R. Lyons and C. Howard, "Improvements to the Sliding Discrete Fourier Transform Algorithm [Tips & Tricks]," in IEEE Signal Processing Magazine, vol. 38, no. 4, pp. 119-127, July 2021, doi: 10.1109/MSP.2021.3075416.

Improvements to the Sliding DFT Algorithm

by Richard Lyons and Carl Howard (July 2021)

This article presents two networks that improve upon the behavior and performance of previously published sliding DFT (SDFT) algorithms.

The proposed networks are structurally simple, computationally-efficient, guaranteed stable networks used for real-time sliding spectrum analysis. The first real-time network computes one spectral output sample, equal to a single-bin output of an *N*-point discrete Fourier transform (DFT), for each input signal sample. The second real-time network is frequency-flexible in that its analysis frequency can be any scalar value in the range of zero to one half the input data sample rate measured in Hz.

Background

The process of sliding spectrum analysis can be defined as successive *z*-transforms of *N* consecutive samples of an *x*(*n*) input sequence evaluated at an arbitrary point z_o on the *z*-plane. Figure 1(a) shows a general sliding spectrum analysis network, comprising a specialized comb filter followed by a complex resonator, that computes those sliding (successive) *z*-transforms. Figure 1(b) shows the network's *z*-plane pole/zero locations when, for example, $z_o = Me^{j\alpha}$ and $N = 8$. The network's pole is located a $z = z_o$, whose magnitude is less than one, where angle α is measured in radians.

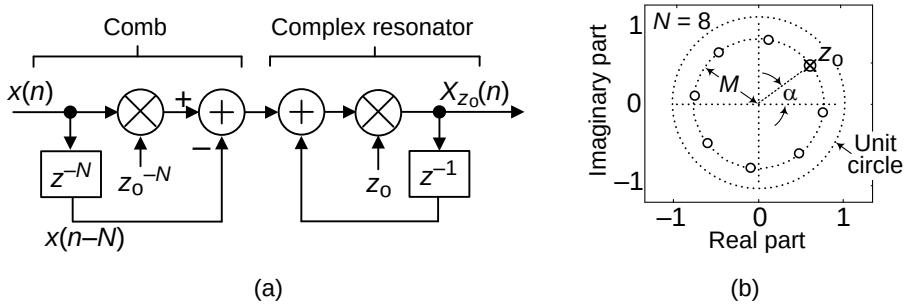

FIGURE 1. General *N*-sample sliding *z*-transform: (a) network structure; (b) *z*-plane pole/zero locations for $z_o = Me^{j\alpha}$ and $N = 8$.

The time-domain difference equation and the *z*-domain transfer function of the Figure 1(a) network are:

$$X_{z_o}(n) = z_o[X_{z_o}(n-1) + x(n)z_o^{-N} - x(n-N)] \qquad (1)$$

and

$$H(z) = \frac{X_{z_o}(z)}{X(z)} = \frac{\left(z_o^{-N} - z^{-N}\right)z_o}{1 - z_o z^{-1}}. \qquad (2)$$

The development of the Figure 1(a) network and derivation of (1) are given in Appendix A. The network in Figure 1(a) is the foundation upon which we will build our proposed SDFT networks.

The Sliding DFT

Of particular interest in sliding spectrum analysis is the sliding discrete Fourier transform (SDFT). The SDFT algorithm is an efficient spectrum analysis technique when only a few frequency bin outputs, on a real-time basis, of an *N*-point DFT are desired [1,2]. The primary application of the SDFT is to detect and estimate the instantaneous peak-to-peak amplitude (instantaneous magnitude envelope) of a single-frequency sinusoid contained within an input signal. The phase angles of

SDFT output samples enable the estimation of the instantaneous phase, or time delay, of an input sinusoid. (An N-point fast Fourier transform (FFT) is not appropriate for such applications because it computes N spectral samples where most of those samples are discarded, as being unwanted.) The SDFT, whose DFT size N is not restricted to be an integer power of two, is real-time in the sense that a new complex output spectral sample is computed for each new input signal sample.

To compute SDFTs we modify the Figure 1(a) network by setting $z_o = e^{j2\pi k/N}$, where frequency variable k is an integer in the set $k \in \{1,2,...,N-1\}$, to produce the traditional integer k SDFT network shown in Figure 2(a). (When k is an integer the Figure 1(a) comb's coefficient $z_o^{-N} = e^{-j2\pi k}$ is equal to unity.) Sequence $X_k(n)$ is an N-point DFT's kth-bin spectral output sequence computed on a sample-by-sample basis.

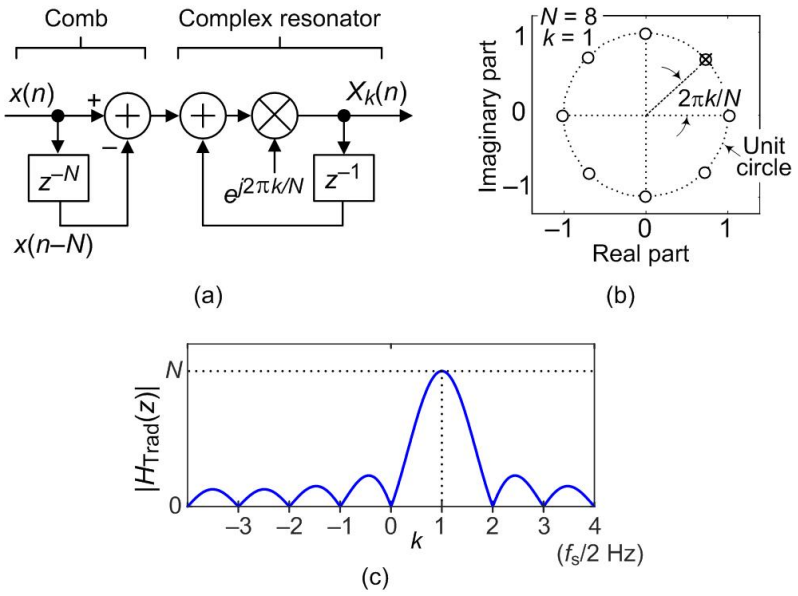

FIGURE 2. Traditional marginally stable, N-sample, integer k SDFT: (a) network structure; (b) z-plane pole/zero locations for $N = 8$ and $k = 1$; (c) frequency magnitude response.

Figures 2(b) and 2(c) show the traditional SDFT network's z-plane pole/zero locations and frequency magnitude response when, for example, $k = 1$ and $N = 8$. The network's single pole is located on the unit circle at an angle of $2\pi k/N$ radians (corresponding to a cyclic frequency of kf_s/N Hz) as shown in Figure 2(b). The frequency magnitude response in Figure 2(c), where the $k = 0$ frequency (DC) is located in the center of the frequency axis, is identical to the frequency magnitude response of the $k = 1$ bin of a conventional 8-point DFT.

The Figure 2 traditional SDFT network's difference equation is:

$$X_k(n) = e^{j2\pi k/N}[X_k(n-1) + x(n) - x(n-N)] \quad (3)$$

and its *z*-domain transfer function is:

$$H_{\text{Trad}}(z) = \frac{X_k(z)}{X(z)} = \frac{(1-z^{-N})e^{j2\pi k/N}}{1-e^{j2\pi k/N}z^{-1}}. \quad (4)$$

The details of the traditional SDFT's properties and behavior are found in [1,2].

The Figure 2 network is called “marginally stable” due to its pole being located on the unit circle only when the network's complex coefficient is numerically infinitely precise. When that coefficient is quantized by rounding and represented by a finite number of binary bits the pole will migrate slightly inside or outside the unit circle and may cause the network to become unstable. Our proposed SDFT network, described in the next section, avoids that potential instability problem.

Proposed Guaranteed Stable Integer *k* SDFT

We develop our proposed guaranteed stable integer *k* SDFT by converting the Figure 2(a) traditional SDFT's complex resonator into a 2nd-order real-valued resonator. To do so we multiply the transfer function numerator and denominator in (4) by $(1-e^{-j2\pi k/N}z^{-1})$ as:

$$H_{\text{Prop}}(z) = \frac{(1-z^{-N})e^{j2\pi k/N}}{1-e^{j2\pi k/N}z^{-1}} \cdot \frac{1-e^{-j2\pi k/N}z^{-1}}{1-e^{-j2\pi k/N}z^{-1}}$$

$$= \frac{e^{j2\pi k/N} - z^{-1} - e^{j2\pi k/N}z^{-N} + z^{-N-1}}{1-2\cos(2\pi k/N)z^{-1}+z^{-2}}. \quad (5)$$

The network corresponding to transfer function (5) is shown in Figure 3(a).

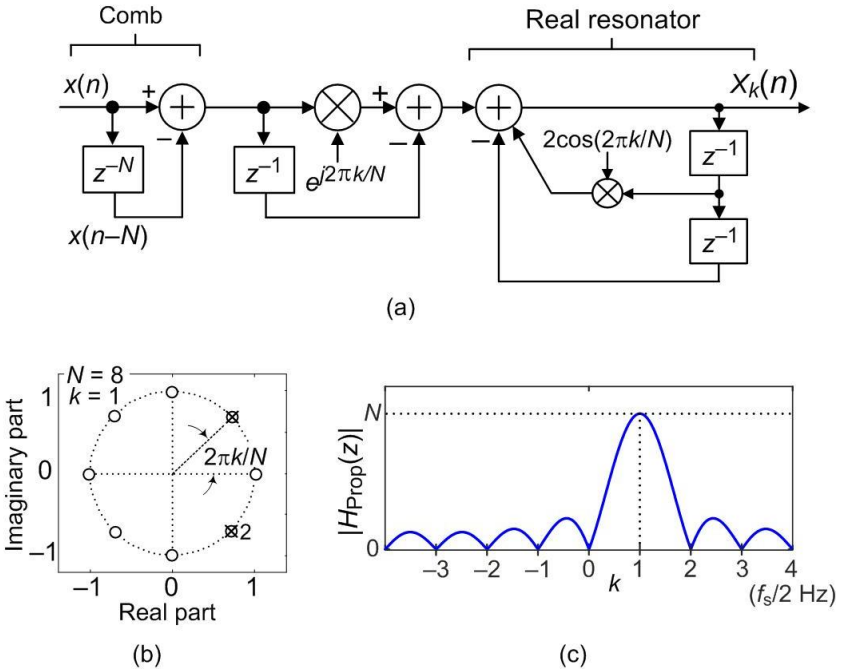

FIGURE 3. Proposed guaranteed stable, *N*-sample, integer *k* SDFT: (a) network structure; (b) *z*-plane pole/zero locations for *N* = 8 and *k* = 1; (c) frequency magnitude response.

In Figure 3 we see the proposed network's real resonator has conjugate poles lying on the *z*-plane's unit circle as shown in Figure 3(b) where the number 2 near the lower pole indicates dual zeros at $z = e^{-j2\pi k/N}$. The second of those dual zeros, produced by the feedforward center stage of the network, cancels the pole at $z = e^{-j2\pi k/N}$. Identical to that of the Figure 2 network, the Figure 3 network's time-domain impulse response is an *N*-sample complex exponential containing exactly *k* cycles. Thus the two networks have identical frequency responses.

Quantization of the center term coefficient in the denominator in (5) will cause the real resonator's poles to shift, as will be discussed later, slightly from their desired positions along the unit circle. However, what is not widely known is that regardless of coefficient quantization the poles will *always* lie exactly on the unit circle guaranteeing resonator numerical stability. (A proof of this important fact is given in Appendix B.) So the proposed Figure 3 network produces the same spectral output samples as a single bin of an *N*-point DFT, and the venerable Goertzel algorithm [4], with the additional advantage that its output spectral samples are updated for each new input signal sample.

Comparison With Previously Published SDFT Algorithms

For completeness, we now compare our proposed Figure 3 SDFT network to several previously published sliding spectrum analysis networks.

Due to their use of complex resonators such as the resonator shown in Figure 2, to avoid potential instability problems [1-3] suggest replacing their complex resonators' feedforward coefficient with $re^{j2\pi k/N}$, where scalar r is close to but slightly smaller than one. This r damping factor forces their networks' single SDFT pole to be located just inside the unit circle. While ensuring numerical stability, unfortunately the r damping factor introduces a small error in the SDFT output that accumulates as each new $X_k(n)$ output sample is computed. In addition, the SDFT in [3] requires phase correction making it much less computationally efficient than our proposed SDFT network in Figure 3.

Several sliding spectrum analysis algorithms have been proposed that compute a single-bin DFT output sample for every N input samples [5-7]. Unlike our proposed Figure 3 network, those algorithms are not strictly real-time because they do not compute one output spectral sample for each input signal sample.

A sliding Goertzel algorithm presented in Figure 2(a) in [8] uses a real resonator and is guaranteed stable. However, it computes real-time integer k single bin DFT output samples having correct magnitudes but incorrect phases. Adding a phase correction stage to that algorithm makes it less computationally efficient than our proposed SDFT network in Figure 3. (Correct phase measurement is mandatory in applications that estimate the time delay between two signals [9].)

A modulated SDFT (mSDFT) network was presented in Figure 3 of [10]. That network frequency translates the N-delay comb stage's output spectral energy, originally centered at kf_s/N Hz, down to zero Hz and implements a complex resonator where exact pole/zero cancellation occurs at $z = 1$ on the z-plane. While that network is guaranteed stable, it is less computationally efficient than is our proposed SDFT network in Figure 3.

A previously published sliding spectrum analysis network cleverly implements a bank of N paths where each path is equivalent to the complex resonator in Figure 2 tuned to a frequency of $2\pi k/N$ radians/sample [11,12]. The value for k in each path are one of the integers in the set $k \in \{1,2,...,N-1\}$. This network, called an “observer-based recursive SDFT” (oSDFT), is advantageous in that it

requires no delay-line comb stage and can be made guaranteed stable. The oSDFT computes real-time spectral outputs for all N bins of an N-point DFT and is best used when we need to simultaneously compute the outputs of multiple DFT bins. Because its paths cannot be pruned the oSDFT network is not computationally efficient in our desire to compute a single-bin DFT output. In the next section we quantify various guaranteed stable SDFT networks' computational requirements.

Proposed Integer *k* SDFT Computational Efficiency

For computing real-time single kth bin samples of an N-point DFT our proposed Figure 3 guaranteed stable integer k SDFT network is computationally more efficient compared to similar real-time, guaranteed stable, integer k SDFT networks discussed in this article. That comparison, for real-valued input signals, is presented in Table 1.

Table 1. Guaranteed stable *N*-point integer *k* SDFT computational complexity per real input sample

SDFT Network:	Real Multiplies:	Real Additions:	Real Divides:
Reference [8] magnitude only, no phase correction	5	4	0
Reference [8] with phase correction	9	6	0
Reference [8]	10	7	0
Reference [11]	$4N$	$6N$-1	2
Proposed Figure 3	4	6	0

A Windowed Integer *k* SDFT

It's worth noting that to reduce the inherent spectral leakage (high spectral sidelobe levels) of the N-point DFT outputs of our Figure 3 SDFT, we can perform the equivalent of time-domain windowing of an input signal by way of frequency-domain convolution using three individual SDFT networks. That process is described in [1].

Sliding Spectrum Analysis for a Non-integer *k* Analysis Frequency

A limitation of the Figure 3 SDFT network is that k, the N-point DFT frequency bin index, must be an integer. However, often there is interest in measuring the spectral value of an input signal of interest whose frequency does not equal a DFT analysis frequency determined by an integer value of k. For example, rotating machinery can experience harmful gearbox or bearing vibrations or internal combustion engines can generate objectionable audio exhaust noise at variable frequencies. It is often desirable to evaluate the spectral magnitude at the exact vibration frequency, which often does

not correspond to an integer value of *k*. As such we now propose networks that compute a single spectral sample of the discrete-time Fourier transform (DTFT) of an *N*-sample input signal. The frequency of the computed spectral sample is a non-integer value of frequency variable *k*.

To perform sliding spectrum analysis for an analysis frequency not restricted to be an integer multiple of f_s/N Hz, we modify the Figure 1(a) network by setting $z_o = e^{j2\pi k/N}$, where scalar *k* is any value in the range $0 \le k < N$, to produce the network shown in Figure 4(a).

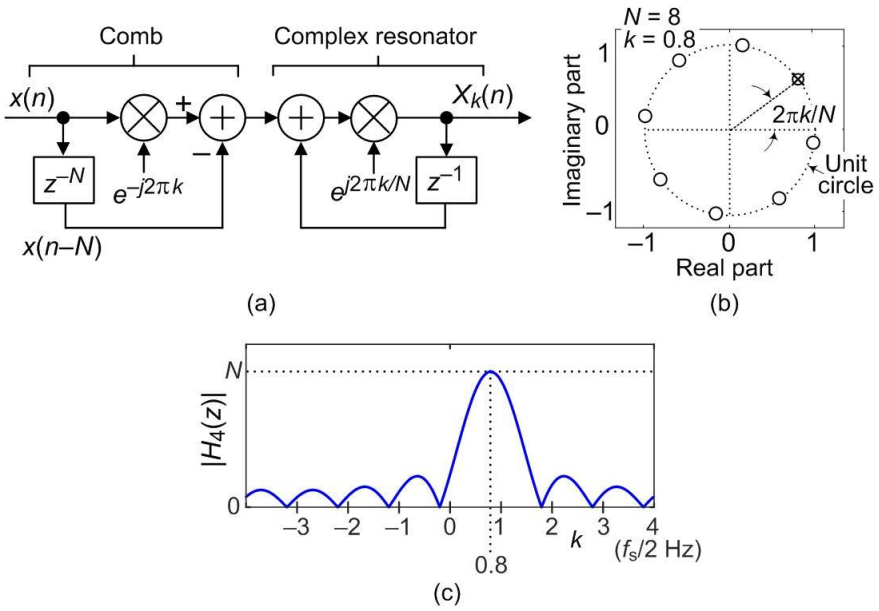

FIGURE 4. Marginally stable, *N*-sample, non-integer *k* sliding spectrum analysis: (a) network structure; (b) *z*-plane pole/zero locations for $N = 8$ and $k = 0.8$; (c) frequency magnitude response.

The complex coefficient in the network's comb stage provides the desired rotation of the angles of the comb's *N* *z*-plane zeros. The *z*-domain transfer function of the Figure 4(a) network is:

$$H_4(z) = \frac{X_k(z)}{X(z)} = \frac{\left(e^{-j2\pi k} - z^{-N}\right)e^{j2\pi k/N}}{1 - e^{j2\pi k/N}z^{-1}}. \qquad (6)$$

The Figure 4(a) non-integer *k* sliding spectrum analysis network, with its single pole on the unit circle, has the same undesirable marginal stability characteristics as the integer *k* SDFT in Figure 2.

We develop our proposed guaranteed stable non-integer k sliding spectrum analyzer by converting the Figure 4(a) complex resonator into a 2nd-order real-valued resonator. We do that by multiplying the transfer function numerator and denominator in (6) by $(1-e^{-j2\pi k/N}z^{-1})$ as:

$$H_5(z) = \frac{(e^{-j2\pi k} - z^{-N})e^{j2\pi k/N}}{1-e^{j2\pi k/N}z^{-1}} \cdot \frac{1-e^{-j2\pi k/N}z^{-1}}{1-e^{-j2\pi k/N}z^{-1}}$$

$$= \frac{e^{-j2\pi k(N-1)/N} - e^{-j2\pi k}z^{-1} - e^{j2\pi k/N}z^{-N} + z^{-N-1}}{1-2\cos(2\pi k/N) + z^{-2}} . \qquad (7)$$

The network corresponding to transfer function $H_5(z)$ is shown in Figure 5(a).

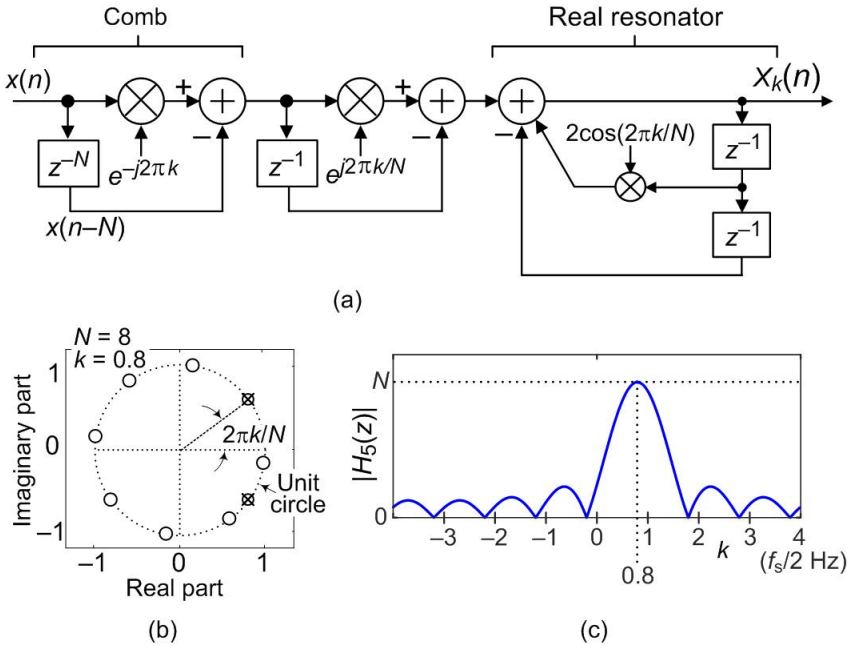

FIGURE 5. Proposed guaranteed stable, N-sample, non-integer k sliding spectrum analysis: (a) network structure; (b) z-plane pole/zero locations for $N = 8$ and $k = 0.8$; (c) frequency magnitude response.

The proposed Figure 5 real-time non-integer k SDFT requires 8 real multiplies and 9 real additions per input sample. It's worth noting that when k is an integer in Figure 5(a) that network then becomes the integer k SDFT network in Figure 3(a).

Again for completeness, we mention a previously published Goertzel-like algorithm that computes non-integer k spectral samples [13]. That algorithm computes a single non-integer k output spectral sample for each block of N input samples. As such, unlike our proposed Figure 5(a) network, it does not have the desired real-time behavior because it does not compute one output spectral sample for each input signal sample.

Effects of Quantized Coefficients

As mentioned earlier, quantization of the $2\cos(2\pi k/N)$ feedback coefficient in the proposed Figure 3 and Figure 5 real resonators will cause the resonators' poles to shift slightly along the unit circle from their desired positions on the z-plane's unit circle. The amount of undesired angular shift of a quantized pole is predictable, and is inversely proportional to the number of binary bits used to represent the feedback coefficients.

Let's assume a real resonator's feedback coefficient, $C = 2\cos(2\pi k/N)$, corresponds to a desired resonator center frequency of f_{des} Hz and places a positive-frequency pole on the z-plane's unit circle at an angle of α_{des} radians. If we quantize C, by way of rounding C to a fixed number of binary bits, to produce a quantized feedback coefficient of C_q, the resonator's actual positive-frequency pole location will be the radian angle α_{act} defined by:

$$\alpha_{act} = \cos^{-1}(-C_q/2). \quad (8)$$

Using (8) we state that a quantized-coefficient real resonator's actual resonant frequency, f_{act}, is defined by:

$$f_{act} = \frac{f_s \cdot \alpha_{act}}{2\pi} = \frac{f_s \cdot \cos^{-1}(-C_q/2)}{2\pi} \text{ Hz} \quad (9)$$

where f_s is the input signal sample rate in Hz.

Estimating the Numerical Accuracy of an Integer *k* SDFT

The long-term numerical accuracy of any given SDFT network depends on the network's structure, the network's resonant frequency, the amount of coefficient quantization, and the length of the network's input signal sequence. Thus to fully characterize the numerical accuracy of an SDFT network is a daunting task.

However, it is straightforward to estimate the numerical accuracy of a single specific integer k SDFT network design. We perform that estimation by applying a unity-amplitude sine wave, having hundreds of thousands or millions of samples, to a previously designed N-point integer k SDFT network having a value of k corresponding to the input signal's frequency. For an ideal SDFT network, after the initial transient N output samples all the remaining magnitude output samples will be unity-valued and the unwrapped output phase samples will increase linearly. For an actual SDFT

network we monitor the network's final magnitude output samples to measure how much their magnitudes deviate from a value of one and examine how the unwrapped output phase samples deviate from linearity.

Integer *k* and Non-integer *k* SDFT Signal Tracking Examples

Our first signal amplitude tracking example demonstrates the signal tracking capability of the proposed integer *k* SDFT network given in Figure 3. Figure 6(a) shows a noise-free 200-sample $x(n)$ time-domain input sine wave, having one cycle over eight samples (a frequency of $f_s/8$ Hz), whose peak-peak amplitude fluctuates over time. When we set the DFT size to $N = 8$ and set the DFT analysis frequency to $k = 1$ ($1 \cdot f_s/8$ Hz) the input sinusoid's frequency is then centered at the DFT's $k = 1$ bin. We note that the value of k is equal to the number of input sinusoidal cycles used in the SDFT processing.

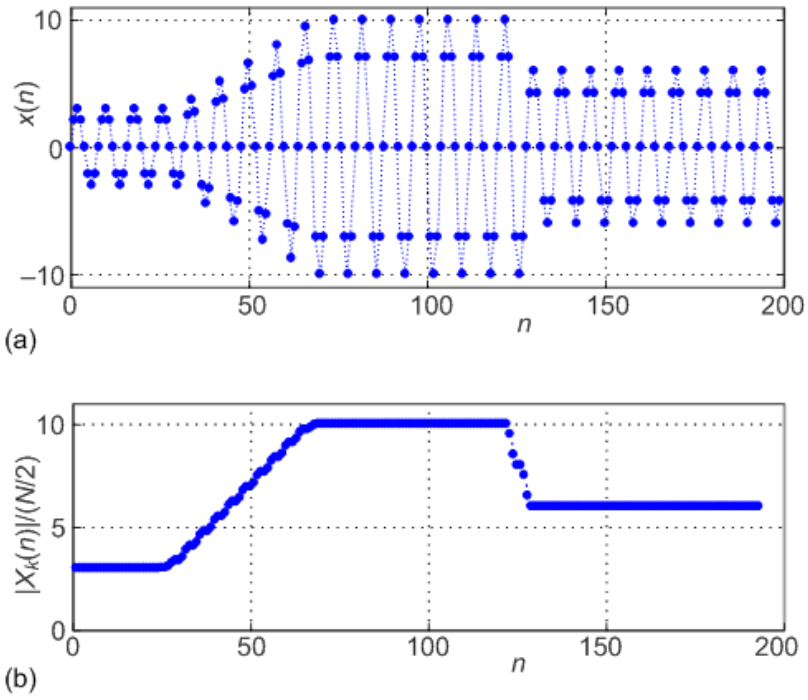

FIGURE 6. Integer *k* SDFT network signal tracking example: (a) input sine wave; (b) Figure 3 network magnitude output samples for $N = 8$ and $k = 1$.

Figure 6(b) shows how the proposed integer *k* SDFT's $|X_k(n)|$ output magnitude samples accurately track the input sinusoid's fluctuating peak amplitude. Those samples were divided by $N/2$ in Figure 6(b) for easy comparison with Figure 6(a).

Our second example illustrates the signal tracking capability of the proposed non-integer *k* SDFT network given in Figure 5. Figure 7(a) shows a noise-free 200-sample $x(n)$ time-domain input sine

wave, having one cycle over ten samples (a frequency of $0.8f_s/8$ Hz), whose peak-peak amplitude fluctuates over time. When we set the DFT size to $N = 8$ and set the DFT analysis frequency to $k = 0.8$ ($0.8f_s/8$ Hz) the input sinusoid's frequency is then centered at the DFT's scalar frequency of $k = 0.8$.

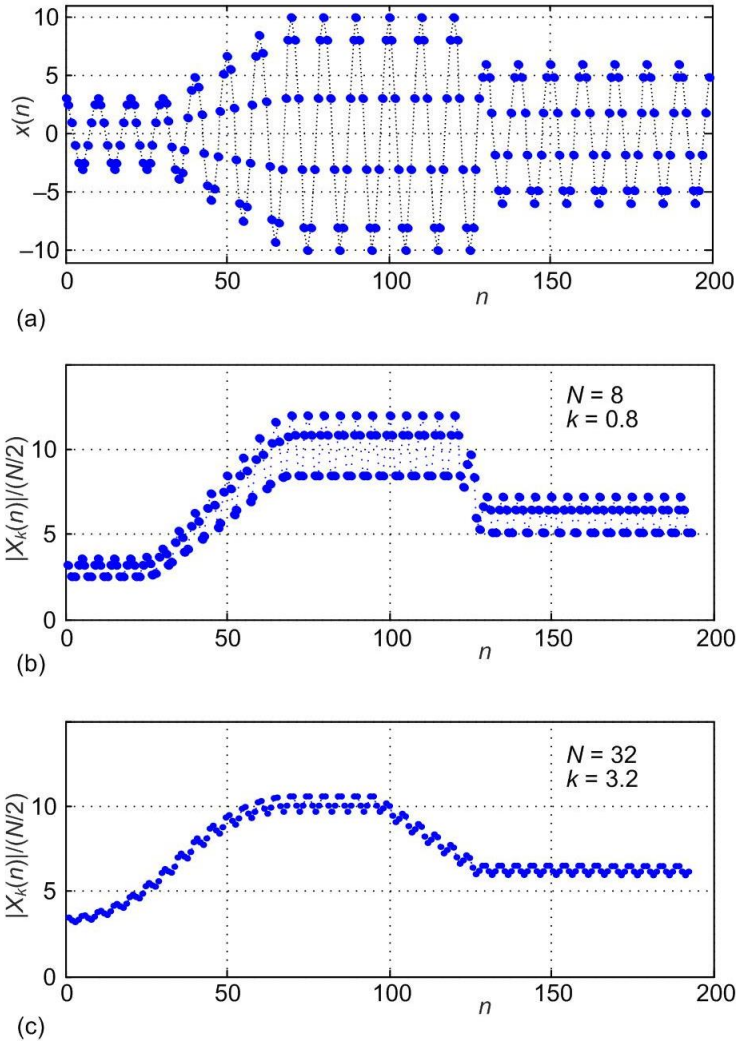

FIGURE 7. Non-integer k SDFT network signal tracking example: (a) input sine wave; (b) Figure 5 network magnitude output samples for N = 8 and k = 0.8; (c) Figure 5 network output samples for N = 32 and k = 3.2.

Figure 7(b) shows how the proposed non-integer k SDFT's $|X_k(n)|$ output magnitude samples track the input sinusoid's fluctuating peak amplitude. Those samples were divided by $N/2$ in Figure 7(b) for easy comparison with Figure 7(a).

Figure 7(b) also shows the output magnitude fluctuations inherent in all real-time DFT (including traditional non-sliding DFT) magnitude tracking when the frequency of a single-frequency real-valued target input sinusoid is not located at a DFT's bin center. In this signal tracking example those magnitude fluctuations are produced because a real-valued input sine wave's negative-frequency

spectral component, located at $k = –0.8$ in Figure 8(a), does not reside at a sinc magnitude response zero (null) as it did with the integer $k = 1$ scenario in Figure 3(c). And the insufficiently-attenuated $k = –0.8$ spectral component contaminates the output of the Figure 5 SDFT network.

Those unwanted non-integer k SDFT signal magnitude tracking fluctuations will also occur when the input signal is periodic containing a fundamental tone and its harmonics. That is because the input harmonics will not reside at a sinc magnitude response zero as they would have with the integer $k = 1$ scenario in Figure 3(c).

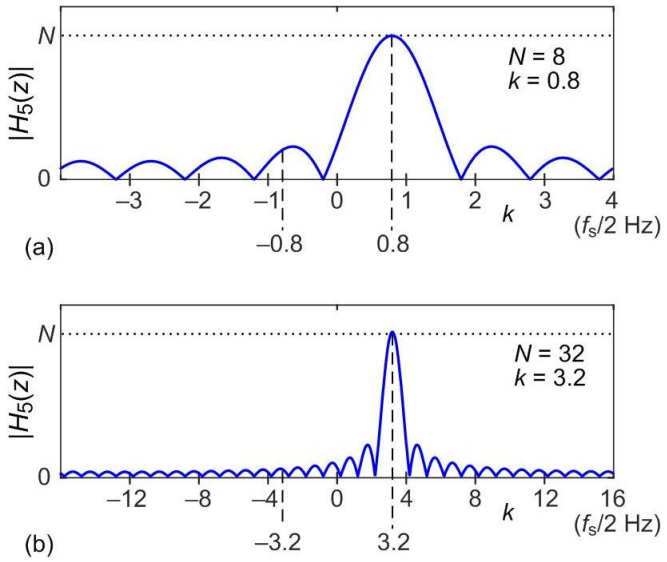

FIGURE 8. Proposed non-integer k SDFT network frequency magnitude response and real-valued target input sine wave spectral component locations at ±k as shown by the vertical dashed lines: (a) for N = 8 and k = 0.8; (b) for N = 32 and k = 3.2.

Keeping in mind that the value k is equal to the number of input sine wave cycles used in our SDFT processing window, we can increase the number of cycles used and reduce the undesirable output magnitude fluctuations by increasing k and N by a common factor while keeping the ratio k/N unchanged. Increasing k and N by a factor of four compresses the Figure 8(a) sinc magnitude response and results in improved attenuation of the unwanted negative-frequency $k = –3.2$ spectral component as shown in Figure 8(b). Setting $N = 32$ and $k = 3.2$ produces a non-integer k SDFT output having reduced magnitude fluctuations as shown in Figure 7(c).

It is worth mentioning that if we can set N freely, it is advisable to choose N so that the corresponding k number—and thus the periods of the signal from which the DFT is computed—is as close to an integer as possible. This will minimize the distance between the zeros of the sinc

frequency response and attenuate unwanted harmonics of the signal, including its negative-frequency components.

Other than increasing *k* and *N*, at the cost of additional computations alternate processing techniques can be effective in reducing the unwanted non-integer *k* SDFT output magnitude fluctuations. Those techniques are: 1) performing simple unity-gain lowpass filtering of the Figure 7(b) output magnitude samples; 2) implementing time-domain windowing of the input signal by means of frequency-domain convolution [1]; 3) if a real-valued input signal is a single tone we can convert that input signal to a positive-frequency only analytic signal, prior to SDFT processing, using a Hilbert transform network [14] (note that unlike the other listed options, this Hilbert option will not reduce the fluctuations caused by additional input frequency components); and 4) using the oSDFT networks in [11,12] will eliminate unwanted signal magnitude tracking fluctuations because their sinc frequency functions stretch rather than shift to provide complete attenuation of a real-valued input signal's negative-frequency spectral components and a periodic input signal's harmonics.

Our third signal amplitude tracking example involves monitoring the exhaust sound pressure of an 8-cylinder diesel engine where Figure 9(a) shows the output of a microphone located near the engine's exhaust pipe. Our goal is to apply an adaptive quarter wave-length tube to the exhaust pipe to reduce the peak-peak amplitude of that exhaust sound pressure signal.

An adaptive quarter wavelength tube is a duct of variable length connected to the engine's main exhaust duct by a T-junction. The length of the tube is manually adjusted until it corresponds to one-quarter of the acoustic wavelength of interest, which causes the amplitude of the exhaust sound pressure to be reduced.

Because each engine cylinder fires once every two crankshaft revolutions in a four-stroke engine, when the crankshaft speed is 1600 rpm (1600/60 revolutions/second) the main cylinder firing frequency is (1600/60)*(8/2) = 106.67 Hz. As such, the primary spectral component of the microphone signal in Figure 9(a) is a 106.67 Hz sinusoid. After digitizing the microphone signal at a sample rate of 1600 Hz, we use an SDFT to monitor the 106.67 Hz signal's amplitude as we adjust the length of the quarter-wavelength tube to minimize the exhaust sound pressure at 106.67 Hz.

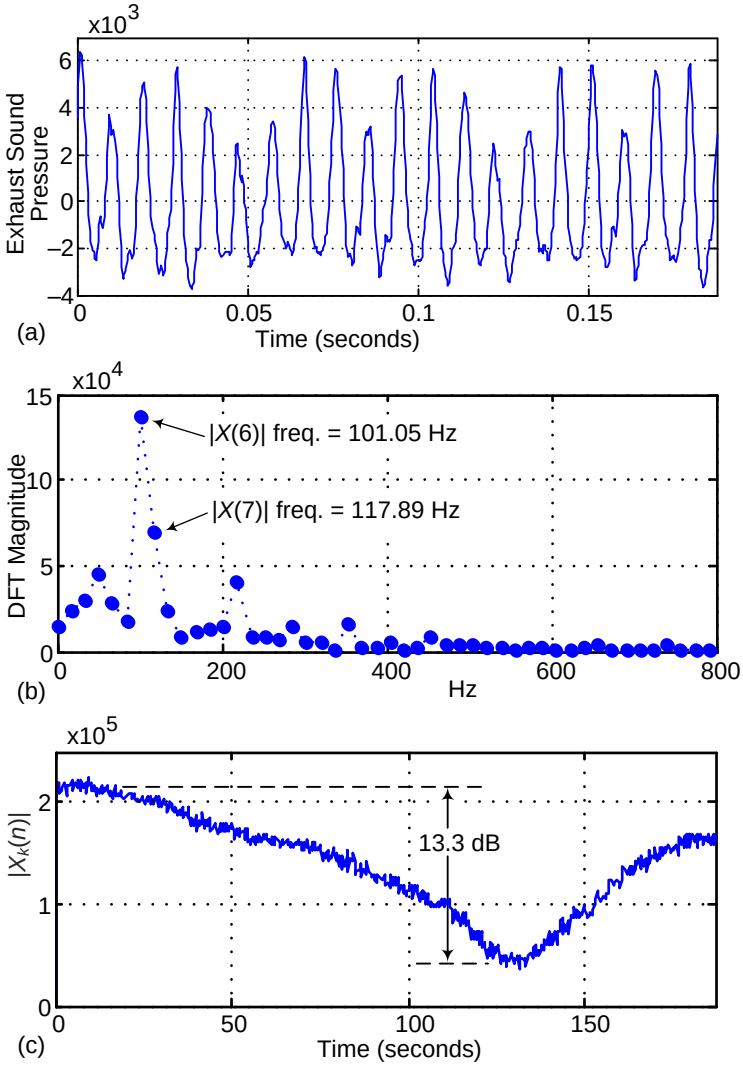

FIGURE 9. Diesel engine exhaust signal tracking example: (a) microphone output signal; (b) 95-point DFT spectral magnitude of the digitized microphone signal; (c) N = 95 and non-integer k = 6.33 SDFT output magnitude samples versus time.

Let's assume that for system design reasons we're restricted to set N = 95. With N = 95, Figure 9(b) shows the N-point DFT spectral magnitude of the microphone signal." In that figure we see that our high-level 106.67 Hz exhaust signal lies between the DFT bins of k = 6 (101.05 Hz) and k = 7 (117.89 Hz). The DFT's non-integer frequency variable k for the 106.67 Hz tone is

$$k = \frac{106.67N}{f_s} = \frac{106.67 \cdot 95}{1600} = 6.33. \qquad (10)$$

Hence, we are compelled to use the Figure 5 non-integer k SDFT network, with k = 6.33 and N = 95, to track the 106.67 Hz exhaust signal.

Figure 9(c) shows the magnitude of the exhaust sound pressure at 106.67 Hz when the length of the adjustable quarter-wavelength tube starts at its shortest length at time = 0 seconds, to simulate an exhaust system with no quarter-wavelength tube installed, and is gradually extended in length. The figure shows that at time 130 seconds the quarter-wavelength tube reduced the exhaust sound pressure by roughly 13.3 dB—which is a good result.

Although the non-integer *k* SDFT in this third example was not implemented in an automatic control system, that algorithm would be beneficial in such a feedback system where the engine exhaust signal is fed back through a non-integer *k* SDFT network whose output controls a mechanical actuator that adjusts the quarter wave tube length in real time as the engine speed varies. Having a non-integer *k* SDFT in the feedback loop would be far more computationally efficient than using an *N*-point FFT algorithm in the feedback loop.

Conclusions

We described the general notion of sliding spectrum analysis and showed the development of the currently most popular marginally stable integer *k* sliding DFT algorithm presented in Figure 2. Next we proposed the computationally efficient integer *k* sliding DFT network shown in Figure 3. That network computes a standard *N*-point DFT's *k*th bin output on a real-time sample for sample basis. Next we proved the proposed Figure 3 network to be guaranteed stable for all coefficient quantization scenarios, and compared the network's characteristics to previously published sliding DFT algorithms. Next we proposed the real-time, computationally efficient, guaranteed stable, non-integer *k* sliding discrete-time Fourier transform (DTFT) network shown in Figure 5 whose frequency parameter *k* can be any real number in the range $0 \le k < N$. Finally we provided three real-time signal tracking examples using our proposed integer *k* and non-integer *k* SDFT algorithms.

Acknowledgments

We thank Pavel Rajmic for the proof given in Appendix B; as well as Editors Balazs Bank and Roberto Togneri, and the anonymous reviewers for their very helpful comments.

Authors

Richard Lyons (R.Lyons@ieee.org) is a consulting signal processing engineer and lawn sprinkler system repairman. Winner of the IEEE 2012 Education Award, he is the author of the textbook *Understanding Digital Signal Processing 3/E* (Prentice-Hall, 2010). Lyons is the Editor of, and contributor to, the book *Streamlining Digital Signal Processing, A Tricks of the Trade Guidebook* (IEEE Press/Wiley, 2007), and co-author of *The Essential Guide to Digital Signal Processing* book (Prentice-Hall, 2014).

Carl Howard (carl.howard@adelaide.edu.au) is a professor at The University of Adelaide, School of Mechanical Engineering, undertaking teaching and research in acoustics and vibrations. He is a fellow of the Australian Acoustical Society, and co-author of the textbooks *Engineering Noise Control, 5th ed* (CRC Press, 2017), and *Acoustic Analyses Using Matlab® and Ansys®* (CRC Press 2017).

References

[1] E. Jacobsen and R. Lyons, "The sliding DFT," *IEEE Signal Processing Mag.*, vol. 20, no. 2, pp. 74–80, Mar. 2003.

[2] E. Jacobsen and R. Lyons, "An Update to the Sliding DFT," *IEEE Signal Processing Mag*., vol. 21, no. 1, pp. 110–111, Jan. 2004.

[3] S. Douglas and J. Soh, "A Numerically Stable Sliding-Window Estimator and Its Application to Adaptive Filters," *Proc. 31st Annu. Asilomar Conf. Signals, Systems, and Computers*, Pacific Grove, CA, vol. 1, pp. 111–115, Nov. 1997.

[4] G. Goertzel, "An Algorithm for the Evaluation of Finite Trigonometric Series," *American Mathematical Monthly*, vol. 65, no. 1, pp. 34–35, Jan. 1958.

[5] J. Chicharo and M. Kilani, "A Sliding Goertzel Algorithm," *Signal Processing*, no. 52, pp. 283-297, March, 1996.

[6] D. Gufovskiy and L. Chu, "An Accurate and Stable Sliding DFT Computed by a Modified CIC Filter ," *IEEE Signal Processing Mag.*, vol. 34, no. 1, pp. 89–93, Jan. 2017.

[7] D. Romero and M. Jimenez, "Simplifying Single-Bin Discrete Fourier Transform Computations," *IEEE Signal Processing Mag*., vol. 38, no. 2, pp. 130–136, Mar. 2021.

[8] C. Orallo and I. Carugati, "Single Bin Sliding Discrete Fourier Transform," Chapter 2 of *Fourier Transforms: High-tech Application and Current Trends*, IntechOpen Limited, London, pp. 25-42, 2017.

[9] A. Harvie, et. al., "A High-Resolution Polarimeter Formed From Inexpensive Optical Parts," Science Reports, Vol. 10, Article No. 5448, 2020, Avail. online: https://doi.org/10.1038/s41598-020-61715-7

[10] K. Duda, "Accurate, Guaranteed Stable, Sliding Discrete Fourier Transform," *IEEE Signal Processing Mag*., vol. 27, no. 6, pp. 124–127, Nov. 2010.

[11] Z. Kollar, et. al., "Observer-Based Recursive Sliding Discrete Fourier Transform," *IEEE Signal Processing Mag.*, vol. 35, no. 6, pp. 100-106, Nov. 2018.

[12] L. Sujbert, et. al., "An Observer-Based Adaptive Fourier Analysis," *IEEE Signal Processing Mag.*, vol. 37, no. 4, pp. 134-143, July 2020.

[13] P. Sysel, and P. Rajmic, "Goertzel Algorithm Generalized to Non-integer Multiples of Fundamental Frequency," *EURASIP Journal on Advances in Signal Processing*, no. 56, pp. 1-8, 2012.

[14] R. Lyons, *Understanding Digital Signal Processing*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2011, pp. 479-498.

Appendix A: Derivation of (1)

The process of sliding spectrum analysis is based on successive *z*-transforms of *N* consecutive samples of an $x(n)$ input sequence evaluated at an arbitrary point z_o on the *z*-plane. For example, the first and second *N*-sample sliding *z*-transforms of an $x(n)$ sequence evaluated at $z = z_o$ can be defined as:

$$X_{z_o}(q) = x(n)z_o^{-0} + x(n+1)z_o^{-1} + x(n+2)z_o^{-2} + \ldots + x(n+N-1)z_o^{-N+1}, \text{ and} \qquad \text{(A-1)}$$

$$X_{z_o}(q+1) = x(n+1)z_o^{-0} + x(n+2)z_o^{-1} + x(n+3)z_o^{-2} + \ldots + x(n+N)z_o^{-N+1}. \qquad \text{(A-2)}$$

Multiplying both sides of (A-2) by z_o^{-1} and substituting (A-1) into the right side of the resulting product allows us to write:

$$X_{z_o}(q+1) = z_o[X_{z_o}(q) - x(n) + x(n+N)z_o^{-N}]. \qquad \text{(A-3)}$$

Finally, with no loss in generality, we can modify (A-3)'s time indexing so the $x(n)$ input samples and the $X_{zo}(q)$ output samples use the same time index *n*. That modification yields our desired recursive sliding *z*-transform of an *N*-sample $x(n)$ sequence evaluated at $z = z_o$ as:

$$X_{z_o}(n) = z_o[X_{z_o}(n-1) + x(n)z_o^{-N} - x(n-N)]. \qquad \text{(A-4)}$$

Appendix B: Proof of Real Resonator Guaranteed Stability

To show the 2nd-order real resonator stage in Figure 3(a), having conjugate poles residing on the *z*-plane's unit circle, to be guaranteed stable we proceed as follows. For mathematical convenience, we multiply the real resonator's transfer function numerator and denominator by z^2 as:

$$H_{\text{Res}}(z) = \frac{1}{1 - 2\cos(2\pi k/N)z^{-1} + z^{-2}} \cdot \frac{z^2}{z^2} = \frac{z^2}{z^2 - 2\cos(2\pi k/N)z + 1}. \qquad \text{(B-1)}$$

If we let $\cos(2\pi k/N) = p$ then (B-1)'s denominator becomes:

$$z^2 - 2pz + 1. \qquad \text{(B-2)}$$

The roots of the (B-2) polynomial, the *z*-plane locations of the resonator's poles, are:

$$z = \frac{2p \pm \sqrt{4p^2 - 4}}{2} = p \pm \sqrt{p^2 - 1} = p \pm \sqrt{-\left(1 - p^2\right)}. \qquad \text{(B-3)}$$

Because p is in the range $-1 \le p \le 1$, then $p^2 \le 1$ and the quantity under the square root sign in (B-3) is always zero or negative allowing us to rewrite (B-3) as:

$$z = p \pm j\sqrt{1 - p^2}\,. \qquad \text{(B-4)}$$

Thus the magnitudes of both roots of polynomial (B-2) are

$$|z| = \sqrt{p^2 + \left(1 - p^2\right)} = \sqrt{1} = 1 \qquad \text{(B-5)}$$

so the Figure 3 real resonator's conjugate pole magnitudes are *always* unity and the poles lie exactly on the z-plane's unit circle providing guaranteed stability.