TIME FREQUENCY SIGNAL ANALYSIS AND PROCESSING TOOLBOX UPDATE 6.2: AN ENHANCED RESEARCH PLATFORM WITH NEW ADVANCED HIGH-RESOLUTION TFDS

Boualem Boashash[1, 2, 3] *and Mubeen Ghafoor*[3]

[1]University of Queensland Centre for Clinical Research, University of Queensland, Brisbane, Australia
[2]Perinatal Research Centre, School of Medicine, University of Queensland, Brisbane, Australia
[3]Qatar University College of Engineering, Doha, Qatar

ABSTRACT

This paper describes the advancements, updates and improvements made in the new Time Frequency Signal Analysis TFSAP toolbox as compared with the previous TFSA toolbox version. The updates and improvements done in TFSA toolbox are in-line with the latest research done in recent few years in the field of time-frequency based signal analysis. TFSA Toolbox has proved in the past to be an efficient and popular tool for analyzing non-stationary signals. The new TFSAP Toolbox is an updated Matlab toolbox that extends the functionality of previous TFSA toolboxes. It provides additional options for generating new time-frequency distributions, and synthesizing a signal from its time-frequency distribution. It also includes options for analyzing real-life signals such as biomedical, speech and radar signals. Several demo scripts are also included in the new version to demonstrate the main functionality of the toolbox and to coach new users to use TFSA toolbox for advanced signal processing applications dealing with non-stationarities. The new version is renamed TFSAP 6.2; it can be downloaded for free as a service to the community.

1. INTRODUCTION

The majority of real-life signals have spectra that change over time, and should therefore be considered as non-stationary. Due to the non-stationarity of such real-life signals, joint Time-Frequency (TF) based analysis techniques are preferred over time only or frequency only based signal processing techniques [1]. For this purpose, previous Time-Frequency Signal Analysis (TFSA) toolbox versions provided users with an array of tools to analyze non-stationary signals accurately and efficiently [2]. This updated version TFSAP 6.2 extends and enhances previous TFSA versions continuously updated since the first release in 1987 at ISSPA 1987 in Brisbane, Australia.

1.1. Features in Previous Versions

The previous versions of TFSA toolboxes broadly addressed the issues of non-stationary signal generation, quadratic time-frequency distribution, Multi-linear time-frequency analysis, Instantaneous frequency estimation and tools for data visualization and plotting [2].
For quadratic time-frequency distributions, the algorithms included were Wigner-Ville Distribution, smoothed Wigner-Ville Distribution, cross Wigner-Ville Distribution, spectrogram, Rihaczek-Margenau, Gaussian Kernel, Born-Jordan and Zhao-Atlas-Marks distributions. Multi-linear time-frequency analysis is also available with the polynomial Wigner-Ville distribution of 4^{th} and 6^{th} order kernels. In addition, time-scale analysis includes algorithms for Daubechies coefficients computation. For instantaneous frequency (IF) estimation, a substantial suite of algorithms were also included. These are: the general phase difference, weighted phase difference, adaptive, zero-crossing, least square polynomial fit, peak of spectrogram, peak of Wigner-Ville distribution and peak of polynomial Wigner-Ville distribution IF estimators [4], [1].
To facilitate the use of TFSA tools and utilities, a graphic user interface (GUI) has been developed in Matlab with computational intensive routines written in C for efficient implementation. For Matlab and C interfacing, mex file formatting is used. The previous package of TFSA was written to run only 32-bit Windows or Linux operating system platforms. A user-guide and tutorial is also provided along with the TFSAP toolbox package. This tutorial and user-guide is designed to give an overview of the features and capabilities of the toolbox as well as provide a step by step procedure from installation to

978-1-4673-5540-7/13/$31.00 ©2013 IEEE

usage. The new version (TFSAP 6.2) extends this to 64-bit Windows and 64-bit Linux platforms.

1.2. Need for new features in TFSA package

Although time-frequency based signal analysis is a mature research area, a continuous improvement is still on for the development of more accurate and efficient algorithms/approaches. In particular, an intensive research effort has been done for designing and implementing high-resolution quadratic time-frequency distributions (TFD) as they have proved to be very effective and allowed improvements in terms of resolution and accuracy in the analysis and characterization non-stationary and/or multi-component signals. Although the quadratic class of TFDs offers good temporal and spectral resolutions simultaneously, these transforms suffer from cross-terms when analyzing multi-component signals. These cross-terms that resulted because of the quadratic nature of the transforms may be useful in some applications such as classification as they provide extra detail for the recognition algorithm. However, in some other applications, these cross-terms needed to be reduced by carefully choosing a smoothing function; this is achieved at the expense of resolution.

These quadratic TFDs can be considered as smoothed versions of the Wigner–Ville distribution (WVD) and the choice of a particular TFD depends on the specific application at hand and the properties that are desirable for that application. As a result, many new members of the quadratic class of TFDs have been proposed and the list is still increasing. So there is a need to update TFSA package with the more recent quadratic TFDs that have been proposed recently.

Some of the new members of the quadratic class of TFDs with new implementations are:

1) B-distribution (BD) [1]
2) Modified B-distribution (MBD) [1]
3) Extended Modified B-distribution (EMBD) [3]
4) Compact Support Kernel TFD (CSK) [5]
5) Extended Compact Support Kernel TFD (ECSK) [current research]

1.3. Test signals Update

In recent years, researchers explored Time-Frequency signal analysis as an effective tool in many new applications, such as in the fields of biomedical, speech processing and radar signal processing. This presented a greater need to improve TFSA toolbox and include the above mentioned quadratic TFD based tools to help researchers perform time-frequency based analysis in such new research areas.

More particularly, in the biomedical field, Time-frequency signal analysis has shown great potential in analyzing Electroencephalogram (EEG) and Heart Rate Variability (HRV) signals [5]. Electroencephalogram (EEG) signals are most commonly used to clinically assess brain activities, and to detect abnormalities such as epilepsy and seizure. However, manual detection of EEG abnormalities is very time consuming as it requires visual scanning of long recordings of EEG signals. Moreover, manual EEG analyses require a neurophysiologist, who could be prone to subjective judgment and error. Heart Rate Variability (HRV) signals are used to study the automatic nervous regulation of cardiovascular function and can be used for detection of perinatal hypoxia. EEG and HRV abnormalities analysis using TFDs is now a hot research area and these datasets are therefore desirable in a platform such as this TFSAP toolbox.

Synthesis of a time domain signal from a given Time-Frequency Distribution has also a few useful applications in signal processing. By using the signal synthesis property, any signal can be reconstructed from TFDs that have a minimum of desirable properties [1], allowing applications such as time-frequency filtering and signal enhancement.

In addition, the previous version of TFSA package supported only 32-bit Windows/Linux operating system. So with the advancement in technology, there is a need to make TFSA toolbox package compatible with new platforms/operating systems/machines; specially TFSA toolbox should be made compatible with 64-bit Windows/Linux operating systems. For all these reasons, the decision was made to expand the toolbox to new version TFSAP 6.2.

1.4. Formulations

Given a signal $s(t)$, we first form its analytic associate $z(t)$ as follows:

$$z(t) = s(t) + jH[s(t)] \quad (1)$$

where H is the Hilbert Transform.

We then define the Quadratic TFD as:

$$\rho_z(t,f) = W_z(\mathrm{t,f}) \underset{t\ f}{**} \gamma(\mathrm{t,f}) \quad (2)$$

where $\gamma(t,f)$ is a 2-D smoothing kernel and $W_z(t,f)$ is the WVD defined as [1, pp 38, 68]:

$$W_z(t,f) = \int_{-\infty}^{\infty} [z(t+\tau/2)z^*(t-\tau/2)]e^{-j2\pi\tau f}d\tau \quad (3)$$

2. UPDATES IN THE NEW VERSION

2.1. List of new Functions

In addition to the existing features of the previous versions of TFSA toolbox, the following updates are included in the new version of TFSA 6.2.

- Generation of time-frequency distributions
  - B distribution
  - Modified-B Distributions
  - Extended Modified-B Distributions

- Compact Support Kernel based Distributions
- Extended Compact Support Kernel based Distributions
- Ambiguity Function

- Signal Synthesis from the following time-frequency distributions:
    - Short-Time Fourier Transform
    - Spectrogram
    - Wigner-Ville distribution
- Demo Signals (signal1, whale1, bat1, eeg1, hrv1)
    - signal1 is an arbitrary Test signal
    - whale1 is a Whale signal
    - bat1 is a Bat signal
    - eeg1 is a Electroencephalogram (EEG) signal
    - hrv1 is a Heart Rate Variability (HRV) signal
- The current version of TFSA toolbox supports for four different Operating System platforms
    1) 32 bit Windows
    2) 64 bit Windows
    3) 32 bit Linux
    4) 64 bit Linux

The current release of TFSA (TFSAP 6.2) supports the following functions:

1. Generation of test signals and noise:
    - Linear FM
    - Quadratic FM
    - Cubic FM
    - Stepped FM
    - Sinusoidal FM
    - Hyperbolic FM
    - Gaussian and uniform noise
2. Generation of time-frequency distributions:
    - Wigner-Ville distribution
    - Smoothed Wigner-Ville distribution
    - Spectrogram
    - Rihaczek-Margenau-Hill distribution
    - Windowed-Rihaczek-Margenau-Hill distribution
    - Gaussian Kernel distribution
    - B distribution
    - Modified-B Distributions
    - Extended Modified-B Distributions
    - Compact Support Kernel based Distributions
    - Extended Compact Support Kernel based Distributions
    - Born-Jordan distribution
    - Zhao-Atlas-Marks distribution
    - Cross Wigner-Ville distribution
    - Polynomial Wigner-Ville distribution (order 6 kernel)
    - Polynomial Wigner-Ville distribution (order 4 kernel)
    - Ambiguity Function
3. Generation of time-scale signal representations:
    - Daubechies 4, 12 and 20 coefficient wavelet transforms
    - Scalogram
4. Instantaneous frequency estimation algorithms:
    - Finite phase difference:
        – First order (FFD)
        – Second order (CFD)
        – Fourth order
        – Sixth order
    - Weighted phase difference
    - Zero crossing
    - Adaptive LMS
    - Adaptive RLS
    - Least squares polynomial coefficients
    - Peak of the spectrogram
    - Peak of the Wigner-Ville distribution
    - Peak of the polynomial Wigner-Ville distribution
5. Signal Synthesis from the following time-frequency distributions:
    - Short-Time Fourier Transform
    - Spectrogram
    - Wigner-Ville distribution
6. Other DSP tools:
    - Analytic signal calculation
    - Power spectrum
    - Demo Signals (signal1, whale1, bat1, eeg1 and hrv1)
7. Data visualization routines
    - Plot
    - Image
    - PseudoColor
    - Waterfall
    - Mesh
    - Surf
    - Contour
    - Tfsapl

2.2. TFSAP Development Process

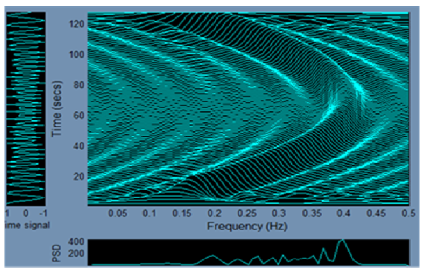
TFSAP has been developed in Matlab and C as an extension of a previous FORTRAN and C toolbox. The GUI of TFSAP is developed in Matlab while the computational intensive scripts are written in C for efficient implementation. The Matlab and C code communicates with each other using mex file formatting.
Adding a new function to TFSAP requires the coding of source files in C to perform the required function. Then TFSAP GUI files are edited to provide access to the new function. For example to add the MBD in TFSAP requires editing of the source code of quadratic function in C, as the MBD is a member of the Quadratic class of TFDs. Afterwards, the source code of quadratic TFD Analysis GUI is updated to incorporate the functionality of the MBD in GUI of the Quadratic TF Analysis.

3. RESULTS OF TIME-FREQUENCY SIGNAL ANALYSIS USING NEW TFSAP TOOLBOX

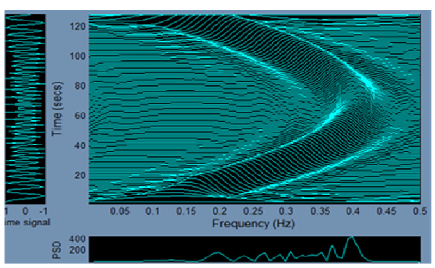
This section illustrates the usability of the updated TFSAP toolbox in real applications.

3.1. Results for a Quadratic FM signal

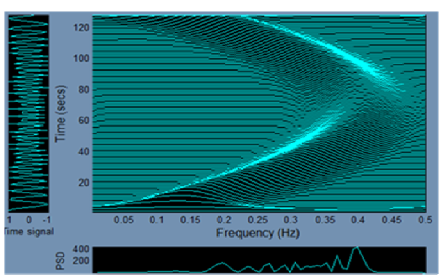
A comparison of Quadratic TFDs for a Quadratic FM signal is shown in figure 1. The frequency of Quadratic FM signal (length = 128 samples) ranges from 0.1 to 0.4 Hz. Figure 1 (a) shows Wigner-Ville distribution; figure 1 (b) shows Gaussian Kernel distribution with sigma 20; figure 1(c) spectrogram with smooth window ‘hamm’ and smooth window length 15; and figure 1(d) shows B-Distribution with beta 0.01 on Quadratic FM signal. All the TFDs are generated with lag window length 15, time-resolution 1 and length of FFT 128.



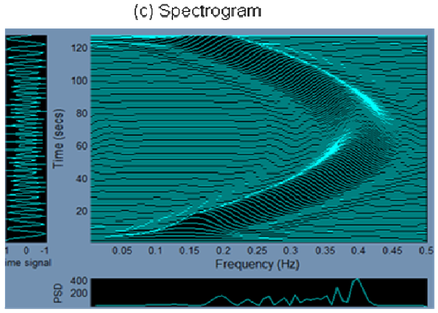
(a) Wigner-Ville distribution



(b) Gaussian Kernel Distribution



(c) Spectrogram



(d) B-distribution

Figure 1: Comparison of Quadratic TFD's for a quadratic FM signal. (a) Wigner-Ville distribution, (b) Gaussian Kernel Distribution, (c) Spectrogram, (d) B-distribution

3.2. Results for IF estimation of Quadratic FM signals

Instantaneous frequency IF estimation of Quadratic FM signal is shown in Figure 2. The IF is computed using three different methods; i.e. Zero-crossing, Phase Differentiation and Peaks of WVD, all available from TFSAP 6.2.

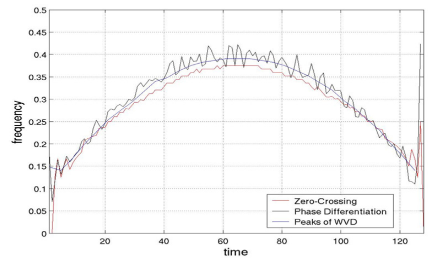
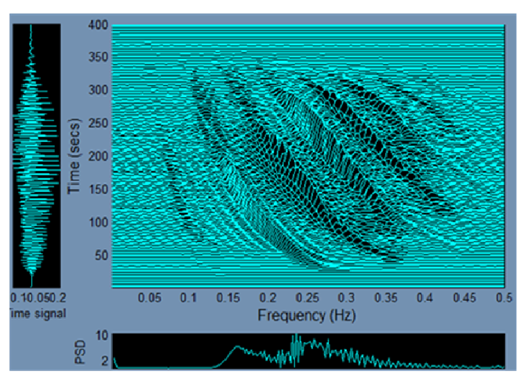


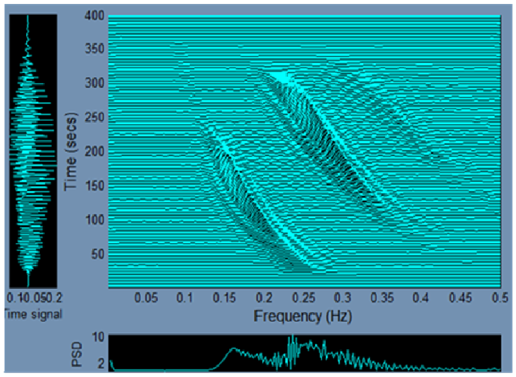
Figure 2: IF estimation of a Quadratic FM signal

3.3. Results for a Bat signal

In order to show the applications of TFSAP, several demo signals are also included in the toolbox i.e. bat signal, whale signal, EEG and HRV signals. A comparison of time-frequency analysis by using two different Quadratic TFDs on a bat signal sampled at 142 kHz is shown in figure 3. Figure 3 (a) shows Wigner-Ville Distribution of bat signal with lag window length 63 and time-resolution 3. While figure 3 (b) shows Modified B-Distribution of bat signal with lag window length 63 and time-resolution 2 and beta 0.01. From figure 3, it can be observed that the bat signal is multi-component with a decreasing non-linear FM. It can be also observed that Modified B-Distribution performs much better in suppressing cross-terms.



(a) Wigner-Ville Distribution of a bat signal

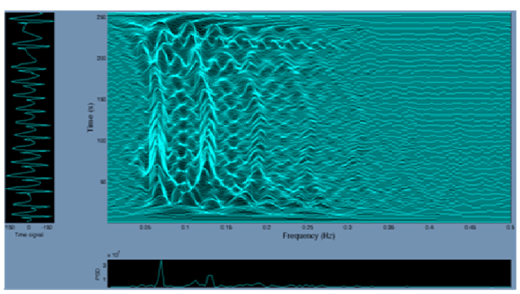


(b) Modified B-Distribution of a bat signal

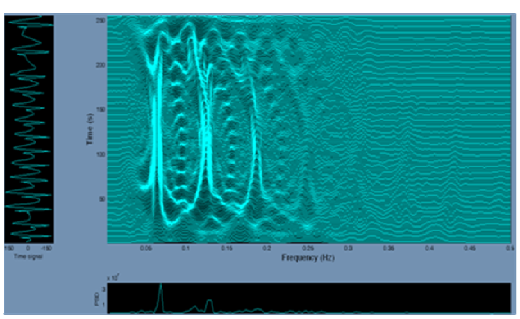
Figure 3: Comparison of WVD and MBD for Bat echolocation signal. (a) WVD of a bat signal, (b) MBD of a bat signal

3.4. Results for a EEG signal

An example of EEG signal analysis based on Quadratic TFDs is shown in figure 4. Figure 4 (a) shows Modified B-Distribution of EEG signal. While figure 4 (b) shows Extended Modified B-Distribution of EEG signal. From figure 4, it can be observed that the Extended Modified B-Distribution not only enhances the ridge structure of multi-component EEG signal better than Modified B-Distribution, but also performs better in suppressing the cross terms [3].



(a) Modified B-Distribution



(b) Extended-Modified B-Distribution

Figure 4: Comparison of MBD and EMBD on EEG signal. (a) MBD of EEG signal, (b) EMBD of EEG signal

3.5. Result for a Whale signal

A whale data is also included in the TFSAP 6.2 package. This signal contains 7000 data points and was collected at a sample rate of 8 kHz. Figure 5 shows EMBD of whale signal. It can be observed that this whale signal is mono-component with varying IF.

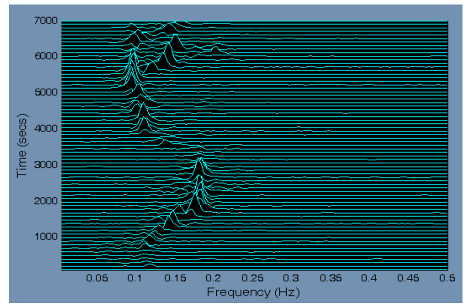
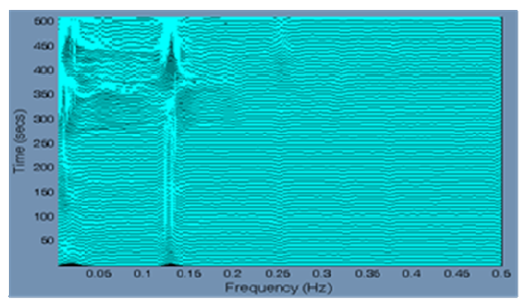


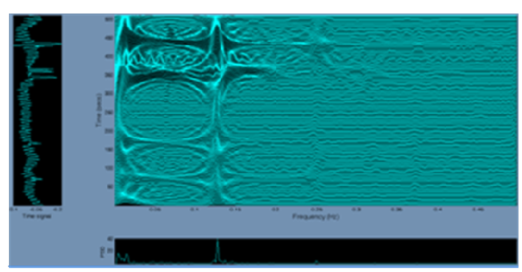
Figure 5: EMBD of whale signal

3.6. Result for a HRV signal

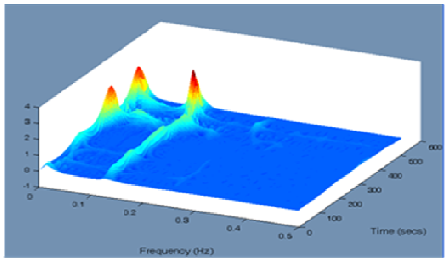
The HRV signals are extracted from Electrocardiogram (ECG) signals [5]. An HRV signal with length of 128 and sampling frequency 4 is included in TFSAP 6.2 package. Figure 6 shows an example of EMBD for this HRV signal. Figure 6 (a) and (b) shows the EMBD of the HRV signal using the tfsapl function but with different parameters of the EMBD. Figure 6 (c) shows the same TFD as in figure 6 (b) by using the surf plotting function.



(a) EMBD of HRV signal using tfsapl option



(b) EMBD of HRV signal using tfsapl option



(c) EMBD of HRV signal using surf option

Figure 6: EMBD of HRV signal, (a) and (b) EMBD of HRV signal using tfsapl function, both (a) and (b) are generated using different parameters of EMBD; (c) EMBD of HRV is plotted using surf function with same EMBD parameters as (b)

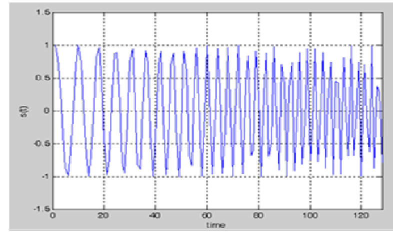
4. SIGNAL SYNTHESIS AND TF FILTERING

An example of signal synthesis using Wigner-Ville distribution method is shown in figure 7. Figure 7 (a) shows a linear FM signal with starting and ending frequency of 0.1 and 0.4 respectively with signal length 127. Figure 7 (b) shows the corresponding Wigner-Ville distribution of signal. Figure 7 (c) shows the synthesized time-domain signal using WVD method. Figure 7 (d) shows the WVD of synthesized time-domain signal. From figure 7, it can be observed that the synthesized and original are similar. Figure 8 shows signal synthesis and filtering of a noisy FM signal with starting and ending frequency of 0.1 Hz and 0.4 Hz respectively and having a Gaussian noise of 20 db.

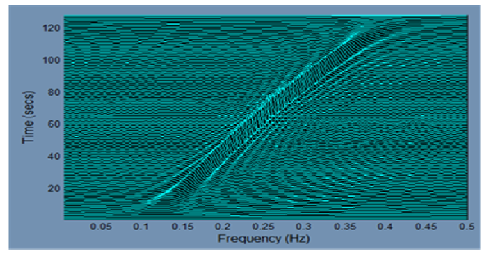
The following steps are performed to remove noise from this signal.

1) The instantaneous frequency of the signal is estimated by detecting peak locations in the WVD of the noisy signal.
2) The IF estimates are then used to create a binary image such that the value is one along the IF points and zero elsewhere.
3) The binary image is smoothed along the frequency axis by convolving it with the 2-D Gaussian filter.
4) The smoothed image is then used to mask the WVD.
5) The masked WVD is then synthesized to construct a filtered time-domain signal.

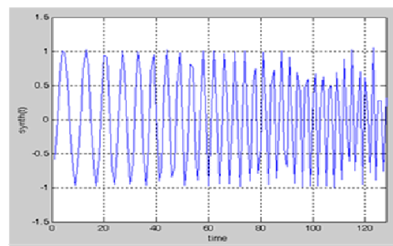
Figure 8(a) shows the time domain representation of the noisy signal. Figure 8(b) shows the WVD of the noisy signal. Figure 8(c) shows the Filtered TFD of the noisy signal. Figure 8(d) shows the time-domain representation of the filtered TFD; it shows that TF filtering has significantly reduced the effect of noise and the reconstructed signal closely matched the original one as shown in Figure 7 (a). In order to show the effect of filtering, the WVD of synthesized filtered signal is again computed as shown in Figure 8 (e). By comparing the results of figure 8 (b) with figure 8 (e), it can be observed that the noise of the synthesized signal is significantly reduced and the WVD of the synthesized signal is closer to the actual WVD of the original signal.



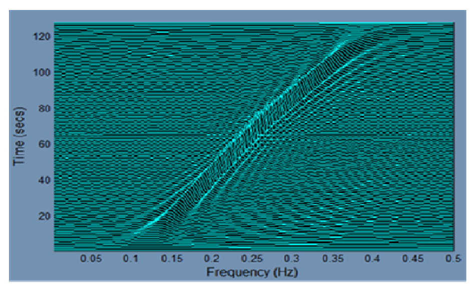
(a) Signal



(b) WVD of Signal

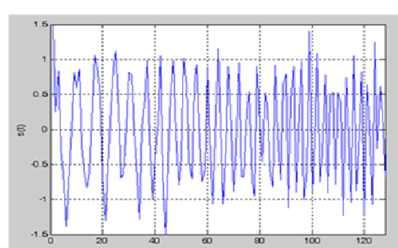


(c) Synthesized Signal



(d) WVD of synthesized signal

Figure 7: Signal Synthesis using WVD method. (a) Signal, (b) WVD of signal, (c) Synthesized Signal, (d) WVD of synthesized signal



(a) Signal + Noise

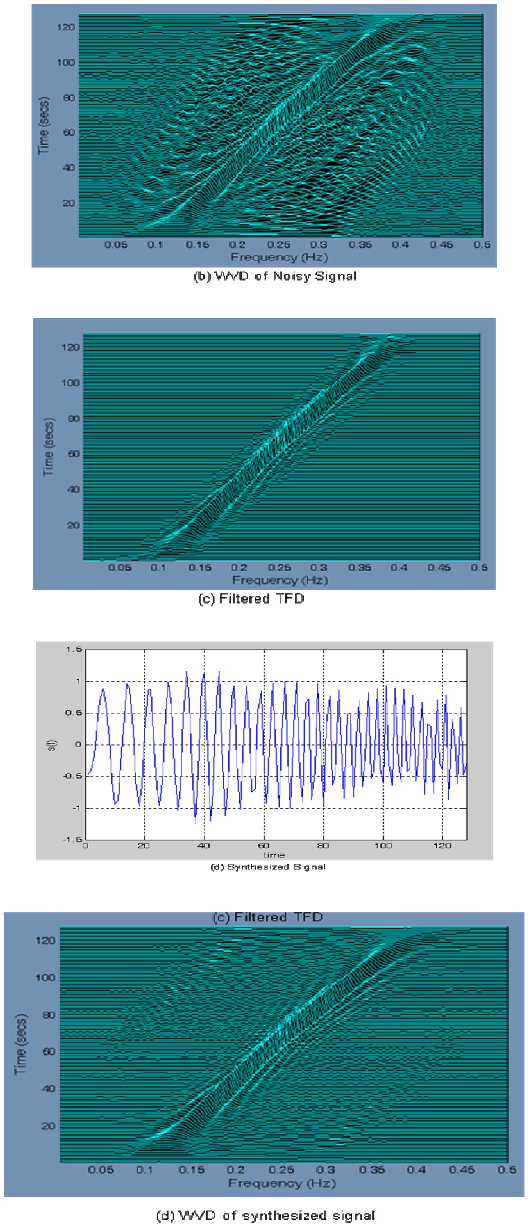


Figure 8: Signal Synthesis of a noisy signal using WVD based TF filtering. (a) Signal + Noise, (b) WVD of noisy signal, (c) Filtered TFD, (d) Synthesized Signal, (e) WVD of synthesized signal

5. CONCLUSION AND FUTURE VERSIONS

This new TFSAP toolbox 6.2 provides an excellent opportunity for new researchers to start their journey in the field of non-stationary signal analysis and allows the users to exploit the wide range of time-frequency analysis tools in different signal processing application. The new version of TFSAP 6.2 updates and extended the functionality of previous version of TFSA toolbox. The toolbox will continue to be updated to reflect new ideas and algorithms, and to improve functionality. Next version TFSAP 7 is intended to be also available in Octave and will include more biomedical data for testing and research.

6. ACKNOWLEDGEMENTS

This research was funded by Qatar Foundation grants NPRP 4-1303-2-517/1/2012, NPRP 09-465-2-174, NPRP 096262243, and UREP 11-148-2-056.
The authors wish to thank Dr. John O'Toole for his useful comments and feedback.

7. REFERENCES

[1] B. Boashash, *Time Frequency Signal Analysis and Processing: A Comprehensive Reference*, Elsevier, Oxford, UK, 2003.

[2] B. Boashash, "Efficient Software Implementation for the Upgrade of a Time-Frequency Signal Analysis Package," in *Proceedings of the Eighth Australian Joint Conference on Artificial Intelligence (AI-95),* Canberra, Australia, pp. 26-31, November 1995.

[3] B. Boashash, T. Ben-Jabeur, "Design of a high-resolution separable-kernel quadratic TFD for improving newborn health outcomes using fetal movement detection," in *Proceedings of the 11th International Conference on Information Science, Signal Processing and their Applications,* 2012, pp. 354–359.

[4] B. Boashash, "Estimating and interpreting the instantaneous frequency of a signal. I. Fundamentals," in *Proceedings of the IEEE,* vol. 80, no. 4, pp. 520- 538, 1992.

[5] B. Boashash, G.Azemi and J. O" Toole, "Time–frequency processing of non-stationary signals to aid diagnosis: highlights from selected medical applications," invited paper, *IEEE Signal Processing Magazine,* (in press)

[6] B. Boashash and A. Reilly, "*Algorithms for Time--Frequency Signal Analysis," in Time--Frequency Signal Analysis: Methods and Applications*, Ed. Melbourne 3205: Wiley Press, 1992.

8. APPENDIX

The Matlab time-frequency signal analysis and processing toolbox can be downloaded from the following link. http://espace.library.uq.edu.au/view/uq:211321. The corresponding code can be found in [6], [1].
This appendix takes a step-by-step approach to the use and understanding of time-frequency signal analysis and processing toolbox. It also has an adjunct purpose in that it familiarizes the user with TFSA 6.2 for MATLAB. TFSAP 6.2 has been written to run on a platform supporting MATLAB. To run TFSAP 6.2 you need to have MATLAB installed. In order to run the TFSAP 6.2 GUI, a minimum system requirement of 32 megabytes RAM on PCs is needed to avoid memory problems.

8.1. Installation

The distribution for Linux and MS-Windows is distributed pre-compiled. To install the distribution, just extract the zipped file and follow the instructions in the README text file.

8.2. Running TFSAP 6.2

To run TFSAP 6.2 on MS-Windows, all you need to do is click the MATLAB icon on the screen and then type tfsa6 at the MATLAB prompt. If you are running TFSAP 6.2 on UNIX workstations, invoke MATLAB from the shell by typing matlab <ENTER>1, and then type tfsa6 at the MATLAB prompt. After invoking TFSAP 6.2, the Main Menu of GUI will appear as shown in figure 9. The GUI of TFSAP 6.2 contains the pop-up menus and interface fields which control the various functions and tools available in the toolbox. The GUI provides an alternative to typing TFSAP 6.2 commands on the MATLAB command line to make it easy for starters.

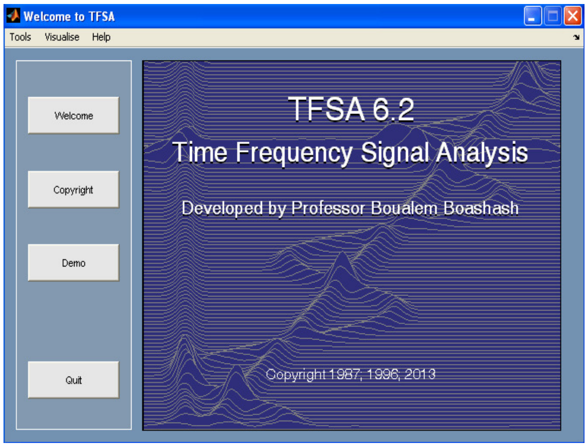


Figure 9: Main Menu of GUI of TFSAP 6.2 toolbox

It contains four buttons. These are:

• **Welcome**: Pressing this button will give general information regarding TFSAP 6.2.

• **Copyright**: The license agreement screen appears by clicking this button.

• **Demo**: Pressing this button gives a demonstration of TFSAP 6.2.

• **Quit**: Pressing this button will abort the TFSAP session.

The TFSAP 6.2 menu bar is composed of three major menu items: **Tools, Visualize,** and **TFSA Help**. Any menu item can be opened by placing the pointer on the corresponding item and then clicking the left hand side button of the mouse. You can access any menu sub-item by clicking with the left hand-side button of the mouse on the corresponding menu item and whilst holding this button down, dragging the mouse across the submenu bar until encountering the required sub-item, then releasing the mouse button.

8.3. Data accesses and Signal Generation GUI

The main GUI through which different types of non-stationary signals can be generated is shown in figure 10. In figure 10 an example of generating a Linear FM signal with length 128 and having starting and ending frequencies of 0.1 Hz and 0.4 Hz is shown.

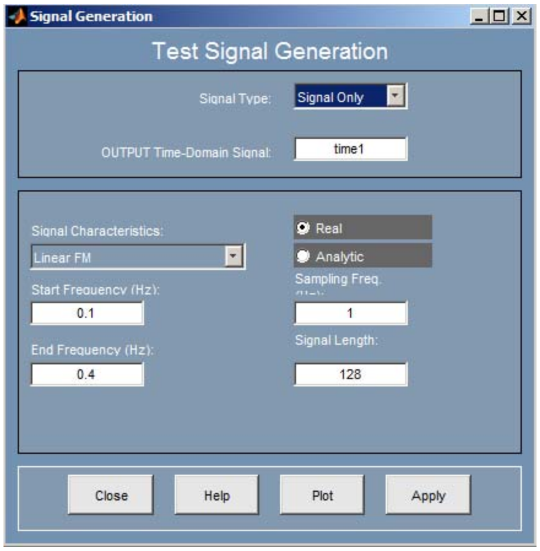


Figure 10: Signal Generation GUI

There is an also an option to generate real or analytic form of test signal. The complex analytic signal is an important concept in time-frequency signal analysis [1]. It can be generated by adding to a time-domain signal an imaginary part using the Hilbert transform. However, in TFSAP, the analytic signal is generated using an equivalent frequency domain method.

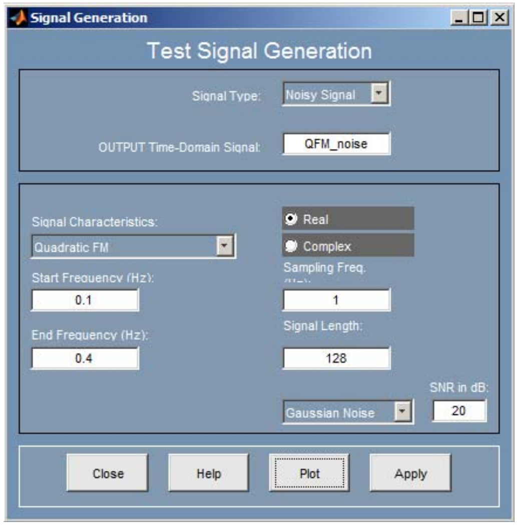


Figure 11: Generating Quadratic FM Signal with noise

The different types of test signals that can be generated from TFSAP 6.2 are listed in section 2.

There is also an option to generate noise only or a test signal with noise. An illustration to generate a Quadratic FM noisy signal is shown in Figure 11. A Gaussian noise of SNR 20 db is added with the signal. Length of the signal is 128 with starting frequency of 0.1 Hz and ending frequency of 0.4 Hz.
As previously mentioned, TSFAP 6.2 also includes some real-world non-stationary signals. These can be generated by selecting Demo Signal as signal type. Then examples of demo signals will appear in Demo Signal Tab as shown in figure 12.

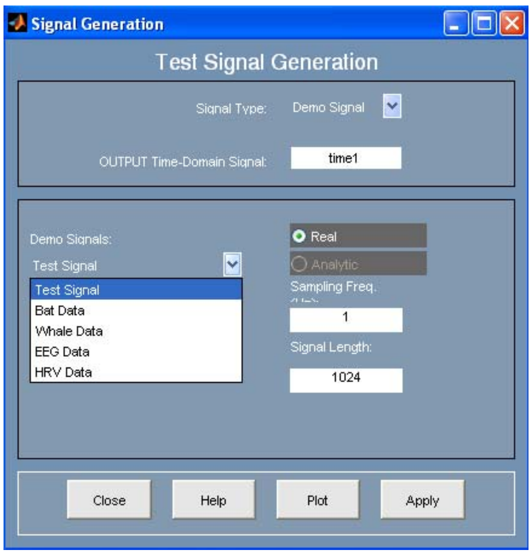


Figure 12: Generating Demo signal

All references to "signal" or "array" names in TFSAP 6.2 refer to variables in the MATLAB workspace. Values are read from and written to this space, and can be accessed normally from the MATLAB command line.
Thus two signals can be added together using the following procedure:
1. Generate time1 from the GUI Signal Generation pop-up.
2. Generate time2 from the GUI Signal Generation pop-up.
3. In the MATLAB command-line window, type
time3 = time1 + time2
4. Then, time3, the sum of the two generated signals, can now be accessed from the GUI.

8.4. Instantaneous Frequency (IF) Estimation GUI

Instantaneous frequency estimation is an important concept related to time-frequency analysis [4]. There are several techniques to compute IF e.g. differentiation schemes, adaptive techniques, peaks of TDS and zero crossings. The detailed list of different functions for estimating IF is given in section 2. For IF estimation of any signal, the user needs to perform the following tasks.

- Generate a time domain signal as explained in section 8.2.
- Open the IF estimation GUI.
- Enter the name of the signal in the Input Time Domain Signal textbox
- Select any IF estimation method from the list of methods shown in the algorithm tab and then press apply button to generate the IF.

Figure 13 shows an example of estimating the IF of a Quadratic FM signal using the IF Estimation GUI.

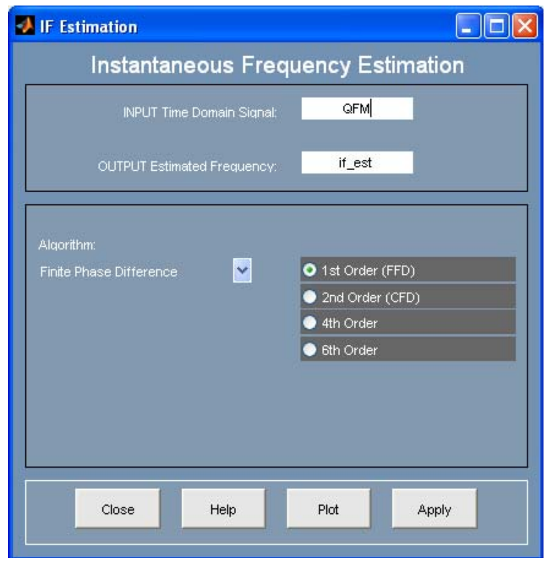


Figure 13: GUI for IF estimation

8.5. Quadratic TFD Analysis GUI

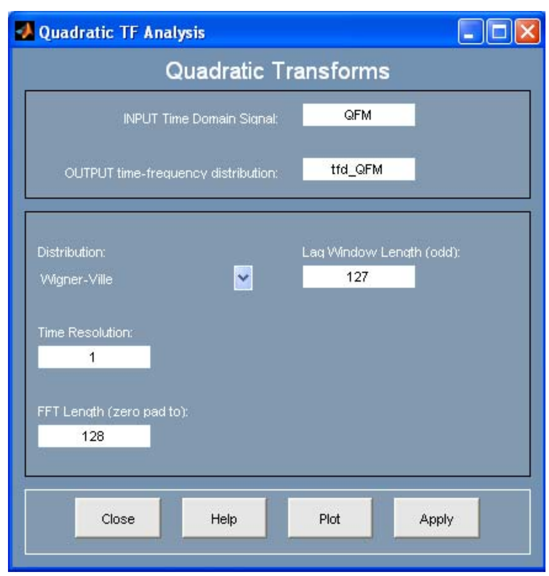


Figure 14: GUI for Quadratic TF Analysis

As explained earlier, Quadratic TFDs are powerful tools for analyzing non-stationary signals. In TFSAP 6.2, a

GUI is included that gives the option to generate different TFDs for any signal. For analyzing the Quadratic TFD of any signal, the user needs to perform the following tasks.

- Generate a time domain signal as explained in section 8.2.
- Open the Quadratic TF Analysis GUI.
- Enter the name of the signal in the Input Time Domain Signal textbox
- Select any Quadratic TFD method from the list of methods shown in the distribution tab
- Set the FFT length and Lag Window length and press apply to generate Quadratic TFD.

Figure 14 shows an example of estimating the WVD of a Quadratic FM signal using the Quadratic TF Analysis GUI.

8.6. Signal Synthesis GUI

As discussed in section 4, a time-domain signal can be efficiently synthesized from any time-frequency distribution. TFSAP 6.2 includes a GUI to help the users to explore and use the different features of signal synthesis in applications such as time-frequency filtering and signal enhancement. The main GUI for signal synthesis is shown in figure 15.

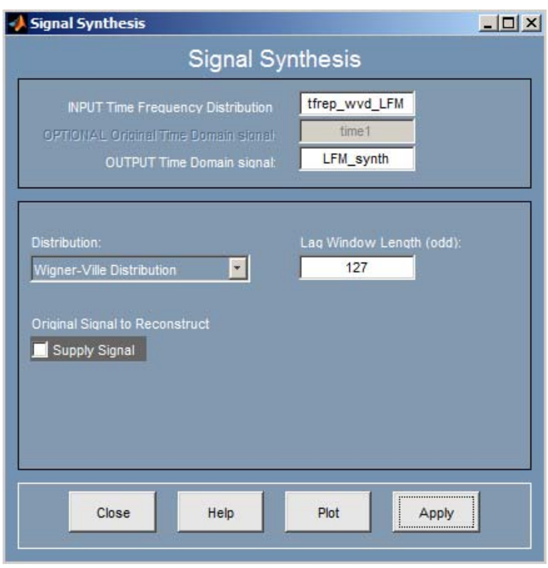


Figure 15: GUI for Signal Synthesis

For synthesizing a time-domain signal from a TFD, the users need to perform the following tasks.

- Generate a time-frequency distribution for any signal,
- Open the signal synthesis GUI
- Enter the name of the time-frequency distribution in the Input Time Frequency Distribution textbox
- Select any synthesis method along with the lag window length and then press apply to generate a time-domain signal.

8.7. Compatibility with other MATLAB functions

The GUI is an interface to the TFSAP 6.2 functions. All normal MATLAB features remain in place. For example, titles, labels, figure properties and variables may all be changed using the MATLAB command-line interface. Figures can be printed using the MATLAB's print command, as normal.

8.8. MATLAB command line interface

TFSAP 6.2 functions can be accessed from MATLAB in the same manner as the standard MATLAB functions.
For a list of functions in TFSA 6.2 type:
help tfsa6
Help on any of the individual functions can be obtained by typing:
help <functionname>
A detailed list of functions that TFSAP 6.2 supports is given in section 2.