat{p}_{k,B}^i - a_{k,B}^i$, is unknown. If Participant A is using the playout delay prediction method then the transmit to playout delay for Participant B, $\hat{p}_{k,B}^i - t_{k,A}^i$, is unknown. These delays may be exchanged between VoIP terminals using RTP or RTCP packet extensions as specified in [35] or they may be approximated based on the equivalent delay at the local host. Estimating unknown network delays with the RTT and playout delay prediction ARTD computation methods can be very inaccurate due to asymmetry in the Internet transmit and receive paths as discussed

in section 7.1.1. It is highly recommended that this delay information be exchanged and not estimated. Unknown acoustic delay information should be approximated since it is typically much smaller than its network delay counterparts.

7.4 Echo Suppression Target Determination

The appropriate echo suppression target can be determined using the psycho-acoustic TELR data introduced in section 7.1 once the ARTD has been measured by either the RTCP RTT method, $ARTD_{RTT}$, or the playout delay prediction method, $ARTD_{PD}$, as discussed in the previous section. The appropriate talker echo loudness rating, $TELR_{VoIP}$, is determined by cross-referencing the ARTD with the psycho-acoustic TELR targets, see Fig. 45.

Fig. 45. Echo suppression target determination

Knowledge of the ARTD allows the system to automatically minimize the amount of required half-duplex signal attenuation and centre-clipping for the conference. In Fig. 45 a mean one-way propagation delay of 100ms requires a TELR of 40dB. The line at 40dB in Fig. 45 shows that this level of echo suppression can be obtained without the addition of any attenuation or centre-clipping from the NLP, $EL_{NLP} = 0$. Any ARTD in excess of 100ms requires that attenuation and/or centre-clipping be added to the conference from the NLP.

8 Conclusions

This thesis focuses on algorithms and techniques that can be used to enhance full-duplex hands-free audio for VoIP telephony systems. Typical VoIP and PSTN systems operate with monaural audio. This thesis proposed replacing monaural audio with synthetically generated stereophonic audio or even surround sound audio to enhance the listener's experience. Multiple loudspeaker channels are especially important in the case of a multiple participant VoIP conference since participants can be identified using spatial cues, not just voice recognition cues.

Hands-free VoIP conferences, typically, need to be able to provide large amounts of echo suppression due to the large end-to-end delays involved in transmitting voice over the IP networks. As the VoIP end-to-end delay increases, the requirement for echo suppression also increases. Full-duplex hands-free VoIP systems require acoustic echo cancellers, attenuators, voice activity detectors and other circuits that are necessary for controlling talker echo. The introduction of multiple loudspeakers means that the architecture of the acoustic echo cancellation structure must be rethought for synthetic stereo or surround sound. This thesis proposed a per-spatial-region acoustic echo cancellation structure appropriate for any number of loudspeakers.

Additional echo suppression is achievable by using a microphone array beamformer. VoIP conferences typically suffer from long end-to-end delays, and additional echo suppression by means of microphone array beamforming is beneficial. Microphone array beamformers also benefit the system by providing

reverberation and noise reduction, and microphone array beamformers can be made steerable to track the user's position, although not the focus of this thesis.

Voice over IP systems must use packet-scheduling algorithms for playout since IP networks introduce a significant amount of delay and jitter. This thesis dealt with playout algorithms for multiple participant conferences that adjust the one-way delay only during silence periods, which requires the use of a codec with silence suppression and voice activity detection. A packet playout-scheduling algorithm delays packets by enough time to reproduce the audio stream without any audible gaps but does not delay the packets enough to disrupt the normal flow of the conversation. These packet playout-scheduling algorithms attempt to simultaneously reduce the average lost packet rate, end-to-end delay and end-to-end delay variation. Unfortunately, these three criteria cannot be optimized simultaneously; therefore a compromise must be made. This thesis proposes a scheduling algorithm that is trained in advance using a library of collected packet delay traces, and switches between several network models in real-time using a hidden Markov model.

A metric for evaluating VoIP conferences in terms of the end-to-end delay that the user perceives is an important tool for system designers. Packet playout scheduling algorithms only provide a small piece of the picture. This thesis proposes a method for computing the acoustic round trip delay (ARTD) for multiple participant VoIP conferences based on packet playout scheduling algorithms such as the HMM packet scheduling technique or from round-trip time measurement information.

The acoustic round trip delay metric is shown to be applicable to computing echo suppression targets, which allows the system to reduce the amount of half-duplex attenuation added to the signal path. Half-duplex attenuation by means of the non-linear processor is undesirable since the system takes time to switch directions, can cause speech to be cutoff, and can switch in the wrong direction inadvertently due to background noise. Precise knowledge of the level of echo suppression required allows the system to minimize half-duplex attenuation automatically.

8.1 Synthetic Stereo Acoustic Echo Cancellation

The synthetic stereo acoustic echo cancellation structure with a single canceler per-spatial-region provides enhanced sound quality and localization cues to the near-end user. The proposed structure eliminates misconvergence caused by cross-correlation of the reference signals without requiring the addition of a front-end preprocessor, which could be audible to the listener or affect spatialization of the signal, or requiring more complex adaptive algorithms. The per-spatial-region synthetic stereo echo cancellation structure obtains good convergence with much lower complexity than other synthetic stereo or true stereophonic structures.

The per-spatial-region structure requires fewer echo cancelers than the per-channel structure when only one echo canceler is allowed to be active at a time, which is normally true except for interruptions in the conference. The per-spatial-region structure does require more echo cancelers than the per-channel structure

when multiple active spatial regions are allowed and there are more than two regions. The horizontal localization blur, size of the video display, and separation of the loudspeakers, however, limits the system to a reasonably small number of spatial regions. When the per-spatial-region structure is used with only one active region at a time, then the computational complexity is greatly decreased and convergence and tracking rates are increased.

The simplicity and effectiveness of the synthetic stereo acoustic echo cancellation structure with a single canceler per spatial region, therefore, makes the proposed structure an attractive option for full-duplex hands-free multiple participant VoIP conferences.

8.2 Synthetic Surround Sound Acoustic Echo Cancellation

The synthetic stereo acoustic echo cancellation structure is then extended to the proposed structure for supporting acoustic echo cancellation with synthetic surround sound for multiple participant VoIP conferencing, which can also be applied to virtual reality systems. The synthetic surround sound structure proposed in this thesis allocates a single acoustic echo canceler per spatial region, which eliminates the problem of reference signal correlation and echo canceler misconvergence. It was shown that the number of loudspeaker channels does not affect the convergence rate or tracking performance of this structure and that convergence is possible with multiple active spatial regions.

The Microsoft DirectX API was used to generate the synthetic surround sound used in the simulations since this is an industry standard used in virtual reality applications and many VoIP applications. Synthetic surround sound can also be generated using traditional ILD and ITD techniques.

The differences between synthetic stereo and synthetic surround sound AEC were discussed and it was shown that the synthetic surround AEC structure is a straightforward extension of the synthetic stereo per-spatial-region AEC structure.

8.3 Near-Field Beamforming and Synthetic Stereo Acoustic Echo Cancellation

Combining a microphone array beamformer with the synthetic stereo echo canceler produces an optimal acoustic front-end for full-duplex hands-free VoIP conferences. The synthetic stereo structure provides an enhanced audio experience for users in a multiple participant conferences by making it easier to distinguish among talkers placed in spatial regions. The single canceler per spatial region structure is used to perform echo cancellation without the fundamental problem of cross-correlation of the reference signals experienced by many other structures. The microphone array reduces noise and reverberation for near-end speech and provides additional ERLE compared to a single microphone for far-end echo. An additional 8.4 dB of ERLE was obtained for the configuration tested in this thesis.

Talker echo loudness rating curves from ITU-T G.131 and ETSI ETR 250 show that large amounts of echo suppression are required when large end-to-

end delays are present, and these are typical in VoIP conferences. Microphone array beamforming is an important ally in providing additional echo suppression without resorting to attenuation using the NLP.

8.4 HMM Playout Delay Prediction

The HMM playout delay prediction technique successfully combines the advantages of several previously proposed delay prediction techniques to increase playout delay prediction accuracy. The HMM technique advantage comes from a combination of statistical signal information with state transition information. The HMM technique is trained a priori off-line using a wide range of network delay traces at different times of the day and can, therefore, recognize a wide range of network delay patterns when predicting playout delays. The proposed technique is made generic by separating playout delay prediction into two components, the mean delay estimate and predicted quantized excess delay, which is predicted through the use of an HMM. No aging parameters are required with the HMM technique since the system is designed to switch seamlessly between pre-computed models in real-time operation based on probability.

Three criteria are introduced that influence the user's perception of the conference QoS: the average number of lost packets, the average end-to-end delay and the average end-to-end delay variation. Simulations show that the HMM technique predicts playout delays that are a good compromise between average end-to-end delay, end-to-end delay standard deviation, average packet loss rate and mean packet burst loss length. The HMM technique obtains a 30%

improvement in playout delay fluctuation over similar playout delay prediction techniques for the Sreenan data, which contributes to an improvement in quality of service from the user's perspective.

The proposed HMM technique for playout delay prediction is, therefore, considered a promising technique in compensating for Internet delay variation that adversely affects packet voice applications.

8.5 Acoustic Round Trip Delay and Echo Suppression Target Determination

Acoustic round trip delay (ARTD) is proposed as an important metric for determination of the end-to-end delay experienced by a hands-free VoIP conference participant. This metric can be used to determine appropriate echo suppression targets using psycho-acoustic echo and delay models. The ARTD can be computed using two different methods, the RTP control protocol (RTCP) round trip time (RTT) method or the playout delay prediction method. The RTCP RTT method does not require VoIP terminal clock synchronization, whereas the playout delay prediction method does. The RTCP RTT method requires coordination with the RTP control protocol, whereas the playout delay prediction method uses standard RTP speech packets and performs clock synchronization through a technique such as the network time protocol (NTP). Accurate determination of echo suppression targets allows the level of attenuation and centre-clipping provided by the nonlinear processor (NLP) to be minimized. It is advantageous to minimize attenuation and centre-clipping thresholds since the half-duplex attenuation and non-linear effects lower overall

conference signal levels, can be subject to switching time that makes normal conversation difficult and introduce noise. Public service telephone networks (PSTN) are currently unable to measure the ARTD but VoIP systems are able to use this information to minimize NLP attenuation and centre-clipping. This new nonlinear processor design methodology relies on ARTD rather than signal power to determine echo suppression thresholds, which is used to optimize the required level of attenuation and centre-clipping.

8.6 Summary of Contributions

8.6.1 Synthetic Stereo Acoustic Echo Cancellation

- Proposed a method for generation of synthetic stereo spatial image using ITDs and ILDs and an associated per-spatial-region acoustic echo cancellation structure that does not suffer from correlation of the reference signals.
- Reference signals are taken prior to spatialization, unlike other competing approaches, which eliminates the problem of echo canceller misconvergence.
- Proved through simulation that it is possible to have more than one acoustic echo canceller for multiple spatial regions active at a time and still achieve convergence.

8.6.2 Synthetic Surround Sound Acoustic Echo Cancellation

- Proposed a per-spatial-region acoustic echo cancellation structure appropriate for surround sound presented over multiple loudspeaker

channels. This structure does not suffer from correlation of the reference signals.

- The proposed per-spatial region structure for acoustic echo cancellation is shown to be applicable to any number of loudspeaker channels.

8.6.3 Near-Field Beamforming and Synthetic Stereo Acoustic Echo Cancellation

- Microphone array beamforming combined with the per-spatial-region acoustic echo cancellation structure was shown to improve near-end talker gain and increase echo suppression levels.

8.6.4 HMM Playout Delay Prediction

- Proposed a playout delay prediction algorithm that is trained a priori to recognize patterns using data from a wide variety of network topologies at various times of the day. Previously, other algorithms were trained on the fly.
- The proposed playout delay model is not adapted in real-time by adjusting model parameters, instead the HMM technique switches seamlessly between several models based on probabilities computed in real-time.
- The proposed algorithm focuses on three criteria: the number of lost packets, the one-way delay and the delay variation. Previously, other algorithms only focused on lost packets and one-way delay.

8.6.5 Echo Suppression Target Determination

- Proposed a metric called acoustic round trip time (ARTD) which is appropriate for computing the appropriate amount of acoustic echo suppression required for hands-free VoIP conferences.
- Proposed a method for computing echo suppression targets by combining ARTD with talker echo loudness rating (TELR) curves. TELR was originally designed for network cancellers in PSTN systems; this thesis extends TELR to acoustic echo in hands-free VoIP conferences.

8.7 Future Work

A method for combining audio presented over multiple channels using head-related transfer functions (HRTF) with an appropriate acoustic echo cancellation structure would add an additional degree of realism to the conference. HRTF are not currently a good choice for transaural presentation of audio over loudspeakers since it is difficult to track the listener's head position accurately. If the listener's head position is not tracked accurately the spatial audio effect is lost. The literature suggests several methods for tracking the listener's head position and determining appropriate loudspeaker decorrelation functions. However, these methods typically require elaborate hardware setups and large amounts of CPU bandwidth. A further complication is that the constant adjustment of the loudspeaker decorrelation functions causes convergence problems for the acoustic echo canceller. A method that combines some of the advantages of HRTF presentation of audio with a less

complicated listener's tracking procedure would be of great value. This method should also take into account the effect on the acoustic echo canceller. Perhaps the loudspeaker decorrelation function can be fed directly to the acoustic echo canceller to aid in tracking the room impulse responses.

The HMM playout delay prediction technique is different than other techniques proposed in the literature since it is trained off-line and then its model is applied to a wide range of network conditions. A real-time implementation of the HMM playout delay prediction technique and the construction of a formal subjective listening test or using an electronic mean opinion score tester would be useful in further evaluation of the effectiveness of this technique. A subjective test of this sort would be difficult to implement due to number of variables. The test must be performed over a large number of network configurations and simply relocating the subjects to different locations around the world is impractical. Simply simulating a wide range of possible network configurations with a network emulator may not be appropriate. The test should be run with a suitable number of different playout delay prediction techniques that must all be implemented in the test software. The user's perception of quality-of-service will be largely affected by the choice of the audio codec, the total audio path delay, the amount of echo, the comfort noise generation and error concealment scheme, interaction with the nonlinear processor and the experience of the subjects with VoIP systems. This thesis analyzed the HMM delay prediction technique using commonly applied metrics since they can easily be compared against existing literature. A real-time platform for subjective analysis would provide additional quality-of-service information although it may be difficult to setup such a

test and identify a methodology for isolating the playout delay prediction technique from the other components that interact with it.

Measurement of talker echo loudness rating (TELR) curves for hands-free VoIP would be an important metric for determination of echo suppression targets for hands-free VoIP. These curves could be used to enhance the material presented in chapter 7. These measurements should account for various room types with varying amounts of reverberation. These measurements should also be performed with varying amounts of background noise. This sort of subjective measurement would either require a large number of participants or perhaps an electronic mean opinion score tester may be used to estimate this new curve.

9 References

- [1] T. Yensen, M. Parperis, I. Lambadaris and R. Goubran, "Echo target determination using acoustic round trip delay for VoIP conferences," in *Proc. IEEE Int'l Symp. Circuits and Systems*, 1999, Vol. 4, pp. 348-351.
- [2] Int. Telecommunications Union Recommendation ITU-T G.131, "Control of talker echo," Aug. 1996.
- [3] S. Haykin, Adaptive Filter Theory, 2nd ed., Prentice-Hall, Englewood Cliffs, N. J., 1991.
- [4] T. Yensen, R. Goubran and I. Lambadaris, "A multiple participant CTI acoustic echo cancellation structure," in *Proc. IEEE Nordic Signal Processing Symp.*, 1998, pp. 193-196.
- [5] M. Knappe and R. Goubran, "Steady state performance limitations of full-band acoustic echo cancellers," in *Proc. IEEE Int'l Conf. Acoustics, Speech and Signal Processing*, 1994, pp. 2073-2077.
- [6] N. Birkett, Nonlinear Adaptive Filtering with Application to Acoustic Echo Cancellation, Ph. D Thesis, Carleton University, Ottawa, Canada, April 1997.
- [7] S. Makino, "Relationship between the "ES family" algorithms and conventional adaptive algorithms," *Proc. Int. Workshop on Acoustic Echo and Noise Control*, pp. 11-14, June 1995.
- [8] S. Gay and S. Tavathia, "The fast affine projection algorithm," in *Proc. IEEE Int'l Conf. Acoustics, Speech and Signal Processing*, 1995, pp. 3023-3026.
- [9] M. Montazeri and P. Duhamel, "A set of algorithms linking NLMS and block RLS algorithms," *IEEE Trans. on Signal Proc.* pp. 444-453, Feb. 1995.
- [10] J. Treichler, C. Johnson and M. Larimore, Theory and Design of Adaptive Filters, John Wiley and Son, Wiley, New York, NY, 1987.

- [11] B. Widrow and S. Stearns, Adaptive Signal Processing, Prentice-Hall, Englewood Cliffs, NJ, 1985.
- [12] A. Gilloire, "Recent advances in adaptive filtering algorithms for acoustic echo cancellation," Proc. Int. Workshop on Acoustic Echo and Noise Control, pp. 115-134, June 1995.
- [13] F. Amand, J. Benesty, A. Gilloire, Y. Grenier, "Multi-channel acoustic echo cancellation," in *Proc. IEEE Int'l Workshop on Acoustic Echo and Noise Control*, 1995, pp. 57-60.
- [14] S. Makino, J. Noebauer, Y. Haneda and A. Nakagawa, "SSB subband echo canceller using low-order projection algorithm," Proc. Int. Conf. Acoustics Speech and Signal Proc., pp. 945-948, May 1996.
- [15] Int. Telecommunications Union Recommendation ITU-T G. 168, "Digital network echo cancellers," Apr. 2000.
- [16] J. Chen, K. Yao and R. Hudson, "Source localization and beamforming," *IEEE Signal Processing Mag.*, vol. 19, no. 2, pp. 30-39, March 2002.
- [17] B. Van Veen and K. Buckley, "Beamforming: a versatile approach to spatial filtering," *IEEE ASSP Mag.*, vol. 5, no. 2, pp. 4-24, April 1988.
- [18] J. Flanagan, J. Johnston, R. Zahn and G. Elko, "Computer-Steered Microphone Arrays for Sound Transduction in Large Rooms," *J. Acoustic Society of America*, vol. 75, no. 3, Nov. 1985, pp 1508-1518.
- [19] J. Ryan, Near-field beamforming using microphone arrays, Ph. D Thesis, Carleton University, Ottawa, Canada, 1999.
- [20] F. Halsall, Data Communications, Computer Networks and Open Systems, 4th ed., Addison-Wesley, Don Mills, Ontario, 1996.
- [21] K. Jeffay, D. Stone and F. Smith, "Transport and display mechanisms for multimedia conferencing across packet-switched networks," *Computer Networks and ISDN*

Systems, vol. 26, no. 10, pp. 1281-1304, 1994.

- [22] Int. Telecommunications Union Recommendation ITU-T H.323, "Packet-based multimedia communications systems," 1998.
- [23] G. Thom, "H.323: The multimedia communications standard for local area networks," *IEEE Communications Mag.*, vol. 32, no. 12, pp. 52-56, Dec. 1996.
- [24] Int. Telecommunications Union Recommendation ITU-T H.245, "Control of communications between visual telephone systems and terminal equipment," 1995.
- [25] Int. Telecommunications Union Recommendation ITU-T H.225.0, "Media stream packetization and synchronization for visual telephone systems on non-guaranteed quality of service LANs," 1996.
- [26] Int. Telecommunications Union Recommendation ITU-T G.711, "Pulse code modulation (PCM) of voice frequencies," 1988.
- [27] Int. Telecommunications Union Recommendation ITU-T G.722, "7 kHz audio-coding within 64 kbit/s," 1988.
- [28] Int. Telecommunications Union Recommendation ITU-T G.723.1, "Dual rate speech codec for multimedia telecommunications transmitting at 6.4 and 5.3 kbit/s," 1995.
- [29] Int. Telecommunications Union Recommendation ITU-T G.728, "Speech coding at 16 kbit/s," 1992.
- [30] Int. Telecommunications Union Recommendation ITU-T G.729, "Speech coding for multimedia telecommunications transmitting at 8/13 kbit/s," 1995.
- [31] Int. Telecommunications Union Recommendation ITU-T H.261, "Video CODEC for audiovisual services at p x 64 kbit/s," 1993.
- [32] Int. Telecommunications Union Recommendation ITU-T G.263, "Video CODEC for narrow telecommunications channels at < 64 kbit/s," 1995.
- [33] Int. Telecommunications Union Recommendation ITU-T T.120, "Data protocols for multimedia conferencing," 1996.

- [34] Internet Engineering Task Force Proposed Standard RFC 2543, "SIP: Session initiation protocol," 1999.
- [35] Internet Engineering Task Force Proposed Standard RFC 1889, "RTP: A transport protocol for real-time applications," 1996.
- [36] Internet Engineering Task Force Proposed Standard RFC 1890, "RTP profile for audio and video conferences with minimal control," 1996.
- [37] I. Dalgic and H. Fang, "Comparison of H.323 and SIP for IP telephony signaling," in *Proc. of SPIE, Multimedia Systems and Applications II*, vol. 3845, 1999, pp. 106-122.
- [38] H. Schulzrinne and J. Rosenberg, "Internet telephony: architecture and protocols – an IETF perspective," *Computer Networks*, vol. 31, no. 3, 1999, pp. 237-255.
- [39] Int. Telecommunications Union Recommendation ITU-T Q.931, "ISDN user-network interface layer 3 specification for basic call control," 1998.
- [40] J. Blauert, *Spatial Hearing: The Psychophysics of Human Sound Localization*, revised ed. Cambridge, MA: MIT Press, 1997.
- [41] J. Bauck, "Generalized transaural stereo and applications," *J. Audio Eng. Soc.*, Vol. 44, No.9, pp. 683-705, Sept. 1996.
- [42] T. Yensen, R. Goubran and I. Lambadaris, "Synthetic stereo acoustic echo cancellation structure for multiple participant VoIP conferences," *IEEE Trans. Speech and Audio Proc.*, vol. 9, no. 2, pp. 168-174, 2001.
- [43] J. Benesty, D. Morgan, J. Hall and M. Sondhi, "Synthesized stereo combined with acoustic echo cancellation for desktop conferencing," *Bell Labs Tech. J.*, Vol. 3, No. 3, pp. 148-158, July-Sept. 1998.
- [44] J. Benesty, D. Morgan, J. Hall and M. Sondhi, "Synthesized stereo combined with acoustic echo cancellation for desktop conferencing," in *Proc. IEEE Int'l Conf. Acoustics, Speech and Signal Processing*, 1999, pp. 853-856.
- [45] S. Lipshitz, "Stereo microphone techniques... Are the purists wrong?," *J. Audio Eng.*

Soc., Vol. 34, No. 9, pp. 716-744, Sept. 1986.

- [46] M. Sondhi, D. Morgan and J. Hall, "Stereophonic acoustic echo cancellation – An overview of the fundamental problem," *IEEE Signal Processing Lett.*, Vol. 2, No. 8, pp. 148-151, Aug. 1995.
- [47] A. Hirano and A. Sugiyama, "A compact multi-channel echo canceller with a single adaptive filter per channel," in *Proc. IEEE Int'l Symp. Circuits and Systems*, 1992, pp. 1922-1925.
- [48] S. Makino, K. Strauss, S. Shimauchi, Y. Haneda and A. Nakagawa, "Subband stereo echo canceller using the projection algorithm with fast convergence to the true echo path," in *Proc. IEEE Int'l Conf. Acoustics, Speech and Signal Processing*, 1997, pp. 299-302.
- [49] J. Benesty, D. Morgan and M. Sondhi, "A better understanding and an improved solution to the problems of stereophonic acoustic echo cancellation," *IEEE Trans. Speech Audio Processing*, Vol. 6, No. 2, pp. 156-165, March 1998.
- [50] A. Gilloire and V. Turbin, "Using auditory properties to improve the behaviour of stereophonic acoustic echo cancellers," in *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, 1998, pp. 3681-3684.
- [51] J. Benesty, P. Duhamel and Y. Grenier, "A multichannel affine projection algorithm with applications to multichannel acoustic echo cancellation," *IEEE Signal Processing Lett.*, Vol. 3, No. 2, pp. 35-37, Feb. 1996.
- [52] D. Minoli and E. Minoli, Delivering Voice over IP Networks, Wiley Computer Publishing, Toronto, Canada, 1998.
- [53] S. Moon, J. Kurose and D. Towsley, "Packet audio playout delay adjustment: performance bounds and algorithms," *Multimedia Systems*, ACM/Springer, v. 5, n. 1, pp. 17-28, Jan 1998.
- [54] J. Rosenberg, L. Qiu, H. Schulzrinne, "Integrating packet FEC into adaptive voice

- playout buffer algorithms on the Internet,” in *Proc. IEEE Infocom*, vol. 3, 2000, pp. 1705-1714.
- [55] R. Ramjee, J. Kurose, D. Towsley, H. Schulzrinne, “Adaptive playout mechanisms for packetized audio applications in wide-area networks,” in *Proc. IEEE Infocom*, vol. 2, 1994, pp 680-688.
 - [56] J. Pinto and K. Christensen, “An algorithm for playout of packet voice based on adaptive adjustment of talkspurt silence periods,” in *Proc. Conf. on Local Computer Networks*, 1999, pp. 224-231.
 - [57] V. Matic, A. Bazant and M. Kos, “Predictive playout delay adaptation for voice over Internet,” in *Proc. Mediterranean Electrotechnical Conf.*, vol. 1, 2000, pp. 348-351.
 - [58] C. Sreenan, J. Chen, P. Agrawal and B. Narendran, “Delay reduction techniques for playout buffering,” *IEEE Trans. on Multimedia*, vol.2, no. 2, pp. 88-100, 2000.
 - [59] J. Benesty, F. Amand, A. Gilloire and Y. Grenier, “Adaptive filtering algorithms for stereophonic acoustic echo cancellation,” in *Proc. IEEE Int’l Conf. Acoustics, Speech and Signal Processing*, 1995, pp. 3099-3101.
 - [60] R. Dunlop, D. Shepherd, M. Martin, et al, Sams Teach Yourself DirectX 7 in 24 Hours, Indianapolis, IN: Sams Publishing, 1999.
 - [61] J. Ryan and R. Goubran, “Nearfield Beamforming for Microphone Array,” *Proc. IEEE Int’l Conf. Acoustics, Speech and Signal Processing*, Munich, Germany, April 1997, pp. 363-366.
 - [62] T. Parks and C. Burrus, *Digital Filter Design*, John Wiley & Sons, New York, NY, 1987.
 - [63] T. Laakso, V. Valimaki, M. Karjalainen and U. Laine, “Splitting the Unit Delay,” *IEEE Signal Processing Magazine*, Vol. 13, Jan. 1996, pp. 30-60.
 - [64] T. Yensen, J. Lariviere, I. Lambadaris, R. Goubran, “HMM delay prediction technique for VoIP,” *IEEE Trans. on Multimedia*, Accepted for publication.

- [65] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257-286, 1989.
- [66] L. Rabiner and B. Juang, Fundamentals of Speech Recognition, Prentice Hall, Englewood Cliffs, New Jersey, 1993.
- [67] P. Brady, "A technique for Investigating On-Off Patterns of Speech," *Bell Systems Technical Journal*, pp. 1-22, Jan. 1965.
- [68] P. Brady, "A model for generating on-off speech patterns in two-way conversation," *Bell Systems Technical Journal*, pp. 2445-2472, Sept. 1969.
- [69] H. Chan and V. Leung, "Reservation arbitrated access for statistical multiplexing voice traffic over dual-bus metropolitan area networks," *IEEE Journal on Selected Areas in Communication*, vol. 17, no. 1, pp. 42-50, Jan. 1999.
- [70] European Telecommunications Standards Institute Technical Report ETR 250, "Speech communication quality from mouth to ear for 3,1 kHz handset telephony across networks," 1996.
- [71] T. Yensen, R. Goubran, J. Lariviere, I. Lambadaris, "Echo suppression target determination for voice over IP," to be submitted to *Proc. Int. Conf. Acoustics Speech and Signal Proc.*
- [72] Internet Engineering Task Force Technical Report, RFC 2679, "A one-way delay metric for IPPM," 1999.
- [73] Internet Engineering Task Force Proposed Standard RFC 2475, "An architecture for differentiated services," 1998.
- [74] Internet Engineering Task Force Proposed Standard RFC 1305, "Network time protocol (version 3) – specification, implementation and analysis," 1992.
- [75] T. Yensen, M. Parperis, I. Lambadaris, and R. Goubran, "Determining acoustic round trip delay for VoIP conferences," in *Proc. IEEE Workshop Multimedia Signal Proc.*, 1998, pp. 161-166.

- [76] P. T. Brady, "Effects of transmission delay on conversational behavior on echo-free telephone circuits," *Bell Labs Technical Journal*, vol. 50, pp. 115-134, 1971.