Interpolate to Enhance for NonStationary Signal Processing

Mahdi Triki
Philips Research, Eindhoven, The Netherlands
mahdi.triki@philips.com

Abstract

Enhancing non-stationary signals is crucial for many applications, such as speech recognition, audio communication, and bio-signals analysis. The present paper investigates a novel processing structure (alternative to the overlap-add scheme), based on an interpolated zero-phase FIR filtering. The proposed structure accounts for slow signal non-stationarity, and also natively supports time and frequency smoothing. Applied for speech denoising and compared to the usual overlap-add framework, the proposed structure has shown promising results in terms of quality, perception, and recognition performances.

Index Terms: speech enhancement, segmentation, linear interpolation, smoothing, Wiener filter.

1. Introduction

Processing time-varying (non-stationary) signals is a challenging task, both from theoretical and practical prespectives. Indeed, most of the usual concepts/models introduced for stationary signals cannot be straightforwardly extended to the non-stationary context. Let's consider, as an example, the basic concept of 'frequency'. For sinusoidal signals, this quantity is well defined. It could be unambiguously extended for stationary signals (as any stationary signal can be represented as a weighted sum of sins and cosines with particular frequencies, amplitudes and phases) [1]. However, for non-stationary signals, the sinusoidal decomposition is no-longer unique, which arises some identification issues and makes the physical interpretation ambiguous [1, 2, 3]. From a practical prespective, even if these ambiguities were alleviated, estimating the model parameters is a challenging task, as the helpful ergodicity assumption is no-longer valid.

Within the group of non-stationary signals, the subclass of the slowly-stationary signals is of particular interest, as it includes some useful signals such as speech, music, and large range of bio-signals. The statistical variations of such signals is so slow that they can be assumed locally constant/stationary; although globally, they are time-varying.
To process such signals, a three step scheme is typically used:

1. Segment the signal into (locally) stationary blocks
2. Process (individually) each block, by usual stationary tools
3. Synthesize the processed blocks

In all the stages, a key ingredient is the selection of the segmentation window. A good choice should ensure:

- Effective non-stationary to locally-stationary decomposition (segmentation stage)
- Good linear to circular convolution approximation (processing stage)
- Smooth (alleviate discontinuity) signal resynthesis (synthesis stage)

With this respect it has been observed that using a smooth window (e.g. Hanning window) is beneficial, and that both time and frequency smoothing are advantageous. On the other hand, full understanding of the effect of time-windowing is still an open issue and is triggering the interest of the signal processing research communities [4, 5], not only for theoretical challenges but also for an effective design of the processing schemes.

In the present paper, the frame-by-frame analysis and synthesis structure is analyzed, and an alternative processing structures are investigated. The remainder of this paper is organized as follows. The problem statement is introduced in Section 2. Section 3 analyses time-segmentation for both frequency-flat and (smooth) frequency-selective processing. Application to Wiener filter design for speech enhancement is considered in Section 4. Finally, a discussion and concluding remarks are provided in Section 5.

2. Problem Statement

We consider a standard frame-by-frame processing scheme (Figure 1). The received signal $y(n)$ (contaminated by noise, reverberation, etc) is first segmented into overlapping frames:

$$\begin{aligned} y^{(k)}(n) &= w^{(k)}(n)\, y(n) \\ &= w(n-kD)\, y(n) \end{aligned} \tag{1}$$

D characterizes the window shift, and $w(n)$ is a function with a finite support (non-zeros elements), i.e.,

$$w(n) = 0 \qquad n \notin [0..B-1] \tag{2}$$

where B denotes the window length.

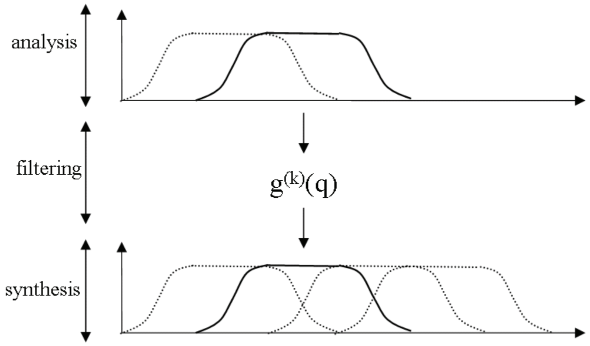

Figure 1: Frame-by-Frame processing scheme.

The segmented blocks are individually processed by the (piecewise constant) filters $g^{(k)}(q)$:

$$x^{(k)}(n) = g^{(k)} * y^{(k)}(n) \tag{3}$$

where $(.*.)$ represents the convolution operator.

978-1-4673-0046-9/12/$26.00 ©2012 IEEE

Finally, the enhanced signal is synthesized

$$x(n) = \sum_k x^{(k)}(n) \tag{4}$$

To ensure signal preservation (when no processing is performed), the segmentation parameters (shape and shift) satisfy

$$\sum_k w^{(k)}(n) = \sum_k w(n - kD) = 1 \tag{5}$$

Notations: Upper- and lower-case boldface letters denote matrices and vectors, respectively. Upper- and lower-case normal letter represent scalar constant and processes, respectively. Either as a subscript, superscript or argument n, t and k refer respectively to the time, time-lag, and frame indexes. f and ν refer respectively to the frequency and frequency-band indexes.

3. Analysis and Interpretation of Segmented Processing

In frequency domain, the enhanced signal can be expressed as:

$$\begin{aligned} x(f) &= \sum_k g^{(k)}(f)\, y^{(k)}(f) \\ y^{(k)}(f) &= w^{(k)}(f)\, y(f) \\ &= \left[e^{-2j\pi kfD/B} w(f)\right] * y(f) \quad f{=}1{:}B \end{aligned} \tag{6}$$

Typically, the segmentation window length and shift are linearly related (i.e. $\exists K \in \mathbb{N},\ B = KD$). Only K frames get involved in the overlap-add synthesis (i.e. only K entries of the infinite sum are non-zero).

Assuming a frequency-flat processing scheme (i.e., $g^{(k)}(f)$ is frequency independent), the synthesized signal is given by

$$\begin{aligned} x(f) &= \left[\sum_k g^{(k)}\ w^{(k)}(f)\right] \ *\ y(f) \\ &= \underbrace{\left[\left(\sum_k e^{-2j\pi fk/K} g^{(k)}\right)\ w(f)\right]}_{g(f)} * y(f) \end{aligned} \tag{7}$$

The frame-by-frame enhancement could be then interpreted in terms of a (slowly-varying) amplitude modulation. The modulating signal $g(n)$ is generated by interpolating the (downsampled) process $\left\{g^{(k)}\right\}_k$ with the smooth spline function $w(n)$ (see appendix A for further details on the interpretation of the interpolation operator in terms of linear filtering).

More generally, the gain function is often designed to be slowly-varying with frequency. Several studies reported that smoothing the gain function over frequency reduces musical noise and enhances auditory results. The smoothing was recommended for both communication [6] and speech recognition [7] applications. With this respect, filter-bank provides a flexible and effective structure to implement the gain smoothing [8]. The input signal $y(n)$ is passed through a bank of M analysis filters $\left\{h^{(\nu)}(q)\right\}_{\nu=1:M}$, each of which preserves a frequency band of uniform bandwidth . An enhancement gain filter $g^{(\nu)}(q)$ is next applied in each frequency band ν. Given that enhancement is slowly varying over frequency, the gain filter (at each subband) can be assumed flat and expressed in terms of a slowly-varying amplitude modulation, i.e.,

$$g^{(\nu)}(n) = g_{\downarrow}^{(\nu)} * w(n) \tag{8}$$

Finally, the subbands are combined by a set of synthesis filters $\left\{f^{(\nu)}(q)\right\}_{\nu=1:M}$ to form the reconstructed signal:

$$\begin{aligned} x(n) &= \sum_\mu x^{(\nu)}(n) \\ x^{(\nu)}(n) &= f^{(\nu)} * \left[g^{(\nu)}(n) \cdot \left(h^{(\nu)} * y(n)\right)\right] && (9) \\ &\approx g^{(\nu)}(n) \cdot \underbrace{\left(f^{(\nu)} * h^{(\nu)} * y(n)\right)}_{y^{(\nu)}(n)} && (10) \end{aligned}$$

The approximation (10) assumes that the gain filter $g^{(\nu)}(n)$ bandwidth is narrow compared to the subband bandwidth. Thus, one may interpret the (smooth) segmented processing in terms of (frequency-selective) amplitude modulation. The amplitude modulating filter is constructed by interpolating the filter coefficients (at different subbands) with the spline window $w(n)$. The frequency-selective AM modulation was also introduced for audio signal modeling. It has been shown to be linked to the way sounds are produced, and effective to express a variety of musical instruments [9, 10], and speech signals [11]. The remainder of this paper investigates an alternative approach, where the time-domain coefficients of the gain filters are interpolated (rather than frequency-domain coefficients). Explicitly, the enhancement gain filters are

1. estimated (assuming stationarity) at each frame
$$g^{(k)}(q) = \sum_t g^{(k)}(t) q^{-t} \tag{11}$$
2. interpolated over frames
$$g_n(t) = \sum_k g^{(k)}(t) w(n - kD) \tag{12}$$
3. applied in time-domain:
$$x(n) = \sum_t g_n(t) y(n - t) \tag{13}$$

Please note that the proposed structure is independent from the design of the gain filters $g^{(k)}(q)$. Moreover, the order of the gain filters is often selected small[1] (compared to the window size B). The additional computational complexity (due to interpolation and convolution) does not alter the overall computational performance.

4. Wiener Filter Design for Speech Enhancement

To investigate and compare the segmented and interpolated enhancement schemes, we consider a signal in noise problem

$$y(n) \ =\ s(n)\ +\ v(n) \tag{14}$$

where $s(n)$ is the clean signal, and $v(n)$ is an additive Gaussian noise (zero-mean and with known spectral density).

If the clean signal is Gaussian stationary, it can be optimally recovered (i.e. achieving the minimum Mean Squared Error (MSE)) using a Wiener filter

$$g(f) = \frac{S_{yy}(f) - S_{vv}(f)}{S_{yy}(f)} \tag{15}$$

[1] Limiting the order of the gain filter leads to a smoothing over the gain frequency components, which has been shown advantageous both audio quality and recognition accuracy

For speech inputs (locally stationary), Wiener filtering can be applied on a frame-by-frame structure: on each frame, a local Wiener filter is designed (using the local statistics) and applied (in frequency domain). Alternatively, an amplitude modulating filter could be generated (by interpolating the Wiener filter taps over frames), then applied (in time-domain). We constrain the amplitude modulating filter to be FIR and zero-phase:

$$\begin{aligned} g_n(-t) &= g_n(t) & -L \leq t \leq L \\ g_n(t) &= 0 & \text{otherwise} \end{aligned}$$

L is the order of the enhancement FIR filter. The estimation of the zero-phase FIR local Wiener filters is depicted in appendix B.

To assess the performance of the schemes described above, we consider the output SNR:

$$\text{SNR}_{\text{out}} = \frac{\sum_n s^2(n)}{\sum_n (x-s)^2(n)} \tag{16}$$

We consider a speech signal sampled at 8 kHz contaminated by an additive white noise (SNR $= 0dB$). The noise statistics are assumed to be known. The received signal is segmented using a Hanning window of length B and 50% overlap (K = 2). If our input is stationary and $B \rightarrow \infty$, the frequency-domain implementation of the Wiener filter maximizes the SNR‘$_{\text{out}}$. However for speech signals, B set a tradeoff between modeling the speech non-stationarity and estimating the signal statistics.

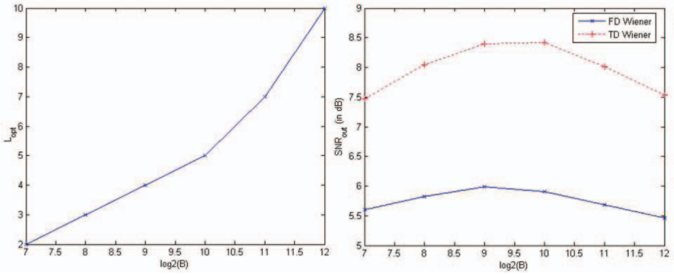

Figure 2: Optimal FIR order L (left) and its corresponding SNR$_{\text{out}}$ function of the segmentation window (B).

Figure 2 shows that the time-domain FIR Wiener (TD Wiener) consistently outperforms the frequency-domain implementation (FD Wiener). The FIR implementation achieves better tradeoff by discarding the noisy covariance estimates ($r_y(t),\ t > 2L$). The filter order set a tradeoff between the modeling error (due to FIR constraint) and the estimation error (due to second order statistics estimation), which justify the observation that the optimal order (leading to the best SNR improvement) increases with the window length.

Next, we investigate the choice of the interpolation window. We consider Bartlett and Hanning interpolating windows (Figure 3). Curves show that Bartlett smoothing is advantageous only at very low sampling-rate (K=2). For $K \geq 4$, Hanning windowing produces better results. These observations are consistent with studies reporting the outperformance of Hanning segmentation, and could be explained by the fact that a smooth window (with energy concentrated around the principal lobe) reduces the interpolation error. The higher the sampling rate (i.e., K), the wider the (allowed) bandwidth of the gain filter variations, and the higher the side-effect of the secondary lobes of the interpolating window.

The output SNR has straightforward interpretation; and it can provide indications of the perceived audio quality in some cases [12]. Unfortunately, the output SNR shows a limited correlation with perceived speech quality. Therefore, we also consider The ITU P.862 PESQ (Perceptual Evaluation of Speech Quality [13, 14]) for speech quality assessment. Figure 4 plots the SNR and the PESQ improvement (between the frequency and time domain implementations) function of the shift factor $K = B/D$. The FIR Wiener filters were interpolated using the 'usual' linear interpolation (Bartlett window).

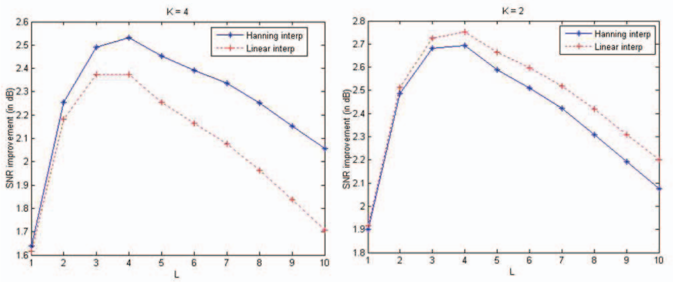

Figure 3: SNR improvement function of filter order for $K = 4$ (left) and $K = 2$ (rigth).

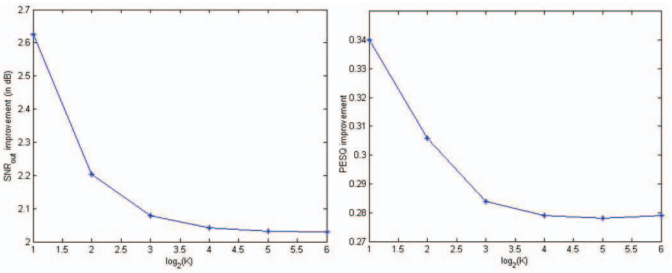

Figure 4: SNR and the PESQ improvement function of the shift factor $K = B/D$.

One can observe that the constrained FIR implementation consistently improves both the audio quality (SNR$_{\text{out}}$), and perception (PESQ). It becomes key in the high shift region (K=2). The proposed scheme was also applied as pre-processing for the DNS speech recognizer. The input signal was sampled at $11.025kHz$, synthetically distorted by an additive white Gaussian noise (SNR$_{\text{in}}$ $= 20dB$), and segmented using a Hamming window ($B = 1024$ and $K = 2$). Table 1 shows that FIR Wiener also enhances the recognition accuracy. However, particular attention should be paid to the selection of the interpolation window.

Pre-Processing	$K = 2$	$K = 4$	$K = 8$
Clean	86.52	86.95	86.63
FD Wiener	69.41	69.09	66.42
TD Wiener (Hanning)	66.20	65.78	65.56
TD Wiener (Bartlett)	77.97	77.01	78.93

Table 1: Speech in white noise: speech recognition rate (in %).

5. Concluding Remarks

In the present paper, we have interpreted the frame-by-frame processing scheme in terms of (smooth) time-varying frequency-selective modulation. We have proposed an alternative (smoothed) FIR Wiener filtering for speech enhancement application. Simulations show that the proposed scheme may outperform the usual overlap-add structure in terms of speech quality, perception and recognition.

6. References

[1] B. Boashash, ”Estimating and Interpreting the Instantaneous Frequency of a Signal–Part 1: Fundamentals,” *In IEEE Proceedings,* pp.519-538, Apr. 1980.

[2] D. Wei and A. C. Bovik, ”On the Instantaneous Frequencies of Multicomponent AM-FM Signals,” *IEEE Signal Processing Letter,* 1998.

[3] P.J. Loughlin and B. Tacer, ”Comments on the Interpretation of Instantaneous Frequency,” *IEEE Signal Processing Letter,* pp.123-125, May 1998.

[4] Y. Pantazis, O. Rosec and Y. Stylianou, ”Iterative Estimation of Sinusoidal Signal Parameters,” *IEEE Signal Processing Letter,* pp.461-464, Apr. 2010.

[5] J.I. Marin-Hurtado and D. Anderson. “Distortions in Speech Enhancement due to Block Processing,” *In Proc. of Int. Conf. on Acoustic, Speech, and Signal Processing (ICASSP)*, March 2010.

[6] A. Favrot and C. Faller, “Perceptually Motivated Gain Filter Smoothing for Noise Suppression,” *In Proc. of Audio Engineering Society (AES) Convention*, Oct. 2007.

[7] S. Sivdas, “Time-Frequency Averaging of Noise Compensation Filters for ASR,” *In Proc. of Int. Conf. on Acoustic, Speech, and Signal Processing (ICASSP)*, May 2006.

[8] P.P. Vaidyanathan, ”Multirate Systems and Filter Banks,” *Englewood Cliffs, NJ: Prentice-Hall,* 1993.

[9] M. Triki, D.T.M. Slock, and Ahmed Triki, “Periodic Signal Extraction with Frequency-Selective Amplitude Modulation and Global Time-Warping for Music Signal Decomposition,” *In Proc. of IEEE Work. on Multimedia Signal Processing (MMSP)*, pp.972-977, Oct. 2008.

[10] M. Triki, “Harmonize-Decompose Audio Signals with Global Amplitude and Frequency Modulations,” *In Proc. of Int. Conf. on Digital Audio Effects (DAFx)*, Sep. 2010.

[11] M. Triki and D.T.M. Slock, “A Novel Voiced Speech Enhancement Approach Based on Modulated Periodic Signal Extraction,” *In Proc. of European Signal Processing Conf. (EUSIPCO)*, Sept. 2006.

[12] S. Voran. “Objective estimation of perceived speech quality - part I: Development of the measuring normalizing block technique,” *IEEE Trans. on Speech, and Audio Processing*, Vol. 7, Issue 4, pp. 371-382, July 1999.

[13] A.W. Rix, J.G. Beerends, M.P. Hollier, A.P. Hekstra, “Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs,” *In Proc. of Int. Conf. on Acoustic, Speech, and Signal Processing (ICASSP)*, Vol. 2, pp. 749-752, May 2001.

[14] *ITU-T Recommendation P.862*, “Perceptual Evaluation of Speech Quality (PESQ): an objective method for end-to-end speech quality assessment of narrow-band telephone network and speech codecs,” 2001.

A. Linear Interpolation for Signal Reconstruction

Mathematical interpolation focuses on the estimation of an unknown value of a function f, defined on a regular grid N. If we restrict our consideration to a linear case, the desired solution will take the following general form:

$$f(x) = \sum_{n \in N} w(x,n) f(n) \tag{17}$$

where $f(x)$ is the unknown value , and $w(x,n)$ is a given linear weight function.
The linear weighting function must verify two properties:

- The interpolation of a constant function $f(n)$ remains constant (i.e., $\sum_{n \in N} w(x,n) = 1$)
- The interpolation at a given point n does not change the value $f(n)$ (i.e., $w(n,n) = 1$)

In addition, one can verify that mathematical interpolation is equivalent to *filtering* an impulse train carrying the signal sample with a continuous-time filter:

$$f(x) = \sum_{n \in \mathbb{N}} w(x-n) f(n) \tag{18}$$

where $w(.)$ characterizes the filter impulse response.
For instance, the nearest-neighbor interpolation can be achieved by filtering the signal using a rectangular window

$$w(x) = \left\{ \begin{array}{ll} 1 & \text{for } |x| < \frac{1}{2} \\ 0 & \text{otherwise} \end{array} \right. \tag{19}$$

The linear interpolation can, also, be performed by filtering the sampled signal with a continuous-time filter having a triangular (Bartlett) window

$$w(x) = \left\{ \begin{array}{ll} 1 - |x| & \text{for } |x| < 1 \\ 0 & \text{otherwise} \end{array} \right. \tag{20}$$

One can also use smoother windows to perform interpolation, such as Hanning window (we can easily verify that the resulting interpolating weight function satisfy the two previous properties). The use of a smooth window (with energy concentrated essentially on the principle lobe) reduces the interpolation errors.

B. Zero-Phase FIR Wiener Filter

In order to derive the coefficients of the FIR Wiener filter, we consider a signal $y(n)$ being fed to a Wiener filter of order L and with coefficients $g_t,\ t = -L, \cdots, L$. The zero-phase constraint is implemented via a symmetric non-causal design, i.e.,

$$g_{-t} = g_t \qquad \forall t \tag{21}$$

The output of the filter (denoted $x(n)$) is given by

$$\begin{aligned} x(n) &= \sum_{t=-L}^{L} g_t\, y(n-t) \\ &= g_0 y(n) + \sum_{t=1}^{L} g_t\ (y(n-t) + y(n+t)) \end{aligned} \tag{22}$$

The Wiener filter is designed so as to minimize the mean square error, i.e,

$$g_t = \arg\min E\left\{e^2(n)\right\} \tag{23}$$

where $E\{.\}$ denote the expectation operator, and $e(n) = x(n) - s(n)$ is the residual error.
Setting the gradient to zero ($\frac{\partial}{\partial g_t} E\left\{e^2(n)\right\} = 0$) leads to the $L+1$ linear system

$$\begin{bmatrix} \mathbf{P}(0,0) & \mathbf{P}(0,1) & \cdots & \mathbf{P}(0,L) \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \mathbf{P}(L,0) & & \cdots & \mathbf{P}(L,L) \end{bmatrix} \begin{bmatrix} g_0 \\ \vdots \\ \vdots \\ g_L \end{bmatrix} = \begin{bmatrix} \mathbf{p}(0) \\ \vdots \\ \vdots \\ \mathbf{p}(L) \end{bmatrix} \tag{24}$$

where $\mathbf{P}$ and $\mathbf{p}$ are functions of the auto $r_{yy}(.)$ and cross $r_{sy}(.)$ correlation of the received signal $y(n)$ with the clean signal $s(n)$:

$$\left| \begin{array}{ll} \mathbf{P}(0,0) = r_y(0) & \\ \mathbf{P}(i,0) = \mathbf{P}(0,i) = 2\, r_y(i) & i = 1:L \\ \mathbf{P}(i,j) = \mathbf{P}(j,i) = 2\ (r_y(i-j) + r_y(i+j)) & i,j = 1:L \end{array} \right.$$

$$\left| \begin{array}{ll} \mathbf{p}(0) = r_{sy}(0) & \\ \mathbf{p}(i) = 2\, r_{sy}(i) & i = 1:L \end{array} \right.$$

The filter coefficients are estimated by inverting (24), and the enhanced signal is processed as in (22).