Carnegie Mellon University

CARNEGIE INSTITUTE OF TECHNOLOGY

THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF Doctor of Philosophy

TITLE **Compensation for Nonlinear Distortion in Noise for Robust Speech Recognition**

PRESENTED BY **Mark Harvilla**

ACCEPTED BY THE DEPARTMENT OF

Electrical and Computer Engineering

ADVISOR, MAJOR PROFESSOR — DATE 10/27/14

DEPARTMENT HEAD — DATE 11/3/14

APPROVED BY THE COLLEGE COUNCIL

DEAN — DATE

Compensation for Nonlinear Distortion in Noise for Robust Speech Recognition

Submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Mark J. Harvilla
B.S., Electrical and Computer Engineering, University of Pittsburgh
M.S., Electrical and Computer Engineering, Carnegie Mellon University

Carnegie Mellon University
Pittsburgh, PA

October 2014

Supervisor:	Dr. Richard Stern Department of Electrical and Computer Engineering
Committee Members:	Dr. Bhiksha Raj Department of Electrical and Computer Engineering
	Dr. Ian Lane Department of Electrical and Computer Engineering
	Dr. Kornel Laskowski Voci Technologies

To my parents, the two most loving people I know.

Acknowledgments

On May 15, 2010, I thought I would be pursuing my graduate education at the University of Pittsburgh, where I was soon to complete my Bachelor's. I had applied to a wealth of graduate schools and I'd heard back from all of them, except one: Carnegie Mellon. Most graduate schools adhere to a mutual decision notification deadline so that prospective students can compare their offers from various schools to one another and make a fair decision. The schools agree to notify prospective students of acceptance or rejection well before this date, and the students must correspondingly notify the school by this date if they wish to accept an offer. This date was May 15.

Earlier in the day, I had spoken to the professor in charge of graduate admissions at Pitt about my dilemma, and he told me that I would have until 5 PM that day to let him know if I wanted to attend Pitt. By 3 PM, I still hadn't heard from CMU. At the very least, I wanted a definitive rejection, just to be sure I wasn't missing an opportunity by accepting Pitt's offer. So, I decided to walk over to CMU's campus. My first stop was Tara Moe's office; as the Associate Director of Graduate Affairs, she was the only person at CMU that I had communicated with in any capacity at the time. Unbeknownst to me, the CMU campus is designed to be navigated by highly intelligent people, as my initial impression was that of a circuitous labyrinth.

Eventually, I found Tara's office in the depths of Hamerschlag Hall and explained to her who I was and why I was there. She reassured me that she would attempt to contact the professors whose research I had found interesting, and in the meantime, I ventured back into the maze of CMU's buildings to find these professors myself. Most of it is a blur in memory, but at least an hour was spent meandering through the halls of Hamerschlag, Wean, and Roberts. I eventually found most of the professors' offices that I wanted to speak with, but not a single one of them was there. After exhausting my admittedly limited navigational intellect, I revisited Tara in Hamerschlag for an update. No responses. She assured me she would continue to do her best to contact some professors, but all in all, her best advice was for me to head back to Pitt.

At about twenty to five, I was walking towards Forbes Ave. past the CIC building, when my cell phone rang. It was an unfamiliar number. I picked it up, and from what I remember, the conversation went something like this.

"Hello," I said.

"Hi, this is Rich Stern," said the voice, "I'm a professor at CMU. I'd like to speak with you."

"Are you aware of the circumstances?" I responded, in reference to my stark lack of time.

"Yes," he said, "Can you come to my office?"

"Okay... Where is it?" At this point I panicked, recalling my marked inability to navigate the campus.

"Porter Hall, B24," Rich said.

"How do I get there? I've been kind of lost all day."

"Well, where are you now?" he asked.

"Um," I looked around utterly perplexed, "I have no idea."

"Where is your left hand?" he asked, in an attempt to understand my orientation. I'll never forget that line.

My left hand, incidentally, was at CIC. I don't recall if I stayed on the phone with Rich or not, but eventually I made it to his office. We spoke about research interests and life in graduate school, admittedly with a high sense of urgency, at least on my part. Eventually, the conversation came to a head, and Rich pronounced, "I have to make some calls." At this point, it was well past 5 PM; I called the admissions coordinator at Pitt to ask for a bit more time. Turns out, however, I didn't need it. When Rich finally got off the phone, he looked at me and said, "You're in." My face lit up. It was a dream come true. We talked a bit more, and before I walked out of the office, Rich extended his hand and said, "Welcome to Carnegie Mellon." I think he later saw me dancing in the parking lot outside.

Now, four-and-a-half years later, as I prepare to defend my Ph.D. thesis, I couldn't be any more thankful for the confluence of circumstances that day and Professor Stern's willingness to take a chance on me. Being a student at CMU has been a life-changing experience; I've met some of the brightest people I'll ever meet and have been awarded opportunities that I otherwise would not have imagined. However my career develops, it is inevitably forever changed for the better because of the events of that day.

In the nearly half of a decade that I've worked with him now, Professor Stern has been an invaluable source of advice, both personal and professional; my academic skills and the quality of my research work have flourished under his guidance. I'll also never forget how to properly

carry a harpsichord. I'd also like to personally thank Kornel Laskowski. Though I've only known him for a relatively short time, Kornel has been a major influence on the direction and quality of my research. Kornel is a brilliant individual; as my supervisor at Voci Technologies, I've had the pleasure of learning so much from him. His diligent approach to problems, his thorough analysis of results, and his seemingly endless drive to engineer a good product have been immensely inspiring.

Bhiksha Raj and Rita Singh are two wonderful, bright, and loving individuals that I'm very happy to have worked with. Both of them have extensively contributed to my work, creatively and intellectually, and their lighthearted, fun-loving personalities have always been refreshing in stressful times. Bhiksha's teaching skills are unparalleled; it is because of his intensive course that I truly understand how speech recognition systems work (and that's not an easy thing to explain). I'd also like to personally thank Horacio Franco, Martin Graciarena, Vikramjit Mitra, and countless others at SRI who played a role on the SCENIC team. The DARPA RATS project supported me during nearly all of my time at CMU, and its associated work served as the technical inspiration for the topics in this thesis. Also, a special thank you to Ian Lane is in order, for being a part of my thesis committee.

Finally, and perhaps most importantly, my friends and family are my rock. Their unconditional love, compassion, and understanding could not be more appreciated. Sometimes I think they know me better than I know myself. I have a remarkably diverse group of loving and supporting individuals in my life, and I wouldn't have accomplished very much at all without them. My mother and father are the two most selfless and loving individuals that I know. To know that they've always supported me, no matter the path I chose in life, is the best gift I've ever received. Everyday that I grow older, it becomes apparent just how lucky I am to be able to call them my parents.

Abstract

The performance, reliability, and ubiquity of automatic speech recognition systems has flourished in recent years due to steadily increasing computational power and technological innovations such as hidden Markov models, weighted finite-state transducers, and deep learning methods. One problem which plagues speech recognition systems, especially those that operate offline and have been trained on specific in-domain data, is the deleterious effect of noise on the accuracy of speech recognition. Historically, robust speech recognition research has focused on traditional noise types such as additive noise, linear filtering, and reverberation. This thesis describes the effects of nonlinear dynamic range compression on automatic speech recognition and develops a number of novel techniques for characterizing and counteracting it. Dynamic range compression is any function which reduces the dynamic range of an input signal. Dynamic range compression is a widely-used tool in audio engineering and is almost always a component of a practical telecommunications system. Despite its ubiquity, this thesis is the first work to comprehensively study and address the effect of dynamic range compression on speech recognition.

More specifically, this thesis treats the problem of dynamic range compression in three ways: (1) blind amplitude normalization methods, which counteract dynamic range compression when its parameter values allow the function to be mathematically inverted, (2) blind amplitude reconstruction techniques, i.e., declipping, which attempt to reconstruct clipped segments of the speech signal that are lost through non-invertible dynamic range compression, and (3) matched-training techniques, which attempt to select the pre-trained acoustic model with the closest set of compression parameters. All three of these methods rely on robust estimation of the dynamic range compression distortion parameters. Novel algorithms for the blind prediction of these parameters are also introduced. The algorithms' quality is evaluated in terms of the degree to which they decrease speech recognition word error rate, as well as in terms of the degree to which they increase a given speech signal's signal-to-noise ratio. In all evaluations, the possibility of independent additive noise following the application of dynamic range compression is assumed.

Contents

List of Figures

Chapter 1

Introduction

The overall intent of this thesis is to develop a set of algorithms to ameliorate the effect of nonlinear distortion on speech, with the primary goal of improving the accuracy of automatic speech recognition (ASR) systems and related technologies.

Nonlinear distortion, generally speaking, is any kind of transformation of a signal that does not preserve scaling and superposition. Some basic examples of nonlinear functions are x^2 and $|x|$. Certain types of nonlinear distortion appear more often in nature, due to the physical properties of the systems generating, transmitting, or processing signals such as speech. For example, the signal processing front end and transmission stages of telecommunications systems often give rise to nonlinear distortion in the form of dynamic range compression (DRC) and limiting (i.e., clipping), or to unnatural frequency shifting due to mistmatched modulation and demodulation.

While such distortions often do not render speech unintelligible to humans, signal nonlinearities can significantly degrade the performance of ASR systems. This is because ASR systems are trained on speech features derived from a frequency-domain representation of the signal, and nonlinear processes often alter the frequency spectrum of a signal in unpredictable ways. As with all noisy data, this causes the input speech features to be mismatched with the ASR system's clean speech models. Furthermore, it is often not possible to train the system on noisy or distorted speech, due to lack of representative data, ignorance of the true testing condition, or both.

Because the class of "nonlinear distortion" is literally infinite in size, this thesis focuses on one particular type−dynamic range compression−that is quite prevalent in real-world telecommunications systems. The mathematical definition for DRC employed in this thesis is borrowed from the field of audio engineering and is a standard way in which proprietary software implementations of DRC are often parameterized. The legitimacy of the larger degradation model used for algorithm development (i.e., the application of DRC to clean speech, followed by a layer of independent additive channel noise) is supported by research on the DARPA RATS project, whose simulated development data were generated according to a similar model. Finally, the problem of DRC, and especially clipping, is not unique to automatic speech recognition robustness, therefore the algorithms developed in this thesis may find much wider applicability in the audio engineering, machine learning, and signal processing worlds.

This thesis document is organized as follows. Chapter 2 presents an overview of historically-relevant robust speech recognition topics, including the fundamentals of how HMM-based ASR

works, how conventional feature extraction is performed, and traditional methods for counteracting additive noise (e.g., spectral subtraction and vector Taylor series). The second half of Ch. 2 develops the formal mathematical framework in which dynamic range compression distortion will be cast, shows how DRC affects the performance of ASR as a function of the DRC parameters, analyzes the relationship between DRC parameters and signal-to-noise ratio (SNR), and briefly summarizes prior research on nonlinear distortion.

Chapter 3 considers the specific case of DRC when the DRC function is invertible, which as will be shown, corresponds to the ratio parameter, R, being finite valued. When the DRC function is mathematically invertible, compensation for its effects becomes a matter of inferring the form of the function and then applying its inverse to the observed speech. This chapter outlines a previously-developed method for blind inference of a nonlinear function, herein termed blind amplitude normalization (BAN), which works by comparing the cumulative distribution function (CDF) of the distorted observation to an estimate of the CDF of clean speech. While this method is highly effective, its reliability breaks down in the presence of noise and consequently, a novel, robust version of BAN is introduced.

Next, Ch. 4 address the more difficult case of saturating (non-invertible) DRC, for which $R = \infty$. This subsumes a distortion commonly referred to as *clipping*, which limits the absolute maximum amplitude of a waveform to a certain level, τ; all values of the signal that exceed that level are mapped to $\pm\tau$. Because clipping is inherently non-invertible, methods that are more mathematically sophisticated than those of BAN are required for compensation. A wealth of research has been done in the area of declipping. Following a thorough literature review, novel declipping algorithms based on least squares interpolation are developed. Two novel algorithms are introduced, one which maximizes declipping performance at the expense of high computational complexity, and another which maximizes efficiency and speed at the expense of less precise declipping performance.

Chapter 5 considers the blind determination of the DRC function parameters, τ and R, from observed speech that may or may not be compressed. Related problems are the conversion of a given τ value to its corresponding percentile value, and the estimation of which samples are clipped in a segment of noisy speech. Subsequently, Chapter 6 presents a unique approach to robust speech recognition. Rather than attempting to directly compensate for the DRC distortion, a large set of acoustic models are trained on speech exposed to DRC with different pairs of parameter values.

Using the techniques of Ch. 5, the acoustic model trained on speech whose DRC parameters most closely match those estimated from the incoming observation is chosen during decoding.

The thesis concludes with Ch. 7, which presents a comprehensive framework for DRC compensation that integrates all of the work from Chapters 3, 4, 5, and 6. The purpose of this final chapter is to illustrate that the algorithms of this thesis can provide substantial improvements in a real-world situation when used in conjunction. Lastly, an overall summary of findings from the thesis, corresponding conclusions, and the most promising directions for future research are given in Ch. 8.

Chapter 2

Background

2.1 Overview of Automatic Speech Recognition

2.1.1 A brief history of ASR

Automatic speech recognition (ASR) refers to the process of using a computer to automatically transcribe spoken words into text. Despite the fact that fairly sophisticated ASR technology is now available on most modern cell phones (e.g., Apple's *Siri*), so-called speaker-independent (SI), large-vocabulary continuous speech recognition (LVCSR) has only become feasible in recent decades due to exponential growth of computational power (cf., Moore's Law [1]).

The first speech recognition systems−circa 1970−were only able to recognize words spoken in isolation by a known speaker; the technology was based on a concept proposed by Vintsyuk called *dynamic time warping* (DTW) [2]. These systems utilized whole-word models; that is, each word in the ASR's vocabulary had to be spoken and stored at least once by the speaker before using the system. Then, upon recognition, the input speech was sequentially compared to each of the whole-word models and the "closest" match gave the prediction of the input word. The DTW algorithm was used when comparing the input word to each of the models, as it is able to compensate for within-word variability in speaking rate, and thus allows for a much more flexible ASR.

The major breakthrough that formed the foundation of today's state-of-the-art ASR technology was the invention of the hidden Markov model (HMM) [3]. An HMM is a first-order probabilistic characterization of a time-varying process that allows for a maximum likelihood prediction of system state sequences given a series of observations. In ASR, the series of observations is the collection of samples of an electrical voltage waveform transduced by the microphone (which are subsequently encoded into a more compact and efficient representation called a *feature*), and the states are either words or some atomic unit of words (e.g., phonemes). The realization that all words can be represented by a small set of atomic sound units called *phonemes* allows for huge vocabularies to be modeled efficiently; for example, the entire English language can be represented using 41 phonemes [4].

2.1.2 The mathematical basis of speech recognition

Speech recognition is a special case of the more general Bayesian classification problem. Given a sequence of observations $\boldsymbol{X}$, determine the most like sequence of phonemes (or words) $\hat{\boldsymbol{W}}$:

$$\hat{\boldsymbol{W}} = \underset{\boldsymbol{W}}{\operatorname{argmax}} \Pr(\boldsymbol{W}|\boldsymbol{X}) \tag{2.1}$$

The expansion of Eq. 2.1 in terms of Bayes rule reveals the two primary system components of a speech recognition system, the *acoustic model*, and the *language model*:

$$\begin{aligned} \hat{\boldsymbol{W}} &= \operatorname{argmax}_{\boldsymbol{W}} \frac{\Pr(\boldsymbol{X}|\boldsymbol{W})\Pr(\boldsymbol{W})}{\Pr(\boldsymbol{X})} \\ &= \operatorname{argmax}_{\boldsymbol{W}} \underbrace{\Pr(\boldsymbol{X}|\boldsymbol{W})}_{\text{acoustic model}} \overbrace{\Pr(\boldsymbol{W})}^{\text{language model}} \end{aligned} \tag{2.2}$$

Using a large set of example utterances from a given language, the language model (LM) characterizes the probability of observing a given sequence of words in that language. LMs are based on the notion of an *n-gram*, which models the probability of a word or phoneme given the previous $n-1$ words or phonemes [5]. Typically, *bigrams* or *trigrams* are used, for which $n = 2$ or 3, respectively.

The acoustic model (AcM), on the other hand, characterizes the probability of observing a particular manifestation of a speech sound in the feature space. When speech is input to an ASR, the audio stream is broken up into overlapping frames−each frame is typically 25 milliseconds in duration, and a new frame of data is pulled from the audio stream every 10 milliseconds. As outlined in Section 2.1.3, each of these frames is transformed into a set of *cepstral coefficients*. For different manifestations of a particular speech sound (e.g., the *uh* in the word *run*), the cepstral coefficients cluster around particular values (see Fig. 2.4a), and it is exactly the shape of this clustering that the AcM captures. Acoustic modeling allows the ASR to make a probabilistically-optimal decision as to what sound is most likely being made and indeed the cepstral coefficients are what the computer effectively "hears."

2.1.3 Feature extraction

The time-domain waveform of speech is a very inefficient way of representing the information relevant to speech recognition. The motivation for speech *feature extraction*, i.e., the transformation of the time-domain audio into a set of cepstral coefficients, is twofold. First, it would be useful to achieve a more compact representation, so that the information relevant to recognition occupies a

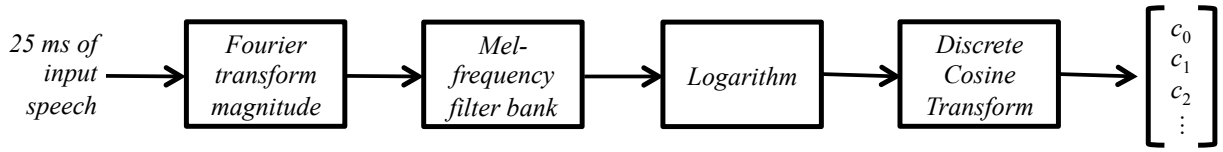


Figure 2.1: Transformation of one frame of speech into a set of Mel-frequency cepstral coefficients.

much smaller dimensional space (e.g., at 16 kHz sampling rate, 25 ms of speech occupies a 400-dimensional feature space; in contrast, approximately 10-15 cepstral coefficients can represent the same information more effectively). Second, a reduction in variability across speakers, ambient conditions, microphone type, distance from microphone, pitch, and so forth, is desired. Both of these goals are achieved with the use of the well-known *Mel-Frequency Cepstral Coefficient* (MFCC) representation [6]. The standard processing stages of MFCC extraction are shown in Fig. 2.1.

Fourier transform magnitude

Small deviations in the time offset between two waveforms makes a direct comparison between them more challenging. Sensitivity to such time offsets would be detrimental to speech recognition performance. By taking the Fourier transform of each frame of speech and then discarding the phase information, this problem is entirely avoided. Furthermore, one of the most useful characteristics of voiced sounds are their associated *formants*. Formants are large peaks in the speech spectrum that represent acoustic resonances of particular configurations of the vocal tract when making different vowel sounds. The locations, in frequency, of these formant peaks are relatively stable across speakers and are thus very reliable identifiers of vowels [7].

Mel-frequency filter bank

While it is generally more useful to analyze speech signals in the frequency domain, a standard discrete Fourier transform (DFT) magnitude contains at least as many real numbers as the underlying signal from which it is derived [8], and it may contain more if the underlying signal was zero padded to make the sequence's length a power of two. To simplify the acoustic modeling process, it is helpful to significantly reduce the dimensionality of the feature space [9].

In contrast to formants, another piece of information contained in the speech spectrum that is particularly *irrelevant* to speech recognition is pitch (provided one is recognizing a language

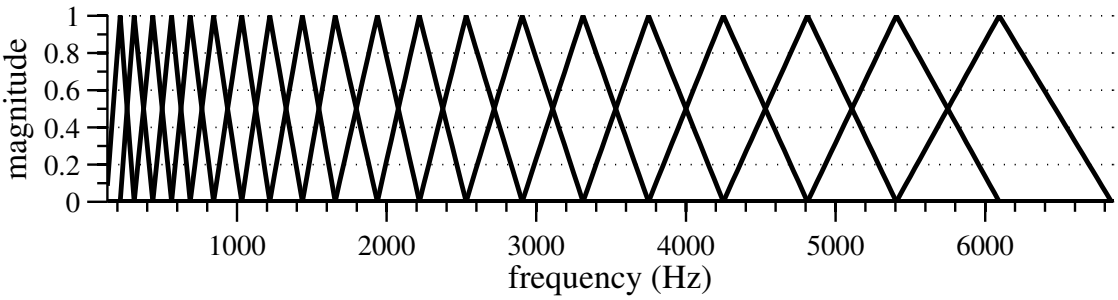


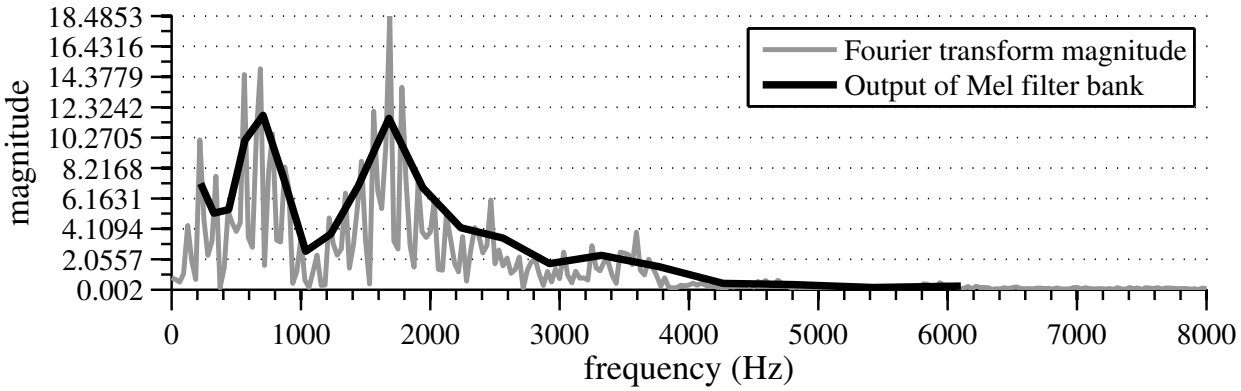
Figure 2.2: Frequency-selective weighting functions spaced in frequency according to the Mel scale.

unlike Mandarin Chinese for which pitch can convey meaning). Incidentally, pitch is also reflected by peaks in the spectrum, but pitch-related peaks occur rapidly and periodically over frequency, giving the raw speech spectrum a jagged, comb-like appearance. Figure 2.3 illustrates the difference between formant- and pitch-related peaks. Given these characteristics, it stands to reason that the dimensionality can be reduced by first *smoothing* the speech spectrum to remove superfluous pitch-related peaks (e.g., by application of a low-pass filter to the spectral magnitude), and then safely downsampling to reduce the feature dimensionality.

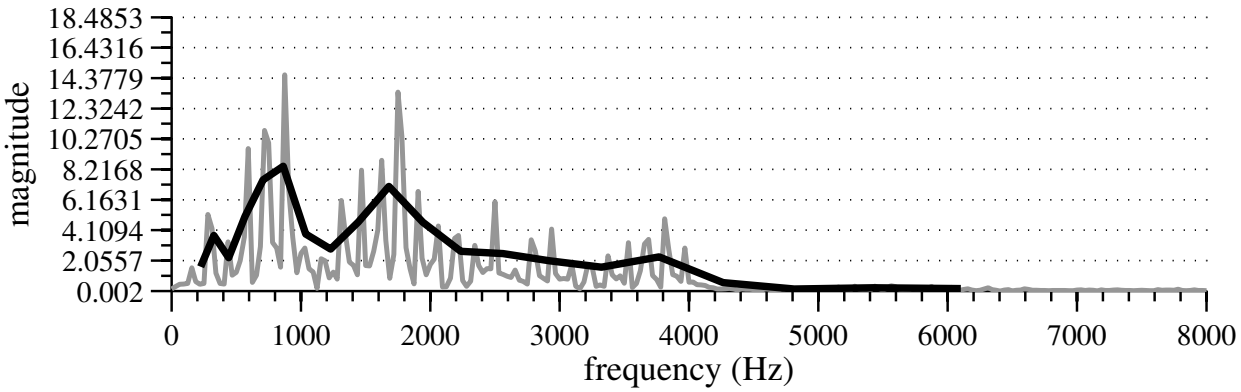
Rather than explicitly low-pass filtering and downsampling, however, MFCC processing efficiently achieves this smoothing and dimensionality reduction by applying a set of frequency-selective weighting functions to the magnitude spectrum. As depicted in Fig. 2.2, the weighting functions are typically triangular in shape and are spaced according to the perceptually-motivated Mel scale [10]. Each feature dimension is computed as the dot product of each triangular weighting function with the Fourier transform magnitude.

Logarithmic magnitude compression

The dynamic range of a typical speech spectrum often spans several orders of magnitude. The primary purpose of applying the logarithmic nonlinearity is to shrink the dynamic range of the observed spectrum, allowing small deviations to be more easily captured by the AcM. Some perceptual models of human hearing have been used to motivate similar compressive nonlinearities such as power-law functions of the form x^a where typical values for a range between 1/15 and 1/3 ([11],[12]).



(a) Spoken at a pitch of approximately 110 Hz.



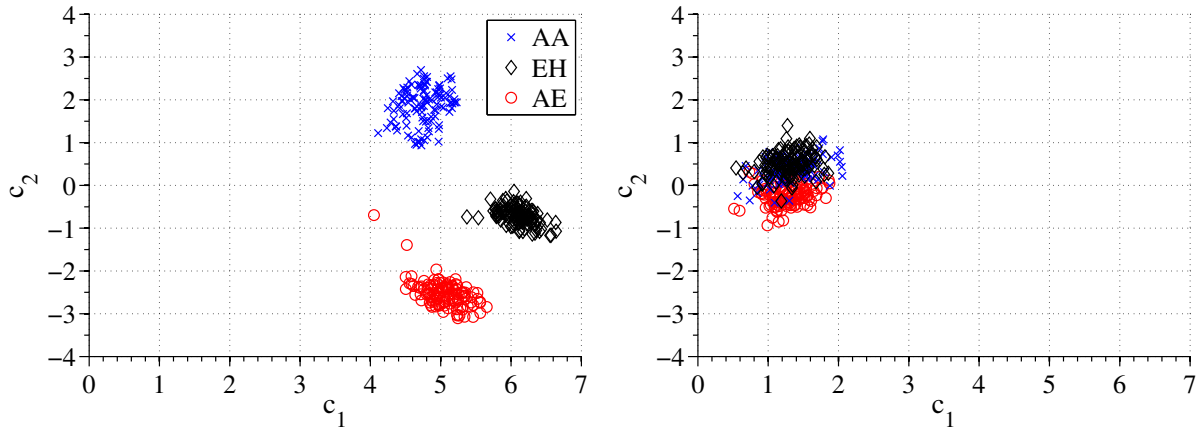
(b) Spoken at a higher pitch of approximately 150 Hz.

Figure 2.3: Spectrum of the vowel AE as in "fast" spoken at two different pitches by a male speaker. The separation of the rapid peaks in the spectrum reflect the pitch. Note that the peaks of the spectrum of the higher-pitched voicing are farther apart, which reflects the more rapid periodic fluctuation of the vocal chords and the consequential higher frequency of the pitch. The peaks in the output of the Mel filter bank represent the formants, which themselves reflect the resonances of the physical configuration of the vocal tract and mouth when making a particular vowel sound. As expected, the formant peaks are independent of the pitch of the voice and approximately match between the two spectra.

Discrete cosine transform

The discrete cosine transform (DCT) is similar in nature to the DFT, except that it represents a sequence as a linear combination of cosines instead of complex exponentials. There are four standard versions of the DCT that differ in their periodicity properties [13]. Discussions of the DCT in this thesis specifically refer to the DCT-2.

The DCT exhibits an *energy compaction* property; that is, the nature of the DCT is such that the energy of its coefficients are highly concentrated at low indices [14]. Consequently, truncating



(a) With no additive noise. (b) With additive white Gaussian noise at 0-dB SNR

Figure 2.4: Visualization of the clustering of cepstral coefficients in $[c_1, c_2]$ space for three different vowels. There is a clear loss of discriminability upon the addition of independent noise.

the DCT causes relatively little information in the signal to be lost. Indeed, this is the reason that the DCT is the basis for many lossy data compression standards such as MP3 audio encoding and JPEG image encoding.

Recalling that acoustic modeling is simplified for features with smaller dimensionality (i.e., less reference training data is required to achieve the same result [9]), application of the DCT in the final stage of feature extraction is well motivated. After applying the DCT, the feature vector is typically truncated to $10 - 15$ cepstral coefficients. Interestingly, the term *cepstral* coefficients derives from the so-called *cepstrum*, which is often defined as the inverse DFT of the log-magnitude of the DFT of a signal [15], a transformation similar in nature to MFCC processing.

2.1.4 Measuring ASR performance

Word error rate

To facilitate successful research in speech recognition, there must exist a standardized performance metric with which to evaluate the accuracy of a given ASR system. Such a metric does in fact exist, and it is referred to as the *word error rate* (WER). The WER measure is based on the Levenshtein distance between two text strings, and it measures the edit distance−i.e., the minimum number of word-by-word edits that must be made to transform the hypothesis sentence output by the ASR to the ground truth reference−between them. The word error rate is computed as the sum of the

number of substitutions, deletions, and insertions, divided by the number of words in the reference. Because the number of possible insertions is unlimited, it is possible for the WER to exceed 100%.

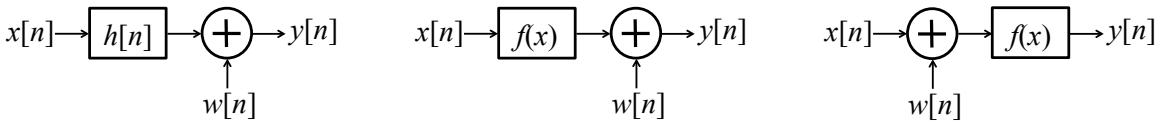
CMU Sphinx-III and MFCC configuration

The algorithms and concepts developed in this thesis are primarily evaluated using the CMU Sphinx-III speech recognition system [16] in conjunction with the DARPA Resource Management (RM1) database. Sphinx-III is a traditional HMM-based system. This particular configuration of Sphinx-III utilizes a standard bigram language model and an eight-component GMM-based acoustic model. The RM1 database is sampled at 16 kHz. Unless otherwise noted, MFCC features are extracted at 100 Hz frame rate, with a window duration of 25.625 ms. The Mel filter bank contains 40 filters spanning 133.3 Hz to 6855.5 Hz. Before feature extraction, each audio file is pre-emphasized with a high pass filter $H(z) = 1 - 0.97z^{-1}$. Cepstral mean normalization (CMN) [17] is applied before acoustic modeling. The baseline WER for the system in this particular configuration on this dataset is 6.4%.

2.1.5 Robust speech recognition

For a speech recognition system to perform optimally, the acoustic characteristics of the speech to be recognized should match those of the speech on which the AcM was trained as closely as possible. In practical usage, this is often not possible because of things like unpredictable environmental noise (e.g., noise from the cabin of a car or the cockpit of a plane), varying equipment characteristics (e.g., the use of different microphones in training and testing), transmission artifacts (e.g., signal dropouts, channel filtering, atmospheric noise), and so on.

Robust speech recognition research is associated with enhancing the reliability of ASR systems in noisy environments, such as any of those previously noted. Figure 2.4 demonstrates the effect of noise on the second and third Mel-frequency cepstral coefficients for three different vowels. Notice, in Fig. 2.4a, each vowel clusters in a distinct area of the cepstral space and has a unique shape. Following the addition of white Gaussian noise at 0-dB SNR, shown in Fig. 2.4b, the clusters for each vowel almost completely overlap and all tend toward a circular shape. The result of the addition of noise is a two-fold loss of discriminability between the vowels and a loss of identifiability of them individually.



(a) Linear channel filtering and independent additive noise.

(b) Nonlinear distortion followed by independent channel noise.

(c) Environmental noise followed by nonlinear distortion.

Figure 2.5: Three models of environmental degradation. The linear filtering and noise model is the classical basis for research in robust speech. Note that, because the filtering and addition of noise are linear operations, the ordering of the elements is mathematically arbitrary. In contrast, the two models that incorporate nonlinear distortion must be treated separately.

Traditionally, research in robust speech recognition has utilized the degradation model in Fig. 2.5a [18]. The pairing of a linear filter and additive noise allows a highly diverse set of noise phenomena to be modeled accurately. This includes things such as the effects of telephone lines and room reverberations, any sort of background noise like the hum of an air conditioner or speech babble (i.e., "the cocktail party effect"), and transmission channel noise. Further motivation for this model is the relative ease with which it can be analyzed mathematically. It can be shown that the cepstral coefficients derived from one frame of $y[n]$ in Fig. 2.5a are approximately equal to [19]:

$$\boldsymbol{C}_Y = \boldsymbol{C}_X + \underbrace{\boldsymbol{C}_H + \mathrm{DCT}(\log(1 + e^{\mathrm{IDCT}(\boldsymbol{C}_W - \boldsymbol{C}_X - \boldsymbol{C}_H)}))}_{g(\boldsymbol{C}_X, \boldsymbol{C}_H, \boldsymbol{C}_W)} \tag{2.3}$$

In Eq. 2.3, $\boldsymbol{C}_X$ is the set of cepstral coefficients of the clean speech, $\boldsymbol{C}_H$ the cepstral coefficients of the filter's impulse response, and $\boldsymbol{C}_W$ the cepstral coefficients of the independent additive noise. The $g(\boldsymbol{C}_X, \boldsymbol{C}_H, \boldsymbol{C}_W)$ term is referred to as the *environment function*, as it captures the way in which the value of the cepstral coefficients of degraded speech deviate from those of clean speech under the proposed noise model associated with the environment.

Cepstral mean normalization

In the absence of additive noise (i.e., with $w[n] = 0$ in Fig. 2.5a), the cepstral coefficients of one frame of $y[n]$ are approximately equal to:

$$\boldsymbol{C}_Y = \boldsymbol{C}_X + \boldsymbol{C}_H \tag{2.4}$$

If the filter's characteristics do not vary with time, which is the case in many practical situations,

then the $\boldsymbol{C}_H$ term in Eq. 2.4 is exactly the same from from frame to frame. That is, introducing a time dependency, where m is the frame index:

$$\begin{aligned} \boldsymbol{C}_Y[m] &= \boldsymbol{C}_X[m] + \boldsymbol{C}_H[m] \\ &\approx \boldsymbol{C}_X[m] + \boldsymbol{C}_H \end{aligned} \tag{2.5}$$

Therefore, the mean of $\boldsymbol{C}_Y$ over time is:

$$\bar{\boldsymbol{C}}_Y = \underbrace{\frac{1}{M}\sum_{m=0}^{M-1} \boldsymbol{C}_X[m]}_{\bar{\boldsymbol{C}}_X} + \boldsymbol{C}_H \tag{2.6}$$

Cepstral mean normalization (CMN) simply subtracts the time average of each cepstral coefficient from each feature vector. Note that, when Eq. 2.5 is satisfied, the following is true:

$$\boldsymbol{C}_Y - \bar{\boldsymbol{C}}_Y = \boldsymbol{C}_X - \bar{\boldsymbol{C}}_X \tag{2.7}$$

Thus, CMN completely eliminates the difference between filtered and unfiltered cepstral coefficients for a time-invariant filter. CMN is very effective in practice, and even provides substantial performance improvements in the presence of additive noise, despite the noise-free assumption made in developing the technique [20].

Spectral subtraction

Spectral subtraction is a classic noise compensation algorithm originally proposed by Boll [21] and improved by Berouti *et al.* [22] which essentially subtracts an estimate of the noise power spectral density (PSD) from the noisy speech spectrum. The PSD estimate of one frame of noisy speech is equal to [23]:

$$|Y[k]|^2 = |X[k]|^2 + |W[k]|^2 + 2|X[k]||W[k]|\cos(\theta) \tag{2.8}$$

Because the speech and noise are typically statistically independent signals, the cross term involving the product of the speech and noise spectra tends to zero on average, thus:

$$|Y[k]|^2 \approx |X[k]|^2 + |W[k]|^2 \tag{2.9}$$

If $|W[k]|$ is known, or can be accurately estimated, $|X[k]|$ can be approximately isolated. Specific implementations of spectral subtraction vary in (1) the way in which the noise spectrum is estimated and (2) the way with which artifacts of inaccurate estimates of the noise are handled. One prosaic method of noise spectrum estimation is to average together the spectra of non-speech frames. Unfortunately, this approach relies on the use of a speech activity detector (SAD), and the accuracy of many SAD algorithms often degrade in the presence of noise. Further, this approach is only effective for stationary (i.e., time-invariant) interference. Many other more sophisticated techniques for noise estimation have been proposed. For example, Martin proposed a technique for noise PSD estimation which tracks the spectral noise floor of the noisy speech on a frame-by-frame basis without the use of a SAD [24].

One of the most common artifacts associated with spectral subtraction is so-called *musical noise*. Any smoothed noise PSD estimate reflects the average value of the true noise PSD and consequently deviates randomly from any particular instance of the noise PSD in a given frame. When the estimate is subtracted from the noisy speech, the values of the true noise spectrum that fall below the estimate are set to zero, but those that exceed the estimate are not completely eliminated. These residual spectral peaks give rise to sporadic tonal interference called musical noise.

To minimize the presence of musical noise, a variant of spectral subtraction called *nonlinear spectral subtraction* was proposed by Berouti. In nonlinear spectral subtraction, the spectrum of one frame of noisy speech after noise subtraction is given by:

$$|\tilde{X}[k]| = \begin{cases} D[k] & \text{if } D[k] > \beta|\tilde{W}[k]| \\ \beta|\tilde{W}[k]| & \text{otherwise} \end{cases} \tag{2.10}$$

In Eq. 2.10, $D[k] = |Y[k]| - \alpha|\tilde{W}[k]|$, where $Y[k]$ is the noisy speech spectrum, $\tilde{W}[k]$ is the estimate of the noise spectrum, and α and β are algorithm parameters referred to as the *oversubtraction factor* and *spectral floor parameter*, respectively. In practice, $\alpha \geq 1$ and $0 < \beta << 1$; the α parameter causes the noise estimate to be "oversubtracted" from the noisy speech spectrum, thus

suppressing the residual peaks that lead to musical noise. In frequency bins (i.e., for particular values of k) for which the oversubtraction would result in undesirably small or negative values of $|\tilde{X}[k]|$, the result is floored at a small, non-zero value proportional to—by β—the noise spectrum estimate.

Vector Taylor series

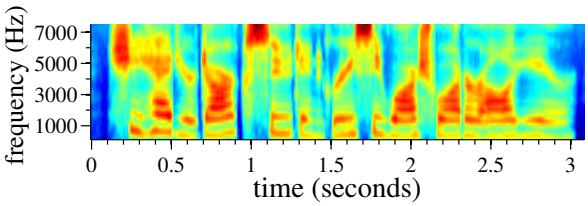
While CMN and spectral subtraction attempt to deal with the issues of filtering and additive noise distinctly, the *vector Taylor series* (VTS) approach to noise compensation is designed to tackle both forms of degradation simultaneously. VTS assumes the particular model of degradation shown in Fig. 2.5a and relies on pre-computed statistics of clean speech features.

By estimating the distribution of MFCCs of clean speech using a Gaussian mixture model (GMM) [25], the distribution of the MFCCs of noisy speech can be theoretically computed using the relationship given by Eq. 2.3. Unfortunately, because of the nonlinear term in Eq. 2.3, there is no closed-form solution for the distribution of noisy speech[1] [25]. VTS solves this problem by replacing the $g(\boldsymbol{C}_X, \boldsymbol{C}_H, \boldsymbol{C}_W)$ term with a zero- or first-order Taylor series approximation. The zeroth-order VTS solution models the effect of noise on clean speech only as a shift of the cepstral means. The first-order solution additionally incorporates the covariance matrix in capturing the effects of noise.

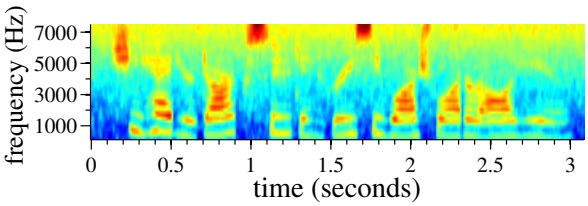
In practice, the VTS environment compensation algorithm uses an expectation maximization (EM)-style approach to iteratively estimate the values of the additive noise and the filter "parameters" (i.e., the values of $\boldsymbol{C}_W$ and $\boldsymbol{C}_H$ in Eq. 2.3), given only the observed noisy speech and the pre-computed GMM of clean speech. Subsequently, VTS computes the means and covariances of the GMM characterizing the noisy speech MFCCs. Finally, $g(\boldsymbol{C}_X, \boldsymbol{C}_H, \boldsymbol{C}_W)$ can be rewritten in terms of the observed noisy speech MFCCs, the GMM characterizing clean speech, and the Taylor series approximation of the environment function; the clean speech MFCCs then can be approximated as follows:

$$\tilde{\boldsymbol{C}}_X = \boldsymbol{C}_Y - g(\boldsymbol{C}_X, \boldsymbol{C}_H, \boldsymbol{C}_W) \tag{2.11}$$

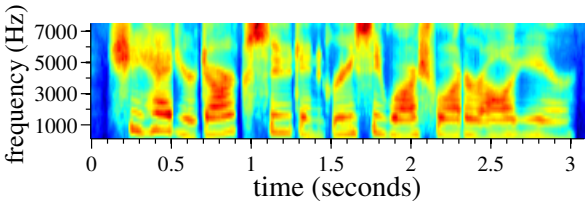
[1] Even if the posterior distribution of the noisy speech given the clean speech and noise parameters is assumed to be Gaussian, there is no closed-form solution for the mean and covariance.



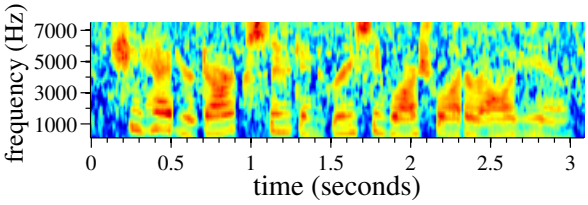
(a) Clean speech.



(b) Speech plus AWGN at 10-dB SNR.



(c) Clean speech processed with CSAWH.



(d) Speech plus AWGN at 10-dB SNR after CSAWH.

Figure 2.6: Log spectra of clean and noisy speech before and after CSAWH processing.

In practice, VTS is often performed on the so-called *log spectra*[2] instead of the cepstral coefficients for better performance.

Histogram-based noise compensation

The VTS approach to noise compensation uses Gaussian mixture models to characterize clean and noisy speech features in an attempt to make a statistically-optimal prediction of the clean speech log spectra or cepstral coefficients given only the noisy observation. In contrast to using GMMs, one can employ the use of a nonparametric distribution (e.g., a histogram) to undo the effects of noise on speech features. The CSAWH ("see-saw") algorithm [26], for *Compensatory Spectral Averaging and Warping using Histograms*, nonlinearly transforms noisy speech log spectra so that their histograms match those of clean speech log spectra. Log spectra of clean and noisy speech before and after CSAWH processing are shown in Fig. 2.6.

2.2 Dynamic Range Compression (DRC)

The effect of additive noise and filtering on automatic speech recognition performance, and a variety of contemporary techniques for counteracting the effects of such noise were reviewed in Section 2.1.5. Both additive noise and filtering are mathematically linear in nature, and as should

[2]The *log spectra* refers to the data in the MFCC processing of Fig. 2.1 after the logarithm and prior to the DCT.

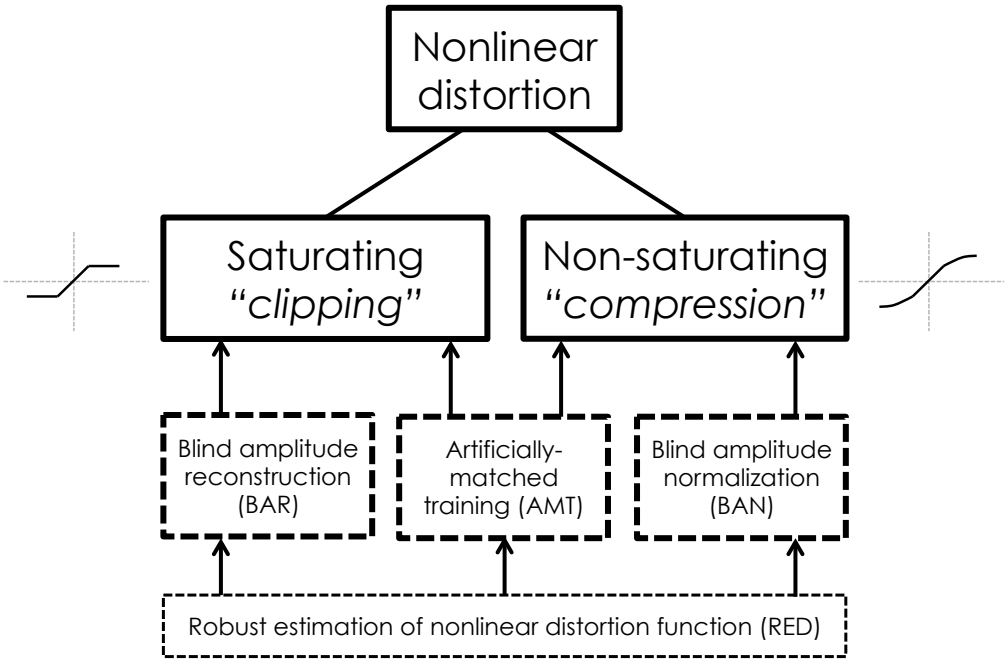


Figure 2.7: Illustration of the two fundamental types of nonlinear distortion considered in this thesis, *clipping* and *compression*, and the corresponding four major approaches to compensation. While BAR and BAN methods are exclusively designed to repair clipped or compressed speech, respectively, AMT techniques provide a comprehensive solution. As will be detailed later in the document, BAR, BAN, and AMT rely directly or indirectly on RED.

be clear from the literature review, there has been an extensive amount of work in the area. In contrast, the effects of *nonlinear distortion* on ASR performance have not been well studied. Nonetheless, nonlinear distortion is a practical consequence of many telecommunications schemes in widespread use (e.g., amplitude modulation (AM), frequency modulation (FM), and rectification are all nonlinear operations). The work in this thesis focuses on one particular type of nonlinear distortion referred to as *dynamic range compression* (DRC), its effects on the performance of speech recognition technology, and innovative techniques for counteracting or circumventing them.

DRC is ubiquitous in the field of audio engineering, used almost universally in television and radio broadcasts to increase perceived signal loudness[3] [27]. Furthermore, because most practical modulation schemes place constraints on the transmitted signal's dynamic range (e.g., AM, FM, PCM), DRC is often necessary, particularly for high crest-factor signals. The most commonly

[3]Perceived signal loudness is closely related to the average per-frame root-mean-square energy of the signal; by reducing signal peaks of a peaky waveform, the average energy of the overall signal can be greatly increased while still meeting the dynamic range limitations of a transmission system.

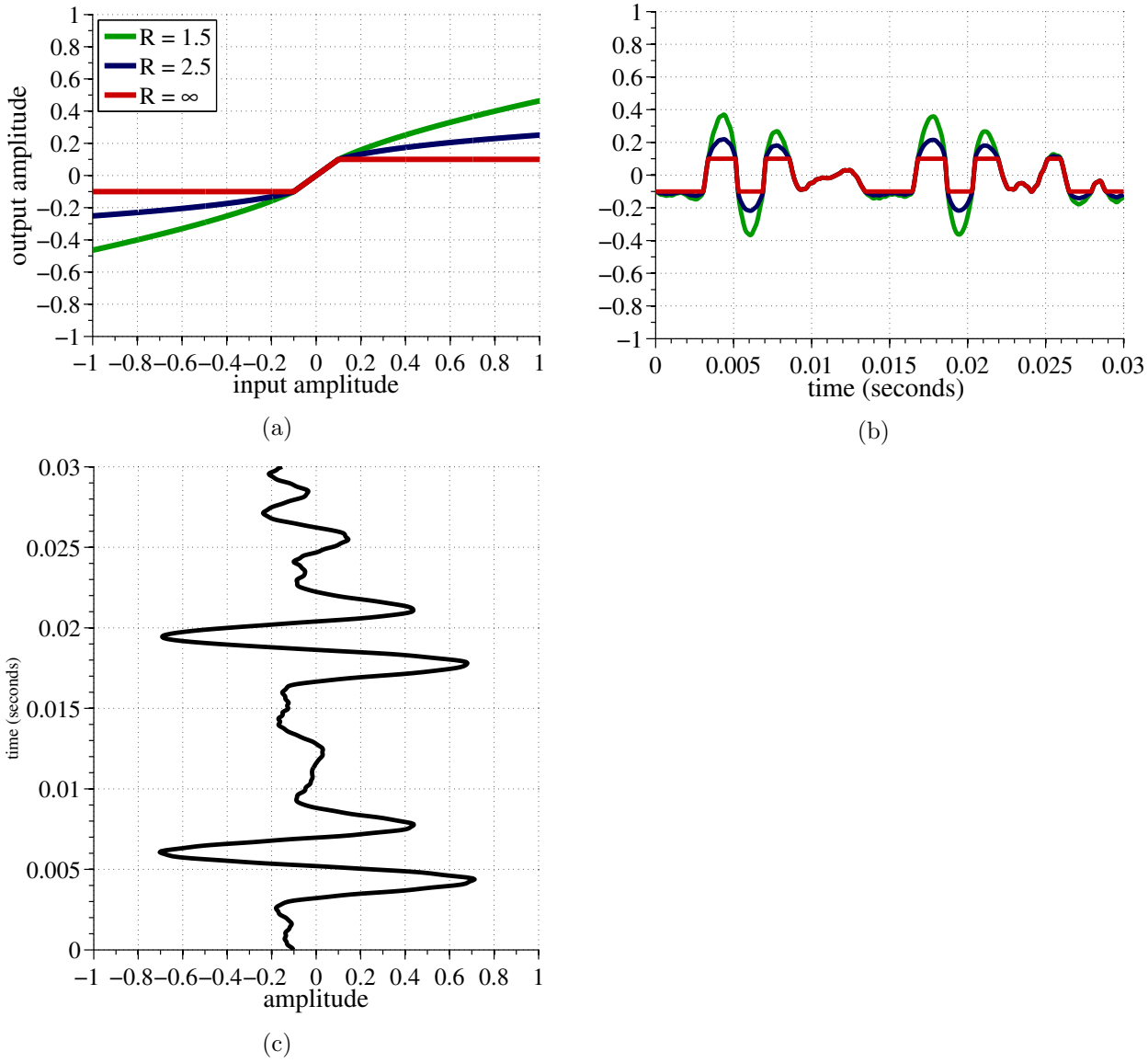


Figure 2.8: Illustration of the mapping of a segment of voiced speech through the DRC function for three different ratio values. Figure 2.8a shows the dynamic range compression function for values of R equal to 1.5, 2.5 and ∞. Figure 2.8c shows an example input signal to the DRC functions, which produce the outputs shown in Fig. 2.8b. All signals are drawn to scale. The plots are best viewed in clockwise progression from 2.8c to 2.8a to 2.8b. Note the decrease in the dynamic range of the output signals with increasing R.

transmitted auditory signals, speech and music, both have high crest factors (defined as the ratio of the peak amplitude of a waveform to its root-mean-square value) and so typically must be "squashed" via DRC. Beyond communications systems, compression is one of the most commonly-used tools in music production [28]. It is especially prominent on vocal tracks of pop music, and is one of the core components of the mastering stage of the music production process. In these cases,

the application of DRC is used to enhance some *perceptual* quality of the signal, e.g., perceived loudness, or consistency of volume, so that a particular audio track "sits well" in a mix. Finally, DRC encompasses the phenomenon of *clipping*: the complete loss of signal peaks due to a system being driven beyond its allowable dynamic range.

For all these reasons, it is quite likely that many acoustic signal capture, transmission, and understanding systems in practical use today and in the future will be exposed to dynamic range compressed speech at some point. This has strong implications for many speech applications inclusive of automatic speech recognition (ASR), keyword spotting (KWS), speech activity detection (SAD), automatic speaker clustering (ASC), and so on. As will be shown, decoding compressed speech with an ASR system trained on clean, uncompressed speech will lead to a significant rise in WER.

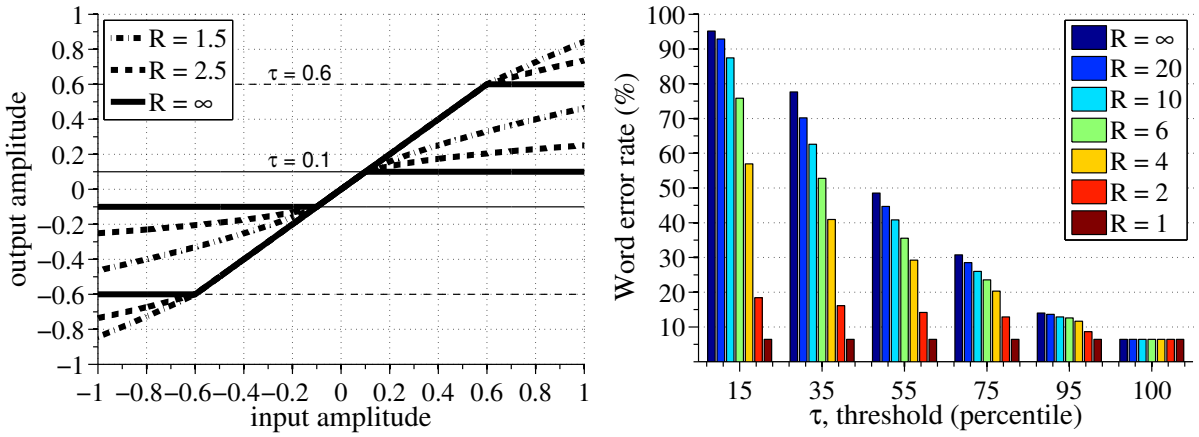
Throughout this work, DRC will be split into two major subtypes: (1) non-saturating nonlinear distortion, or *compression*, wherein the speech signal's dynamic range is reduced by an *invertible* nonlinear function and (2) saturating nonlinear distortion, or *clipping*, for which the speech signal's dynamic range is reduced by a *non-invertible* function. Figure 2.7 illustrates this dichotomy. The dashed boxes in the figure depict the four major compensation techniques that will be treated in the thesis. Methods designed to repair clipped signals will be collectively referred to as *blind amplitude reconstruction* (BAR) techniques; methods for inverting a compressive function will be referred to as *blind amplitude normalization* (BAN) techniques. *Artifically-matched training* (AMT) will theoretically treat both cases, and all three rely on *robust estimation of the nonlinear distortion function* (RED).

2.2.1 Mathematical characterization of DRC

A possible mathematical definition of an instantaneous dynamic range compressor [29] is:

$$f(x[n]) = \begin{cases} \tau^{(1-\frac{1}{R})}|x[n]|^{\frac{1}{R}} \operatorname{sgn} x[n] & \text{if } |x[n]| \geq \tau \\ x[n] & \text{if } |x[n]| < \tau \end{cases} \tag{2.12}$$

This is a piecewise nonlinear function parameterized by a threshold level, τ, which controls the absolute amplitude beyond which the nonlinearity takes effect, and a ratio level, R, which



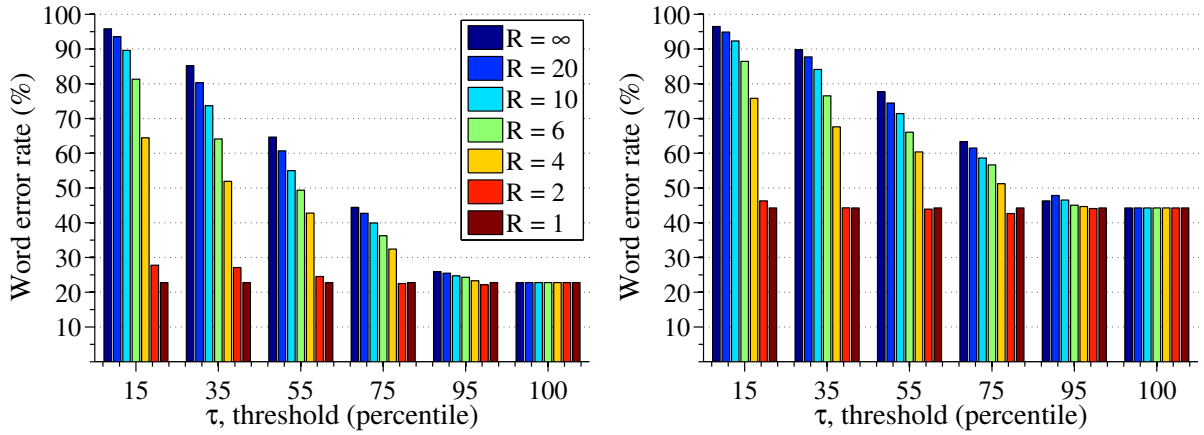
(a) Examples of DRC functions.

(b) WER as a function of DRC parameter values.

Figure 2.9: Figure 2.9a depicts the dynamic range compression function of Eq. 2.12 for various values of τ and R. For signal values normalized to the range $[-1, 1]$, their valid ranges are $0 < \tau \leq 1$ and $1 \leq R < \infty$. Figure 2.9b shows WER results using CMU Sphinx-III trained on clean speech with MFCC features. The effect of dynamic range compression on speech recognition is very detrimental. Recall that ratio $R = \infty$ represents clipping.

controls the intensity of the nonlinearity above the threshold. Examples of the nonlinear function for different values of τ and R are shown in Fig. 2.9a. Note that R is the inverse slope of the nonlinearity above the threshold in log-log space. When $R < \infty$, the DRC function has a unique inverse and, as will be shown in Ch. 3, it is relatively simple to blindly estimate the nature of the nonlinearity in the absence of further additive noise. When $R = \infty$, Eq. 2.12 simulates clipping, and the function no longer has a unique inverse.

A segment of voiced speech after compression with various values of R at $\tau = 0.1$ is shown in Fig. 2.8b. It is clear that the signal is unchanged below the threshold. There is a clear reduction in dynamic range with increasing R. Note that actual compression circuits and compression algorithms often employ additional parameters such as attack and release times, peak vs. RMS sensing, and hard-knee vs. soft-knee transitions. Non-zero attack and release times cause the compressor to gradually activate and deactive after the input signal exceeds or drops below the threshold, respectively. Rather than using the absolute peak value of the input signal to activate the compressor, RMS sensing uses the RMS value of the signal over a small time window. Finally, soft-knee transitions cause the compressor to gradually increase the effective ratio value as the input signal nears the threshold. This can be envisioned graphically as smoothing the transition between the linear and power-law segments of $f(x[n])$. All in all, these additional features do not



(a) With AWGN at 20-dB SNR.

(b) With AWGN at 15-dB SNR.

Figure 2.10: WER as a function of DRC parameters with AWGN using the degradation model of Fig. 2.5b.

change the basic functionality of the compressor, and relate primarily to perceptual characteristics of the output signal. To be thorough, Eq. 2.12 describes a peak-sensing, hard-knee compressor with instantaneous attack and release times.

2.2.2 Effect of DRC on automatic speech recognition performance

Dynamic range compression has an interesting effect on the performance of speech recognition. Figure 2.9b shows WER results for recognition experiments run on the RM1 ASR experiment described in Section 2.1.4. These results illustrate that, for a fixed ratio, R, the WER as a function of τ is approximately linear. In contrast, the WER as a function of R with τ fixed appears to rise exponentially and saturate as R approaches ∞.

As noted, a common situation is that either independent *channel noise* is added after the application of the nonlinearity, or independent *environmental noise* is present before the nonlinearity. The degradation models for these situations are depicted in Figs. 2.5b and 2.5c, respectively. The addition of white noise after DRC, as in Fig. 2.5b, has a rather unsurprising effect on ASR performance; results are shown in Fig. 2.10. With respect to Fig. 2.9b, the minimum of the WER values is progressively increased with decreasing SNR[4].

[4]For these experiments, the SNR is measured with respect to the compressed signal.

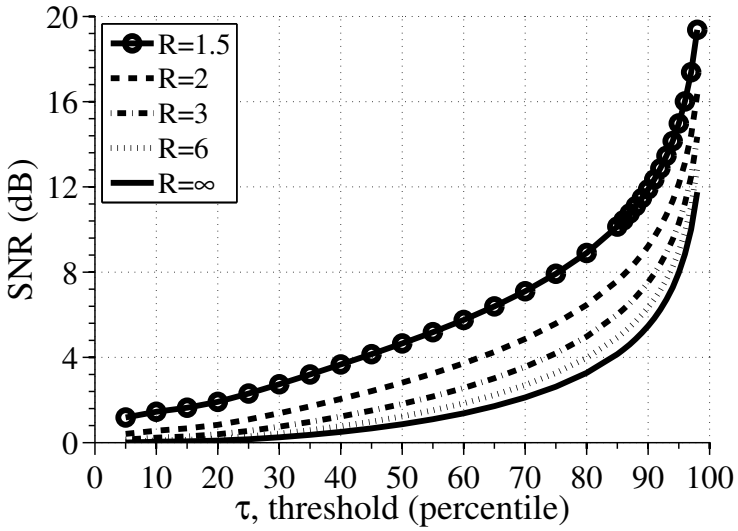


Figure 2.11: Mean SNR of a set of speech audio files as a function of the DRC parameters, τ and R. Note that these particular SNR values were computed according to Eq. 2.13 and are not associated with any independent additive noise.

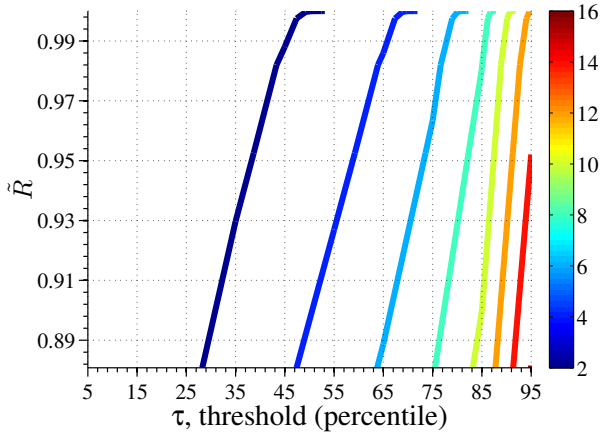
2.2.3 Relationship between DRC and signal-to-noise ratio

The bar graphs in Figs. 2.9b and 2.10 depict the WER of the ASR system in terms of the DRC parameters R and τ. As can be seen from the trends in WER, this is a useful and intuitive characterization of the nonlinear distortion (i.e., as expected, the WER increases with increasing R and decreasing τ, in some cases, nearly linearly). Nonetheless, it is not obvious how these parameters relate to the more common measure of noise intensity: signal-to-noise ratio (SNR). Despite the fact that SNR is usually computed under the assumption that the noise source is statistically independent of the signal−which, in the case of DRC, it is not−the SNR can be approximated as follows, where $x[n]$ is a clean speech signal and $f(x[n]; R, \tau)$ is the output of Eq. 2.12 given particular values of R and τ:

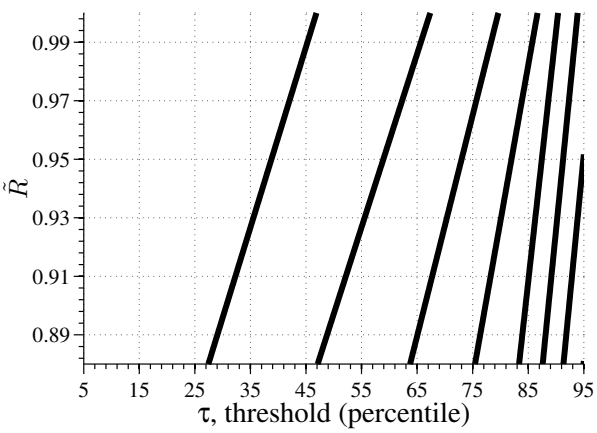
$$\mathrm{SNR}(R,\tau) = 10\log_{10}\left(\frac{\sum_{n=0}^{N-1} x^2[n]}{\sum_{n=0}^{N-1}\left(f(x[n];R,\tau)-x[n]\right)^2}\right) \tag{2.13}$$

Using Eq. 2.13, the average SNR[5] associated with each WER in Fig. 2.9b could be computed.

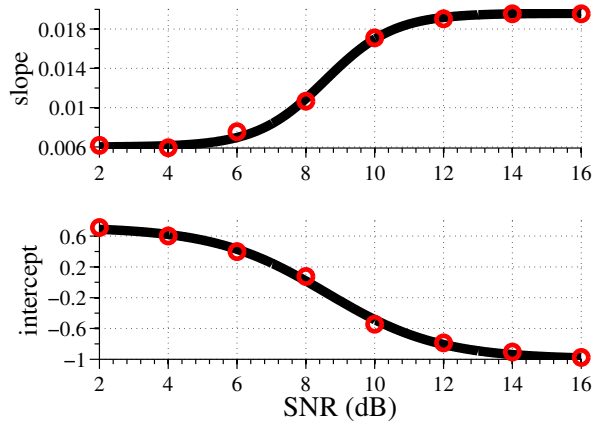
[5]Only the average SNR associated with each WER can be computed because, even for fixed R and τ, the SNR will vary across audio files.



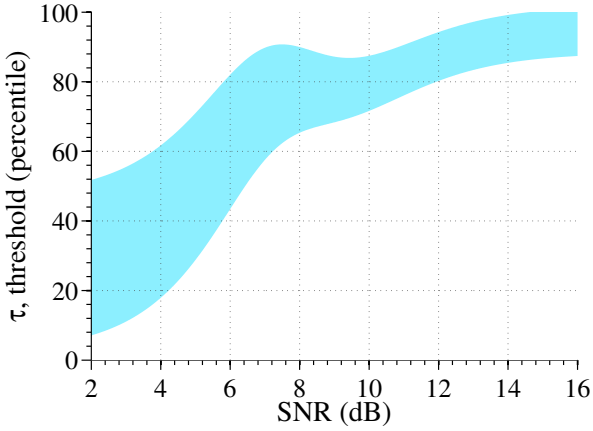
(a) The curves indicate points in $(\tilde{R}, \tau)$ space that correspond to equal SNR. Each color corresponds to a particular fixed SNR value.



(b) Result of simple linear regression on the isoclines of Fig. 2.12a to the left. These fits ignore the unusual behavior near $\tilde{R} = 1$.



(c) Approximate relationships between SNR and slope (top) and intercept (bottom) of the linear fits to the isoclines of Fig. 2.12a.



(d) The turquoise shaded region indicates the region in (SNR, τ) space which corresponds to legitimate values of $R \geq 1$.

Figure 2.12: Illustration of the computational stages required to determine the relationship between DRC parameters and SNR.

The average SNR of the full test data set of the RM1 database, as a function of R and τ, is shown in Fig. 2.11. In contrast, given only a desired SNR value, it is not as simple to determine the associated R and τ parameters because the mapping from R and τ to SNR is many-to-one (i.e., there are various (R, τ) pairs that map to the same SNR). In this section, a procedure is developed for obtaining R and τ values yielding a desired SNR.

To illustrate the many-to-one property, first define $\tilde{R} = \frac{1}{1+e^{-R}}$. Figure 2.12a shows *isoclines* of SNR in $(\tilde{R}, \tau)$ space. All the points that lie on an isocline equate to the same SNR. Replacing R by $\tilde{R}$ causes the isoclines to be approximately linear. Ignoring the nonlinear behavior of the isoclines

SNR (dB)	slope	intercept
2	0.006168	0.7108
4	0.005923	0.601
6	0.007553	0.3986
8	0.01066	0.07606
10	0.0171	-0.5453
12	0.01904	-0.7878
14	0.01955	-0.9056
16	0.01955	-0.9762

Table 2.1: Slope and intercept parameters obtained by performing linear regression on the isoclines in Fig. 2.12a to obtain the lines in Fig. 2.12b, each defined by the equation $\tilde{R} = \text{slope} \cdot \tau + \text{intercept}$.

near $\tilde{R} = 1$, simple linear regression gives the line approximations to the isoclines shown in Fig. 2.12b and defined by the slope and intercept parameters in Table 2.1. Further, the data points in Table 2.1 are shown as red circles in Fig. 2.12c. By fitting logistic functions to the slope and intercept parameters as a function of SNR, the following approximate relationships are obtained:

$$\text{slope}(\text{SNR}) = \frac{0.0136}{1 + e^{8.568 - \text{SNR}}} + 0.006 \tag{2.14a}$$

$$\text{intercept}(\text{SNR}) = \frac{1.718}{1 + e^{0.611(\text{SNR} - 8.611)}} - 1 \tag{2.14b}$$

The solid lines in Fig. 2.12c illustrate Eq. 2.14 graphically. Finally, for a given SNR, the valid region in (R, τ) space must be determined:

$$1 \leq R < \infty \tag{2.15a}$$

$$-1 \geq -R > -\infty \tag{2.15b}$$

$$1 + e^{-1} \geq 1 + e^{-R} > 1 \tag{2.15c}$$

$$\frac{1}{1 + e^{-1}} \leq \frac{1}{1 + e^{-R}} < 1 \tag{2.15d}$$

$$\frac{1}{1 + e^{-1}} \leq \tilde{R} < 1 \tag{2.15e}$$

That is, $\tilde{R}$ lies on the half-closed interval, $[\frac{e}{1+e}, 1)$. Recalling that $\tilde{R} = \text{slope(SNR)} \cdot \tau + \text{intercept(SNR)}$, the following bounds on τ as a function of SNR are implied:

$$\frac{1}{\text{slope(SNR)}} \left(\frac{e}{1+e} - \text{intercept(SNR)} \right) \leq \tau < \frac{1 - \text{intercept(SNR)}}{\text{slope(SNR)}} \tag{2.16}$$

This relationship is depicted graphically by Fig. 2.12d. Without loss of generality, one could alternatively fix the bounds of τ and determine the corresponding bounds on $\tilde{R}$ as a function of SNR. Given this information, the following steps can be taken to acquire a (R, τ) pair to achieve some desired SNR:

1. Use Eq. 2.14 to compute the slope and intercept of the associated $(\tilde{R}, \tau)$ isocline.
2. Obtain an appropriate value of τ within the bounds specified by Eq. 2.16, either manually or stochastically.
3. Compute $\tilde{R} = \text{slope(SNR)} \cdot \tau + \text{intercept(SNR)}$
4. Compute $R = -\log(\frac{1}{\tilde{R}} - 1)$

2.3 Prior research on compensation for nonlinear distortion

Despite the very serious degradation that DRC entails for ASR, relatively little work has been done on the problem. Of the work that has been done, most concerns the reconstruction of clipped signals. Admittedly, in the absence of superimposed additive noise, correction for invertible DRC (i.e., when $R < \infty$) is straightforward, as will be shown in Chapter 3. Nonetheless, compensation for nonlinearly distorted speech in the presence of noise has not been addressed and is not a simple problem. It appears that no work has been done specifically addressing the problem of repairing dynamic range compressed speech in noise.

Approaches to the reconstruction of clipped speech are varied. One of the most common themes is the use of an autoregressive model to predict the missing samples, e.g., as in linear predictive coding (LPC) [30]. Perhaps the most widely-cited work that utilizes autoregressive modeling for speech declipping was done by Janssen *et al.* [31]. Dahimene *et al.* also use LPC directly, by estimating the prediction coefficients from the unclipped samples, and filling in the clipped portions

using backwards prediction [32]. Work by Fong and Godsill utilizes autoregressive modeling, but not directly. It is instead used as the underlying statistical model of a particle filter from which a sequence of samples is drawn, and then used to fill in those that are missing [33]. An analysis of clipped signal restoration techniques based on bandwidth constraints was presented by Abel [34]. Other more recent approaches include reconstructions based on sparsity ([35], [36]) and recursive vector projection [37].

In all of this prior work, the quality of the clipped speech reconstruction is typically measured by either the mean-squared error between the original clean speech and the clipped speech, or by subjective listening experiments. It appears there has been no comprehensive review of the effect of clipping, dynamic range compression, or any associated restoration algorithms on the performance of automatic speech recognition or other speech-based systems, aside from the modicum of speaker identification results reported in [38].

Chapter 3

Blind Amplitude Normalization (BAN)

3.1 Introduction

One approach to compensating for the effects of dynamic range compression is by inverting the DRC function. From Eq. 2.12, the inverse DRC function is:

$$x[n] = \begin{cases} \tau^{1-R}|f(x[n])|^R \operatorname{sgn} f(x[n]) & \text{if } |f(x[n])| \geq \tau \\ f(x[n]) & \text{if } |f(x[n])| < \tau \end{cases} \tag{3.1}$$

Provided R is finite, so that Eq. 3.1 converges, knowledge of the DRC parameters τ and R is sufficient to undo the compression. Such *parametric* techniques are dicussed in Chapter 5. This chapter, however, introduces a classic *nonparametric* technique for inferring the inverse of any type of instantaneous nonlinear amplitude distortion, including that of DRC.

3.2 The Balchandran Algorithm

Balchandran and Mammone proposed an algorithm, herein referred to as the Blind Amplitude Normalization (BAN) algorithm, to undo the effects of instantaneous nonlinear amplitude compression or expansion in the context of speaker identification [38]. BAN is essentially a CDF matching algorithm, utilizing the same principle as CSAWH from Section 2.1.5.

Blind inference of the DRC function using BAN requires an estimate of the CDF of clean speech, which should be easy to obtain. The fundamental working principle of the algorithm can be understood by realizing that the "probability" of a given sample point in the input waveform remains the same after passing through the DRC function, the sample just takes on a new value. Mathematically, this means the following, where x is the clean speech waveform sample, y is the compressed waveform sample (i.e., the output of Eq. 2.12), and C_X and C_Y are their corresponding cumulative distribution functions:

$$C_X(x) = C_Y(y) \tag{3.2}$$

Equation 3.2 also implies that the DRC function itself is given by:

$$f(x) = C_Y^{-1} C_X(x) \tag{3.3}$$

Similarly, the inverse DRC function is given by:

$$x = f^{-1}(y) = C_X^{-1} C_Y(y) \tag{3.4}$$

Balchandran and Mammone similarly propose an efficient mechanism for estimating the CDF of a data set. The value of the CDF of a given sample represents the probability of an instance of the random variable (RV) falling below that value. That is, the CDF of an arbitrary RV X is:

$$C_X(x) = \Pr(X \leq x) \tag{3.5}$$

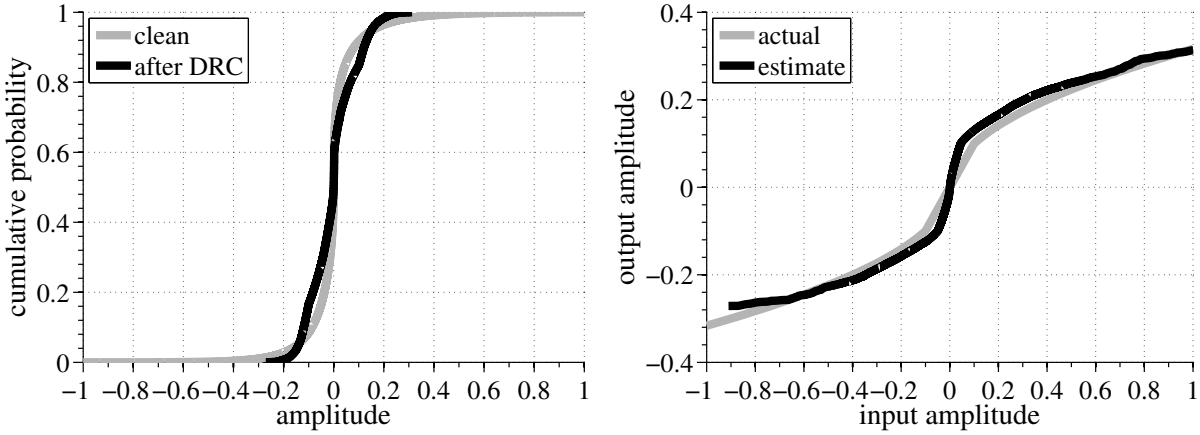
This implies that, given a relatively large data set of N independent, identically-distributed (IID) samples of a particular RV, the CDF can be estimated by counting, for each sample, the number of samples that have a lower value and then dividing by the total number of samples in the data set. If the data are sorted and no sample value is repeated, there are $i - 1$ samples in the set lesser in value than the i^{th} sample. Therefore,

$$C_X(x_i) \approx \frac{i}{N} \tag{3.6}$$

By assuming an audio waveform is a set of IID samples from a fixed probability distribution, estimating the CDF reduces to a sorting operation, which typically can be executed efficiently with algorithms such as `quicksort` [39]. Note that this CDF estimation technique results in a nonuniformly sampled estimate because the abscissa points are the observed waveform samples, which are not guaranteed to be uniformly spaced along the support of the distribution.

Figure 3.1a shows the CDFs of clean and compressed speech computed using the sorting technique previously described. Note the inflection points in the solid black curve at $\pm\tau$ where the nonlinearity takes effect. In this case, the clean CDF was computed from a large independent speech database. The BAN nonlinearity estimate inferred by matching these CDFs is shown in Fig. 3.1b on top of the true DRC function. It is apparent that BAN produces a reasonably accurate estimate of the nonlinearity in this case.

Of course, the relevant metric of BAN's overall effectiveness is its ability to improve speech recognition performance, which it does quite well. Figure 3.2a shows the same speech recognition



(a) Target CDF of clean speech and observed CDF of compressed speech with $\tau = 0.1 \approx \mathrm{P}_{70}$ and $R = 2$.

(b) Resulting inference of DRC nonlinearity obtained by matching the CDFs in Fig. 3.1a.

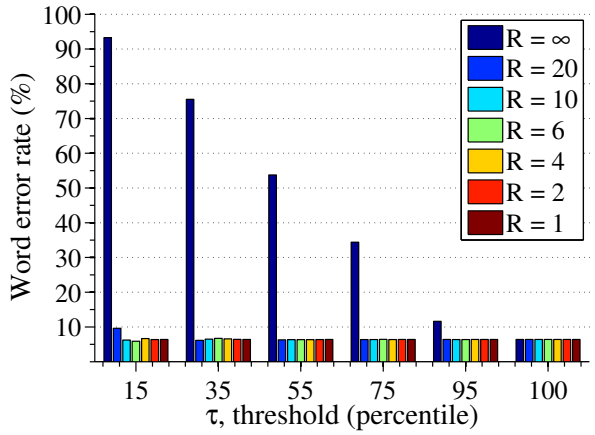
Figure 3.1: Illustration of inference of the DRC function using the Balchandran algorithm.

experiments presented in Fig. 2.9b but after processing both training and testing data with BAN, as depicted by Fig. 3.2b. Evidently, there is a blanket improvement in ASR performance, with BAN almost entirely mitigating the effects of DRC, except for the case of $R = \infty$, for which BAN should not be expected to work because the DRC function is noninvertible.

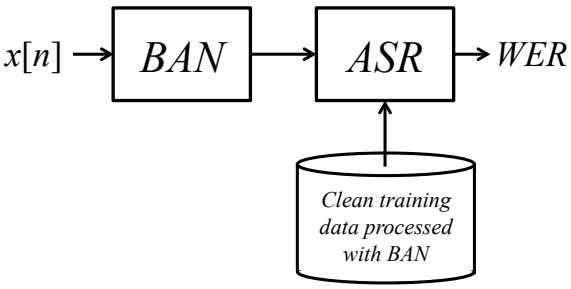
3.3 The Effects of Noise

Inversion of the DRC function unsurprisingly becomes more difficult with the addition of channel noise according to Fig. 2.5b. Two main problems arise. First, the presence of noise naturally decreases the accuracy of the nonlinearity estimate; secondly, even given oracle knowledge of the true nonlinearity, the perturbations due to noise are often amplified when passing through the inverse function (the opposite of *compression* is *expansion*). This amplification is illustrated by Fig. 3.3a.

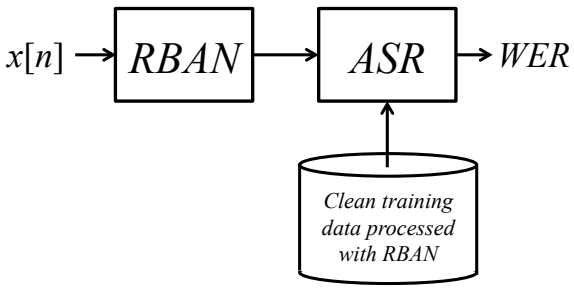
The performance of BAN when noise is added at 20-dB and 15-dB SNR is illustrated in Figures 3.5b and 3.6b, respectively. In comparison to Figures 2.10a and 2.10b (the results of which are repeated in Figures 3.5a and 3.6a, respectively), BAN nonetheless provides substantial improvements, primarily when $R > 20$ for the selection of DRC parameter values considered. Despite the inherent robustness of BAN, improvements to its baseline performance in noise are possible. Some approaches to this end are considered in the following sections.



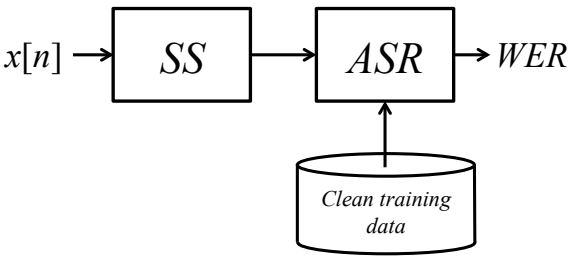
(a) ASR performance after processing training and testing speech with BAN as in Fig. 3.2b. Here, the speech contains no additive noise. BAN is ineffective for clipped signals, when $R = \infty$.



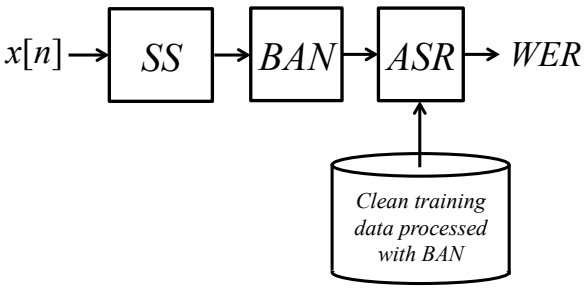
(b) Diagram of the experimental setup to measure BAN's effectiveness. Results with no additive noise are shown in Fig. 3.2b; results with additive noise are shown in Figures 3.5b and 3.6b.



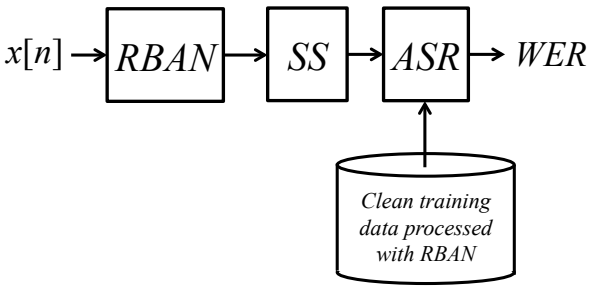
(c) Diagram of the experimental setup to measure Robust BAN's effectiveness. Results are shown in Figures 3.5c and 3.6c.



(d) Diagram of the experimental setup to measure the ASR improvement attributable to SS alone. Results are shown in Figures 3.5d and 3.6d.

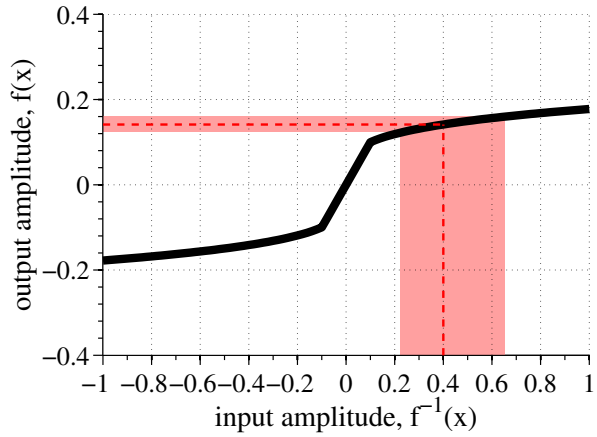


(e) Diagram of the experimental setup to measure BAN's effectiveness when combined with SS. Results are shown in Figures 3.5e and 3.6e.

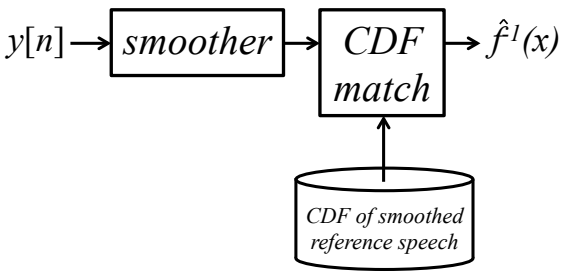


(f) Diagram of the experimental setup to measure Robust BAN's effectiveness when combined with SS. Results are shown in Figures 3.5f and 3.6f.

Figure 3.2: Baseline BAN results, an illustration of DRC estimation after noise removal, and diagrams of the experimental setups used to measure the efficacy of BAN-based algorithms.



(a) This plot demonstrates the amplification of channel noise added to the compressed signal when inverting DRC. Here, $R = 4$ and $\tau = 0.2$.



(b) A robust estimate, $\hat{f}^{-1}(x)$, of the inverse DRC function can be obtained by matching the distribution functions of smoothed speech.

Figure 3.3: Illustrations of the amplification of noise when inverting DRC (left) and an approach to obtaining a robust estimate of the DRC function (right). In the left plot, the height of the salmon-colored horizontal strip represents one standard deviation of AWGN at 15 dB around a compressed signal sample at $f(x) = 0.1414$, shown as the dashed horizontal line. As the sample passes through the inverse function, the power in the noise grows significantly, reflected by the increased width of the shaded vertical strip.

3.3.1 Circumventing the noise

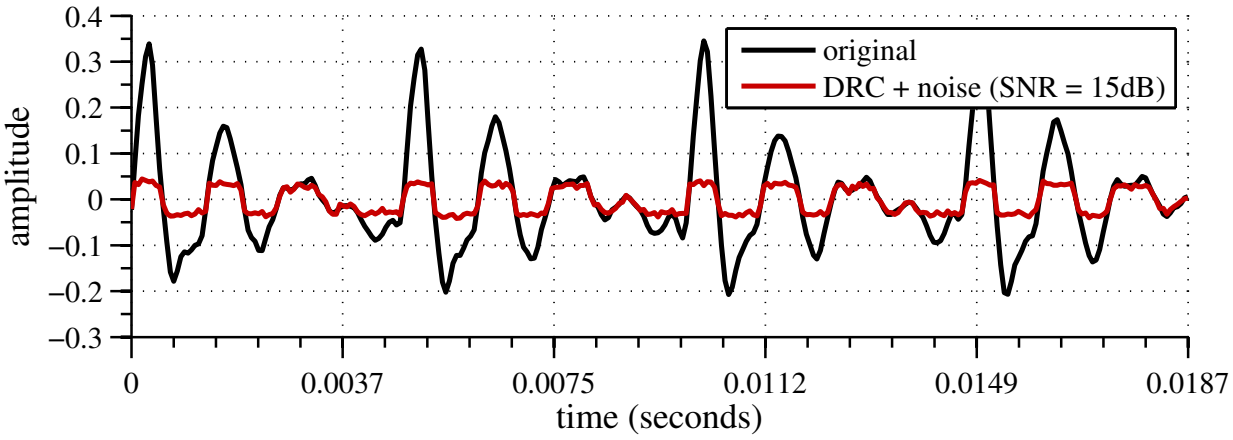
The diagram in Fig. 3.3b depicts a proposed system for obtaining a more accurate estimate of the inverse DRC function, which will be denoted $\hat{f}^{-1}(x)$. The smoothing block is intended to eliminate or attenuate perturbations due to additive noise. One possible smoothing mechanism is a simple moving average filter:

$$M_n(x) = \frac{1}{2N+1} \sum_{k=-N}^{N} x[n+k] \tag{3.7}$$

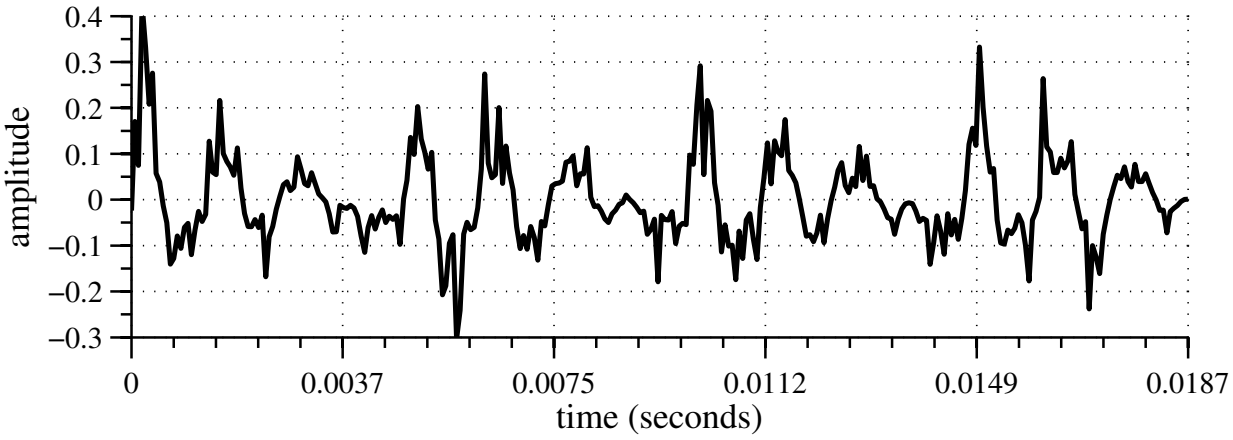
Given the estimate of the inverse DRC function, the original noisy signal can be decompressed as follows, where $y[n]$ is the noisy and compressed signal in accordance with Fig. 2.5b:

$$\hat{x}[n] = y[n] + \hat{f}^{-1}(M_n(y)) - M_n(y) \tag{3.8}$$

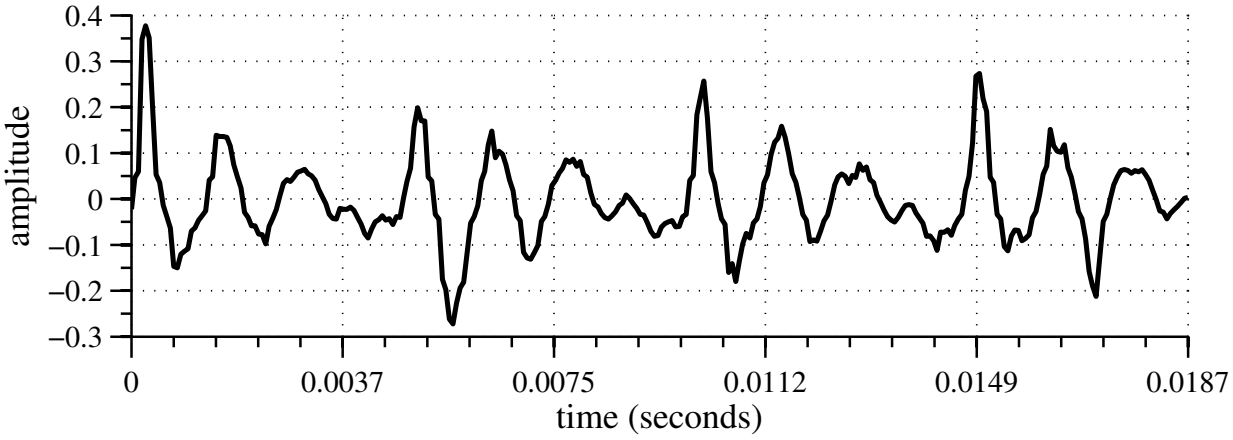
In Eq. 3.8, the $\hat{f}^{-1}(M_n(y)) - M_n(y)$ term is an offset that represents the amount by which the smoothed signal is vertically shifted when passing through $\hat{f}^{-1}$. In this way, the noisy samples of $y[n]$ are decompressed according to the underlying smoothed signal, M_n, and the inversion is



(a) Examples of a voiced speech waveform after DRC and noise addition.



(b) After inversion of the noisy DRC speech with BAN. Note the amplification of the noise.



(c) After inversion of the noisy DRC speech with Robust BAN.

Figure 3.4: An illustration of the effectiveness of Robust BAN in circumventing the perturbations due to noise when inverting the DRC nonlinearity. The waveforms in Figures 3.4b and 3.4c were obtained from the red curve in Fig. 3.4a.

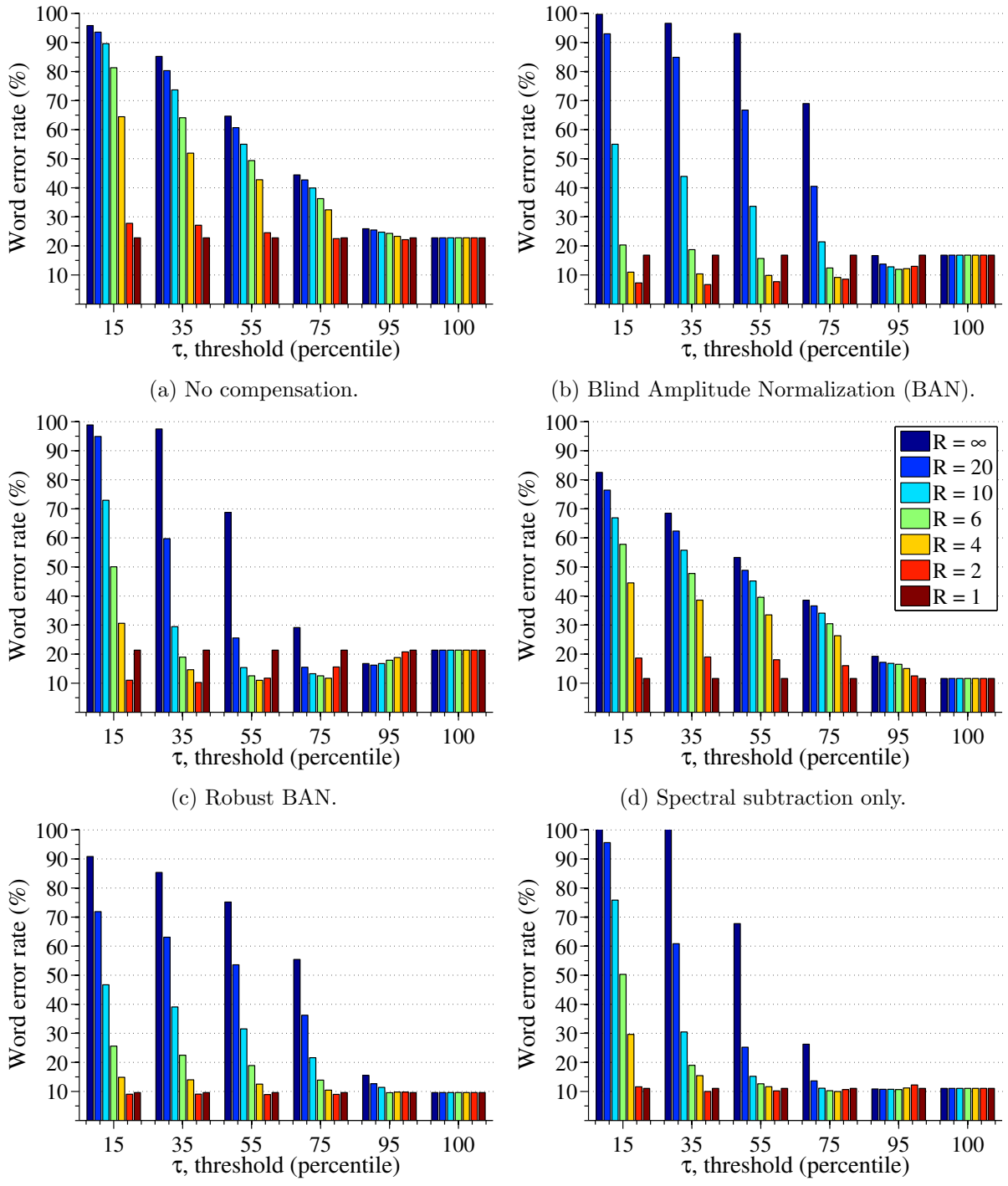
generally less sensitive to perturbations due to independent additive noise. This approach will be referred to as Robust BAN (RBAN). Figure 3.4 demonstrates RBAN in practice. The waveforms shown in Figures 3.4b and 3.4c were obtained from the red curve in Fig. 3.4a, using the original BAN and Robust BAN approaches, respectively. While both figures demonstrate successful signal decompression, Fig. 3.4b depicts a clear amplification of the noise, and Fig. 3.4c reflects a relatively cleaner signal. In this figure, as well as for all RBAN experimental results presented, the smoothing parameter $N = 2$, which equates to a moving average window of 5 samples.

3.3.2 Removing the noise

An alternative, and perhaps more prosaic, approach to improving the robustness of BAN is to simply attempt to remove the noise prior to inverting the DRC function. As exemplified by the overview in Section 2.1.5, a variety of practical noise reduction algorithms have been developed over the years. To demonstrate the efficacy of this approach, traditional *spectral subtraction* (SS) is used in the system in Fig. 3.2e to obtain the results of Figures 3.5e and 3.6e, for AWGN at 20-dB and 15-dB SNR, respectively. Similarly, the WER values obtained by using spectral subtraction alone, without BAN, are also shown in Figures 3.5d and 3.6d, again for AWGN at 20-dB and 15-dB SNR, respectively. It is clear that SS alone consistently lowers the WER across all conditions over the baseline of no compensation. Following SS with BAN provides further improvement, often substantial (e.g. the $R = 10$ cases), over SS.

3.3.3 Combining approaches to robustness

A natural extension to the noise removal and noise circumvention approaches is to combine them. Because Robust BAN inverts the nonlinearity in the presence of noise and spectral subtraction removes additive noise, a sensible ordering is to first apply Robust BAN to produce noisy but decompressed audio, then apply spectral subtraction. The experimental setup to test this combination is illustrated in Fig. 3.2f. Corresponding results for the case of AWGN injected at SNRs of 20 dB and 15 dB are shown in Figures 3.5f and 3.6f, respectively.



(a) No compensation.

(b) Blind Amplitude Normalization (BAN).

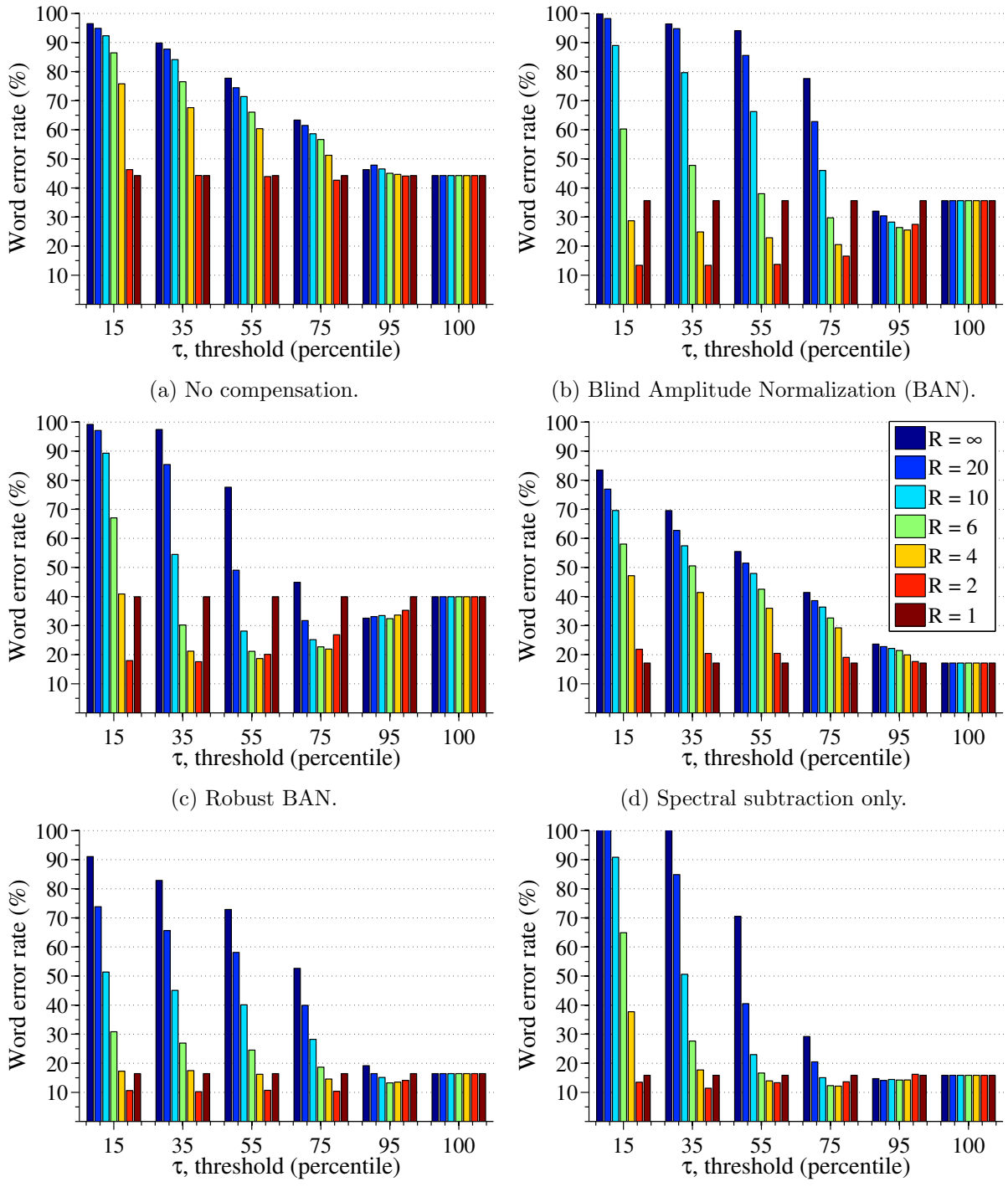
(c) Robust BAN.

(d) Spectral subtraction only.

(e) Spectral subtraction followed by BAN.

(f) Robust BAN followed by spectral subtraction.

Figure 3.5: Results of speech recognition experiments using variants of BAN on compressed speech containing AWGN at an SNR of 20 dB.



(a) No compensation.

(b) Blind Amplitude Normalization (BAN).

(c) Robust BAN.

(d) Spectral subtraction only.

(e) Spectral subtraction followed by BAN.

(f) Robust BAN followed by spectral subtraction.

Figure 3.6: Results of speech recognition experiments using variants of BAN on compressed speech containing AWGN at an SNR of 15 dB.

τ \ R	2	4	6	10	20	∞
15	*BAN*	*BAN*	*BAN*	*SS +BAN*	*SS +BAN*	*SS*
35	*BAN*	*BAN*	*BAN*	*RBAN*	*RBAN*	*SS*
55	*BAN*	*BAN*	*RBAN*	*RBAN +SS*	*RBAN +SS*	*SS*
75	*BAN*	*BAN*	*RBAN +SS*	*RBAN +SS*	*RBAN +SS*	*RBAN +SS*
95	*SS +BAN*	*SS +BAN*	*SS +BAN*	*RBAN +SS*	*RBAN +SS*	*RBAN +SS*
100	*SS +BAN*	*SS +BAN*	*SS +BAN*	*SS +BAN*	*SS +BAN*	*SS +BAN*

(a) With AWGN at 20-dB SNR.

τ \ R	2	4	6	10	20	∞
15	*SS +BAN*	*SS +BAN*	*SS +BAN*	*SS +BAN*	*SS +BAN*	*SS*
35	*SS +BAN*	*SS +BAN*	*SS +BAN*	*SS +BAN*	*SS*	*SS*
55	*SS +BAN*	*RBAN +SS*	*RBAN +SS*	*RBAN +SS*	*RBAN +SS*	*SS*
75	*SS +BAN*	*RBAN +SS*	*RBAN +SS*	*RBAN +SS*	*RBAN +SS*	*RBAN +SS*
95	*SS +BAN*	*SS +BAN*	*SS +BAN*	*RBAN +SS*	*RBAN +SS*	*RBAN +SS*
100	*RBAN +SS*	*RBAN +SS*	*RBAN +SS*	*RBAN +SS*	*RBAN +SS*	*RBAN +SS*

(b) With AWGN at 15-dB SNR.

Figure 3.7: These charts indicate the best performing algorithm for the particular pair of DRC parameters indicated.

3.4 Interpreting the Results

While the charts of Figures 3.5 and 3.6 are comprehensive, the sheer volume of data may be challenging to interpret usefully. Instead of studying the specific WER values produced by each algorithm in each condition, the tables in Fig. 3.7 identify the algorithm that gives rise to the lowest WER for a particular triplet of R, τ, and SNR.

In general, these tables suggest that BAN or BAN combined with spectral subtraction (SS+BAN) is most effective in the northwest region of the table, i.e. for low τ and low R, whereas Robust BAN or Robust BAN combined with spectral subtraction (RBAN+SS) is most effective in the southeast region of the table, i.e. for higher τ and higher R. Clearly, the utility of RBAN increases with decreasing SNR (as evidenced by its higher prevalence in Table 3.7b over Table 3.7a). Also note that all of the best-performing techniques in Table 3.7b incorporate SS, an intuitively reasonable outcome.

One final observation is that BAN and RBAN provide improvement beyond that of spectral subtraction in some cases when $R = \infty$. This result is somewhat counterintuitive but suggests that the perturbations due to noise allow BAN or RBAN to incidentally "declip" the signal by forcing the clipped signal samples to deviate from $\pm\tau$. The smoothing function used in RBAN may also play a role in this behavior.

3.5 Summary

This chapter considered approaches to compensation for dynamic range compression when the ratio parameter, R, is finite. As discussed in Section 2.2.1, the DRC function is invertible so long as $R < \infty$. Here, the blind amplitude normalization algorithm, first developed by Balchandran and Mammone, was introduced. Two variations of BAN, designed to be more robust to independent additive noise were developed. The first approach attempts to *circumvent* the noise by decompressing the signal based on a smoothed reference, the second approach attempts to remove the noise with the use of spectral subtraction prior to the application of standard BAN. The two approaches, while individually effective, are also complementary, and their combination was demonstrated to give rise to substantial improvements in speech recognition accuracy when decoding noisy, compressed speech. It was determined that the combination of spectral subtraction with BAN is most

useful for noisy speech compressed with relatively low threshold and ratio values. Robust BAN is more useful for noisy speech compressed with relatively high threshold and ratio values. In general, the utility of RBAN increases with decreasing SNR, as expected. The next chapter considers approaches to repairing clipped speech.

Chapter 4

Blind Amplitude Reconstruction (BAR)

4.1 Introduction

In this chapter, approaches to compensating for the effects of dynamic range compression when the ratio parameter equals ∞ are considered. Colloquially, DRC with $R = \infty$ is often referred to as *clipping*, owing to the appearance of the audio waveform as having been clipped on the top and bottom with a pair of scissors, as shown in Fig. 4.1. Mathematically, clipping is defined as follows:

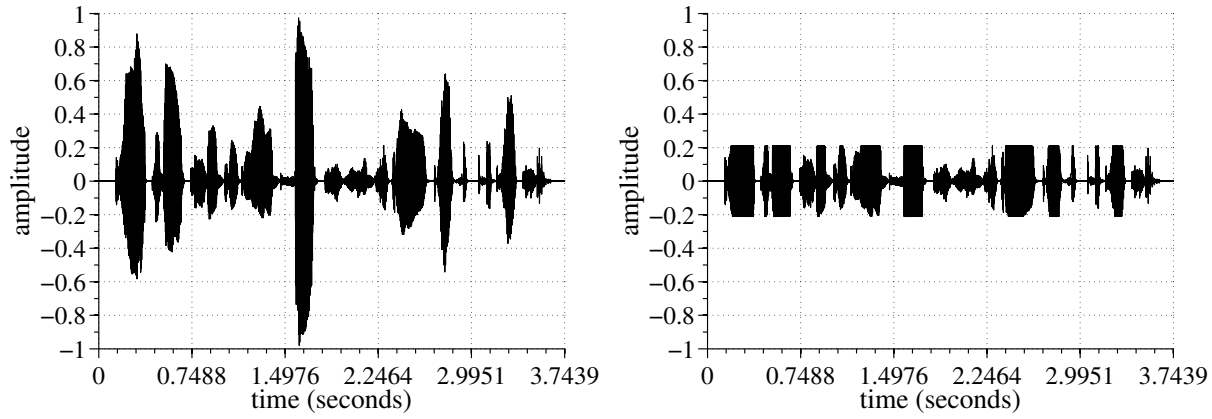
$$g(x[n]) = \begin{cases} \tau \cdot \operatorname{sgn} x[n] & \text{if } |x[n]| \geq \tau \\ x[n] & \text{if } |x[n]| < \tau \end{cases} \tag{4.1}$$

As noted in Section 2.2.1, clipping is a mathematically noninvertible transformation. Therefore, by definition, the blind amplitude normalization methods of Chapter 3 are not effective in compensating for clipping. This is most succinctly demonstrated by Fig. 3.2a, which shows that BAN provides no discernible improvement in ASR performance for $R = \infty$ (and in some cases slightly increases the WER).

Clipping is generally regarded as a form of undesirable distortion, and generally occurs either (1) during signal capture, as a result of exceeding the dynamic range limitations of an analog-to-digital (A/D) converter (e.g., by yelling loudly into a microphone and not properly adjusting the pre-amplifier gain) or (2) as a result of writing improperly normalized audio data to a file (e.g., MATLAB's popular `wavwrite` function requires values in the range $[-1, 1]$). In some cases, a signal is clipped on purpose, to achieve some desirable perceptual characteristic or maximally reduce the signal's dynamic range (e.g., for mastering music).

4.2 Existing approaches

Signal declipping has a rich history, with some approaches dating back to the 1980s. This section reviews some of the most popular and widely cited techniques, and considers their efficacy in the specific context of automatic speech recognition. Note that the quality of signal declipping algorithms is typically measured by perceptual experiments or mean-squared signal reconstruction error. The use of WER to measure the effectiveness of declipping constitutes novel research.



(a) Clean and unclipped speech waveform.

(b) Clipped waveform ($\tau = 0.21$).

Figure 4.1: Visualization of clipping. The clipped waveform (right) is obtained from the original (left) by clipping the positive and negative peaks. Clipping is the most extreme from of DRC and constitutes a mathematically noninvertible transformation.

4.2.1 Autoregressive modeling of speech for declipping

In 1986, Janssen *et al.* published a widely-cited speech declipping algorithm based on an autoregressive (all-pole) model of speech [31]. While this research has not found the Janssen declipping algorithm to be useful for speech recognition, its prevalence in the declipping literature necessitates a brief review. Before introducing Janssen's algorithm, however, an overview of relevant background information is presented. The following sections outline the source-filter model of speech production and linear predictive modeling of speech.

Source-filter model of speech production

The *source-filter* (SF) model of speech production is a way of characterizing the physical production of human speech that lends itself well to mathematical analysis [40]. The SF model dichotomizes all of human speech into two classes: *voiced* speech, i.e., speech with pitch (e.g., all vowel sounds, such as AA in "father"), and *unvoiced* speech, or speech with no pitch (e.g., fricatives, such as F in "for" and plosives like the P in "pop").

Given this dichotomy, the input or *source* of the speech is either an impulse train, in the case of voiced speech, or white noise, in the case of unvoiced speech. In either case, the source signal is then passed through a linear, shift-invariant (LSI) *filter* with an impulse response (IR) that reflects the target sound. For voiced sounds, the IR of the speech production filter will have peaks in its

Fourier transform magnitude centered around the formants that define that particular vowel (e.g., for the vowel AE as in "fast" the Fourier transform magnitude of the IR would look similar to either of the solid black curves in Fig. 2.3).

Figure 4.2 shows a diagram of the SF model. For each stationary segment of speech produced, the switch chooses the input signal as either $p[n]$, a pulse train for voiced speech, or $w[n]$, a noise source for unvoiced speech. The impulse train is defined as:

$$p[n] = \begin{cases} 1 & \text{for } n = kN; k \in \mathbb{Z} \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

The discrete-time Fourier transform (DTFT) of $p[n]$ is also an impulse train in the frequency domain:

$$P(e^{j\omega}) = \begin{cases} \frac{2\pi}{N} & \text{for } \omega = k\frac{2\pi}{N}; k \in \mathbb{Z} \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

The time-domain impulses are spaced according to the fundamental pitch period in samples; the frequency-domain impulses are spaced according to the fundamental frequency ($\frac{2\pi}{N}$) in radians per second. These quantities are inversely proportional to one another; as the pitch increases, the time-domain impulses move closer together and the frequency-domain impulses spread out. As the pitch decreases, the time-domain impulses spread out and the frequency-domain impulses move closer together.

Because each pulse of $p[n]$ represents a burst of air through the vocal cords, the value of N in the above equation reflects the fundamental period of the voice in samples (i.e., the inverse of the fundamental frequency, F0, or the *pitch*). E.g., if a male is speaking at a typical pitch of 100 Hz and the signal is sampled at 16 kHz, then $N = \frac{16{,}000}{100} = 160$. For voiced speech, every impulse input to the vocal tract filter elicits the impulse response, $h[n]$, and the observed waveform, $x[n]$, is approximated as a sum of scaled (by G) and delayed (by multiples of N) impulse responses:

$$\hat{x}[n] = G \sum_{l=-\infty}^{\infty} h[n - lN] \tag{4.4}$$

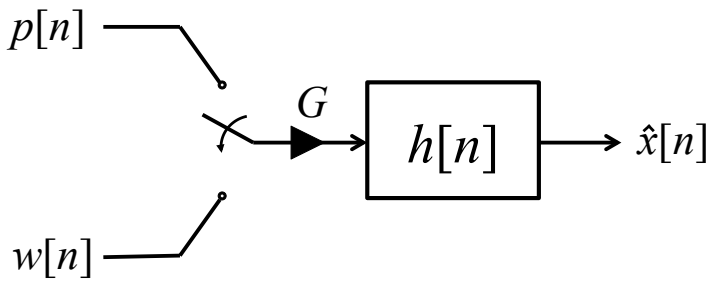


Figure 4.2: Diagram of the source-filter model of speech production.

Note that G and $h[n]$ vary with time,[1] but are approximately invariant within one frame of speech, on the order of 25 ms. Equation 4.4 is the result of the convolution of $h[n]$ and $p[n]$. Recalling that convolution in the time domain is equivalent to multiplication in the frequency domain, the DTFT of $\hat{x}[n]$ can simply be written as follows, where $H(e^{j\omega})$ is the frequency response of the filter, i.e., the DTFT of $h[n]$:

$$\hat{X}(e^{j\omega}) = G \cdot P(e^{j\omega}) \cdot H(e^{j\omega}) \tag{4.5}$$

Equation 4.5 explains the jagged appearance of the Fourier transform magnitudes of Fig. 2.3. Each sharp local peak is the result of an impulse being multiplied by the filter's underlying frequency response. Note that, as expected, the sharp peaks spread out slightly in going from Fig. 2.3a to 2.3b as the pitch increases. One of the purposes of the Mel-frequency filter bank stage of MFCC feature extraction was to eliminate this pitch information by smoothing out the local peaks.

Linear prediction

The defining characteristic of the SF model of speech production is the definition of the vocal tract filter, $h[n]$. While the source signal controls whether the speech is voiced or unvoiced, and in the former case, also controls the pitch of the speech, the vocal tract filter specifically defines the phoneme to be produced. To further facilitate mathematical analysis of the SF model, it is common practice to specify $h[n]$ to be an *all-pole filter*:

[1] Changes in G over time represent natural changes in volume as a person speaks, such as those resulting from emphasizing particular syllables of words; changes in $h[n]$ over time represent changes in the atomic speech sounds being made, the concatenation of which creates meaningful utterances.

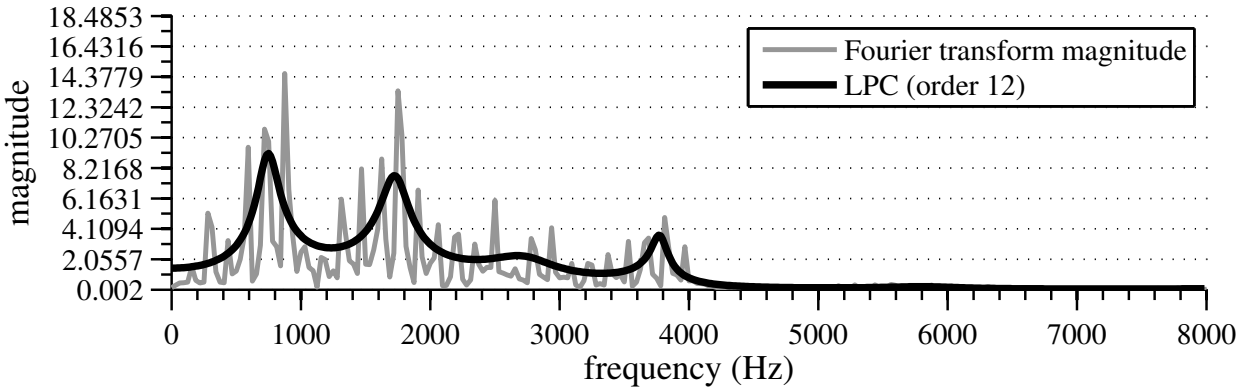


Figure 4.3: Spectrum of the vowel AE as in "fast" from Fig. 2.3b spoken at a pitch of approximately 150 Hz. The frequency response of the 12^{th}-order all-pole filter, whose coefficients were computed using Eq. 4.10 is shown. This could be used as the frequency response of a vocal tract filter in the SF model.

$$H(e^{j\omega}) = \frac{1}{1 - \sum_{k=1}^{K} \alpha_k z^{-k}} \tag{4.6}$$

With reference to Fig. 4.2, and defining the Fourier transform of the input signal (either $p[n]$ or $w[n]$) to be $E(e^{j\omega})$, the DTFT of the output is:

$$\hat{X}(e^{j\omega}) = \frac{G}{1 - \sum_{k=1}^{K} \alpha_k z^{-k}} E(e^{j\omega}) \tag{4.7}$$

This implies:

$$\hat{X}(e^{j\omega}) \left(1 - \sum_{k=1}^{K} \alpha_k z^{-k}\right) = G \cdot E(e^{j\omega}) \tag{4.8}$$

Distributing the $\hat{X}(e^{j\omega})$ term on the left side and taking the inverse z-transform yields:

$$\hat{x}[n] - \sum_{k=1}^{K} \alpha_k \hat{x}[n-k] = G \cdot e[n] \tag{4.9}$$

The summation term on the left hand side of Eq. 4.9 is a weighted sum of the past K samples of $\hat{x}[n]$. Thus, Eq. 4.9 states that in the SF model, if the vocal tract filter is defined to be an all-pole filter, the source signal is proportional to the error incurred by approximating the current

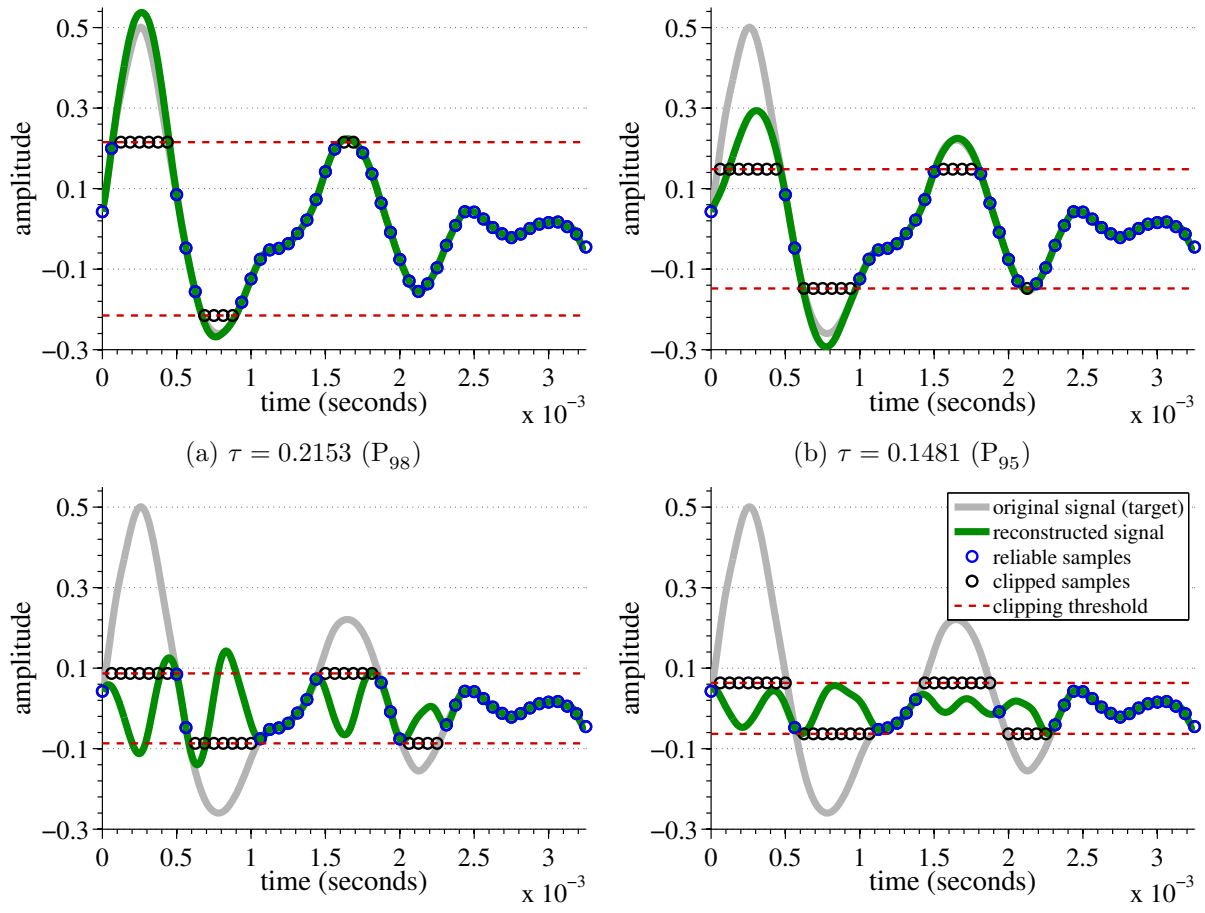
signal sample as a linear combination of the previous K signal samples. Each weight, α_k, in the linear combination is appropriately referred to as a *linear prediction* (LP) coefficient.

If both the error term, $e[n]$, and the value of K, i.e., the order of the all-pole filter, are relatively small, then linear prediction can be used as an efficient coding mechanism for speech. For each frame of speech, only the pitch, gain, voiced/unvoiced decision, and LP coefficients need to be transmitted−the speech can then be resynthesized at the receiver using the SF model. Such linear predictive coding (LPC) is widely used in modern telecommunications systems. Naturally, however, the key to using LPC in practice is the ability to reliably compute the LP coefficients from some observed speech. It can be shown that the minimum mean-squared error (MMSE) solution for the LP coefficients can be found by solving the following matrix equation for $\boldsymbol{\alpha}$ [30]:

$$\underbrace{\begin{bmatrix} \phi[0] & \phi[1] & \phi[2] & \cdots & \phi[K-1] \\ \phi[1] & \phi[0] & \phi[1] & \cdots & \phi[K-2] \\ \phi[2] & \phi[1] & \phi[0] & \cdots & \phi[K-3] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi[K-1] & \phi[K-2] & \phi[K-3] & \cdots & \phi[0] \end{bmatrix}}_{\boldsymbol{\Phi}} \underbrace{\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \vdots \\ \alpha_K \end{bmatrix}}_{\boldsymbol{\alpha}} = \underbrace{\begin{bmatrix} \phi[1] \\ \phi[2] \\ \phi[3] \\ \vdots \\ \phi[K] \end{bmatrix}}_{\boldsymbol{P}} \tag{4.10}$$

In Eq. 4.10, $\boldsymbol{\Phi}$ is an autocorrelation matrix whose entries consist of values of the autocorrelation of the speech signal. In practice, $\phi[n]$ is directly estimated from the observed signal, $x[n]$, typically in one of two ways: using the *autocorrelation method* (shown, resulting in a Toeplitz autocorrelation matrix) or the *covariance method* [30]. Because the autocorrelation method results in a Toeplitz matrix, and can thus utilize Levinson-Durbin recursion to solve for $\boldsymbol{\alpha}$, the autocorrelation method is more commonly used.

Figure 4.3 shows the magnitude spectrum of Eq. 4.6 for $K = 12$, where the LP coefficients were computed using the solution to Eq. 4.10. (The autocorrelation values were estimated from the observed audio waveform whose magnitude spectrum is shown in the figure.) By comparing Figs. 4.3 and 2.3, it can be seen that the LPC modeling and Mel filter bank frequency weighting of MFCC feature extraction both have the effect of smoothing the speech magnitude spectrum, eliminating pitch information, but retaining formant peaks. Indeed, LPC is used as the basis for another prevalent feature extraction algorithm.



(a) $\tau = 0.2153$ (P_{98})

(b) $\tau = 0.1481$ (P_{95})

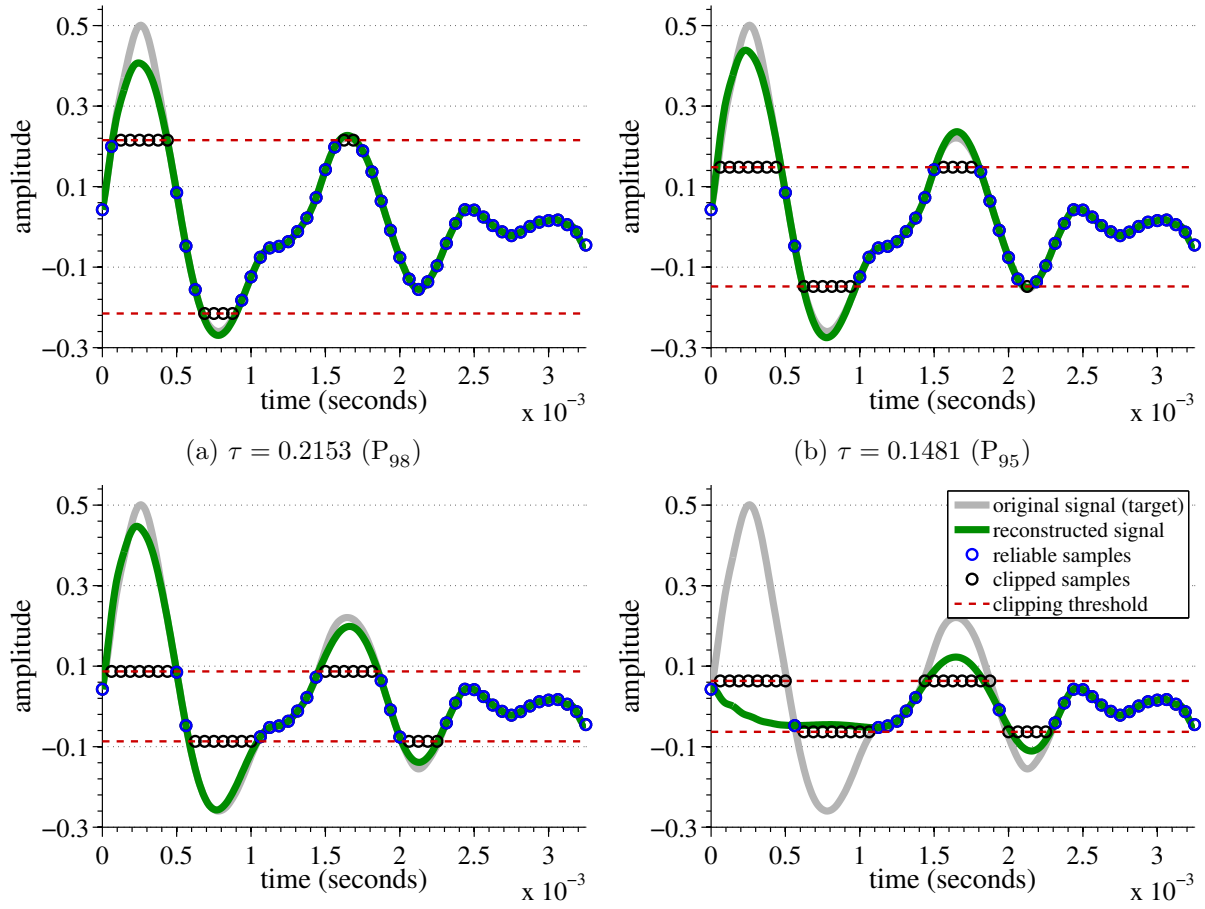
(c) $\tau = 0.0868$ (P_{85})

(d) $\tau = 0.0631$ (P_{75})

Figure 4.4: Examples of declipping outcomes using the Janssen-AR algorithm. The quality of the reconstructions rapidly deteriorates with decreasing τ.

The Janssen declipping algorithm

If the vector of LP coefficients, $\boldsymbol{\alpha}$, is known *a priori*, then the missing samples could be interpolated sequentially as a weighted linear combination of the previous K samples. Unfortunately, in practice, the value of the LP coefficients are not known and must be estimated from the observed data, which is incomplete due to clipping. The Janssen declipping algorithm (Janssen-AR) solves this problem with an iterative technique for simultaneously updating the LP coefficients and interpolating the missing signal samples. The technique is similar in nature to the well-known expectation maximization (EM) algorithm [41]. For a signal segment of length L, the vector of LP coefficients, $\boldsymbol{\alpha}$, and the vector of clipped (unknown) samples, $\boldsymbol{x}_c$, the following function is iteratively minimized:



(a) $\tau = 0.2153$ (P_{98})

(b) $\tau = 0.1481$ (P_{95})

(c) $\tau = 0.0868$ (P_{85})

(d) $\tau = 0.0631$ (P_{75})

Figure 4.5: Examples of declipping outcomes using the Selesnick-LS algorithm. The quality of the reconstructions is relatively stable in Figs. 4.5a−4.5c, and deteriorates suddenly as τ drops to P_{75} in Fig. 4.5d.

$$Q(\boldsymbol{\alpha}, \boldsymbol{x}_c) = \sum_{l=K}^{L-1} \left| \sum_{k=0}^{K} \alpha_k x[l-k] \right|^2 \tag{4.11}$$

The minimization of Eq. 4.11 follows a two-stop process. First, $\boldsymbol{x}_c$ is initialized to $\mathbf{0}$ and $Q(\boldsymbol{\alpha}, \boldsymbol{x}_c)$ is minimized with respect to $\boldsymbol{\alpha}$ with $\boldsymbol{x}_c$ fixed. Given the updated α_k values, the unknown samples in $\boldsymbol{x}_c$ are re-estimated by minimizing $Q(\boldsymbol{\alpha}, \boldsymbol{x}_c)$ with respect to $\boldsymbol{x}_c$ with $\boldsymbol{\alpha}$ fixed. This process repeats until convergence or a predetermined maximum number of iterations is reached.

Figure 4.4 depicts reconstructions of a portion of a speech signal that has been clipped at four thresholds: $0.2153, 0.1481, 0.0868$, and 0.0631 corresponding to the 98^{th}, 95^{th}, 85^{th}, and 75^{th} percentiles of the absolute value of the underlying waveform, respectively. Despite the well-motivated

development of the algorithm, these figures depict that Janssen-AR is only effective in accurately reconstructing the waveform for extremely high threshold values. The quality of the reconstructions rapidly deteriorate with decreasing τ. Shown in Fig. 4.13, Janssen-AR is similarly ineffective at decreasing the WER relative to the baseline error obtained by directly decoding unrepaired clipped speech.

4.2.2 Least squares declipping

Rather than using a model-based approach to reconstructing clipped regions of a signal, as with the Janssen-AR algorithm, a conceptually simpler approach to signal interpolation is possible based on the observation that signal peaks are generally parabolic in shape. Ivan Selesnick proposed an unpublished technique for signal declipping based on this observation.

Define $\boldsymbol{x}$ to be a column vector of length L which contains all the samples of a frame of clipped speech. Suppose there are R reliable samples contained in the vector $\boldsymbol{x}_r$ and $C = L - R$ clipped samples contained in the vector $\boldsymbol{x}_c$. Let $\boldsymbol{S}_r$ be the R x L matrix obtained from the L x L identity matrix by removing all rows corresponding to a clipped sample. Similarly, let $\boldsymbol{S}_c$ be the C x L matrix obtained from the L x L identity matrix by removing all rows corresponding to reliable samples. Finally, let $\boldsymbol{D}_i$ represent the i^{th} derivative, a linear operator. Note the following relationship is true [42]:

$$\boldsymbol{x} = \boldsymbol{S}_r^T \boldsymbol{x}_r + \boldsymbol{S}_c^T \boldsymbol{x}_c \tag{4.12}$$

The idea is to solve for $\boldsymbol{x}_c$ such that the third derivative of $\boldsymbol{x}$, i.e., $\boldsymbol{D}_3\boldsymbol{x}$, is minimized. By minimizing the third derivative, the reconstructed samples tend towards a parabolic shape, since the third derivative of a parabola is zero. Mathematically, the interpolation is obtained as follows:

$$\hat{\boldsymbol{x}}_c = \underset{\boldsymbol{x}_c}{\operatorname{argmin}} \, ||\boldsymbol{D}_3 \left(\boldsymbol{S}_r^T \boldsymbol{x}_r + \boldsymbol{S}_c^T \boldsymbol{x}_c \right) ||_2^2 \tag{4.13}$$

Recall that the least-squares solution to the standard matrix equation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{w}$ is [43]:

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \, ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}||_2^2 = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{y} \tag{4.14}$$

The solution of Eq. 4.14 can be used to solve Eq. 4.13 by making the following associations:

$$\hat{\boldsymbol{x}}_c = \underset{\boldsymbol{x}_c}{\operatorname{argmin}} || \underbrace{\boldsymbol{D}_3 \boldsymbol{S}_r^T \boldsymbol{x}_r}_{\boldsymbol{y}} + \underbrace{\boldsymbol{D}_3 \boldsymbol{S}_c^T}_{-\boldsymbol{A}} \boldsymbol{x}_c ||_2^2 \tag{4.15}$$

Equation 4.14 now directly yields the solution:

$$\hat{\boldsymbol{x}}_c = -(\boldsymbol{S}_c \boldsymbol{D}_3^T \boldsymbol{D}_3 \boldsymbol{S}_c^T)^{-1} \boldsymbol{S}_c \boldsymbol{D}_3^T \boldsymbol{D}_3 \boldsymbol{S}_r^T \boldsymbol{x}_r \tag{4.16}$$

Finally, the overall signal can be resynthesized using Eq. 4.12:

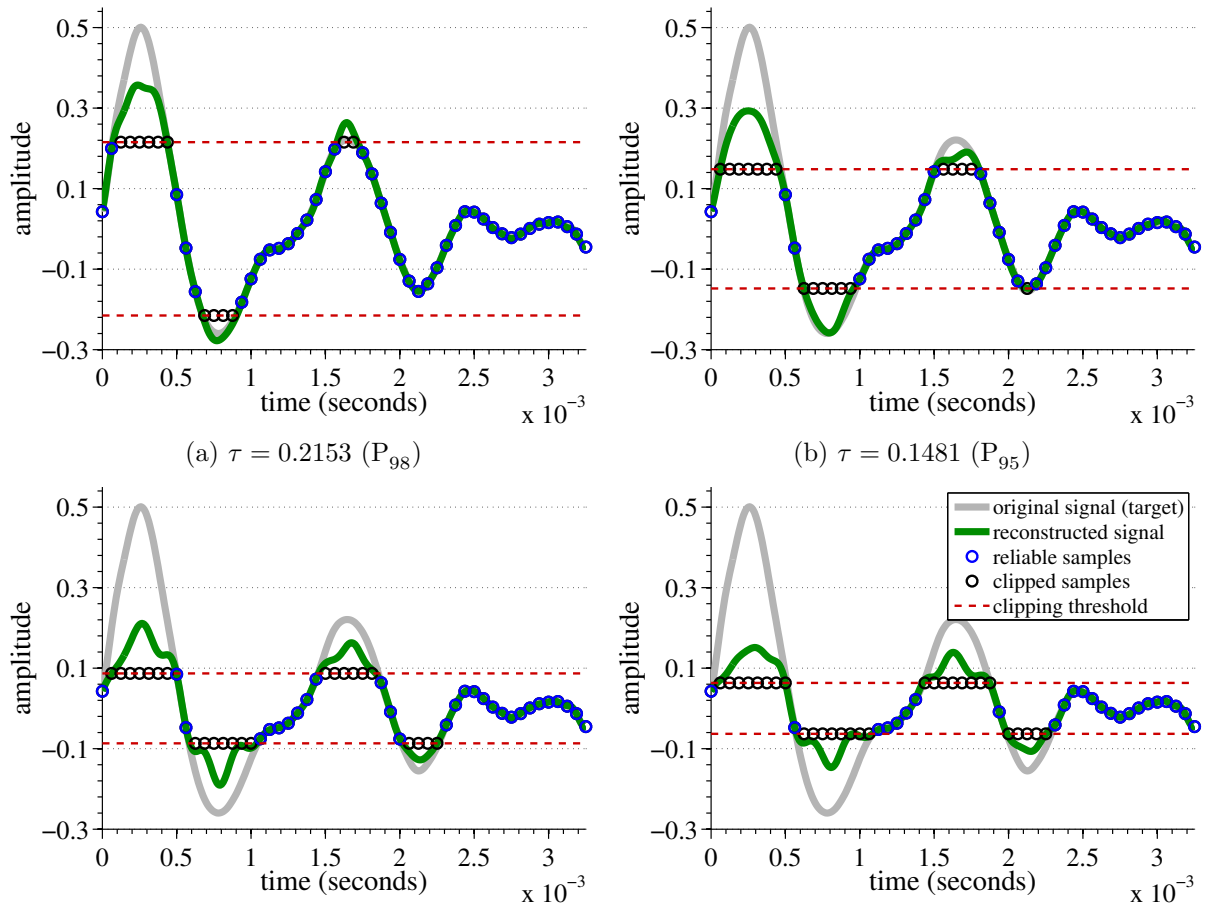
$$\boldsymbol{x} = \boldsymbol{S}_r^T \boldsymbol{x}_r + \boldsymbol{S}_c^T \hat{\boldsymbol{x}}_c \tag{4.17}$$

This declipping algorithm will be referred to as Selesnick-LS. Analogous to Fig. 4.4, Fig. 4.5 depicts reconstructions of the same portion of a speech signal that has been clipped at four thresholds: $0.2153, 0.1481, 0.0868, 0.0631$ corresponding to the 98^{th}, 95^{th}, 85^{th}, and 75^{th} percentiles of the absolute value of the underlying waveform, respectively. Upon comparison, it is evident that Selesnick-LS provides a more stable reconstruction over a larger range of threshold values than does Janssen-AR, though the reconstruction quality provided by Selesnick-LS similarly deteriorates below $\tau = \mathrm{P}_{85}$. As will be discussed later in the chapter, the increasing scarcity of reliable (unclipped) samples with decreasing τ necessitates the use of additional information (such as hard constraints) to guide a more accurate reconstruction.

4.2.3 Sparsity-based declipping

Sparsity-based declipping is another model-based approach to signal declipping that has gained significant popularity in recent years [35],[36]. In general, *sparsity*[2] in this context refers to the representation of a signal by a fixed, small number of basis vectors relative to the full dimensionality of the subspace. The number of basis vectors allowed in the representation is termed the *sparsity level*, S. For example, given a speech signal segment $\boldsymbol{x}$, a sparse representation of the signal in the subspace spanned by the column vectors of $\boldsymbol{\Psi}$ is found as follows:

[2] A *sparse* vector has only a small number of non-zero entries relative to its length.



(a) $\tau = 0.2153$ (P_{98})

(b) $\tau = 0.1481$ (P_{95})

(c) $\tau = 0.0868$ (P_{85})

(d) $\tau = 0.0631$ (P_{75})

Figure 4.6: Examples of declipping outcomes using the Kitic-IHT algorithm. Despite the undesirable high-frequency fluctuations and insufficient amplitude of the declipped signal segments, the quality of the reconstructions is stable over the range of thresholds considered.

$$\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \frac{1}{2}||\boldsymbol{x} - \boldsymbol{\Psi}\boldsymbol{\alpha}||^2 \text{ s.t. } ||\boldsymbol{\alpha}||_0 \leq S \tag{4.18}$$

In Eq. 4.18, sparsity is imposed by the constraint that the ℓ_0-norm of $\boldsymbol{\alpha}$ be less than or equal to S. The representation is sparse when S is much smaller than the number of columns in $\boldsymbol{\Psi}$. The solution to Eq. 4.18 can be obtained using the Iterative Hard Thresholding (IHT) algorithm [44]. Given a sparse representation, $\boldsymbol{\alpha}$, of a signal segment, $\boldsymbol{x}$, in terms of the basis vectors in $\boldsymbol{\Psi}$, the signal segment can be approximated as:

$$\boldsymbol{x} \approx \boldsymbol{\Psi}\boldsymbol{\alpha} \tag{4.19}$$

Therefore, if an accurate estimate of $\boldsymbol{\alpha}$ can be obtained from a clipped signal observation, then declipping can be achieved through the simple linear transformation of Eq. 4.19. The work by Kitic *et al.* [36], herein deemed the Kitic-IHT declipping algorithm, approximates $\boldsymbol{\alpha}$ from a clipped signal segment by solving the following modification of Eq. 4.18:

$$\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \frac{1}{2}||C(\boldsymbol{x} - \boldsymbol{\Psi}\boldsymbol{\alpha})||^2 \text{ s.t. } ||\boldsymbol{\alpha}||_0 \leq S \tag{4.20}$$

In Eq. 4.20, C is an operator that separates the observed signal into three subsets: (1) unclipped samples, (2) samples clipped at the positive threshold, $+\tau$, and (3) samples clipped at the negative threshold, $-\tau$. The C operator also imposes positive and negative "thresholding" on the clipped subsets. If $\boldsymbol{M}_r$, $\boldsymbol{M}_c^+$, and $\boldsymbol{M}_c^-$ are *masking matrices* representing each of the three subsets, respectively, and which transform all out-of-set samples of a vector to zero (i.e., resulting in vectors of the same length as the original signal segment vector $\boldsymbol{x}$, but with the out-of-set samples set to zero), then:

$$C(\boldsymbol{x}) = \boldsymbol{M}_r\boldsymbol{x} + ((\boldsymbol{M}_c^+\boldsymbol{x}))_+((\boldsymbol{M}_c^-\boldsymbol{x}))_- \tag{4.21}$$

The $((\cdot))_+$ and $((\cdot))_-$ are positive and negative thresholding operators, respectively, and operate element-wise on their arguments as follows:

$$((x))_\pm = \pm\frac{x \pm |x|}{2} \tag{4.22}$$

Note that if $x \leq 0$, then $((x))_+ = 0$; and inversely, if $x \geq 0$, then $((x))_- = 0$. Finally, $((x))_\pm = x$ otherwise. As more thoroughly outlined in [36], the implication of using C in Eq. 4.20 is that the minimization results in a vector $\hat{\boldsymbol{\alpha}}$ such that $\boldsymbol{\Psi}\hat{\boldsymbol{\alpha}}$ is (1) equal to $\boldsymbol{x}$ where the signal was originally unclipped, (2) greater than or equal to τ for which the original samples were clipped at $+\tau$, and (3) less than or equal to $-\tau$ where the original samples were clipped at $-\tau$. Mathematically,

$$\boldsymbol{M}_r \boldsymbol{\Psi} \hat{\boldsymbol{\alpha}} = \boldsymbol{M}_r \boldsymbol{x} \tag{4.23a}$$

$$\boldsymbol{M}_c^+ \boldsymbol{\Psi} \hat{\boldsymbol{\alpha}} \geq +\tau \boldsymbol{M}_c^+ \mathbf{1} \tag{4.23b}$$

$$\boldsymbol{M}_c^- \boldsymbol{\Psi} \hat{\boldsymbol{\alpha}} \leq -\tau \boldsymbol{M}_c^- \mathbf{1} \tag{4.23c}$$

Examples of declipped signal reconstructions are shown in Fig. 4.6. The utility of the added constraints from Eqs. 4.23 is particularly evident at a lower threshold value as in Fig. 4.6d, which depicts an acceptable reconstruction relative to those generated by Janssen-AR (Fig. 4.4d) and Selesnick-LS (Fig. 4.5d).

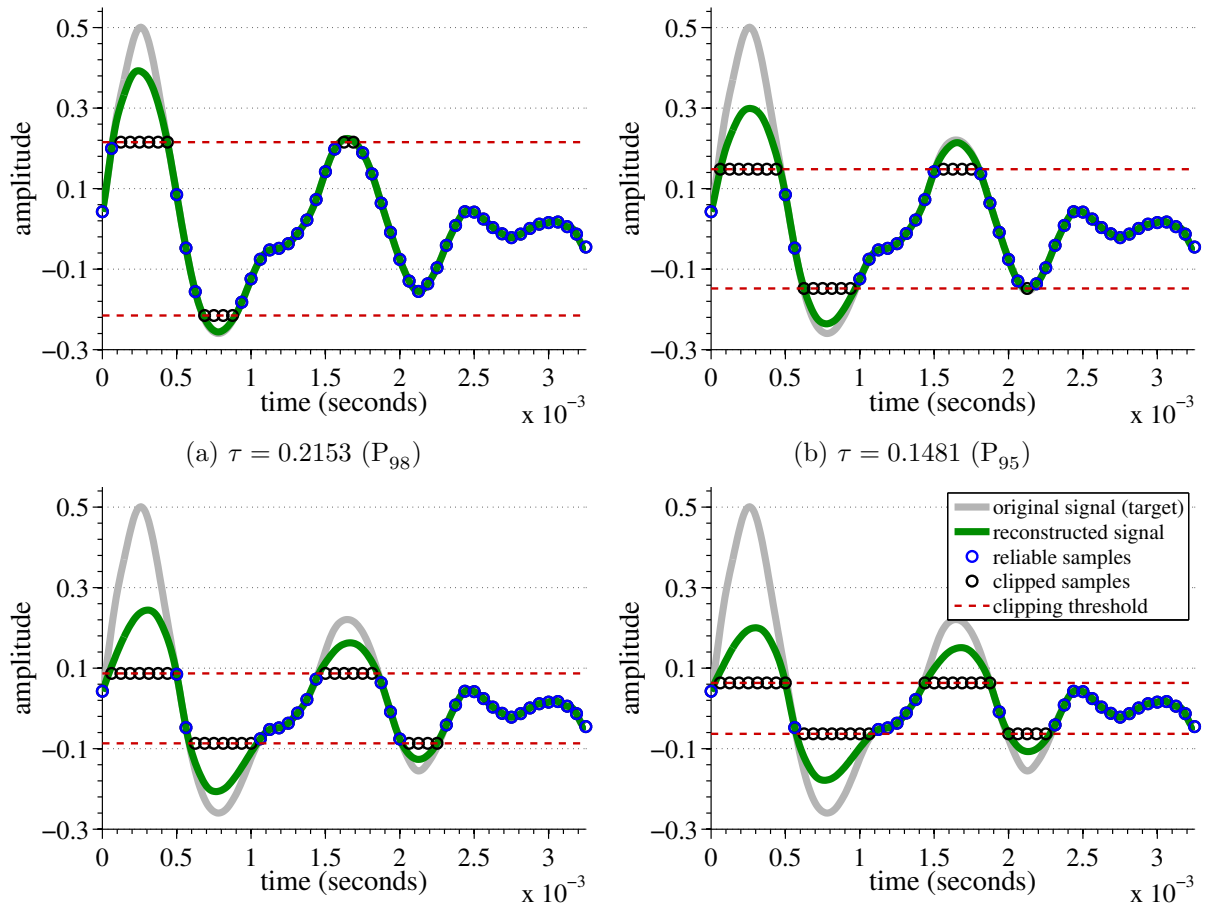
4.3 Constrained Blind Amplitude Reconstruction (CBAR)

This section introduces a novel declipping algorithm based on a combination of principles from the Selesnick-LS and Kitic-IHT methods. *Constrained Blind Amplitude Reconstruction*, or CBAR, solves the following nonlinear constrained optimization problem:

$$\begin{aligned} \underset{\boldsymbol{x}_c}{\text{minimize}} \quad & ||\boldsymbol{D}_2 \left(\boldsymbol{S}_r^T \boldsymbol{x}_r + \boldsymbol{S}_c^T \boldsymbol{x}_c \right)||_2^2 \\ \text{subject to} \quad & \boldsymbol{x}_c \circ \operatorname{sgn} \boldsymbol{S}_c \boldsymbol{x} \geq +\tau \mathbf{1} \end{aligned} \tag{4.24}$$

Equation 4.24 finds a sequence of samples $\boldsymbol{x}_c$ to replace the clipped samples, such that the reconstructed signal's second derivative is minimized within the search space allowed by the specified constraint on $\boldsymbol{x}_c$.

In the constraint term of Eq. 4.24, the $\circ$ represents the Hadamard (elementwise) product of two vectors or matrices. From the notation introduced in Section 4.2.2, recall that $\boldsymbol{S}_c$ is a C x L matrix obtained from the L x L identity matrix by removing each row corresponding to a reliable sample. The product $\boldsymbol{S}_c \boldsymbol{x}$, then, is a C x 1 vector containing the clipped samples from the original signal frame, $\boldsymbol{x}$, but with the reliable samples removed. Where the observed clipped sample is equal to $+\tau$, the underlying unclipped sample (the value of which is to be estimated) must be greater than or equal to $+\tau$. Inversely, where the observed clipped sample is equal to $-\tau$, the underlying



(a) $\tau = 0.2153$ (P_{98})

(b) $\tau = 0.1481$ (P_{95})

(c) $\tau = 0.0868$ (P_{85})

(d) $\tau = 0.0631$ (P_{75})

Figure 4.7: Examples of declipping outcomes using the CBAR algorithm. Though the amplitude of the declipped signal segments tends to undershoot the target amplitude, the reconstructions are smooth and their quality is stable over the range of thresholds considered.

unclipped sample must be less than or equal to $-\tau$. Requiring (each element of) the elementwise product of $\boldsymbol{x}_c$ and the sign of the corresponding observed clipped samples to be greater than τ incorporates this knowledge.

In the actual implementation of CBAR, the optimization problem is solved sequentially, on a frame-by-frame basis, for non-overlapping signal segments of at least 5 ms in length. To avoid irregular discontinuities in the recovered signal, it is ensured that the last sample of each frame (and consequently, the first sample of the next frame to be processed), is an unclipped sample, or else the frame length is incremented until it is.[3] Algorithm 1 summarizes CBAR in pseudocode.

In CBAR, the second derivative of the recovered signal is minimized, in contrast to the use of the

[3]Naturally, this approach reasonably assumes that unclipped samples are not spaced inordinately far apart.

Algorithm 1 Constrained Blind Amplitude Reconstruction

$N \leftarrow$ length of *observed_signal*
$minFrameSize \leftarrow 80$
$startPoint \leftarrow 0$
$outerWhileVar \leftarrow 1$
while $outerWhileVar$ **do** ▷ Step through frames until end of signal is reached.
    $doDeclipping \leftarrow 0$
    $endPoint \leftarrow startPoint + minFrameSize - 1$
    $innerWhileVar \leftarrow 1$
    while $innerWhileVar$ **do**
        if $endPoint > N - 1$ **then**
            $outerWhileVar \leftarrow 0$
            $endPoint \leftarrow N - 1$
            break
        end if
        if $\text{abs}(observed\_signal[endPoint]) = \tau$ **then**
            $endPoint \leftarrow endPoint + 1$ ▷ Increment last sample of frame if it's clipped.
        else
            $innerWhileVar \leftarrow 0$
        end if
    end while
    $j \leftarrow 0$
    $k \leftarrow 0$
    for $i = startPoint$ to $endPoint$ **do** ▷ Copy segment frame and check for clipping.
        $seg[k] \leftarrow observed\_signal[i]$
        $k \leftarrow k + 1$
        if $\text{abs}(observed\_signal[i]) = \tau$ **then**
            $signVect[j] \leftarrow \text{sgn}(observed\_signal[i])$
            $c_0[j] \leftarrow 1.25 \cdot \tau \cdot \text{sgn}(observed\_signal[i])$
            $doDeclipping \leftarrow 1$
            $j \leftarrow j + 1$
        end if
    end for

third derivative in Selesnick-LS. Minimization of the second derivative encourages a mathematically-smooth reconstruction; the parabolic shape encouraged by minimization of the third derivative is no longer necessary with the inclusion of the hard constraint. Nonetheless, there may be benefit to minimizing higher-order derivatives of the signal instead of the second derivative (or alternatively, finding a solution such that all derivatives of the signal up to order n are minimized or take on some predetermined characteristic value). This work is reserved for future research.

Figure 4.7 illustrates the efficacy of CBAR. In comparison to Figs. 4.4, 4.5, and 4.6 it yields the most consistent reconstruction and appears to be the most resilient to lower threshold values.

Algorithm 1 Constrained Blind Amplitude Reconstruction (continued)

```
    if doDeclipping then                                   ▷ Declip the segment.
        c ← solve Equation 4.24 given seg, signVect, τ; minimization initialized at c_0
    end if
    j ← 0
    for i = startPoint to endPoint do                      ▷ Copy repaired and unaltered samples.
        if abs(observed_signal[i]) = τ then
            repaired_signal[i] ← c[j]
            j ← j + 1
        else
            repaired_signal[i] ← observed_signal[i]
        end if
    end for
    startPoint ← endPoint                                  ▷ Shift to next frame.
    if startPoint + minFrameSize − 1 > N − 1 then          ▷ Check for end of signal.
        outerWhileVar = 0
    end if
end while
```

4.3.1 Nonlinear constrained optimization

The CBAR objective function defined by Eq. 4.24 requires the use of a sophisticated constrained optimization algorithm to solve. The current implementation of CBAR uses a line search [45]. A *line search* is an iterative algorithm that minimizes an objective function by computing a descent direction followed by a step size on each iteration. In the current implementation, the descent direction is computed using the quasi-Newton method, the benefit of which is that a full second-order derivative Hessian matrix does not need to be computed. The line search method is an *active-set* method because, on each iteration, the current "active" constraints (i.e., the points which lie on the constraint boundary) are maintained. Knowledge of these points allows one to determine the largest possible step size on each iteration.

4.4 Regularized Blind Amplitude Reconstruction (RBAR)

Both Kitic-IHT and CBAR, introduced in Sections 4.2.3 and 4.3 respectively, have run times much less than real time on a "typical" laptop computer due to the computational complexity of the underlying algorithms. In the case of Kitic-IHT, Eq. 4.20 must be solved on a frame-by-frame basis; similarly, in the case of CBAR, the constrained optimization of Eq. 4.24 is also solved

Algorithm 2 Regularized Blind Amplitude Reconstruction

$N \leftarrow$ length of *observed_signal*
$minFrameSize \leftarrow 800$
$frameStep \leftarrow 200$
$\lambda \leftarrow 0.05$
$numFrames \leftarrow 1 + \lceil \frac{N - minFrameSize}{frameStep} \rceil$
$newLength \leftarrow (numFrames - 1) \cdot frameStep + minFrameSize$
for $i = 0$ to $newLength - 1$ **do** ▷ Pad signal with zeros for integer number of frames.
    if $i < N$ **then**
        $padded\_signal[i] \leftarrow observed\_signal[i]$
        $repaired\_signal[i] \leftarrow 0$ ▷ Initialize repaired signal samples to zero.
    else
        $padded\_signal[i] \leftarrow 0$
        $repaired\_signal[i] \leftarrow 0$
    end if
end for
for $j = 0$ to $minFrameSize - 1$ **do** ▷ Compute a Hamming window.
    $hamming\_window[j] \leftarrow 0.54 - 0.46 \cos \frac{2\pi j}{minFrameSize - 1}$
end for
for $k = 0$ to $numFrames - 1$ **do** ▷ Cycle over frames.
    $startPoint \leftarrow k \cdot frameStep$
    $endPoint \leftarrow k \cdot frameStep + (minFrameSize - 1)$
    $frameLen \leftarrow endPoint - startPoint + 1$
    $n \leftarrow 0$
    $clippedFrame \leftarrow 0$
    $voicedFrame \leftarrow 0$
    for $m = startPoint$ to $endPoint$ **do** ▷ Extract signal frame and check for clipping.
        $seg[n] \leftarrow padded\_signal[m]$
        if abs$(seg[n]) = \tau$ **then**
            $clippedFrame \leftarrow 1$
        end if
        $n \leftarrow n + 1$
    end for
    $cepstral\_max \leftarrow$ **compute** peak of cepstrum as described in Section 4.4.4.
    if $cepstral\_max \geq 3$ **then**
        $voicedFrame \leftarrow 1$
    end if

frame-wise. Despite the efficacy of these techniques, a less computationally-intensive approach is desirable.

The innovation vis-à-vis CBAR is the realization that the Selesnick-LS technique (cf., Section 4.2.2) is ineffective due to the fact that its interpolating samples take on illegitimate values in the context of clipping (i.e., they may be less than $|\tau|$). CBAR addresses this problem by minimizing a similar objective function (the energy of the 2nd derivative instead of the 3rd) subject to the

Algorithm 2 Regularized Blind Amplitude Reconstruction (continued)

if *clippedFrame* AND *voicedFrame* **then** ▷ Declip if frame is clipped and voiced.
  $numPrependedSamples \leftarrow 0$
  $innerWhileVar \leftarrow 1$
  while $innerWhileVar$ **do** ▷ Prepend samples to frame if first sample is clipped.
    if abs($seg[0]$) $= \tau$ AND $startPoint > 0$ **then**
      $new\_seg[0] \leftarrow padded\_signal[startPoint - 1]$
      for $l = 0$ to $frameLen - 1$ **do**
        $new\_seg[l + 1] \leftarrow seg[l]$
      end for
      $numPrependedSamples \leftarrow numPrependedSamples + 1$
      $startPoint \leftarrow startPoint - 1$
      $frameLen \leftarrow frameLen + 1$
      $seg \leftarrow new\_seg$
    else
      $innerWhileVar \leftarrow 0$
    end if
  end while
  $numAppendedSamples \leftarrow 0$
  $innerWhileVar \leftarrow 1$
  while $innerWhileVar$ **do** ▷ Append samples to frame if last sample is clipped.
    if abs($seg[frameLen - 1]$) $= \tau$ AND $endPoint < N - 1$ **then**
      for $l = 0$ to $frameLen - 1$ **do**
        $new\_seg[l] \leftarrow seg[l]$
      end for
      $new\_seg[frameLen] \leftarrow padded\_signal[endPoint + 1]$
      $numPrependedSamples \leftarrow numPrependedSamples + 1$
      $endPoint \leftarrow endPoint + 1$
      $frameLen \leftarrow frameLen + 1$
      $seg \leftarrow new\_seg$
    else
      $innerWhileVar \leftarrow 0$
    end if
  end while
  $numClippedSamples \leftarrow 0$
  for $l = 0$ to $frameLen - 1$ **do**
    if $abs(seg[l]) = \tau$ **then** ▷ Compute fraction of clipped samples (ρ) in frame.
      $numClippedSamples \leftarrow numClippedSamples + 1$
    end if
  end for
  $\rho \leftarrow \frac{numClippedSamples}{frameLen}$
  if $\rho \leq 0.9$ **then**
    $\phi \leftarrow e^{2.481\rho}$
  else
    $\phi \leftarrow 271.7493\rho^{59.9519} + 8.8361$
  end if

Algorithm 2 Regularized Blind Amplitude Reconstruction (continued)

```
        c ← solve Equation 4.33 given seg, τ, φ, λ
        seg ← solve Equation 4.17
        origStartPoint ← startPoint + numPrependedSamples
        origEndPoint ← endPoint − numAppendedSamples
        for m = origStartPoint to origEndPoint do                    ▷ Overlap-add.
            repaired_signal[m] = repaired_signal[m]+hamming_window[m−origStartPoint]·
    seg[m − startPoint]
        end for
    else                    ▷ Frame is either not voiced, not clipped, or both; just copy data.
        for m = startPoint to endPoint do                            ▷ Overlap-add.
            repaired_signal[m] = repaired_signal[m] + hamming_window[m − startPoint] ·
    seg[m − startPoint]
        end for
    end if
end for
```

constraint that the interpolating samples have legitimate amplitude. Rather than enforcing this hard constraint, which leads to a computationally-intensive solution, one can employ *regularization* to "encourage" (though not guarantee) the interpolating values to fall in a legitimate range. The use of regularization allows for a closed-form solution.

This section describes the specific technical details of a frame-based declipping algorithm called *Regularized Blind Amplitude Reconstruction*, or RBAR, which performs comparably to Kitic-IHT and CBAR, but processes data at a much faster rate. Algorithm 2 summarizes RBAR in pseudocode.

4.4.1 Regularization

Regularization is often used to modify a least-squares problem statement such that the solution vector likely has more desirable characteristics. For example, rather than solving the standard least-squares problem described by Eq. 4.14, one may be interested in finding a solution with relatively low energy. This can be achieved by solving the following problem:

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}||_2^2 + \lambda||\boldsymbol{H}\boldsymbol{w}||_2^2 \tag{4.25}$$

If $\boldsymbol{H} = \boldsymbol{I}$, the energy of $\boldsymbol{w}$ is minimized in the original space, otherwise its energy is minimized in the space defined by the linear operator $\boldsymbol{H}$. Naturally, λ is an adjustable real-valued parameter

that quantifies the relative importance of the regularizing term in the minimization. Note that any number of linear regularizing terms can be added to the objective function and a closed-form solution is still possible. The form of regularization most relevant to this discussion is as follows:

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}}\, J(\boldsymbol{w}) \tag{4.26a}$$

$$= \underset{\boldsymbol{w}}{\operatorname{argmin}}\, ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}||_2^2 + \lambda_0||\boldsymbol{t}_0 - \boldsymbol{H}_0\boldsymbol{w}||_2^2 + \lambda_1||\boldsymbol{t}_1 - \boldsymbol{H}_1\boldsymbol{w}||_2^2 \tag{4.26b}$$

In Eq. 4.26, $\boldsymbol{t}_0$ and $\boldsymbol{t}_1$ are target vectors used to guide the solution in the spaces defined by $\boldsymbol{H}_0$ and $\boldsymbol{H}_1$, respectively. For declipping, these terms will be used to guide the solution toward values greater than $+\tau$, where the signal is clipped at $+\tau$, and less than $-\tau$, where the signal is clipped at $-\tau$. The solution vector, $\hat{\boldsymbol{w}}$, is obtained by finding the matrix derivative of Eq. 4.26 and setting it equal to $\mathbf{0}$, as follows.

$$\frac{\partial}{\partial \boldsymbol{w}} J(\boldsymbol{w}) = 2\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{w} - \boldsymbol{y}) + 2\lambda_0\boldsymbol{H}_0^T(\boldsymbol{H}_0\boldsymbol{w} - \boldsymbol{t}_0) + 2\lambda_1\boldsymbol{H}_1^T(\boldsymbol{H}_1\boldsymbol{w} - \boldsymbol{t}_1) \tag{4.27}$$

Setting Eq. 4.27 equal to $\mathbf{0}$, as noted, yields the following solution:

$$\hat{\boldsymbol{w}} = (\boldsymbol{A}^T\boldsymbol{A} + \lambda_0\boldsymbol{H}_0^T\boldsymbol{H}_0 + \lambda_1\boldsymbol{H}_1^T\boldsymbol{H}_1)^{-1}(\boldsymbol{A}^T\boldsymbol{y} + \lambda_0\boldsymbol{H}_0^T\boldsymbol{t}_0 + \lambda_1\boldsymbol{H}_1^T\boldsymbol{t}_1) \tag{4.28}$$

4.4.2 Applying regularization to declipping

To understand the application of regularization to declipping, it is useful to consider conceptual parallels with CBAR from Section 4.3. The CBAR algorithm declips a signal by finding an interpolating sequence of samples such that the second derivative of the reconstructed signal is minimized. Fundamentally, the CBAR objective function is the same as the Selesnick-LS objective function, except with the second derivative operator, $\boldsymbol{D}_2$, replacing the third derivative operator, $\boldsymbol{D}_3$:

$$\hat{\boldsymbol{x}}_c = \underset{\boldsymbol{x}_c}{\operatorname{argmin}}\, ||\boldsymbol{D}_2\left(\boldsymbol{S}_r^T\boldsymbol{x}_r + \boldsymbol{S}_c^T\boldsymbol{x}_c\right)||_2^2 \tag{4.29}$$

From Section 4.2.2, recall that the $\boldsymbol{S}_r$ and $\boldsymbol{S}_c$ matrices serve to isolate the reliable samples and

the clipped samples of the observed signal, respectively. To easily employ regularization in the form of Eq. 4.28, it is useful to further define two additional matrices that separate the *clipped* samples, i.e., $\boldsymbol{x}_c = \boldsymbol{S}_c\boldsymbol{x}$, into two sets: $\boldsymbol{x}_c^+$, containing samples clipped at the positive threshold, and $\boldsymbol{x}_c^-$, containing samples clipped at the negative threshold.

As before, define $\boldsymbol{x}$ to be a column vector of length L which contains all the samples of a frame of clipped speech. Suppose there are R reliable samples contained in the vector $\boldsymbol{x}_r$ and $C = L - R$ clipped samples contained in the vector $\boldsymbol{x}_c$. Further suppose there are C_p positively-clipped samples (i.e., samples clipped at $+\tau$) and C_n negatively-clipped samples (i.e., samples clipped at $-\tau$). Define $\boldsymbol{S}_c^+$ to be the C_p x C matrix obtained from the C x C identity matrix by removing all rows corresponding to a negatively-clipped sample. Similarly, let $\boldsymbol{S}_c^-$ be the C_n x C matrix obtained from the C x C identity matrix by removing all rows corresponding to positively-clipped samples. Note the following relationship is true:

$$\boldsymbol{x}_c = (\boldsymbol{S}_c^+)^T\boldsymbol{x}_c^+ + (\boldsymbol{S}_c^-)^T\boldsymbol{x}_c^- \tag{4.30}$$

Given the signal decomposition of Eq. 4.30, the regularized objective function for declipping can be framed as follows:

$$\hat{\boldsymbol{x}}_c = \underset{\boldsymbol{x}_c}{\operatorname{argmin}} \, ||\boldsymbol{D}_2\boldsymbol{S}_r^T\boldsymbol{x}_r + \boldsymbol{D}_2\boldsymbol{S}_c^T\boldsymbol{x}_c||_2^2 + \lambda||\boldsymbol{t}_0 - \boldsymbol{S}_c^+\boldsymbol{x}_c||_2^2 + \lambda||\boldsymbol{t}_1 - \boldsymbol{S}_c^-\boldsymbol{x}_c||_2^2 \tag{4.31}$$

The first term in Eq. 4.31 represents the energy of the 2$^{\text{nd}}$ derivative of the reconstructed signal; the second and third terms represent the squared-error between target vectors, $\boldsymbol{t}_0$ and $\boldsymbol{t}_1$, and the positively-clipped and negatively-clipped sample sets, respectively. Equation 4.28 can be used to solve Eq. 4.31 by making the following associations to Eq. 4.26, and noting that $\boldsymbol{x}_c$ replaces $\boldsymbol{w}$.

$$\hat{\boldsymbol{x}}_c = \underset{\boldsymbol{x}_c}{\operatorname{argmin}} \, ||\underbrace{\boldsymbol{D}_2\boldsymbol{S}_r^T\boldsymbol{x}_r}_{\boldsymbol{y}} + \underbrace{\boldsymbol{D}_2\boldsymbol{S}_c^T}_{-\boldsymbol{A}}\boldsymbol{x}_c||_2^2 + \lambda||\boldsymbol{t}_0 - \underbrace{\boldsymbol{S}_c^+}_{\boldsymbol{H}_0}\boldsymbol{x}_c||_2^2 + \lambda||\boldsymbol{t}_1 - \underbrace{\boldsymbol{S}_c^-}_{\boldsymbol{H}_1}\boldsymbol{x}_c||_2^2 \tag{4.32}$$

Therefore,

$$\hat{\boldsymbol{x}}_c = -(\boldsymbol{S}_c\boldsymbol{D}_2^T\boldsymbol{D}_2\boldsymbol{S}_c^T + \lambda(\boldsymbol{S}_c^+)^T\boldsymbol{S}_c^+ + \lambda(\boldsymbol{S}_c^-)^T\boldsymbol{S}_c^-)^{-1} \\ (\boldsymbol{S}_c\boldsymbol{D}_2^T\boldsymbol{D}_2\boldsymbol{S}_r^T\boldsymbol{x}_r - \lambda(\boldsymbol{S}_c^+)^T\boldsymbol{t}_0 - \lambda(\boldsymbol{S}_c^-)^T\boldsymbol{t}_1) \tag{4.33}$$

The overall signal frame is then resynthesized using Eq. 4.17.

4.4.3 Amplitude prediction

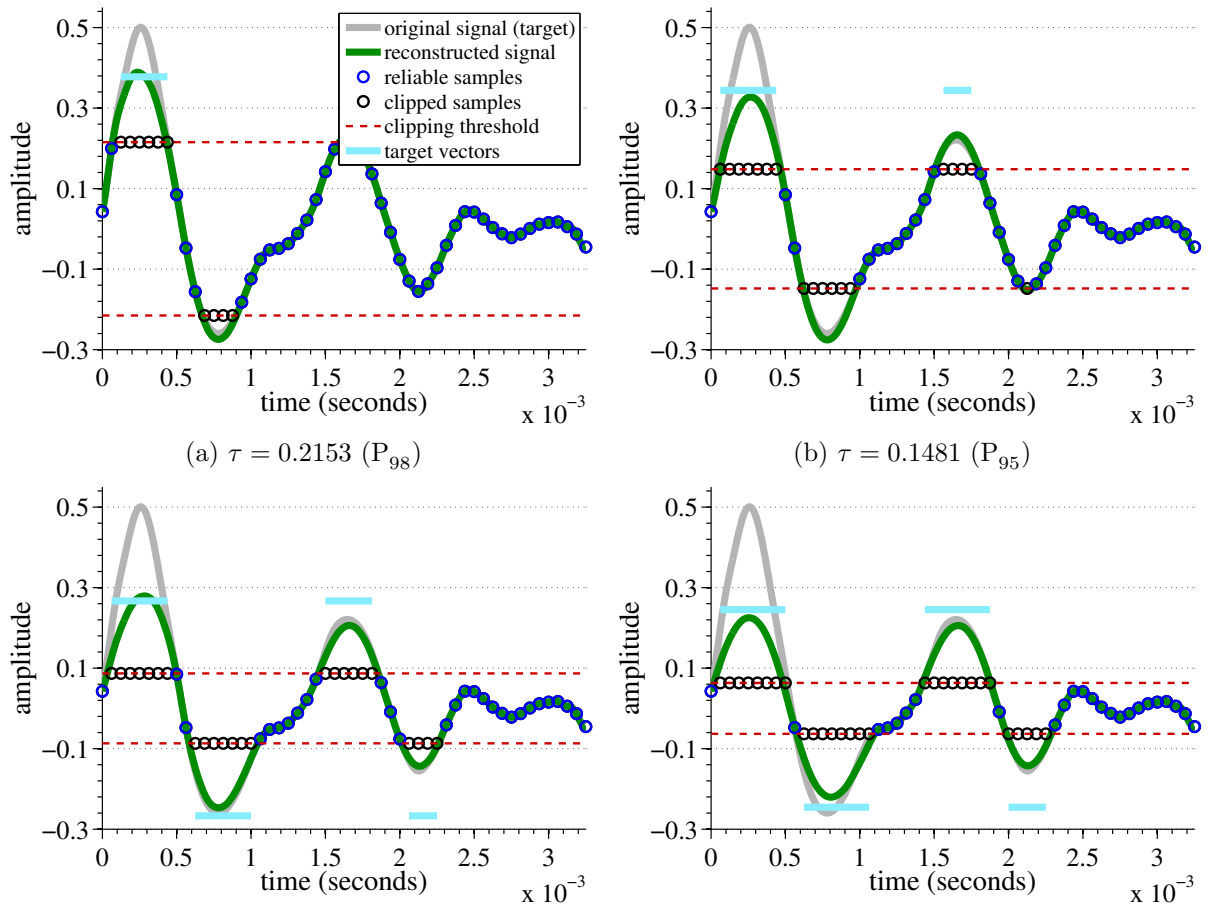
In order to compute $\hat{\boldsymbol{x}}_c$ in Eq. 4.33, values must be assigned to the target vectors, $\boldsymbol{t}_0$ and $\boldsymbol{t}_1$. In the ideal case, $\boldsymbol{t}_0$ and $\boldsymbol{t}_1$ would contain the exact sample values of the original signal in its clipped regions. Of course, if this information were available, a *blind* declipping algorithm would be unnecessary. Instead, as mentioned in Section 4.4, the target vectors will be assigned such that the interpolation tends toward a legitimate solution in which the interpolating samples fall above τ in positively-clipped segments, and below $-\tau$ in negatively-clipped segments.

Because the first term of Eq. 4.31 enforces a smooth reconstruction, it is reasonable to dynamically assign the target vectors to a constant value equal to some robust measure of the peak amplitude in a given clipped frame. That is, $\boldsymbol{t}_0$ and $\boldsymbol{t}_1$ should vary on a frame-by-frame basis as a function of some statistic of the clipped frame. Generally speaking, the peak amplitude of a signal segment is positively correlated with the energy of the segment, and further, for some fixed τ, the fraction of clipped samples of a segment of speech is positively correlated with the pre-clipping energy of the segment. Therefore, the fraction of clipped samples can be used to predict the peak amplitude of a signal segment as a function of τ.

Figure 4.9a shows a scatter plot of the ratio of the 95^{th} percentile of a frame of speech before clipping, to the clipping threshold, τ (i.e., $\frac{\text{P}_{95}}{\tau}$) as a function of the fraction of clipped samples in each frame of speech. The points on the scatter plot were obtained by artificially clipping a clean database of speech (independent of the testing data) at five different thresholds:[4] P_{15}, P_{35}, P_{55}, P_{75}, and P_{95}, and using the pre-clipped clean data to determine the ratio, $\frac{\text{P}_{95}}{\tau}$.

Nonlinear least-squares can be used to fit a regression function to the data in Fig. 4.9a. The

[4]The clipping thresholds for setting τ and artificially clipping the speech are determined from the percentiles over an entire utterance; the threshold used in the ratio, $\frac{\text{P}_{95}}{\tau}$, is associated with an individual short-duration frame.



(a) $\tau = 0.2153$ (P_{98})

(b) $\tau = 0.1481$ (P_{95})

(c) $\tau = 0.0868$ (P_{85})

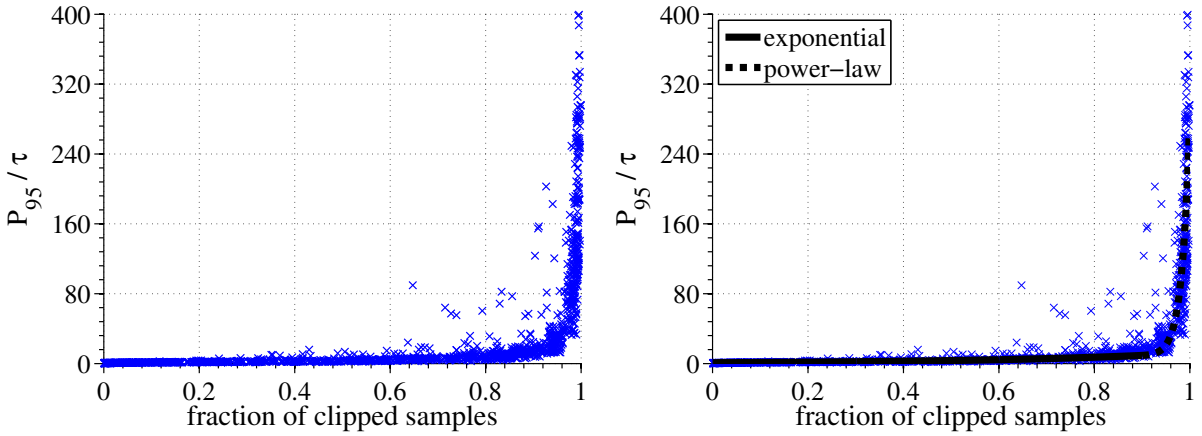
(d) $\tau = 0.0631$ (P_{75})

Figure 4.8: Examples of declipping outcomes using the RBAR algorithm. The target vectors, defined by Eq. 4.35, are plotted time-aligned to the corresponding clipped samples. The reconstructions are similar to CBAR (Fig. 4.7), despite being the result of a closed-form, computationally-efficient solution.

optimal fit was found to be a piecewise combination of exponential and power-law functions. Denoting the ratio as $\phi = \frac{P_{95}}{\tau}$ and the fraction of clipped samples as ρ, the resulting regression function is given by:

$$\phi(\rho) = \begin{cases} e^{2.481\rho} & \text{for } \rho \leq 0.9 \\ 271.7493\rho^{59.9519} + 8.8361 & \text{for } \rho > 0.9 \end{cases} \tag{4.34}$$

Given the value of ρ for an incoming frame of clipped speech (which can be computed trivially with the knowledge of which samples are clipped), the target vectors are then set as follows:



(a) Scatter plot.

(b) Scatter plot with regression.

Figure 4.9: Scatter plots showing the relationship between the ratio $\frac{P_{95}}{\tau}$ and the fraction of clipped samples in a frame of clipped speech. The right plot shows a piecewise least-squares fit to the data, which is used to assign the target vectors in Eq. 4.33.

$$\boldsymbol{t}_0 = \phi(\rho)\tau\mathbf{1};\ \boldsymbol{t}_1 = -\phi(\rho)\tau\mathbf{1} \tag{4.35}$$

Note the the value of $\phi(\rho)$ never falls below 1 in accordance with the fact that the interpolating samples should always be greater than $|\tau|$ in the absolute sense. Further, note that $\boldsymbol{t}_0$ is a C_p x 1 vector and $\boldsymbol{t}_1$ is a C_n x 1 vector.

4.4.4 Voicing detection

It was empirically observed that Eq. 4.33 destroys the spectral characteristics of unvoiced speech as a result of smoothing. Despite the incorporation of the target vector framework, the interpolating samples tend to fall between $-\tau$ and τ for signal segments with extremely high frequency fluctuations. This is illustrated by Fig. 4.10. The loss of fricatives significantly decreases the intelligibility of an utterance, and causes a correspondingly large increase in WER when the processed speech is decoded using ASR. For these reasons, and because clipping has a somewhat negligible effect on unvoiced speech, the RBAR algorithm does not repair clipped speech segments likely to be unvoiced. This, of course, requires some form of voicing detection.

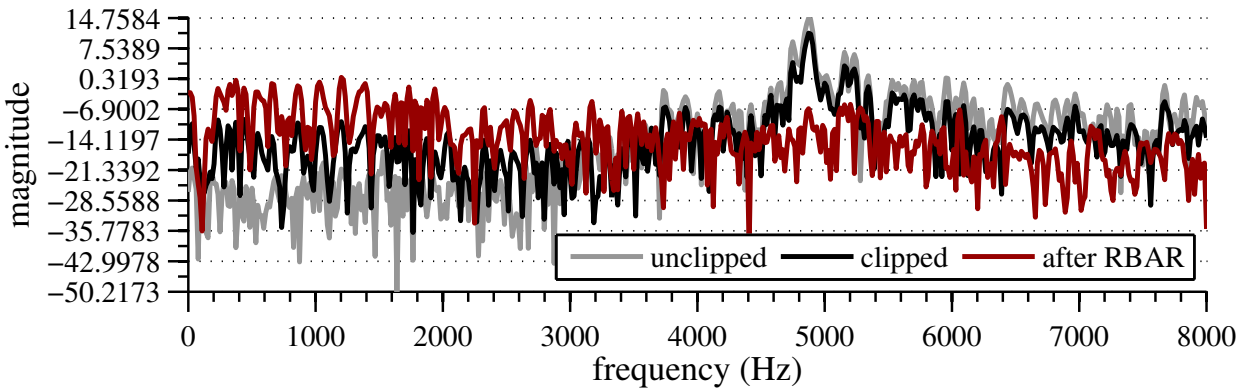


Figure 4.10: Spectrum of the fricative S as in "say," before clipping, after clipping at $\tau = \mathrm{P}_{75} = 0.0288$, and after declipping using RBAR. The figure illustrates that the use of RBAR further increases the spectral error with respect to the original spectrum, beyond that of the unrepaired, clipped spectrum. In this situation, RBAR yields an undesirably-smooth interpolation in the time domain, reflected by the attenuation of high-frequency components.

Cepstral analysis

Pitch−and consequently, voicing−detection can be achieved through the use of cepstral analysis [46]. As briefly introduced in Section 2.1.2, the form of cepstrum used in this work is defined as follows:

$$C[n] = \mathrm{DCT}(\log|X(e^{j\omega})|) \tag{4.36}$$

In words, the *cepstrum* is defined here as the discrete cosine transform (DCT) of the logarithm of the magnitude of the Fourier transform, $|X(e^{j\omega})|$, of the underlying signal. Many sources define the cepstrum using the inverse Fourier transform in place of the DCT; because the cosine transform is real-valued, it was chosen to be used here for convenience. The cepstrum is colloquially referred to as a "spectrum of a spectrum" because it concisely reflects periodic content in the magnitude spectrum of the signal. Recalling again from Section 2.1.2 (and as illustrated by Fig. 2.3), pitch manifests itself as high-frequency spikes in the magnitude spectrum, which look like the teeth of a comb. These rapid periodic variations in the magnitude spectrum are transformed into a single peak in the cepstrum located at the pitch period. Thus, to determine the pitch, one can search for a cepstral peak over a range of time values that correspond to reasonable values of human pitch.[5]

[5]Human pitch, across genders, will likely always fall between 50 Hz and 400 Hz [47], which for a 16 kHz sampling

For the purposes of RBAR, the specific value of the pitch is not important, only the degree to which the speech signal is voiced, which is quantified by the magnitude of the cepstral peak corresponding to the pitch (if any). In the absence of voicing, the maximum value will be relatively small. Figure 4.12 shows the change in the cepstral peak value over time, for the waveform shown, before and after clipping at $\tau = \mathrm{P}_{55}$. This figure clearly depicts the relationship of the cepstral peak value to speech voicing (e.g., the segments from 0.7 s to 0.9 s, 1.8 s to 1.95 s, and 2.2 s to 2.35 s all contain speech energy but have correspondingly low-valued cepstral peaks because they are unvoiced). Further, based on the similarity of the two cepstral peak time series, it is clear that the cepstral peak is relatively invariant to clipping, and is therefore an ideal feature to use for this purpose. Based on empirical observations and preliminary ASR experimental results, a voiced/unvoiced decision threshold of 3 was chosen.[6] That is, if the cepstral peak of a given speech frame is greater than or equal to 3, the frame is deemed voiced, otherwise it is unvoiced.

4.5 Results

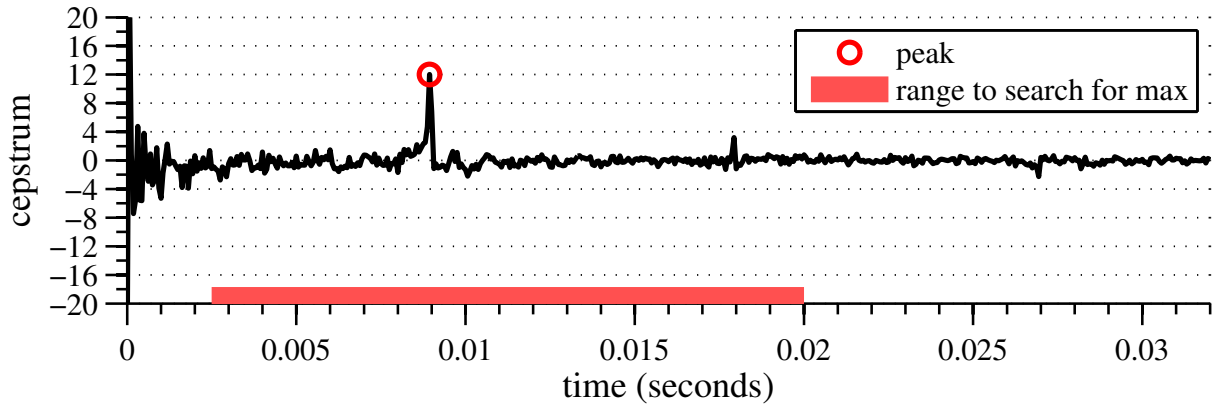
4.5.1 Declipping performance

The results of speech recognition experiments using all of the previously-described algorithms are shown in Fig. 4.13. Of the five non-trivial clipping thresholds considered (the 100^{th} percentile means no clipping), CBAR is the best performing algorithm 80% of the time. For $\tau = \mathrm{P}_{95}$ and $\tau = \mathrm{P}_{75}$, RBAR performs second best, the performance of Kitic-IHT and RBAR are matched at $\tau = \mathrm{P}_{55}$, and Kitic-IHT performances second best for lower thresholds. Kitic-IHT slightly outperforms CBAR for $\tau = \mathrm{P}_{15}$. As will be shown in Section 4.5.2, RBAR is the fastest-running algorithm, by far, 100% of the time.

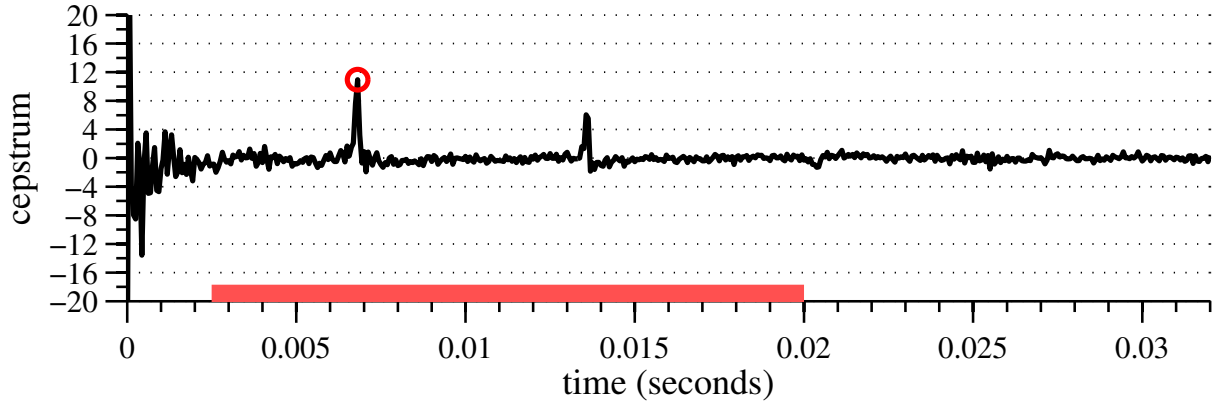
Relative to baseline ASR performance with no declipping, CBAR provides an average WER improvement of 40.4%. The maximum relative WER improvement is 56.2% at $\tau = \mathrm{P}_{75}$; the minimum relative WER improvement is 8.4% at $\tau = \mathrm{P}_{15}$; there were no test conditions under which

rate, corresponds to pitch periods between 2.5 ms and 20 ms.

[6] In truth, the scientifically-proper way to determine the optimal voiced/unvoiced decision threshold would be to perform a *forced alignment* on a large database of speech clipped at varying thresholds, separate voiced and unvoiced speech segments, compute the cepstral peak values for each segment, and finally choose the threshold for which the least amount of probabilistic error occurs based on the shape of the distributions of the cepstral peaks for voiced and unvoiced speech.



(a) Spoken at a pitch of approximately 110 Hz.



(b) Spoken at a higher pitch of approximately 150 Hz.

Figure 4.11: Cepstrum of the vowel AE as in "fast" spoken at two different pitches by a male speaker as shown originally in Fig. 2.3. The red circles indicate the maximum value of the cepstrum in the range logical for human voicing (50 Hz - 400 Hz; indicated by the red stripe). The top plot's peak occurs at 8.9 ms corresponding to a pitch of 112 Hz; the bottom plot's peak occurs at 6.8 ms, which corresponds to a frequency of 146.8 Hz.

CBAR did not improve performance. Relative to baseline ASR performance with no declipping, RBAR provides an average WER improvement of 28.1%. The maximum relative WER improvement is 49.6% at $\tau = \mathrm{P}_{75}$; the minimum relative WER improvement is -7.5% at $\tau = \mathrm{P}_{15}$ (i.e., performance was degraded relative to baseline), the smallest *positive* relative WER improvement is 21.7% occurring at $\tau = \mathrm{P}_{35}$. Relative to Kitic-IHT, CBAR provides an average WER improvement of 13.2%. The maximum relative WER improvement is 24.4% at $\tau = \mathrm{P}_{75}$; the minimum relative WER improvement is 8.6% at $\tau = P95$. CBAR degrades performance by 4.8% relative to Kitic-IHT for $\tau = \mathrm{P}_{15}$. Relative to Kitic-IHT, RBAR provides an average WER improvement of -4.9%; it is maximally 12.9% better than Kitic-IHT at $\tau = \mathrm{P}_{75}$; and 23% worse at $\tau = \mathrm{P}_{15}$.

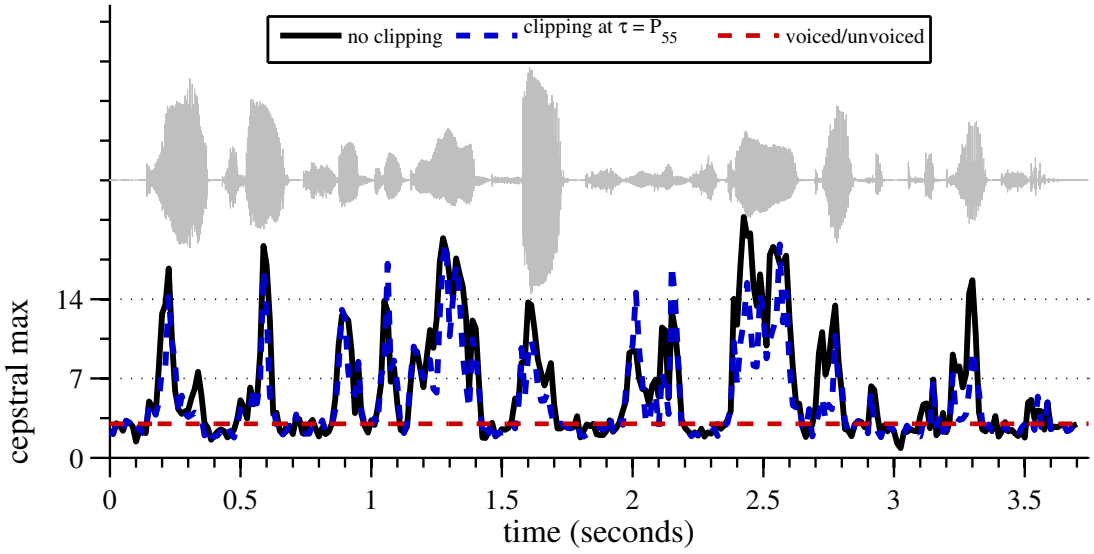


Figure 4.12: Evolution of the cepstral maximum over time for the waveform shown, before and after clipping at $\tau = \mathrm{P}_{55}$. The cepstral maximum is shown to be robust to clipping. Speech frames for which the cepstral maximum is greater than or equal to 3 are taken to be voiced, and are deemed unvoiced otherwise. This is not the optimal Bayesian decision threshold. The waveform is not drawn to scale.

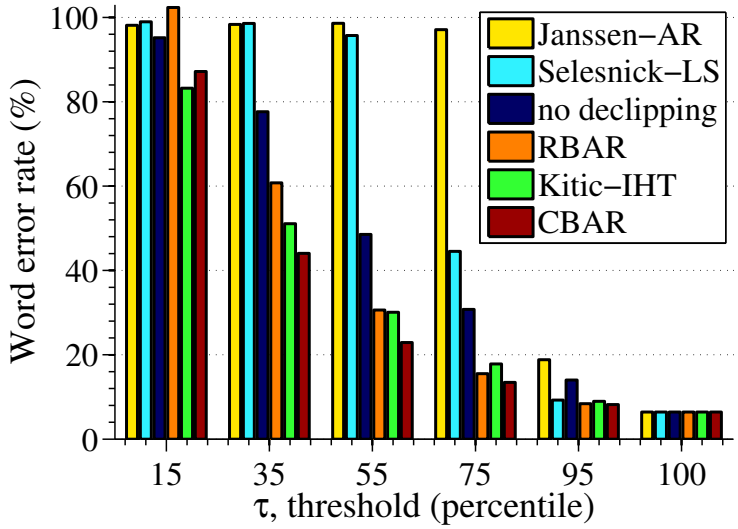
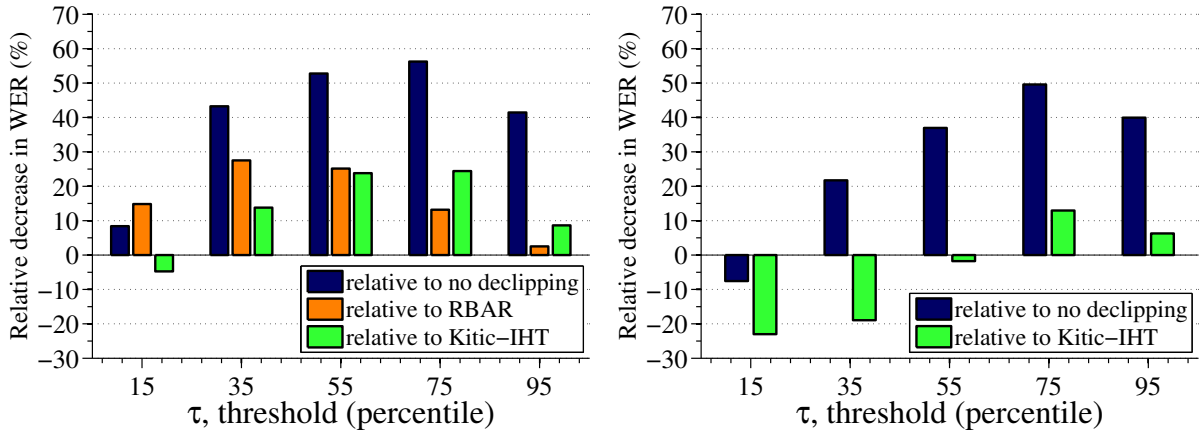


Figure 4.13: Results of speech recognition experiments on speech clipped at varying thresholds and then declipped with the indicated algorithm. The ASR system was trained on clean, unclipped speech features. Note that CBAR is the best performing algorithm in 80% of the test cases. RBAR matches or exceeds the performance of Kitic-IHT for $\tau \geq \mathrm{P}_{55}$.



(a) Constrained BAR.

(b) Regularized BAR.

Figure 4.14: Relative decrease in WER using Constrained and Regularized Blind Amplitude Reconstruction algorithms. These percentages are derived from the underlying word error rates in Fig. 4.13.

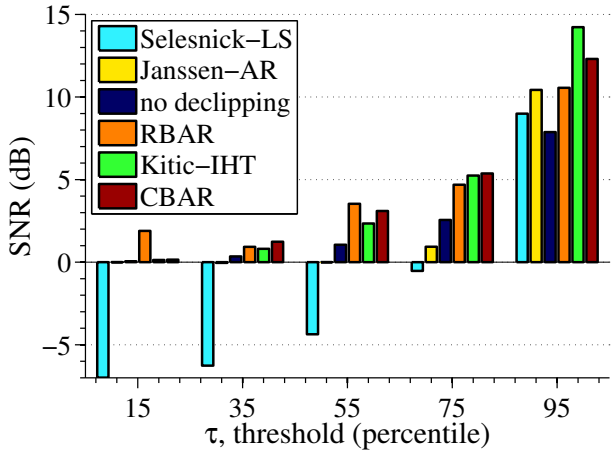
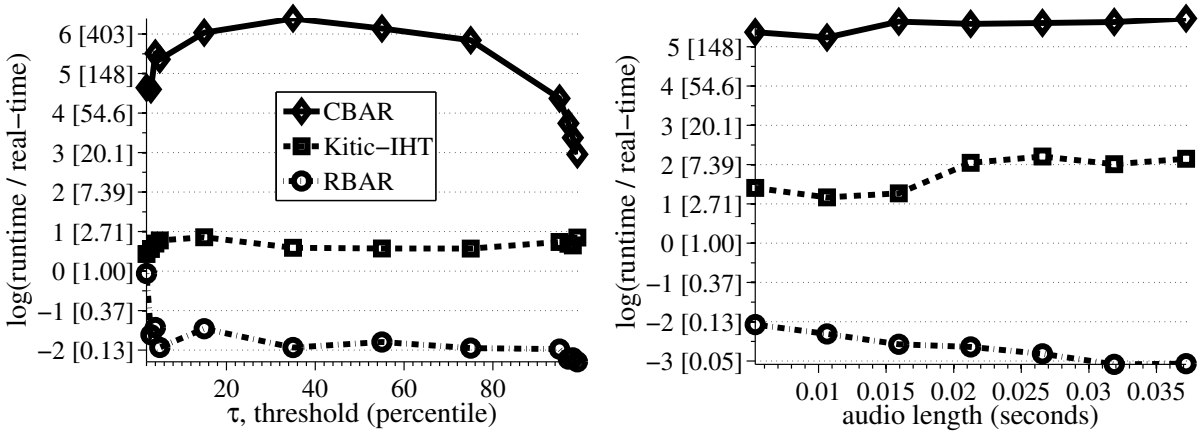


Figure 4.15: Median SNR of the RM1 speech database clipped at varying thresholds and then declipped with the indicated algorithm. The SNR of each waveform was computed using Eq. 2.13.

4.5.2 Algorithm efficiency

Despite the efficacy of CBAR and Kitic-IHT in increasing the SNR of clipped speech, and reducing the WER when clipped speech is decoded, both algorithms are relatively computationally inefficient. The motivation behind the development of RBAR was the utility of having a simultaneously efficient and effective solution. To understand the differences in algorithm efficiency in a pragmatic setting, the following definition of "times real-time" will be used:



(a) Runtime as a function of clipping threshold for a 5.3125 ms speech segment.

(b) Runtime as a function of segment length with $\tau = \mathrm{P}_{95}$.

Figure 4.16: Average runtime of declipping algorithms over 50 independent trials when used to repair a voiced speech segment. The plotted data depicts the natural logarithm of the ratio of the runtime to the duration of the segment; the actual "times real-time" value is indicated in brackets on the left vertical axis.

$$\text{times real-time} = \frac{\text{runtime}}{\text{real-time}} \tag{4.37}$$

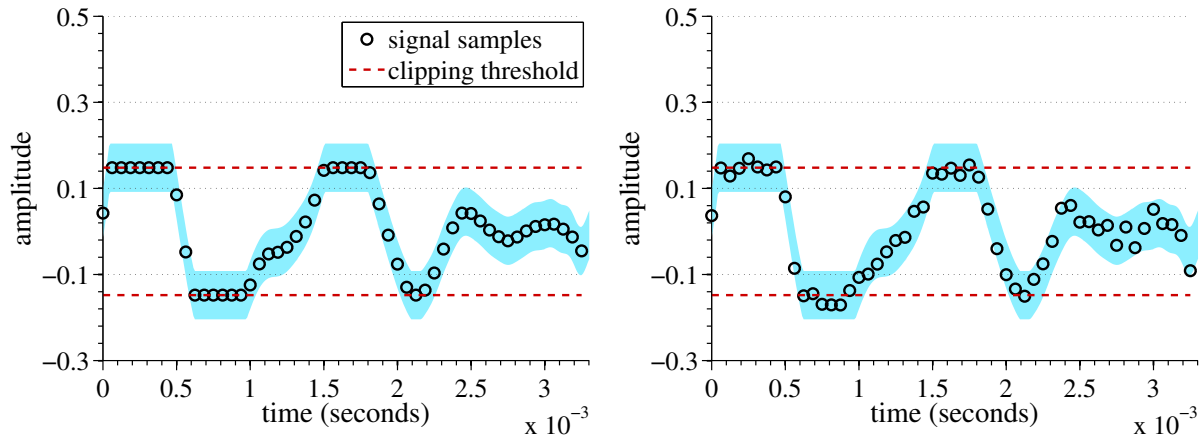
For example, if a given algorithm takes 4 seconds to process an audio segment of duration 2 seconds, its *times real-time* value is $4/2 = 2$. Figure 4.16a depicts the logarithm[7] of the average[8] *times real-time* value for each of the three relevant declipping algorithms (Kitic-IHT, CBAR, and RBAR) for a 5.3125 ms segment[9] of clipped speech, as a function of the clipping threshold, τ. The true (linear) *times real-time* value is indicated in brackets on the left vertical axis.

Kitic-IHT has relatively constant efficiency, across τ, between 1.5 and 2.5 times real-time. RBAR has a similarly constant efficiency across τ, generally between 0.1 and 0.2 times real-time. On the other hand, CBAR has an efficiency that varies as a function of τ. At best, CBAR runs at approximately 20 times real-time, for very high τ; at worst, CBAR's efficiency exceeds 400 times real-time for $\tau \approx \mathrm{P}_{35}$. Again, for very low τ, CBAR's efficiency drops to approximately 100 times real-time. This result suggests that CBAR's efficiency is dependent on the ratio of the number of clipped samples to the number of unclipped samples. When the ratio is nearly zero (i.e., $\tau \to \mathrm{P}_{100}$),

[7]The logarithm is used due to the large differences in runtime across the algorithms.

[8]The average is taken over 50 independent trials. The runtime experiments were run on a 2011 Apple MacBook Air with a 1.7 GHz Intel Core i5 processor and 4 GB of RAM. The software was run in MATLAB R2013a.

[9]85 samples at 16 kHz sampling rate

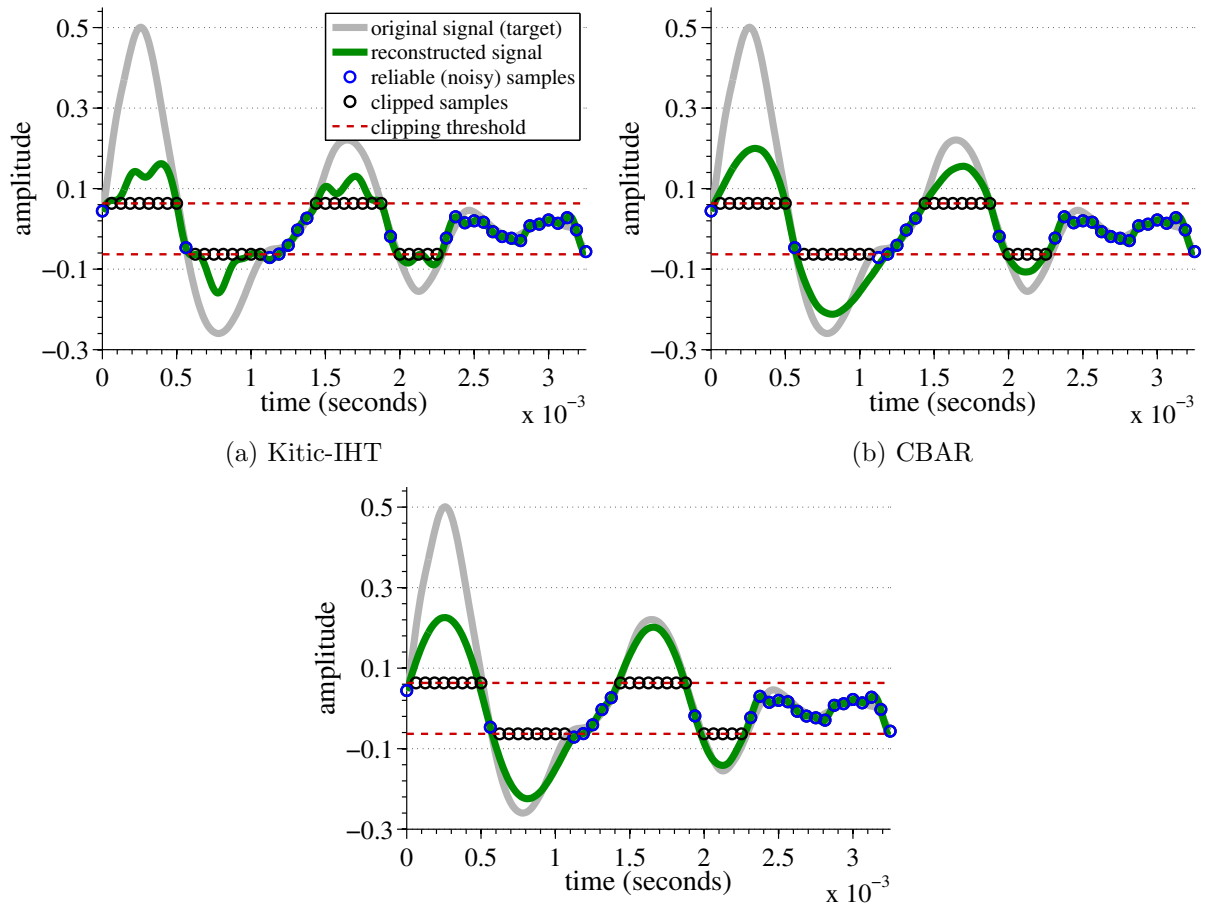


(a) SNR = ∞ dB (b) SNR = 10 dB

Figure 4.17: One pitch period of voiced speech clipped at $\tau = 0.1481$ (P_{95}) before and after the addition of white Gaussian noise at 10-dB SNR. The shaded region around the signal samples represents three standard deviations of the Gaussian noise ($\sigma = 0.0187$); i.e., after the addition of noise at 10-dB SNR, each signal sample has a 99.7% likelihood of being vertically perturbed by an amount that causes its value to lie in the shaded region.

there are few samples to infer, thus shrinking the optimization algorithm's necessary search space. When the ratio is very large, (i.e., $\tau \to P_0$), there are very few reliable samples to match and it is simpler for the optimization algorithm to find candidate interpolating sequences with sufficiently small second derivatives. When the ratio is near 1, however, approximately half the signal segment must be interpolated, while simultaneously matching to the remaining half of reliable samples. This confluence of circumstances renders the optimization much more challenging to solve and thus causes CBAR to be exceedingly slow.

Figure 4.16b again shows the logarithm of the declipping algorithms' *times real-time* efficiency for a voiced speech segment clipped at $\tau = P_{95}$. In this plot, however, the efficiency is plotted as a function of segment length. Note that CBAR's efficiency is reasonably invariant to audio length (i.e., it runs in linear time). In contrast, Kitic-IHT exhibits an increased runtime when the audio length is quadrupled from the 5.3125 ms duration, jumping from approximately 4 times real-time to 8 times real-time. Finally, RBAR exhibits constant time efficiency, with its times real-time value decreasing with increasing segment length.



(a) Kitic-IHT

(b) CBAR

(c) RBAR

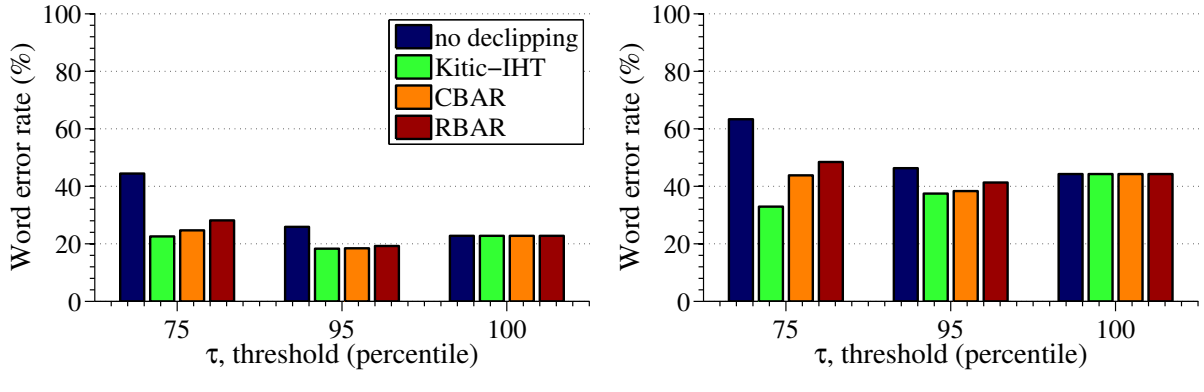
Figure 4.18: Examples of declipping outcomes using the Kitic-IHT, CBAR, and RBAR algorithms in AWGN at 10-dB SNR. Note that the reconstructions are all visually comparable to the noise-free case in Figs. 4.6d, 4.7d, and 4.8d implying that the algorithms are reasonably robust. Oracle knowledge of the clipped samples is assumed to be known a priori.

4.6 Declipping in noise

The challenge of restoring a clipped signal is compounded by the effects of independent channel noise, which randomly perturb the individual samples of an observed signal, as demonstrated in Fig. 4.17. The introduction of channel noise after clipping presents two primary challenges: (1) the clipping threshold, τ, can no longer be trivially identified as the maximum value of the observed waveform,[10] and (2) given knowledge of which samples were clipped before the addition of noise, the accuracy of the reconstruction may be disrupted by noise in the remaining reliable samples.

For the purposes of this chapter, it is assumed that clipped samples can be identified with 100%

[10]Even if the value of τ is known, the identification of which samples were clipped before noise addition is no longer trivial because the clipped samples will have been randomly perturbed about the clipping threshold.



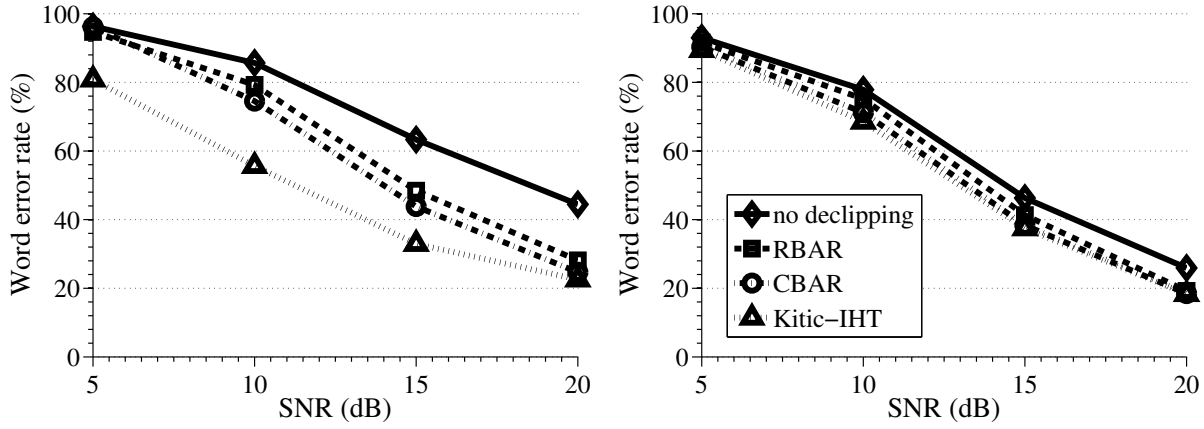
(a) Clipped speech plus AWGN at 20-dB SNR

(b) Clipped speech plus AWGN at 15-dB SNR

(c) Clipped speech plus AWGN at 10-dB SNR

(d) Clipped speech plus AWGN at 5-dB SNR

Figure 4.19: Results of declipping in noise using the indicated algorithm. Additive white Gaussian noise was superimposed on the clipped signal at the indicated SNR. Recall that $\tau = \mathrm{P}_{100}$ indicates no clipping. The SNR reflects the intensity of the additive noise only, and is measured with respect to the power in the clipped signal.



(a) $\tau = \mathrm{P}_{75}$

(b) $\tau = \mathrm{P}_{95}$

Figure 4.20: Results of declipping in noise using the indicated algorithm, plotted as a function of SNR. These results match those of Fig. 4.19, but simply are plotted in a different fashion.

accuracy, even in the presence of noise. The identification of τ and the subsequent identification of clipped samples in noise is the topic of Chapter 5. Figures 4.19 and 4.20 depict WER after declipping in noise at four different signal-to-noise ratios, 5 dB, 10 dB, 15 dB, and 20 dB. While all of the algorithms provide improvement in performance over the baseline no-declipping case, the Kitic-IHT algorithm is observed to be the most robust to noise, especially as the SNR and clipping threshold decrease.

4.7 Summary

This chapter thoroughly explored the phenomenon of *clipping*, which is a special case of DRC when the ratio parameter is infinite. Various algorithms have been developed over the past few decades to repair clipped signals, but many of them are only effective for the most benign clipping thresholds (e.g., $\tau \geq \mathrm{P}_{95}$). The key principle behind developing successful declipping algorithms for relatively low sampling rate signals (e.g., 8 or 16 kHz speech) lies in the need to guide the interpolation such that it remains "above" the clipping threshold, in the absolute sense. The two novel algorithms outlined in this chapter, CBAR and RBAR, achieve this goal through hard constraints and regularization, respectively. Topics of associated interest that were discussed include voiced speech detection by cepstral analysis and amplitude prediction by modeling a pre-clipped speech signal's amplitude as a function of the fraction of clipped samples in a given frame. The CBAR algorithm is shown to be more effective in reducing ASR WER than Kitic-IHT, the previous state-of-the-art declipping algorithm. On the other hand, while RBAR's performance is slightly worse, it is significantly more computationally efficient. Both algorithms are reasonably robust to additive noise.

Chapter 5

Robust Estimation of Nonlinear Distortion (RED)

5.1 Introduction

Chapter 2 of this thesis introduced the notion of dynamic range compression (DRC), defined by the following two-parameter deterministic function:

$$f(x[n]; R, \tau) = \begin{cases} \tau^{(1-\frac{1}{R})} |x[n]|^{\frac{1}{R}} \operatorname{sgn} x[n] & \text{if } |x[n]| \geq \tau \\ x[n] & \text{if } |x[n]| < \tau \end{cases} \tag{5.1}$$

DRC achieves exactly what its name implies; that is, the DRC function reduces the dynamic range of any signal passing through it, provided the signal's maximum value exceeds the DRC threshold, τ. When R is finite, Eq. 5.1 imposes *soft clipping* or *compression*; when $R = \infty$, DRC devolves to *hard clipping*, or simply, *clipping*. The accuracy of automatic speech recognition software decreases in relation to the intensity of the dynamic range compression; i.e., when either τ decreases or R increases, the word error rate monotonically increases, as illustrated by the results of Fig. 2.9b. Various novel techniques designed to compensate for the DRC with limited use of *a priori* information are discussed in Chapters 3 and 4, which concern Blind Amplitude Normalization (for compression) and Blind Amplitude Reconstruction (for clipping), respectively.

Despite the intent of limiting the use of *a priori* information in developing compensation algorithms, it is an oft-unavoidable situation. For example, BAN is capable of producing a non-parametric (look-up table) estimate of the DRC nonlinearity, but it requires an estimate of the CDF of the uncompressed signal. Fortunately, this is usually trivial to obtain, given knowledge of the application domain (e.g., speech or music). For BAR, declipping requires the knowledge of which samples are clipped, which in the absence of noise, is equivalent to knowing the clipping threshold, τ. This is also trivial to obtain, as it is simply equal to the maximum value of the waveform. In the presence of noise, however, τ becomes harder to estimate with certainty, and even given oracle knowledge of τ, the determination of which samples are clipped is subject to stochastic phenomena. Finally, the classification of speech as either clipped or compressed in general, at least requires a determination of the finiteness of R. To these ends, this chapter discusses the challenges and implications, and approaches to, estimating the DRC threshold parameter, τ. The estimation of R and the subsequent classification of speech as either clipped or compressed is reserved for future research.

5.2 Pre-compression gain

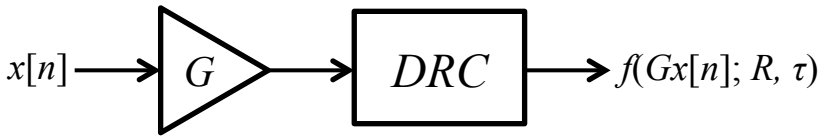


Figure 5.1: A practical signal path for DRC which includes a pre-compression gain stage. Here, the value of the gain, G, and the actual value of τ together determine the "effective" threshold (e.g., in terms of percentiles of the input signal).

In Chapters 2, 3, and 4, the threshold parameter, τ, has been expressed in terms of percentiles of the (absolute value of the) input signal. This allows for a more general presentation of experimental results, because the percentile value of a signal varies in proportion to the scaling of the waveform (thus, for example, the experimental results presented thus far would remain the same if the data were scaled arbitrarily). To prove that this is true, consider the following.

The definition of *percentile* as used throughout this document is:

$$\mathrm{P}_p = x \text{ s.t. } \Pr(X \leq x) = 0.01p \tag{5.2}$$

So, for example, the 75^{th} percentile, P_{75}, is the value x such that the probability that the random variable X is less than or equal to x is 0.75. Defining a new random variable:

$$Y = G \cdot X \tag{5.3}$$

In Eq. 5.3, G is a constant scaling factor, or *gain*. This relationship implies the following:

$$\Pr(X \leq x) = \Pr\left(\frac{Y}{G} \leq x\right) = \Pr(Y \leq Gx) \tag{5.4}$$

Therefore, if the p^{th} percentile of random variable X is x, and the random variable $Y = GX$, then the p^{th} percentile of Y is Gx. Thus, the effect of DRC is independent of G when τ is expressed in terms of percentiles of the input.

In a real-world application of DRC, however, τ is set in terms of its actual amplitude value (usually expressed in dB). As depicted in Fig. 5.1, this means that the effect of DRC on an input signal is actually a function of three parameters: τ, R, and the pre-compression gain factor, G.

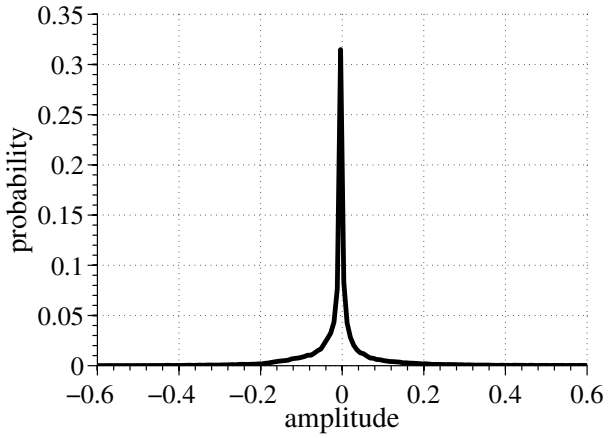


Figure 5.2: Waveform amplitude distribution of a typical speech utterance. The speech was normalized to a maximum amplitude of 1.0.

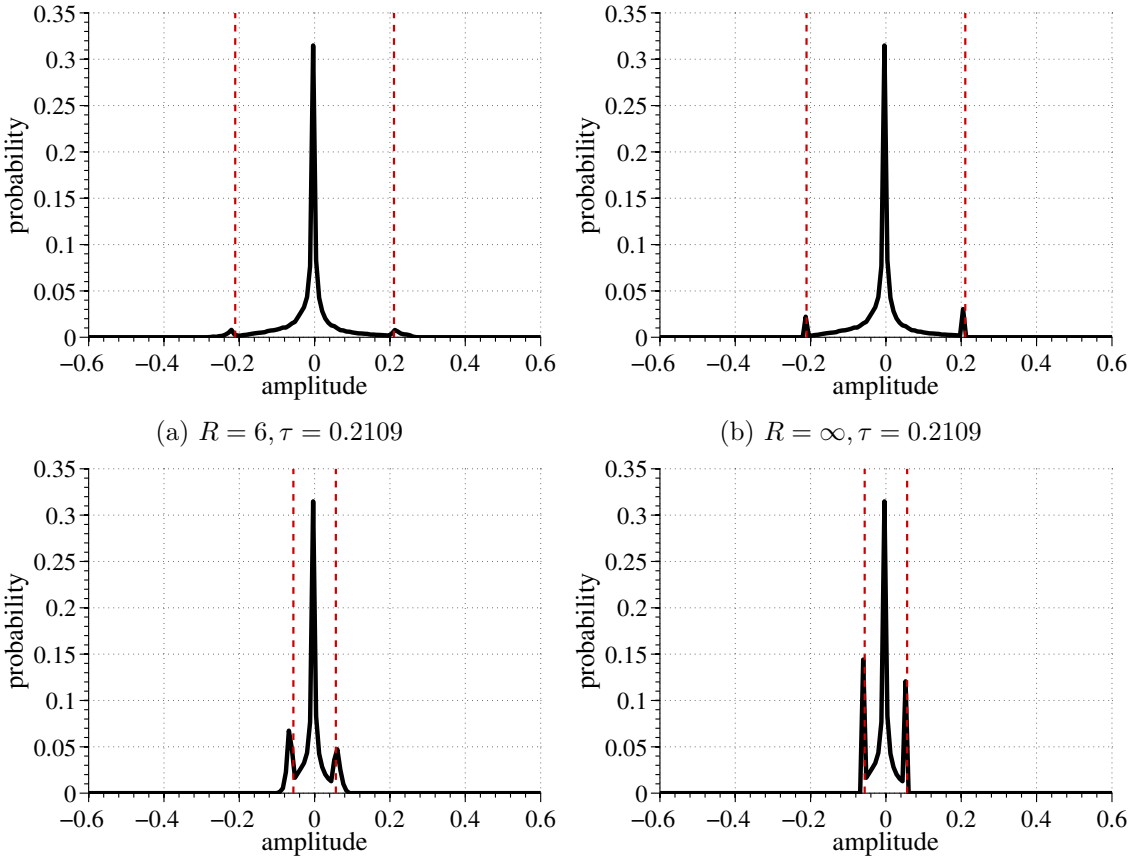
When DRC is inadvertently applied to a signal (e.g., as a result of exceeding the dynamic range limitations of a system), one can imagine that $\tau = 1$ (and $R = \infty$ for a digital system or some large, finite value for an analog system) and the degree to which a signal is clipped is a function of G alone, which controls the amount by which the input signal exceeds the system's dynamic range capabilities.

5.3 DRC threshold estimation

5.3.1 Amplitude value of τ

In the case of clipping for which $R = \infty$, and assuming the degradation model of Fig. 2.5c, or in the absence of noise ($w[n] = 0$) when using the degradation model of Fig. 2.5b, the amplitude value of τ is equal to the maximum value of the waveform. This is the simplest case.

When $R < \infty$ and/or when noise is present in the degradation model of Fig. 2.5b, it is not obvious how to estimate τ by inspection of the waveform. As most clearly evidenced by BAN (Ch. 3), the application of DRC modifies the waveform's amplitude distribution. This is explicitly depicted in Fig. 5.3, which shows the distribution of the speech (whose unmodified distribution is shown in Fig. 5.2) after DRC with a number of parameter combinations. The corresponding distributions, after the addition of AWGN at 15-dB SNR according to the degradation model of Fig. 2.5b, are shown in Figs. 5.4 and 5.5.



(a) $R = 6, \tau = 0.2109$

(b) $R = \infty, \tau = 0.2109$

(c) $R = 6, \tau = 0.0564$

(d) $R = \infty, \tau = 0.0564$

Figure 5.3: Waveform amplitude distributions of the same speech utterance used in Fig. 5.2 after DRC. Dashed red vertical lines indicate the location of $\pm\tau$.

If two statistically independent random variables, X and Y, are summed together, the probability density function (PDF) of the resulting random variable is equal to the convolution of the PDFs of X and Y [48]. The effect of this can be seen in Fig. 5.5 where three Gaussian-shaped lobes appear at the spikes corresponding to $\pm\tau$ and 0.

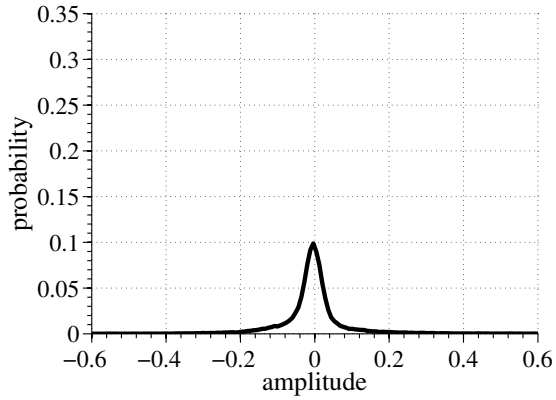
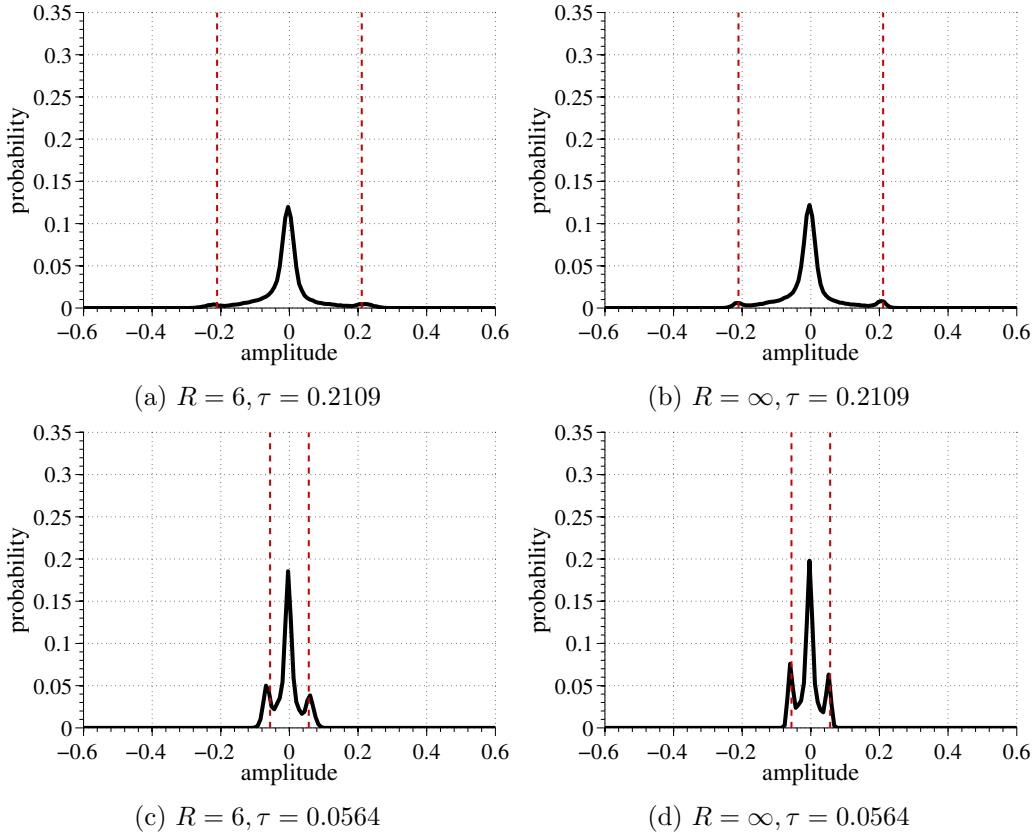


Figure 5.4: Waveform amplitude distribution of speech plus noise at 15-dB SNR. The speech was normalized to a maximum amplitude of 1.0.



(a) $R = 6, \tau = 0.2109$

(b) $R = \infty, \tau = 0.2109$

(c) $R = 6, \tau = 0.0564$

(d) $R = \infty, \tau = 0.0564$

Figure 5.5: Waveform amplitude distributions of the same speech utterance used in Fig. 5.4 after DRC and noise addition at 15-dB SNR. Dashed red vertical lines indicate the location of $\pm\tau$.

Observation of Figs. 5.2−5.5 illustrate that the value of τ is closely related to the location of the outer peak values in the compressed waveform distributions, whether or not noise is present. This observation can be leveraged to design a method of estimating τ from a waveform amplitude distribution. Consider a sequence of data with K peaks whose locations with respect to the independent variable are $\{k_0, k_1, k_2, ..., k_{K-1}\}$. If a *peak-finding algorithm*[1] is applied to a speech waveform distribution, an estimate of the value of τ is given by:

$$\tilde{\tau} = \frac{1}{K-1} \sum_{i=0}^{K-1} |k_i| \tag{5.5}$$

When clipping or compression has occurred, $K = 3$ and the individual peaks theoretically should equal $k_0 = -\tau$, $k_1 = 0$, $k_2 = \tau$; the sum in Eq. 5.5 is then effectively $\frac{2\tau}{2} = \tau$. If no compression has occurred, $K = 1$, and the result diverges to ∞, which is correct. Thus, this technique simultaneously performs *regression* to predict τ and *binary classification* to determine whether or not the speech has been exposed to DRC at all. The accuracy of this method in performing regression is presented in Figs. 5.6−5.9, which show the distributions of the estimator when $\tau = \mathrm{P}_{75}$ and the SNR varies between 5 dB and 20 dB, and Figs. 5.10−5.13, which show the means and standard deviations of the estimator for variable τ, and again, as the SNR varies between 5 dB and 20 dB. The technique is remarkably accurate in all cases except SNR = 5 dB.

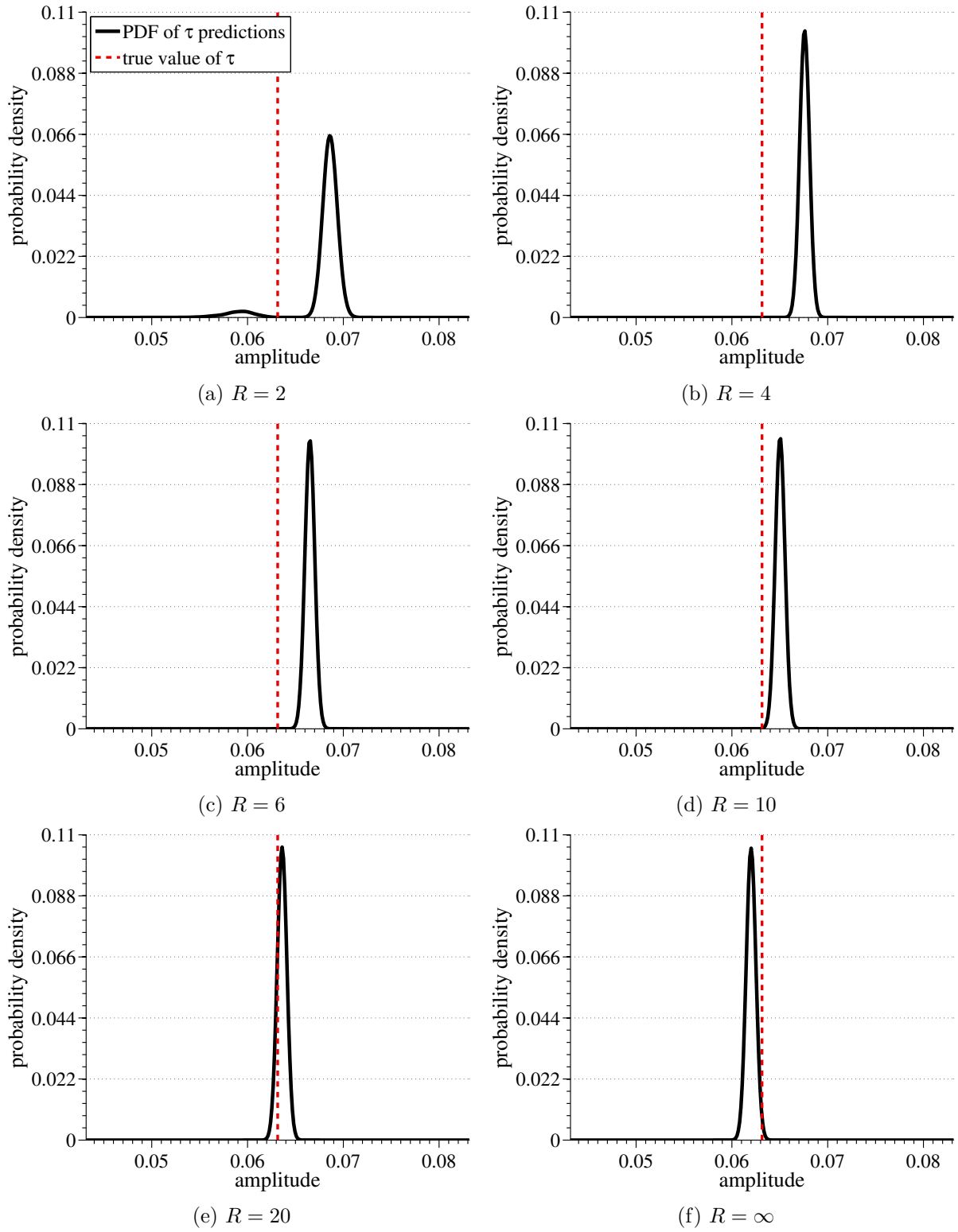
5.3.2 Percentile value of τ

For declipping applications, the amplitude value of the clipping threshold, τ, is sufficient to determine which samples need to be interpolated.[2] As will be shown in Ch. 6, it is often also of interest to determine the corresponding percentile value at which the speech has been compressed or clipped. Referring to Fig. 5.1, this is equivalent to estimating the pre-DRC gain value, G.

Nonetheless, rather than attempting to estimate G and then inferring the percentile, the percentile value can be approximated directly. In fact, the percentile value of τ is approximately equal to the integral (cumulative sum) of the probability density function of the observed speech between $-\tau$ and $+\tau$. Mathematically, where $c(x)$ is the PDF of the observed speech, and $C(x)$ is

[1] From basic calculus, the peaks of a signal can be found by finding the zeros of the first derivative of the signal. For discrete-time processing, the first derivative is approximated using the first difference.

[2] This is only true in the absence of additional noise, the presence of which complicates the determination of which samples are clipped, and deserves a separate discussion, provided in Section 5.4.

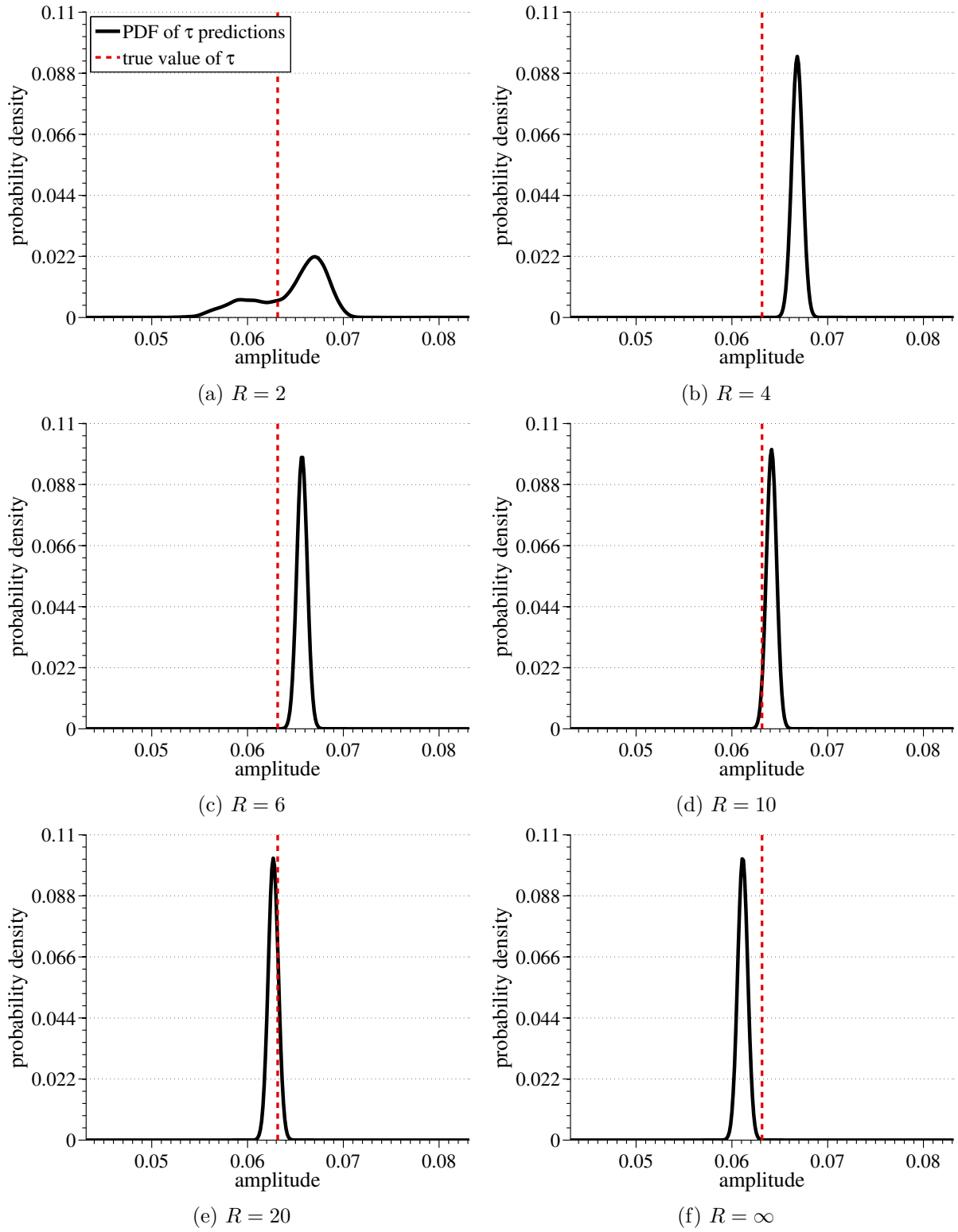


(a) $R = 2$ (b) $R = 4$

(c) $R = 6$ (d) $R = 10$

(e) $R = 20$ (f) $R = \infty$

Figure 5.6: Results of blindly predicting τ using a basic peak-finding algorithm in conjunction with Eq. 5.5. Here, the compressed speech ($\tau = \mathrm{P}_{75}$, variable R) is added to AWGN to achieve 20-dB SNR according to Fig. 2.5b. For a given (R, τ) pair, the value of τ is predicted over 10,000 independent trials of the same compressed speech added to a newly-generated white noise sequence.

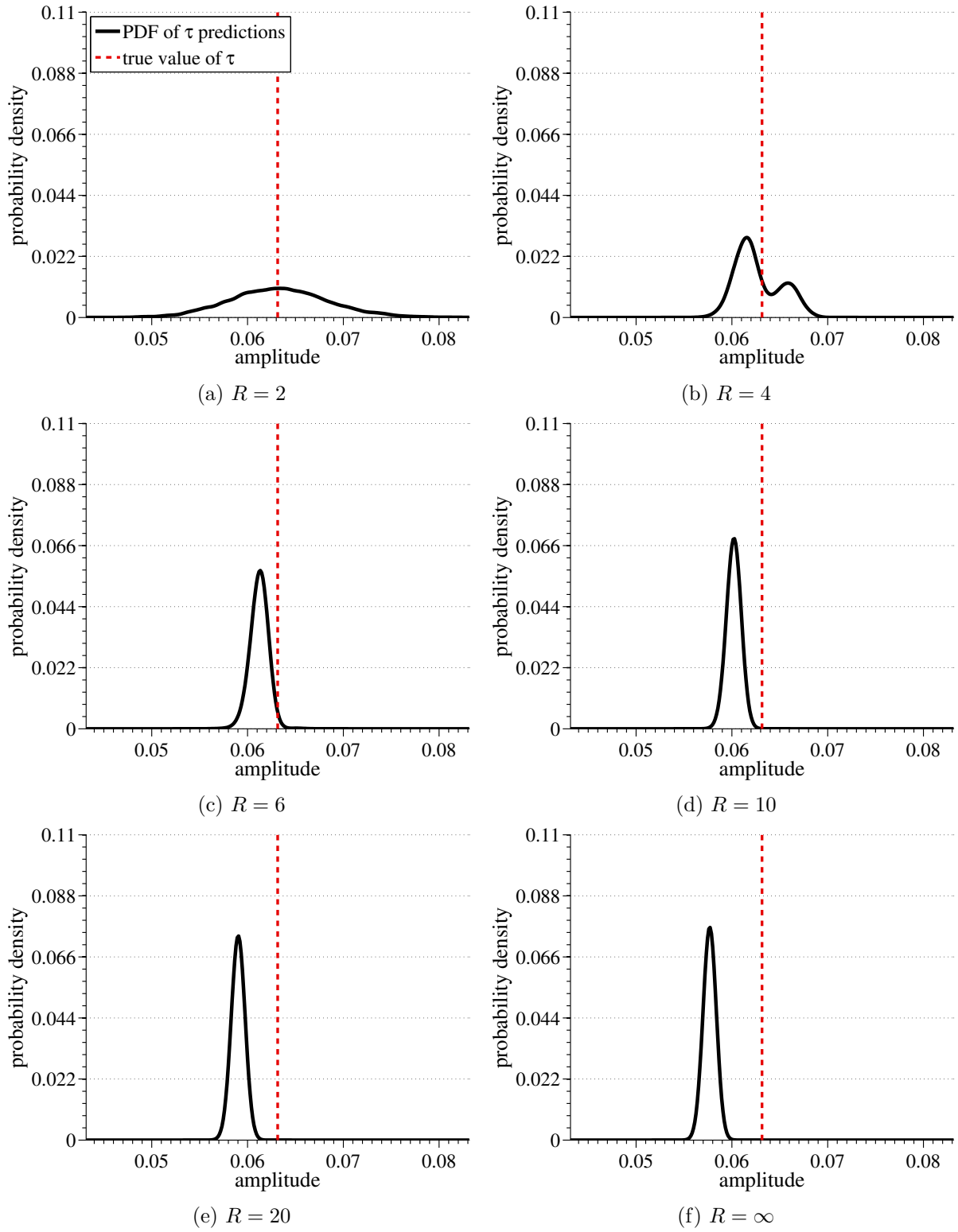


(a) $R = 2$ (b) $R = 4$

(c) $R = 6$ (d) $R = 10$

(e) $R = 20$ (f) $R = \infty$

Figure 5.7: Results of blindly predicting τ using a basic peak-finding algorithm in conjunction with Eq. 5.5. Here, the compressed speech ($\tau = \mathrm{P}_{75}$, variable R) is added to AWGN to achieve 15-dB SNR according to Fig. 2.5b. For a given (R, τ) pair, the value of τ is predicted over 10,000 independent trials of the same compressed speech added to a newly-generated white noise sequence.

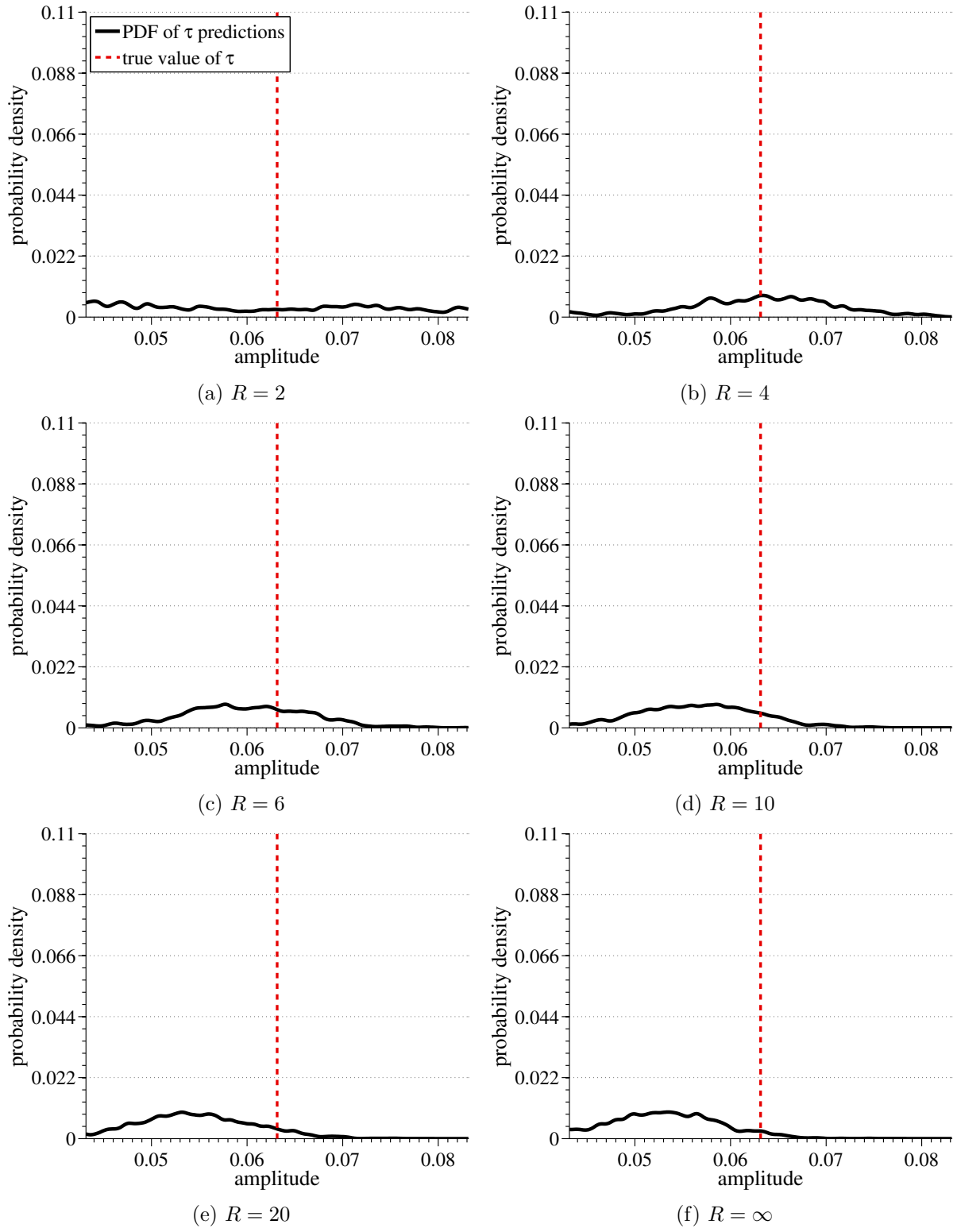


(a) $R = 2$ (b) $R = 4$

(c) $R = 6$ (d) $R = 10$

(e) $R = 20$ (f) $R = \infty$

Figure 5.8: Results of blindly predicting τ using a basic peak-finding algorithm in conjunction with Eq. 5.5. Here, the compressed speech ($\tau = \mathrm{P}_{75}$, variable R) is added to AWGN to achieve 10-dB SNR according to Fig. 2.5b. For a given (R, τ) pair, the value of τ is predicted over 10,000 independent trials of the same compressed speech added to a newly-generated white noise sequence.

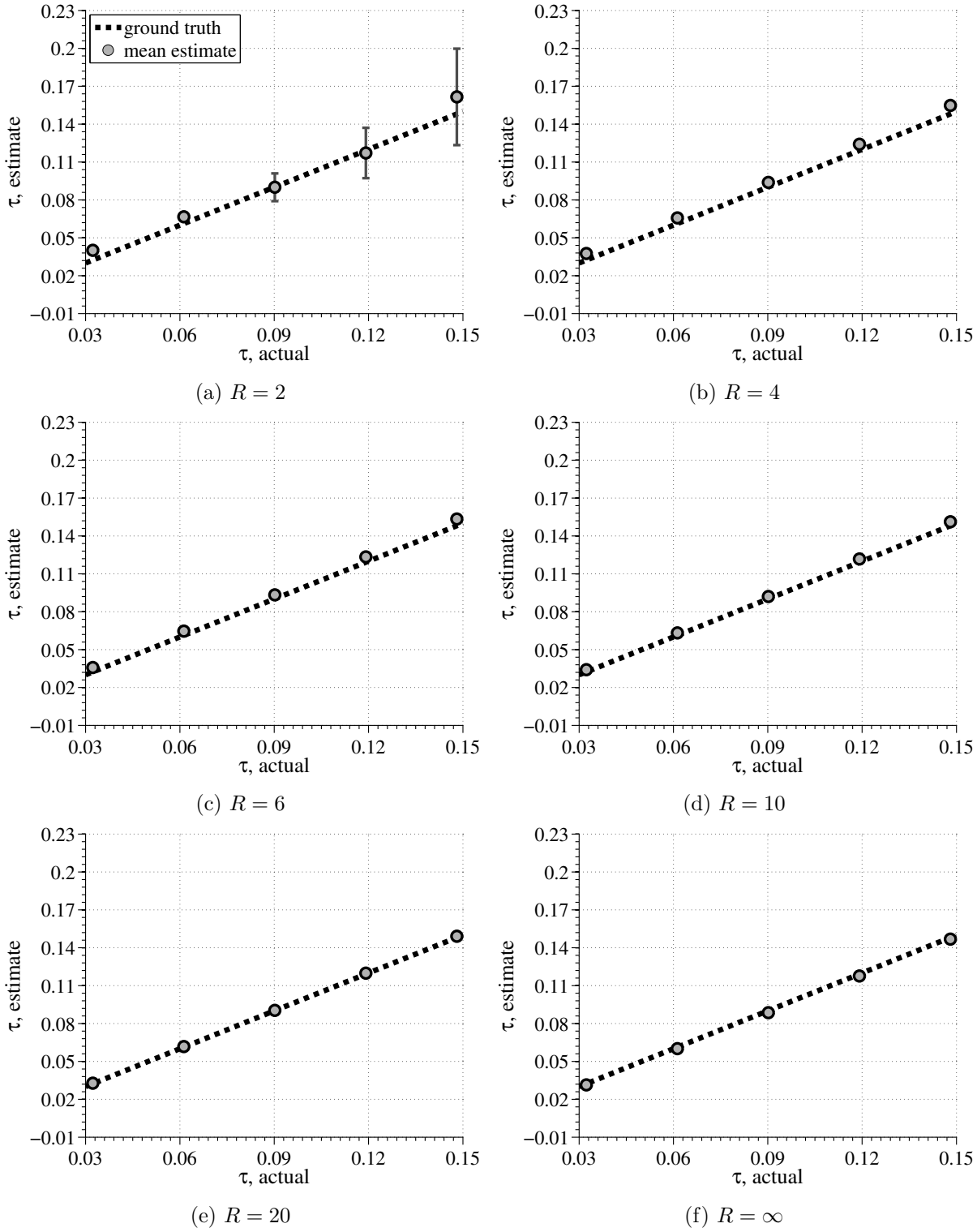


(a) $R = 2$ (b) $R = 4$

(c) $R = 6$ (d) $R = 10$

(e) $R = 20$ (f) $R = \infty$

Figure 5.9: Results of blindly predicting τ using a basic peak-finding algorithm in conjunction with Eq. 5.5. Here, the compressed speech ($\tau = \mathrm{P}_{75}$, variable R) is added to AWGN to achieve 5-dB SNR according to Fig. 2.5b. For a given (R, τ) pair, the value of τ is predicted over 10,000 independent trials of the same compressed speech added to a newly-generated white noise sequence.

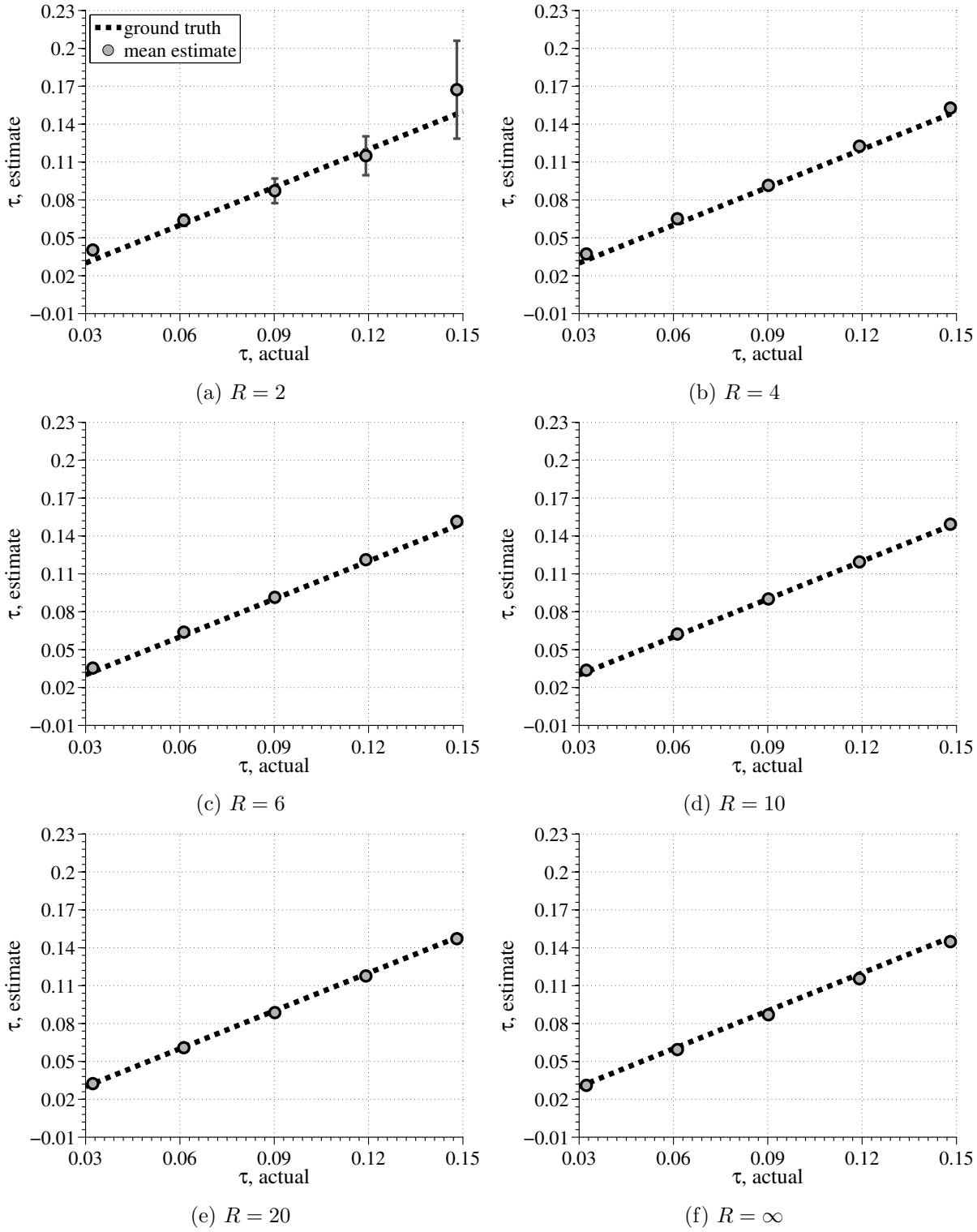


(a) $R = 2$ (b) $R = 4$

(c) $R = 6$ (d) $R = 10$

(e) $R = 20$ (f) $R = \infty$

Figure 5.10: Results of blindly predicting τ using a basic peak-finding algorithm in conjunction with Eq. 5.5. Here, the compressed speech is added to AWGN to achieve 20-dB SNR according to Fig. 2.5b. For a given (R, τ) pair, the value of τ is predicted over 500 independent trials of the same compressed speech added to a newly-generated white noise sequence; the markers show the sample mean of the τ predictions; the error bars extend one standard deviation above and below the mean.

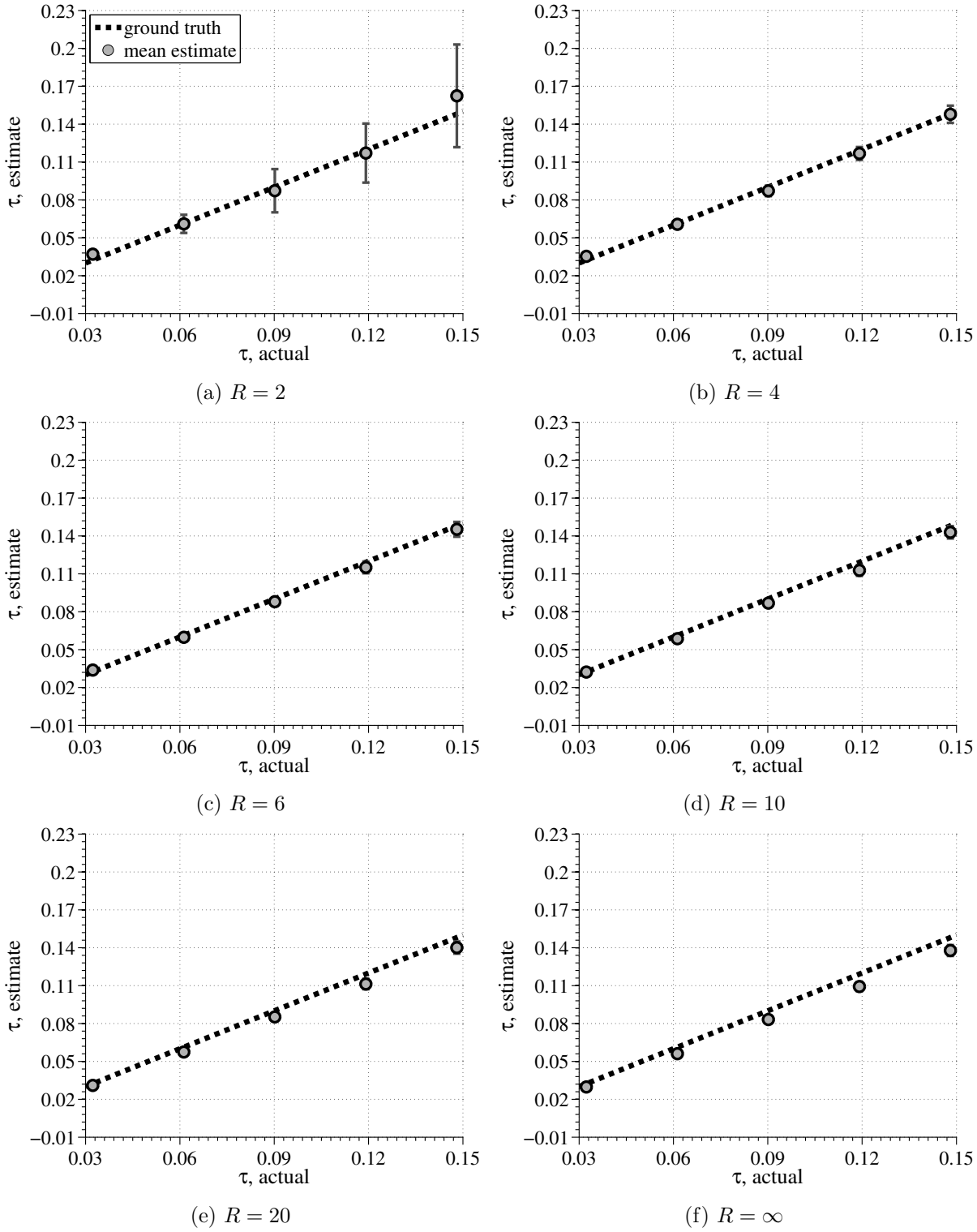


(a) $R = 2$ (b) $R = 4$

(c) $R = 6$ (d) $R = 10$

(e) $R = 20$ (f) $R = \infty$

Figure 5.11: Results of blindly predicting τ using a basic peak-finding algorithm in conjunction with Eq. 5.5. Here, the compressed speech is added to AWGN to achieve 15-dB SNR according to Fig. 2.5b. For a given (R, τ) pair, the value of τ is predicted over 500 independent trials of the same compressed speech added to a newly-generated white noise sequence; the markers show the sample mean of the τ predictions; the error bars extend one standard deviation above and below the mean.

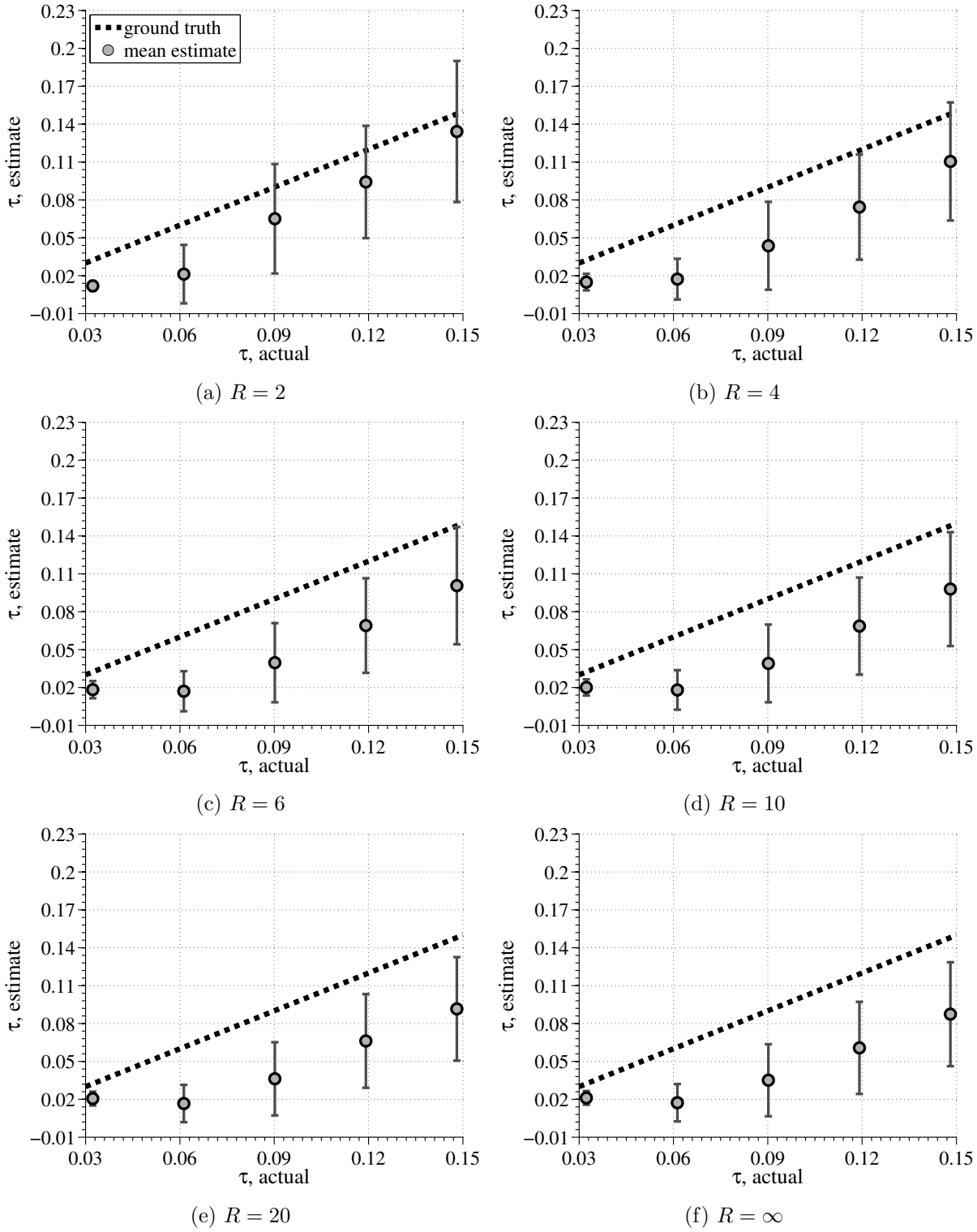


(a) $R = 2$ (b) $R = 4$

(c) $R = 6$ (d) $R = 10$

(e) $R = 20$ (f) $R = \infty$

Figure 5.12: Results of blindly predicting τ using a basic peak-finding algorithm in conjunction with Eq. 5.5. Here, the compressed speech is added to AWGN to achieve 10-dB SNR according to Fig. 2.5b. For a given (R, τ) pair, the value of τ is predicted over 500 independent trials of the same compressed speech added to a newly-generated white noise sequence; the markers show the sample mean of the τ predictions; the error bars extend one standard deviation above and below the mean.

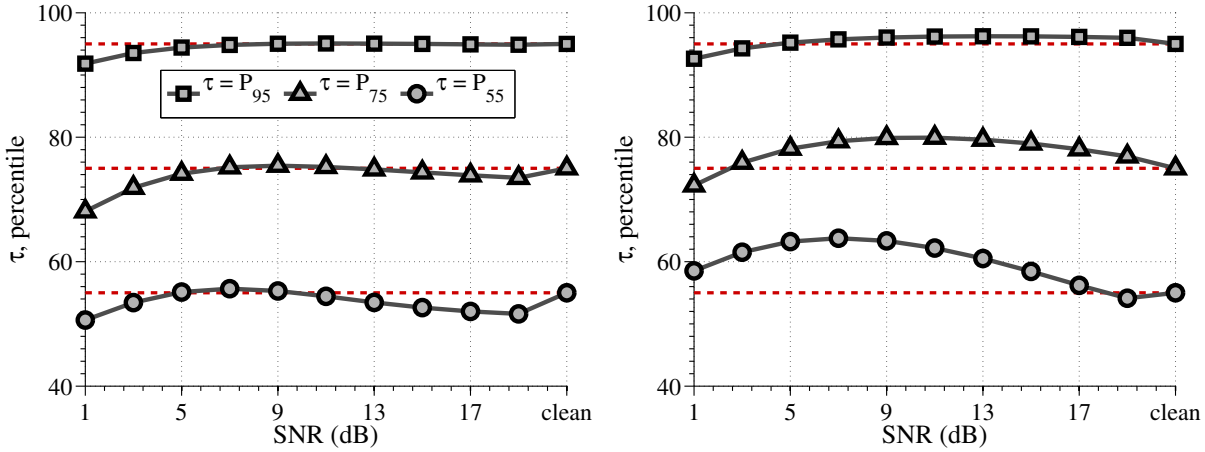


(a) $R = 2$ (b) $R = 4$

(c) $R = 6$ (d) $R = 10$

(e) $R = 20$ (f) $R = \infty$

Figure 5.13: Results of blindly predicting τ using a basic peak-finding algorithm in conjunction with Eq. 5.5. Here, the compressed speech is added to AWGN to achieve 5-dB SNR according to Fig. 2.5b. For a given (R, τ) pair, the value of τ is predicted over 500 independent trials of the same compressed speech added to a newly-generated white noise sequence; the markers show the sample mean of the τ predictions; the error bars extend one standard deviation above and below the mean.



(a) $R = 4$ (b) $R = \infty$

Figure 5.14: Results of blindly predicting the percentile value of τ by accumulating the probability density of the observed (noisy) speech between $-\tau$ and $+\tau$. The markers reflect the sample mean of 500 independent predictions of the percentile value of τ, where a new white noise sequence was generated for each trial. The red dashed lines indicate the target (true) percentiles.

the corresponding cumulative distribution function (CDF):

$$\text{percentile value of } \tau = \int_{-\tau}^{+\tau} c(x)dx = \int_{-\infty}^{+\tau} c(x)dx - \int_{-\infty}^{-\tau} c(x)dx \tag{5.6a}$$

$$= C(\tau) - C(-\tau) \tag{5.6b}$$

The effectiveness of this method of estimation is depicted in Fig. 5.14. For example, Fig. 5.14a shows estimation of the percentile value of τ when $R = 4$ for three different percentiles: 55, 75, and 95. The mean estimated value of the percentile value of τ over 500 independent trials is plotted as a function of SNR. On each trial, a newly-generated white noise sequence was added to clean speech to achieve the indicated SNR; subsequently, the CDF was estimated from the noisy speech, and the percentile value was estimated according to Eq. 5.6.

5.4 Clipped sample estimation

In the presence of noise in the degradation model of Fig. 2.5b, the identification of which samples are clipped−even given the value of τ−is not trivial. Because the addition of noise perturbs

the amplitude of the signal samples, it is no longer possible to know with certainty whether the underlying speech signal's samples were clipped in a certain interval of time. A probabilistic approach is necessary to make an informed decision concerning whether or not a given (series of) sample(s) is clipped.

In particular, the identification of clipped samples is a binary classification problem (i.e., a sample is either clipped or not). For simplicity, it may be assumed that the probability of any given sample being clipped is only a function of its observed amplitude, y_n, the signal's power, σ_y^2, the variance (power) of the white Gaussian noise, σ_w^2, and the (given) value of τ. It would be useful to determine the conditional probability that the output of the DRC function in Fig. 2.5b is equal to $\pm\tau$, given the above information. Proceeding mathematically, the intention is to compute:

$$\Pr(f(x_n) = \pm\tau | y_n, \sigma_y^2, \sigma_w^2, \tau) \tag{5.7}$$

Using Bayes' theorem [49],

$$\Pr(f(x_n) = \pm\tau | y_n, \sigma_y^2, \sigma_w^2, \tau) = \frac{\Pr(y_n | f(x_n) = \pm\tau, \sigma_y^2, \sigma_w^2, \tau)\Pr(f(x_n) = \pm\tau | \sigma_y^2, \sigma_w^2, \tau)}{\Pr(y_n | \sigma_y^2, \sigma_w^2, \tau)} \tag{5.8}$$

The numerator can be simplified slightly by noting that the probability of $f(x_n)$ being clipped is independent of the signal and noise power, and as will be shown, the probability of y_n given that $f(x_n) = \pm\tau$ is independent of the overall signal power; finally, the denominator can be expanded, as follows.

$$\Pr(f(x_n) = \pm\tau | y_n, \sigma_y^2, \sigma_w^2, \tau) = \frac{\Pr(y_n | f(x_n) = \pm\tau, \sigma_w^2, \tau)\Pr(f(x_n) = \pm\tau | \tau)}{\Pr(y_n | f(x_n) = \pm\tau, \sigma_w^2, \tau)\Pr(f(x_n) = \pm\tau | \tau) + \Pr(y_n | f(x_n) \neq \pm\tau, \sigma_y^2, \sigma_w^2, \tau)\Pr(f(x_n) \neq \pm\tau | \tau)} \tag{5.9}$$

Under the assumption of zero-mean AWGN with variance σ_w^2, the probability of the noisy signal having observed value y_n given that $f(x_n) = \pm\tau$ is:

$$\Pr(y_n|f(x_n) = \pm\tau, \sigma_w^2, \tau) = \lim_{\epsilon \to 0} \int_{|y_n|-\epsilon}^{|y_n|+\epsilon} \frac{1}{\sqrt{2\pi}\sigma_w} e^{-\frac{(t-\tau)^2}{2\sigma_w^2}} dt \tag{5.10}$$

Moreover, the probability of a given sample $f(x_n)$ being equal to $\pm\tau$ is related to the percentile value of τ:

$$\Pr(f(x_n) = \pm\tau|\tau) = 1 - \text{percentile value of } \tau \tag{5.11}$$

Furthermore,

$$\Pr(f(x_n) \neq \pm\tau|\tau) = 1 - \Pr(f(x_n) = \pm\tau|\tau) \tag{5.12}$$

The last term to define is the conditional probability of the observed sample, y_n, given that the underlying noise-free sample is not clipped. Note that $y_n = f(x_n) + w_n$, where both $f(x_n)$ and w_n are random variables. As described in Sec. 5.3.1, the PDF of y_n would be equal to the convolution of the PDF of $f(x_n)$ with the PDF of w_n. Thus, this term requires the estimation of the PDF of $f(x_n)$, which is not directly observable. To avoid the complications involved in this density estimation, it will be assumed that the conditional PDF of y_n given that $f(x_n) \neq \pm\tau$ can be modeled as a Gaussian distribution with zero-mean and variance, σ_y^2, equal to the sample variance of the observed noisy speech waveform. Therefore,

$$\Pr(y_n|f(x_n) \neq \pm\tau, \sigma_y^2, \tau) = \lim_{\epsilon \to 0} \int_{|y_n|-\epsilon}^{|y_n|+\epsilon} \frac{1}{\sqrt{2\pi}\sigma_y} e^{-\frac{t^2}{2\sigma_y^2}} dt \tag{5.13}$$

With these quantities, it is now also possible to compute the posterior probability of a sample of the noise-free signal being unclipped:

$$\Pr(f(x_n) \neq \pm\tau|y_n, \sigma_y^2, \sigma_w^2, \tau) = \frac{\Pr(y_n|f(x_n) \neq \pm\tau, \sigma_y^2, \sigma_w^2, \tau)\Pr(f(x_n) \neq \pm\tau|\tau)}{\Pr(y_n|f(x_n) = \pm\tau, \sigma_w^2, \tau)\Pr(f(x_n) = \pm\tau|\tau) + \Pr(y_n|f(x_n) \neq \pm\tau, \sigma_y^2, \sigma_w^2, \tau)\Pr(f(x_n) \neq \pm\tau|\tau)} \tag{5.14}$$

A given observed signal sample, y_n, can be classified as either "clipped" or "unclipped" according

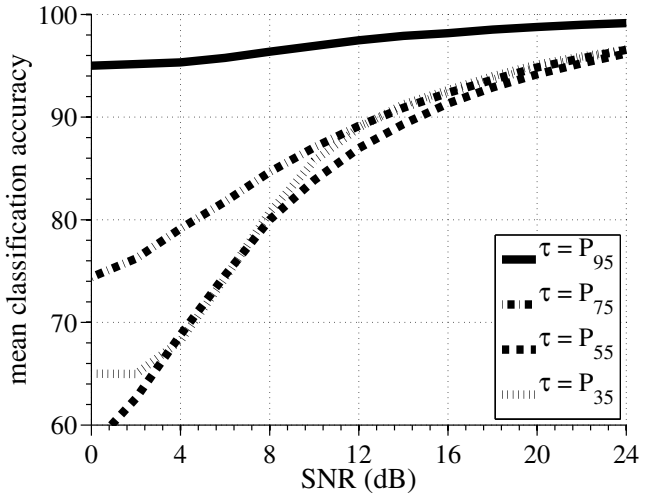


Figure 5.15: Mean classification accuracy for classifying individual noisy signal samples as either clipped or not clipped using the rule in Eq. 5.15.

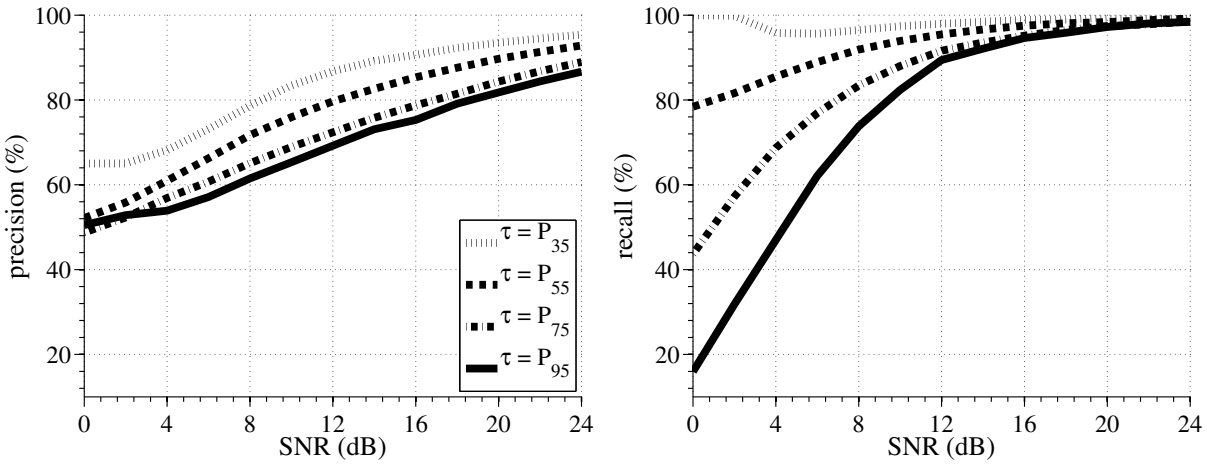
to the optimal Bayesian decision threshold [50] as follows:

$$\text{class of } y_n \;=\; \begin{cases} \text{clipped} & \text{if } \Pr(f(x_n) = \pm\tau | y_n, \sigma_y^2, \sigma_w^2, \tau) \geq \Pr(f(x_n) \neq \pm\tau | y_n, \sigma_y^2, \sigma_w^2, \tau) \\ \text{unclipped} & \text{otherwise} \end{cases} \tag{5.15}$$

Given oracle knowledge of the amplitude and percentile values of τ, the classifier described by Eq. 5.15 produces the mean classification accuracies shown in Fig. 5.15. The classifier yields at least 85% accuracy for all clipping thresholds considered whenever the SNR is greater than or equal to 12 dB. Figure 5.16 shows the precision and recall[3] of the classifier for the same test data as Fig. 5.15. From Fig. 5.16b, the recall is relatively high for SNR greater than 8 dB. This implies that the classifier is detecting most of the samples that are clipped (i.e., low false negative). From Fig. 5.16a, the precision is less variable but slightly lower on average than the recall, implying a slightly higher rate of false positives (classifying samples as clipped that actually are not).

To gain a more thorough understanding of the behavior of the classifier, Fig. 5.17 shows the posterior distributions, $\Pr(f(x_n) = \pm\tau | y_n, \sigma_y^2, \sigma_w^2, \tau)$ and $\Pr(f(x_n) \neq \pm\tau | y_n, \sigma_y^2, \sigma_w^2, \tau)$, which govern the classification rule of Eq. 5.15, plotted as a function of y_n.

[3]In this case, the *precision* equals the percentage of samples deemed clipped that are actually clipped. The *recall* equals the percentage of actually-clipped samples that are found to be clipped.



(a) Precision (b) Recall

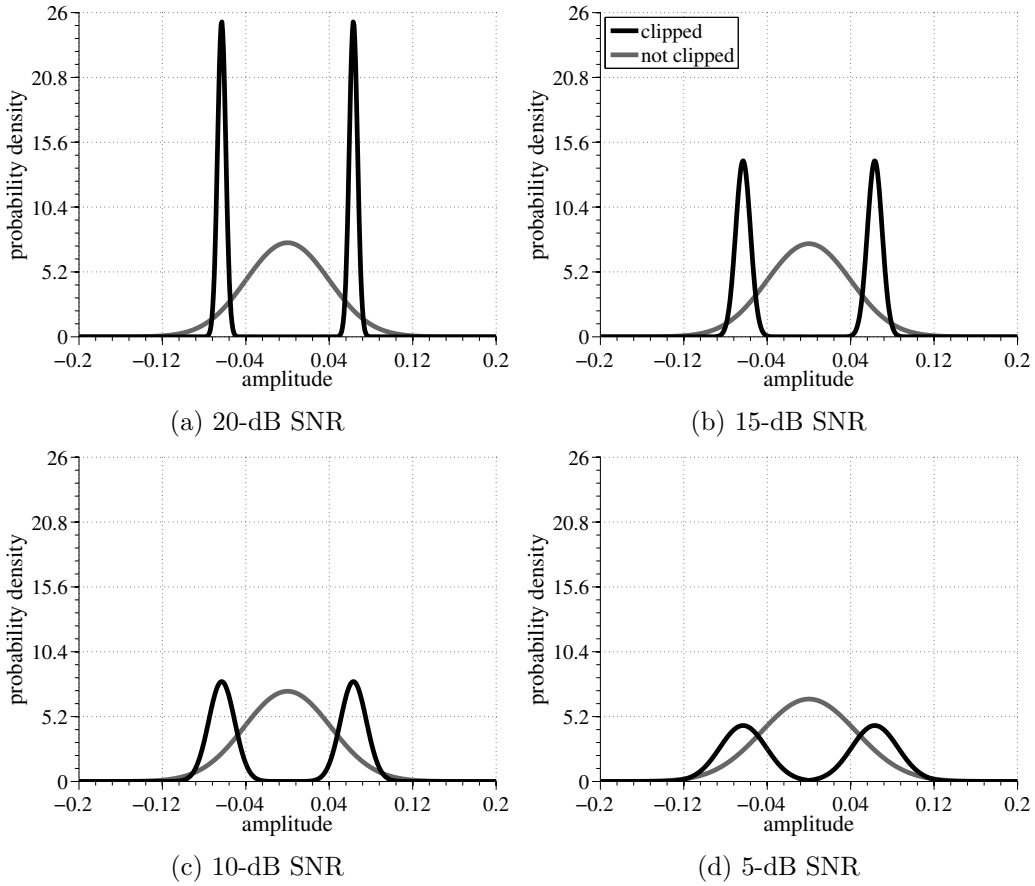
Figure 5.16: Precision and recall of classifying individual noisy samples as either clipped or not clipped using Eq. 5.15, and corresponding to the classification accuracies shown in Fig. 5.15.

5.5 DRC ratio estimation

One method for estimating the ratio, R, of an observed speech waveform is by artificially adding white Gaussian noise to the incoming speech signal (at a fixed, pre-determined SNR), and then measuring the kurtosis of the top and bottom halves of the outer lobes of the noisy waveform amplitude distribution. This method relies on knowledge of the value of τ. The kurtosis of a random variable is defined here as:

$$\gamma = \frac{\mu_4}{\sigma^4} - 3 \tag{5.16}$$

The kurtosis of a Gaussian distribution is zero, and is thus a measure of "Gaussianity." To use this knowledge for estimating R, consider again the artificial addition of white Gaussian noise to compressed speech. Recalling that the sum of two random variables convolves their probability distributions, two "smeared" replicas of the Gaussian distribution will appear at $\pm\tau$. The sharper the peaks around $\pm\tau$, the more the Gaussian distributions will retain their shape in these locations after convolution. Because the sharpness of these peaks is correlated with R, so is the kurtosis of the upper and lower halves of the distributions.



(a) 20-dB SNR

(b) 15-dB SNR

(c) 10-dB SNR

(d) 5-dB SNR

Figure 5.17: Depiction of the posterior probability density functions of Eq. 5.15 as a function of the observed amplitude, y_n.

The steps of the algorithm to measure R are as follows:

1. Add white Gaussian noise, $w[n]$, to the speech signal, $x[n]$, at a given SNR[4] to generate $y[n] = x[n] + w[n]$.

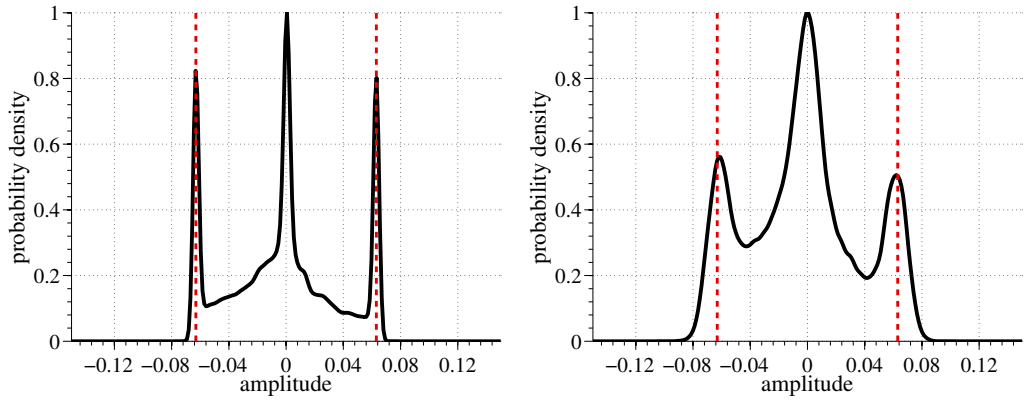
2. Form the subset of samples $z[n] = \{y[n] : |y[n]| \geq \tau\}$.

3. Apply the transformation $g[n] = z[n] - \text{sgn}(z[n]) \cdot \tau$. The $g[n]$ data is referred to as the center-clipped waveform.

4. Measure the kurtosis of $g[n]$ and map to R.

The probability distribution of compressed speech data for $R = \infty$ at each stage of the algorithm is depicted in Fig. 5.18.

[4]A reasonably high SNR, such as 25 dB, is likely adequate.



(a) $x[n]$; clipped speech, no noise

(b) $y[n]$; clipped speech, plus noise

(c) $z[n]$; upper and lower halves of outer lobes

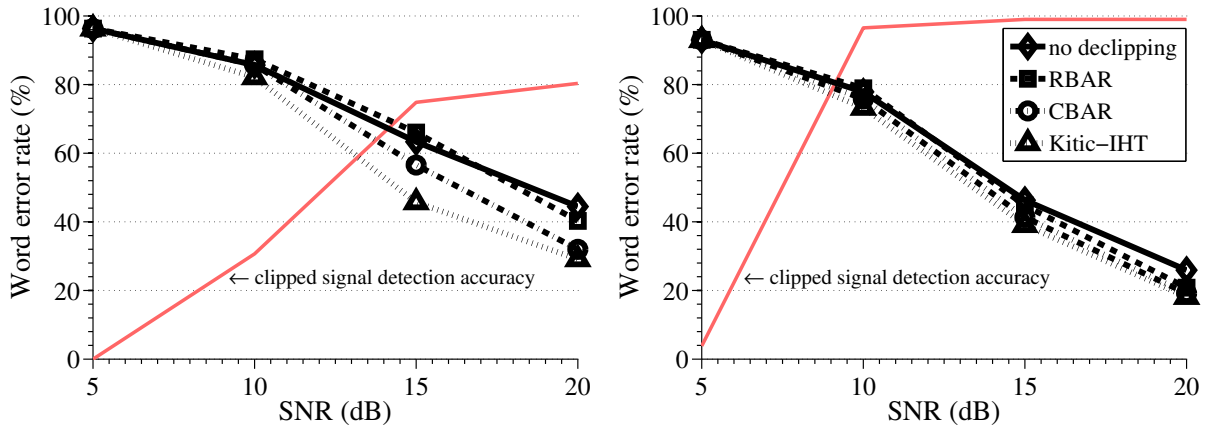
(d) $g[n]$; center-clipped speech whose kurtosis predicts R

Figure 5.18: Each panel depicts the probability distribution of speech data at each stage of the R estimation algorithm. Here, $R = \infty$ and $\tau = \mathrm{P}_{75}$; τ is indicated by the dashed vertical red lines.

5.6 ASR performance

The real measure of the efficacy of the RED algorithms is the degree to which they contribute to the overall goal: the repair of clipped or compressed speech and the subsequent improvement in ASR performance. Figure 5.19 shows the WER as a function of additive-noise SNR for clipped speech repaired using each of the three declipping algorithms (Kitic-IHT, CBAR, and RBAR) introduced in Ch. 4. This figure is directly comparable to Fig. 4.20 except that, in this case, the presence of clipping, the amplitude and percentile values of τ, and the estimation of which samples are clipped is done blindly with no oracle information.

The lack of oracle information about the clipping parameters evidently reduces the benefit of the declipping algorithms. While it has been shown in this chapter that the individual τ estimation and clipped sample estimation algorithms are relatively accurate on their own, the initial detection



(a) $\tau = \mathrm{P}_{75}$ (b) $\tau = \mathrm{P}_{95}$

Figure 5.19: Results of declipping in noise using the indicated algorithm, plotted as a function of SNR. Here, no information about the incoming signal is assumed known. Whether or not the signal is clipped, the amplitude and percentile values of τ, and the estimate of which samples are clipped all are inferred blindly according to the algorithms in this chapter. These plots can be directly compared to Fig. 4.20, for which oracle knowledge of which signals and samples are clipped is given. The light red lines in the plots reflect the clipped signal detection accuracy, i.e., the percentage of signals detected to contain any amount of clipping according to Eq. 5.5. The clipped signal detection accuracy is hypothesized to be the main contributing factors to the performance differential between the oracle-knowledge and no-oracle-knowledge situations.

of the presence of clipping in noise is the biggest factor contributing to this decrease in performance. In addition to WER, Fig. 5.19 shows the clipped signal detection accuracy, which is the percentage of utterances detected to contain any degree of clipping. This clipped signal detection is achieved implicitly with the τ amplitude estimation algorithm from Sec. 5.3.1 (i.e., a signal is deemed not clipped when Eq. 5.5 diverges to ∞). In both Figs. 5.19a and 5.19b, the clipped signal detection accuracy for SNR = 5 dB is 0.0%. This means that none of the signals are even attempted to be declipped at 5-dB SNR. Even if the signals were known to be clipped, there would be no way of detecting the amplitude value of τ with the current algorithms. This is an important area of future research.

5.7 Summary

This chapter introduced and evaluated a novel technique for blindly inferring the value of τ from observed speech that had been subjected to DRC and independent additive white Gaussian noise. The peak-finding algorithm was thoroughly evaluated and shown to be effective in the presence of

noise down to a signal-to-noise ratio of 10 dB. A second novel algorithm for inferring the percentile value corresponding to a particular amplitude value of τ and an observed speech waveform was introduced. Finally, an accurate and robust classification algorithm for detecting clipping in noisy speech was discussed and evaluated.

The algorithms' composite effect on ASR performance was found to be generally helpful. The largest observed benefit of 35% relative occurred on speech clipped at $\tau = \mathrm{P}_{75}$ at SNR = 20 dB when the RED techniques were paired with the Kitic-IHT declipping algoirthm. At very low SNR and τ, no ASR improvements were observed. This substantial performance differential between oracle-knowledge and no-oracle-knowledge situations in low-SNR, low-τ conditions was attributed to the difficulty of detecting the presence of clipping and subsequently inferring the value of τ.

Chapter 6

Artificially-Matched Training Techniques (AMT)

6.1 Introduction

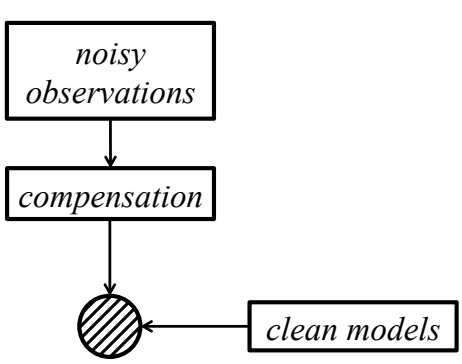
Chapters 3 and 4 considered techniques for explicitly restoring the quality of audio signals subjected to invertible or noninvertible dynamic range compression, respectively. Because the focus of this research is to improve the performance of automatic speech recognition systems, however, such generalizable techinques are not always necessary. This chapter describes two approaches to designing a robust ASR system that are fundamentally different from BAN and BAR. To a large extent, the accuracy of a speech recognizer is a function of the degree to which the acoustic models of phonemes correctly characterize the observed instances of those phonemes. Any introduction of noise, as discussed in Section 2.1.5, skews the statistical properties of the observations, thus making the models less accurate. Robustness can be achieved through any means of reducing that mismatch.

6.2 Classifying approaches to robustness

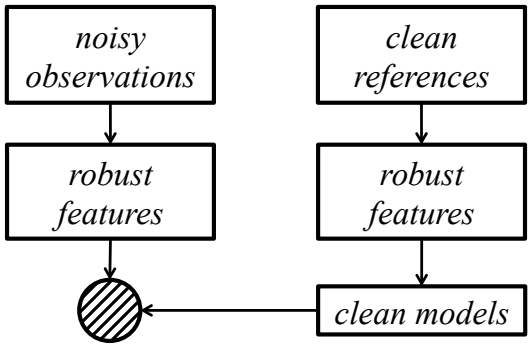
Given some noisy and distorted speech to decode, there are three ways to reduce acoustic model (AcM) mismatching: (1) clean the noisy observations (C-I), (2) "meet in the middle" by representing the training and testing speech data using only qualities of the speech signal that are relatively invariant to the noise and distortion (C-II), or (3) use an AcM trained on data that represents the observed distortion, i.e., "matched" training (C-III). These categories are graphically illustrated in Fig. 6.1 and are useful for conceptualizing robustness algorithms. The categories are not mutually exclusive and algorithms from each class can be combined. For example, standard MFCC feature extraction is of class C-II and is often paired with a noise reduction algorithm from class C-I while also being used in a system trained on matched-condition audio, as in class C-III. All novel algorithms discussed in the thesis thus far (e.g., BAN and BAR and their variations) fall into class C-I and constitute cleaning a noisy observation.

6.2.1 Artificially-matched training

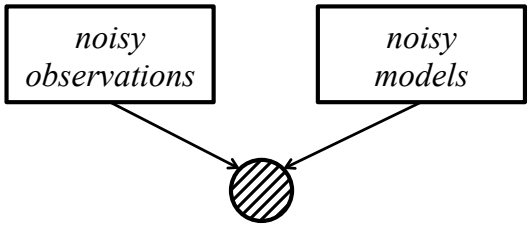
Algorithms from class C-I involve the use of a static AcM derived from clean speech training data, generally with no modification (e.g., BAN and BAR have no effect on clean speech, and noise reduction algorithms like spectral subtraction usually introduce negligible artifacts when no noise



(a) **C-I**: Cleaning noisy observations.



(b) **C-II**: Isolating invariant characteristics of the signals.



(c) **C-III**: Matching the acoustic model to noisy observations.

Figure 6.1: Three distinct approaches to noise-robust speech recognition. The previously-introduced BAN and BAR techniques fall under "cleaning noisy observations" of Fig. 6.1a. Isolating invariant characteristics and matching the acoustic model to noisy observations, i.e., Figs. 6.1b and 6.1c, are collectively referred to as artificially-matched training (AMT) techniques.

is present). In contrast, class C-II and C-III algorithms either implicitly or explicitly modify both the testing and training data (and thus, the AcM) or just the training data in order to reduce the AcM mismatch. Thus, classes C-II and C-III are collectively referred to as *artificially-matched training* (AMT) techniques.

Class C-II: Isolating invariant characteristics of signals

Dynamic range compression, as defined by Eq. 2.12, preserves zero crossings of the input signal, $x[n]$. This immediately suggests that, if a speech feature were designed to extract information from only the zero crossings of $x[n]$, the feature would be entirely invariant to DRC. One simple way of retaining only zero crossing information of a signal is to apply *hard limiting*[1]:

[1]In many audio engineering contexts, the terms hard limiting, limiting, and clipping are used interchangeably to refer to the output of Eq. 2.12 with $R = \infty$. In this thesis, however, hard limiting specifically refers to retaining only the sign of the input signal, as described by Eq. 6.1.

$$HL(x[n]) = \gamma \cdot \text{sgn}(x[n]) \tag{6.1}$$

If $f(x[n]; R, \tau)$ represents Eq. 2.12 for some specific values of R and τ, the following statement is true:

$$HL(x[n]) = HL\left(f(x[n]; R, \tau)\right) \qquad \forall \tau, R \tag{6.2}$$

A naïve approach to DRC-robust feature extraction using hard limiting is illustrated in Fig. 6.2. All input test data is purposefully passed through a hard limiter, from which features are then extracted, and then decoded using an ASR trained on hard-limited data. This system is referred to as *artificially-matched training with hard limiting* (AMT-HL). The utility of AMT-HL is based on whether the particular mismatch between the training and testing data being used outweighs the loss of information incurred by purposefully hard limiting all of the speech. One direction of future research is to consider more intelligent ways of leveraging information contained in the zero crossings.

Despite the marked simplicity of the hard-limiting approach, the ASR performance improvement for compressed speech is remarkable. In the absence of noise, the WER of the standard Sphinx-III RM1 speech recognition experiment[2] using hard-limiting feature extraction is 16.9%. While this represents a 164% increase in the WER relative to clean performance with no hard limiting (6.4%), this WER is completely invariant to DRC and is (often substantially) lower than the WER values achieved on compressed speech using basic MFCC features for $R \geq 4$ and $\tau \leq 75$, by inspection of Fig. 2.9b.

Figure 6.4 shows performance in noise, in accordance with the degradation model of Fig. 2.5b. At first, the results of Fig. 6.4 may be counterintuitive. The WER is, in fact, *decreasing* with *increasing* R and *decreasing* τ. This characteristic occurs, however, not due to the hard limiting of Fig. 6.2, but due to the fact that, as R increases and τ decreases, the underlying compressed speech looks more like hard-limited speech, thus implicitly reducing the mismatch between the training and testing data. Because of the additional layer of additive noise after DRC, the purposeful

[2] As a reminder, the specifics of this standard experiment are described in Section 2.1.4 and the baseline WER on clean speech using MFCC features is 6.4%.

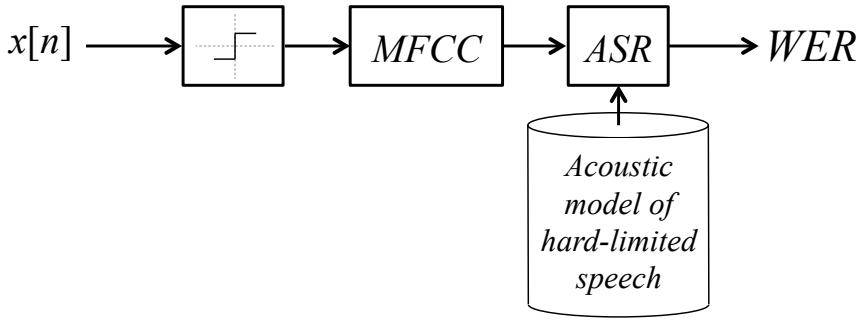


Figure 6.2: Processing flow for extracting features which will be invariant to DRC. The input speech is first purposefully hard limited, and then input to a speech recognizer trained on hard-limited speech. This system is referred to as *artificially-matched training with hard limiting* (AMT-HL).

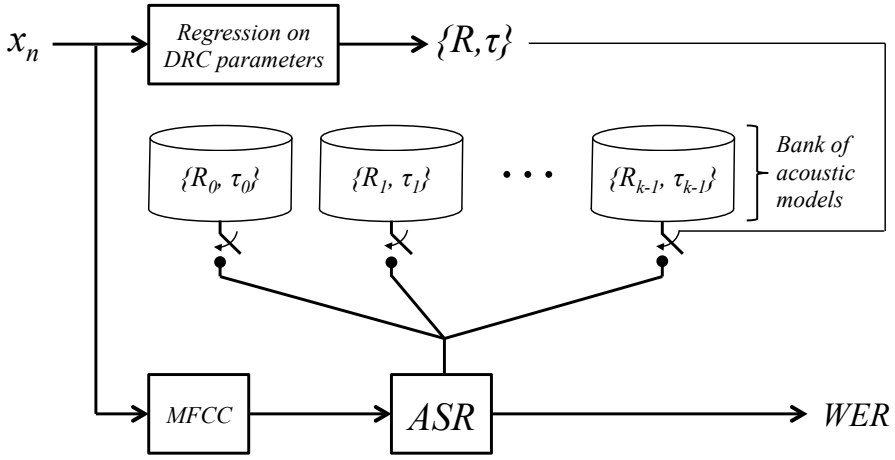
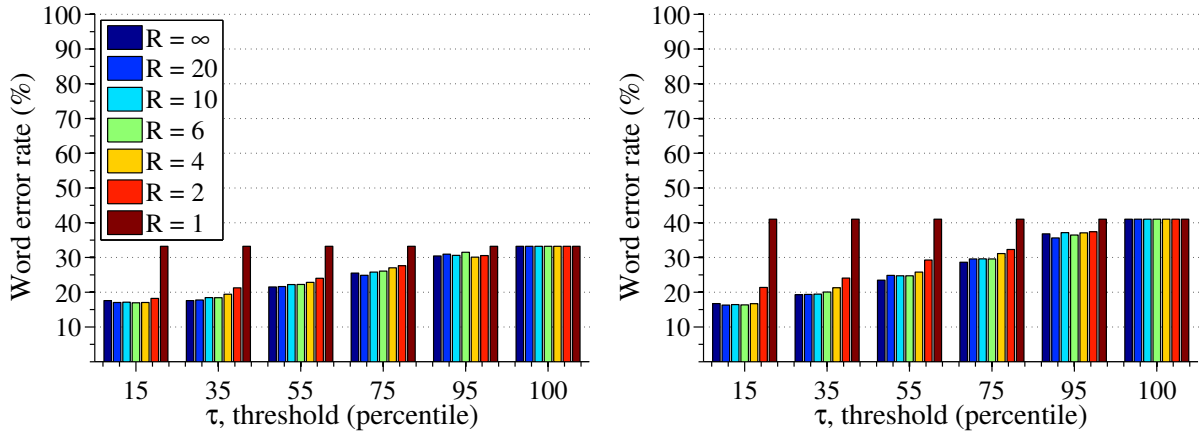


Figure 6.3: Processing flow diagram for a system capable of switching between a set of acoustic models based on the best estimate of the DRC parameters of the input speech. This system is referred to as *artificially-matched training with acoustic model selection* (AMT-AMS).

hard limiting can no longer directly counteract the effects of DRC. Note that this very simplistic approach outperforms BAN in noise under many conditions (particularly large R and small τ).

Class C-III: Matching the acoustic model to noisy observations

The diagram in Fig. 6.3 depicts a system which switches between a set of pre-trained acoustic models based on the best estimate of DRC parameters. This system is referred to as *artificially-matched training with acoustic model selection* (AMT-AMS). For each incoming utterance the values of R and τ are inferred (e.g., using the techniques of Ch. 5) and the acoustic model whose DRC



(a) With AWGN at 20-dB SNR.

(b) With AWGN at 15-dB SNR.

Figure 6.4: WER of the system in Fig. 6.2 as a function of DRC parameters with AWGN at the indicated SNR.

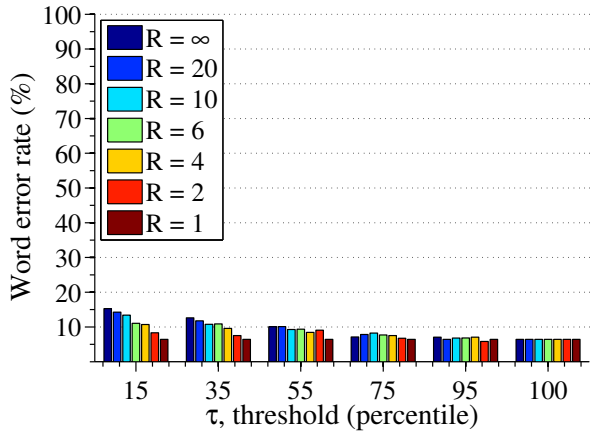
parameters most closely match is chosen. Given infinitesimal granularity in the DRC space and 100% precision when inferring DRC parameters, the WERs shown in Fig. 6.5 are achieved.

In practice, because the algorithm for inferring R from Sec. 5.5 has not been fully developed, the system of Fig. 6.3 is implemented using a set of AcMs with a fixed value of R. The results in Fig. 6.7 illustrate the performance of the system using six separate AcMs trained from data with τ drawn from $\{\mathrm{P}_{15}, \mathrm{P}_{35}, \mathrm{P}_{55}, \mathrm{P}_{75}, \mathrm{P}_{95}\}$ and with $R = \infty$ with an additional AcM trained on clean data.

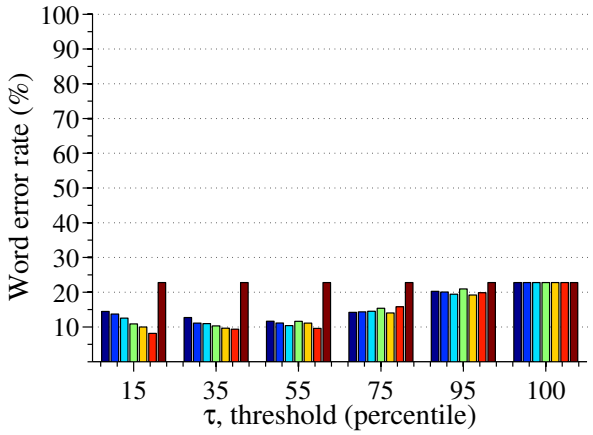
6.3 Interpreting the results

To understand the benefits of the AMT-HL and AMT-AMS systems of Figs. 6.2 and 6.3, their WERs over the range of parameter values considered must be compared to each other as well as the other algorithms presented in this thesis. Generally speaking, the HL system is best suited for cases where DRC or clipping is very extreme. It is most beneficial for $\tau \leq \mathrm{P}_{55}$ and $R \geq 4$ (e.g., when comparing Figs. 6.4 and 6.7).

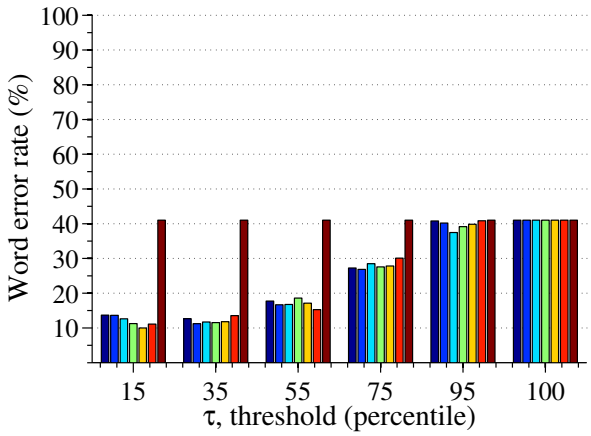
The initial results of Fig. 6.7 show that the AMS system is rarely optimal compared to the host of other algorithms considered in this thesis (e.g., BAN, BAR, and the HL system), but is still generally beneficial, especially for values of $\tau \geq \mathrm{P}_{75}$. In Ch. 7, it will be shown that the AMT-AMS system is particularly useful as a complement to BAR. The subset of poor AMT-AMS



(a) Training perfectly matched to DRC parameters; test data contains no additive noise.



(b) Training perfectly matched to DRC parameters with test data including white noise at 20-dB SNR.

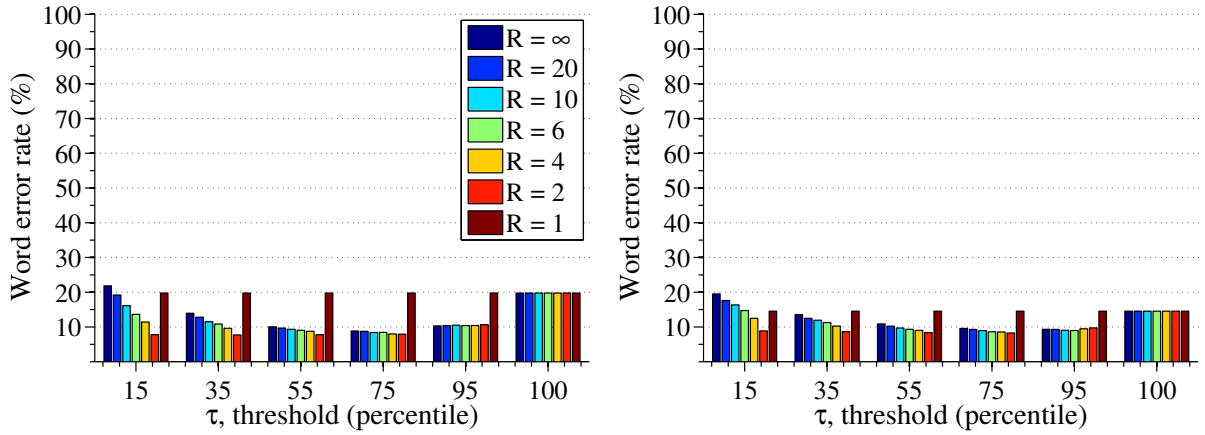


(c) Training perfectly matched to DRC parameters with test data including white noise at 15-dB SNR.

Figure 6.5: WER of the system in Fig. 6.3 with R and τ perfectly estimated from the incoming test data.

results is likely attributable to (1) the fact that a method for estimating the value of R has not been developed, and (2) the τ estimation algorithm of Sec. 5.3.1 often misclassifies compressed or clipped speech as uncompressed when the value of τ is very low (in these situations, the AMS system then incorrectly references a clean AcM when decoding). Future research in the area of robust DRC parameter estimation would likely give rise to substantial improvements in the performance of the AMS system, as evidenced by Fig. 6.5 which shows the best-case performance of AMS with perfect AcM matching.

Figure 6.6 shows the performance of an ASR system when trained on a so-called "multistyle" training set. A multistyle database is a training set composed of a heterogeneous mix of speech



(a) Training data contains even amounts of all test conditions; clean is underrepresented by 23%.

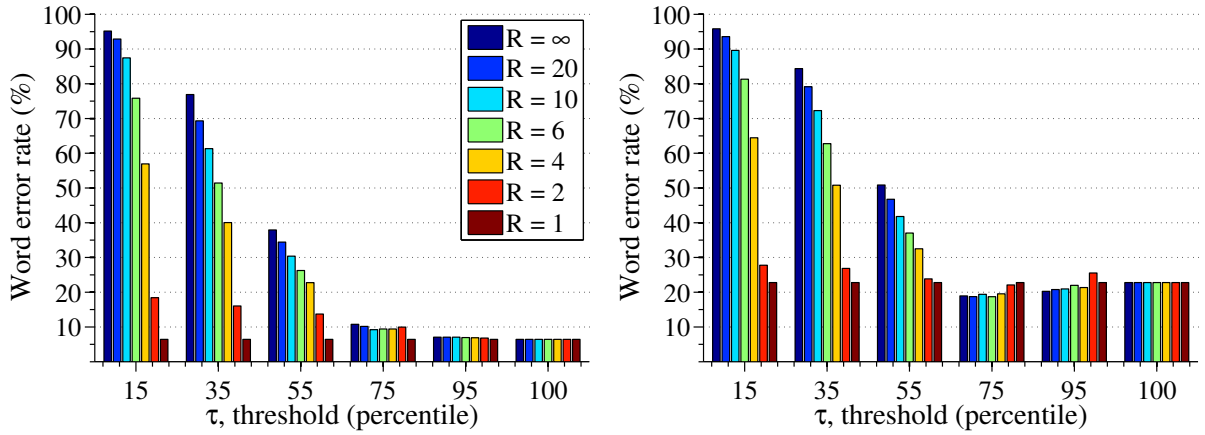
(b) Training data contains even amounts of clipped data conditions; clean is underrepresented by 28%.

Figure 6.6: WER of the ASR system when trained on a heterogeneous mix of data.

containing examples across all expected environmental conditions. In Fig. 6.6a, the training set includes equal amounts of all environmental conditions, including clean, with the clean condition underrepresented by approximately 23% relative to the other conditions. Similarly, Fig. 6.6b shows the results when the training set includes equal amounts of clipped data only, plus clean, and again where clean data is underrepresented by approximately 28%. Note that these results are very good, and indicate that in the absence of accurate parameter estimation algorithms—and with a confident estimate of the expected environmental conditions upon system deployment—performance comparable to the ideal AMT-AMS system can still be achieved.

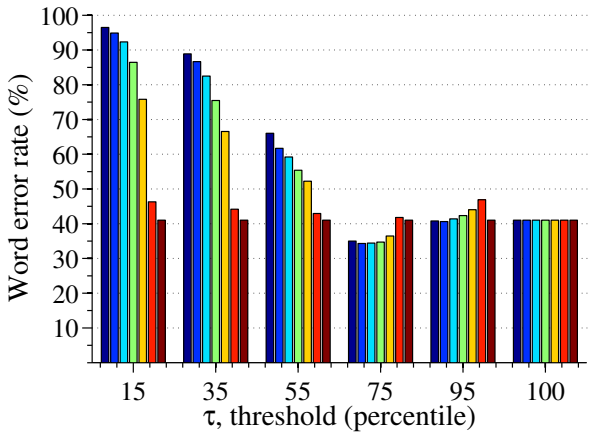
6.4 Summary

This chapter introduced a pair of techniques for DRC-robust speech recognition. Artificially-matched training contrasts with the earlier BAN and BAR algorithms in that, rather than trying to clean a noisy speech observation to "look like" clean speech, one of two methods is used to artificially match the acoustic model reference to the characteristics of the incoming speech features. Artificially-matched training with hard limiting, or AMT-HL, achieves this match by simply hard limiting all of the input speech before feature extraction. Because DRC retains zero crossing information, hard limiting generates a signal that is entirely independent of DRC in the absence of additional noise.



(a) Training matched to the percentile value of τ; test data contains no additive noise.

(b) Training matched to the percentile value of τ with test data including white noise at 20-dB SNR.



(c) Training matched to the percentile value of τ with test data including white noise at 15-dB SNR.

Figure 6.7: WER of the system in Fig. 6.3 with τ blindly inferred using the amplitude and percentile estimation methods of Secs. 5.3.1 and 5.3.2. There are six reference acoustic models: one clean and five corresponding to $\tau = \{\mathrm{P}_{15}, \mathrm{P}_{35}, \mathrm{P}_{55}, \mathrm{P}_{75}, \mathrm{P}_{95}\}$, all with $R = \infty$.

A more sophisticated approach, artificially-matched training with acoustic model selection, or AMT-AMS, utilizes the τ estimation algorithm of Sec. 5.3.1 to select the AcM associated with the closet pair of DRC values to the observed speech. While AMT-AMS is extremely useful in theory (Fig. 6.5), it is of limited use in practice (Fig 6.7) for values of τ less than P_{75} until the DRC parameter estimation techniques can be improved. In particular, the τ estimation algorithm's ability to differentiate between compressed/clipped speech and clean speech must be improved. Furthermore, the R estimation algorithm must be more fully developed and tested. Despite these somewhat lackluster results, it will be shown in Ch. 7 that the AMT-AMS system is of particular practical use on stochastically clipped data and when paired with BAR techniques from Ch. 4.

Chapter 7

A Practical Framework for DRC Compensation

7.1 Introduction

This thesis has thus far considered the problem of dynamic range compression, its effects on the performance of automatic speech recognition systems, and algorithms designed to compensate for any resulting degradation in word recognition accuracy. The DRC function gives rise to two distinct types of distortion−invertible *compression* or non-invertible *clipping*−depending on whether or not the ratio parameter, R, is finite valued. In practical situations, the difficulty of compensating for DRC is often amplified by the presence of independent additive channel noise appearing after the imposition of DRC. The previous chapters of this thesis have considered the DRC sub-problems of compression (Ch. 3), clipping (Ch. 4), and differentiation between the two (Ch. 5) in isolation. This chapter attempts to illustrate how the individual algorithms developed thus far can work synchronously to compensate for the effects of DRC in a practical situation.

7.2 A comprehensive solution to DRC

In a real-world situation, and in the absence of all other information, a usefully-engineered system for DRC compensation should be able to accurately traverse the flowchart in Fig. 7.1. As will be shown, the successful implementation of this system will utilize each of the individual algorithms developed in this thesis, leading to a comprehensive solution to nonlinear DRC distortion.

7.2.1 Is the audio exposed to DRC?

The determination of whether or not audio has been exposed to DRC can be accomplished through the application of the τ-estimation algorithm introduced in Sec. 5.3.1. In theory, if no DRC is present, there will be only one peak in the waveform amplitude probability distribution and Eq. 5.5 should diverge to ∞. In the presence of DRC, the τ estimator theoretically returns the correct value of τ, potentially useful later on, if declipping is necessary.

7.2.2 Is the audio clipped?

Given the presence of DRC, the classification of the audio as either compressed ($R < \infty$) or clipped ($R = \infty$) is required. This task can be accomplished through the application of the R-estimation algorithm introduced in Sec. 5.5. It is worth noting that, here, performance should be somewhat

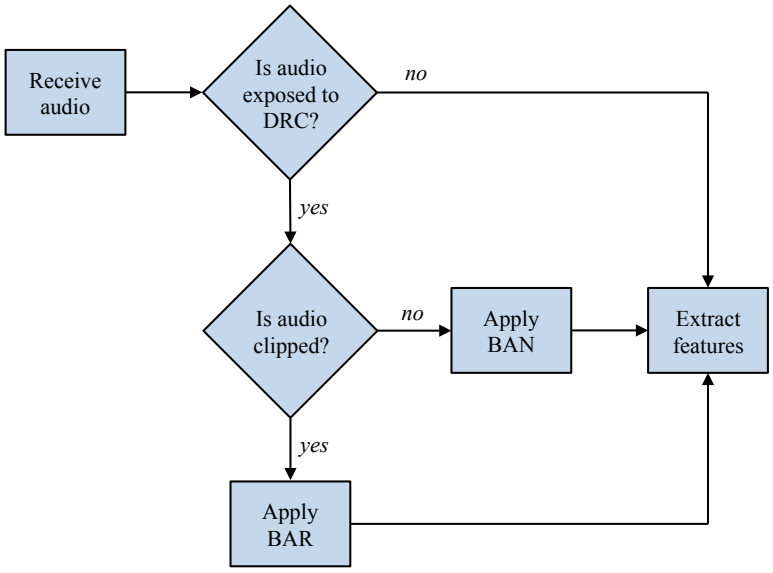


Figure 7.1: Flowchart illustrating a practical system for counteracting the effects of DRC in the absence of additional information about the distortion.

insensitive to the precision of the R estimator as it is only necessary to determine whether or not R is sufficiently large to deem the audio clipped.

7.2.3 Applying BAN

In the case of determining that the audio is not clipped but compressed, straightforward application of some variation of blind amplitude normalization (e.g., BAN, Robust BAN) as described in Ch. 3 is sufficient to counteract DRC. Note that the application of BAN to audio that has not been exposed to DRC is often beneficial or at least ineffectual, therefore, the accuracy of the initial classification of the speech as having been exposed to DRC or not does not have to be exceedingly high.

7.2.4 Applying BAR

The application of blind amplitude reconstruction techniques requires a more sophisticated aggregation of information about the DRC and noise parameters (e.g., τ, its corresponding percentile value, and the variance of any additive noise). The "Apply BAR" block of Fig. 7.1 is thus expanded as shown in Fig. 7.2. A standard voice activity detection (VAD) algorithm, as described in [51],

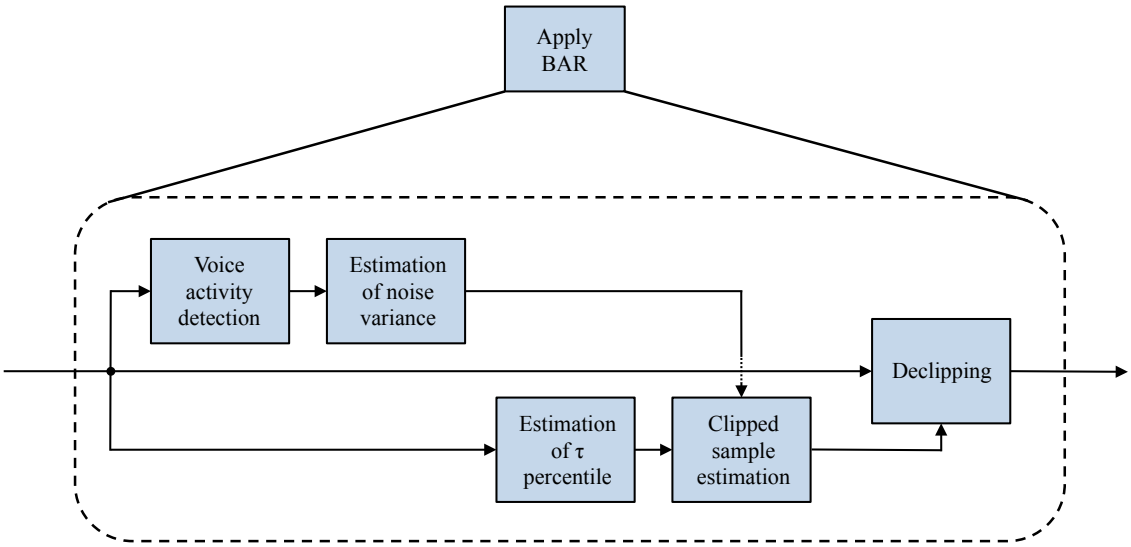


Figure 7.2: Expansion of the "Apply BAR" block from the flowchart in Fig. 7.1.

is applied to the speech to isolate *non-speech* regions, which are then averaged to estimate the long-term noise variance. Independently, the percentile value of τ is estimated from the previously estimated amplitude value of τ. Then, given the estimate of the noise variance and the percentile value of τ, a prediction of which samples have been clipped is made. Finally, any desired declipping algorithm can be applied.

7.3 Simulating real-world conditions

In order to test the practicality of the proposed configurations from Sec. 7.2, a database of distorted, noisy speech that simulates the unpredictability of real-world conditions must be created. As outlined below, this will be achieved by drawing distortion parameters from a uniform probability distribution and compressing speech and injecting noise correspondingly, on an utterance-by-utterance basis.

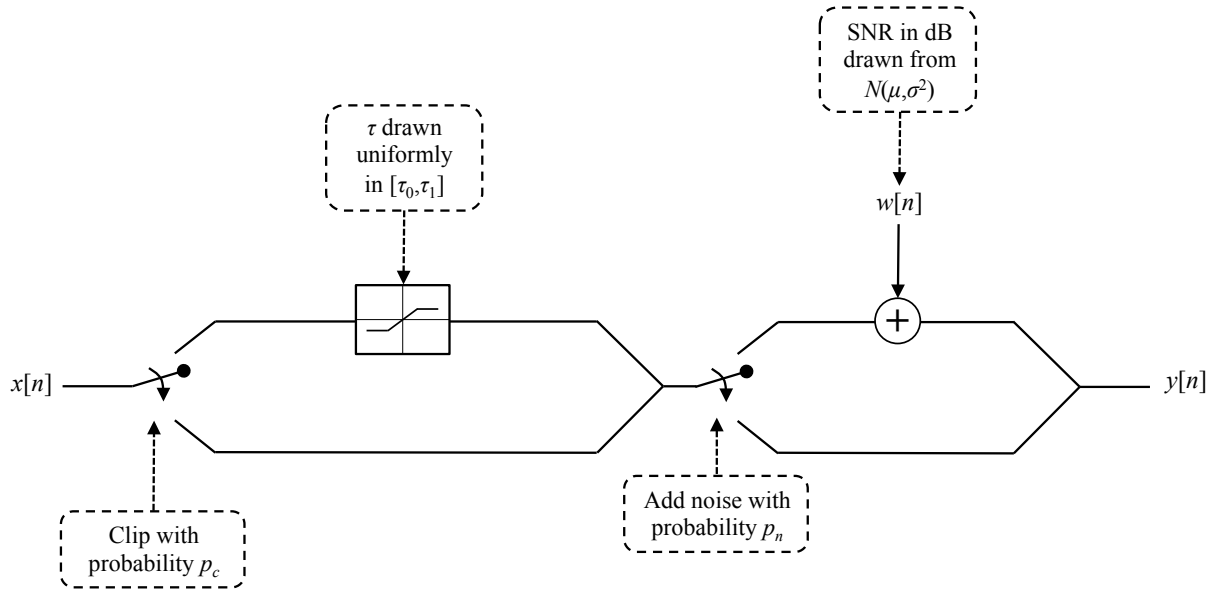


Figure 7.3: Sequence of processing steps for a single audio file when generating a stochastically-clipped database.

7.3.1 Generating stochastically-distorted data

The classification of speech as either *compressed* or *clipped* can be achieved by determining whether or not R is finite valued. Unfortunately, the technology for the prediction of R, as briefly introduced in Sec. 5.5, is not yet developed well enough to warrant sufficiently accurate classification of speech as either clipped or compressed. For this reason, the distinction between clipped and compressed speech will be assumed as oracle knowledge in the following test cases.

Clipped data

Figure 7.3 depicts a proposed processing flow for generating a database containing some clipped speech that simulates a realistic lack of knowledge concerning the intensity of noise, the clipping threshold, or the presence of clipping at all.

Given a clean speech audio file, $x[n]$, the decision of whether or not to clip the audio at all is made with probability p_c, i.e., with probability p_c the audio is clipped and with probability $1 - p_c$ the audio is not clipped. Given that the audio should be clipped, the value of the clipping threshold, τ, is drawn from a uniform distribution bounded by τ_0 and τ_1.

Next, whether or not the audio has been clipped, the decision of whether or not to add noise

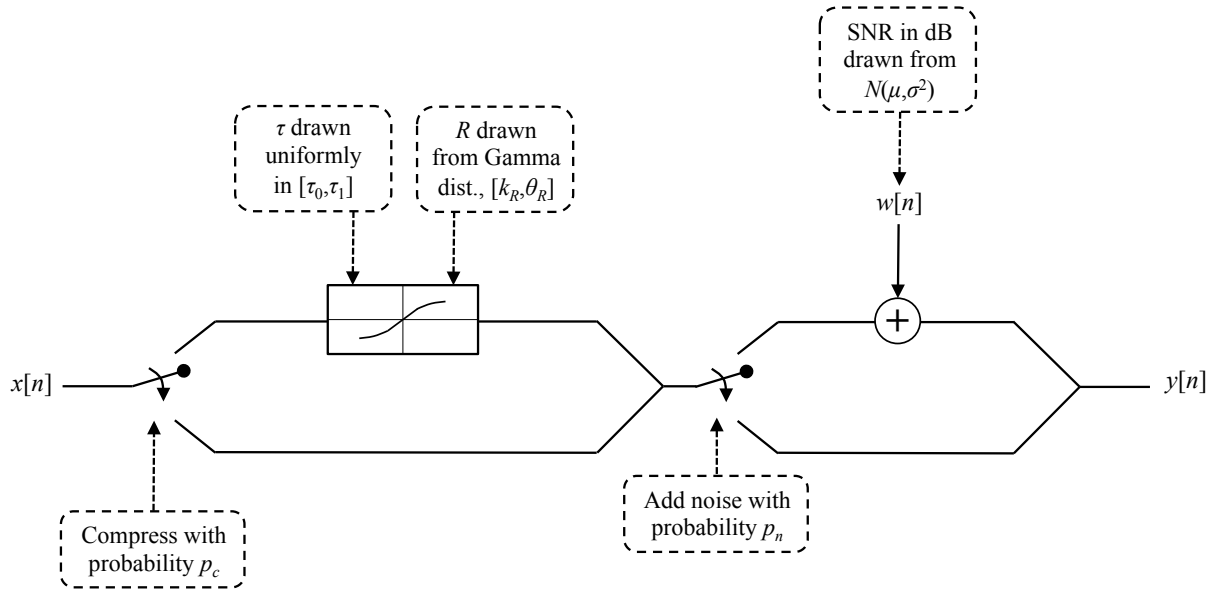


Figure 7.4: Sequence of processing steps for a single audio file when generating a stochastically-compressed database.

to the (possibly) clipped signal is made with probability p_n, i.e., with probability p_n independent AWGN is layered on the signal, and with probability $1 - p_n$ no noise is added to the signal. Given that noise should be added to the audio, the SNR in dB that should be achieved after noise addition is drawn from a Gaussian distribution characterized by mean μ and variance σ^2.

The results reported in this thesis chapter utilize the following parameter values:

$$p_c = 0.9$$

$$\tau_0 = 60$$

$$\tau_1 = 98$$

$$p_n = 0.75$$

$$\mu = 20$$

$$\sigma^2 = 25$$

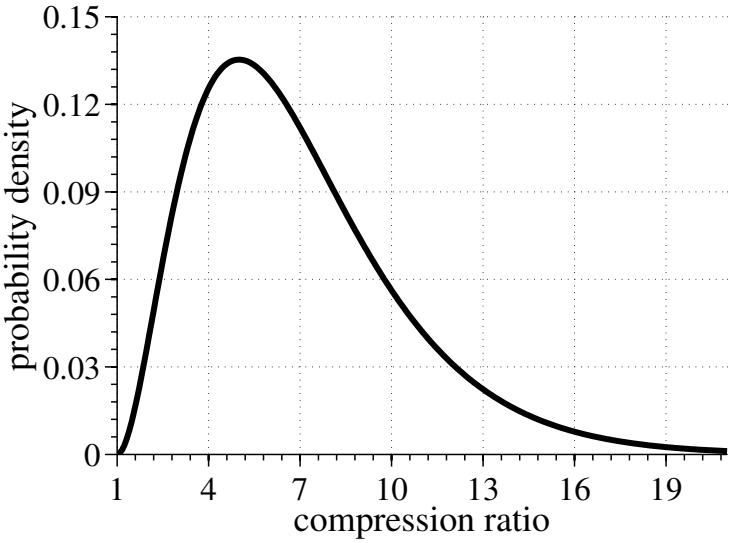


Figure 7.5: Shifted Gamma probability distribution used for randomly specifying the value of R for a given audio file in the processing flow of Fig. 7.4.

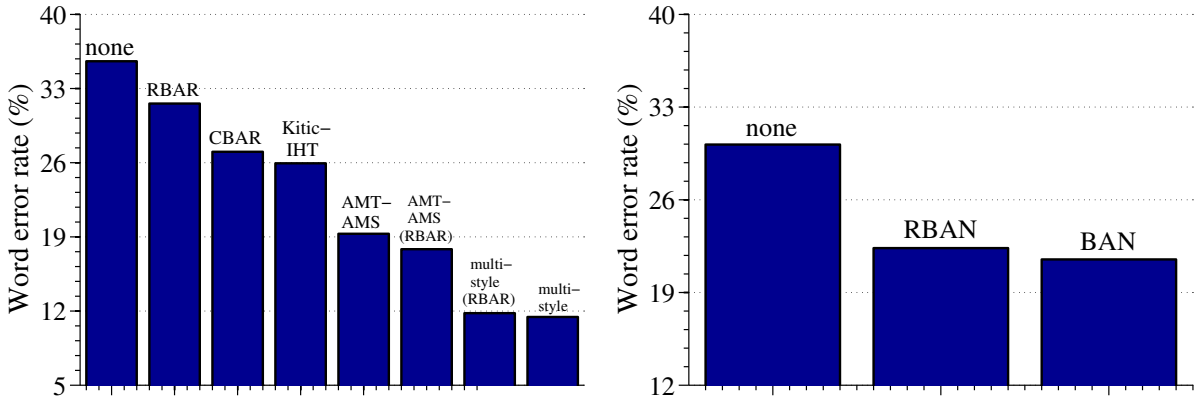
Compressed data

Figure 7.4 depicts a proposed processing flow for generating a database containing some compressed speech that simulates a realistic lack of knowledge concerning the intensity of noise, the threshold, ratio, or the presence of compression at all.

The decisions of whether or not to compress the audio and whether or not to add noise follow the same probabilities as the clipping case shown in Fig. 7.3. The only difference between the compression and clipping case is the need to generate a finite value for R in the compression case. Given that the audio should be compressed, the value of R is drawn from a shifted Gamma distribution [52] with shape parameter, k, equal to 3 and scale parameter, θ, equal to 2. All draws from the distribution are added to 1, effectively shifting the support of the distribution from $(0, \infty)$ to $(1, \infty)$. With these choices of parameter values and shifting, the resulting Gamma distribution has a mean of 7, a mode of 5, and a standard deviation of approximately 3.4. The distribution is shown in Fig. 7.5.

7.3.2 System performance

The performance of the speech recognition system on stochastically-clipped and stochastically-compressed data (according to Figs. 7.3 and 7.4, respectively) is shown in Fig. 7.6. As can be seen



(a) Results of declipping the data generated according to Fig. 7.3 and using the declipping system outlined in Figs. 7.1 and 7.2, where the "Declipping" block is replaced by the indicated algorithm.

(b) Results of decompressing data generated according to Fig. 7.4 using the two types of blind amplitude normalization introduced in Sections 3.2 and 3.3.

Figure 7.6: Word error rate results of declipping (left) and decompressing (right) the stochastically-generated datasets according to Figs. 7.3 and 7.4, respectively.

from Fig. 7.6a, with no prior knowledge of the distortion parameters, the use of the DRC parameter estimation algorithms developed in Ch. 5 can be utilized alongside any of the previously-described declipping algorithms to provide a significant improvement in WER (27% relative with Kitic-IHT, 24% relative with CBAR, and 11.2% relative with RBAR). Note that for any practical application of a declipping algorithm in noise, some form of estimation of the distortion parameters is necessary to inform the declipping algorithm which sections of the signal need to be repaired.

Also from Fig. 7.6a, it can be seen that the AMT-AMS technique from Ch. 6 and described by Fig. 6.3 can also be combined with the RED and BAR techniques. AMT-AMS alone provides a remarkable 46% relative improvement in WER over baseline. When AMT-AMS is paired with RBAR (i.e., acoustic models are trained on clipped speech repaired using RBAR and the incoming speech is also repaired using RBAR before decoding and acoustic model selection), an additional 7.5% relative improvement over AMT-AMS and a total 50% relative improvement over baseline are observed. Pairing the AMT-AMS system with the BAR techniques represents a full integration of the algorithms developed in this thesis.

Figure 7.6a also indicates that the multistyle training method introduced in Ch. 6 outperforms all previously-developed methods, and does not benefit from pairing with RBAR. This does not indicate that the thesis' algorithms are pointless−both AMT-AMS and multistyle training require an accurate anticipation and data representation of the distortion and noise to be encountered,

which is not always possible. Further note that, in comparing Figs. 6.5a and 6.6a, the AMT-AMS system still outperforms multistyle training for various DRC parameter combinations.

The compensation for invertible DRC is similarly effective using both BAN and Robust BAN (cf., Sec. 3.2, 3.3), as illustrated in Fig. 7.6b. BAN and RBAN are comparable in performance; BAN yields a 28.8% relative improvement over baseline, and Robust BAN yields a 26% relative improvement. As observed in the results of Sec. 3.4, Robust BAN would likely outperform BAN for values of R greater in magnitude. When R is of large finite value, the inverse DRC function is more sensitive to additive noise and the benefits of Robust BAN are more pronounced. In this experiment, however, the values of R are drawn from the shifted Gamma distribution in Fig. 7.5 which comprises values of R that are relatively "small" so as to sufficiently distinguish this database from that of the stochastically-clipped data generated according to Fig. 7.3.

Note that, unlike the case of declipping, no explicit DRC parameter estimation is required for decompression using BAN, or variations thereof, because it generates a nonparametric estimate of the (inverse) nonlinearity. Despite this, the relative improvements observed with the declipping case are just as significant as those observed with decompression owing to the accuracy and utility of the parameter estimation methods of Ch. 5.

7.4 Summary

This chapter provided a concise summary of the way in which the BAN, BAR, and RED methods of Chapters 3, 4, and 5 respectively, can be combined in a complementary framework. The generation of the clipped and compressed databases with stochastically-generated parameters intended to demonstrate that the framework would be useful in a practical situation. The illustrated performance gains for both clipping and compression are substantial and promising. The only missing link is the combination of BAN with the declipping configuration of Fig. 7.2, which relies on the ability to accurately differentiate between clipped and compressed speech (i.e., answering the question "Is audio clipped?" in Fig. 7.1). One way of accomplishing this task is through blind inference of the value of R, which may be possible with further development of the basic idea presented in Sec. 5.5.

Chapter 8

Summary and Conclusions

This thesis comprises the introduction of a previously untapped problem in robust speech recognition: dynamic range compression. Dynamic range compression, or DRC, is a common type of distortion that is often a vital component in telecommunications systems. Because telecommunications systems and protocols limit the acceptable dynamic range of the signals they transmit, it is often necessary to nonlinearly "squash" the signal's amplitude so as to adhere to a given system's dynamic range limitations while maintaining adequate signal power. Because of the ubiquity of DRC in telecommunications and audio engineering, it is inevitable that speech systems like ASR will be exposed to dynamic range compressed speech at some point, and as illustrated in this thesis, the effects of DRC can be extremely detrimental to ASR performance. For these reasons, among others, the development of algorithms to blindly detect, infer the parameters of, and compensate for DRC are well motivated.

A phenomenon related to, and special case of, dynamic range compression is audio clipping. From the standpoint of distortion intensity and reversibility, clipping is the most insidious form of DRC, and occurs when the DRC function of Eq. 2.12 is no longer mathematically invertible ($R = \infty$). Clipping often occurs in one of three ways: (1) during signal capture, as a result of exceeding the dynamic range limitations of an A/D converter, (2) during signal write, as a result of not properly amplitude normalizing the audio data before writing to a file, or (3) on purpose, to achieve some desirable perceptual characteristic (e.g., as with the mastering of music). In most cases, audio clipping is perceptually undesirable and essentially always detrimental to ASR performance.

The thesis treats the problem in four complementary distinctions: (1) *blind amplitude normalization* (BAN) methods for counteracting the effects of compressive DRC associated with $R < \infty$, (2) *blind amplitude reconstruction* (BAR) methods, i.e., declipping algorithms, for reconstructing a signal that has been clipped through noninvertible DRC where $R = \infty$, (3) *robust estimation of nonlinear distortion* (RED) comprising a set of algorithms for detecting the presence of DRC, and then subsequently determining the values of DRC parameters, R and τ, with as much insensitivity to noise as possible, and (4) *artificially-matched training* (AMT) methods which attempt to match the ASR AcM to the incoming observation either through isolating invariant characteristics of the speech signal (e.g., zero crossings) with AMT-HL, or by selecting from a set of pre-trained AcMs, as with AMT-AMS. While BAN, BAR, and AMT are basically independent of one another, both BAR

and AMT heavily rely on RED techniques for use in a practical setting. These interdependencies are illustrated by Fig. 2.7.

In Ch. 3, the blind amplitude normalization techniques attempt to undo the effects of invertible DRC by eliciting a non-parametric estimate of the inverse DRC function through comparison of the CDF of the observed speech to a reference CDF obtained over an unadulterated clean speech dataset. While BAN works extremely well in the absence of additive noise, its utility decreases with the intensity of the additive noise. This motivated the development of a variant algorithm entitled Robust BAN. Robust BAN effectively infers the non-parametric inverse DRC function estimate from a low-pass filtered version of the noisy signal, which reduces the impact of additive noise on the estimate. It was observed that Robust BAN is particularly useful for speech that has been exposed to DRC that is nearly saturated (e.g., $R \geq 10$). It was also demonstrated that combining the BAN methods with noise reduction techniques such as spectral subtraction has the potential to substantially improve decompression performance.

Chapter 4 comprises the development of two novel declipping algorithms, *Constrained Blind Amplitude Reconstruction* (CBAR) and *Regularized Blind Amplitude Reconstruction* (RBAR). These time-domain-based algorithms achieve declipping through least squares minimization of the energy of the second derivative of the reconstructed signal. At typical speech sampling rates like 16 kHz, unconstrained minimization of the energy of the derivative of a signal produces an illegitimate interpolation that falls below $|\tau|$, as depicted in Fig. 4.5d. CBAR addresses this issue by imposing a hard constraint on the reconstruction such that the solution agrees with the sign of the observed signal and is greater than $|\tau|$.

While CBAR's declipping performance is state-of-the-art, the imposition of a hard constraint on the least squares minimization causes the algorithm to be highly computationally complex and leads to very slow processing times. At the (slight) expense of declipping quality, RBAR dramatically improves processing speed by removing the hard constraint, using regularization instead to "encourage" (but not force) the solution to lie in a legitimate range. The regularizing terms in RBAR minimize the difference between the solution vector and a constant target vector that floats above $|\tau|$ at some predicted target amplitude. The current version of RBAR produces target amplitude predictions as a function of the fraction of clipped samples in the speech frame being processed.

Chapter 5 considers the tasks of blindly determining if a given speech utterance is clipped, compressed, or unadulterated. In the former two cases, algorithms are developed to automatically infer the amplitude and percentile values of τ, and in the case of clipping specifically, a classification algorithm for differentiating between clipped and unclipped speech samples is developed. This is a crucial precursor to the use of declipping algorithms in practice.

Chapter 6 presents a unique approach to robust ASR, which attempts to match the reference AcM to the potentially noisy observations, rather than repair the noisy observations to look like clean speech. Two approaches to AcM matching are evaulated: AMT-HL and AMT-AMS. The former isolates the zero crossings of the input speech, which are invariant to DRC in the absence of noise, the latter utilizes the techniques of Ch. 5 to choose the closest matching AcM from a discrete set. It is shown that AMT-HL is quite effective for highly compressed or clipped utterances. While AMT-AMS does not perform optimally in practice, its theoretical best-case performance is superb, and it is shown to be of complementary benefit in Ch. 7.

The thesis concludes with the work of Ch. 7, which presents a framework for using the algorithms of Chapters 3, 4, 5, and 6 to achieve a comprehensive solution to DRC for ASR. Each of the individual algorithms are intelligently combined, and the total system is demonstrated to work on a stochastically-generated database. The one missing link of the comprehensive solution proposed in Fig. 7.1 is the ability to answer the question "Is audio clipped?". In theory, this relies on a determination of the value of R, the algorithm for which was introduced in Sec. 5.5 but not sufficiently developed. Nonetheless, considering BAN and BAR separately, and including AMT-AMS, substantial performance improvements are observed on the stochastically-degraded databases, generated according to Figs. 7.3 and 7.4.

In conclusion, this thesis has treated the largely unconsidered problem of dynamic range compression and clipping in robust speech recognition. The mathematical framework for DRC is borrowed from audio engineering and is thus widely applicable. The DRC problem was shown to be radically damaging to ASR performance, and a number of novel algorithms were developed to treat the different manifestations of DRC. Beyond the development of individual algorithms, a practical framework for incorporating these algorithms into a useful whole was developed, and a number of promising future research directions have been suggested.

Bibliography

[1] G. Moore, "Cramming more components onto integrated circuits," *Electronics Magazine*, 1965.

[2] T. Vintsyuk, "Speech discrimination by dynamic programming," *Kibernetika*, vol. 4, pp. 81–88, 1968.

[3] L. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *The Annals of Mathematical Statistics*, vol. 37, no. 6, pp. 1554–1563, 1966.

[4] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm and System Development.* Prentice Hall, 2001, ch. 2, p. 38.

[5] D. Jurafsky and J. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*, 1st ed. Prentice-Hall, 2000, ch. 6.

[6] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[7] D. Deterding, "The Formants of Monophthong Vowels in Standard Southern British English Pronunciation," *Journal of the International Phonetic Association*, vol. 27, pp. 47–55, 1997.

[8] A. Oppenheim and R. Schafer, *Discrete-time Signal Processing*, 3rd ed. Prentice Hall, 2010, ch. 8.

[9] C. Bishop, *Pattern Recognition and Machine Learning.* Springer, 2006, ch. 1, pp. 33–38.

[10] D. O'Shaughnessy, *Speech communication: human and machine.* Addison-Wesley, 1987, p. 150.

[11] C. Kim and R. M. Stern, "Power-normalized cepstral coefficients (PNCC) for robust speech recognition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Kyoto, Japan, March 2012.

[12] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *The Journal of the Acoustical Society of America*, vol. 87, p. 1738, 1990.

[13] A. Oppenheim and R. Schafer, *Discrete-time Signal Processing*, 3rd ed. Prentice Hall, 2010, ch. 8, pp. 673–679.

[14] ——, *Discrete-time Signal Processing*, 3rd ed. Prentice Hall, 2010, ch. 8, pp. 679–682.

[15] ——, *Discrete-time Signal Processing*, 3rd ed. Prentice Hall, 2010, ch. 13, pp. 981–982.

[16] C. S. S. Consortium, "CMU sphinx open source toolkit for speech recognition," http://cmusphinx.sourceforge.net/wiki/download/.

[17] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm and System Development.* Prentice Hall, 2001, ch. 10, pp. 517–519.

[18] A. Acero, "Acoustical and environmental robustness in automatic speech recognition," Ph.D. dissertation, Carnegie Mellon University, 1990.

[19] P. Moreno, B. Raj, and R. Stern, "A vector taylor series approach for environment-independent speech recognition," in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, vol. 2, May 1996, pp. 733–736 vol. 2.

[20] J. Droppo and A. Acero, "Environmental robustness," in *Springer Handbook of Speech Processing*, J. Benesty, M. M. Sondhi, and Y. Huang, Eds. Springer, 2008, ch. 33, pp. 658–659.

[21] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 27, no. 2, pp. 113–120, Apr 1979.

[22] M. Berouti, R. Schwartz, and J. Makhoul, "Enhancement of speech corrupted by acoustic noise," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '79.*, vol. 4, Apr 1979, pp. 208–211.

[23] J. Droppo and A. Acero, "Environmental robustness," in *Springer Handbook of Speech Processing*, J. Benesty, M. M. Sondhi, and Y. Huang, Eds. Springer, 2008, ch. 33, p. 665.

[24] R. Martin, "Noise power spectral density estimation based on optimal smoothing and minimum statistics," *Speech and Audio Processing, IEEE Transactions on*, vol. 9, no. 5, pp. 504–512, Jul 2001.

[25] P. Moreno, "Speech recognition in noisy environments," Ph.D. dissertation, Carnegie Mellon University, 1996.

[26] M. Harvilla and R. Stern, "Histogram-based subband power warping and spectral averaging for robust speech recognition under matched and multistyle training," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, March 2012, pp. 4697–4700.

[27] J. Follansbee, *Hands-on guide to streaming media: an introduction to delivering on-demand media.* Upper Saddle Ridge, New Jersey: Focal Press, 2006.

[28] M. Senior, *Mixing secrets for the small studio.* Burlington, Massachusetts: Elsevier/Focal Press, 2011.

[29] D. Giannoulis, M. Massberg, and J. Reiss, "Digital dynamic range compressor design-a tutorial and analysis," *Journal of the Audio Engineering Society*, pp. 399–408, July 2012.

[30] L. R. Rabiner and S. R.W., *Digital Processing of Speech Signals.* Prentice Hall, 1978, ch. 8, pp. 396–455.

[31] A. Janssen, R. Veldhuis, and L. Vries, "Adaptive interpolation of discrete-time signals that can be modeled as autoregressive processes," *IEEE Trans. on Acoust., Speech and Signal Processing*, pp. 317–330, April 1986.

[32] A. Dahimene, M. Noureddine, and A. Azrar, "A simple algorithm for the restoration of clipped speech signal," in *Informatica*, 2008, pp. 183–188.

[33] W. Fong and S. Godsill, "Monte carlo smoothing for nonlinearly distorted signals," in *IEEE Int. Conf. on Acoust., Speech and Signal Processing*, May 2001.

[34] J. Abel and J. Smith, "Restoring a clipped signal," in *IEEE Int. Conf. on Acoust., Speech and Signal Processing*, April 1991.

[35] A. Adler, V. Emiya, M. Jafari, M. Elad, R. Gribonval, and M. Plumbley, "Audio Inpainting," *IEEE Trans. on Acoust., Speech and Signal Processing*, pp. 922–932, April 2012.

[36] S. Kitic, L. Jacques, N. Madhu, M. Hopwood, A. Spriet, and C. D. Vleeschouwer, "Consistent iterative hard thresholding for signal declipping," in *IEEE Int. Conf. on Acoust., Speech and Signal Processing*, May 2013.

[37] S. Miura, H. Nakajima, S. Miyabe, S. Makino, T. Yamada, and K. Nakadai, "Restoration of clipped audio signal using recursive vector projection," in *TENCON*, November 2011.

[38] R. Balchandran and R. Mammone, "Non-parametric estimation and correction of non-linear distortion in speech systems," in *IEEE Int. Conf. on Acoust., Speech and Signal Processing*, May 1998.

[39] C. Hoare, "Algorithm 64: Quicksort," *Communications of the ACM*, vol. 4, no. 7, pp. 10–15.

[40] R. Schafer, "Homomorphic systems and cepstrum analysis of speech," in *Springer Handbook of Speech Processing*, J. Benesty, M. M. Sondhi, and Y. Huang, Eds. Springer, 2008, ch. 9, p. 166.

[41] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, vol. 39, no. 1, pp. 1–38, 1977.

[42] "Least squares with examples in signal processing," March 2013. [Online]. Available: http://eeweb.poly.edu/iselesni/lecture_notes/least_squares/least_squares_SP.pdf

[43] G. Strang, *Linear Algebra and its Applications*, 4th ed. Cengage Learning, 2005.

[44] T. Blumensath and M. Davies, "Iterative thresholding for sparse approximations," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 629–654, 2008.

[45] *Constrained Nonlinear Optimization Algorithms*, The MathWorks, Natick, MA, 2014. [Online]. Available: http://www.mathworks.com/help/optim/ug/constrained-nonlinear-optimization-algorithms.html

[46] A. Noll and M. Schroder, "Short-time "cepstrum" pitch detection," *Journal of the Acoustical Society of America*, vol. 36, no. 5, p. 1030, 1964.

[47] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm and System Development.* Prentice Hall, 2001, ch. 2, p. 26.

[48] A. Papoulis and S. Pillai, *Probability, Random Variables, and Stochastic Processes*, 4th ed. Tata McGraw-Hill, 2002, ch. 6.

[49] A. Drake, *Fundamentals of Applied Probability Theory.* McGraw-Hill, Inc., 1967, ch. 1.

[50] C. Bishop, *Pattern Recognition and Machine Learning.* Springer, 2006, ch. 1.

[51] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection," *Signal Processing Letters, IEEE*, vol. 6, no. 1, pp. 1–3, Jan 1999.

[52] A. Papoulis and S. Pillai, *Probability, Random Variables, and Stochastic Processes*, 4th ed. Tata McGraw-Hill, 2002, ch. 4.