A PRESENTATION ON A TWO-STAGE ALGORITHM FOR ONE-MICROPHONE REVERBERANT SPEECH ENHANCEMENT

By Md. Shamim Hussain (0417062229)

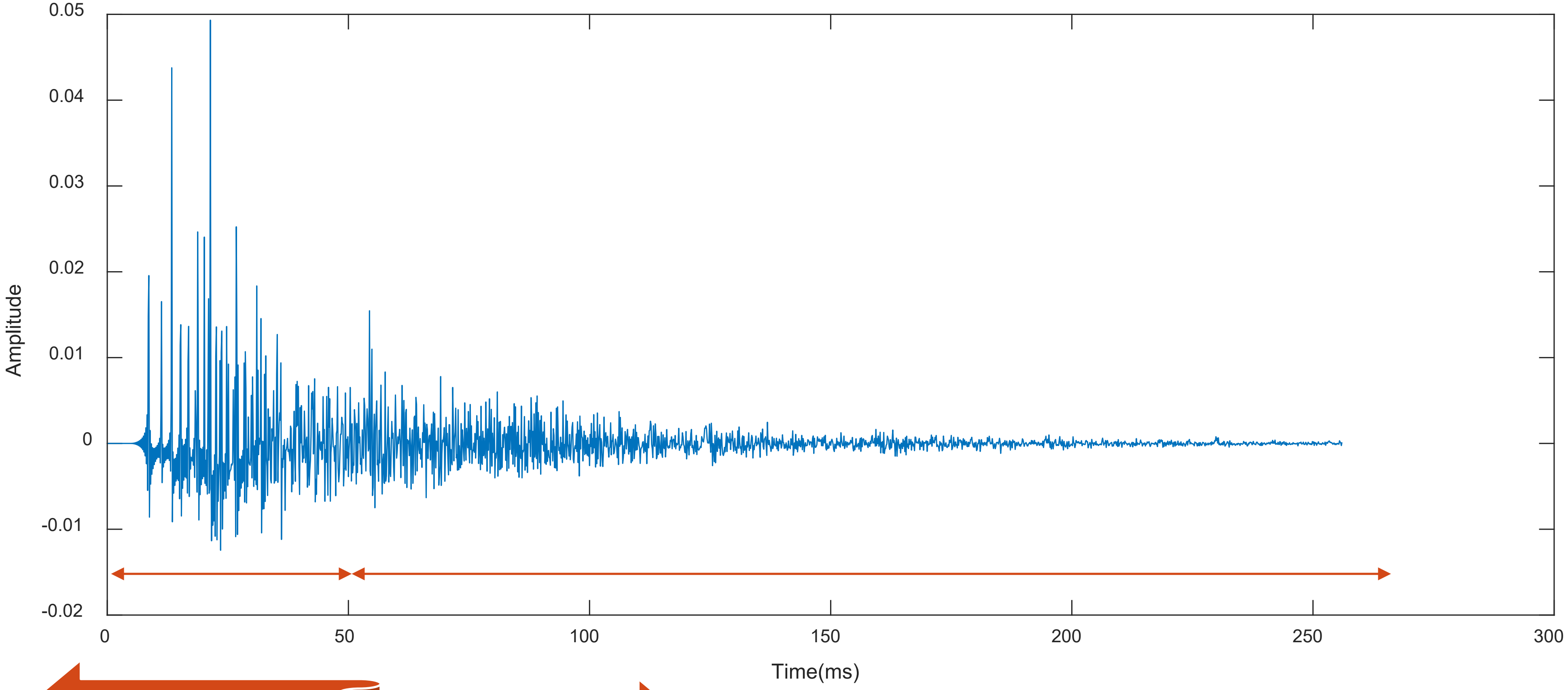
REVERBERATION

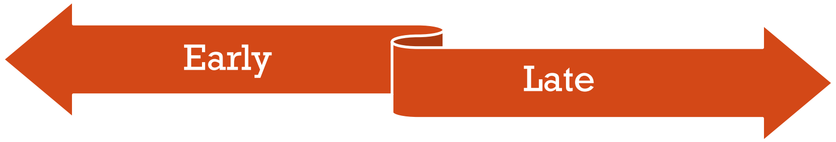
- When the speaker is at a distance from the input device, speech signal *x(n)* is convolved by the room impulse response *h(n)*. *w(n)* is additive noise. The reverberant and noisy speech is modeled as:

$$y(n) = x(n) * h(n) + w(n)$$

- Reverberation causes degradation of speech quality which causes hearing fatigue, reduces intelligibility of speech which causes reduction in efficiency of ASR (Automatic Speech Recognition) and speaker recognition.
- To reduce the degradation of quality, we must try to nullify the effect of room impulse response *h(n)*, which works as a filter on the speech signal.

THE ROOM IMPULSE RESPONSE





THE ROOM IMPULSE RESPONSE

- The early part of the impulse response ($t<50ms$) looks like a train of impulses, which indicate the early reflections in the room.
- The late part of the impulse response looks more random. They are due to the late reflections in the room.
- Due to the different nature of the two parts of the impulse response they produce two types of degradations, namely coloration and long-term reverberation.
- A two-stage algorithm addresses these two effects in two stages. First we design an inverse filter to reduce the effect of early reflections. Next we treat the late part of the reverberant speech like uncorrelated signal and remove its effect by modified spectral subtraction.

LP RESIDUAL

- According to the linear prediction model, the speech signal is the output *y(n)* of an all-pole filter *1/A(z)* excited by *X(n)*

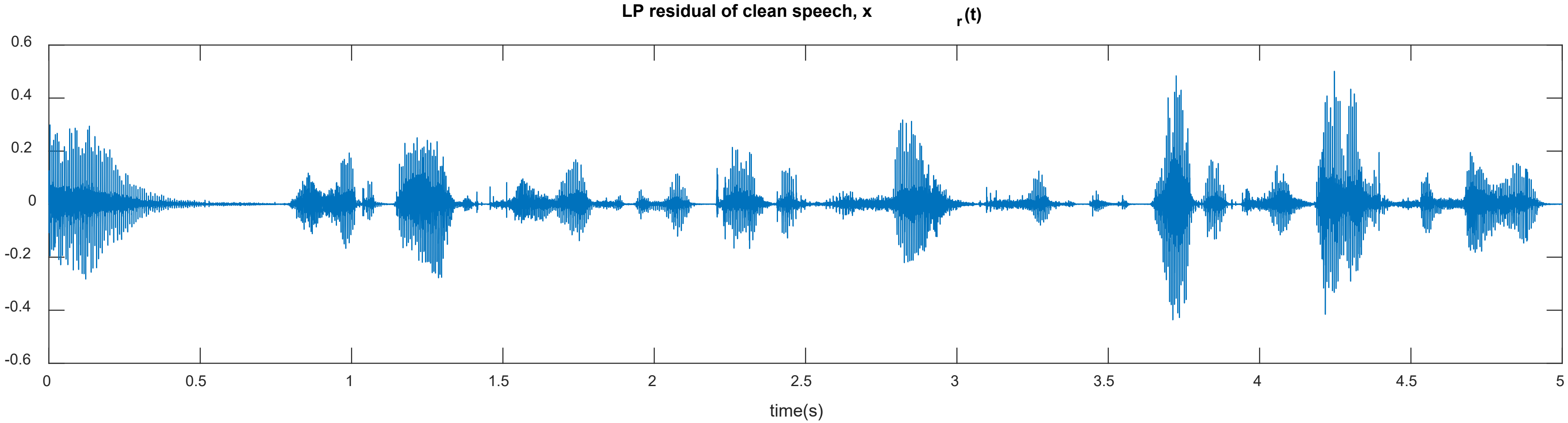
$$Y(z) = \frac{X(z)}{1 + \sum a_k z^{-k}} = X(z)/A(z)$$

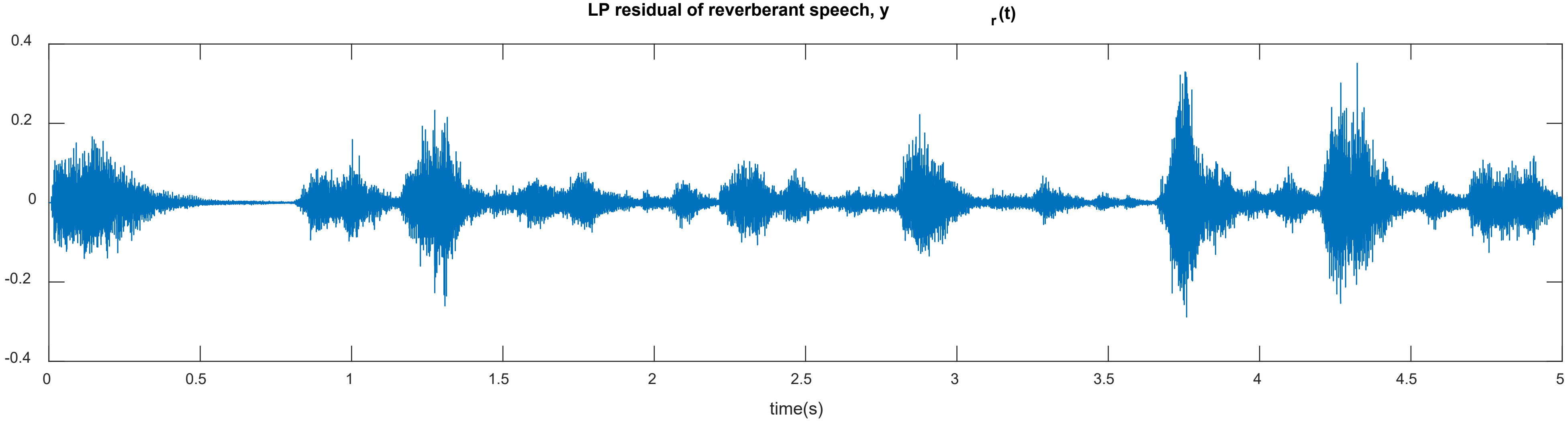
- The excitation signal can be a sequence of pulses(voiced part) or white noise(unvoiced part).
- The linear prediction residual is the inverse filtered signal:

$$y_r(n) = y(n) - \sum a_k y(n-k)$$

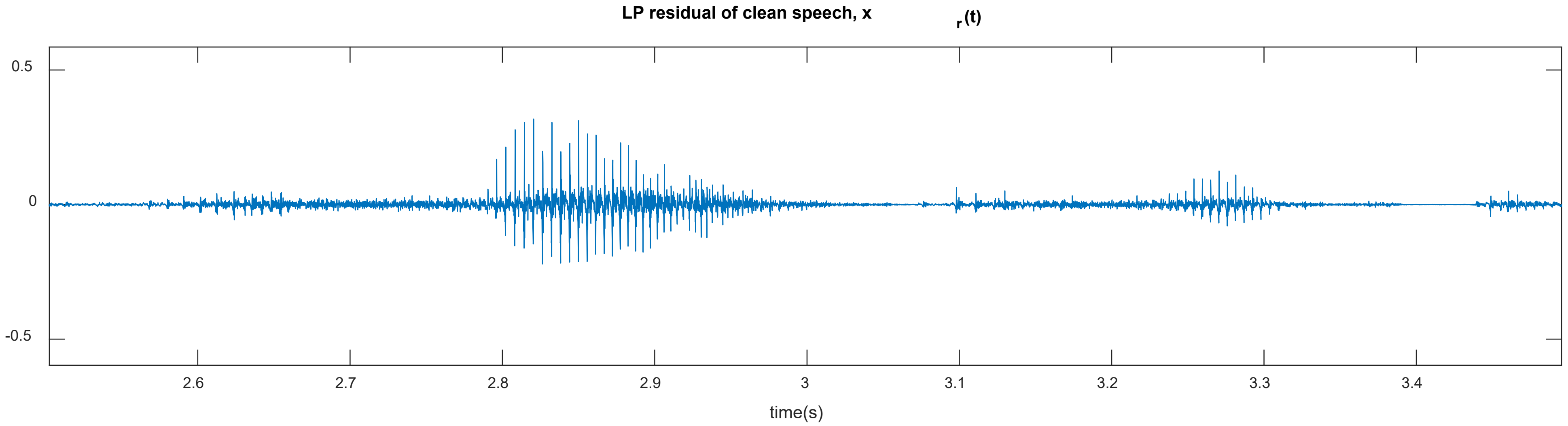
a_k are chosen to minimize the prediction error.

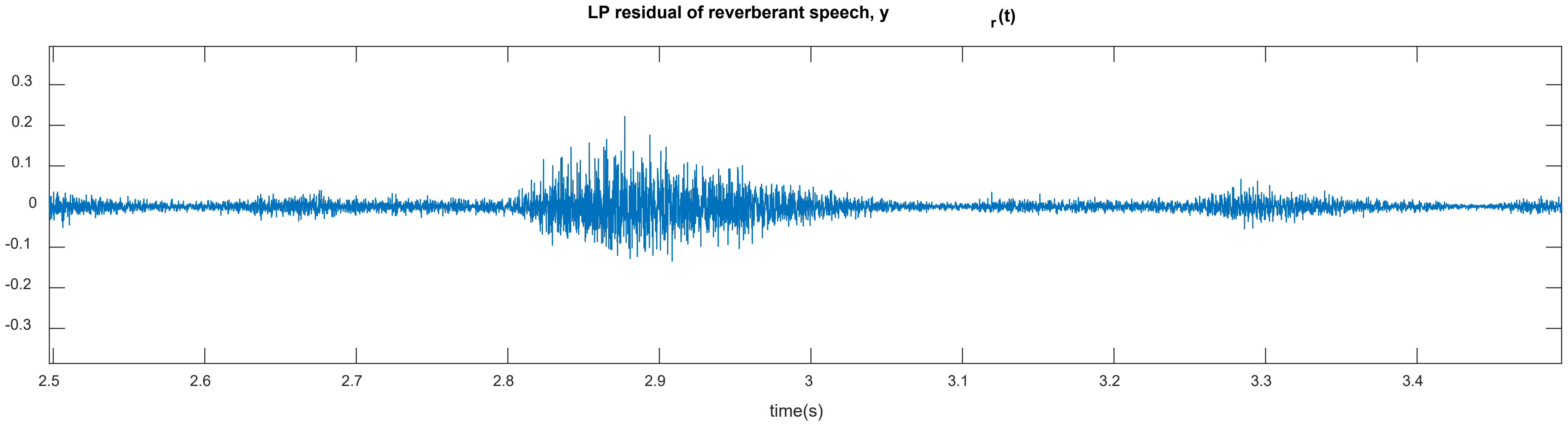
LP RESIDUAL


LP residual of clean speech, $x_r(t)$
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
time(s)


LP residual of reverberant speech, $y_r(t)$
0.4
0.2
0
-0.2
-0.4
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
time(s)

LP RESIDUAL : MAGNIFIED VIEW



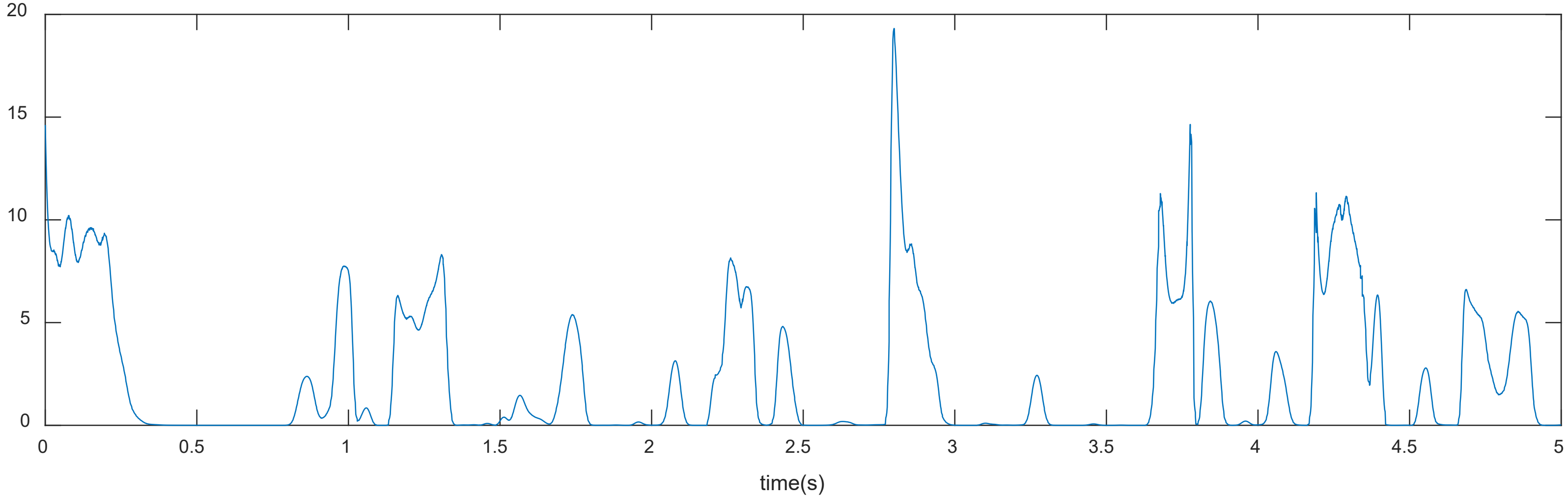


LP RESIDUAL

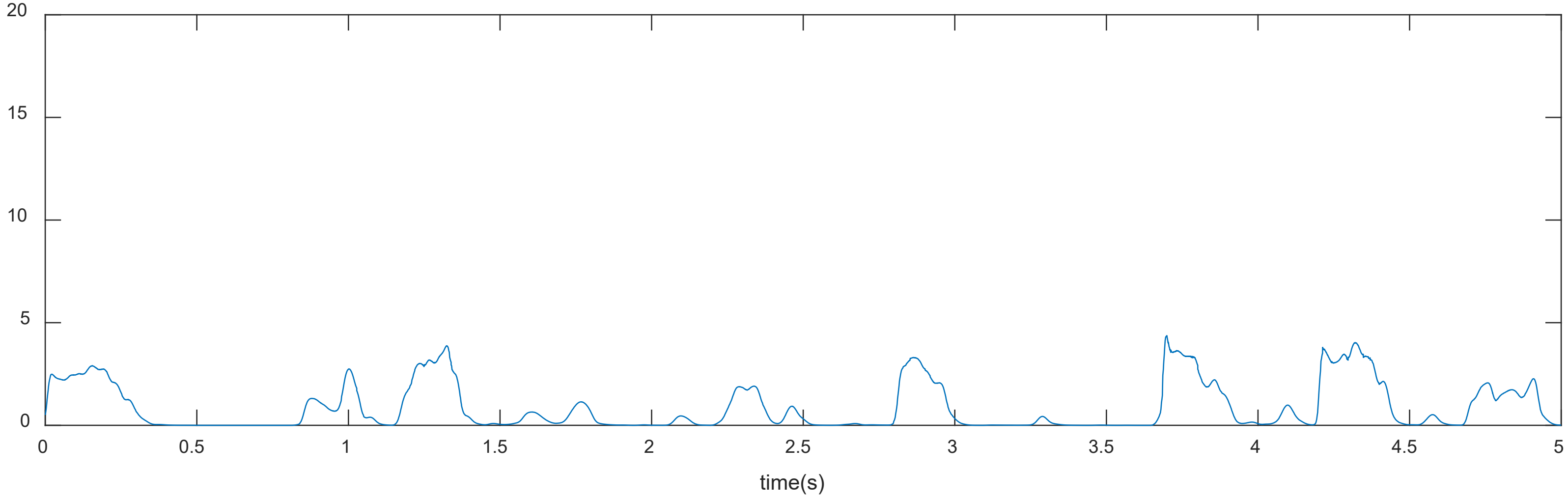
- The LP residual of clean speech is more impulse-train like, the interval between impulses look like damped sinusoids [3].
- The LP residual of reverberant speech is more white noise like, i.e. the peakedness of the clean speech is lost.
- We want to associate this property to a statistical parameter, i.e. kurtosis [3].
- The kurtosis of LP residual is given by

$$Kurt[y_r] = \frac{E\{y_r^4(n)\}}{E^2\{y_r^2(n)\}}$$

KURTOSIS OF LP RESIDUAL



Kurtosis of LP residual of clean speech, kurt(x)



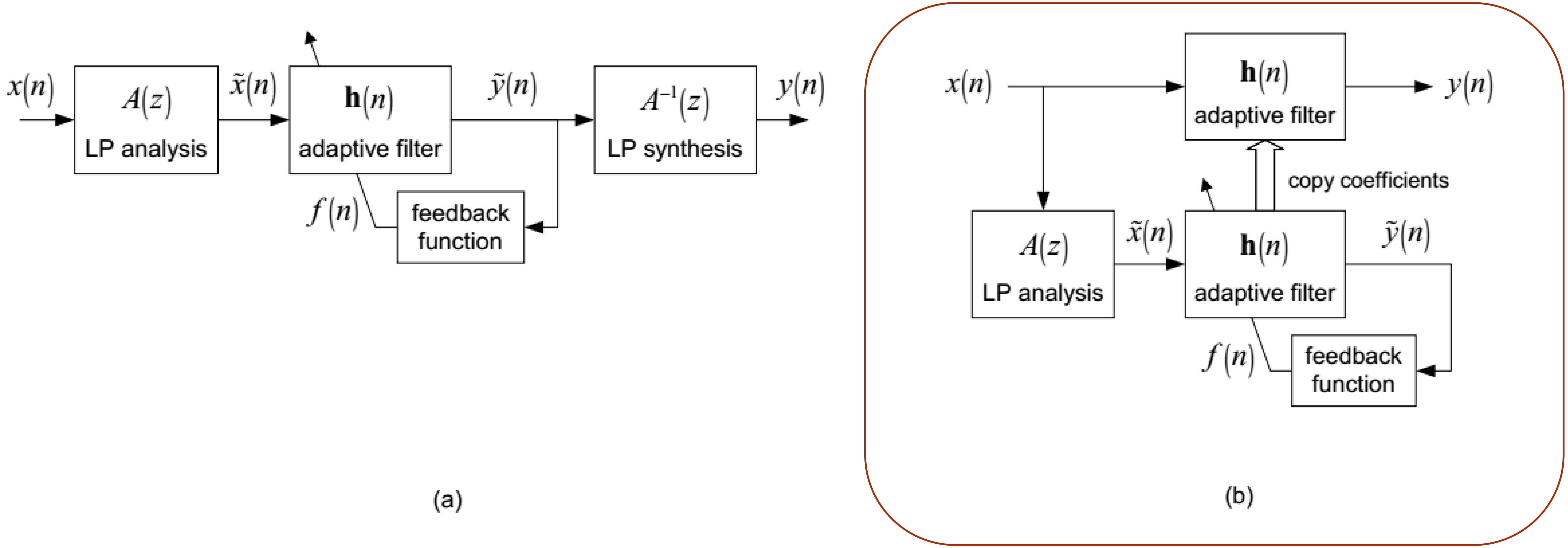
Kurtosis of LP residual of reverberant speech, kurt(y_r)

STAGE I: INVERSE FILTER

INVERSE FILTER

- Due to the smearing of the LP residual of reverberant speech, the kurtosis ot the reverberant speech is lower.
- We attempt to design an inverse filter, which seeks to maximize the kurtosis of the LP residual of the reverberant speech.
- The inverse filter cannot completely nullify the effect of RIR on the signal because it is non-minimum phase (may be possible in multi-channel implementation [2] where there are no common zeros). In single channel implementation the inverse filter is only assigned the task of reducing the effect of early reflections.
- Two arrangements of such inverse filter is shown in the following figure.

INVERSE FILTER ARRANGEMENTS



(a)

(b)

To avoid LP synthesis and resulting artifacts, we instead use a copy of the inverse filter to process the reverberant speech.

INVERSE FILTER UPDATE EQNS

- Let the filter coefficients be ***g(n)***. The inverse filtered speech is

$$z(n) = \boldsymbol{g}(\boldsymbol{n})^{\boldsymbol{T}}\boldsymbol{y}(\boldsymbol{n})$$

- The inverse filtered LP residual is

$$\tilde{z}(n) = \boldsymbol{g}(\boldsymbol{n})^{\boldsymbol{T}}\boldsymbol{y}_{\boldsymbol{r}}(\boldsymbol{n})$$

- We want to maximize the excess kurtosis given by

$$J(n) = \frac{E\{\tilde{z}^4(n)\}}{E^2\{\tilde{z}^2(n)\}} - 3$$

- The gradient is given by

$$\nabla_{\boldsymbol{g}}J(n) = \frac{4(E\{\tilde{z}^2(n)\}E\{\tilde{z}^3(n)\boldsymbol{y}_{\boldsymbol{r}}(n)\} - E\{\tilde{z}^4(n)\}E\{\tilde{z}(n)\boldsymbol{y}_{\boldsymbol{r}}(n)\})}{E^3\{\tilde{z}^2(n)\}} \approx f(n)\boldsymbol{y}_{\boldsymbol{r}}(n)$$

INVERSE FILTER UPDATE EQNS

- The update equation is

$$\boldsymbol{g}(n+1) = \boldsymbol{g}(n) + \mu f(n)\boldsymbol{y_r}(n)$$

- An implementation in frequency domain is crucial for faster convergence. In frequency domain, as suggested by the authors

$$\boldsymbol{G}(n) = \boldsymbol{G}(n) + \mu \boldsymbol{F}(n)\boldsymbol{Y}_{\boldsymbol{r}}^{*}(n)$$

- We implement the inverse filter in the frequency domain in a block adaptive configuration and make multiple iterations on 20s speech sample.

NOTE ON IMPLEMENTATION

- In block adaptive setting, the excess kurtosis can be approximated from the sample averages instead of ensemble averages:

$$\hat{J}(m) = \frac{\sum \tilde{z}^4(n)/L}{[\sum \tilde{z}^2(n)/L]^2} - 3$$

Where the sums are taken over the *m'th* block and L is the number of samples in the block.

- We can maximize this blockwise approximation of excess kurtosis

$$\nabla_{\boldsymbol{g}} \hat{J}(m) = \frac{4(\sum[\tilde{z}^2(n)]\sum[\tilde{z}^3(n)\boldsymbol{y}_{\boldsymbol{r}}(n)] - \sum[\tilde{z}^4(n)]\sum[\tilde{z}(n)\boldsymbol{y}_{\boldsymbol{r}}(n)])L}{\{\sum \tilde{z}^2(n)\}^3} \approx \sum f(n)\boldsymbol{y}_{\boldsymbol{r}}(n)$$

- We used a frequency domain implementation of this expression using overlap save method for filter update

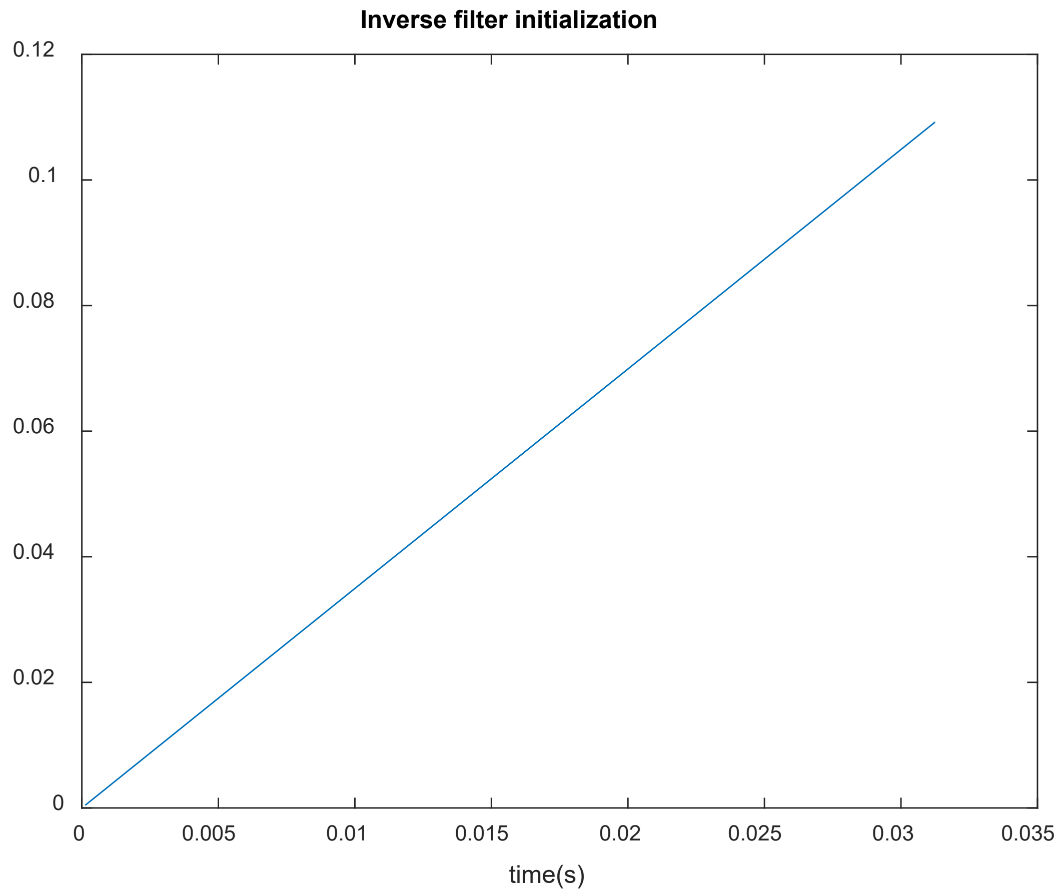
$$\hat{\boldsymbol{g}}(m+1) = \hat{\boldsymbol{g}}(m) + \mu \nabla_{\boldsymbol{g}} \hat{J}(m)$$

NOTE ON IMPLEMENTATION

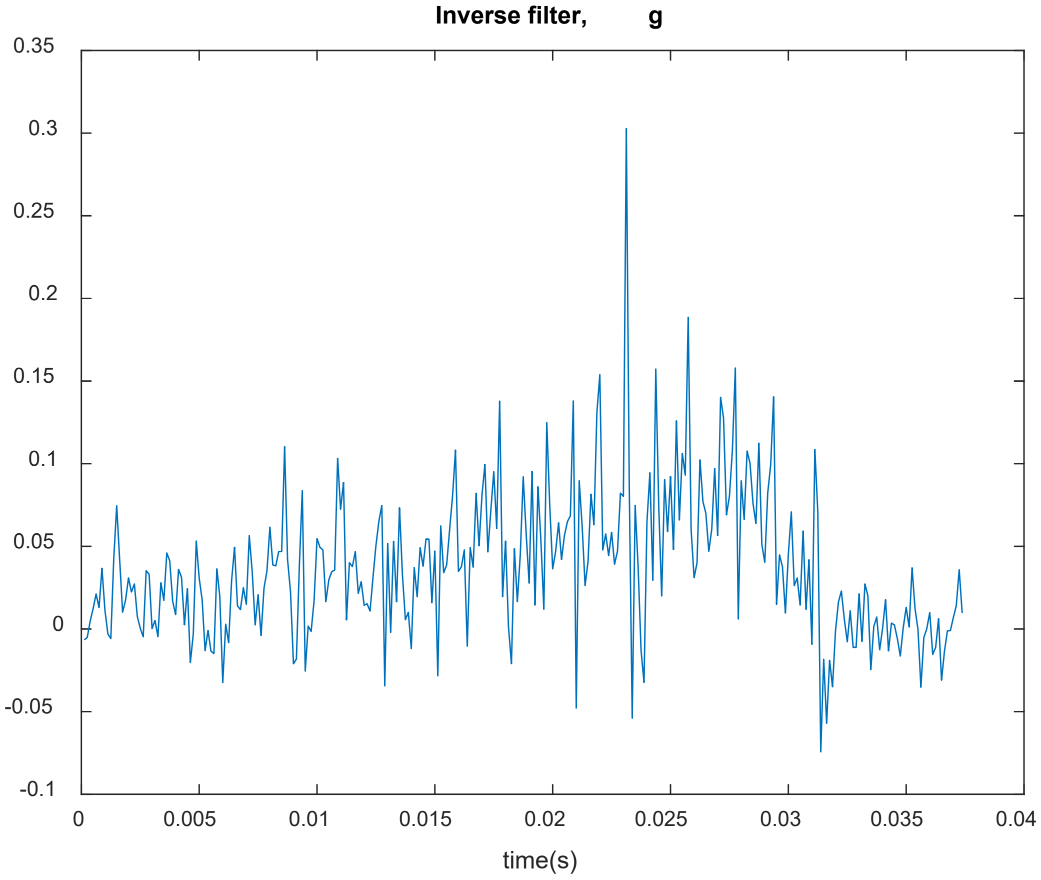
- It is important to understand that kurtosis is based on 4'th order statistics, it has local maximas and is not very smooth. To ensure convergence we should
  - Start with a good initial value. This cuts down time of convergence a lot.
  - Make the learning rate small enough to avoid divergence.
  - Make multiple iterations on the same data.

- Our objective is to remove the effect of early reflections. Increasing filter length will obviously maximize kurtosis further, but will degrade the effect of late reflections and add additional delay. So we limit filter length to a value that makes a good trade off between these two effects.

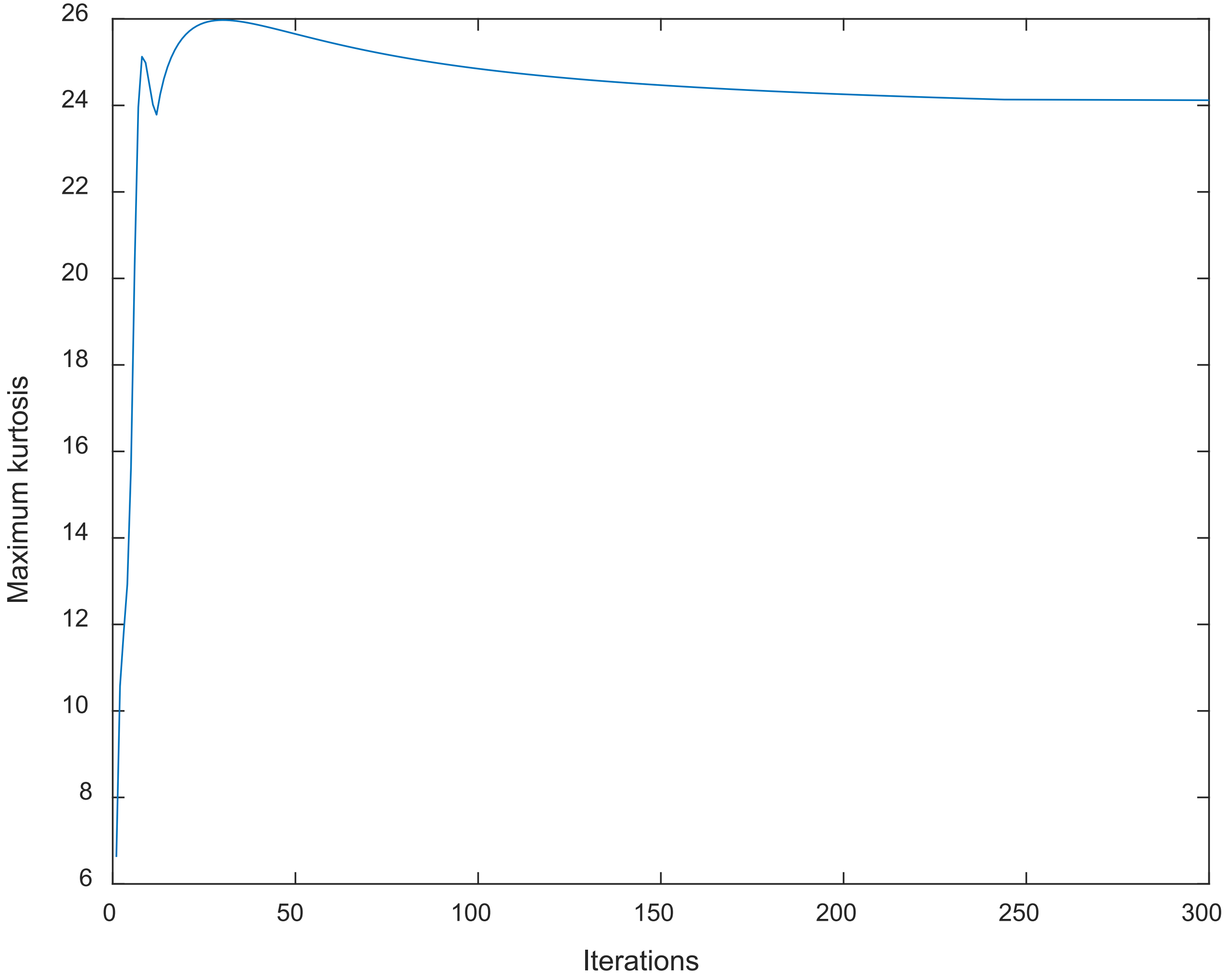
INVERSE FILTER INITIALIZATION



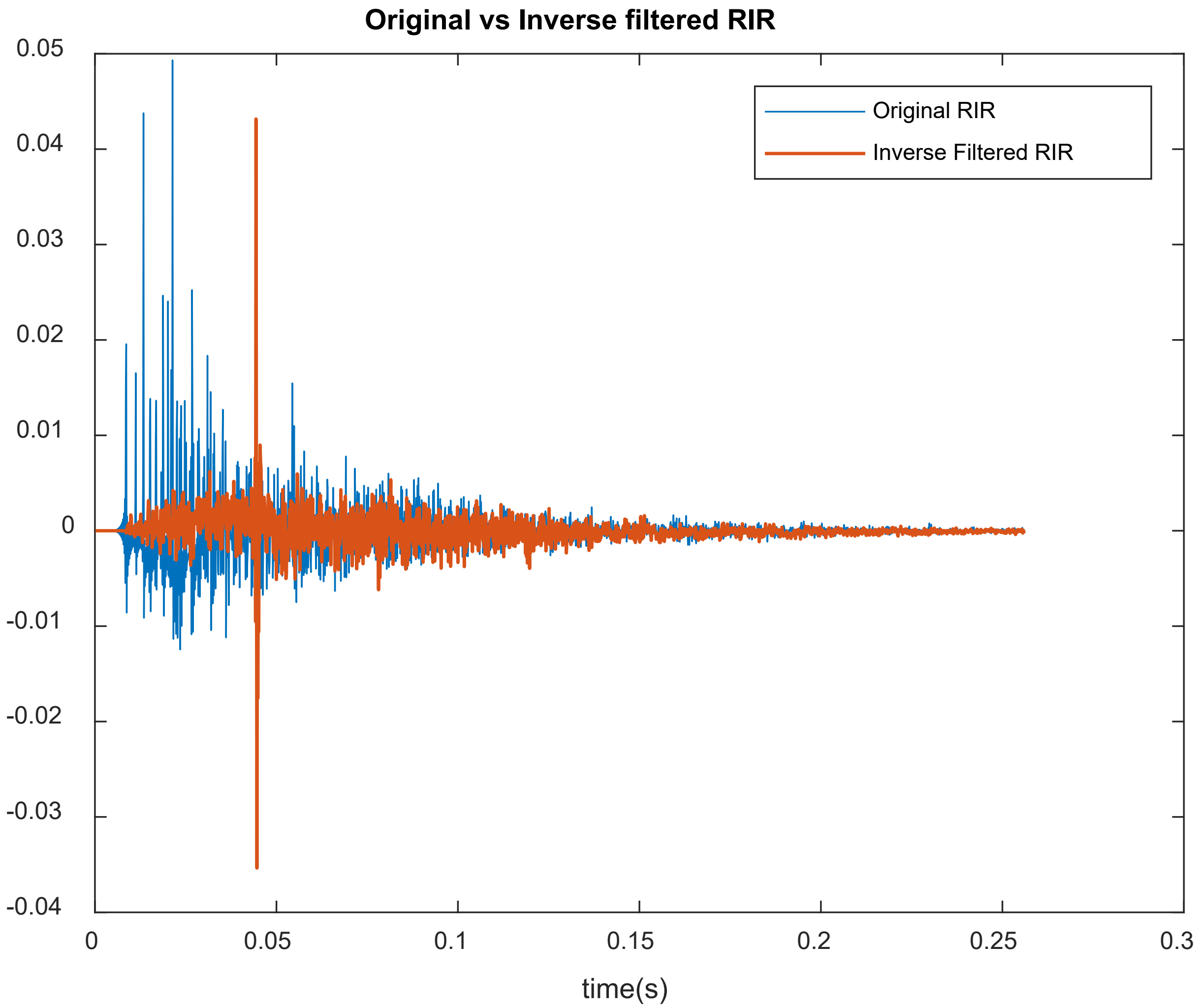
CONVERGED INVERSE FILTER

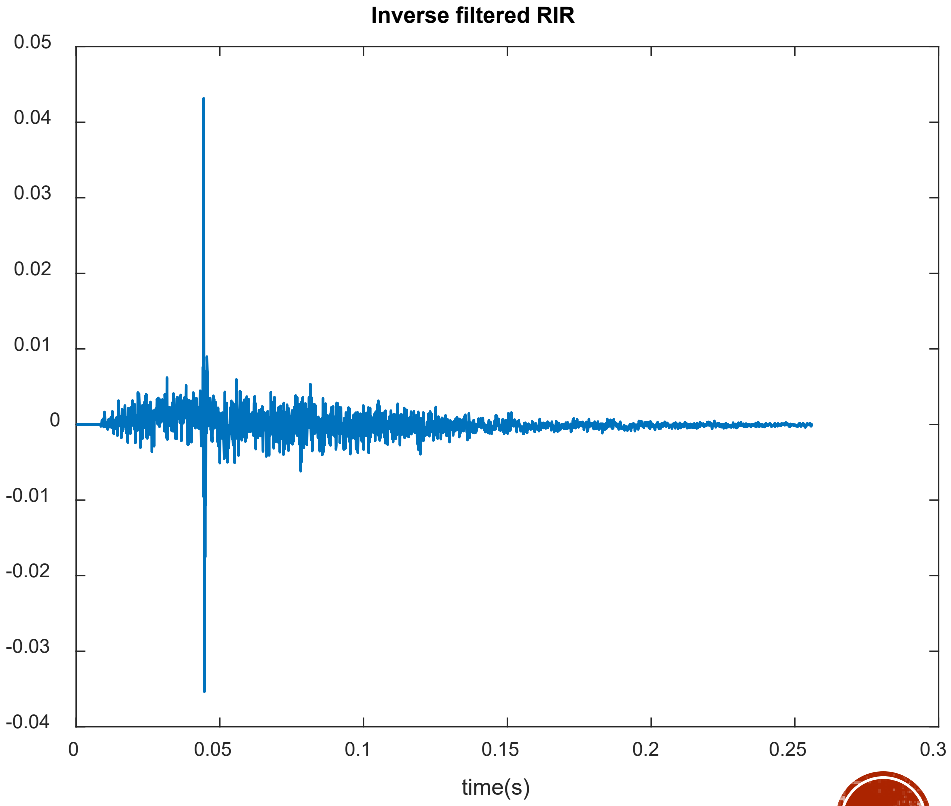


INCREASING KURTOSIS WITH ITERATIONS

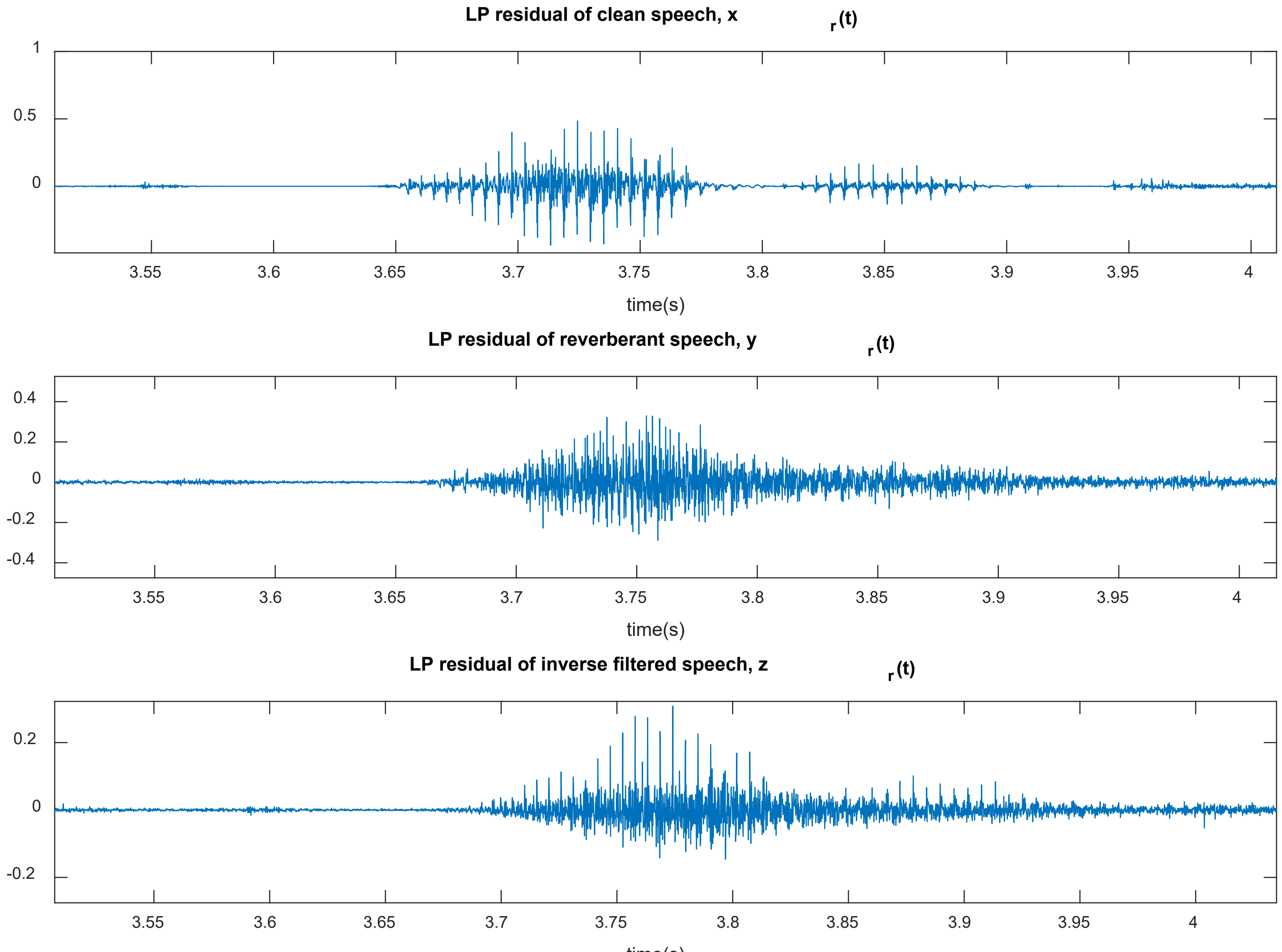


RESULT OF INVERSE FILTERING: RIR



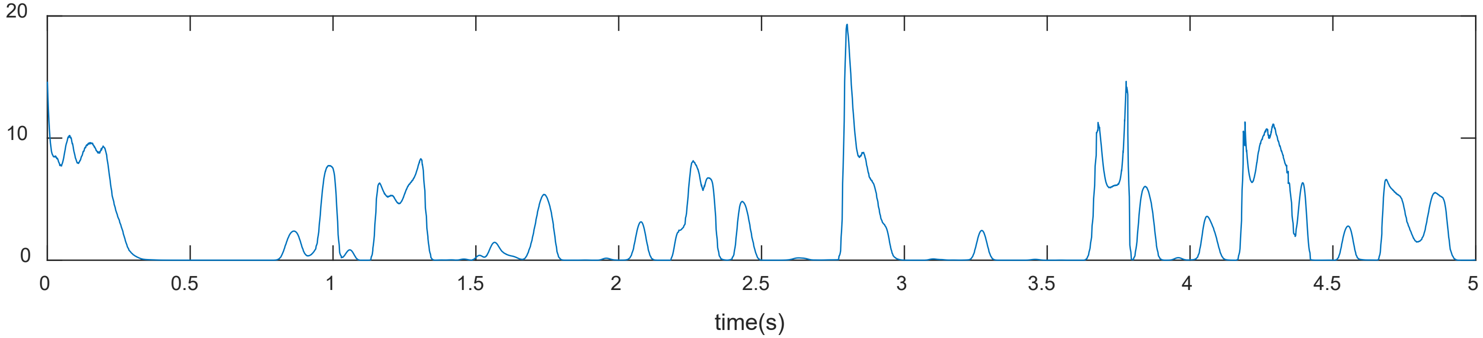


RESULT OF INVERSE FILTERING: RESIDUAL

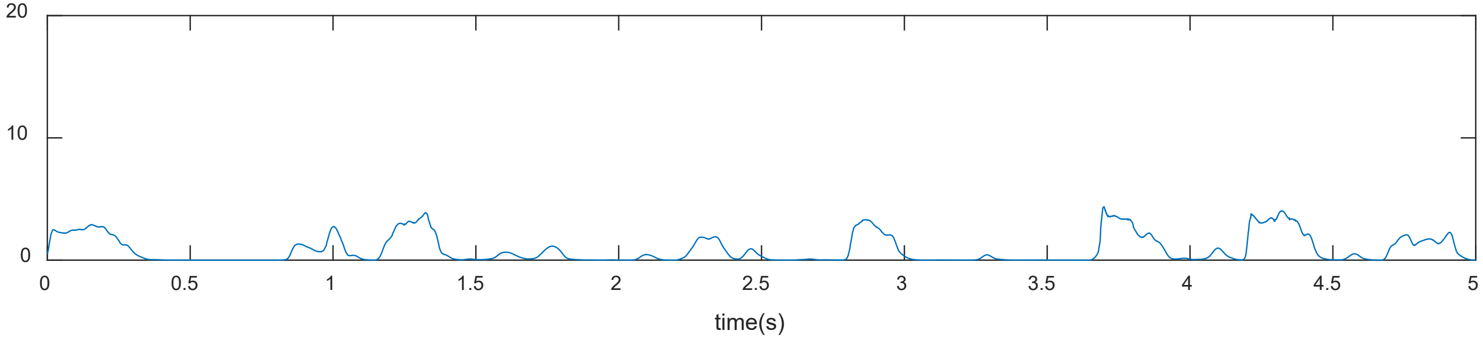


RESULT OF INVERSE FILTERING: KURTOSIS

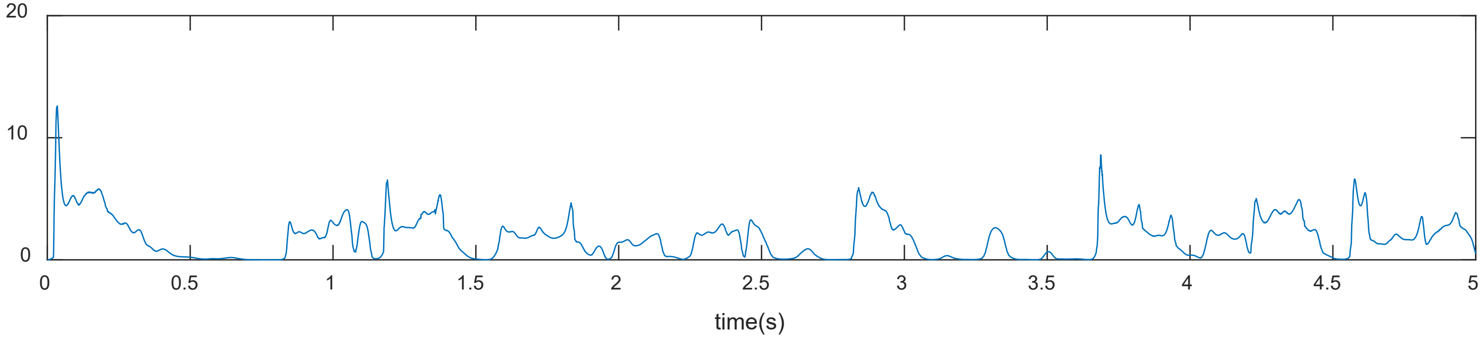
Kurtosis of LP residual of clean speech, kurt(x)



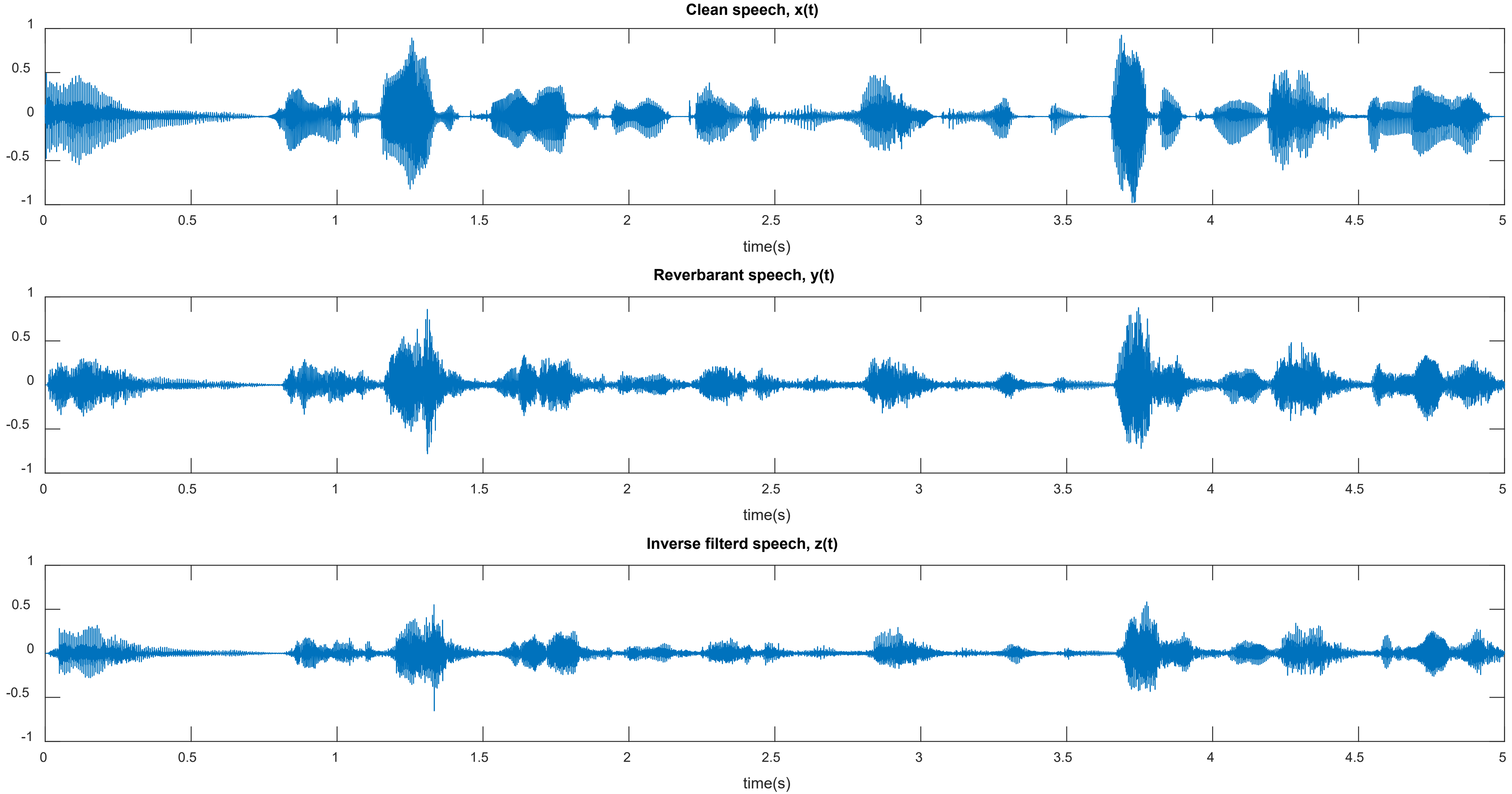
Kurtosis of LP residual of reverbarant speech, kurt(y)



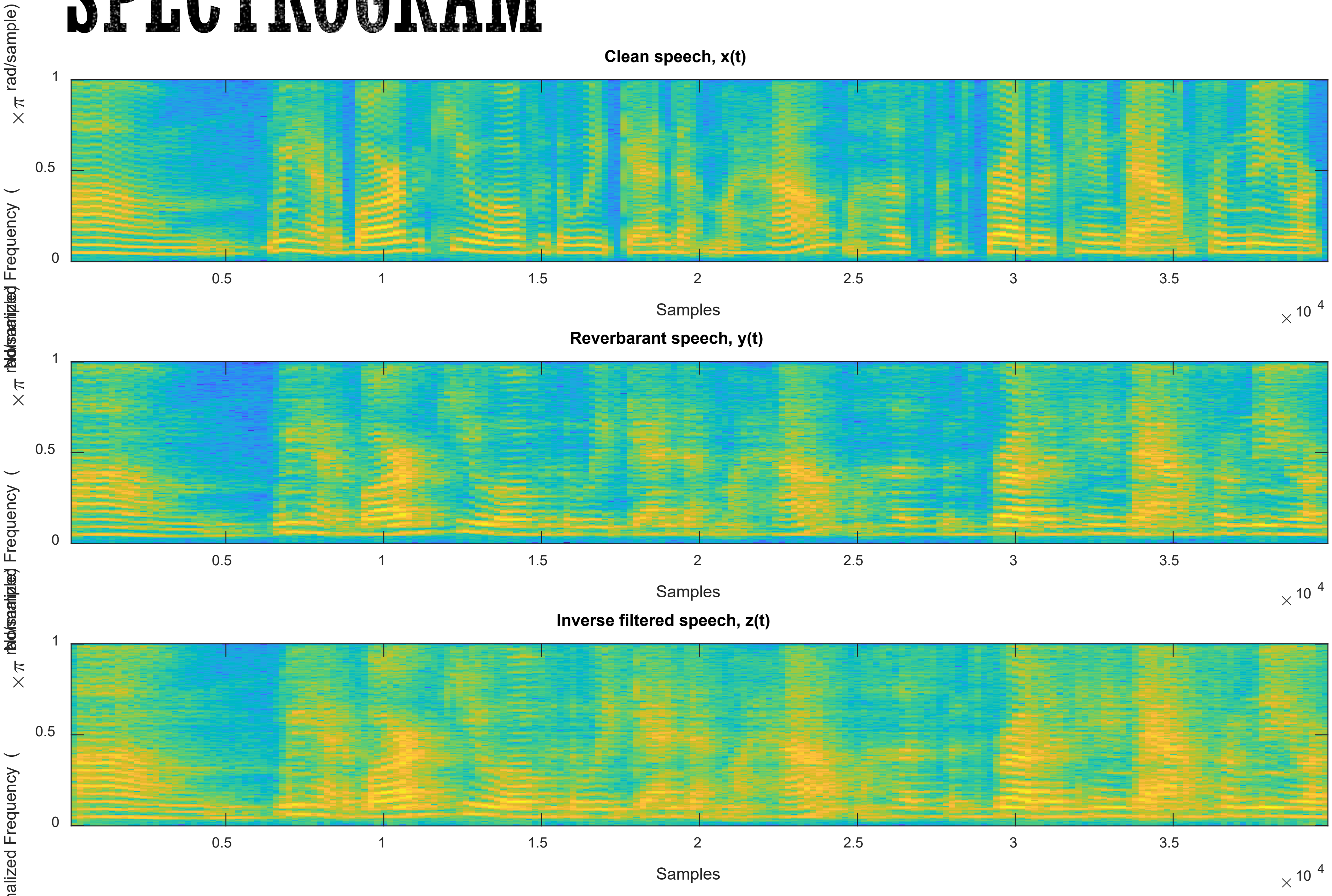
Kurtosis of LP residual of inverse filtered speech, kurt(z)



RESULT OF INVERSE FILTERING: WAVEFORMS



SPECTROGRAM



Clean speech, x(t)

Reverbarant speech, y(t)

Inverse filtered speech, z(t)

RESULT OF INVERSE FILTERING

- Inverse filtering reduces the effect of early reflections, but the late reflections are still dominant, as seen from the inverse filtered RIR.
- From the waveforms, we see that the signal has nearly regained its original wave shape i.e. it seems to reduces the smearing effect in time domain.
- From the spectrogram, the inverse filtered signal looks like a smudged version of the clean speech signal i.e. does not do anything to reduce the smearing in frequency spectrum.
- Inverse filtering also adds some delay. This is because delay is necessary for the filter to be causal.
- From auditory experience, the inverse filtered speech sounds almost like reverberant speech, although the resonant nature seems to go down a slightly. This is because the human ear cannot detect the effect of early reflections.

STAGE II: MODIFIED SPECTRAL SUBTRACTION

MODIFIED SPECTRAL SUBTRACTION

- We assume that the power spectrum of late-impulse components is a smoothed and shifted version of the power spectrum of the inverse-filtered speech z(n).

$$|S_l(k;i)|^2 = \gamma w(i-\rho) * |S_z(k;i)|^2$$

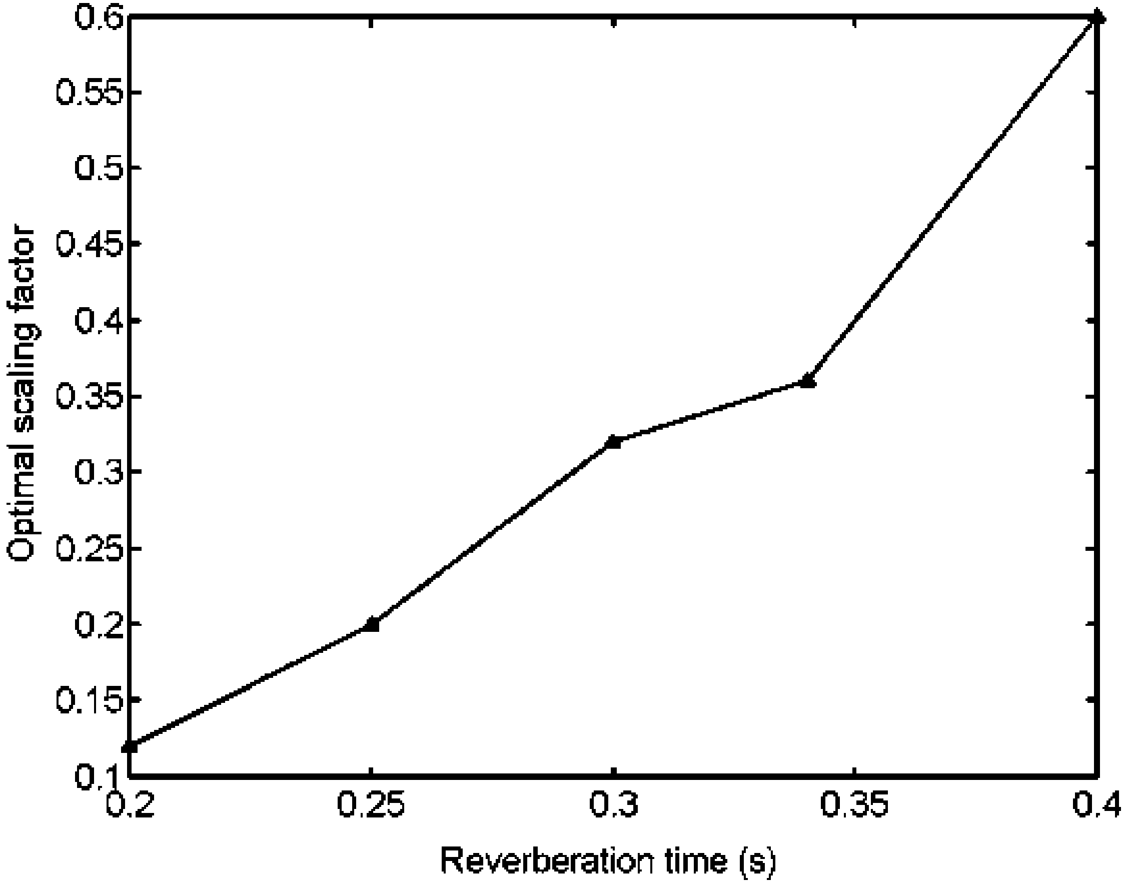
假设晚期rir脉冲分量的功率谱是逆滤波语音z(n)功率谱的平滑移位版本

- γ is a scaling factor and $\omega(i)$ is a smoothing function given by

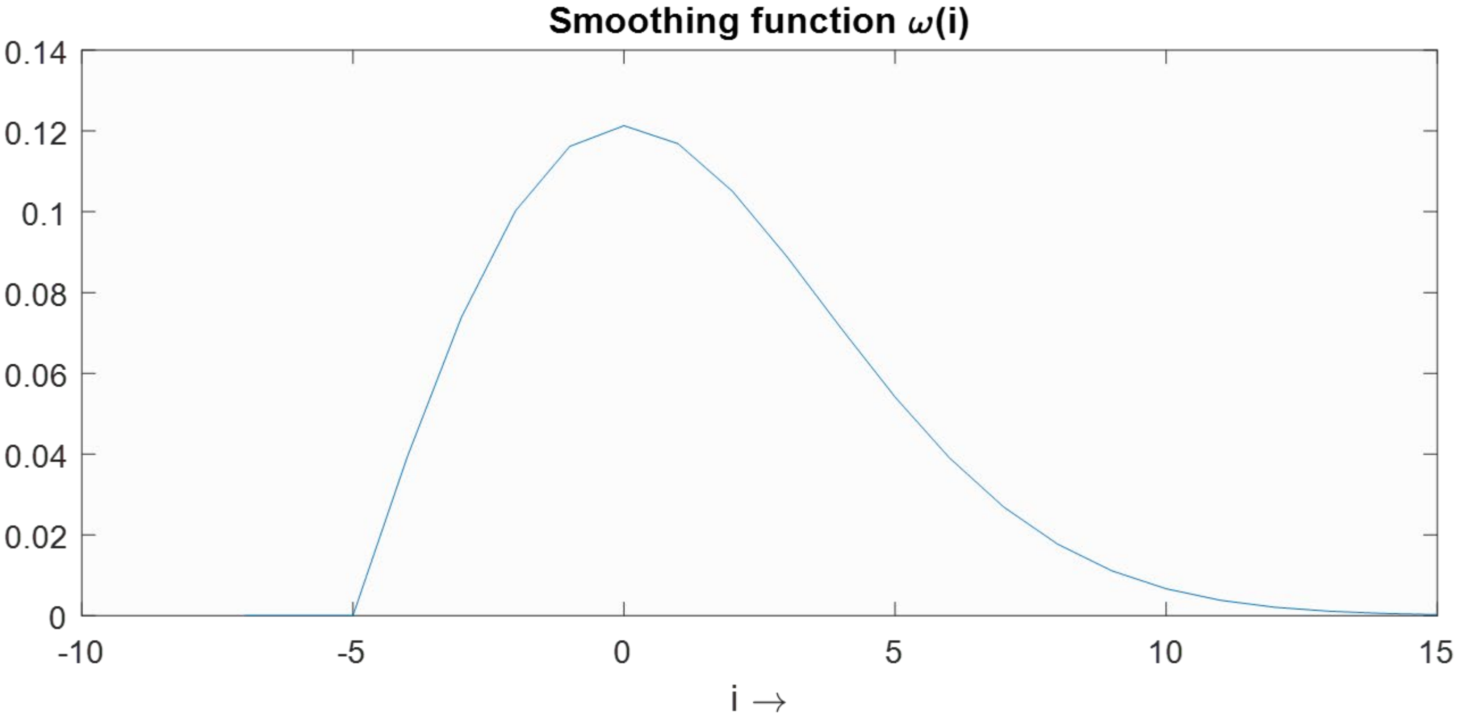
$$\begin{cases} w(i) = \frac{i+a}{a^2}\exp\left(\frac{-(i+a)^2}{2a^2}\right), & \text{if } i > -a \\ w(i) = 0, & \text{otherwise.} \end{cases}$$

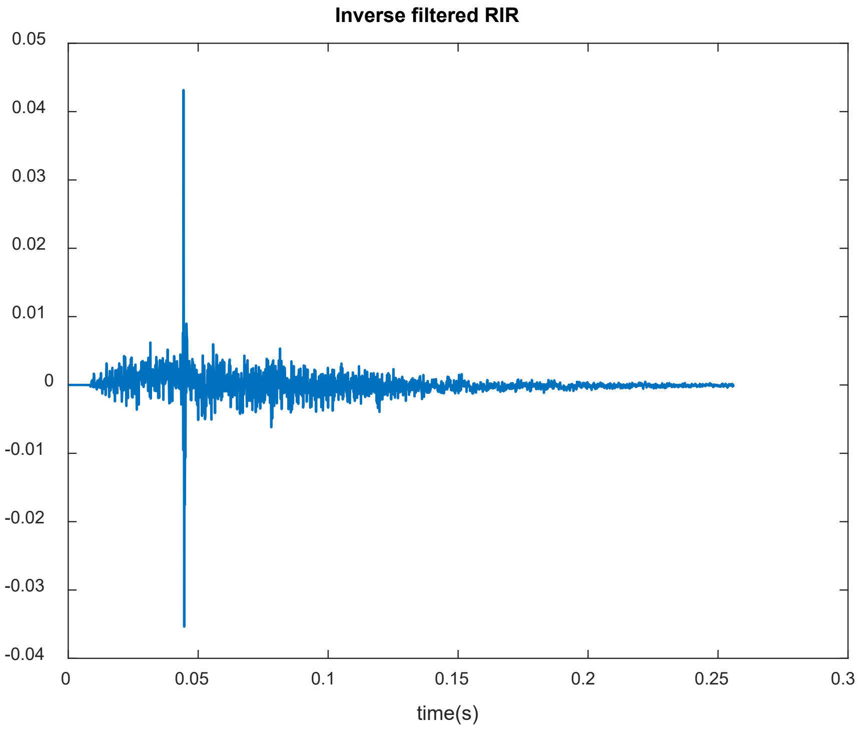
- The shift delay ρ indicates the relative delay of the late-impulse components. Here $a < \rho$.
- For a time shift of 8ms $\rho = 50/8 \approx 7$.

HOW TO CHOOSE THE SCALING FACTOR



THE SMOOTHING FUNCTION





MODIFIED SPECTRAL SUBTRACTION

- Assuming the early and late components are mutually uncorrelated, the power spectrum of the early-impulse components can be estimated by subtracting the power spectrum of the late-impulse components from that of the inverse-filtered speech.
- So, we take the STFT of the inverse filtered speech and estimate the final processed speech signal as

$$|S_{\tilde{x}}(k;i)|^2 = |S_z(k;i)|^2 \max\left[\frac{|S_z(k;i)|^2 - \gamma w(i-\rho) * |S_z(k;i)|^2}{|S_z(k;i)|^2}, \varepsilon\right]$$

- Here ϵ is a noise threshold. For $\epsilon = 0.001$ we define a maximum attenuation of 30dB.

IMPLEMENTATION NOTES

- The modified spectral subtraction process takes place between STFT and ISTFT.
- After taking STFT we perform the squared magnitudes, i.e power in each frequency bins, and perform the convolution operation as shown before to estimate the power due to late reflection in that particular bin.
- Then we finish the spectral subtraction by determining the attenuation factors and multiplying the powers accordingly.
- We get the magnitudes from the squared roots of the powers and the phase is kept unchanged.
- Then we take ISTFT to get the final signal.

APPLYING CRITERION FOR SILENCE

- A frame is silenced when
  - The energy of the inverse filtered speech $E_z(i)$ is less than a threshold ϑ_1. i.e.

$$E_z(i) < \vartheta_1$$

And,

  - The energy ratio of the inverse filtered speech $E_z(i)$ to the spectral subtracted speech $E_{\tilde{x}}(i)$ is greater than a threshold ϑ_2. i.e.

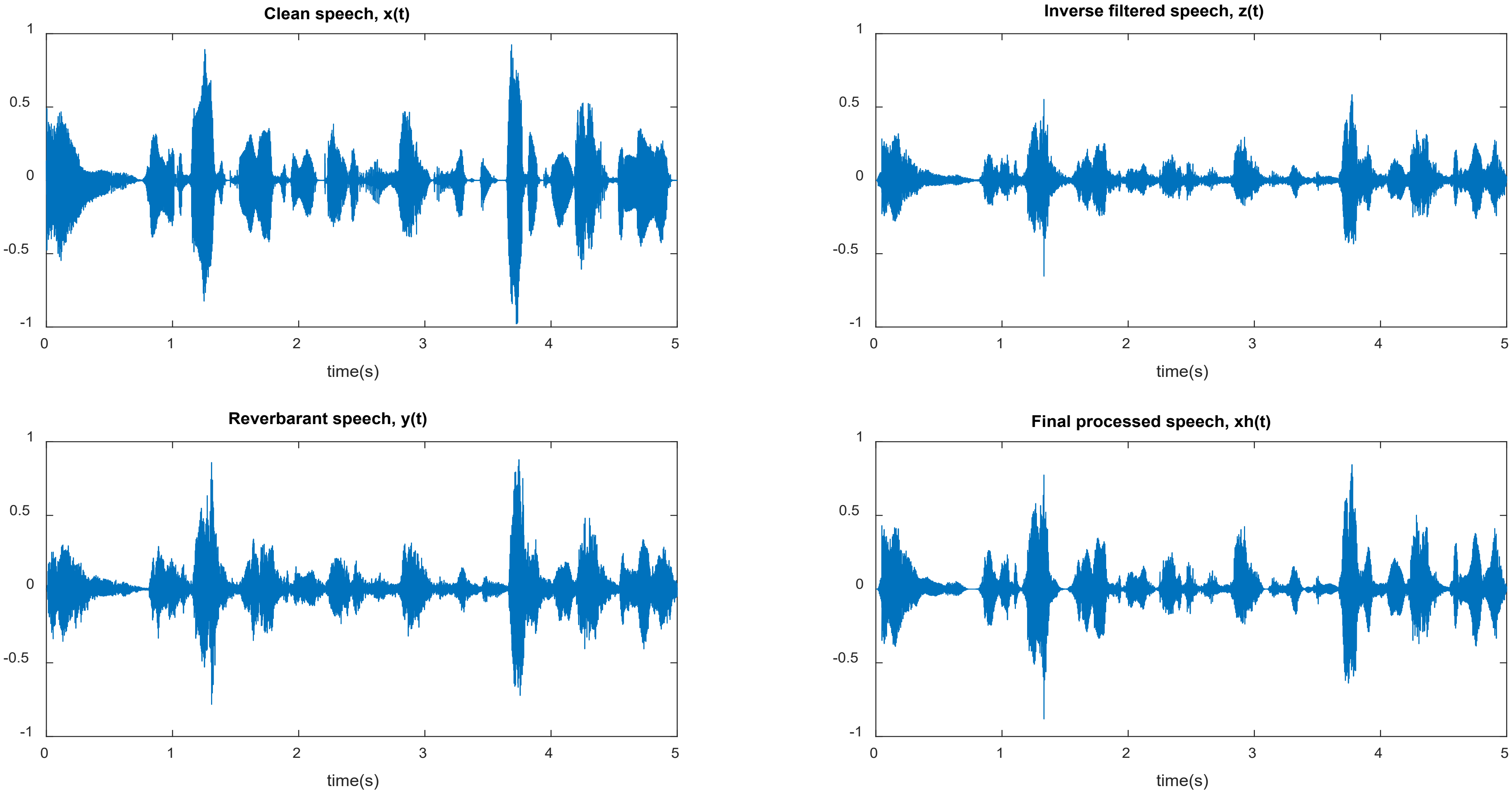
$$\frac{E_z(i)}{E_{\tilde{x}}(i)} > \vartheta_2$$

- However we used this criterion sparingly because too hard a criterion degrades speech quality (from auditory experience).

RESULTS

And relevant discussions

RESULTS: WAVEFORMS



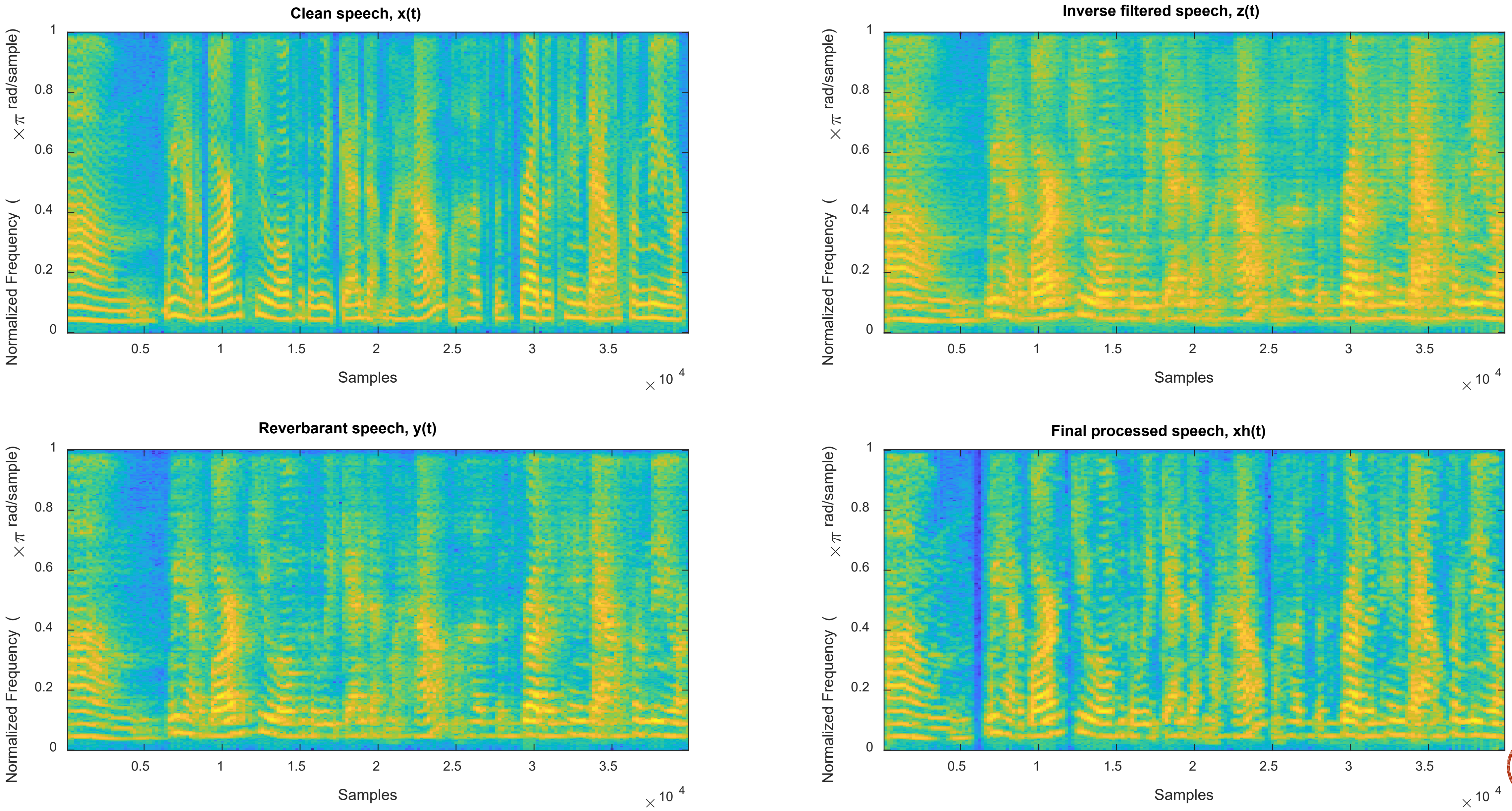
Clean speech, x(t)

Inverse filtered speech, z(t)

Reverbarant speech, y(t)

Final processed speech, xh(t)

RESULTS: SPECTROGRAM



Clean speech, x(t)

Inverse filtered speech, z(t)

Reverbarant speech, y(t)

Final processed speech, xh(t)

RESULTS: PESQ AND SPEECH QUALITY

Sample1: Single Female Monologue

Signal	PESQ[4]	
	Raw-MOS	MOS-LQO
Clean speech, x (maximum)	4.500	4.549
Reverberant speech, y	2.384	1.960
Inverse filtered speech, z	2.425	2.046
Final (spectral subtracted), $\hat{x}$	2.505	2.142

RESULTS: PESQ AND SPEECH QUALITY

Sample2: Two Male Conversation

Signal	PESQ[4]	
	Raw-MOS	MOS-LQO
Clean speech, x (maximum)	4.500	4.549
Reverberant speech, y	2.437	2.060
Inverse filtered speech, z	2.472	2.102
Final (spectral subtracted), $\hat{x}$	2.636	2.307

RESULTS

- Spectral subtraction removes the smeared-ness power spectrum, which is evident in the spectrogram.
- Silencing the low power parts generates natural gaps in speech (blue vertical lines in the spectrogram).
- The speech quality increases in each stage as evident from PESQ and auditory experience.
- SRR improves in each stage.
- Although it may seem from the auditory clues that the first stage (inverse filtering) does not improve SRR much, it is essential for the second stage to be effective.

DISCUSSIONS

- The convergence of the algorithm was found to be very slow. It took us 40s to process 20s speech. So it may not be applicable to real time speech processing, where the RIR varies frequently.
- A multi-channel implementation may facilitate real time processing [2].
- When designing inverse filter there is actually a trade off between reducing early and late reverberant components, and we end up increasing the late components slightly. But the spectral subtraction stage makes up for this degradation.
- The algorithm is subject to a few tuning parameters, the automated selection of these parameters in real world situation may result in trade offs between robustness and performance.
- The inverse filtering process is the most time consuming part of the algorithm.
- The algorithm has no inherent ability to remove additive noise from reverberant speech, as was found from experimentation. Further processing may be required for removal of additive noise.

FURTHER RESEARCH

- Further investigations can be carried out to increase the speed of convergence of the inverse filtering process and make it suitable for online processing.
- The spectral subtraction method is an effective way of removing late reverberation components. It can be added to other one stage algorithms to enhance their performances.
- Instead of using fixed parameters, we may look for some methods for estimating the parameters automatically from the data.

REFERENCES

1. Wu, Mingyang, and DeLiang Wang. "A two-stage algorithm for one-microphone reverberant speech enhancement." IEEE Transactions on Audio, Speech, and Language Processing 14.3 (2006): 774-784.
2. Gillespie, Bradford W., Henrique S. Malvar, and Dinei AF Florêncio. "Speech dereverberation via maximum-kurtosis subband adaptive filtering." *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*. Vol. 6. IEEE, 2001.
3. Yegnanarayana, Bayya, and P. Satyanarayana Murthy. "Enhancement of reverberant speech using LP residual signal." *IEEE Transactions on Speech and Audio Processing* 8.3 (2000): 267-281.
4. http://www.opticom.de/technology/pesq.php

Thank you