SnT 2019
CTBT: SCIENCE AND TECHNOLOGY CONFERENCE

A Novel Approach for Signal Sparse Time-Frequency Representations

Victor Sucic[1*], Ivan Volaric[1], Goetz Bokelmann[2], Ronan Le Bras[3]
[1] University of Rijeka, Croatia; [2] University of Vienna, Austria; [3] CTBTO, Vienna, Austria.
*Corresponding author: vsucic@riteh.hr

T3.5-P5

Abstract

By exploiting the fact that most real-life signals are sparse in the time-frequency (TF) domain, a significant suppression of the unwanted cross-terms can be achieved in the signal TF representation. In this work, we propose a sparse reconstruction algorithm, based on the two-step iterative shrinkage/thresholding (TwIST) algorithm, in which the soft-thresholding value is adaptively determined by the Fast Intersection of the Confidence Intervals (FICI) rule. First, the TF region with the lowest mean value is determined, and then the thresholding value is set to the largest sample within the region. Examples of synthetic and real-life signals confirm that the performance of the proposed reconstruction algorithm is competitive to the performance of its state-of-the-art counterparts.

Quadratic Time-Frequency Distributions

The Wigner-Ville distribution (WVD), $W_z(t,f)$, is defined as the Fourier transform of the signal quadratic localized auto-correlation function, $R_z(t,\tau)$ [1]:

$$W_z(t,f)=\int_{-\infty}^{\infty}R_z(t,\tau)e^{-j2\pi f\tau}d\tau, \quad (1a)$$

$$R_z(t,\tau)=z\left(t+\frac{\tau}{2}\right)z^*\left(t-\frac{\tau}{2}\right). \quad (1b)$$

When the signal $z(t)$ contains only one linear frequency modulated component, the WVD provides an almost ideal TF representation. When this is not the case, the WVD introduces the unwanted artifacts [1]:

$$R_z(t,\tau)=\sum_{i=1}^{N_c}z_i\left(t+\frac{\tau}{2}\right)z_i^*\left(t-\frac{\tau}{2}\right)+\sum_{i=1}^{N_c}z_i\left(t+\frac{\tau}{2}\right)\sum_{\substack{j=1\\ j\neq i}}^{N_c}z_j^*\left(t-\frac{\tau}{2}\right), \quad (2a)$$

$$W_z(t,f)=\sum_{i=1}^{N_c}W_{z_i}(t,f)+2\sum_{i=1}^{N_c}\sum_{\substack{j=1\\ j\neq i}}^{N_c}\mathrm{Re}\left\{W_{z_{i,j}}(t,f)\right\}, \quad (2b)$$

where $W_{z_i}(t,f)$ and $W_{z_{i,j}}(t,f)$ are respectively the WVD of the i-th signal component (auto-term) and the cross-WVD between the i-th and j-th signal components (cross-term). In the ambiguity function, $A_z(\nu,\tau)$, calculated as the 2D Fourier transform of the WVD [1]:

$$A_z(\nu,\tau)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}W_z(t,f)e^{j2\pi(f\tau-\nu t)}dtdf, \quad (3)$$

the cross-terms are dislocated from the domain origin; thus, they can be removed by filtering the AF with a low-pass filter:

$$A_z'(\nu,\tau)=A_z(\nu,\tau)g(\nu,\tau), \quad (4)$$

where $g(\nu,\tau)$is the low-pass filter function, while $A_z'(\nu,\tau)$ is the filtered AF. However, since the auto-terms are not strictly located at the AF domain origin, the filtering process also affects the auto-terms, lowering their TF concentration. The class of TFDs obtained in such a way is commonly referred to as the Quadratic class of TFDs (QTFD).

Sparse Time-Frequency Distributions

In sparse TFDs the idea is to take a relatively small number of AF samples, in the process commonly referred to as the compressive sensing (CS). The selected samples should belong exclusively to the auto-terms; while the rest of the AF is calculated in a way which will produce the sparsest TFD. Let us rewrite (3) in the matrix form [2,3]:

$$\boldsymbol{\vartheta}_z(t,f)=\psi^H\mathbf{A}_z'(\nu,\tau), \quad (5)$$

where $\boldsymbol{\vartheta}_z(t,f)$ and $\mathbf{A}_z'(\nu,\tau)$ are respectively the filtered WVD and the AF, while $\boldsymbol{\psi}$ is the domain transformation matrix. Following the CS notation, we can also rewrite (4):

$$\mathbf{A}_z'(\nu,\tau)=\begin{cases}\mathbf{A}_z(\nu,\tau), & (\nu,\tau)\in\Omega,\\ 0, & \text{otherwise},\end{cases} \quad (6)$$

where $\mathbf{A}_z(\nu,\tau)$ is the discretized AF, while Ω is the $N_\nu'\times N_\tau'$ area located around the AF plane origin. Since a high resolution sparse TFD contains $N_t\times N_f$ samples, and $\mathbf{A}_z'(\nu,\tau)$ has only $N_\nu'\times N_\tau'\ll N_t\times N_f$ samples, Eq. (5) has multiple solutions. This is a well-known unconstrained optimization problem, which can be rewritten as [2,3]:

$$\widehat{\boldsymbol{\vartheta}}_z(t,f)=\arg\min_{\boldsymbol{\vartheta}_z(t,f)}\frac{1}{2}\|\boldsymbol{\vartheta}_z(t,f)-\boldsymbol{\psi}^H\mathbf{A}_z'(\nu,\tau)\|_2^2+\lambda c\left(\boldsymbol{\vartheta}_z(t,f)\right), \quad (7)$$

where $c\left(\boldsymbol{\vartheta}_z(t,f)\right):\mathbb{R}^2\to\mathbb{R}$ is the sparsity inducing regularization function, while λ is the regularization parameter. The first approach in solving this problem is centered around the ℓ_0-norm based regularization function. The ℓ_0-norm is the best sparsity inducing function; however, solving (7) becomes NP-hard, thus an iterative greedy algorithm needs to be implemented in order to find a good local minimum of (7), instead of the global one. The second approach relies on the easier to solve ℓ_1-norm, the only convex norm among the sparsity inducing ℓ_q-norms. The convexity of the ℓ_1-norm guaranties that the local minimum is also a global one, making (7) much easier to solve. By using the ℓ_1-norm as the sparsity inducing function, we can further rewrite (7) as [2,3]:

$$\boldsymbol{\vartheta}_z^{\ell_1}(t,f)=\arg\min_{\boldsymbol{\vartheta}_z(t,f)}\|\boldsymbol{\vartheta}_z(t,f)\|_1,\ \text{s.t.}\ \|\boldsymbol{\vartheta}_z(t,f)-\boldsymbol{\psi}^H\mathbf{A}_z'(\nu,\tau)\|_2^2\leq\epsilon, \quad (8)$$

where ϵ is the user-predefined parameter which defines the acceptable solution tolerance. By introducing the proximity operator to (8), we can further ease this problem by finding the closed-form expression which can be iteratively solved:

$$\boldsymbol{\vartheta}_z^{\ell_1}(t,f)=\mathrm{soft}_\lambda\left\{\boldsymbol{\vartheta}_z^{\ell_1}(t,f)\right\}, \quad (9)$$

$$\mathrm{soft}_\lambda\left\{\boldsymbol{\vartheta}_z^{\ell_1}(t,f)\right\}=\mathrm{sgn}\left(\boldsymbol{\vartheta}_z^{\ell_1}(t,f)\right)\max\left(\left|\boldsymbol{\vartheta}_z^{\ell_1}(t,f)-\lambda\right|,0\right). \quad (10)$$

One simple, yet effective algorithm built around this idea is the TwIST algorithm:

$$\left[\boldsymbol{\vartheta}_z^{\ell_1}(t,f)\right]^{[n+1]}=(1-\alpha)\left[\boldsymbol{\vartheta}_z^{\ell_1}(t,f)\right]^{[n-1]}+(\alpha-\beta)\left[\boldsymbol{\vartheta}_z^{\ell_1}(t,f)\right]^{[n]}+\beta\mathrm{soft}_\lambda\left\{\left[\boldsymbol{\vartheta}_z^{\ell_1}(t,f)\right]^{[n]}+\boldsymbol{\psi}^H\left(\mathbf{A}_z'(\nu,\tau)-\boldsymbol{\psi}\left[\boldsymbol{\vartheta}_z^{\ell_1}(t,f)\right]^{[n]}\right)\right\}, \quad (11)$$

where $0<\alpha\leq 1$ and $0<\beta<2\alpha$ are the TwIST parameters. With a low threshold value, the algorithm converges slower and achieves better accuracy. On the other hand, a larger threshold value decreases the algorithm execution time by lowering the algorithm accuracy. We can achieve advantages of both the small and high threshold values by starting with a larger value, and decreasing it with each iteration. With this idea in mind, we introduce the FICI rule in order to adaptively calculate the threshold value. Let us denote the vectorized and ascendingly sorted argument of the soft-thresholding operator in (11) with $[\varsigma_{z1D}'(i)]^{[n+1]}$. In the first step of the FICI algorithm, the hard-thresholding is performed [3]:

$$[\varsigma_{z1D}(i)]^{[n+1]}=\mathrm{hard}_{\lambda_h}\{[\varsigma_{z1D}'(i)]^{[n+1]}\}, \quad (12)$$

$$\mathrm{hard}_{\lambda_h}\{\varsigma_{z1D}'(i)\}=\begin{cases}\varsigma_{z1D}'(i), & |\varsigma_{z1D}'(i)|>\lambda_h,\\ 0, & \text{otherwise},\end{cases} \quad (13)$$

with a low value hard-thresholding parameter λ_h. The idea behind this thresholding is to eliminate the low value amplitudes, which would only increase the FICI algorithm execution time. Starting from the first non-zero element, $\varsigma_{z1D}(i_0)$, we calculate the upper and lower bounds of the confidence interval [3]:

$$D_u(i_0+\Delta i)=\overline{\varsigma_{z1D}}(i_0+\Delta i)+\Gamma\,\widehat{\varsigma_{z1D}}(i_0+\Delta i), \quad (14a)$$

$$D_l(i_0+\Delta i)=\overline{\varsigma_{z1D}}(i_0+\Delta i)-\Gamma\,\widehat{\varsigma_{z1D}}(i_0+\Delta i), \quad (14b)$$

where $0\leq\Delta i\leq(N_tN_f-i_0)$ is the current window size, Γ is the user defined ICI threshold value, while $\overline{\varsigma_{z1D}}(i_0+\Delta i)$ and $\widehat{\varsigma_{z1D}}(i_o+\Delta i)$ are respectively the mean value and the standard deviation of the samples inside the current window. In the next step of the FICI algorithm, we calculate the relative amount of intersection [3]:

$$R(i_0+\Delta i)=\frac{D_{u_{min}}(i_0+\Delta i)-D_{l_{max}}(i_0+\Delta i)}{2\Gamma\,\overline{\varsigma_{z1D}}(i_0+\Delta i)}, \quad (15)$$

$$D_{u_{min}}(i_0+\Delta i)=\min\left(D_u(i_0),\ldots,D_u(i_0+\Delta i)\right), \quad (16a)$$

$$D_{l_{max}}(i_0+\Delta i)=\max\left(D_u(i_0),\ldots,D_u(i_0+\Delta i)\right). \quad (16b)$$

The FICI algorithm runs iteratively until: $R(i_0+\Delta i)\geq R_c$, where $0<R_c\leq 1$ is the threshold value. With the obtained value of Δi^+, we can set the threshold value for the current TwIST iteration:$[\lambda]^{[n+1]}=[\varsigma_{z1D}(i_0+\Delta i^+)]^{[n+1]}$.

Experimental Results

The performance of the FICI-TwIST algorithm is tested on modeled, $z_{MOD}(t)$, and real-life seismic signals, $z_{RL}(t)$, shown in Fig. 1. The algorithms are evaluated based on their execution time and the TFD concentration measure:

$$M_z^S=\frac{1}{N_tN_f}\left[\sum_{t,f}^{N_tN_f}\left(\boldsymbol{\vartheta}_z(t,f)\right)^{1/2}\right]^2. \quad (17)$$

Table I: Sparse TFD concentration measures.

	FICI-TwIST[3]	NESTA[4]	YALL1 [5]
$z_{MOD}(t)$	0.0130	0.3634	0.0060
$z_{RL}(t)$	0.0099	0.3179	0.0101

Table II: Reconstruction algorithm execution times.

	FICI-TwIST[3]	NESTA[4]	YALL1[5]
$z_{MOD}(t)$	27.552	25.302	189.27
$z_{RL}(t)$	1.7301	7.9177	24.753

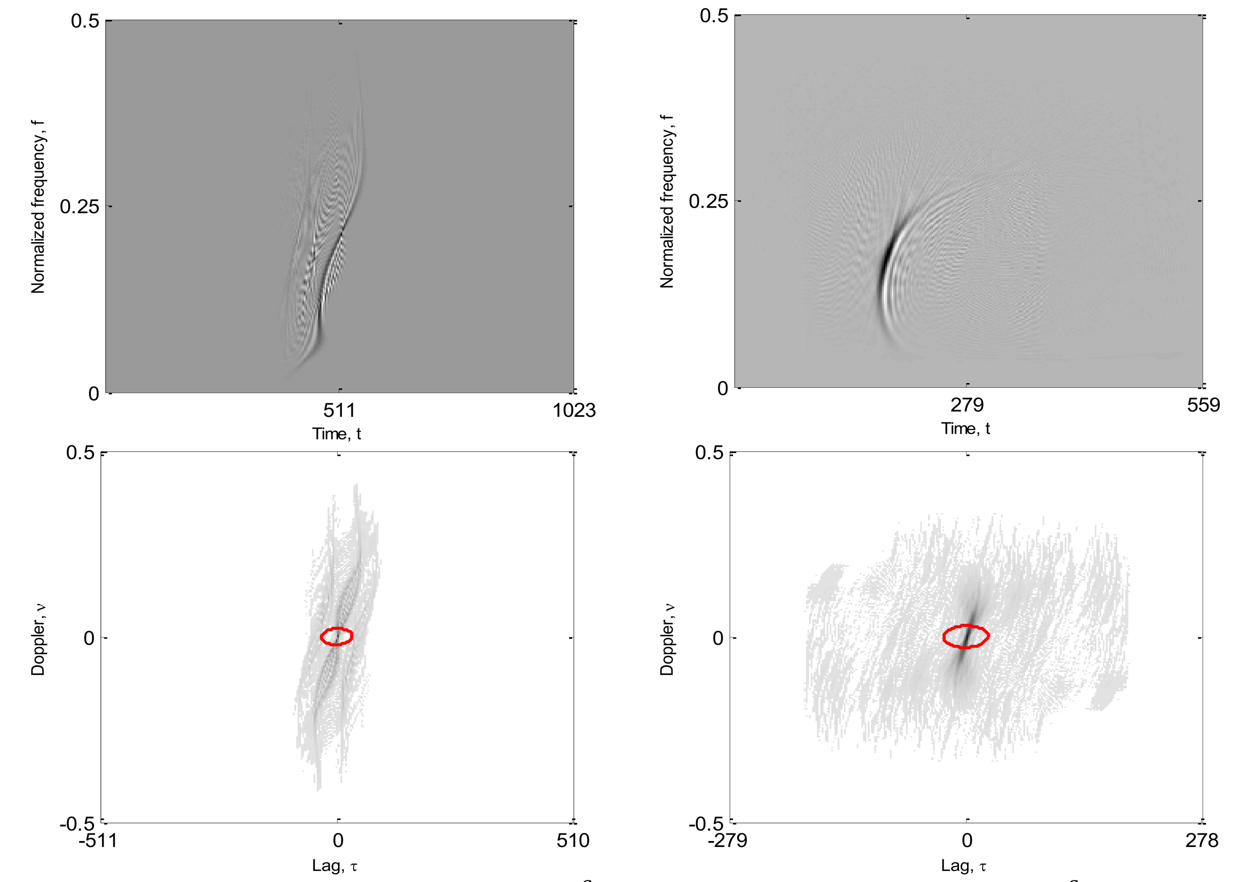

Fig. 1. Test signals: (a) WVD of the modeled signal, $M_z^S=0.4987$; (b) WVD of the real-life signal, $M_z^S=2.1266$; (c) AF of the modeled signal with the selected CS-AF area (ellipse enclosed within $N_\tau'=47, N_\nu'=31$); (d) AF of the real-life signal with the selected CS-AF area (ellipse enclosed within $N_\tau'=37, N_\nu'=23$).

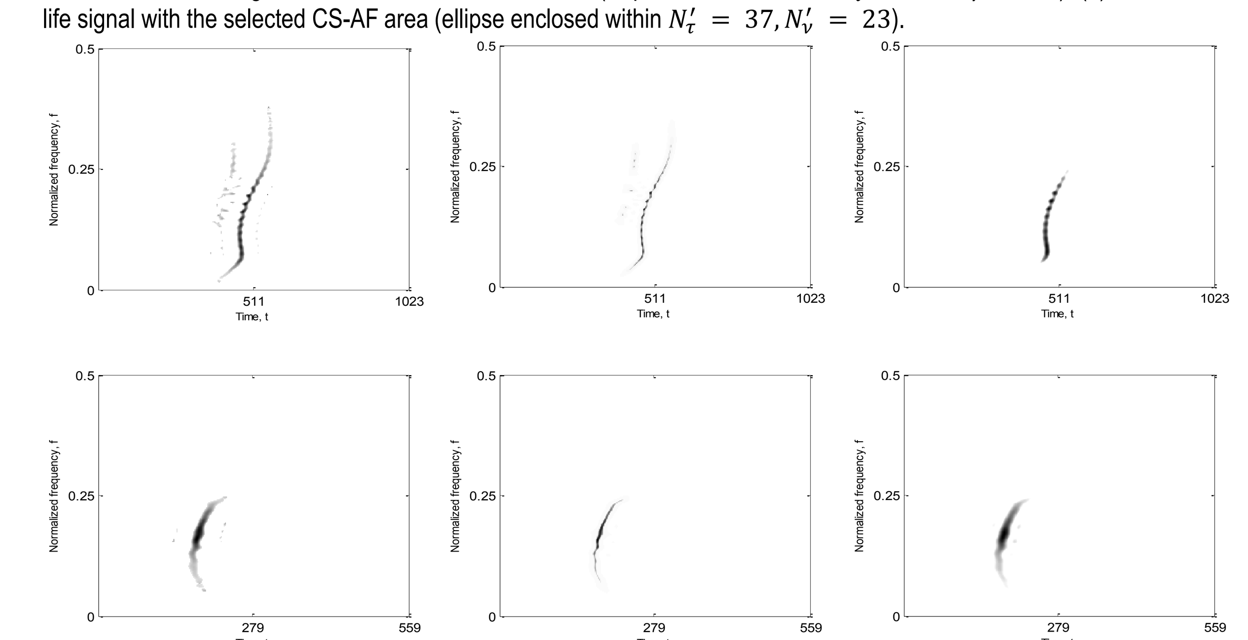

Fig. 2. Sparse TFDs of the: (a) modeled signal, FICI-TwIST; (b) modeled signal, NESTA; (c) modeled signal, YALL1; (d) real-life signal, FICI-TwIST; (e) real-life signal, NESTA; (f) real-life signal YALL1.

Conclusion

We have presented the FICI-TwIST algorithm for the sparse TFD reconstruction. The algorithm eliminates the need for threshold value selection, as required by the original TwIST algorithm, by introducing the FICI rule for the adaptive threshold value calculation. The FICI rule finds the TFD region with the lowest amplitude values based on the intersection of the confidence intervals and accordingly sets the threshold value. The theoretical considerations have been confirmed by tests on both modeled and real-life seismic signals.

References

[1] B.Boashash. Time-Frequency *Signal Analysis and Processing, 2nd Edition A Comprehensive Reference*, Elsevier 2016.
[2] I. Volaric, V.Sucic, S. Stankovic: „A data driven compressive sensing approach for time-frequency signal enhancement",*Signal Processing*,vol.141,pp.229-239, 2017.
[3] I. Volaric, V. Sucic: „Sparse time-frequency distributions based on the l1-norm minimization with the fast intersection of confidence intervals rule", *Signal, Image and Video Processing*, vol. 13, no. 3, pp. 499-506, 2019.
[4] S. Becker, J. Bobin, E.J. Candes: „NESTA: a fast and accurate first-order method for sparse recovery", *SIAM Journal on Imaging Sciences*, vol.4, pp. 1-39, 2011.
[5] J. Yang, Y. Zhang: „Alternating direction algorithms for l1-problems in compressive sensing", *SIAM Journal on Scientific Computing*, vol. 33, pp.250-278, 2011.

Acknowledgement

This work has been fully supported by the University of Rijeka under the project number UNIRI-TEHNIC-18-67.