AUDIO RESTORATION: CLICK REMOVAL

Junhao Wang
McGill University
MUMT 501 Final Project

1. INTRODUCTION

In audio signal processing, the term "click" refers to a type of localized finite-duration degradation present in various types of audio media. Clicks can occur at random positions in the audio signal. They differ from global audio degradation such as background noise and hiss noise in that clicks belong to impulsive noise that only affect certain parts of the waveform while the global noise affects all samples. In most situations, audio recordings corrupted by impulsive noise have at least 90% of the samples unaffected [1]. This makes it possible to detect and remove the clicks using the surrounding undegraded samples.

The objective of this project is to study and implement the click removal algorithm proposed by [3]. Specifically, the click removal algorithm incorporates two components: detection [3] and interpolation [4]. In the detection process, the degraded samples are identified and located in the mixture. Then, in the interpolation process, the values of detected noise samples are replaced with more appropriate values, which are obtained by interpolation. The whole algorithm works under the assumption that the underlying audio signal can be modeled by short-term stationary autoregressive processes.

2. DETECTION SIGNAL

Clicks can be detected by identifying outliers in the waveform. The audio signal can typically be modeled accurately as an autoregressive process. In this case, the corrupted audio signal x is considered a mixture of the original signal s and the impulsive noise n

$$x_t = s_t + n_t. \tag{1}$$

Potential background noise and other non-impulsive noise elements are ignored, as we only focus on detecting clicks. It is assumed that the original clean signal s is drawn from a locally stationary autoregressive process, where each sample can be modeled as a linear combination of the p preceding samples using a known set of autoregressive coefficients $\mathbf{a} = [a_1, \ldots, a_p]$ and white noise (residual error) e_t

$$s_t = -\sum_{k=1}^{p} a_k s_{t-k} + e_t. \tag{2}$$

In practice, the autoregressive parameters $\mathbf{a}$ and the variance σ_e^2 of the excitation signal e are unknown and must be estimated from the mixture x. A robust algorithm is thus needed to estimate the correct autoregressive parameters from the corrupted signal. In this project, Yule-Walker equations are used, which exploit the relationship between autoregressive parameters and the autocorrelation function. If the autoregressive model is fitted to the mixture x, large errors would occur at the degraded samples because the click samples are outliers, which are unrelated to their neighbors. Combining Eqn (1) and Eqn (2), the mixture x can be rewritten, and the noise terms can be grouped together

$$x_t = -\sum_{k=1}^{p} a_k s_{t-k} + e_t + n_t, \tag{3}$$

$$x_t = -\sum_{k=1}^{p} a_k \left(x_{t-k} - n_{t-k}\right) + e_t + n_t, \tag{4}$$

$$d_t = x_t + \sum_{k=1}^{p} a_k x_{t-k} = e_t + n_t + \sum_{k=1}^{p} a_k n_{t-k}. \tag{5}$$

As the excitation signal e is random and generally much smaller than the original signal s, the term d_t is a good measure of the noise. The detection signal $|d_t|$ will take on large values at noisy samples and small values otherwise. In particular, for a given sample x_t, if the previous p samples are all undegraded, d_t is exactly the sum of n_t and e_t. Contrarily, if impulsive noise is present in the previous p samples, the impulsive noise will propagate and affect the detection accuracy. Depending on their values, noisy samples in the same vicinity may build up constructively or cancel each other out, leading to false positives or false negatives in the detection.

To visualize the detection signal, an experiment was conducted with clean audio signal and artificial impulsive noise. 2,000 samples were taken from an audio file and $N_{max} = 50$ samples of gaussian noise were added to the middle of the signal. The order p of the autoregressive model was set to $3 * N_{max} + 2 = 152$ according to empirical results in [2]. The clean signal, signal with artificial noise, and the detection signal computed from the mixture are shown in Figure 1.

3. THRESHOLDING

As seen in Figure 1, the magnitude of the detection signal $|d_t|$ generally reflects the level of the noise. We can roughly locate the degraded samples by naively imposing a threshold λ on the detection signal. The aim is then to

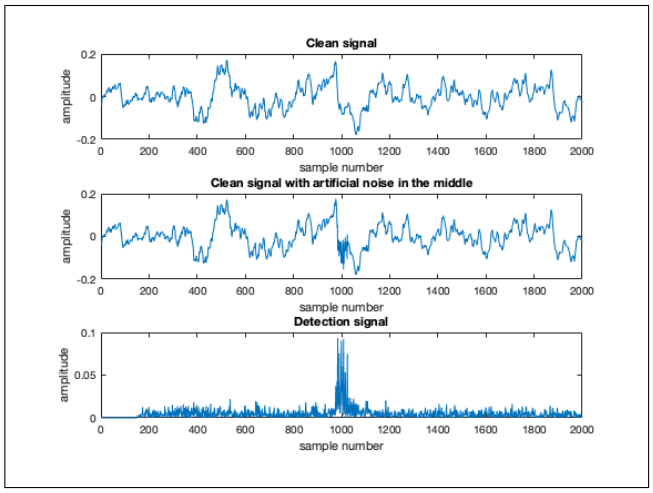

Figure 1. Clean signal with artificial clicks

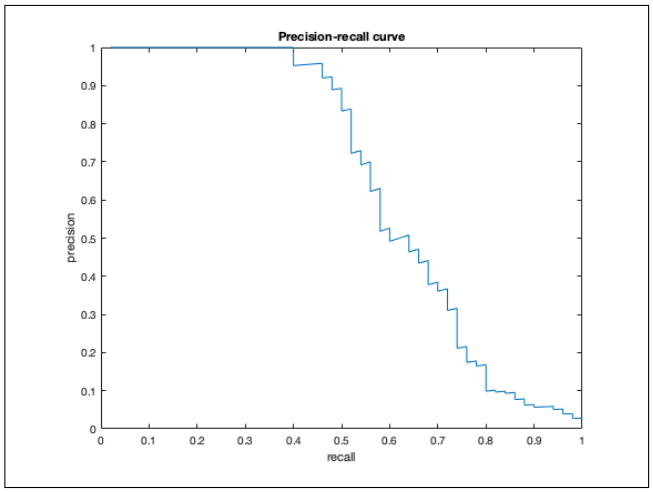

Figure 2. Precision-recall curve obtained by direct thresholding

find the best threshold that gives the best detection performance. As this is a binary classification problem, *precision* and *recall* are good metrics suitable for this task. Precision is the ratio between the number of true positives and detected positives, and recall is the ratio between the number of true positives and actual positives. In selecting the threshold, there is a trade-off between high precision and high recall. In a perfect detection algorithm, the precision and recall should both be 1.

3.1 Direct Thresholding

A wide range of threshold values λ_K are tested.

$$\lambda_K = K\sigma_e, \tag{6}$$

where σ_e is the standard deviation estimated in the previous section, and K lies between 0 and 12 with an increment of 10^{-4}. For each value of K, a threshold is computed and imposed to the detection signal. By comparing the prediction and the ground-truth labels, the precision and recall curve is obtained and plotted in Figure 2. Obviously, with the best compromise around 0.6 for both precision and recall, direct thresholding did not achieve satisfactory performance. There are two reasons why direct thresholding does not work well: First, even though the burst region of detection signal generally shows high amplitudes, low values do exist within the same region, which are probably caused by destructive interference. This leads to false negatives. Second, the noisy samples may give rise to the following p samples, due to the nature of autoregressive process. This leads to false positives.

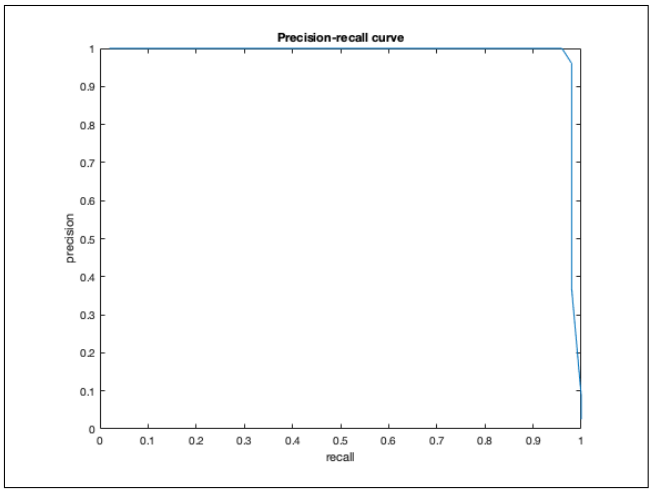

Figure 3. Precision-recall curve obtained by thresholding with post-processing

3.2 Thresholding with Post-processing

To alleviate this to some extent, knowledge about the nature of impulsive noise is helpful. In real-world recordings, clicks often occur in groups. It is rare for a single impulse to appear in the corrupted waveform. In fact, clicks often have some finite width between 5 and 100 samples at a 44.1kHz sampling rate, which corresponds to the width of physical scratch or irregularity in the recorded medium [1]. Therefore, it is reasonable to assume that the samples within a certain range around a detected noise sample also belong to the noise. In the previous example, as we know that there is only one burst in the waveform, we can take the first and the last sample that exceeds the threshold and assume all samples in between are noise. Using this strategy, a new precision-recall curve is obtained as shown in Figure 3. Now the algorithm achieves almost perfect performance, with a precision of 1.0 and recall of 0.96.

However, in real-world applications, multiple bursts could be present in the audio signal (e.g., Figure 4). In this case, it's impractical to assume all samples between the first and last noise sample belong to noise. Alternatively, using a similar idea, a fusion parameter b is defined, which measures the maximum number of consecutive samples within a burst whose values are lower than the threshold [3]. For any two samples where the detection signal exceeds the threshold, if the distance between them is smaller than b samples, all samples between them are considered noise.

In summary, the detection of clicks is controlled by the

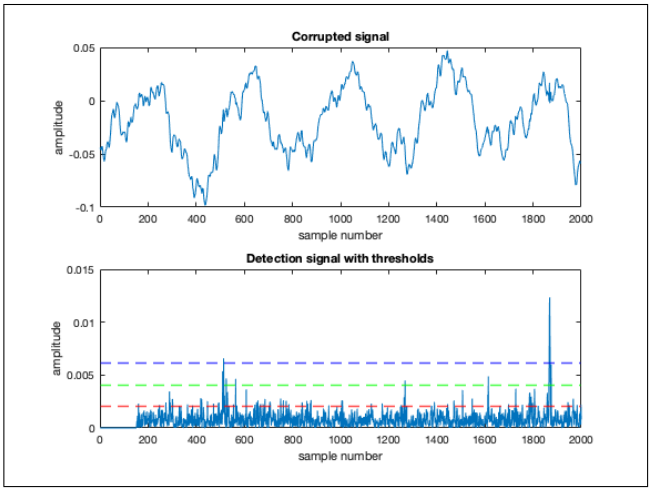

Figure 4. Real audio example with multiple bursts

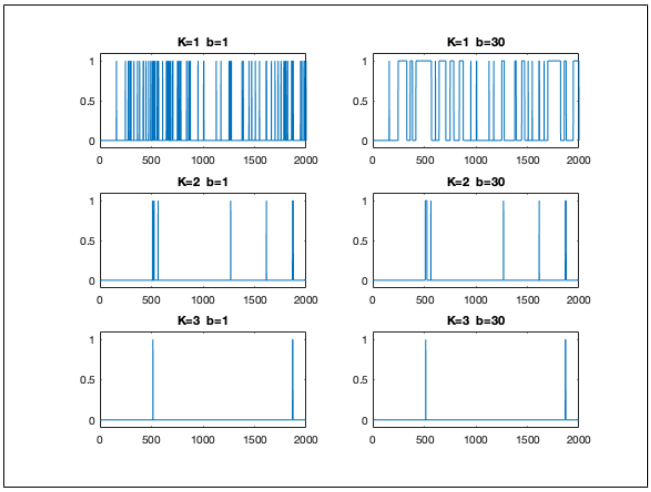

Figure 5. Detected clicks using different parameter values

threshold parameter K and the fusion parameter b. For the same signal, different K and b values would result in different number of detected clicks. Figure 5 illustrates the position and number of detected clicks using different parameter combinations. A value of 1 on the vertical axis corresponds to a detected click. Generally, a higher threshold leads to less positive detection. Using larger value for b tends to merge together the bursts that are close to each other. In experimenting with these two parameters, it is found that the actual detection performance depends heavily on the signal itself. Different signals would require different set of parameters to accurately detect the bursts. Fine-tuning these parameters is crucial for satisfactory click removal.

4. INTERPOLATION

After determining the positions of degraded samples, the declicked signal can be constructed by replacing the degraded samples with more appropriate values. For estimating these values, it is assumed that the degraded samples do not contain any information about the underlying signal. Therefore, missing samples at the degradation can be estimated by interpolation based on the surrounding samples.

The interpolation scheme based on autoregressive modeling [2] [4] can be applied in this case because two conditions are satisfied: First, the positions of the missing samples are known (estimated at the detection step). Second, the missing samples are surrounded by a sufficient number of known (undegraded) samples. Using the same order and autoregressive parameters estimated at the detection step, the missing samples can be easily estimated.

As an example, a short segment of corrupted signal is shown in Figure 6, along with the detection function, detected clicks, and reconstructed signal. The predicted values for samples at the detected clicks fit well into the overall structure of the waveform, leading to little audible distortion.

5. EXPERIMENTS ON LONGER SIGNALS

For testing the algorithm in practice, three real-world music recordings are selected [1] . The recordings are corrupted by various levels of impulsive noise from small crackles to large bursts. In the following, we only discuss experiments performed on the classical music excerpt by Mussorgsky, as it is the same corrupted extract as used in [3].

5.1 Overlapping Frames

Since the autoregressive assumption is only valid on a local scale, the audio track is first split into overlapping frames with a frame length of N_w and a hop size of N_h. The same parameter settings are used as [3], where $N_h = N_w/4$, corresponding to a 75% overlap. Every frame is processed by the detection and interpolation procedures discussed in the previous sections. To reconstruct the signal, each processed frame is multiplied by a Hamming window of size N_w and added iteratively with a 75% overlap. In order to perfectly reconstruct the signal, the hamming window is scaled to

$$w(k) = \frac{1}{4 \times 0.54}\left(0.54 - 0.46\cos\left(2\pi\frac{k-1}{N_w}\right)\right). \quad (7)$$

5.2 Parameters

There are three parameters to tune in this click removal algorithm: the fusion parameter b, the threshold parameter K, and the number of iterations I. It is claimed in [3] that $K = 2$ and $b = 20$ yield good performance in various types of audio signals, and iterating the algorithm several times generally improves the result. In this section, we explore different values for these parameters. For other parameters such as N_{max} and p, we directly use the same setting as [3]. Since there are no ground-truth annotations for the audio files, it is hard to quantitatively evaluate the performance of the algorithm and to search for the best parameter combinations accordingly. Therefore, we rely primarily on subjective hearing tests to evaluate the effectiveness of this algorithm.

[1] http://www-sigproc.eng.cam.ac.uk/Main/SJGSpringer

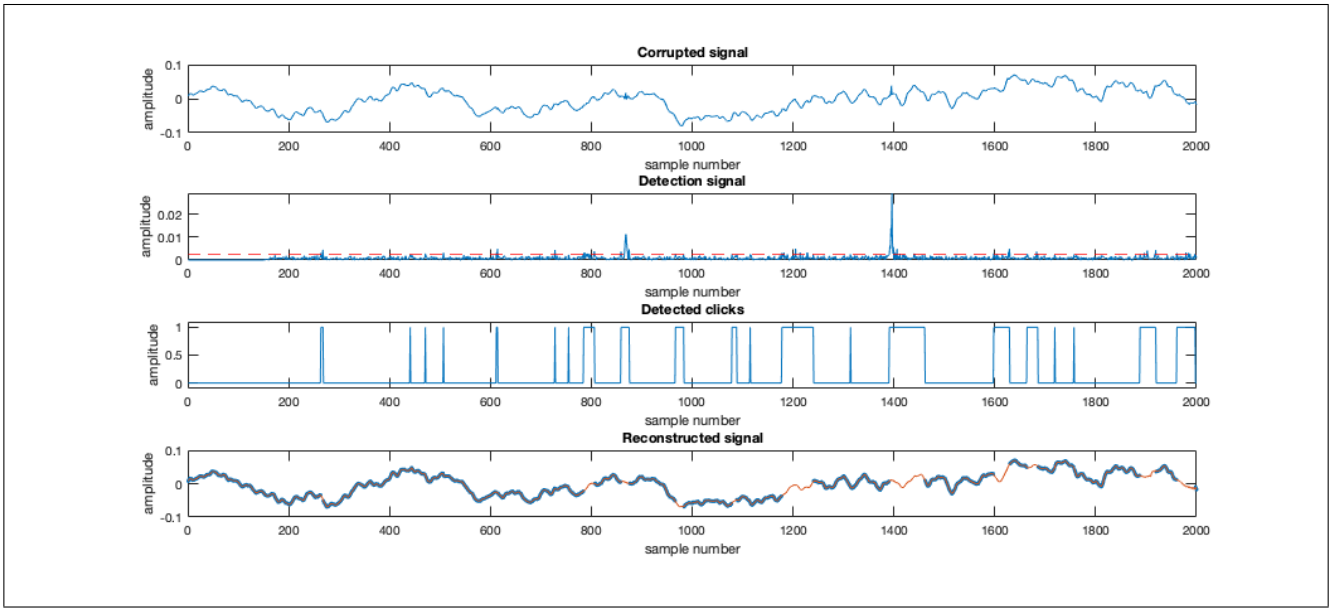

Figure 6. A corrupted signal with the detection function and the interpolation result

5.2.1 Threshold parameter K

The threshold parameter controls the threshold. The higher the K, the higher the threshold and thus the lower detection sensitivity. If the threshold is set too high, few samples will exceed the threshold and clicks remain in the signal. If the threshold is set too low, normal samples would be detected as noise and distortion may occur. Figure 7 shows the 36-second classical music excerpt by Mussorgsky, and its declicked versions using different thresholds. At $K = 0.5$, the signal is clearly distorted from the original signal. At $K = 8$, some impulses remain in the reconstructed signal. These discrepancies are clearly audible in the resulting audio files [2] .

5.2.2 Fusion parameter b

As discussed in section 3, a large fusion parameter b tends to merge together the bursts that are close to each other. Therefore, this parameter controls the length of detected clicks. The same excerpt by Mussorgsky was processed using different values for b and the results are plotted in Figure 8. When using $b = 1$, the detection algorithm is the same as direct thresholding, which did not achieve satisfactory performance in detecting the degraded samples. This is verified by observing the waveform as well as listening to the output. Clicks remain in the signal when using such small value for b. For $b = 20$ and $b = 40$, in this case, there is no significant difference between the two results [3] .

5.2.3 Number of iterations I

The click detection process relies heavily on the estimated autoregressive parameters. Therefore, the whole process can be iterated for several times to attempt for a better result. At the first iteration, the autoregressive parameters are estimated from the corrupted signal. After one iteration, many of the clicks are detected and replaced by more appropriate values. If we compute the autoregressive parameters again on the declicked signal, we will likely end up with better parameters because the declicked signal is cleaner than the corrupted signal that we started with.

Figure 9 shows the same audio example and its declicked versions after one and two iterations ($K = 2, b = 20$). It is obvious that the samples degraded by impulsive noise, which have significantly higher amplitudes than other samples in the vicinity are replaced by values that better fit the waveform in the declicked versions. However, there is no visible difference between the two declicked waveforms. After listening to the three audio files, it was determined that the click removal algorithm effectively removed the impulsive noise and did not introduce audible distortion and artifacts [4] . The difference between the one-iteration and two-iteration versions is very subtle and hardly audible. Therefore, it is reasonable to conclude that in this specific case, one iteration is sufficient for impulsive noise removal.

6. CONCLUSION

In this project, the click removal algorithm proposed by [3] is implemented and explored. A subset of parameters are investigated and tested on real-world music recordings. The algorithm achieved good performance in both detecting and removing the impulsive noise in the audio signal.

The algorithm is simple and works entirely in the time domain. One limitation of this algorithm is that it is not self-adaptive. The right combination of parameters is crucial for good performance, and it may vary from one

[2] Please consult the attached audio files: *MussK0.5b20I1.wav*, *MussK2b20I1.wav*, and *MussK8b20I1.wav*

[3] Please consult the attached audio files: *MussK2b1I1.wav*, *MussK2b20I1.wav*, and *MussK2b40I1.wav*

[4] Please consult the attached audio files: *MussK2b20I1.wav* and *MussK2b20I2.wav*

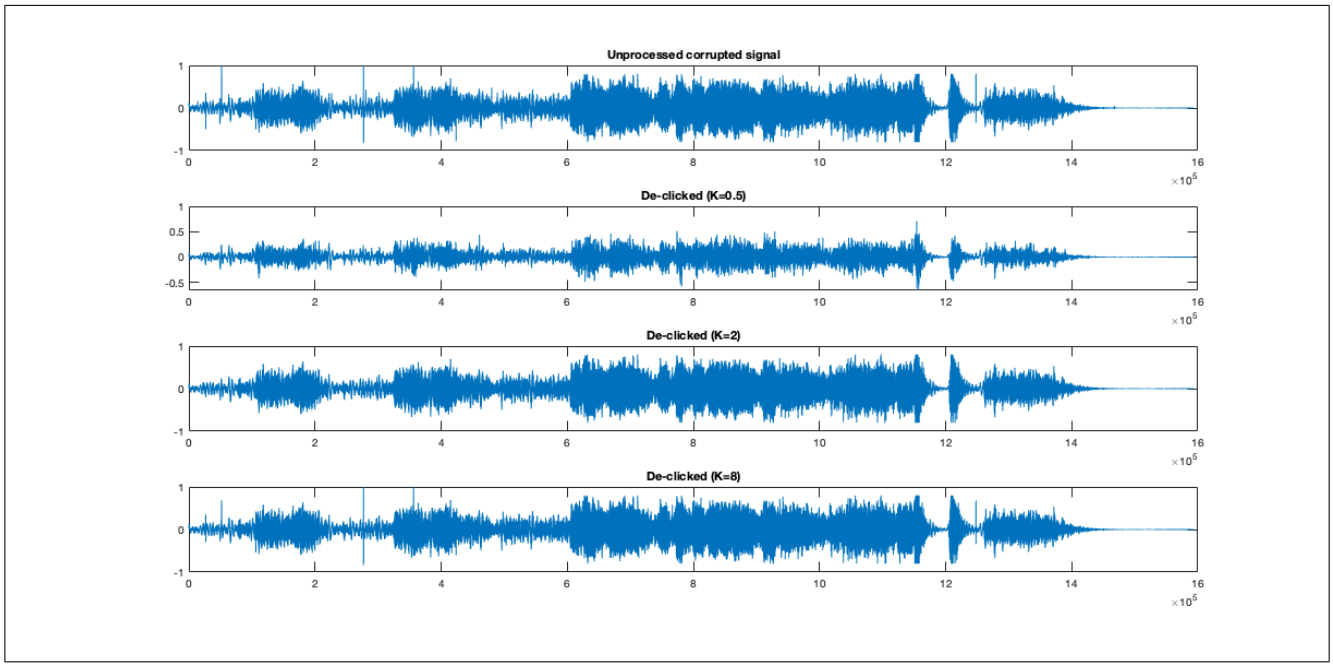

Figure 7. Click removal results using different thresholds

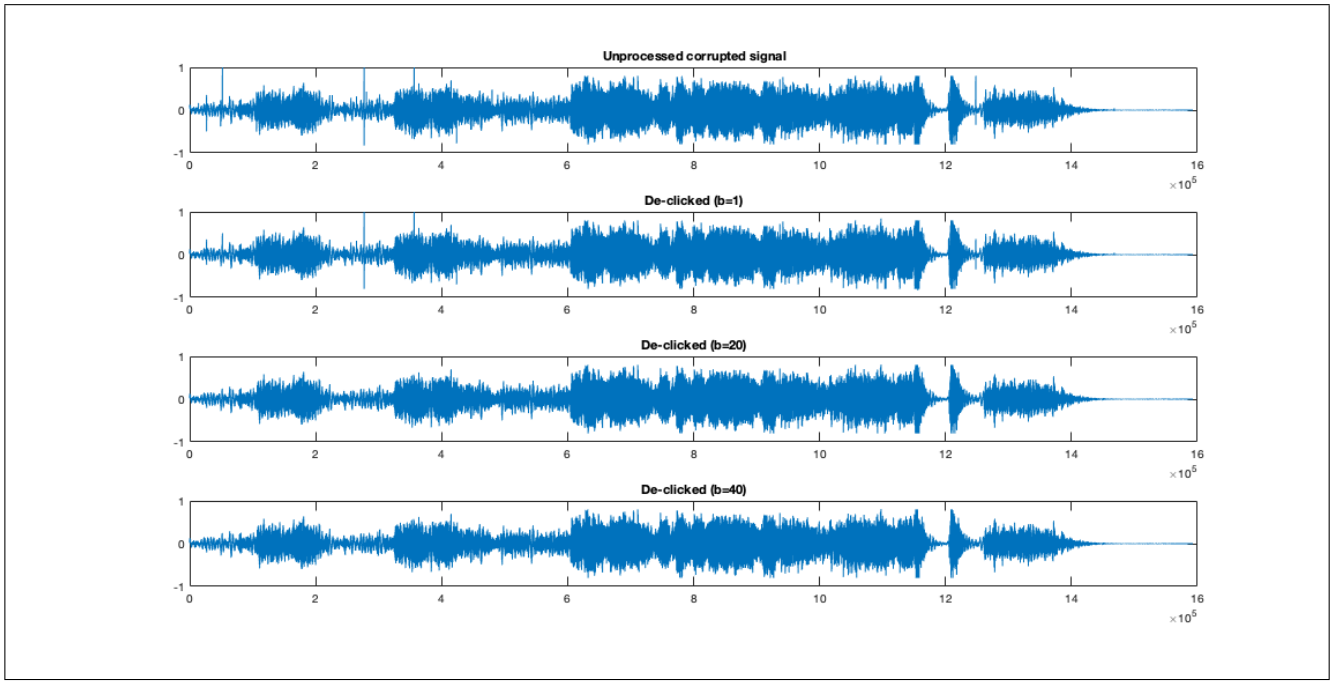

Figure 8. Click removal results using different values for the fusion parameter

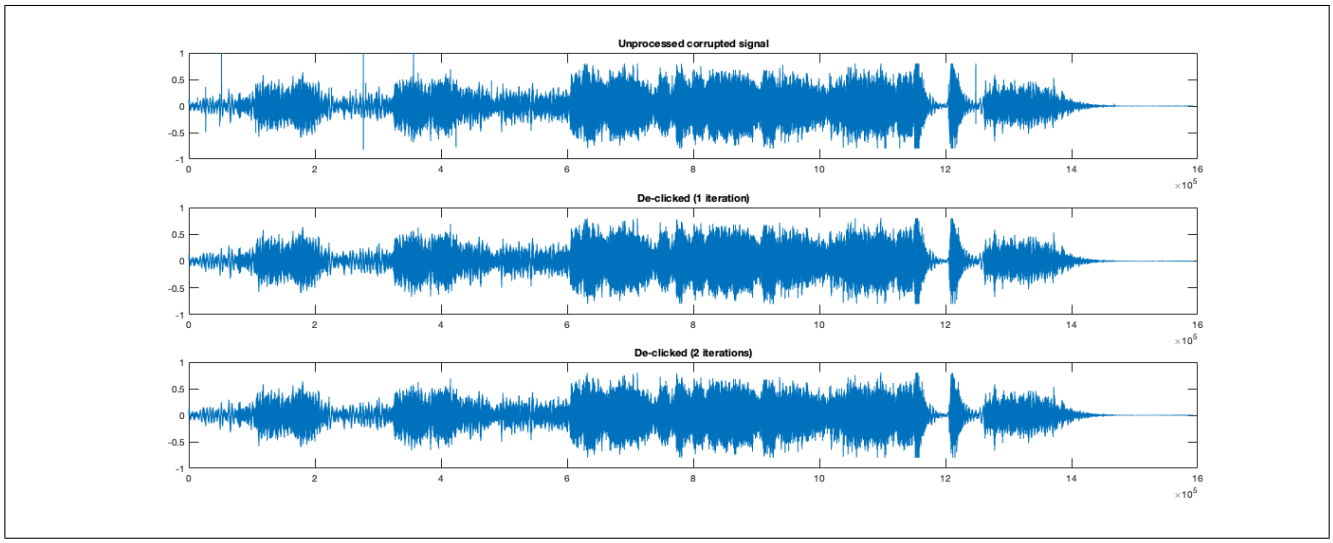

Figure 9. Click removal results using different number of iterations

recording to another. Therefore, for each corrupted recording, the parameter values need to be determined through trial and error.

In implementing this algorithm, the interpolation part was supposed to use the same autoregressive parameters estimated in the detection stage. However, this part of implementation is rather complex and the *fillgaps* function in the signal processing toolbox of MATLAB is used, which works in the same way as proposed by [4] but does not take pre-computed autoregressive parameters as input. This is found to significantly slow down the experiments and to affect the efficiency of the algorithm. Future improvements may focus on reducing the computational complexity and improving the speed performance of the implementation.

7. IMPLEMENTATION DETAILS

This project is implemented entirely in MATLAB. Full source code is available online [5] .

wholeWorkflow.m implements the entire detection and interpolation process in one script. It reads an audio file and write the declicked signal into another one. To make experiments easier, it is functionized in *deClick.m* and called in the master experiment script *main.m*. The other scripts in the repository are for generating the plots presented in this report. Please see *README.md* for more details.

8. REFERENCES

[1] Simon J. Godsill and Peter J. W. Rayner. Removal of Clicks. In Simon J. Godsill and Peter J. W. Rayner, editors, *Digital Audio Restoration: A Statistical Model Based Approach*, pages 99–134. Springer, London, 1998.

[2] Augustus J.E.M. Janssen, Raymond N.J. Veldhuis, and Lodevijk B. Vries. Adaptive interpolation of discrete-time signals that can be modeled as autoregressive processes. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(2):317–330, April 1986.

[3] Laurent Oudre. Automatic Detection and Removal of Impulsive Noise in Audio Signals. *Image Processing On Line*, 5:267–281, November 2015.

[4] Laurent Oudre. Interpolation of Missing Samples in Sound Signals Based on Autoregressive Modeling. *Image Processing On Line*, 8:329–344, October 2018.

[5] https://github.com/jwang44/Impulsive-Noise-Removal