Observer-Based Recursive Sliding Discrete Fourier Transform [Tips & Tricks]

Article *in* IEEE Signal Processing Magazine · November 2018

DOI: 10.1109/MSP.2018.2853196

3 authors, including:

Zsolt Kollar
Budapest University of Technology and Economics

Observer based recursive sliding discrete Fourier transform

Zsolt Kollár, *Member, IEEE,* Ferenc Plesznik and Simon Trumpf

I. INTRODUCTION

IN the field of digital signal analysis and processing, the ubiquitous domain transformation is the Discrete Fourier Transform (DFT), which converts the signal of interest within a limited time window from discrete time to the discrete frequency domain. The active use in real-time or quasi real-time applications has been made possible by a family of fast implementations of the DFT, called the Fast Fourier Transform (FFT) algorithms.

Although highly optimized and efficient FFT algorithms are available, their operation remains block oriented with non-recursive operations. An alternative approach to this technique is the Sliding DFT (SDFT) where the calculations are performed for a fixed-size sliding window.

The basic idea behind the SDFT algorithm is to recursively calculate the DFT spectrum of the input stream [1], [2]. It is based on a Lagrange structure, built up on a comb filter and complex resonators for the various frequency bins. The biggest disadvantage of this algorithm is that it suffers from stability problems caused by numerical imperfections. Various solutions have been proposed to counteract this effect, keeping the original functionality. The Modulated SDFT (mSDFT) [3] addresses the problem with a modified structure moving the complex multiplication factor out of the resonator. Another SDFT variant is the Hopping SDFT (hSDFT) [4] which is optimized for the calculation of the SDFT with larger steps (L) than a single sample but smaller than the observation window: $L = 2^a < N$.

In this paper, we investigate a less known alternative solution for the recursive calculation of the DFT which is based on the observer theory – the Observer based SDFT (oSDFT). It was originally developed by Hostetter [5], and generalized by Péceli [6]. Software implementation issues of the structure were recently presented in [7]. The structure is proved to be stable, with a small sensitivity to numerical imperfections. Throughout this paper we will compare it to the SDFT and mSDFT structures.

The paper is organized as follows. First the principles of the SDFT and its variant – the mSDFT – is briefly discussed. Following this, the structure of the oSDFT with two implementation alternatives is introduced. In Section IV the computational complexity of the investigated SDFT schemes is briefly analyzed. Section V presents the effects of quantization errors on the estimated spectrum for the presented SDFT schemes. Finally, in Section VI the conclusions are drawn.

Manuscript received May 4, 2018.

II. SLIDING DFT

A. SDFT

In this section the basic principle of the SDFT is presented according to [1], [2].

The formula for calculating the DFT coefficient in the k^{th} frequency position over the N samples block of $x[n]$ is given as

$$X_k = \sum_{n=0}^{N-1} x\left[n\right] W_N^{-kn}, \quad k = 0 \dots N-1, \tag{1}$$

where $W_N = e^{j\frac{2\pi}{N}}$ with j being the imaginary unit. The calculation of Eq. (1) in a sliding manner, the DFT component can be expressed as:

$$X_k^C\left[n+1\right] = \sum_{m=0}^{N-1} x\left[q+m\right] W_N^{-km}, \tag{2}$$

where $k = 0 \dots N-1$ and $q = n - N + 1$. Through this operation we obtain a rotating DFT coefficient, a complex DFT component $X_k^C[n]$, since $x[n]$ slides while W_N^{-km} stands still relative to the sampling window. The upper index C in $X_k^C[n]$ refers to the component nature of the DFT value in order to distinguish it from the DFT coefficient $X_k[n]$. Given a periodic signal with periodicity of N, the DFT component equals the DFT coefficient at every N^{th} step:

$$X_k^C[n] = X_k, \quad n = 0, N, 2N \dots . \tag{3}$$

The recursive equivalent of Eq. (2) can be expressed based on the previous DFT component $X_k^C[n]$, the current signal sample $x[n]$ and the former signal sample $x[n-N]$ as:

$$X_k^C[n+1] = W_N^k \Big(X_k^C[n] + \big(x[n] - x[n-N] \big) \Big). \tag{4}$$

Eq. (4) can be implemented as a comb filter followed by a resonator stage as shown in Fig. 1. The resonator stage is an

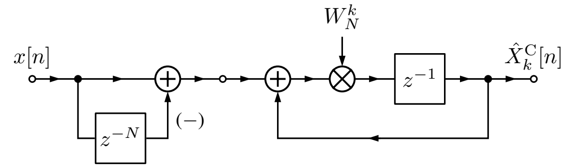

Fig. 1. A single SDFT branch for the calculation of the k^{th} frequency bin

integrator containing a complex multiplication factor, which is an Infinite Impulse Response (IIR) filter. The transfer function of the SDFT structure can be expressed as:

$$H_{\mathrm{k,SDFT}}(z) = \frac{X_k^C(z)}{X(z)} = \left(1 - z^{-N}\right) H_k(z), \tag{5}$$

where the transfer function of the resonator $H_k(z)$ can be determined as:

$$H_k(z) = \frac{W_N^k z^{-1}}{1 - W_N^k z^{-1}}. \tag{6}$$

This structure is considered to be only marginally stable in practice [1] as the W_N^k poles – in presence of numerical imperfections – may be located inside or outside the unit circle. To avoid a potential divergence in the results, without altering the structure, a straightforward method is given by the rSDFT [1] enforcing the poles inside the unit circle by applying a constant multiplication factor r – slightly smaller than 1 – to all W_N^k factors. As a drawback it leads to a modified DFT calculation thus it gives inaccurate results [3].

B. mSDFT

A slightly modified structure of the SDFT is the mSDFT [3], which aims to solve the aforementioned stability issue without sacrificing accuracy through utilizing the DFT's frequency shift theorem property. The mSDFT based structure calculating the k^{th} frequency bin is shown in Fig. 2.

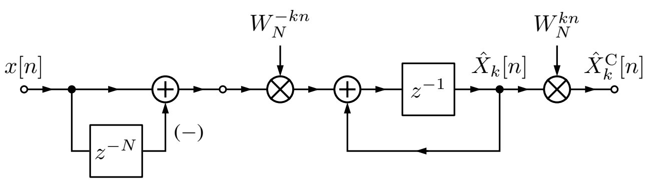

Fig. 2. A single mSDFT branch for the calculation of the k^{th} frequency bin

First, it transforms the k^{th} frequency bin to DC ($k = 0$) by a complex multiplication with the sequence W_N^{-kn}, than the calculations of Eq. (2) is applied for $k = 0$. Finally, it transforms the result back by up-conversion with a multiplication of the sequence W_N^{kn}. With this described technique the resonators became stable integrators performing simple averaging.

Via down-conversion the mSDFT calculates the DFT coefficients, recursively as:

$$\hat{X}_k[n+1] = \left(\hat{X}_k[n] + W_N^{-kn}\Big(x[n] - x[n-N]\Big)\right). \tag{7}$$

In order to get the same output as the SDFT in Eq. (2), namely the DFT component, an up-conversion sequence has to be applied by multiplying the DFT coefficient X_k with W_N^{kn},

$$\hat{X}_k^C[n] = \hat{X}_k[n] W_N^{kn}. \tag{8}$$

As a result, the transfer function of an mSDFT branch is theoretically identical with the transfer function of an SDFT branch presented in Eq. (5).

III. Observer based sliding DFT

In this section we introduce a less known alternative approach to the SDFT problem, the Observer based SDFT, the oSDFT. The main idea behind the oSDFT – the application of the state observer – is widely used in system control theory [8], and can be successfully adapted for digital signal processing purposes [5], [6] as well.

A. The observer theory model

The observer theory supposes the system model that the measured signal ($x[n]$) is a linear combination of the elements of a given basis system:

$$x[n] = \sum_{k=0}^{N-1} X_k c_k[n], \tag{9}$$

where N is the rank of the basis system, $c_k[n]$ is the k^{th} basis vector and X_k its matching weighting factor.

This system model is considered for the signal construction and can be seen as the generator of the signal $x[n]$ on the left side of Fig. 3, wherein weighting factors are stored in discrete integrators as initial values.

The observer (it can be seen on the right side of Fig. 3), by mirroring the system model's structure, estimates the $x[n]$ input signal's X_k weighting factors in its internal state variables $\hat{X}_k$ through signal decomposition. For the refinement of this estimation, a negative feedback is created with a reconstructed signal $y[n]$ from the estimated $\hat{X}_k$ weighting factors. This negative feedback also acts as a stabilizing control loop for our state observer [6], [9].

The k^{th} state variable can be expressed as:

$$\hat{X}_k[n+1] = \hat{X}_k[n] + g_k[n]\Big(x[n] - y[n]\Big), \tag{10}$$

where $y[n]$ can be expressed as:

$$y[n] = \frac{1}{N}\sum_{k=0}^{N-1} c_k[n]\hat{X}_k[n] = \frac{1}{N}\sum_{k=0}^{N-1} \hat{X}_k^C[n]. \tag{11}$$

Péceli proves the following 4 statements in [6], which are crucial from the SDFT aspect :

1) The observer is convergent, if $c_k[n]$ and $g_k[n]$ are basis-reciprocal basis systems for $n = 0 \ldots N-1$ with a normalization factor of $\frac{1}{N}$:

$$\frac{1}{N}\sum_{n=0}^{N-1} c_k[n] g_k[n] = 1, \forall k. \tag{12}$$

Moreover, in this scenario the system is dead-beat in N-step, i.e. after N steps $\hat{X}_k[n] = X_k$.

2) The state variables $\hat{X}_k$ of the observer are the DFT coefficients according to Eq. (9), if $g_k[n] = W_N^{-kn}$ and $c_k[n] = W_N^{kn}$. The modulated state variables $\hat{X}_k^C[n+1]$ are the sliding DFT components of the input signal $x[n]$ as presented in Eq. (2).

3) Based on the fact that the oSDFT structure is a control loop with a negative feedback, the transfer function of the k^{th} branch of the oSDFT can be expressed as:

$$H_{k,\mathrm{oSDFT}}(z) = \frac{X_k^C(z)}{X(z)} = \frac{H_k(z)}{1 + \frac{1}{N}\sum\limits_{k=0}^{N-1} H_k(z)}, \tag{13}$$

where $H_k(z)$ is given in Eq. (6).

4) The oSDFT structure is equivalent to the SDFT structure presented in Fig. 1 in such a way, that their transfer functions of the k^{th} branches are equal. The proof of the theoretical equivalence can be found in Appendix A.

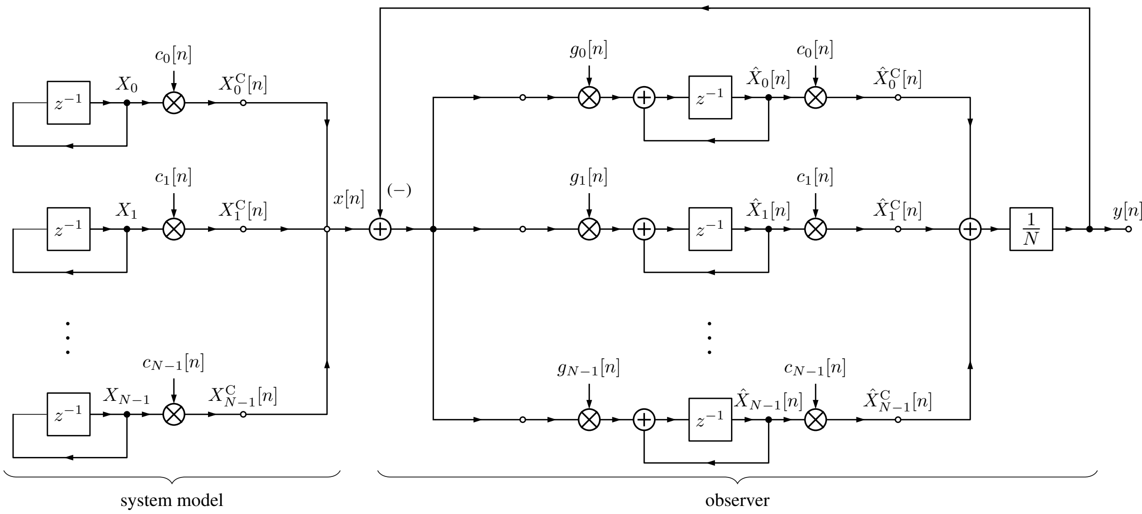

Fig. 3. The observer theory model: system model and observer

B. Resonator based oSDFT

An alternative version of the oSDFT structure is depicted in Fig. 4, which is based solely on resonators – which are IIR filters – without down- and up-converters, similarly to the SDFT structure formerly presented in Section II-A. The proof of the equivalence of the two oSDFT structures is provided in Appendix B.

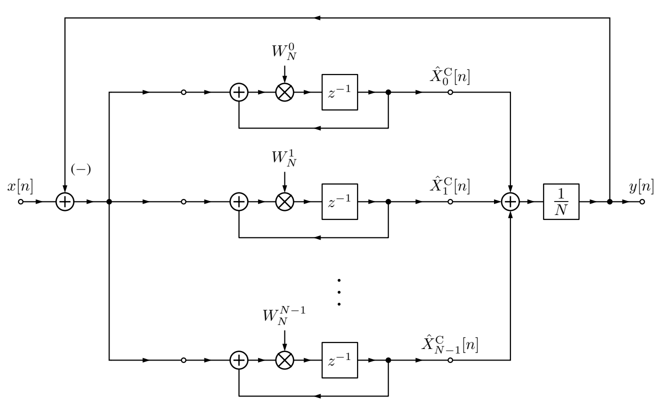

Fig. 4. Resonator based oSDFT

IV. Complexity analysis

In this section we analyze the computational complexity and memory requirements for the various SDFT structures when calculating all N DFT components. The comparison will be performed based on the calculation of a single input sample. All elements are considered to be complex valued. The requirements are summarized in Table I.

Independent from the chosen algorithms, N registers are required for storing the state variables of the resonators or the integrators. The SDFT and mSDFT algorithms use N additional registers for the comb filter's N-step delay line. Furthermore, the SDFT and the resonator based oSDFT structure require N memories to store the multiplication factors $\left(W_k^N\right)$ for all branches. The mSDFT and the oSDFT structures can obtain the values of the modulators and demodulators signal $\left(W_N^{nk}\right.$ and $\left.W_N^{-nk}\right)$ from a Look-Up-Table (LUT). The LUT stores N samples for each branch, as the values are periodic to N.

TABLE I
Complexity comparison of the various SDFT structures

	Memory			Multipliers	Adders	
Type	ROM	RAM	LUT_N		2-input	N-input
SDFT	N	$N+N$	0	N	$N+1$	0
mSDFT	0	$N+N$	1	$2N$	$N+1$	0
oSDFT	0	N	1	$2N$	$N+1$	1
oSDFT (res.)	N	N	0	N	$N+1$	1

Resonator based implementations (SDFT and resonator based oSDFT) require N multipliers, whereas the demodulation and modulation approaches (mSDFT, oSDFT) use $2N$ multipliers.

For all algorithms, each branch requires one 2-input adder. In case of the oSDFT structures, they both apply an N-input adder to calculate the feedback signal $y[n]$ and a 2-input adder is used to calculate the difference of the input and the feedback signal as shown in Eq. (10).

The biggest advantage of these structures compared to the FFT based block-wise calculation is that the operational load can be distributed between the incoming samples, as the SDFT structure can operate continuously. As soon as the N-th sample of a block has arrived, the calculation with the last input sample can be executed in a single step with parallel calculations and the spectral components will be available faster compared to the block-wise operational FFT where this can be performed in $\log_2 N$ steps.

V. Simulations

A. Floating point implementation

A simulation environment for the comparison of the aforementioned sliding DFT algorithms (mSDFT, oSDFT, resonator based oSDFT) was developed in MATLAB2017a (x64 PC). For the algorithms we applied 32 bit – single precision –

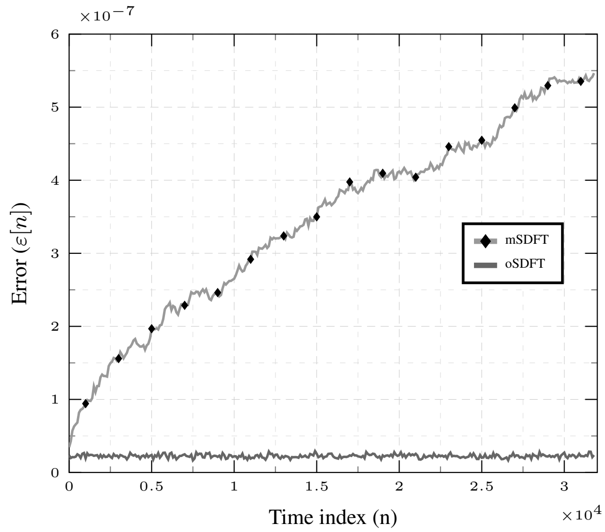

Fig. 5. Error progress of the mSDFT and the oSDFT algorithms

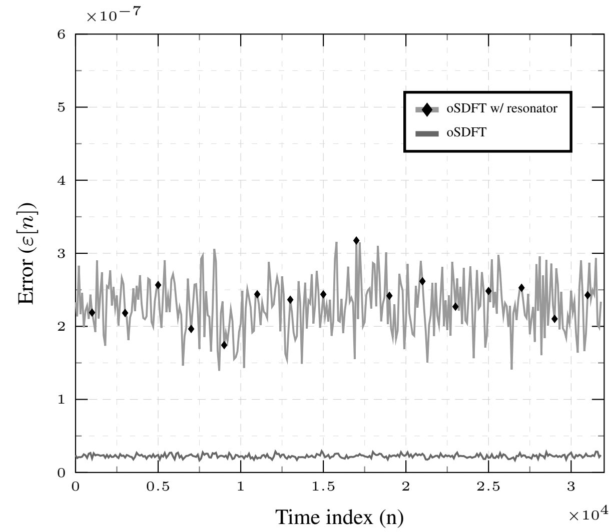

Fig. 6. Error progress of the oSDFT and the resonator based oSDFT algorithms

floating point arithmetic, and compared the numerical imperfections of the various methods to the results of a 64 bit – double precision – arithmetic sliding FFT, utilizing the built-in *fft* function. We applied the following simulation scenario: within an $N = 64$ frequency bin setup an aperiodic white gaussian noise was used, where the noise signal was generated using the built-in *randn* function operating with default seed option and a unit variance as:

```
rng('default'); % setting the seed
var = 1;        % variance of the noise signal
% noise signal with single precision
x = var * randn(1,32000,'single');
```

The usage of white noise as excitation signal ensures, that all the branches system-wide are statistically equally excited, so the behavior of each structure can be better characterized and evaluated as a dynamic system.

The results of the various SDFT methods were compared through double precision arithmetic to the results of the sliding FFT, and the average error signal over the branches was formulated as:

$$\varepsilon[n] = \frac{1}{N}\sum_{k=0}^{N-1}\left|\hat{X}^{C}_{k,\mathrm{xSDFT}}[n] - \hat{X}^{C}_{k,\mathrm{FFT}}[n]\right|, \tag{14}$$

where xSDFT stands for the mSDFT and oSDFT algorithms.

In Fig. 5 the error progress in function of the time index is compared in case of mSDFT and oSDFT. It can bee seen that the error of the mSDFT is not stable, it slowly drifts over the time samples. On the contrary, the oSDFT algorithm is stable, but it is noisy as well, due to the numerical errors.

In Fig. 6 the error progress in function of the time index is shown for the oSDFT and the resonator based oSDFT algorithms. It can be seen that both algorithms produce a stable but noisy error over the discrete time samples. It can be also observed that the oSDFT outperforms the resonator based oSDFT.

The observed performance difference between the two oSDFT structure has two attributes with a common root cause: multiplication within the resonators with constant W_N^k, at every time step. The two distinct differences experienced in Fig. 6 are an offset and a higher noise variance.

The offset is caused by the fact, that for every step of n the W_N^{kn} modulator and demodulator values for the oSDFT are taken periodically from a pre-computed sequence stored in a LUT, and within this LUT the error introduced by rounding (finite precision storage) is averaged out over a sequence period. This way the oSDFT's modulation and demodulation process will be more precise regarding frequency accuracy in average over a sequence period, than the resonator based oSDFT, where the numerical error in the constant W_N^k pole can't be averaged out over the same period, so thus leading to a constant frequency offset in the center frequency of the resonators.

The higher variance of the error signal comes from the fact, that the finite precision multiplication by W_N^k within the resonator's loop is an additional noise source which will dominate the variance due to the structure, thus it will lead to slightly misplaced W_N^k poles in a random manner over the complex plain.

B. Fixed point implementation

As to further investigate and cover wider use-case scenarios, the results for fixed point implementations are also presented. During the comparison simulations with the 32 bit single precision floating point variants the signed fixed point calculations were implemented with a word length of 32 bits, from which 31 bits were used for the fractional part, and a rounding towards zero method was applied to maintain stability of the feedback structures. Otherwise, the simulation environment, the test signals and the error term definition were the same as with the floating point scenario presented in Section V-A.

The error progress of the various SDFT structures for signed Q0.31 format fixed point implementation can be seen in Fig. 7. It can be observed that the results are similar to the case where the structures are implemented using single precision, although

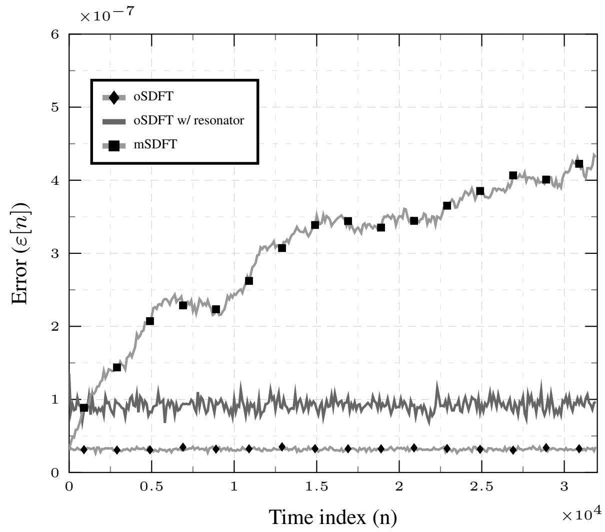

Fig. 7. Error progress of the oSDFT, the resonator based oSDFT and the mSDFT algorithms with fixed point implementation in signed Q0.31 format using 32 bit with 31 bit fractional part

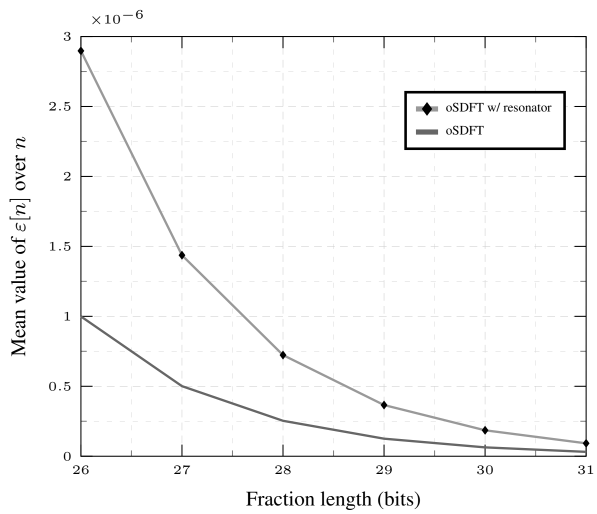

Fig. 8. Mean error of the oSDFT and the resonator based oSDFT in function of the fraction length of the signed 32 bit fixed point implementation

the overall errors are slightly smaller for each method. This can be explained by the fact that for the IEEE 754-2008 single precision standard – used by MATLAB – the fractional part is defined as only 23 bits, so thus within the same range it offers lower resolution, resulting in more imprecise W_N^{kn} modulator and W_N^k pole values.

Furthermore, the averaged error $\varepsilon[n]$ over the samples n for the fixed point implementation in function of the fraction part is shown in Fig. 8. It can be observed that for both methods with enlarged fraction part the error is exponentially decreasing.

VI. Summary

In this paper an alternative structure for the calculation of the SDFT, the oSDFT, was presented which is based on the observer theory – a method taken from control theory. The core structure of the oSDFT is similar to the other SDFT methods, but it applies a recursive overall feedback branch, which allows the elimination of the feed-forward comb filter and its N-tap delay line, achieving long-term stability in contrast to other well-known SDFT methods. The various SDFT structures were also compared based on their memory and arithmetical requirements.

It was also shown that the oSDFT structure has a lower sensitivity to numerical imperfections compared to other SDFT structures. The oSDFT is stable for input signals containing aperiodic white noise as well, due to the control loop feedback structure, and keeps its stability and behavior with fixed point implementations as well.

The application of the oSDFT structure can be especially advantageous if not only the long-term stability but also a large percentage of the N DFT components are required to be calculated in a sliding manner. From a practical aspect, the oSDFT structure can be advantageously used as a tunable filter [10] or as a nonlinear adaptive frequency estimator [11] as well.

Appendix A
Proof of the equivalence of the SDFT and oSDFT

In order to prove the equivalence of the SDFT and oSDFT structures, we will show that the transfer functions for each branch – $H_{\mathrm{k,SDFT}}(z)$ and $H_{\mathrm{k,oSDFT}}(z)$ – are equal. The transfer function of the SDFT and the oSDFT structures are expressed according to Eq. (5) and Eq. (13) as:

$$H_{k,\mathrm{SDFT}}(z) = \left(1 - z^{-N}\right) \cdot H_k(z), \tag{A-15}$$

$$H_{k,\mathrm{oSDFT}}(z) = \frac{H_k(z)}{1 + H_0(z)} = \frac{H_k(z)}{1 + \frac{1}{N}\sum\limits_{k=0}^{N-1} H_k(z)}, \tag{A-16}$$

where $H_k\left(z\right)$ is the transfer function of the k^{th} resonator and $H_0(z)$ is the transfer function of the open loop in the oSDFT structure. The transfer function $H_k\left(z\right)$ is determined as:

$$H_k\left(z\right) = \frac{W_N^k z^{-1}}{1 - W_N^k z^{-1}}. \tag{A-17}$$

First, we will prove that

$$\sum_{k=0}^{N-1} H_k\left(z\right) = \sum_{k=0}^{N-1} \frac{W_N^k z^{-1}}{1 - W_N^k z^{-1}} = N\frac{z^{-N}}{1 - z^{-N}}. \tag{A-18}$$

As we unfold and rearrange the first part of (A-18) using the formula for the sum of a geometric series, we obtain

$$\sum_{k=0}^{N-1} \frac{W_N^k z^{-1}}{1 - W_N^k z^{-1}} = \sum_{k=0}^{N-1} \left[\left(W_N^k z^{-1}\right)^1 + \left(W_N^k z^{-1}\right)^2 + \dots \right.$$
$$\left. \dots + \left(W_N^k z^{-1}\right)^N + \dots\right] = \sum_{p=1}^{\infty} \left[\left(z^{-1}\right)^p \cdot \sum_{k=0}^{N-1} W_N^{kp}\right]. \tag{A-19}$$

Emphasizing the fact, that for the sum of the powers of a unit root the following expression is valid

$$\sum_{k=0}^{N-1} W_N^{kp} = \begin{cases} N, & \text{if } p = 0, N, 2N, \dots \\ 0, & \text{otherwise.} \end{cases}, \tag{A-20}$$

Eq. (A-19) can be simplified – using the formula for the sum of a geometric series – to

$$\sum_{k=0}^{N-1} \frac{W_N^k z^{-1}}{1-W_N^k z^{-1}} = \Big[z^{-N} \cdot N + z^{-2N} \cdot N + \ldots$$
$$\ldots + z^{-NN} \cdot N + \ldots \Big] = N \frac{z^{-N}}{1-z^{-N}}. \qquad \text{(A-21)}$$

This is what we wanted to prove.

Now, if we substitute Eq. (A-18) into Eq. (A-16) we get the following simplified equation:

$$H_{k,\mathrm{oSDFT}}(z) = \frac{H_k(z)}{1+\frac{1}{N}N\frac{z^{-N}}{1-z^{-N}}} = \frac{\left(1-z^{-N}\right)H_k(z)}{\left(1-z^{-N}\right)+z^{-N}} =$$
$$= \left(1-z^{-N}\right)H_k(z). \qquad \text{(A-22)}$$

As a result, we have proved that the transfer function of the two structures according to Eq. (A-15) and Eq. (A-16) are equivalent. ■

APPENDIX B
PROOF OF THE EQUIVALENCE OF TWO OSDFT STRUCTURES

Both oSDFT algorithms are built upon either one of the two main substructures, namely the down-conversion–integrator–up-conversion or the resonator scheme as shown in Fig. 9. Here, we intend to show the theoretical equivalence of these two substructures. We present this statement through an alternative graphical method, while a mathematical approach can be found in [3], [12].

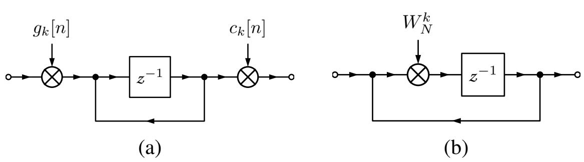

Fig. 9. oSDFT substructures: down-conversion–integrator–up-conversion (a) and resonator scheme (b)

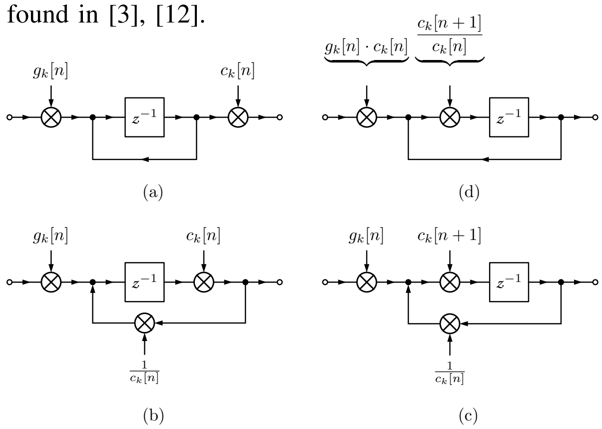

Fig. 10. Steps for proving the equivalence of the two oSDFT substructures

Starting from Step (a) in Fig. 10 :

(b) Push the up-converting sequence into the loop, before the feedback exit point. To ensure the same functionality, we have to compensate for the effect of the newly introduced in-loop multiplication into the feedback path as well.

(c) Move the up-converting sequence even further, through the delay element. Due to the delay element, only the time indexing has to be modified.

(d) Push the compensating term, introduced in Step (b), further down the feedback-loop, till it stands after the feedback entry point. Additionally, in order to counter preserve the functionality, we have to also divide the input (signal) with the compensating term. Finally, as they are in the same position, we can contract the modulating and de-modulating sequences into one term. As a result we have reached the same structure as presented in Fig. 9(b) based on the following equivalences:

$$g_k[n] \cdot c_k[n] = W_N^{-nk} \cdot W_N^{nk} = 1, \qquad \text{(B-23)}$$

$$\frac{c_k[n+1]}{c_k[n]} = c_k[1] = W_N^k. \qquad \text{(B-24)}$$

■

ACKNOWLEDGMENT

We are thankful to Prof. Péceli Gábor, for his helpful comments and suggestions. This work was supported by the János Bolyai Research Fellowship of the Hungarian Academy of Sciences.

REFERENCES

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

[2] ——, "An update to the sliding DFT," *IEEE Signal Processing Magazine*, vol. 21, no. 1, pp. 110–111, Jan 2004.

[3] K. Duda, "Accurate, guaranteed stable, sliding discrete fourier transform," *IEEE Signal Processing Magazine*, vol. 27, no. 6, pp. 124–127, Nov 2010.

[4] C. S. Park and S. J. Ko, "The hopping discrete fourier transform," *IEEE Signal Processing Magazine*, vol. 31, no. 2, pp. 135–139, March 2014.

[5] G. Hostetter, "Recursive discrete fourier transformation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 2, pp. 184–190, Apr 1980.

[6] G. Péceli, "A common structure for recursive discrete transforms," *IEEE Transactions on Circuits and Systems*, vol. 33, no. 10, pp. 1035–1036, Oct 1986.

[7] M. Kovács and Z. Kollár, "Software implementation of the recursive discrete fourier transform," in *2017 27th International Conference Radioelektronika (RADIOELEKTRONIKA)*, April 2017, pp. 1–5.

[8] A. V. Oppenheim and G. C. Verghese, *Signals, Systems, and Inference*. Pearson Education Limited, 2016.

[9] G. Péceli and G. Simon, "Generalization of the frequency sampling method," in *Joint Conference: IEEE Instrumentation and Measurement Technology Conference and IMEKO Tec*, vol. 1, 1996, pp. 339–343.

[10] G. Péceli, "Resonator-based digital filters," *IEEE Transactions on Circuits and Systems*, vol. 36, no. 1, pp. 156–159, Jan 1989.

[11] G. Simon and G. Péceli, "Convergence properties of an adaptive fourier analyzer," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 46, no. 2, pp. 223–227, Feb 1999.

[12] C. S. Park, "Fast, accurate, and guaranteed stable sliding discrete fourier transform," *IEEE Signal Processing Magazine*, vol. 32, no. 4, pp. 145–156, July 2015.

Zsolt Kollár (kollar@hvt.bme.hu) is an associate professor at the Department of Broadband Infocommunications and Electromagnetic Theory at the Budapest University of Technology and Economics, Budapest, Hungary. He is the head of the MATLAB laboratory at the University. His research interests are digital signal processing, wireless communication and quantization issues.

Ferenc Plesznik received his M.Sc. in 2016 from the Department of Measurement and Information Systems at the Budapest University of Technology and Economics, Budapest, Hungary. His main focus is digital signal processing and wireless communication systems.

Simon Trumpf is a currently an M.Sc. Student at the Karlsruhe Institute of Technology (KIT). He received his B.Sc. in 2018 from Karlsruhe Institute of Technology, Karlsruhe, Germany.