

# Observer-Based Recursive Sliding Discrete Fourier Transform

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 *fast Fourier transform (FFT)* algorithms.

Although highly optimized and efficient FFT algorithms are available, their operation remains block oriented with nonrecursive 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 fac-

tor 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 article, we investigate the observer-based SDFT (oSDFT), a lesser-known alternative solution for the recursive calculation of the DFT that is based on the observer theory. 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 article we will compare it to the SDFT and mSDFT structures.

## SDFT

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

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

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

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

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 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

$$X_k^C[n] = X_k, \quad n = 0, N, 2N \dots \quad (3)$$

The recursive equivalent of (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

$$X_k^C[n+1] = W_N^k (X_k^C[n] + (x[n] - x[n-N])). \quad (4)$$

Figure 1 shows how (4) can be implemented as a comb filter followed by a resonator stage. The resonator stage is an 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

$$H_{k, \text{SDFT}}(z) = \frac{X_k^C(z)}{X(z)} = (1 - z^{-N}) H_k(z), \quad (5)$$

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

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

This structure is considered to be only marginally stable in practice [1] as the  $W_N^k$  poles, in the 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] <AU: Should “r” be italic here? If not, the please spell out “rSDFT”.> enforcing the poles inside the unit circle by applying a constant multiplication factor  $r$ , slightly smaller than one, to all  $W_N^k$  factors. As a drawback, it leads to a modified DFT calculation, thus it gives inaccurate results [3].

### 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 Figure 2.

First, it transforms the  $k$ th frequency bin to dc ( $k = 0$ ) by a complex multiplication with the sequence  $W_N^{-kn}$ , then the calculations of (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

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

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

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

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

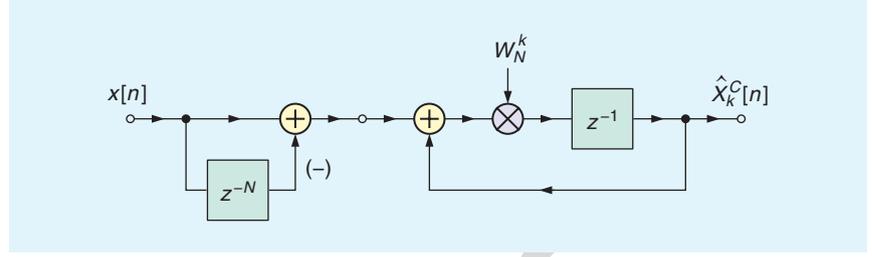


FIGURE 1. A single sliding discrete Fourier transform branch for the calculation of the  $k$ th frequency bin.

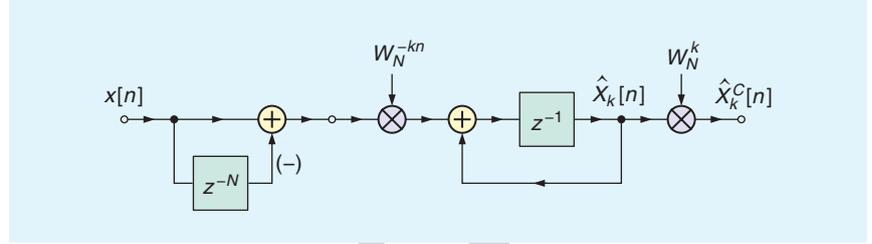


FIGURE 2. A single mSDFT branch for the calculation of the  $k$ th frequency bin.

### oSDFT

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

#### 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

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

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 Figure 3, wherein weighting factors are stored in discrete integrators as initial values.

The observer, which can be seen on the right side of Figure 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

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

where  $y[n]$  can be expressed as

$$\begin{aligned} y[n] &= \frac{1}{N} \sum_{k=0}^{N-1} c_k[n] \hat{X}_k[n] \\ &= \frac{1}{N} \sum_{k=0}^{N-1} \hat{X}_k^C[n]. \end{aligned} \quad (11)$$

Péceli proves the following four 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  $n = 0 \dots N-1$  with a normalization factor of  $1/N$ :

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

Moreover, in this scenario the system is deadbeat 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 (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

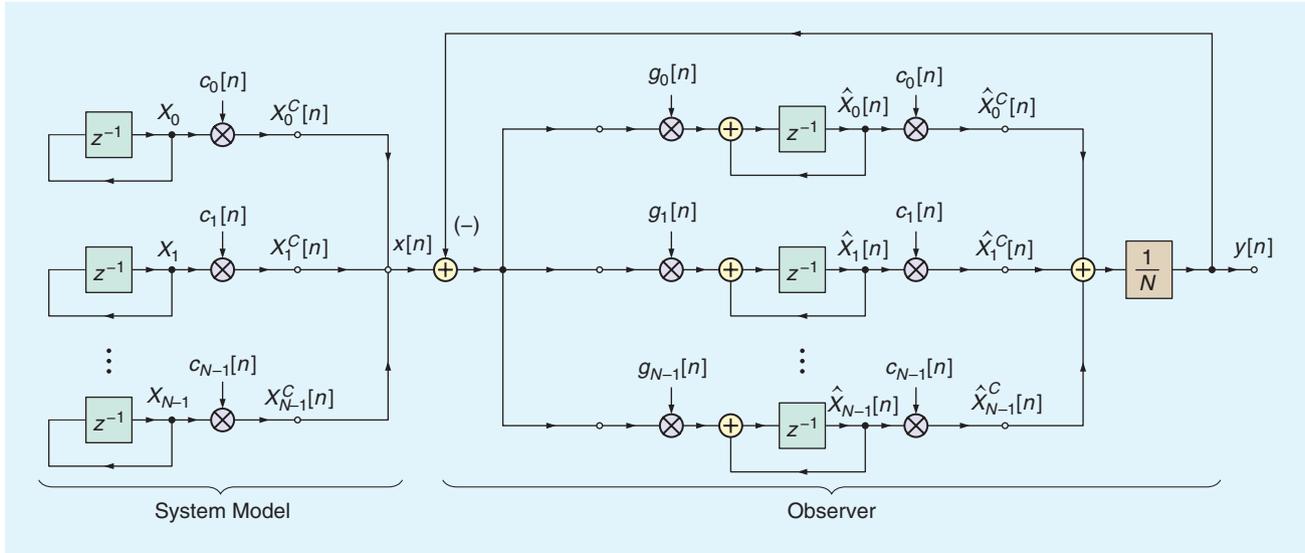


FIGURE 3. The observer theory model: system model and observer.

$x[n]$  as presented in (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

$$H_{k, \text{oSDFT}}(z) = \frac{X_k^C(z)}{X(z)} = \frac{H_k(z)}{1 + \frac{1}{N} \sum_{k=0}^{N-1} H_k(z)}, \quad (13)$$

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

- 4) The oSDFT structure is equivalent to the SDFT structure presented in Figure 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 “Proof of the Equivalence of the SDFT and oSDFT.”

### Resonator-based oSDFT

An alternative version of the oSDFT structure is depicted in Figure 4, which is based solely on resonators, which are IIR filters, without down- and up-converters, similar to the SDFT structure. The proof of the equivalence of the two oSDFT structures is provided in “Proof of the Equivalence of Two oSDFT Structures.”

### 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 1.

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 used  $N$  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 ( $W_N^k$  and  $W_N^{-nk}$ ) from a look-up-table (LUT). The LUT stores  $N$  samples for each branch, as the values are periodic to  $N$ .

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

For all algorithms, each branch requires one two-input adder. In case of the oSDFT structures, they both apply

an  $N$ -input adder to calculate the feedback signal  $y[n]$  and a two-input adder is used to calculate the difference of the input and the feedback signal as shown in (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. The spectral components will be available faster compared to the block-wise operational FFT where this can be performed in  $\log_2 N$  steps.

### Simulations

#### Floating-point implementation

A simulation environment for the comparison of the aforementioned sliding DFT algorithms (i.e., mSDFT, oSDFT, and resonator-based oSDFT) was developed in MATLAB2017a (x64 PC). For the algorithms we applied 32-bit, single-precision, 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  $N = 64$

## Proof of the Equivalence of the SDFT and oSDFT

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

$$H_{k,\text{SDFT}}(z) = (1 - z^{-N}) \cdot H_k(z), \quad (S1)$$

$$H_{k,\text{oSDFT}}(z) = \frac{H_k(z)}{1 + H_0(z)} = \frac{H_k(z)}{1 + \frac{1}{N} \sum_{k=0}^{N-1} H_k(z)}, \quad (S2)$$

where  $H_k(z)$  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  $H_k(z)$  is determined as

$$H_k(z) = \frac{W_N^k z^{-1}}{1 - W_N^k z^{-1}}. \quad (S3)$$

First, we will prove that

$$\sum_{k=0}^{N-1} H_k(z) = \sum_{k=0}^{N-1} \frac{W_N^k z^{-1}}{1 - W_N^k z^{-1}} = N \frac{z^{-N}}{1 - z^{-N}}. \quad (S4)$$

As we unfold and rearrange the first part of (S4) using the formula for the sum of a geometric series, we obtain

$$\sum_{k=0}^{N-1} \frac{W_N^k z^{-1}}{1 - W_N^k z^{-1}} = \sum_{k=0}^{N-1} [(W_N^k z^{-1})^1 + (W_N^k z^{-1})^2 + \dots + (W_N^k z^{-1})^N + \dots] = \sum_{p=1}^{\infty} \left[ (z^{-1})^p \cdot \sum_{k=0}^{N-1} W_N^{kp} \right]. \quad (S5)$$

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

$$\sum_{k=0}^{N-1} W_N^{kp} = \begin{cases} N, & \text{if } p = 0, N, 2N, \dots \\ 0, & \text{otherwise.} \end{cases}, \quad (S6)$$

We can simplify (S5) using the formula for the sum of a geometric series to

$$\sum_{k=0}^{N-1} \frac{W_N^k z^{-1}}{1 - W_N^k z^{-1}} = [z^{-N} \cdot N + z^{-2N} \cdot N + \dots + z^{-NN} \cdot N + \dots] = N \frac{z^{-N}}{1 - z^{-N}}. \quad (S7)$$

This is what we wanted to prove.

Now, if we substitute (S4) into (S2) we get the following simplified equation:

$$H_{k,\text{oSDFT}}(z) = \frac{H_k(z)}{1 + \frac{1}{N} N \frac{z^{-N}}{1 - z^{-N}}} = \frac{(1 - z^{-N}) H_k(z)}{(1 - z^{-N}) + z^{-N}} = (1 - z^{-N}) H_k(z). \quad (S8)$$

As a result, we have proved that the transfer function of the two structures according to (S1) and (S2) are equivalent. ■

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.

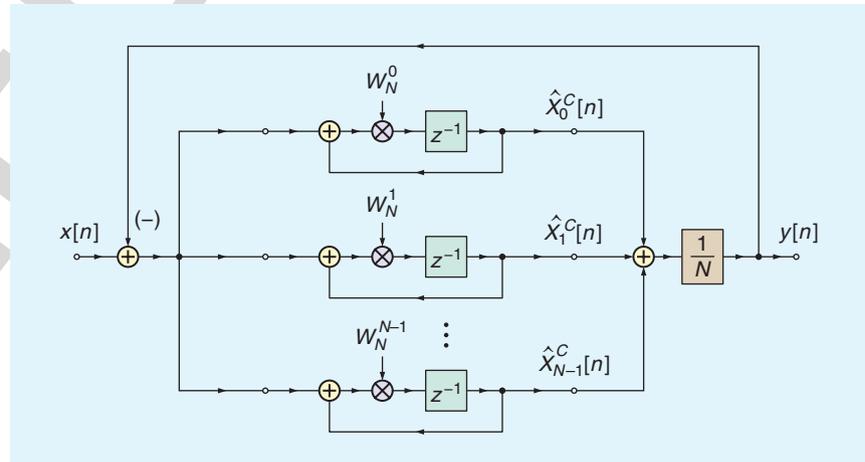


FIGURE 4. A resonator-based oSDFT.

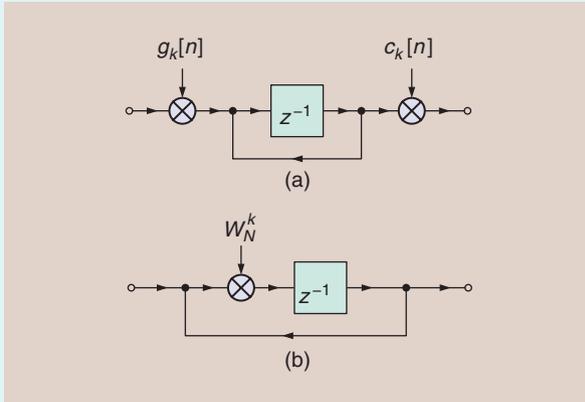
The results of the various SDFT methods were compared through double-precision arithmetic to the results of the sliding FFT, and the average error signal over the branches was formulated

as

$$\varepsilon[n] = \frac{1}{N} \sum_{k=0}^{N-1} |\hat{X}_{k,\text{SDFT}}^C[n] - \hat{X}_{k,\text{FFT}}^C[n]|, \quad (14)$$

## 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 Figure S1. 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] and [12].



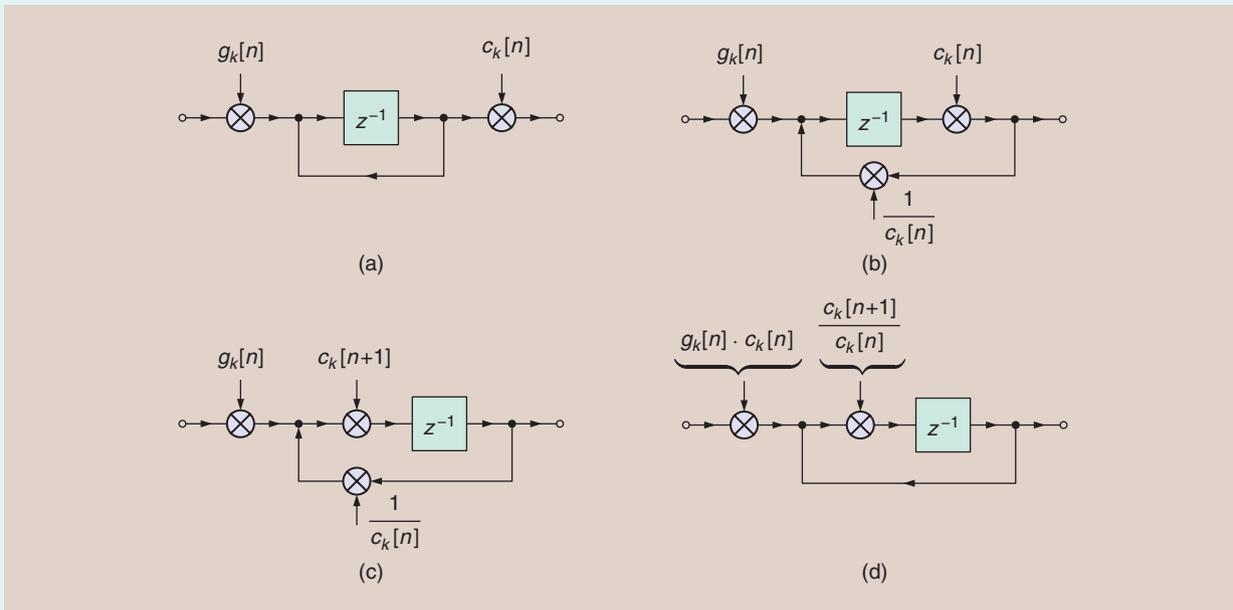
**FIGURE S1.** Substructures of the oSDFT: the (a) down conversion–integrator–up conversion and (b) resonator scheme.

Starting from Step (a) in Figure S2:

- (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, until it stands after the feedback entry point. Additionally, to counter preserve the functionality, we have to also divide the input (i.e., signal) with the compensating term. Finally, as they are in the same position, we can contract the modulating and demodulating sequences into one term. As a result, we have reached the same structure as presented in Figure S1(b) based on the following equivalences:

$$g_k[n] \cdot c_k[n] = W_N^{-nk} \cdot W_N^{nk} = 1, \quad (\text{S9})$$

$$\frac{c_k[n+1]}{c_k[n]} = c_k[1] = W_N^k. \quad (\text{S10})$$



**FIGURE S2.** (a)–(d) The steps for proving the equivalence of the two oSDFT substructures.

where xSDFT stands for the mSDFT and oSDFT algorithms.

In Figure 5, the error progress in the function of the time index is compared

in the case of mSDFT and oSDFT. The error of the mSDFT is not stable and 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 Figure 6 the error progress in function of the time index is shown

for the oSDFT and the resonator-based oSDFT algorithms. Both algorithms produce a stable-but-noisy error over the discrete time samples. Additionally, the oSDFT outperforms the resonator-based oSDFT.

The observed performance difference between the two oSDFT structures 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 Figure 6 are an offset and a higher noise variance.

The offset is caused by the fact that for every step of  $n$  the  $W_N^k$  modulator and demodulator values for the oSDFT are taken periodically from a precomputed sequence stored in a LUT, and within this LUT the error introduced by rounding (i.e., finite precision storage) is averaged out over a sequence period. This way the oSDFT's modulation and demodulation process will be more precise regarding the average frequency accuracy 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, 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

**Table 1. The complexity comparison of the various SDFT structures.**

Type	Memory		Adders			
	Read-Only Memory	Random-Access Memory	LUT <sub>N</sub>	Multipliers	Two-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 (resonator)	$N$	$N$	0	$N$	$N + 1$	1

**<AU: Kindly confirm whether "res." has been correctly spelled out as "resonator".>**

variance due to the structure, thus it will lead to slightly misplaced  $W_N^k$  poles in a random manner over the complex plane.

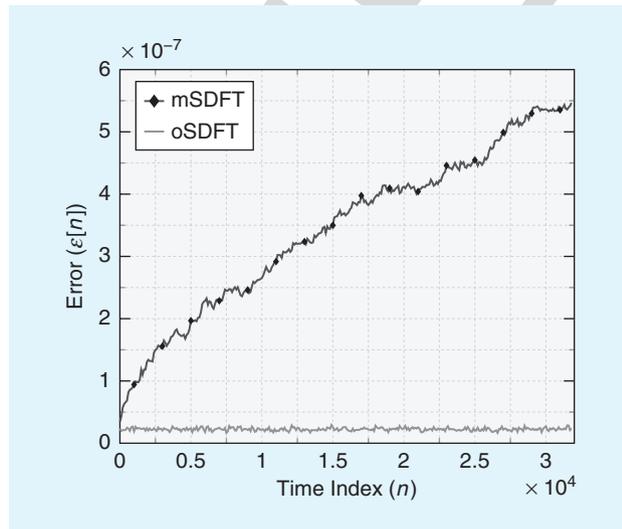
### Fixed-point implementation

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 toward zero method was applied to maintain the 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 earlier.

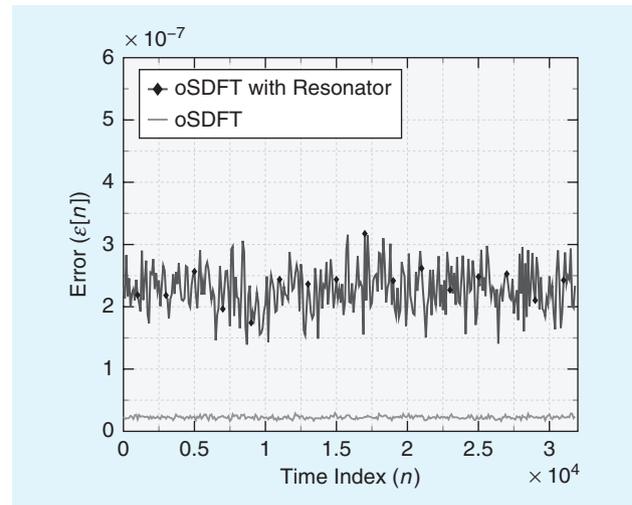
The error progress of the various SDFT structures for signed Q0.31 for-

mat fixed-point implementation can be seen in Figure 7. The results are similar to the case where the structures are implemented using single-precision, although the overall errors are slightly smaller for each method. The reason for this is that, the IEEE 754-2008 single-precision standard, used by MATLAB, the fractional part is defined as only 23 bits, thus within the same range, it offers lower resolution, resulting in more imprecise  $W_N^k$  modulator and  $W_N^k$  pole values.

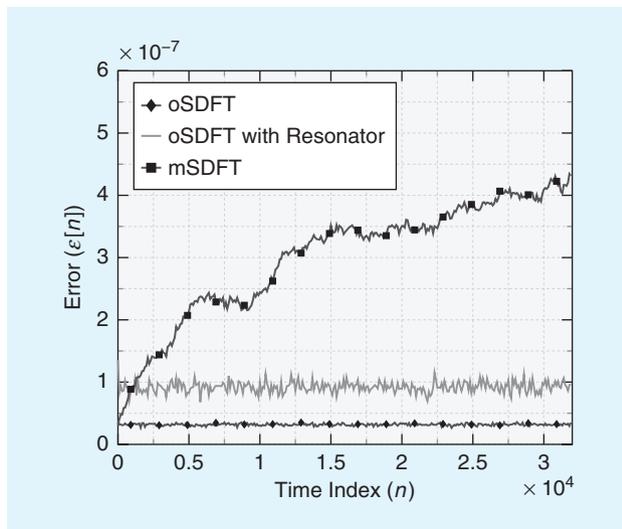
Furthermore, the averaged error  $\varepsilon[n]$  over the samples  $n$  for the fixed-point implementation in function of the fraction part is shown in Figure 8. For both methods with an enlarged fraction part, the error is exponentially decreasing.



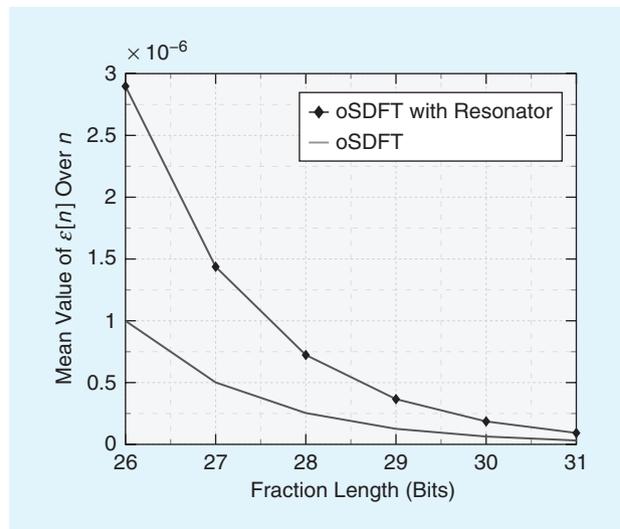
**FIGURE 5.** The error progress of the mSDFT and the oSDFT algorithms.



**FIGURE 6.** The error progress of the oSDFT and the resonator-based oSDFT algorithms.



**FIGURE 7.** The error progress of the oSDFT, the resonator-based oSDFT, and the mSDFT algorithms with fixed-point implementation in signed Q0.31 format using a 32- or 31-bit fractional part.



**FIGURE 8.** The mean error of the oSDFT and the resonator-based oSDFT in function of the fraction length of the signed 32-bit fixed-point implementation.

## Summary

In this article, 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 not only for the long-term stability but because 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].

## 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.

## Authors

**Zsolt Kollár** (kollar@hvt.bme.hu) <AU: Please provide undergraduate and graduate information.> is an associate professor at the Department of Broadband Infocommunications and Electromagnetic Theory at the Budapest University of Technology and Economics, Budapest, Hungary, where he is the head of the MATLAB laboratory. His research interests are digital signal processing, wireless communication, and quantization issues.

**Ferenc Plesznik** (plesznik@yahoo.co.uk) received his M.Sc. degree in 2016 from the Department of Measurement and Information Systems at the Budapest University of Technology and Economics, Hungary. His main focus is digital signal processing and wireless communication systems. <AU: Please provide undergraduate degree information and clarify what field the M.Sc. degree is in.>

**Simon Trumpf** (simon.trumpf@student.kit.edu) received his B.Sc. degree in 2018 from Karlsruhe Institute

of Technology, Germany. <AU: Please clarify in what field.> He is a currently an M.Sc. degree student at the Karlsruhe Institute of Technology. <AU: Please clarify in what field.>

## References

- [1] E. Jacobsen and R. Lyons, "The sliding DFT," *IEEE Signal Process. Mag.*, vol. 20, no. 2, pp. 74–80, Mar. 2003.
- [2] E. Jacobsen and R. Lyons, "An update to the sliding DFT," *IEEE Signal Process. Mag.*, vol. 21, no. 1, pp. 110–111, Jan. 2004.
- [3] K. Duda, "Accurate, guaranteed stable, sliding discrete Fourier transform," *IEEE Signal Process. Mag.*, vol. 27, no. 6, pp. 124–127, Nov. 2010.
- [4] C. S. Park and S. J. Ko, "The hopping discrete Fourier transform," *IEEE Signal Process. Mag.*, vol. 31, no. 2, pp. 135–139, Mar. 2014.
- [5] G. Hostetter, "Recursive discrete Fourier transformation," *IEEE Trans. Acoustics, Speech, Signal Process.*, vol. 28, no. 2, pp. 184–190, Apr. 1980.
- [6] G. Péceli, "A common structure for recursive discrete transforms," *IEEE Trans. Circuits Syst.*, vol. 33, no. 10, pp. 1035–1036, Oct. 1986.
- [7] M. Kovács and Z. Kollár, "Software implementation of the recursive discrete Fourier transform," in *Proc. 2017 27th Int. Conf. Radioelektronika*, April 2017, pp. 1–5.
- [8] A. V. Oppenheim and G. C. Verghese, *Signals, Systems, and Inference*. Upper Saddle River, NJ: Pearson Education Limited, 2016.
- [9] G. Péceli and G. Simon, "Generalization of the frequency sampling method," in *Proc. IEEE Instrumentation and Measurement Technology Conf. IMEKO Tec*, vol. 1, 1996, pp. 339–343.
- [10] G. Péceli, "Resonator-based digital filters," *IEEE Trans. Circuits Syst.*, vol. 36, no. 1, pp. 156–159, Jan. 1989.
- [11] G. Simon and G. Péceli, "Convergence properties of an adaptive Fourier analyzer," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 46, no. 2, pp. 223–227, Feb. 1999.

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

SP

IEEE Proof

Although highly optimized and efficient FFT algorithms are available, their operation remains block oriented with nonrecursive operations.

The basic idea behind the SDFT algorithm is to recursively calculate the DFT spectrum of the input stream.

With this described technique, the resonators became stable integrators performing simple averaging.

Independent from the chosen algorithms,  $N$  registers are required for storing the state variables of the resonators or the integrators.

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.

IEEE PROCEEDINGS