ACOUSTIC ECHO CANCELLATION IN A CHANNEL WITH RAPIDLY VARYING GAIN

Sumit Basu

Microsoft Research
sumitb@microsoft.com

ABSTRACT

We present a method for performing acoustic echo cancellation in a channel with rapidly varying gain and thus a rapidly varying channel characteristic. This is a situation in which standard AEC approaches perform poorly. Our method involves learning a scale-free channel characteristic ($\tilde{H}$). We then apply this to a windowed version of the signal and remove the *projection* of the transformed signal from the output signal. We also develop a "ramp projection" method that allows for a linear variation in gain within the window. We show results in a telephony application with 3 dB to more than 8 dB of improvement over conventional AEC using the simple projection and an additional 1 dB using the ramp projection.

1. INTRODUCTION

There exists a vast literature on acoustic echo cancellation for slowly-varying channels (see [4] for a survey). However, when a channel has a "hidden" but rapidly-varying gain factor, as can be the case in systems with Automatic Gain Control (AGC), this assumption is no longer valid and conventional methods apply quite poorly. This situation can occur when there is an amplification stage prior to the signal combination (i.e., inside a black box) whose output we do not have access to. We illustrate such a situation in Figure 1.

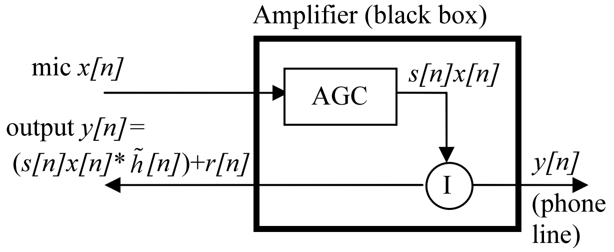

Figure 1. A representative scenario for AEC in the presence of an internal gain. The output of the gain stage is inductively coupled onto the phone line via I and we do not have access to its raw values (s[n]x[n]). The goal is to remove the transformed version of the near-end signal, (s[n]x[n]*h[n]), from the output y[n] to produce a clean version of the remote signal r[n].

In our particular scenario, the microphone signal *x[n]* was coming from a headset microphone and the black box was the headset amplifier that processed the signal and coupled it to the phone line. Thus we had access only to *x[n]* and *y[n]* shown above. The goal was then to remove the effect of the near-end signal *x[n]* from the output *y[n]*, so we could have a clean version of the remote caller's signal *r[n]*. The presence of some kind of gain stage became clear when we drove the input with a modulated sine wave. The input and output are shown in Figure 2 below. Clearly a rapid channel variation is occurring somewhere within the amplifier.

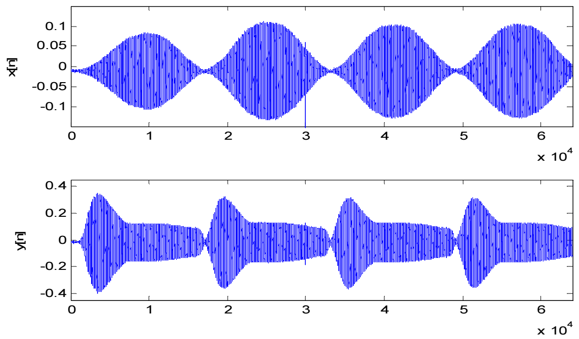

Figure 2. Input signal x[n] and resulting output y[n] in the absence of the remote signal, showing the significant effect of the internal amplifier.

2. MODELING THE CHANNEL

Though there could in fact be an arbitrary channel variation that resulted in the output shown above, it seemed likely that a gain stage could account for much of the variation. We thus modeled the channel *H* between *x[n]* and *y[n]* as the product of a slowly-varying characteristic $\tilde{H}$ and a continuous scaling *s[n]*. In the time domain, this means:

$$y[n] = s[n](\tilde{h}[n] * x[n]) + r[n] \qquad (2.1)$$

where $\tilde{H}$ is modeled as an all-zero filter, i.e.

$$\tilde{H}(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_{N_b} z^{-N_b} \qquad (2.2)$$

Note that we have reversed the order of the gain stage and the channel as a tractable approximation of the real model. We refer to $\tilde{H}$ as the scale-free channel model. This type of scale-free modeling has been applied in other areas of signal processing, such as vector quantization [3]; our work brings this powerful representation to adaptive filtering scenarios.

3. FINDING THE SCALE-FREE CHANNEL MODEL

We first obtain a pair of training signals *x[n]* and *y[n]* in the absence of any remote signal, in order to characterize the transfer function $\tilde{H}$. We then normalize each sample *y[n]* by the norm of the *relevant region of samples from x* that will be used to predict that sample given our transfer function. With the channel model above, this gives us:

$$\tilde{y}[n] = \frac{\| y[n-N_b : n] \|}{\| x[n-N_b : n] \|} y[n] \qquad (3.1)$$

Given this normalized signal, we can now fit the filter coefficients *b* using the method of least squares [1]:

$$\begin{bmatrix} x[0] & \dots & x[N_b-1] \\ x[1] & \dots & x[N_b] \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} b_{N_b} \\ \vdots \\ b_0 \end{bmatrix} = \begin{bmatrix} \tilde{y}[N_b-1] \\ \tilde{y}[N_b] \\ \vdots \end{bmatrix} \qquad (3.2)$$

which we can express as $Xb = y$. We can then solve for *b*:

$$b = (X^T X)^{-1} y \qquad (3.3)$$

The length of the training signal pair can vary, but should be at least an order of magnitude longer than the number of filter taps.

4. APPLYING THE MODEL

Once we have determined $\tilde{H}$, we can predict the contribution to *y[n]* from *x[n]* in a scale-free sense:

$$\hat{y}[n] = s[n](x[n] * \tilde{h}[n]) \qquad (4.1)$$

where *r[n]* is the remote signal, but we are still in need of *s[n]*. We also have the additional assumption that *s[n]* varies more slowly than y[n] itself (though much faster than $\tilde{H}$). Otherwise, we could trivially find an *s[n]* that would completely eliminate *y[n]*, but would eliminate *r[n]* as well.

In this section, we present two ways to model and estimate *s[n]*. Both methods have the same basic approach: break *y[n]* and *x[n]* into windows, then remove the best fit of *x[n]*h[n]* from *y[n]*. The two methods are (1) ordinary vector projection, which models *s[n]* as piecewise constant over windows, and (2) "ramp projection," which models *s[n]* as piecewise linear over windows.

4.1. Ordinary projection

Consider that we have a window of *W* samples from *x[n]*h[n],* which we will call $\tilde{y}[n]$, and the corresponding *W* samples of *y[n]*. Let us model *s[n]* as being constant over this region. We thus want to find the scale factor α such that $\alpha\tilde{y}[n]$ is as close to *y[n]* as possible in a least-squares sense, i.e.

$$\min_{\alpha} \sum_{n=0}^{W-1} (y[n] - \alpha\tilde{y}[n])^2 \qquad (4.2)$$

If we expand this expression and take the derivative with respect to α, we find there is a single minimum at

$$\alpha = \frac{\sum y[n]\tilde{y}[n]}{\sum \tilde{y}^2[n]} = \frac{<y,\tilde{y}>}{<y,y>} \qquad (4.3)$$

This is the familiar vector projection of *y[n]* onto $\tilde{y}[n]$.

4.2. Ramp projection

The problem with the ordinary projection is that *s[n]* may in fact be changing over the course of the window – in the example of Figure 2, it is quite clear that the gain is changing continuously. We thus introduce the ramp projection, which allows s[n] to be linear within a window, starting at some offset α and rising with slope β. This leads to the following minimization problem:

$$\min_{\alpha,\beta} \sum_{n=0}^{W-1} (y[n] - (\alpha + \beta n)\tilde{y}[n])^2 \qquad (4.4)$$

When we expand this and take derivatives, we again find that there is a global minimum, found by solving the following system of equations:

$$\begin{bmatrix} \sum \tilde{y}^2[n] & \sum n\tilde{y}^2[n] \\ \sum n\tilde{y}^2[n] & \sum n^2\tilde{y}^2[n] \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \sum \tilde{y}[n]y[n] \\ \sum n\tilde{y}[n]y[n] \end{bmatrix} \qquad (4.5)$$

4.3. Choosing an appropriate window length

Intuitively, the shorter the window length, the better the fit of $\tilde{y}[n]$ to *y[n]* will be. However, this comes at a price: shorter windows will result in more abrupt changes in the fit signal $s[n]\tilde{y}[n]$. This is a subtle issue, since in the absence of a remote signal *r[n]*, this will often result in further reducing the power of the residual, which is the desired effect. However, when *r[n]* is present, shorter windows will lead to greater distortion of the remote signal. This is because shorter windows lead to a greater number of degrees of freedom in $s[n]$, i.e., the size of the pieces of our piecewise constant or piecewise linear estimate. Since the minimization is trying to cancel $y[n]$, this method will attempt to cancel $r[n]$ along with the transformed version of $x[n]$, as we will see in section 5.3.

4.4. Example

To illustrate the operation of the projection and ramped projection operators, we present a simple example of a modulated sine wave that we are trying to fit over several windows using both techniques. In Figure 3 below, we have set $x[n]$ to be a fixed frequency sine wave, $\tilde{H}$ is identity, $r[n]=0$, and $s[n]$ and $y[n]$ are shown below.

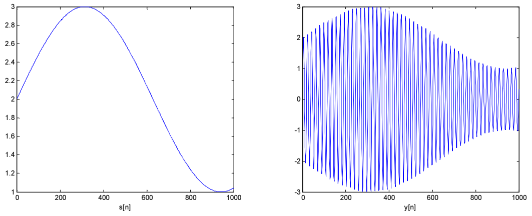

Figure 3. Scaling function s[n] and the resulting y[n]=x[n]s[n]. x[n] is a sine wave at a fixed frequency.

We choose a window length W of 150 samples for illustrative purposes, and then apply the techniques above to fit $y[n]$ with the carrier signal $x[n]$ (which equals $\tilde{y}[n]$ since $\tilde{H}$ is identity). In Figure 4 below, we can see how the projection method chose the best piecewise constant scalefactor for each window, though this still results in a distorted final signal..

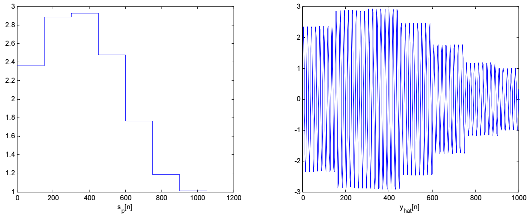

Figure 4. Estimated s[n] and resulting fit to y[n] using ordinary projection with windows of 150 samples.

In Figure 5 below, we can see how the piecewise linear model of $s[n]$ found via ramp projection provides a much better fit, and though the original scaling function was sinusoidal, the resulting signal is very close to the original, as well as a great deal smoother.

5. RESULTS

All of the results in this section are in the context of the telephony setup described above. Specifically, $x[n]$ is coming from a headset microphone, and $y[n]$ is the output on the phone line, which contains the output of the headset amplifier and the remote speaker $r[n]$. All audio was sampled at 16 kHz.

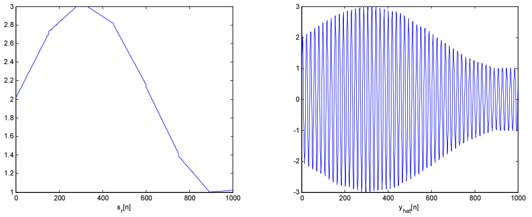

Figure 5. Estimated s[n] and resulting fit to y[n] using ramp projection with windows of 150 samples.

5.1. Examining s[n] for the modulated sine wave

We begin by examining the modulated sine wave from Figure 2. We fit the signal with windows of 1000 samples and show the resulting estimated $s[n]$ in Figure 6 below.

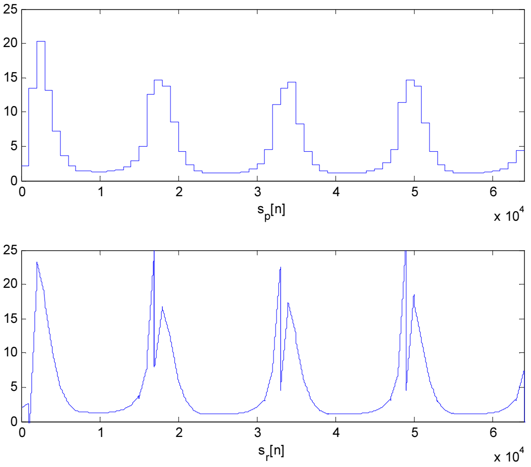

Figure 6. Estimated s[n] for the modulated sine signal sent through the telephony system via ordinary projection (top) and ramp projection (bottom).

Note that the real system is very much like our illustrative example: the scaling function is continuously varying, and the ramp projection is able to make a smoother fit. This is borne out by the results in the table below: the simple projection yields an SNR improvement (original phone signal vs. canceled signal) of 15.6 dB, while the ramped projection yields 22.2 dB ($r[n]$=0). At the same time, the sharp variations at the peaks of the estimated $s[n]$ are cause for some concern – it means that the estimate is not continuous at those points. We investigate this by trying the simple projection with a smaller window size (W =100) in Figure 7 below.

The discontinuities now appear in the simple projection method as well, implying that the actual gain function does contain discontinuities, which were being accurately modeled by the ramp projection with a much larger window size. Note that this smaller window size gives us an SNR improvement of 22.3 dB, but with a window size this short it is likely we would distort the remote signal *r[n]*, as we will show in Section 5.3.

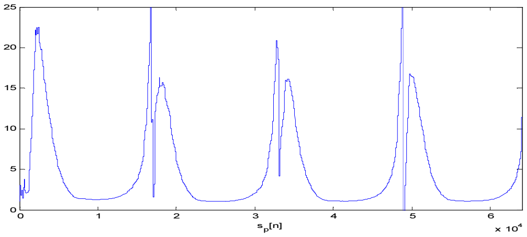

Figure 7. The estimate of s[n] using simple projection where W=100. Note that the discontinuities still appear.

5.2. SNR improvement vs. method

In Table 1 below, we show the improvement in the SNR in the presence of a remote signal for various methods: a fixed channel model, the NLMS (Normalized Least Mean Squares) method [2], and our method using both simple and ramp projection. We looked at two signal scenarios: the modulated sine wave and speech input. In all cases, we added a known remote signal *r[n]* containing speech recorded from the channel.

The filter $\tilde{H}$ had 201 taps (100 causal, 100 anticausal) and was trained *once* on 64000 samples of speech, and the resulting filter was used for all the experiments below. A window size of 1000 samples was used for the projection methods. We define the SNR as:

$$SNR = 10 \log \| r[n] \|^2 / \| (y - \hat{y}[n]) \|^2 \quad (5.1)$$

Note that the line itself is noisy even in the absence of any local or remote signal ("silence"), and the maximum SNR improvement (signal power vs. "silence" power) would be 17.7 dB.

The "fast" vs. "slow" versions of NMLS correspond to setting μ to the maximum prescribed value in [1] (2.06) vs. a more conservative value (1e-3). Note that the latter would more typical, as most systems reduce or event stop adaptation (i.e., use a fixed *H*) when a remote signal is present. In this case, our methods would outperform NLMS by more than 8 dB. In addition to the numerical results, there was a substantial perceptual improvement when using our projection methods.

	Modulated Sine	Speech
Fixed *H*	-2.8	0.2
NLMS (slow)	1.7	1.1
NLMS (fast)	2.7	6.1
Simple Projection	11.3	9.7
Ramp Projection	12.2	10.8

Table 1. SNR improvement (dB) for various algorithms. Note that the maximum improvement would be 17.7 dB.

5.3. SNR improvement vs. window size

In this section, we examine the effect of window size on the SNR reduction. The results are shown in Figure 8 below; note that these results are averaged over a smaller sample than Table 1 above.

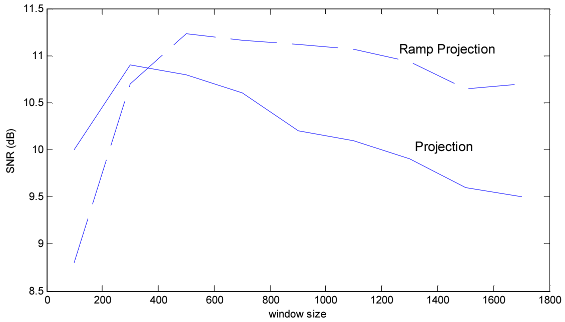

Figure 8. SNR vs. window size for ordinary projection (solid line) and ramp projection (dashed line).

As expected, we see that the distortion will increase with windows that are too small (due to overly aggressive fitting) and too large (due to a poor estimate of *s[n]*). Also note, though, how much more robust the performance of the ramped projection is with respect to window size, due to its greater flexibility in modeling *s[n]* as seen in Section 5.1.

6. DISCUSSION

We have presented a method for doing acoustic echo cancellation in the presence of a rapidly varying gain via estimating a scale-free channel model $\tilde{H}$ and then performing a window by window projection (ordinary or ramp) onto the target signal *y[n]*. While we have presented this method in the context of signal separation in a telephony system, we expect there are many other applications where there is an unknown gain stage whose pre-combination output cannot be accessed.

7. REFERENCES

[1] S. Haykin, *Adaptive Filter Theory, Fourth Edition,* Prentice-Hall, Englewood Cliffs, NJ, 2002.

[2] J. Nagumo and A. Noda, "A learning method for system ID," *IEEE Trans. Autom. Control*, vol. AC-12, pp. 282-287. 1967.

[3] M. Sabin and R. Gray. "Product Code Vector Quantizers for Waveform and Voice Coding." IEEE Trans on Acoustics, Speech, and Sig. Proc. 32(3): pp. 474-488. Jun. 1984.

[4] R. Storn, "Echo cancellation techniques for multimedia applications – a survey," *Int'l Comp. Sci Inst. TR-96-046*, Berkeley, CA, Nov. 1996.