# Audio Restoration: An Investigation of Digital Methods for Click Removal and Hiss Reduction

Joseph Nuzman
jnuzman@gmail.com

March 29, 2004

**Abstract**

This paper investigates methods for restoring, in the digital domain, audio signals that have been corrupted by noise. Experiments are performed to explore existing techniques and point to possible enhancements. For the purposes of click removal, methods based on autoregressive time series modeling are considered. For hiss reduction, short-time spectral attenuation is considered in the form of standard spectral subtraction as well as Ephraim and Malah enhancements. Successful audio restoration of real degraded audio is demonstrated.

## 1 Introduction

Digital noise reduction for audio signals has been an area of investigation since computers became powerful enough to manipulate digital audio in a practical way. The basis for many of the methods described here were motivated by applications in the speech recognition domain. Subsequent enhancements and generalizations have been motivated by the stricter fidelity requirements of commercial and archival restoration of old recordings.

The survey of methods in this paper is motivated by two applications:

**The restoration of a degraded copy of a commercial recording.** An individual may own a copy of an out-of-print commercial recording.

The degradation may be an artifact of the recording (all forms of analog recording exhibit hiss noise), or a result of corruption of fragile media (eg. scratches on a phonograph). When transferring such a recording to a modern digital format, there is an opportunity to reduce these unwanted artifacts. Without access to the undegraded original, an individual needs to make use of audio restoration of the copy.

**The restoration of a personal recording.** An individual may possess a personal recording that represents the only available record of an event. Degradation of the audio signal may be a result of the limitations of commonly available recording media, limitations of inexpensive recording equipment, or errors in the operation of the recording equipment. Audio restoration can help to improve such recordings.

In these two applications, both the nature of the degradations and the quality of restoration desired are very similar to those of commercial and archival restoration. However, the latter applications may suppose an experienced and knowledgeable restoration engineer, who can afford to spend time tuning a method to a particular recording. In contrast, processes for our applications should be as automated as possible, and not require extensive knowledge or tuning.

This paper will focus on effective techniques for the restoration of audio signals degraded by impulsive clicks or by broadband hiss. Processing will be performed off-line, so computational performance of the methods will not be of primary importance. The implementation of the various methods considered will emphasize clarity and correctness over performance wherever practical.

It is intended that a practical system addressing our two applications could be straightforwardly developed from this paper.

The bulk of this paper is divided into two (mostly independent) sections. Section 2 explores the process of click removal, and section 3 investigates hiss reduction.

Please refer to Appendix A for information about accessing the audio samples and program code referenced in this paper.

# 2 Click Removal

The term clicks refers to localized bursts of impulsive noise present in an audio signal. Clicks are commonly caused by particles or scratches on the surface of a phonograph record. They may be observed as quiet distinct ticks, louder pops, or as a crackling sound.

This section investigates time series modeling of audio signals, detection of click noise, and correction of identified clicks. Click removal is demonstrated for both artificial and real corruption, and possible future research directions are suggested.

The evaluations in this section specifically focus on the removal of clicks from 33 RPM long play phonograph records, where clicks are primarily the result of scratches or debris on the medium. That said, the methods used in this section would also likely be effective for removing impulsive noises introduced by other mechanisms.

Throughout this section, we make use of formulas and notation from [8].

## 2.1 Modeling

For all the methods we will explore, we will make use statistical modeling of audio signals to help in the process of click removal. If we can characterize well the desired audio signal, we can hope to distinguish the unwanted noise.

### 2.1.1 Autoregressive modeling

An example of a short sequence from an audio waveform is illustrated in Figure 1. This 44.1 kHz clip is taken from a professionally recorded compact disc track [2] featuring vocals, guitar, bass, dobro, and mandolin.

A fundamental model that we will use is the autoregressive (AR) model. In this model, an audio signal is considered to be the output of a linear time-invariant all-pole filter applied to a white noise process. Each output sample $x_n$ of the output process can be considered to be the weighted sum of a limited number of previous samples plus a single sample $e_n$ from the random white noise process.

$$x_n = \sum_{i=1}^{P} a_i x_{n-i} + e_n \tag{1}$$

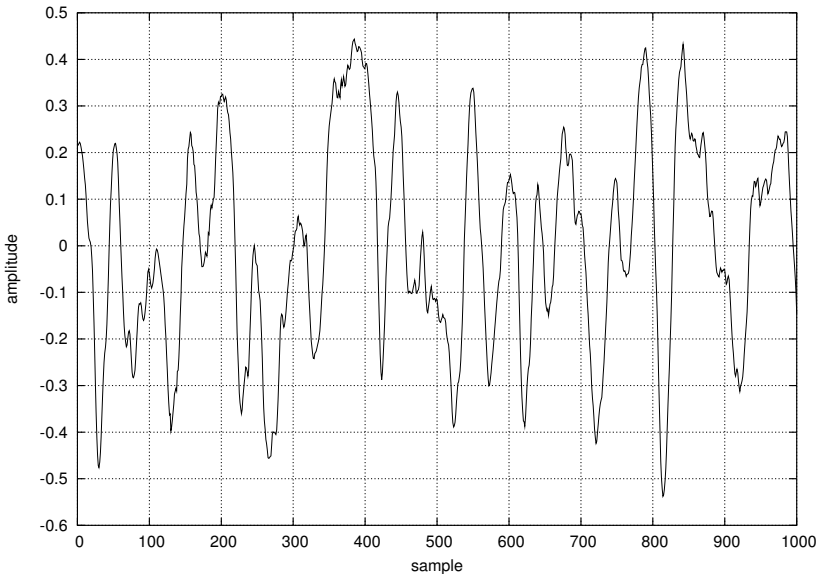


Figure 1: Example audio waveform

Here, $P$ is referred to as the order of the AR process $x$. The $P$ coefficients $a_i$ are referred to as the AR coefficients. The transfer function for the all-pole filter is:

$$H(z) = \frac{1}{A(z)}$$

where

$$A(z) = 1 - \sum_{i=1}^{P} a_i z^{-i}$$

The white noise process is variously called the *innovations* or *excitation* process due to its role in the conceptual synthesis of the AR process. When equation (1) is instead considered as a predictor, the term *prediction error* sequence is often used for the $e_i$.

The choice of the autoregressive model is motivated by two factors:

- The random source with filter model correlates closely to the physical production of many audio signals. The mechanism of a human voice, for example, can be considered random excitations shaped or filtered by the physical characteristics of the speaker.

- Assumptions of a finite-parameter all-pole filter and of Gaussian white excitation allow for straightforward analysis in many cases. More sophisticated models, such as the autoregressive moving-average (ARMA) model, may also be excellent models of audio processes, but can be more difficult to analyze or can present numerical problems. Note that a finite-order AR process can be constructed to approximate any ARMA process arbitrarily well, although possibly using more coefficients.

We will see that the stationary Gaussian AR model can be a very good fit for short blocks of the audio signals considered here.

If we are given a block of $N$ samples and wish to estimate the coefficients $a_i$ of the AR model of order $P$, it is useful to reformulate (1) in matrix notation. We are given the vector $\mathbf{x}$:

$$\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_N \end{bmatrix}^T$$

and wish to estimate the parameter vector $\mathbf{a}$:

$$\mathbf{a} = \begin{bmatrix} a_1 & a_2 & \cdots & a_P \end{bmatrix}^T$$

We form the autoregressive matrix $\mathbf{G}$ from $\mathbf{x}$ as:

$$\mathbf{G} = \begin{bmatrix} x_P & x_{P-1} & \cdots & x_2 & x_1 \\ x_{P+1} & x_P & \cdots & x_3 & x_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{N-2} & x_{N-3} & \cdots & x_{N-P} & x_{N-P-1} \\ x_{N-1} & x_{N-2} & \cdots & x_{N-P+1} & x_{N-P} \end{bmatrix}$$

We use the convention that $\mathbf{x_0}$ is the first $P$ samples of $\mathbf{x}$ and $\mathbf{x_1}$ is the remaining $N - P$ samples, such that:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x_0} \\ \mathbf{x_1} \end{bmatrix}$$

We are now prepared to rewrite (1) as:

$$\mathbf{x_1} = \mathbf{G}\mathbf{a} + \mathbf{e} \tag{2}$$

The innovations vector $\mathbf{e}$ corresponds to:

$$\mathbf{e} = \begin{bmatrix} e_{P+1} & e_{P+2} & \cdots & e_N \end{bmatrix}^T$$

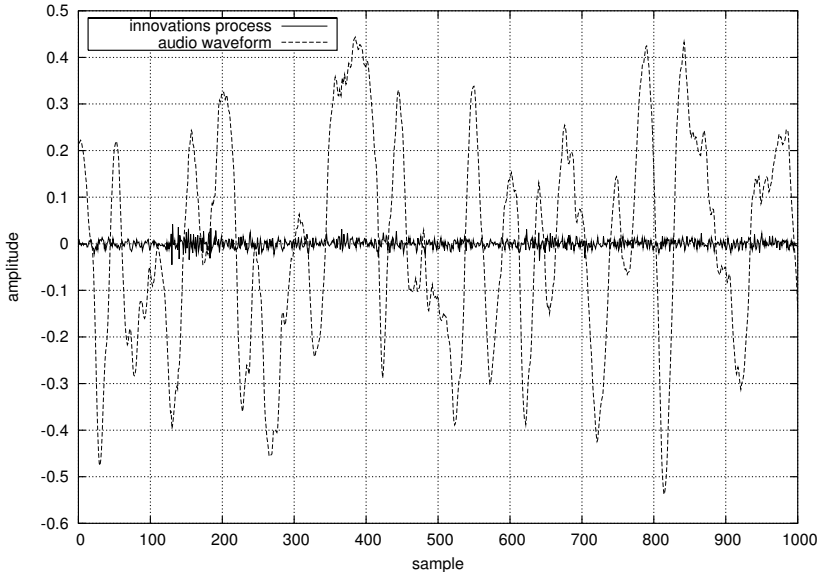


Figure 2: Audio waveform and innovation sequence

An estimate for $\mathbf{a}$ can be produced by minimizing $\mathbf{e}$ in the least squares sense. This estimate is known as the covariance estimate:

$$\mathbf{a} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{x_1}$$

If the innovations process is assumed to be zero-mean Gaussian, it can be shown (see [8]) that this estimate is equivalent to choosing the $\mathbf{a}$ which maximizes the likelihood of $\mathbf{x_1}$ given $\mathbf{a}$ and $\mathbf{x_0}$. If $N >> P$ then this probability is an approximation of the likelihood of the full $\mathbf{x}$ given $\mathbf{a}$. Note that, while the likelihood depends on the variance $\sigma_e^2$ of the excitation sequence, the maximization (and hence the estimator) does not.

To give an idea of how an excitation sequence might compare to the original audio if corresponds to, both are plotted in Figure 2. This uses the same audio clip from [2]. Coefficients for an AR model of order $P = 15$ are estimated using least squares over a window of length 1024. The inverse filter is then applied to produce the excitation sequence shown.

### 2.1.2 Sinusoid extensions

Another possible model of audio signals is as the sum of deterministic bases. This model, where the bases are complex exponentials, is at the heart of Fourier analysis.

For general audio, the AR model tends to better match the inherent randomness of the signal. However, many musical examples exhibit a strong tonal nature. With such signals, a limited-order AR model can miss much of the long-term correlation present. It is often possible to select a few fixed basis vectors (perhaps sinusoids of particular frequencies) that can be linearly combined to model the majority of the signal. The remaining randomness can then be modeled using the AR approach.

This combined AR/sinusoid method can be expressed as:

$$x_n = \sum_{i=1}^{Q} c_i \psi_i[n] + r_n \tag{3}$$

where $Q$ is the number of basis vectors and $\psi_i[n]$ is the $n$th element of the fixed basis vector $\psi_i$. The residual is treated as an AR process:

$$r_n = \sum_{i=1}^{Q} a_i r_{n-i} + e_n$$

Selecting the basis vectors $\psi_i$ becomes part of the parameter estimation problem, along with the determination of the coefficients $c_i$ and $a_i$. One choice is to select $Q/2$ frequencies $\omega_i$, and then create basis vector pairs $\psi_{2i-1} = \cos(\omega_i nT)$ and $\psi_{2i} = \sin(\omega_i nT)$. These pairs can be linearly combined to create an arbitrary amplitude and phase sinusoid at frequency $\omega_i$. A very simple choice for the $\omega_i$ are the frequencies corresponding to the $Q/2$ bins of the discrete Fourier transform of the signal with maximum amplitudes [8].

It is straightforward to find the basis coefficients which minimize the residual in a least squares sense. We rewrite equation (3) in matrix notation:

$$\mathbf{x} = \mathbf{G}\mathbf{c} + \mathbf{r} \tag{4}$$

where $\mathbf{G}$ is now the basis vector matrix:

$$\mathbf{G} = \begin{bmatrix} \psi_1 & \psi_2 & \cdots & \psi_Q \end{bmatrix}$$

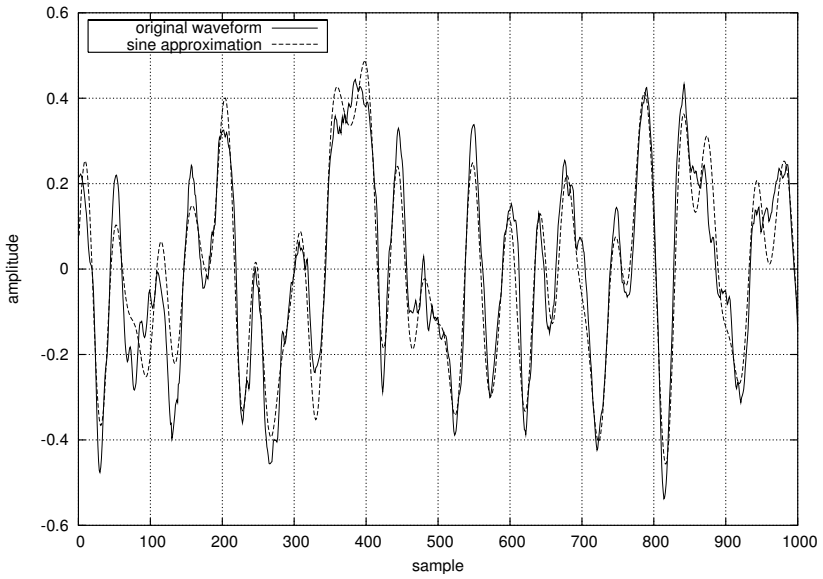


Figure 3: Audio waveform and sinusoid approximation

The least squares estimator is given by:

$$\mathbf{c} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{x}$$

Substituting this $\mathbf{c}$ into equation (4) yields the residual $\mathbf{r}$, which can be treated as an AR process as before.

Figure 3 illustrates the sinusoid modeling. A model with $Q = 31$ basis vectors was used to approximate the waveform from [2]. The vectors corresponded to DC plus the 15 highest amplitude frequencies from the short-time DFT. A window size of 1024 was used.

### 2.1.3 Modeling experiments

To provide an idea of how well the AR/sinusoid model can fit real audio consider Figure 4. Here the ratio of average residual power to average signal power is plotted for values of $P$ and $Q$ ranging between 0 and 121. The audio sample used is 5.5 seconds from the same source as before [2]. In all cases, the AR/sinusoid parameters were estimated for each block of 1024 samples. It is evident that even an AR order of $P = 5$ models the data quite well,

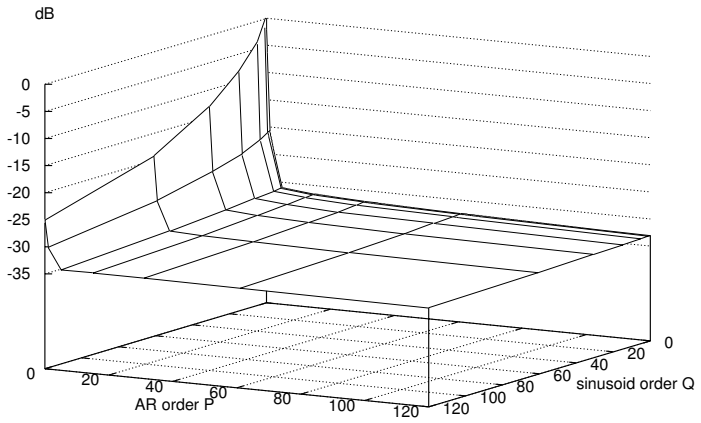


Figure 4: Residual to signal power ratio for various values of $P$ and $Q$

with a power reduction of around 30 decibels. To provide a better look at the contour of the flatter region, Figure 5 clamps the graph above $-30$ decibels.

The choice of block sizes for the AR and sinusoid modeling must be such that the simplifying assumption of stationarity is valid. When the modeling is applied to noisy data, the block sizes should be large enough that a localized degradation does not exhibit too much influence on the parameter estimation for a block.

In general the block sizes for AR coefficient estimation, sinusoid coefficient estimation, and sinusoid frequency selection, as well as the granularity of frequency selection, can all be independently chosen. For these experiments, the AR coefficients, the sinusoid coefficients, and the sinusoid frequencies are estimated for each fixed-size block. Furthermore, the granularity of frequency selection is determined by the granularity of the FFT of a block.

In Figure 6, the power residual to signal ratio is plotted for several choices of block size. $P$ and $Q$ are both fixed at 31. The graph exhibits somewhat of a knee at $12 = \log_2(4096)$. (The up-tick at $7 = \log_2(128)$ can be explained by the details of implementation. In order to produce forward and backward prediction error values for every sample in a block, estimation is actually

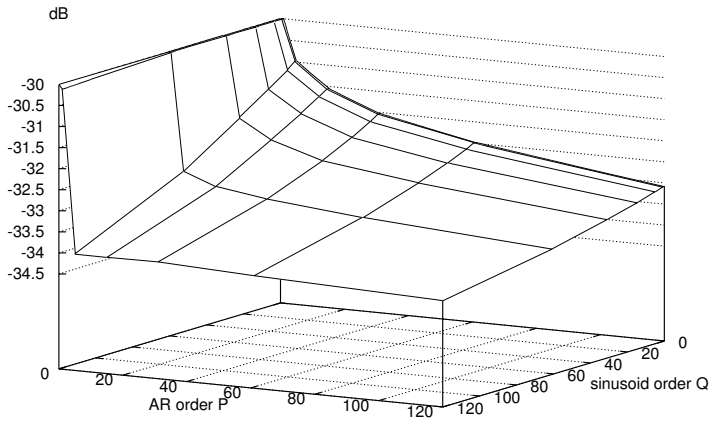


Figure 5: Residual to signal power ratio for various values of $P$ and $Q$ (clamped at $-30$dB)

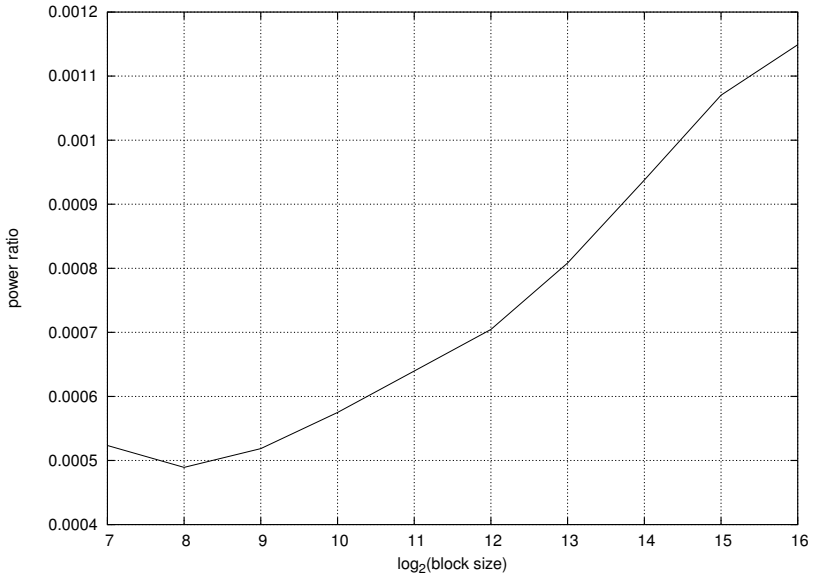


Figure 6: Residual to signal power ratio for various values of $\log_2$(block size)

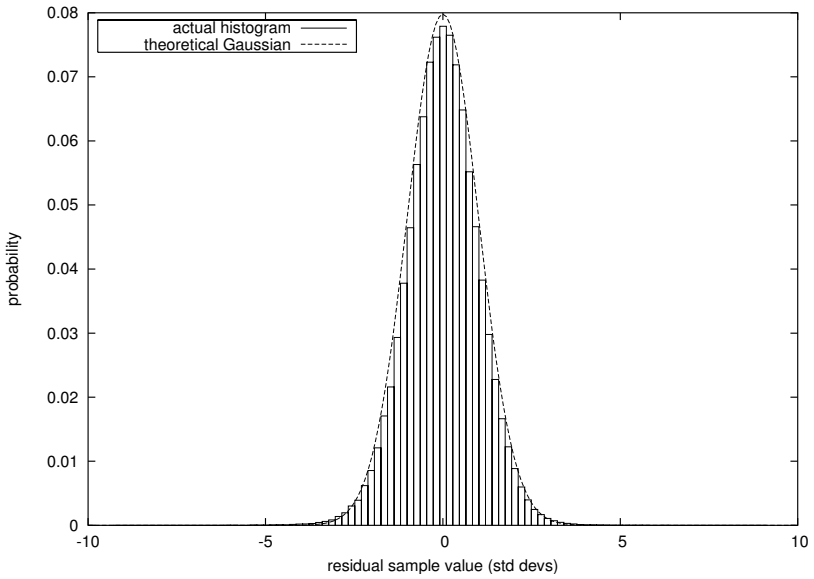


Figure 7: Histogram of normalized residual values

performed using the samples in the block plus the $P$ samples before and the $P$ samples after. The fact that the minimization applies to more than just the block becomes more significant with smaller block sizes. It does not appear the up-tick is related to the granularity of the FFT.) This graph can provide some insight into the stationarity of the audio. Assume that an audio sample can be considered stationary across a block of a particular size. In this case we'd expect the residual power to increase by a relatively small amount when the same parameters are used across the whole block, versus the residuals of the two halves of the block with independently estimated parameters.

Recall that an assumption of our model is that the innovations sequence is Gaussian white noise. Rather than rigorous testing of this hypothesis, we just present a couple demonstrations from real data. We consider the residual resulting from estimating an AR/sinusoid model with $P = 31$ and $Q = 31$ with a block size of 1024 samples. Each block of the sequence is independently normalized to a variance of one. The histogram plotted in Figure 7 presents a beautiful Gaussian curve (plotted against the theoretical normal curve).

As an illustration of the whitening effect, Figure 8 plots the estimated

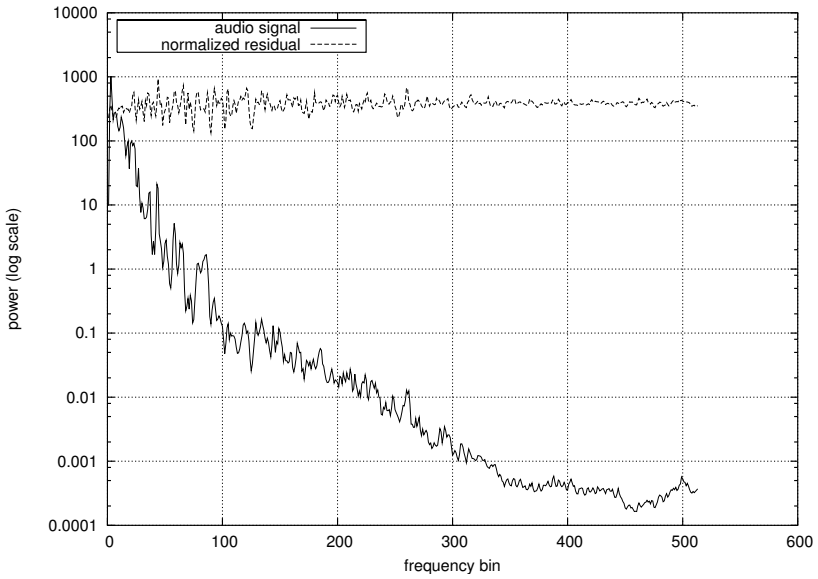


Figure 8: Estimated power spectrum of residual

power spectrums of the original audio and the normalized residual. The power spectrum was estimated using a time-averaged short-time DFT of 1024 sample width.

When considering the modeling results presented here, one should be careful about drawing conclusions:

- All the results were reached from a single audio sample. Producing more general results would require testing with many more types of audio. A complex orchestral recording, for example, might demand a higher AR order to achieve good results.

- Noise corrupted audio will exhibit different behavior than the clean sample used here. Broadband noise could raise the noise floor, broadening the area where increasing model order yields diminishing returns.

- The end goal is more than modeling a clean audio signal. From Figure4 it would appear that sinusoid modeling yields relatively little benefit if the AR model is sufficiently high. However, interpolation of large missing gaps (see section 2.3) can be significantly improved with sinusoid

modeling. Also, estimation for the sinusoid model might prove more robust to localized clicks.

## 2.2 Detection

In section 2.1 we established models that fit clean audio signals. The next step is to identify corrupted samples within noise-degraded audio.

### 2.2.1 Noise modeling

For detection of clicks, we assume an additive noise model. In this model, the corrupted audio process $y_t$ is the sum of an underlying clean AR/sinusoid process $x_t$ and an additive noise process. Because clicks are a localized disturbance, we allow the noise process to take on nonzero values only at certain times. For the purposes of click detection, background noise will be assumed zero at other times. We can formulate this model as:

$$y_t = x_t + i_t n_t$$

The combined process $i_t n_t$ represents the localized noise process. The binary-valued process is 1 when a click is present, and 0 when a click is not present. The process $n_t$ represents the additive amplitude of the click when $i_t$ is 1. The procedure of click detection can be defined as estimating the process $i_t$.

From observing real click degradations, it is possible to describe some of the characteristics of the noise process. Both $i_t$ and $n_t$ are assumed to be independent of $x_t$. However, they may be highly correlated. Individual samples of $i_t$ cannot be considered to be independently distributed, as the noise tends to occur in bursts. Such bursts are observed to typically be between 1 and 200 samples in length for 44.1 kHz audio [8]. The amplitude of clicks, and hence of $n_t$ can vary greatly. It is observed that often the amplitude of a click need not be large relative to the signal amplitude in order to be noticeable to a listener.

While it is possible to try to model all these characteristics of the noise, the simplest detection methods only make use of the assumed independence of the noise from the original signal. If we apply an inverse filter based on the AR/sinusoid model of $x_t$, we produce a prediction error residual process. The expectation is that the power from $x_t$ should be significantly reduced (perhaps 30 dB), but that the clicks should be largely unaffected. Thus,

the clicks are highly amplified relative to the underlying audio, and may be detected by simply thresholding the prediction error.

### 2.2.2 Parameter estimation

The first difficulty when applying this method of click detection is that the parameters of the underlying process $x_t$ are unknown. One simple method of estimation is simply to use the standard estimation techniques on the corrupted process $y_t$. However, for AR estimation at least, Kleiner and Martin [10] have shown that AR estimates can be greatly affected by even relatively low amplitude impulsive noise.

They propose a scheme based on iterated weighted least squares to fit an AR model to the data that is robust to additive impulsive noise. With an improved parameter estimate, we would expect better detection performance.

As a kind of limit study as to the benefit of robust estimation procedures, we perform the following experiment. Parameter estimation is performed directly on a clean audio sample. Then, the audio is artificially corrupted with click-type noise.[1] Parameter estimation is also performed directly on the corrupted waveform. Then, two prediction error sequences are produced from the corrupted audio. One sequence, the *noisy estimate*, uses model parameters estimated directly from the corrupted data. This represents the result of using the most straightforward method. Another sequence, the *perfect estimate*, uses the "true" model parameters estimated from the original audio. This is to represent a hypothetical ideal robust estimator.

We also produce a third prediction error sequence. This sequence, the *previous estimate*, approximates a method that takes advantage of the presumed short-term stationarity of the audio. It is assumed that the audio is cleaned block-by-block, and that model estimates are performed on a cleaned block before moving to the next block. The "clean" model estimate from the previous block is used to produce the error sequence for the current block.

[1] The noise generation is based on an explicit noise model used for some advanced click detection techniques [8]. The noise switching process $i_t$ is considered to be a two-state Markov chain process. The zero state indicates no noise; the one state indicates noise present. The process $i_t$ transitions from state 0 to 1 with probability 0.07, and from 1 to 0 with probability 0.35. The noise samples $n_t$ are modeled as independent Gaussian variables with time-varying variance $\sigma_{n_t}$. Each $\sigma_{n_t}$ is sampled from the heavy-tailed inverse gamma distribution with parameters $\alpha = 0.8$ and $\beta = 10000$. The resulting noise $i_t n_t$ is scaled by $1/32728$ and the noisy data $y_t$ is capped to an amplitude of 1.

| | forward | minimum |
|---|---|---|
| noisy estimate | 8.63 | 17.04 |
| previous estimate | 4.71 | 28.85 |
| perfect estimate | 5.81 | 72.15 |

Table 1: Ratio of MS prediction error of corrupted to uncorrupted samples

This technique is approximated for these experiments by using at each block the *perfect estimate* model for the previous block.

To provide some insight into the noise amplification of different procedures (without introducing any hard thresholds), we construct the ratio of mean squared amplitudes of corrupted to uncorrupted samples from the prediction error sequence. These values are plotted in the first column of Table 1. All experiments were performed with a block size of 1024 and model orders of $P = 31$ and $Q = 31$. Each block was independently normalized before producing mean squared estimates. The normalization was based on a robust estimator $\hat{\sigma}$ for the standard deviation of the sequence [10]:

$$\hat{\sigma} = \text{median}(|e_i|)/0.6745$$

This estimator is robust to outliers, and is non-biased in the case of a zero-mean Gaussian distribution for the $e_i$.

We would expect a larger ratio to allow for better click detection. Curiously, the *perfect estimate* method yields a lower ratio than the simplest method. It turns out that the mean squared amplitude of the "uncorrupted" prediction error samples is dominated by those immediately following "corrupted" samples. This can be explained by considering the auto-regressive equation (1). When producing the prediction error, the additive noise affects the sample where it occurs, as well as the $P$ samples following. Thus, we observe a smearing effect that hampers localization and throws off our noise amplification metrics.

Note that smearing of the prediction error sequence occurs only in the samples immediately *after* a corrupted audio sample. It is possible to take a given AR model, reverse the coefficients, and apply a filter backwards to produce a backwards prediction error sequence. We would expect to see the smearing effect also in this sequence, however the smearing would only be evident in the samples immediately *prior* to a corrupted audio sample.

It is possible to combine the forward and backward prediction error sequences to yield better detection localization. One simple method is to take the minimum amplitude prediction error between the forward and backward values at a given sample. The intuition is that corrupted samples should be high amplitude in both sequences, while the smearing should only appear in one or the other at a given position. The improved localization for all estimation methods is evident in the second column of Table 1. Also, we see a potential for detection improvements with either previous block or robust estimation methods.

### 2.2.3 Noise identification

In a practical detection system, a decision must made as to whether each sample is corrupted or uncorrupted. For this we use a threshold to flag samples with prediction error beyond a certain magnitude. The threshold is chosen relative to the standard deviation of the prediction error sequence for a block (or a robust estimate of the uncorrupted standard deviation). It is straightforward, then, to calculate the likelihood of false positive detections in the case of uncorrupted Gaussian excitation. The likelihood of false positives (and of course of false negatives) in the presence of noise will depend on the characteristics of the noise and of the AR model.

Although we have emphasized localization of the click noise, in practice we will want to "spread" the detections. In the correction stage, all samples not marked as corrupted will be fixed at their original value. If a false negative appears in the middle of a click sequence, it can be impossible to eliminate the noise in the correction stage. Simple pulse spreading techniques can be applied to combat this problem. One method is to create a "fattened" version of the original detection vector, where all samples within some number of samples of a positive detection in the original vector are flagged positive. It may be possible to refine such methods by taking advantage of the observation that the forward prediction error tends to define the start of a click very well, while the backward detection error tends to define the end of a click very well. In these experiments, we will use simple symmetric spreading.

It is very difficult to define a useful metric for a given detection method. Most useful metrics probably require the output of both the detection and correction stages (see section 2.4). Here we will consider the likelihoods of false positive and false negative detection for each sample.

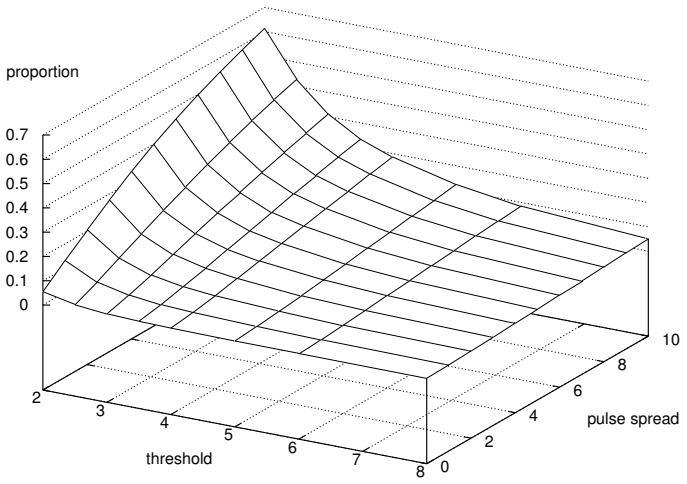


Figure 9: Proportion of false positives among uncorrupted samples

We perform some detection experiments with the same artificial corruption example as before. Straightforward parameter estimation is performed on the noisy data, and the forward prediction error is used for detection. In Figure 9, the proportion of false positive detections among uncorrupted samples is plotted for various threshold and pulse spreading values. As expected, the false positives increase with decreasing threshold and increasing pulse spread. Figure 10 illustrates the proportion of false negatives among corrupted samples. (Note that this figure is rotated 180 ° relative to Figure 9.) False negatives increase with increasing threshold and decreasing pulse spread.

The proportion of false positives and false negatives are plotted together with the pulse spread fixed at 4 samples (Figure 11) and with the threshold fixed at 2 (Figure 12).

## 2.3 Correction

After identifying which samples have been corrupted by impulsive noise, the next step is to perform correction to eliminate the noise. While it is possible

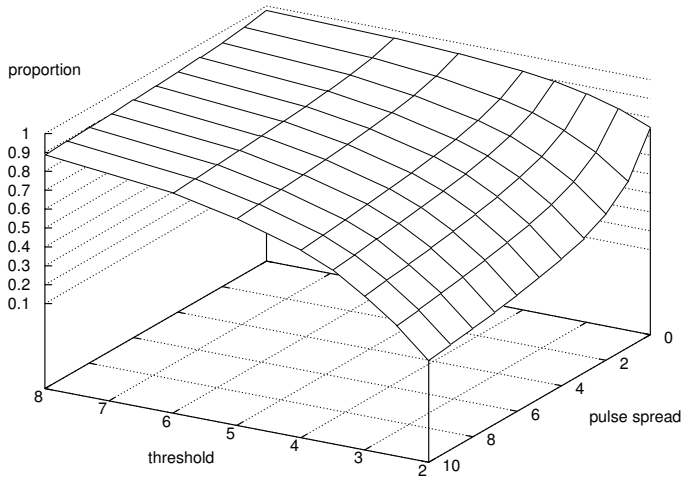


Figure 10: Proportion of false negatives among corrupted samples

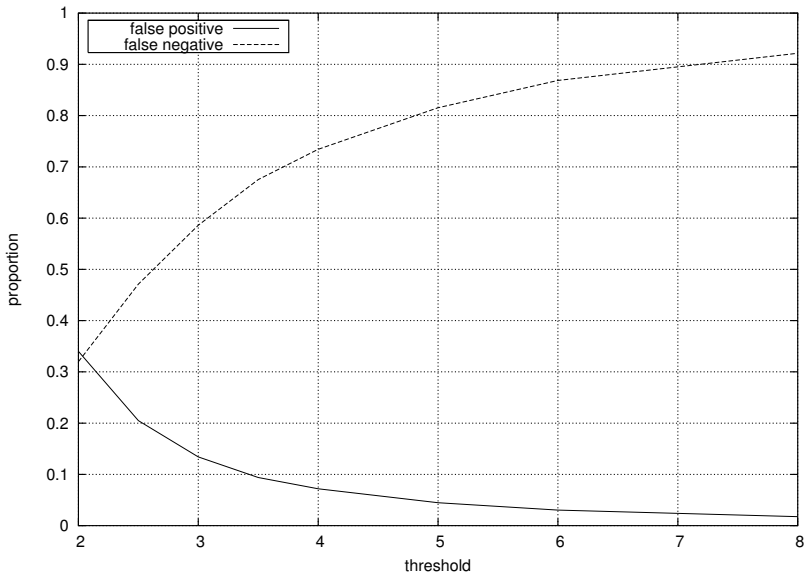


Figure 11: Proportion of false positive and false negative detections (spread = 4 samples)

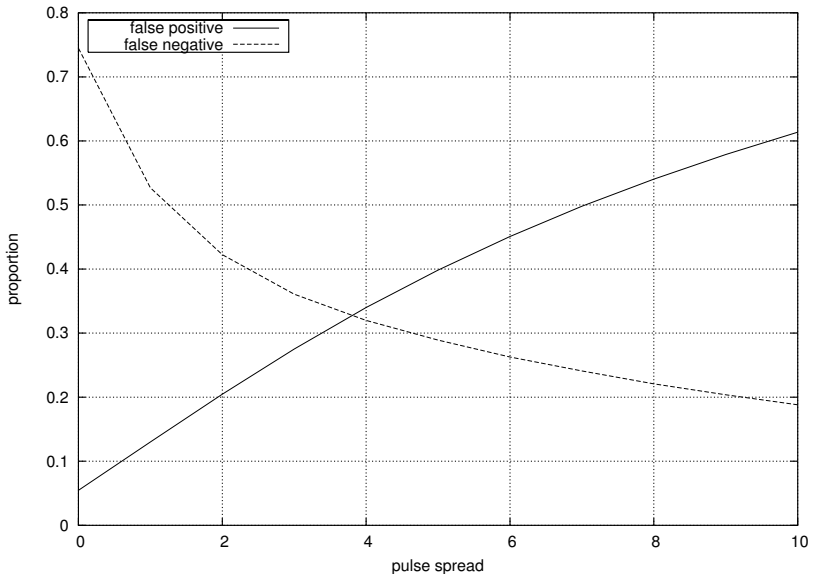


Figure 12: Proportion of false positive and false negative detections (threshold = 2)

to try to model the additive noise process, we will simply consider corrupted samples as missing data, and formulate the correction problem as interpolation. Since typically only a small proportion of samples are corrupted, it is usually possible to use the regularity of an audio sample to achieve high quality restoration.

### 2.3.1 Least squares interpolation

For the purposes of interpolation, we will use a new form of the vector autoregressive equation (2). Suppose we are working with a length $N$ block of data $\mathbf{x}$ from an AR process. We form a $N - P$ by $N$ matrix $\mathbf{A}$ from the AR vector $\mathbf{a}$ like so:

$$\mathbf{A} = \begin{bmatrix} -a_P & \cdots & -a_1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & -a_P & \cdots & -a_1 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -a_P & \cdots & -a_1 & 1 & 0 \\ 0 & \cdots & 0 & 0 & -a_P & \cdots & -a_1 & 1 \end{bmatrix}$$

We can then express the prediction error sequence in terms of $\mathbf{A}$ and $\mathbf{x}$:

$$\mathbf{e} = \mathbf{A}\mathbf{x} \tag{5}$$

If $\mathbf{i}$ is the binary vector where a value of one indicates a missing sample, then we partition $\mathbf{x}$ into two vectors. One, $\mathbf{x}_{(\mathbf{i})}$, represents the missing values and the other, $\mathbf{x}_{-(\mathbf{i})}$, represents the known values. We can perform the corresponding partitioning of the columns of $\mathbf{A}$ such that (5) becomes:

$$\mathbf{e} = \mathbf{A}_{-(\mathbf{i})}\mathbf{x}_{-(\mathbf{i})} + \mathbf{A}_{(\mathbf{i})}\mathbf{x}_{(\mathbf{i})}$$

Assume for the moment that the AR coefficients are known. Then, the objective of the interpolation procedure is to estimate $\mathbf{x}_{(\mathbf{i})}$, given $\mathbf{x}_{-(\mathbf{i})}$, $\mathbf{i}$, and $\mathbf{a}$. The least squares AR interpolator [8] is the estimator which minimizes the sum squared of the prediction error $\mathbf{e}$. This estimate is given by:

$$\mathbf{x}_{(\mathbf{i})} = -(\mathbf{A}_{(\mathbf{i})}{}^T\mathbf{A}_{(\mathbf{i})})^{-1}\mathbf{A}_{(\mathbf{i})}{}^T\mathbf{A}_{-(\mathbf{i})}\mathbf{x}_{-(\mathbf{i})}$$

In the case of Gaussian excitation, and if there is no missing data in the first $P$ samples of the block, the least squares interpolator maximizes the likelihood of $\mathbf{x}_{(\mathbf{i})}$ given $\mathbf{x}_{-(\mathbf{i})}$ [8]. This restriction on the first $P$ samples is usually observed in practice by prepending the last $P$ samples from the previous de-noised data block.

The sinusoid model can also be incorporated into the interpolation procedure. The sinusoids can be filtered out of each block, AR interpolation performed on the residual, and then the sinusoids reintroduced.

### 2.3.2 Iterative estimation

In practice, the AR coefficients and sinusoid parameters are unknown beforehand. Part of the interpolation, then, is to estimate the modeling parameters as well as the missing data. A simple method [8] to achieve this joint estimation is to first choose some initial estimates for the model parameters and the unknown data, and then to iteratively re-estimate the data and model parameters sequentially. For an initial choice of the unknown data, it is possible to simple use the corrupted audio, or initialize to zero. The model parameters can be estimated directly from the initialized data. Alternatively, robust estimation procedures may be useful, or a robust estimate from the detection procedure may be available.

It is possible to incorporate the sinusoid coefficient estimation simultaneous with the unknown data estimation stage, but such a formulation is not explored here – each component is estimated sequentially.

Practical tests with this iterative block-based scheme reveal an important implementation detail. If missing data occurs at the end of a block, data beyond the block does not inform the estimation of the missing data. The result can be a discontinuity which is assumed fixed during the next block interpolation. The result can be an audible click, which wasn't present even in the uncorrected noisy audio. This artifact can be avoided by interpolating $P$ samples beyond the actual block end, and simply throwing away any samples estimated from this extra portion before moving to the next block.

### 2.3.3 Performance

To test click removal performance, we will again use the clean signal with artificial corruption as with click detection. The true noise presence vector is used to indicate which samples are to be interpolated, simulating a perfect detection process. To judge the convergence of the algorithm, the scheme is iterated eight times at each block. Model parameters $P = 31$ and $Q = 31$ are used with a block size of 1024 samples.

Figure 13 plots the mean squared error of corrupted samples for each iteration. For this example, the algorithm converges very quickly, stabilizing after only two iterations. Informal listening tests indicate the interpolated audio sequence to be indistinguishable from the original.

## 2.4 Demonstration

The full click removal process for a given block consists of performing sequentially the prediction error-based click detection, followed by the iterative interpolation. This process is performed for each subsequent block to restore the entire audio sequence.

### 2.4.1 Artificial corruption

When considering the full process, we may hope to find better metrics to guide the choice of detection parameters. With this in mind, we return to the clean audio with artificial corruption scenario we've used throughout this section. In Figure 14 the ratio of squared error for processed audio to that

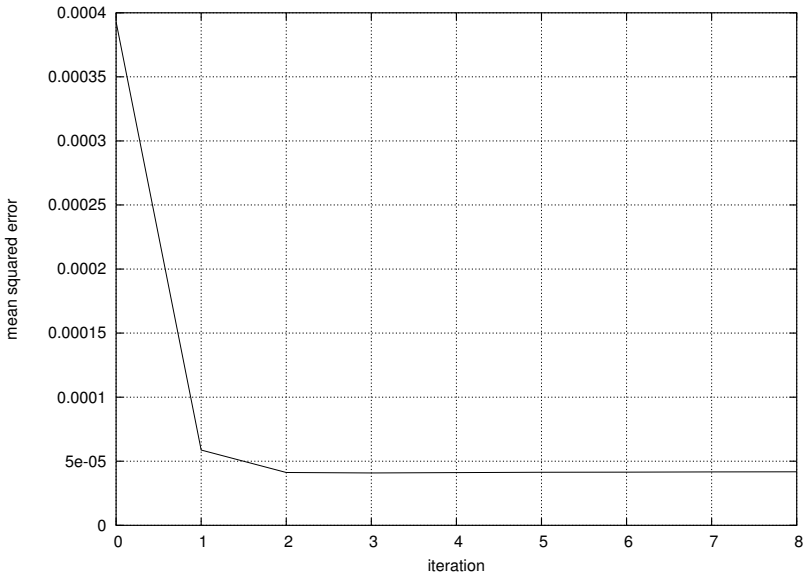


Figure 13: MSE of corrupt data for each interpolation iteration

of the unprocessed corrupted audio is plotted for three choices of the pulse spreading detection parameter. The lines represent a pulse spread of 2, 4, and 8 samples plotted for threshold values between 3 and 40. In all cases $P = 31$, $Q = 31$, the block size is 1024 samples, and the number of interpolation iterations is 3. The error ratio continues to ramp up quickly for threshold values (not shown) less than 3. The error ratio for all three lines would reach the value 1 if the threshold was chosen high enough that no samples were flagged.

Informal listening tests reveal the limitations of mean squared error as a click removal metric. The minimum MSE for the configurations tested was a threshold of 5 and a pulse spread of 2 samples. Comparing the processed audio[2] to the degraded audio[3] reveals that all the most noticeable clicks and pops have been removed without any noticeable distortion to the original signal[4] . There does remain a small amount of (subtle) low-amplitude noise. The configuration with a threshold of 3 and a pulse spread of 4 samples yields

[2] `clicksamps/iwproc-50-02.wav` (local, web)
[3] `clicksamps/iwclicks.wav` (local, web)
[4] `source/iwoke.wav` (local, web)

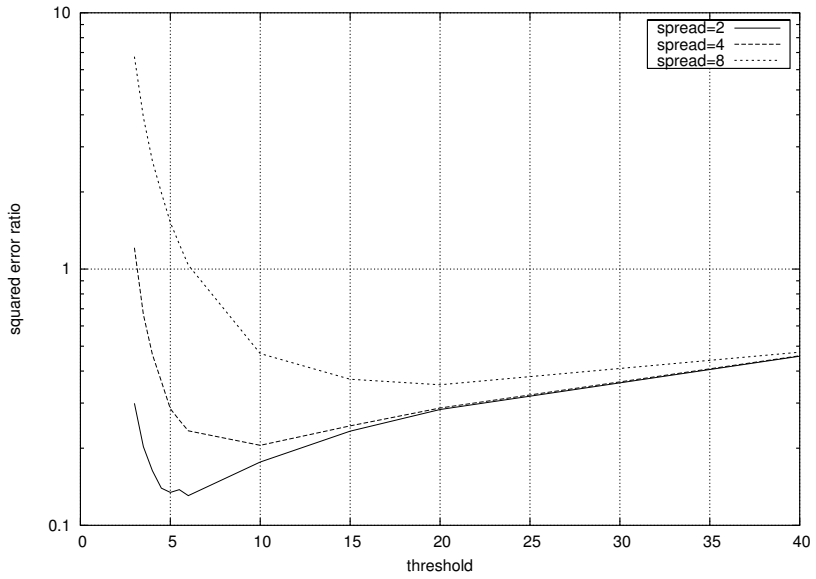


Figure 14: Squared error ratio of processed audio to unprocessed audio (log scale)

audio[5] with almost all the low-amplitude noise removed, but with a small amount of signal distortion. The choice between the two might be a subjective decision. However, the processed audio of the second configuration has an error ratio well above 1, indicating it is worse than the original corrupted audio by the MSE metric. It would seem that MSE criteria does not match human perception particularly well.

### 2.4.2 Real examples

The gauge the subjective performance of these methods on real degraded audio, click removal was performed on digital recordings made from 33 RPM phonograph records. All processing was done with the same parameters: $P = 31$, $Q = 31$, block size is 1024 samples, the threshold is 5, the pulse spread is 4 samples, and the number of interpolation iterations is 3.

The first example is from a recording [6] featuring two male vocalists

[5] `clicksamps/iwproc-30-04.wav` (local, web)

singing *a cappella*. The original sample[6] contains medium-sized, well-spaced pops. The restoration[7] removes all audible pops with no audible distortion.

The second example is from an instrumental recording [11] featuring horns, drums, and other percussive instruments. The original sample[8] contains fairly-constant low-level crackle, as well as the occasional pop. The restoration[9] successfully removes both crackle and pops, without noticeable distortion.

The next example is from a recording [1] featuring a male vocalist accompanied by guitar and trombone. The original sample[10] exhibits short sections of rapid, high-amplitude clicks, probably correlated to the rotation of the disc. Again, the restoration[11] does a fine job of removing the degradation without disturbing the signal.

The last example is from a recording [5] featuring a female vocalist accompanied by guitar. The original sample[12] contains small amounts of bursty click noise. In parts it also exhibits a low-level crackle-type noise that appears to correlate to high signal amplitude. The restoration[13] removes the individual clicks, and also reduces the correlated crackle to the point that it is largely masked by the signal.

## 2.5 Future Directions

There are many opportunities for improvement within the framework described in this section. One such area is implementing model estimation techniques that are robust to impulsive noise. One suitable technique is to perform estimation based on iterative weighted least squares. This involves minimization with an influence curve (other than the typical quadratic) which de-emphasizes the influence of outliers. Another possible technique is to incorporate model estimates from already de-noised data in previous blocks.

Once a model has been selected, detection accuracy may be improved by more sophisticated use of both the forward and backward prediction error.

[6] `source/tina.wav` (local, web)
[7] `clicksamps/tina-proc-50-04.wav` (local, web)
[8] `source/hatari.wav` (local, web)
[9] `clicksamps/hatari-proc-50-04.wav` (local, web)
[10] `source/brady.wav` (local, web)
[11] `clicksamps/brady-proc-50-04.wav` (local, web)
[12] `source/cali.wav` (local, web)
[13] `clicksamps/cali-proc-50-04.wav` (local, web)

One simple technique to be investigated would involve asymmetric spreading of the binary detection sequence corresponding to each prediction error sequence. The forward detection sequence would spread towards increasing time, while the backward detection sequence would spread towards decreasing time. The final detection sequence would be the logical and of the two spread sequences. In this way, the forward sequence clearly defines the start of a click, and the backward sequence clearly defines the end.

The interpolation stage may also be improved, for example by constraining interpolation to a minimum innovation variance to prevent underestimating variance [8]. However, for the short interpolation lengths encountered in click removal, the current scheme has worked quite well.

More recent work in this area has explored explicit noise modeling within a fully Bayesian framework [8]. Such methods have the potential to perform better than the more *ad hoc* methods considered here.

The methods studied here, as well as any extensions or alternative methods, require much more study in relation to both audio and noise with more widely varying characteristics.

# 3 Hiss Reduction

Analog recordings of all types suffer from broadband noise to some degree. Magnetic audio tape is one example that tends to exhibit highly stationary broadband hiss noise. Hiss reduction attempts to attenuate the broadband noise, without introducing unpleasant artifacts or signal degradation.

This section investigates short-time Fourier transform modeling of audio, spectral noise suppression, residual noise conditioning, and the Ephraim and Malah noise reduction method. Hiss reduction is demontrated for real examples of corrupted audio, and possible future research directions are suggested.

## 3.1 Modeling

In Section 2, the localized nature of click-type noise led us to use time-domain methods such as auto-regressive modeling. For the purposes of hiss reduction, we will employ frequency-domain methods. In particular, we will make use of the assumption that hiss-type noise is long-term stationary. We will also make use of the approximation that the audio signal itself is short-term stationary.

The observed noise $y_t$ will be written as:

$$y_t = x_t + d_t$$

We assume here that the noise $d_t$ is additive and independent of the original audio signal $x_t$.

All noise suppression methods used here will make use of the short-time discrete Fourier transform (STFT). The observed data $y_t$ is divided into overlapping subframes of length $N$, with $M < N$ being the distance between the start of successive subframes. A STFT can be performed on each of the subframes (after applying a windowing function) to produce the complex STFT value $Y(p, \omega_k)$ for the $k$th frequency bin ($0 \leq k < N$) from subframe $p$.

The representation $Y(p, \omega_k)$ can be transformed back to the time series representation $y_t$ by applying an inverse STFT to each subframe, and then combining the results using overlap-add. A gain compensation function restores the time domain sequence to the correct amplitude.

All the methods studied here will perform noise reduction on the $Y(p, \omega_k)$ representation before transforming back to the time domain. Every method applies a positive real-valued gain $G(p, \omega_k)$ to each bin of each subframe.

$$\hat{X}(p, \omega_k) = G(p, \omega_k) Y(p, \omega_k)$$

Thus the phase of the restored audio components $\hat{X}(p, \omega_k)$ will be the same as that of the corrupted audio. In fact, Ephraim and Malah have shown [7] that, under certain assumptions, the phase of the observed data is the minimum mean squared error (MMSE) estimator of the original phase.

It is claimed that the human ear is relatively insensitive to phase. To test this claim as it relates to our purposes, we construct a perfect zero-phase noise reduction filter. We take a clean audio signal from [2], and add Gaussian white noise of variance 0.0015. We then perform the STFT transformation, and scale each bin so that its amplitude is equal to the amplitude of the corresponding bin of the uncorrupted signal. This ideal gain is:

$$G(p, \omega_k) = \frac{|X(p, \omega_k)|}{|Y(p, \omega_k)|}$$

which ensures that $|\hat{X}(p, \omega_k)| = |X(p, \omega_k)|$. From casual listening, it is difficult to distinguish the original sample[14] from the perfect zero-phase recon-

[14] `source/iwoke.wav` (local, web)

struction[15] .

An important parameter to STFT analysis is the window size. Increased window duration leads to increased spectral resolution. Analysis [4] has shown that window durations of at least 40 to 50 ms are required to avoid damage to quasi-stationary signal components in the presence of noise. However, increased window duration increases the spreading of sharp transients. For many types of audio, the transient spreading is not noticeable with windows well above 50 ms. However, some signals may exhibit transients that are damaged using windows of greater than 40 ms duration. Uniform STFT methods may not be suitable for such audio.

The window duration used throughout this section will be 2048 samples, or 46 ms at 44.1 kHz. The Hanning window will be used for both the analysis and synthesis windows. Windows will be overlapped by a factor of 4.

## 3.2 Spectral Noise Suppression

In practical noise suppression, we don't know $X(p, \omega_k)$ or $D(p, \omega_k)$. Due to the assumed stationarity of the noise, it is usually possible to deduce an estimate for the noise spectrum. If a portion of the audio signal can be identified which contains noise only (there is no signal, $x_t = 0$), then an estimate for the power spectrum $\hat{S}_d(\omega_k)$ can be obtained by averaging the magnitude squared of the STFT bins for $P$ subframes from the noise only region:

$$\hat{S}_d(\omega_k) = \frac{1}{P} \sum_p |Y(p, \omega_k)|^2$$

Then, $\hat{S}_d(\omega_k)$ should approximate $E\{|D(p, \omega_k)|^2\}$ and can be used as an estimate for $|D(p, \omega_k)|^2$ in every subframe.

A noise only region can often be easily manually identified. It would also be possible to attempt to automatically identify such a region, but this possibility is not explored here.

Various noise suppression rules are used to set the gain function used for spectral noise suppression. The simplest rules can be expressed as a function of the ratio $Q(p, \omega_k)$ of observed power to expected noise power for a given bin.

$$Q(p, \omega_k) = \frac{|Y(p, \omega_k)|^2}{\hat{S}_d(\omega_k)} \tag{6}$$

[15] `clicksamps/iwideal.wav` (local, web)

The power spectral subtraction suppression rule subtracts the expected noise power from the power spectrum. This is equivalent to estimating the restored amplitude as the square root of the maximum likelihood estimator for the signal variance in each bin [7]. The spectral gain $G$ is given by:

$$G(p, \omega_k) = \sqrt{1 - \frac{1}{Q(p, \omega_k)}}$$

Another suppression rule is known as the Wiener filter. This rule estimates the restored amplitude as equal to the amplitude of the minimum mean squared error estimator of the complex signal component for each bin [7]. The spectral gain $G$ is given by:

$$G(p, \omega_k) = 1 - \frac{1}{Q(p, \omega_k)}$$

This rule is simply the square of the power spectral suppression rule. Both these rules clamp the lower gain limit at zero.

Figure 15 illustrates the effect of spectral noise suppression on the time domain waveform. The top graph shows a section of clean audio. The middle shows the same section corrupted with Gaussian white noise of variance 0.015. The bottom graph shows the restored audio after processing with the Wiener suppression rule using a window size of 2048 samples.

## 3.3 Residual Noise Conditioning

For the purposes of high quality audio restoration, the key to noise suppression performance is not the amount of noise reduction, but the nature of the residual noise (along with the amount of distortion to the signal, of course). The most noticeable effect from simple spectral noise suppression is what's known as *musical noise*. This effect is caused by the randomness in the STFT of the noise. From subframe to subframe, the noise power in a given bin randomly fluctuates around its average. In bins with low signal power, the fluctuations can result in wildly varying estimates of signal to noise ratio, and hence gain. The name for the phenomenon comes from the pure tones that randomly appear in a subframe, creating a rapid, pseudo-musical "tinkling" effect. This effect is evident in the Wiener-processed restoration[16] of the white-noise corrupted[17] .

[16] `hisssamps/iwwien.wav` (local, web)
[17] `hisssamps/iwhiss.wav` (local, web)

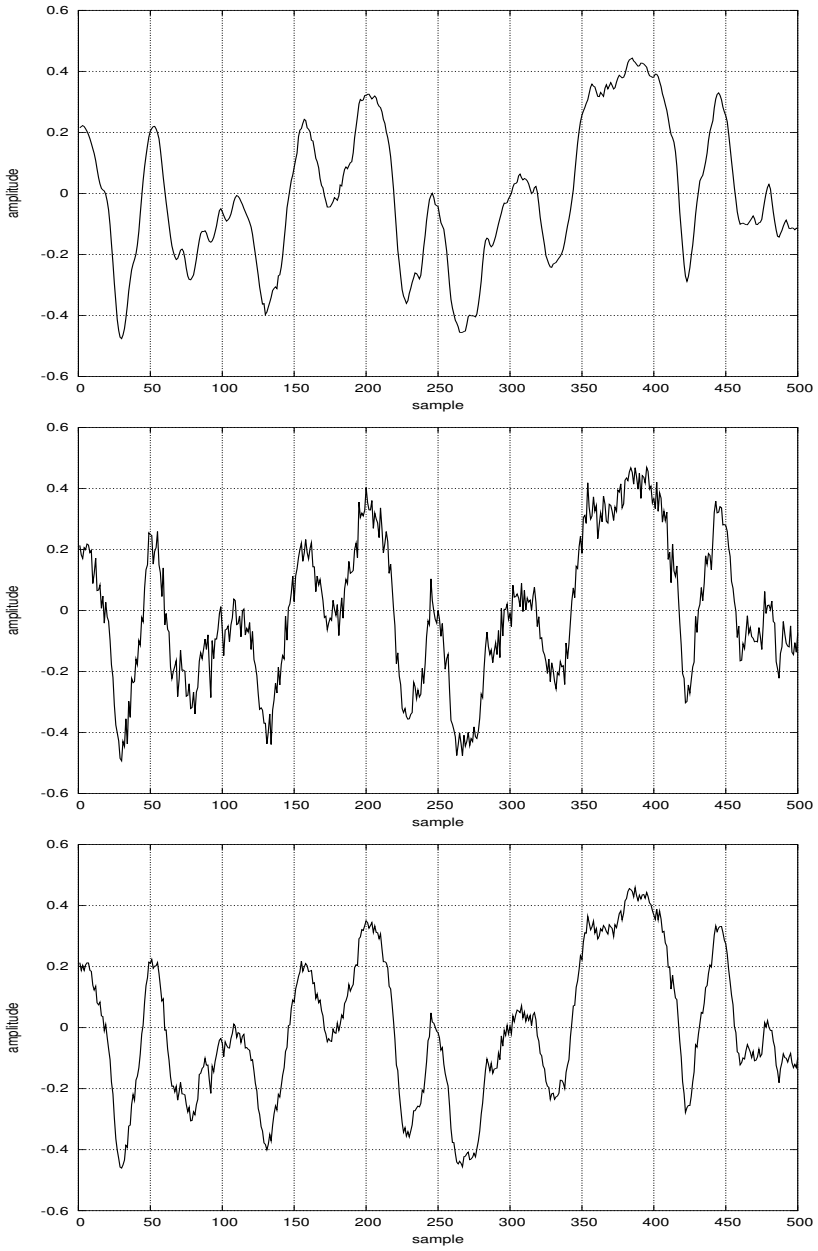


Figure 15: Top: clean audio. Middle: corrupted with white noise. Bottom: after Wiener processing

One method used to reduce musical noise is to overestimate the noise power by a factor $\alpha$. This has the effect of clamping more of the low-amplitude bins to zero. A random fluctuation in a noise component is less likely to push above the clamping threshold. Of course, the overestimation also introduces more signal distortion in bins with significant signal power. Thus, $\alpha$ trades off signal fidelity to reduce musical noise. Typical values for $\alpha$ are between 3 and 6 [9].

Another simple method to reduce the effect of musical noise is to introduce a noise floor to conceal colored residual noise. This is usually achieved by placing a lower limit $\beta$ on the filter gain. While increasing the amount of residual noise, the result may sound much more natural. Typical values for $\beta$ are between 0.01 and 0.1 for processing high noise levels [9].

Both noise overestimation and a noise floor can easily be included in the noise suppression rules discussed. The Wiener filter gain, for example, would become:

$$G(p,\omega_k) = \max\left(\beta\,,\quad 1-\frac{\alpha}{Q(p,\omega_k)}\right)$$

The result[18] of Wiener processing with $\alpha = 5$ and $\beta = 0.1$, exhibits less of the musical effect, at the cost of reduced noise attenuation and increased distortion.

These two simple methods for reducing musical noise trade signal distortion or residual noise for a reduction of the musical noise effect. More sophisticated methods try to take advantage of the characteristics of the STFT bins over time. One observation is that the power of noise fluctuates randomly from subframe to subframe, while the signal components tend to have a smoother envelope. Various types of smoothing can be applied to the frequency bins in the time dimension. One simple method is to average the magnitude of a component across several subframes. This smoothed estimate is used to compute the signal to noise ratio, providing a more stable gain. A method which is found to have a less severe effect on the signal envelope is to take a median amplitude rather than a mean [8].

## 3.4 Ephraim and Malah noise reduction

Ephraim and Malah [7] developed a minimum mean squared error estimator for signal amplitude. The resulting suppression rule is observed to result in

[18] `hisssamps/iwwien-50-10.wav` (local, web)

a relatively colorless residual.

### 3.4.1 MMSE spectral attenuation

When constructing their estimator, Ephraim and Malah distinguish between two quantities.[19] $R_{prio}$ is the *a priori* estimate for the power signal to noise ratio, and $R_{post}$ is the *a posteriori* power signal to noise ratio.

In their model, the *a priori* distribution of the STFT coefficients of both the signal and noise are assumed known. In this case, $R_{prio}$ is defined as:

$$R_{prio}(p, \omega_k) \triangleq \frac{S_x(p, \omega_k)}{S_d(p, \omega_k)}$$

where $S_x(p, \omega_k)$ is the *a priori* expected signal power, and $S_d(p, \omega_k)$ is the *a priori* expected noise power. In practice, the noise characteristics are considered stationary and $S_d(\omega_k)$ is estimated as before. The signal characteristics are also estimated from the data, as will be explained later.

The *a posteriori* ratio is estimated from the observed STFT:

$$R_{post}(p, \omega_k) \triangleq \frac{|Y(p, \omega_k)|^2}{S_d(p, \omega_k)} - 1$$

Observe that when the noise is considered stationary, $R_{post}$ is simply $Q$ from (6) minus 1.

The spectral attenuation for the Ephraim and Malah noise suppression rule is a function of these two quantities (with implicit dependence on $p$ and $\omega_k$):

$$G(p, \omega_k) = \frac{\sqrt{\pi}}{2} \sqrt{\left(\frac{1}{1 + R_{post}}\right)\left(\frac{R_{prio}}{1 + R_{prio}}\right)}\, \mathbf{M}\left[(1 + R_{post})\left(\frac{R_{prio}}{1 + R_{prio}}\right)\right] \tag{7}$$

where $\mathbf{M}$ is a function defined as:

$$\mathbf{M}[\theta] = \exp\left(-\frac{\theta}{2}\right)\left[(1 + \theta) I_0\left(\frac{\theta}{2}\right) + \theta I_1\left(\frac{\theta}{2}\right)\right]$$

where $I_0$ and $I_1$ are the modified Bessel functions of zero and first order, respectively.

[19] We use the formulation from [3], which is slightly different but equivalent to [7]

To get an idea of the behavior of this suppression rule, we will compare it to the Wiener and power spectral subtraction rules. Figure 16 plots the gain of the two traditional rules against $R_{post}$. Figure 17 plots the gain of the Ephraim and Malah rule versus $R_{prio}$, for three different values of $R_{post}$. In Figure 17, the dominant parameter is $R_{prio}$. For values of $R_{post}$ greater than 20 dB or less than -20 dB, the gain curves don't differ much from the 20 dB curves. Notice that in the left half of Figure 17, the curve for $R_{post}$ = -20 dB looks very much like the curve for power subtraction (although the power subtraction curve is a function of $R_{post}$ rather than $R_{prio}$). Similarly, the curve for $R_{post}$ = 20 dB looks very much like the curve for the Wiener rule. When we plot the gain for the Ephraim and Malah rule with $R_{post} = R_{prio}$ (Figure 18), the curve looks very much like power subtraction throughout the plot.

The value of $R_{prio}(p, \omega_k)$ must be estimated from the data. One method is to recursively combine the best previous estimate for the signal to noise ratio (as evidenced by the gain used) with weight $\alpha$, with the *a posteriori* SNR with weight $1 - \alpha$. Thus, we have:

$$R_{prio} = \alpha \frac{|G(p-1, \omega_k) Y(p-1, \omega_k)|^2}{S_d(\omega_k)} + (1-\alpha) P[R_{post}(p, \omega_k)]$$

where $P[x] = \max(x, 0)$. The expression $|G(p-1, \omega_k) Y(p-1, \omega_k)|^2$ is the power of the restored signal in the previous subframe. The choice of $\alpha$ will be discussed later.

In the formulation described above, Ephraim and Malah assume that a signal is present. They also consider a model where uncertainty about signal presence is included. A parameter $q$ representing the probability that a signal is not present is added, and the formulas for $R_{prio}$ and $G$ are adjusted to reflect this. See [7] for details.

### 3.4.2 Analysis

Cappé has performed some analysis [3] which explains how the Ephraim and Malah method reduces the musical noise effect. The first reason is in the use of $R_{prio}$ as the dominant parameter. In frequency bins with no signal component, the $R_{prio}$ resembles a smoothed version of the wildly fluctuating $R_{post}$. This smoothing reduces the musical noise. When significant signal is present, $R_{prio}$ follows $R_{post}$ with very little smoothing. This lessens the distortion of the signal. If $\alpha$ is chosen to be very close to one, the smoothing

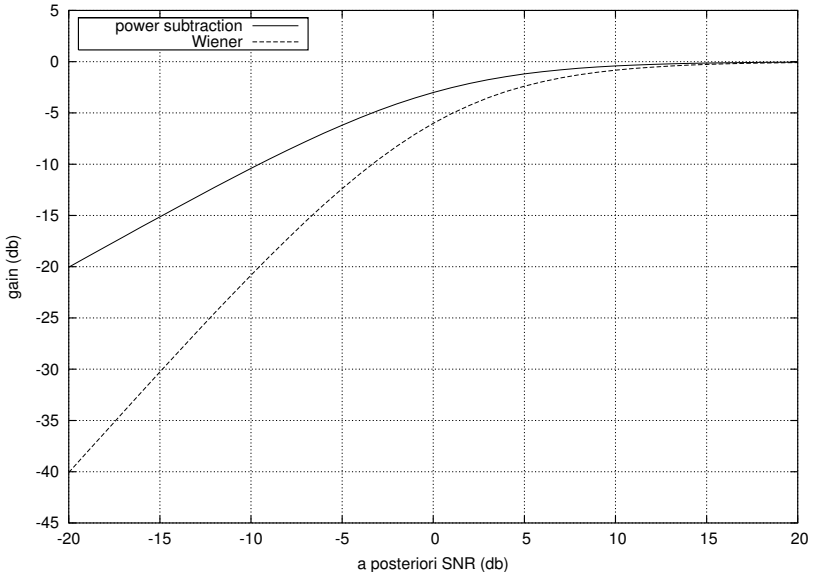


Figure 16: Gain curves for power subtraction and Wiener rules

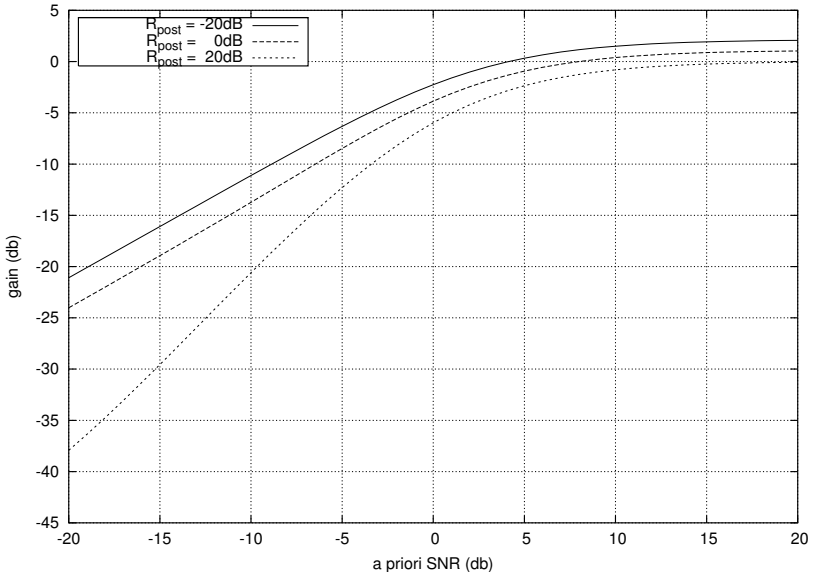


Figure 17: Gain curves for Ephraim and Malah rule

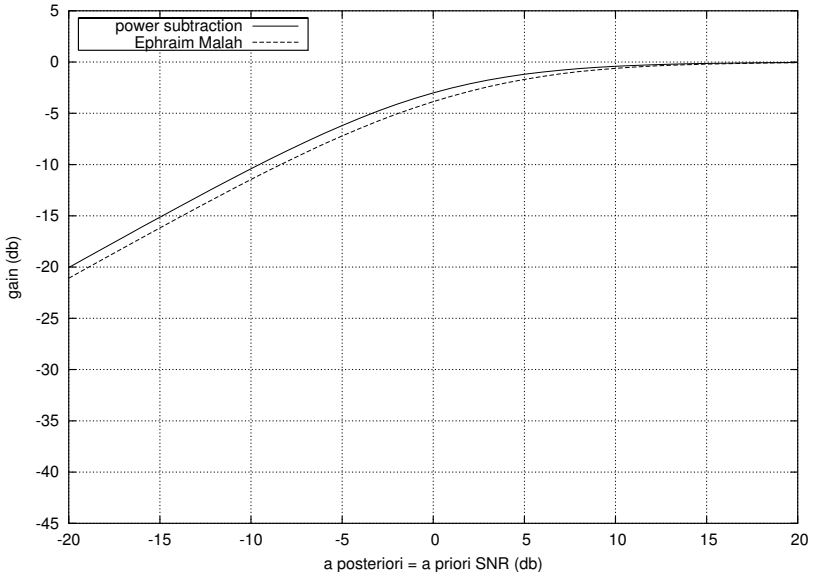


Figure 18: Gain curves for power subtraction and Ephraim and Malah rules with $R_{prio} = R_{post}$

of noise-only sections is increased (reducing musical noise), but the transition to less smoothing happens slower (increasing the signal distortion). If signal presence is not considered, the recommended value of $\alpha$ is 0.98. If an uncertainty of $q = 0.2$ is modeled, the recommended value of $\alpha$ is 0.99.

The smoothing effect described above is also observed if the Wiener suppression rule is used as a function of $R_{prio}$ rather than $R_{post}$. The Ephraim and Malah rule (7) exhibits another effect. Consider again Figure 17. Notice on the left side (where $R_{prio}$ is low – implying low signal levels) that high values of $R_{post}$ yield increased attenuation. Thus when the instantaneous value of $R_{post}$ disagrees with the smoother $R_{prio}$, increased suppression is applied. This effect further reduces musical noise.

Some musical noise may still remain after processing with the Ephraim and Malah method. Cappé suggests to constrain $R_{prio}$ to be greater than a threshold $R_{min}$. In this case, the noise power reduction in noise only bins is approximately $1/R_{min}$. Thus, residual noise can be used to mask the musical noise effect.

Ephraim and Malah processing was performed on the artificial Gaussian

white noise sample. The standard procedure was performed with $\alpha = 0.98$ for values of $R_{min}$ of 0[20] and 0.06[21] . The uncertainty-modeling procedure was performed with $\alpha = 0.99$ and $q = 0.2$ for values of $R_{min}$ of 0[22] and 0.016[23] . Both models reduce the musical noise phenomenon, however the second model yields significantly less colored noise than the original model. For this example audio, the Ephraim and Malah method seems to produce an echo-like effect with certain tones. Further investigation will be necessary to understand this effect.

## 3.5 Demonstration

To see the behavior of the described noise suppression with real degraded material, hiss reduction was performed on digital recordings made from a magnetic tape source. The original recordings were made outdoors on consumer grade audio tape with an inexpensive internal-microphone tape recorder. All processing was done using the Ephraim and Malah method with $\alpha = 0.99$, $q = 0.2$, $R_{min} = 0.016$, a block size of 2048 samples, and an overlap factor of 4.

The first example [12] features a solo female voice (accompanied by cricket chirps). The original sample[24] contains significant broadband noise. The restoration[25] yields a much reduced, natural-sounding residual without noticeable signal distortion.

It has been observed [4] that low-level, lower-frequency signals components are more likely to be decimated by spectral attenuation than higher frequency components. In the second example [13], the female vocalist from the previous example is joined by a lower level, lower-frequency male voice. The original[26] exhibits the same kind of noise as the previous example. The restoration[27] again is successful in strongly reducing the noise, while preserving the deeper voice.

[20] `hisssamps/iwem.wav` (local, web)
[21] `hisssamps/iwem-06.wav` (local, web)
[22] `hisssamps/iwemq.wav` (local, web)
[23] `hisssamps/iwemq-016.wav` (local, web)
[24] `source/gma-letter.wav` (local, web)
[25] `hisssamps/gma-letter-emq-016.wav` (local, web)
[26] `source/gma-utah.wav` (local, web)
[27] `hisssamps/gma-letter-emq-016.wav` (local, web)

### 3.6 Future Directions

Hiss reduction is a very subjective process, and there is a lot of room for tuning the methods described here. In particular, it would be interesting to explore additional methods for smoothing the random noise fluctuations, or performing time domain signal separation across the STFT bins.

Hoeldrich and Lorber [9] have added additional parameters to the methods described here, and additionally consider perceptual frequency masking when deciding which bins to attenuate. Perceptual criteria could play a very useful part in improving hiss reduction.

One way to improve the automation of these methods would be to automatically identify noise-only sections of input audio. Such an algorithm should be robust to different noise and signal characteristics. Time series modeling might aid in the identification of desired audio signal presence.

## 4 Conclusion

This paper has explored several methods for click removal and hiss reduction. Excellent results were demonstrated for click removal using a hybrid autoregressive/sinusoidal basis model. Clicks were detected by thresholding prediction error, and were removed using least squares interpolation. Excellent results for hiss reduction were also demonstrated. Natural-sounding hiss reduction was achieved by using the Ephraim and Malah suppression rule for short-time spectral attenuation.

## A Audio Samples and Program Code

Demonstration audio samples are referenced in footnotes throughout this paper. If you are reading this online, you may be able to access the files either locally or from the Internet by clicking on the "local" or "web" links, respectively. Alternatively, you can find an index to the samples locally as `audio_samples.html` or on the web as `http://jnuzman.github.io/audio-restoration-2004/audio_samples.html`.

All the experiments performed in this paper were implemented in code running under Octave [14], a high-level language for numerical computations. Specifically, Octave version 2.1.57 was used along with the Octave-Forge [15] distribution 2004.02.12. All the code used for this paper is available either

locally with this document as program_code.html or on the web as http://jnuzman.github.io/audio-restoration-2004/program_code.html.

A gzip'ed tar archive, including this document along with all audio samples and program code, is available at http://jnuzman.github.io/audio-restoration-2004/audio.tar.gz.

# References


[1] Hoyt Axton (performer), "They've Been On Their Jobs Too Long," *Saturday's Child*, Vee-Jay Records, Los Angeles, CA, VJLP-1127, 1963. Phonograph record.

[2] Norman Blake (performer), "You Are My Sunshine," *O Brother, Where Art Thou?*, Mercury Records, Universal City, CA, 088 170 069-2, 2000. Compact disc.

[3] O. Cappé, "Elimination of the musical noise phenomenon with the Ephraim and Malah noise suppressor," *IEEE Transactions on Speech and Audio Processing*, 2(2):345-349, 1994.

[4] O. Cappé and J. Laroche, "Evaluation of short-time spectral attenuation techniques for the restoration of musical recordings," *IEEE Transactions on Speech and Audio Processing*, 3(1):84-93, 1995.

[5] Judy Collins (performer), "Maid of Constant Sorrow," *A Maid of Constant Sorrow*, Elektra Records, New York, NY, EKS-7209, 1961. Phonograph record.

[6] Bud Dashiel and Travis Edmonson (performers), "Tina," *Bud and Travis*, Liberty Records, Hollywood, CA, LST 7125, 1959. Phonograph record.

[7] Y. Ephraim and D. Malah, "Speech enhancement using optimal nonlinear spectral amplitude estimation," *Proc. International Conference on Acoustics, Speech and Signal Processing*, pp. 1118-1121, 1983.

[8] S.J. Godsill and P.J.W. Rayner, *Digital Audio Restoration*, Springer-Verlag, London, 1998.

[9] R. Hoeldrich and M. Lorber, "Broadband noise reduction based on spectral subtraction," *Proc. IEEE Workshop on Audio and Acoustics*, 1997.

[10] B. Kleiner, R.D. Martin, and D.J. Thomson, "Robust estimation of power spectra," *Journal of the Royal Statistical Society, Series B*, 41(3):313-351, 1979.

[11] Henri Mancini and His Orchestra (performers), "Theme from Hatari!," *Hatari!*, RCA Victor, Hollywood, CA, LSP 2559, 1962. Phonograph record.

[12] Vera Mortensen Nuzman (performer), "Letter Edged in Black," field recording, date unknown. Audio cassette.

[13] Vera Mortensen Nuzman and Karl Leland Nuzman (performers), "The Utah Trail," field recording, date unknown. Audio cassette.

[14] Octave. Computer program.
URL http://www.octave.org/

[15] Octave-Forge. Computer program.
URL http://octave.sourceforge.net/