On the Development of STFT-analysis and ISTFT-synthesis Routines and their Practical Implementation

Hristo ZHIVOMIROV

Department of Theory of Electrical Engineering and Measurements, Technical University of Varna, 1 Studentska Str., Varna, Bulgaria

Abstract **– This work introduces the development of two software routines for Short-Time Fourier Transform (STFT) and Inverse Short-Time Fourier Transform (ISTFT) along with know-how about their practical implementation. The proposed algorithms and the corresponding novel Matlab® functions form a conjugated analysis-synthesis pair and assist the time-frequency analysis, processing, resynthesis and visualization of real-world non-stationary signals. In the paper, background information is given about the mathematical framework of the STFT, ISTFT and the perfect reconstruction condition. Further, the proposed algorithms and routines are described, based on DFT, IDFT and the overlap-add (OLA) synthesis approach. Comprehensive notes and recommendations are given concerning the applied aspects of their usage. Finally, examples are provided to confirm the consistency of the algorithms and routines, including analysis, spectral modification and resynthesis of non-stationary signals, along with example for the outperformance of the proposed analysis function in comparison with the corresponding in-built Matlab® function.**

Keywords **– STFT, ISTFT, analysis, synthesis, development, routine.**

DOI: 10.18421/TEM81-07
https://dx.doi.org/10.18421/TEM81-07

Corresponding author: Hristo ZHIVOMIROV,
Department of Theory of Electrical Engineering and Measurements, Technical University of Varna, Varna, Bulgaria
Email: hristo_car@abv.bg

Received: 10 December 2018.
Accepted: 29 January 2019.
Published: 27 February 2019.

1. Introduction

The behavior of the real-world signals (which are predominantly non-stationary) could be observed in fullness by time-frequency (TF) representation into the time-frequency domain (TFD), providing information for both the time and frequency localization of the signal. Moreover, one might be interested in performing modifications into the TFD (*e.g.*, a time-varying thresholding) and then resynthesizing of the modified signal.

The TF analysis and (re)synthesis are used in various applications and signal processing techniques: speech enhancement, perceptual audio coding, cross-synthesis, time-varying filtering, noise reduction, pitch shifting, time-scale modification, spectral subtraction, etc.; in the fields of the signal processing, acoustics and vibration, radar and sonar engineering, machinery diagnosis, power system analysis, measurements, medicine, etc. [1, 2].

One of the most intuitive and straightforward approaches to perform TF analysis and synthesis is to use a pair of localized forms of the Fourier transform termed Short-Time Fourier Transform (STFT) and Inverse Short-Time Fourier Transform (ISTFT) – a generalizations of the Gabor transform and Gabor expansion [3], respectively, presented by D. Gabor in 1946. Further, the idea of the STFT/ISTFT analysis/synthesis is extensively developed between 1965 and 1980, mainly associated with the names of pioneers J. L. Flanagan, R. M. Golden, M. R. Portnoff, L. R. Rabiner, R. W. Shafer, J. B. Allen, R. E. Crochiere. In our days, the names of B. Boashash and J. O. Smith could be distinguished.

B. Boashash wrote [1]: "at the last years, the focus became more about refining implementations and simplifying the concepts, methods, and techniques to make them available to a wider audience", so this is a purpose of the publication as well.

The present paper is focused on the development of specialized routines for performing STFT and ISTFT and their practical implementation in the

Matlab® software environment, based on a generalized algorithm firstly described by R. E. Crochiere in [4]. The newly proposed Matlab® functions form a conjugated analysis-synthesis pair and are a new impact in the field of the applied signal processing (to date, there is no such in-build pair of functions in Matlab®). The paper also gives a straightforward and consistent explanation of the mathematical background, as well as an insight on the practical aspects of the TFD processing.

2. Background

Mathematical framework of the STFT

The mathematical framework of the STFT is well described by the author in [5]. Let $x[n]$ be a real-world (*i.e.*, mathematically real and time limited) discrete signal. Let's extract signal frames (segments), denoted as $x_l[m]$ at regular time intervals using a finite window function $w[m]$, expressed as [6]

$$x_l[m] = x[m + lH]w[m], \tag{1}$$

where $m \in \{1, 2, ..., M\}$ is the local time index (*i.e.*, an index relative to the start of the sliding extraction window), $M \in \mathbb{N}$ is the analysis window length, $l \in \{0, 1, ..., L-1\}$ is the frame index, $L \in \mathbb{N}$ is the total number of frames; $H \in \mathbb{N}$ is the hop size (*i.e.*, the time advance, expressed in samples, from one signal frame to the next). In this manner every signal frame is actually translated to time zero.

Further, Discrete Fourier Transform (DFT) is performed on every frame $x_l[m]$, given a localized two-sided spectrum [6]

$$\dot{X}[k,l] \triangleq \frac{1}{M}\sum_{m=1}^{K} x_l[m] e^{-j2\pi\frac{mk}{K}}, \tag{2}$$

where $k \in \{1, 2, ..., K\}$ is the frequency bin index and $K \in \mathbb{N}$ is the DFT size.

The term $\dot{X}[k,l]$ is called STFT of $x[n]$ and corresponds to the local time-frequency behavior of the signal around the time index lH and the frequency bin k. If the sampling frequency is denoted as f_s, the above indices correspond to time $t = {lH}/{f_s}$ and linear frequency $f = {kf_s}/{K}$ [6].

Usually $K = M$; for $K > M$ the sequence $x_l[m]$ is zero padded and the spectrum is interpolated. Similarly, when $H < M$ the STFT is *sui generis* interpolated over the time axis. One must be aware that neither of these interpolations actually improves the frequency or time resolution.

Mathematical framework of the ISTFT. Perfect reconstruction overlap-add condition.

First of all, some implicit requirements must be met so STFT to be invertible [6, 7]:

- $\dot{X}[k,l]$ should exist, hence $w[m] \neq 0$ over its length;
- the signal should be not undersampled into the time domain (to avoid frequency domain aliasing), that is $H \leq M$;
- the signal should be not undersampled into the frequency domain (to avoid time domain aliasing) *i.e.*, $K \geq M$.

Moreover, recovering a given time sequence $\{\dot{X}[k,l] \mid k = \text{const.}\}$ with respect to the time variable l require that the time sampling interval H meets the Nyquist criterion based on the bandwidth of the analysis window function (that is, the bandwidth of the signal frame) [7]

$$\frac{f_s}{H} \geq 2B\frac{f_s}{M} \Rightarrow H \leq \left\lfloor \frac{M}{2B} \right\rfloor, \tag{3}$$

where B is the half-main-lobe width of the analysis window (measured to the first zero), normalized to that of the rectangular one $\left(\equiv {f_s}/{M}\right)$.

The ISTFT begins with inverse DFT of $\dot{X}[k,l]$ to recover $\tilde{x}_l[m]$ [6]:

$$\tilde{x}_l[m] \triangleq \sum_{k=1}^{K} \dot{X}[k,l] e^{+j2\pi\frac{mk}{K}}. \tag{4}$$

As previously mentioned, in some applications, one is interested in making modifications to the STFT-matrix. When such TF processing is performed (*e.g.*, time-varying thresholding), discontinuities may appear at the synthesis frame edges. In this case, a new windowing procedure must be performed on every signal frame after the IDFT with a synthesis window $v[m]$, in order to defeat the possible signal artefacts [8].

Further, the signal frame is translated from the local time index $m = n - lH$ back to the global time index $n = m + lH$:

$$\begin{aligned} \tilde{x}_l[m]v[m] &= \tilde{x}_l[n-lH]v[n-lH] = \\ &= \tilde{x}[n]w[n-lH]v[n-lH] \end{aligned}. \tag{5}$$

Finally, in order to obtain the original signal one must overlap and add (OLA) all (modified) weighted signal frames in the time domain [6]:

$$\begin{aligned}\tilde{x}[n] &= \frac{H}{E_{wv}} \sum_{l=1}^{L} \tilde{x}[n] w[n-lH] v[n-lH] \\ &= \frac{H}{E_{wv}} \tilde{x}[n] \sum_{l=1}^{L} w[n-lH] v[n-lH]\end{aligned} , \quad (6)$$

where $E_{wv} = \sum_m w[m] v[m]$ is the mutual energy (synergy) of the analysis and synthesis window functions (also interpreted as coherent gain of a new window function $w[m]v[m]$).

From Eq. (6) follows that the OLA must be conducted in such a way, so both the analysis and synthesis windows to be cancelled in the time-domain, that is, to overlap-add to a constant [2, 6]:

$$\sum_l w[n-lH] v[n-lH] = const., \; n \in \{1,2,...,N\} . \quad (7)$$

Eq. (7) is called OLA-constraint or perfect reconstruction OLA condition. If Eq. (7) is satisfied (that is, the perfect OLA is achieved), the constant is equal to $\frac{E_{wv}}{H}$ [2], so the reciprocal is the correction factor in Eq. (6).

The resulting processing routine is termed Weighted-OLA (WOLA) – a generalized case of ISTFT synthesis when different, but synergy complimentary analysis and synthesis windows are used. A practical example for WOLA is depicted in Fig. 1.

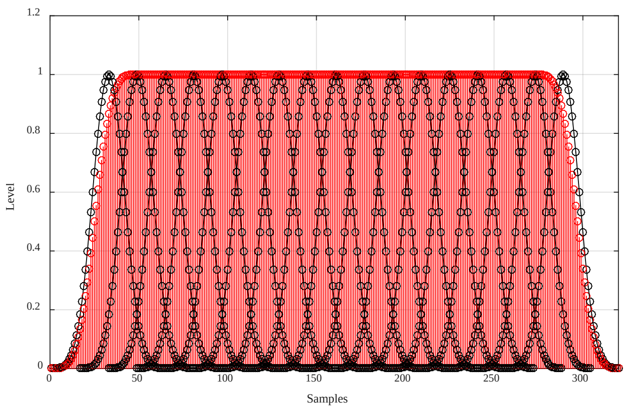

Figure 1. An example of a perfect WOLA using Blackman window for the STFT analysis and Hamming window for the ISTFT synthesis. For both windows $M = 64$ and $H = 16$. Note that the windows are synergy complementary.

Two specific cases are arisen in the applied ISTFT [2]: (i) the synthesis window is rectangular. This means that actually no synthesis window is used, since $v[m]=1, \; m \in \{1,2,...,M\}$. In this case, one denotes OLA as Constant-OLA (COLA) and it is said that the analysis window is amplitude complementary *i.e.*, $\sum_l w[n-lH] = const.$; (ii) the synthesis window is the same as the analysis one *i.e.*, $v[m] \equiv w[m]$, so it is said that the window functions are power (or energy) complementary.

At this point, one must be aware that the only way to satisfy the OLA-constraint is via proper choice of the hop size *H*. The maximum value of the hop size that still satisfy OLA-constraint is denoted as $H_{\max}$ and has different value for COLA and WOLA. Furthermore, in presence of modifications on $\dot{X}[k,l]$ the maximum hop size decreases to a new value, denoted as H_{mod}. As a rule of thumb $H_{\mathrm{mod}} \cong H_{\max}/2$ [2]. In Tab. 1 $H_{\max}$ and H_{mod} for some window functions are shown, in case of COLA and WOLA.

Table 1. Specific hop sizes for some window functions when COLA or WOLA is used, in presence or absence of TFD modifications

Window function	Specific hop sizes			
	COLA		WOLA†	
	H_{mod}	$H_{\max}$	H_{mod}	$H_{\max}$
Rectangular (no-window)	$M/2$	M	$M/2$	M
Bartlett (triangular)	$M/4$	$M/2$	∄	∄
Tukey ($\beta = 0.5$)	$M/12$ ‡	$M/6$ ‡	$M/16$ ‡	$M/8$ ‡
Hann	$M/4$	$M/2$	$M/6$	$M/3$
Hamming	$M/4$	$M/2$	$M/6$	$M/3$
Nuttall (minimum 4-term)	$M/8$	$M/4$	$M/14$	$M/7$
Blackman	$M/6$	$M/3$	$M/10$	$M/5$
Blackman-Harris (minimum 4-term)	$M/8$	$M/4$	$M/14$	$M/7$
FlatTop (minimum 4-term)	$M/10$	$M/5$	$M/18$	$M/9$
Chebyshev (-100 dB)	$M/8$ ‡	$M/4$ ‡	$M/14$ ‡	$M/7$ ‡
Common case [2]	$\frac{M}{2B}$	$\frac{M}{B}$	$\frac{M}{4B-2}$	$\frac{M}{2B-1}$

† – in case that the synthesis window is the same as the analysis one;
‡ – the OLA is not perfect, but the amplitude ripples are below 0.1 %.

When using the STFT for signal processing, the OLA-constraint is a must, while for measurement and display purposes this requirement can be neglected.

3. Description of the Proposed STFT & ISTFT Routines

The number of the successive signal frames across the length of the analyzed time sequence (*cf.* Fig. 2) is determined as

$$L = 1 + \left\lfloor \frac{N-M}{H} \right\rfloor. \tag{8}$$

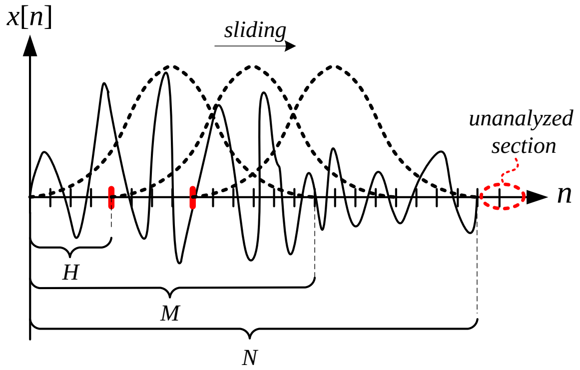

Figure 2. Graphical representation of the extraction and windowing of signal frames into the time-domain

The complex spectrum calculated from Eq. (2) exhibits Hermitian symmetry, as shown in Fig. 3 [9]. It is relevant to use and manipulate only the left half of the spectrum and then to make a conjugate flipped copy of it, in order to reconstruct the whole one. The number of unique spectrum lines in the spectrum is calculated as [10]

$$K_{uniq} = \left\lceil \frac{M+1}{2} \right\rceil. \tag{9}$$

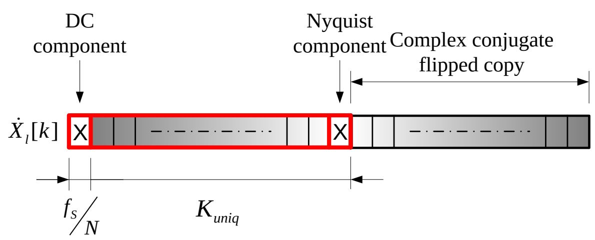

Figure 3. Structure of the discrete complex spectrum of an analyzed signal frame. The red frame indicates the unique part of the spectrum

Based on the Eq. (8) and Eq. (9), the size of the STFT-matrix is determined as $X \in \mathbb{C}^{K_{uniq} \times L}$, thus it could be preallocated in the code beforehand for sake of computational efficiency.

The main steps of the proposed STFT algorithm are as follows:

(1) a particular segment (*i.e.*, a signal frame) l of M samples is selected from the analyzed signal $x[n]$ and then is multiplied by the analysis window as $x[1+lH,...,M+lH] \cdot w[m]$. In this manner, the windowed segment of the signal is translated (slid) in the local (frame) time reference $x_l[m]$, instead of the global time reference $x_l[n-lH]$;

(2) a classical K-point FFT is made on the sequence $x_l[m]$ to produce the local spectrum $\dot{X}_l[k]$, according to Eq. 2;

(3) a STFT-matrix $\dot{X}[k,l]$ is formed with time increasing across the columns and frequency increasing down the rows. Every column of the matrix contains (only the unique) complex spectrum coefficients $\dot{X}_l[k]$. Every row of the matrix contains a set of DFT coefficients for a given fixed frequency versus time and could be interpreting as an output of a narrow-band-pass filter;

(4) the process is repeated along the whole length of the signal, with hop size of H samples.

A similar procedure could be derived to perform ISTFT, using the following basic steps:

(1) a reconstruction of the whole spectrum is made from the input one-sided STFT-matrix $\dot{X}[k,l]$;

(2) an IFFT on every STFT-matrix column is made. This operation leads to formation of new data matrix $x[m,l]$ with the initial windowed signal frames (*i.e.*, time series) across the rows and with time running across the columns;

(3) the resulting signal frames are then windowed by the synthesis window, converted from a frame time reference to the global time reference and summed in an overlap-add manner as per Eq. (6).

For convenience, the nomenclature compliance between the article and Matlab® routine notations are shown in Tab. 2.

Table 2. Nomenclature compliance between the article and Matlab® routine notations

In article	In routine	Data type
$x[n]$ / $\tilde{x}[n]$	`x`	vector
$x_l[m]$	`xw`	vector
$w[m]$	`win/awin`	vector
$v[m]$	`swin`	vector
N	`xlen`	scalar
M	`wlen`	scalar
H	`hop`	scalar
K	`nfft`	scalar
K_{uniq}	`NUP`	scalar
L	`L`	scalar
l	`l`	scalar
$\dot{X}_l[k]$	`X`	matrix
$\dot{X}[k,l]$	`STFT`	matrix
t	`t`	vector
f	`f`	vector
f_s	`fs`	scalar

STFT algorithm

Input: $x[n]$, $w[m]$, H, K, f_s

Determination of the signal length N

Determination of the analysis window length M

Determination of the K_{uniq} and L

for $l = 0, 1, \ldots, L\text{-}1$

step 1: signal segment windowing with the analysis window $x[1+lH,\ldots,M+lH]\cdot w[m]$

step 2: K-point FFT on the windowed segment $x_l[m]$ resulting the DFT vector $\dot{X}_l[k]$

step 3: update of the matrix $\dot{X}[k,l]$

$$\dot{X}[1,\ldots,K_{uniq},\ 1+l] = \dot{X}_l[1,\ldots,K_{uniq}]$$

end for

Time and frequency vectors generation

$$t = \left(\frac{M}{2}, \frac{M}{2}+H, \ldots, \frac{M}{2}+(L-1)H\right)\Big/ f_s$$

$$f = \left(0,\ldots,K_{uniq}-1\right)\cdot f_s / K$$

Output: $\dot{X}[k,l]$, t, f

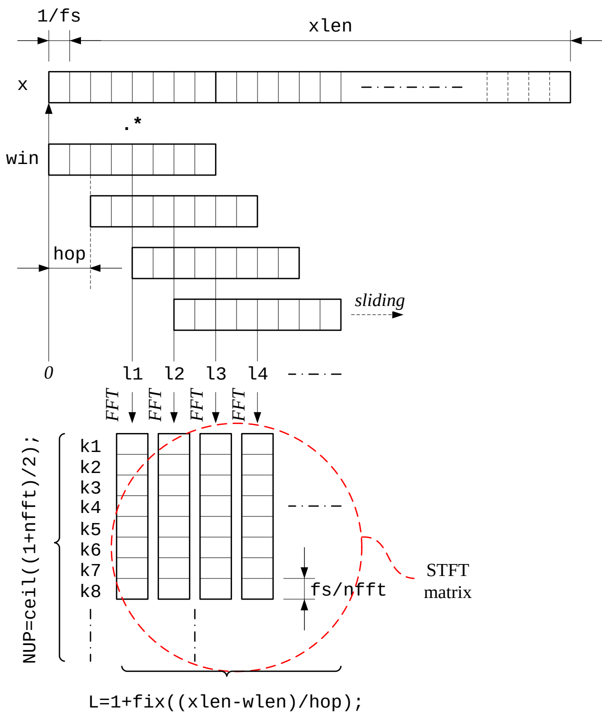

Figure 4. The proposed STFT processing workflow. For convenience, the Matlab® routine notations are used (cf. Tab. 2).

ISTFT algorithm

Input: $\dot{X}[k,l]$, $w[m]$, $v[m]$, H, K, f_s

Determination of the number of frames L

Determination of the synthesis window length M

Estimation of the signal length $\tilde{N}$

Reconstruction of the whole localized spectrum STFT-matrix $\dot{X}[k,l]$

Columwise IDFT on the $\dot{X}[k,l]$ resulting the time series matrix $x[m,l]$

for $l = 1, 2, \ldots, L$

step 1: weighted overlap-add (WOLA) of the signal segments using the synthesis window

$$x[1+(l-1)H,\ldots,M+(l-1)H] = x[m,l]\cdot v[m]$$

end for

Amplitude correction of the reconstructed signal by factor

$$H \Big/ \sum_m w[m]\cdot v[m]$$

Time vector generation

$$t = (0,\ldots,N-1)/f_s$$

Output: $\tilde{x}[n]$, t

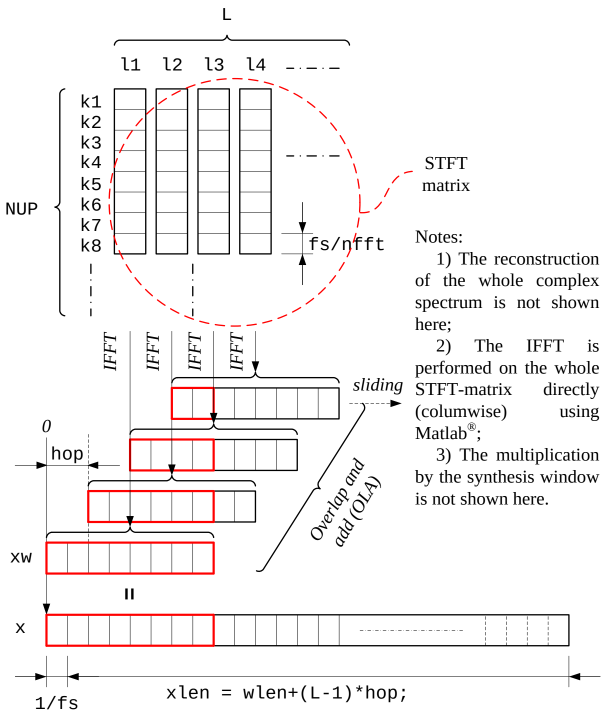

Figure 5. The proposed ISTFT processing workflow. For convenience, the Matlab® routine notations are used (cf. Tab. 2).

Software improvements

Several improvements are implemented into the developed software routines yielding better performance and more accurate results:

- the STFT-matrix is preallocated in advance, yielding acceleration of the code;

- the STFT complex coefficients are restricted to K_{uniq}, so the size of the STFT-matrix is reduced to half, providing memory optimization;

- the timestamps in the time vector t are localized at the temporal centers of the sliding windows (the time instances corresponding to the windows maximums in the time domain) instead at their beginnings, assuring more precise orientation when the TFD is visualized.

The proposed STFT and ISTFT Matlab® functions could be rewritten using entirely matrix techniques instead loop-structures, with creative use of the *buffer* and *bsxfun* functions and *padarray*, *circshift* and *arrayfun* functions, respectively, but on the expense of worsen readability and unnecessary complexity.

4. Analysis and Synthesis Savoir-Faire

Here, some useful addendums and good practices are noted to support the applied STFT analysis and ISTFT synthesis.

***OLA potpourri* [2]:**

- the OLA constraint could not be satisfied for hop sizes $H > \frac{1}{2}M$;

- all windows satisfy the OLA constraint for hop size $H = 1$ (aka sliding DFT);

- the rectangular window is the only one that satisfies the OLA for hop sizes $H = 0$ and $H = M$;

- once the OLA constraint is satisfied for hop size H, it remains true for all $H/h \in \square$, where $h = \{1, \ldots, H\} \in \square$.

***Tips and tricks*:**

- set $K \propto \{2^q \mid q \in \square\}$ for sake of computation acceleration;

- ensure $K > M + R - 1$ when filtering is implemented using filter with impulse response length R [6];

- prefer to use one and the same window for the analysis and synthesis in order to facilitate the attain of the WOLA;

- prefer to use Hann or Hamming windows due to their good properties and OLA behavior;

- avoid hop sizes less than the necessary since the successive signal frames become strongly correlated and hence no novel information could be extract, yet the execution time increases;

- always check if the OLA constraint is satisfied for the given window(s) length and hop size using an appropriate tool and never act just by intuition;

- if a visualization of the amplitude and/or phase of the $\dot{X}[k,l]$ is going to be implemented as (one-sided) spectrogram and/or phasogram, the following relations should be used, based on [11]:

$$Spectrogram \,\square\, \frac{2}{\sum_{m=1}^{M} w[m] \big/ N} \left|\dot{X}[k,l]\right|, \tag{10}$$

$$Phasogram \,\square\, \arg\left(\dot{X}[k,l]\right), \tag{11}$$

where $k \in \{1, \ldots, K_{uniq}\}$ and $\arg(\square)$ is the argument (phase) of $(\square)$.

- equation (8) indicates that unanalyzed samples may remain at the end of the signal $x[n]$, as far as the sliding window could not reach them, as can be seen in Fig. 6. In this case, the length of the synthesized signal $\tilde{x}[n]$ would be $M + (L-1)H < N$. This truncation could be avoided if the original signal $x[n]$ is padded with

$$M + \left\lceil \frac{N-M}{H} \right\rceil H - N \tag{12}$$

zeros at the end. Respectively, the same number of zeros must be removed from the end of the reconstructed signal $\tilde{x}[n]$;

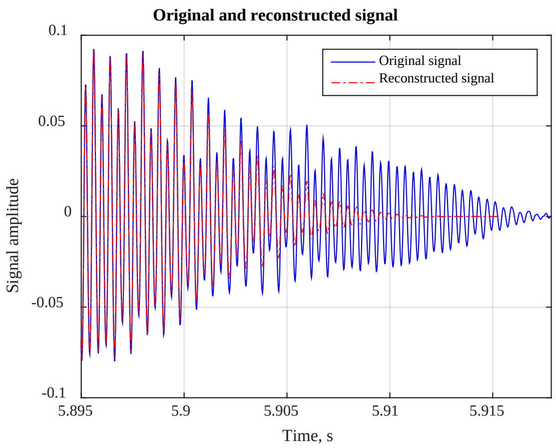

Figure 6. An example of amplitude attenuation and length reduction of the resynthesized signal, in comparison with the original (analyzed) one

- certain amplitude attenuation emerged at both ends of the resynthesized signal as a result of the insufficient number of overlapping windows (see Fig. 6.). This phenomenon could be overcome, when both ends of the signal are padded with $M-H+1$ zeros, so that the analyzed signal has length of $\tilde{N} = N + 2(M-H+1)$. Apparently, the same number of zeros must be removed from the edges of the resynthesized signal. When this correction is done simultaneously with the above one, $\tilde{N}$ should be used instead of N in Eq. 12.

An example of the overall result of both corrections is shown in Fig. 7.

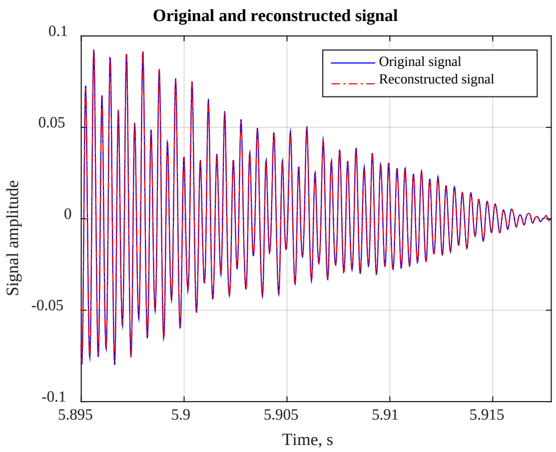

Figure 7. Comparison of the analyzed and resynthesized signals when special precautions are taken, as is described in Section 4. Note the perfect reconstruction

Resolution considerations

By TF-resolution one means the ability to distinguish two spectral peaks in the frequency domain and two individual events in the time domain. In the STFT/ISTFT context the resolution is determined by the analysis window function, which should have: (i) narrow main-lobe to ensure good frequency resolution; (ii) small level of the side-lobes for good amplitude resolution; (iii) shorter duration of the window (*i.e.*, window length) in order to resolve the closely spaced time events. Hence, a good choice will be a window with duration-bandwidth product as lower as possible. However, this product could not be arbitrary small and it is restricted by the Gabor limit [3]. Thus, for the sake of the applied TF analysis one must note:

- the window duration ($\cong M/f_s$) should be approximately equal to the typical time duration at which the signal is considered to be essentially stationary (*e.g.*, 20÷40 ms for human speech). The shorter window degrades the computational performance and the longer one leads to underestimation of the signal dynamics;
- when the resolution is paramount, the Blackman window (not to be confused with Blackman-Harris one) should be used – it has small bandwidth-duration product along with strong side-lobes attenuation and medium main-lobe width [12];
- if the time and frequency resolution that one can achieve in a spectrogram is insufficient, a two different spectrograms could be made: one with a short window for making more precise time measurements, and one with a larger window for frequency measurements. The two extreme cases are $M=1$, so $\dot{X}[k,l] = x[n]$ and $M = N$ yielding $\dot{X}[k,l] = \mathrm{DFT}\{x[n]\} = \dot{X}[k]$;
- when one considers the TF-visualization, two additional types of resolution are introduced: time grid resolution which is defined as the time between the beginnings of the successive frames, depending of the hop size H and frequency grid resolution depending on the DFT-size K. However, these quantities must not be confused with the actual time and frequency resolutions which depend on the analyzing window's length M and bandwidth B, respectively.

For detailed information about the window functions and their parameters the reader could refer to [11], [13].

5. Examples

A few examples are given using Matlab® to examine the proposed STFT/ISTFT functions.

First, a real-world sample of human speech – record "*DR2_FRAM1_SI522*" from the TIMIT database [14] is analyzed using the developed STFT function. The signal has duration T = 6.21 s and length $N = 99431$ samples at sample rate $f_s = 16000$ Hz. The resulting spectrogram is show in Fig. 8.

Further, for illustration purposes, two audio effects are implemented using the STFT/ISTFT pair – speech robotization and whisperization. The robotization effect is obtained by setting the STFT phase spectrum to zero, and the whisperization – by randomizing the phase spectrum [15].

The visual examination of the resulting spectrograms shows the effects of the TFD processing on the harmonic structure of the resynthesized signals. The robotized signal (Fig. 9.) has reshaped harmonic with constant pitch, but the vocal formants are preserved, so the speech sounds monotonically, yet recognizable. In the whispered signal (Fig. 10.) the vocal formants are also maintained, but any sense of pitch is completely eliminating, yielding smearing of the harmonics and thus imitates the whispered aspect of the voice. The analysis/synthesis parameters are: Hamming window (for both the analysis and synthesis) with $M = 1024$, $H = 128$ and $K = 4096$ for the robotization, and with $M = 128$, $H = 16$ and $K = 512$ for the whisperization, respectively.

The produced examples of analysis-modification-resynthesis and visualization are a good testimonial of the developed functions.

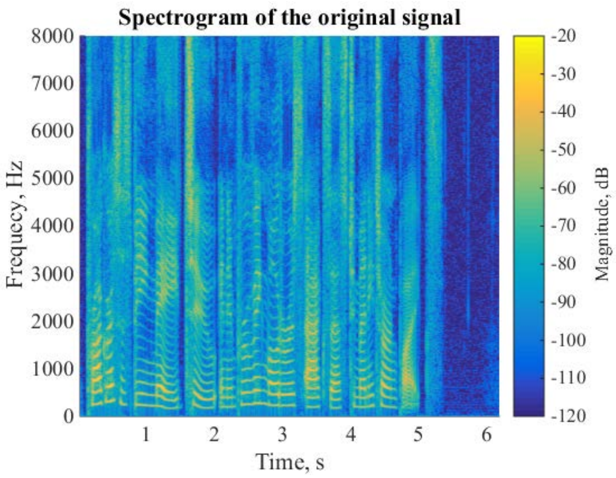

Figure 8. Spectrogram of the original speech signal. It reveals the harmonic structure of the signal and shows that its pitch is a function of time

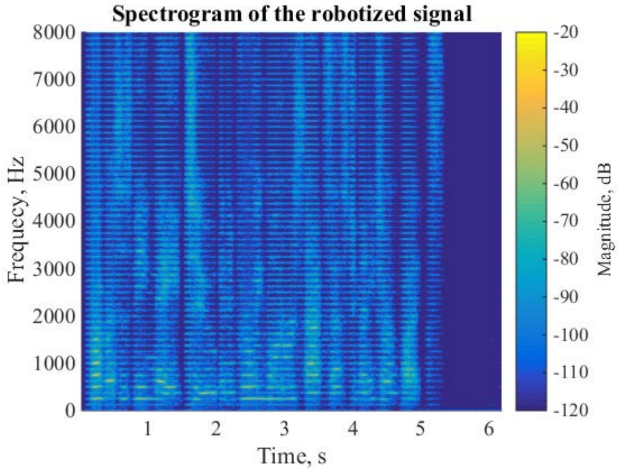

Figure 9. Spectrogram of the robotized signal. Note the constant pitch provided by setting all phases to zero

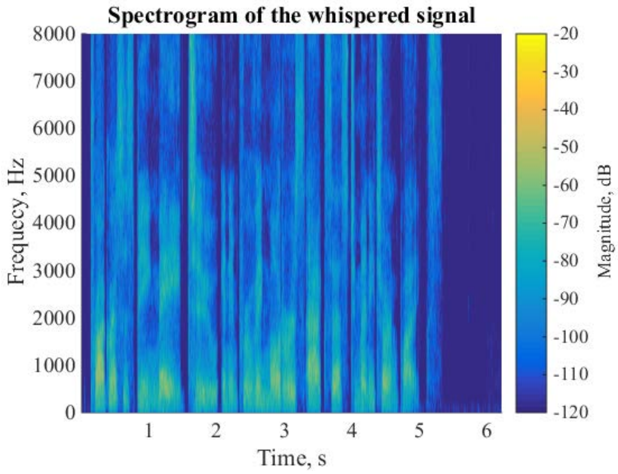

Figure 10. Spectrogram of the whispered signal. Note that any sense of pitch is completely eliminating due to the randomization of the phases

Finally, a speed test is conducted in order to show the outperformance of the proposed STFT routine in comparison with the in-build Matlab® function *spectrogram* [16]. The Matlab® in-built signal "*train*" was used for the tests, with duration T = 1.57 s and length $N = 12880$ samples at sampling frequency $f_s = 8192$ Hz, together with Hamming periodic window. The analysis conditions and test results are listed in Tab. 3. The data shows that the author's function is about 2.5 times faster than the Matlab® one.

Table 3. Execution time of the proposed and the in-built Matlab® STFT functions, averaged over 1000 runs

Analysis parameters	Author's function time, ms	Matlab® function time, ms	Execution time decreasing
M = 256, H = 256, K = 256	0.138	0.310	55.5 % (2.2 times)
M = 512, H = 256, K = 512	0.163	0.403	59.6 % (2.5 times)
M = 1024, H = 64, K = 4096	2.750	7.254	62.1 % (2.6 times)

6. Conclusions

In the presented work, novel routines are developed for TFD processing – STFT analysis and ISTFT synthesis (or perfect reconstruction), based on DFT, IDFT and the Weighted-OLA approach.

The STFT and ISTFT algorithms are implemented and tested in the Matlab® environment as a complementary ready-to-use functions, accessible at [17], [18], accompanied with a software product named "*OLAExam*" that assists the proper choice of the analysis/synthesis windows' length and hop size, in order to achieve synergy compliance and thus perfect reconstruction.

The statement is supported with comprehensive notes and recommendations concerning the applied aspects of the functions usage, along with an original approach for defeating of the edges' amplitude attenuation and length reduction of the resynthesized signal.

Examples of analysis, spectral modification and resynthesis of non-stationary signals are provided that confirm the consistency of the algorithms and routines, along with examples for the outperformance of the proposed analysis function in comparison with the corresponding in-built Matlab® function.

The developed software routines are a new impact in the TFD processing practice. They are of particular importance to the signal processing specialists. The paper also assists the better understanding of the STFT and ISTFT concepts and their practical application and therefore has high methodological value.

Acknowledgements

The author gratefully acknowledges the financial support by the project "Study of the electric power system stability and frequency control at a predominant share of renewable energy generation DN07/27 15.12.2016", in the framework of the program „Competition for financial support for fundamental research – 2016" by National Science Fund, Ministry of Education and Science, Bulgaria.

References

[1]. Boashash, B. (2015). *Time Frequency Signal Analysis and Processing: A Comprehensive Reference*. Oxford, Elsevier.

[2]. Smith, J. (2011). *Spectral Audio Signal Processing*. W3K Publishing.

[3]. Gabor, D. (1946). Theory of communication. Part 1: The analysis of information. *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering*, *93*(26), 429-441.

[4]. Crochiere, R. (1980). A weighted overlap-add method of short-time Fourier analysis/synthesis. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *28*(1), 99-102.

[5]. Zhivomirov, H. (2018). A Novel Visual Representation of the Signals in the Time-Frequency Domain. *University Politehnica of Bucharest Scientific Bulletin Series C-Electrical Engineering And Computer Science*, *80*(3), 75-84.

[6]. Benesty, J., Sondhi, M., & Huang, Y. (2008). *Springer Handbook of Speech Processing: Chapter 12*. Berlin, Springer.

[7]. Quatieri, T. (2001). *Discrete Time Speech Signal Processing: Chapter 7*. Upper Saddle River, Prentice Hall Press.

[8]. Dutoit, T., & Marques, F. (2009). *Applied Signal Processing: A MATLAB-based Proof of Concept*. New York, Springer.

[9]. Zhivomirov, H. (2018). A Method for Colored Noise Generation. *Romanian Journal of Acoustics and Vibration*, *15*(*1*), 14-19.

[10]. Ingle, V., & Proakis, J. (2012). *Digital Signal Processing using MATLAB: Chapter 5*. Boston, Cengage Learning.

[11]. Heinzel, G., Rudiger, A., & Schilling, R. (2002). *Spectrum and spectral density estimation by the Discrete Fourier transform (DFT), including a comprehensive list of window functions and some new flat-top windows*. Hannover, Max-Planck-Institut für Gravitationsphysik.

[12]. Zielinski, T. (2001). Joint time-frequency resolution of signal analysis using Gabor transform. *IEEE Transactions on Instrumentation and Measurement*, *50*(*5*), 1436-1444. DOI: 10.1109/19.963221.

[13]. Prabhu, K. (2014). *Window Functions and Their Applications in Signal Processing*. Boca Raton, CRC Press.

[14]. Garofolo, J. et al. (1993). *TIMIT Acoustic-Phonetic Continuous Speech Corpus LDC93S1*. Philadelphia, Linguistic Data Consortium.

[15]. Zölzer, U. (2011). *DAFX: Digital Audio Effects*. Chichester, John Wiley & Sons.

[16]. The MathWorks Inc. (2017). *Signal Processing Toolbox Reference*. Retrived from: https://www.mathworks.com/help/pdf_doc/signal/signal_ref.pdf [accessed: 10 December 2018].

[17]. Zhivomirov, H. (2018). Short-Time Fourier Transform (STFT) with Matlab. Retrieved from: https://www.mathworks.com/matlabcentral/fileexchange/45197-short-time-fourier-transform-stft-with-matlab , [accessed: 10 December 2018].

[18]. Zhivomirov, H. (2018). Inverse Short-Time Fourier Transform (ISTFT) with Matlab. Retrieved from: https://www.mathworks.com/matlabcentral/fileexchange/45577-inverse-short-time-fourier-transform-istft-with-matlab, [accessed: 10 December 2018].