Frequency-Domain Implementations of Variable Digital FIR Filters Using the Overlap-Save Technique

Oksana Moryakova and Håkan Johansson
Division of Communication Systems, Department of Electrical Engineering, Linköping University, Linköping, Sweden
Email: oksana.moryakova@liu.se, hakan.johansson@liu.se

***Abstract*—The paper introduces frequency-domain implementations of variable digital filters using the overlap-save method. Expressions for implementation and design complexities are derived for real-valued impulse responses. Design examples include implementations of a variable bandwidth (VBW) filter alone as well as a cascade of a VBW filter and a variable fractional delay (VFD) filter. Compared to a time-domain implementation and a filter bank approach, the proposed structures can reduce the implementation complexity significantly and achieve savings up to 95% in the multiplication rate and up to 89% in the addition rate.**

***Index Terms*—Variable digital filter, frequency-domain implementations, implementation complexity, overlap-save.**

I. Introduction

Variable digital filters (VDFs) [1] find many applications in digital signal processing due to the increasing demand for reconfigurable systems required in, for example, medical devices [2], [3] and wireless communication systems [4], [5]. The Farrow structure [6] has received great attention in realizing VDFs, especially for variable bandwidth (VBW) [4], [5], [7] and variable fractional delay (VFD) filters [8]–[11]. In this technique, the overall transfer function is expressed as a weighted linear combination of several fixed linear-phase finite-impulse-response (FIR) filters. This offers offline filter design and simple online update when switching from one mode to another. However, the fact that an input sequence is processed in the time domain may still cause a rather high computational complexity for stringent requirements.

To reduce the complexity, this paper introduces new implementations of VDFs using the overlap-save method[1] [12]. The proposed structures are based on three facts. Firstly, a filter implemented in the frequency domain can be computationally more efficient than in the time domain [13]–[16]. Secondly, the overlap-save method employs fast Fourier transform/inverse fast Fourier (FFT/IFFT) transform pairs which can be shared for several subfilters in a frequency-domain implementation of a VDF. Thirdly, the filter's discrete Fourier transform (DFT) coefficients (hereafter also referred to as weights) can be expressed as a linear combination of fixed and general multiplications. The overlap-save technique has been utilized earlier, for example, in chromatic dispersion compensation filters [14], [17] and in filter banks [18], [19], but it has not been explored in the context of variable filters.

This paper is organized as follows. Sections II gives a brief overview of the overlap-save method. In Section III, the proposed frequency-domain implementations of VDFs are presented and the corresponding complexity expressions are derived. Section IV provides examples of applying the proposed structures for a VBW filter and a cascade of a VBW filter and a VFD filter. Finally, Section V concludes the paper.

II. Overlap-Save Method

The output sequence of an FIR filter can be expressed in the time domain as linear convolution of an input sequence and the filter impulse response $h(n)$ of length L. In the frequency domain, it corresponds to a multiplication of DFT coefficients and can be implemented using the overlap-save technique [12], which employs overlapping input segments $x_m(n) = x(n + mM)$, $n = 0, 1, ...N-1$, where m is a segment index, and a zero-padded impulse response sequence according to

$$h_z(n) = \begin{cases} h(n), & n = 0, 1, ..., L-1, \\ 0, & n = L, L+1, ..., N-1. \end{cases} \tag{1}$$

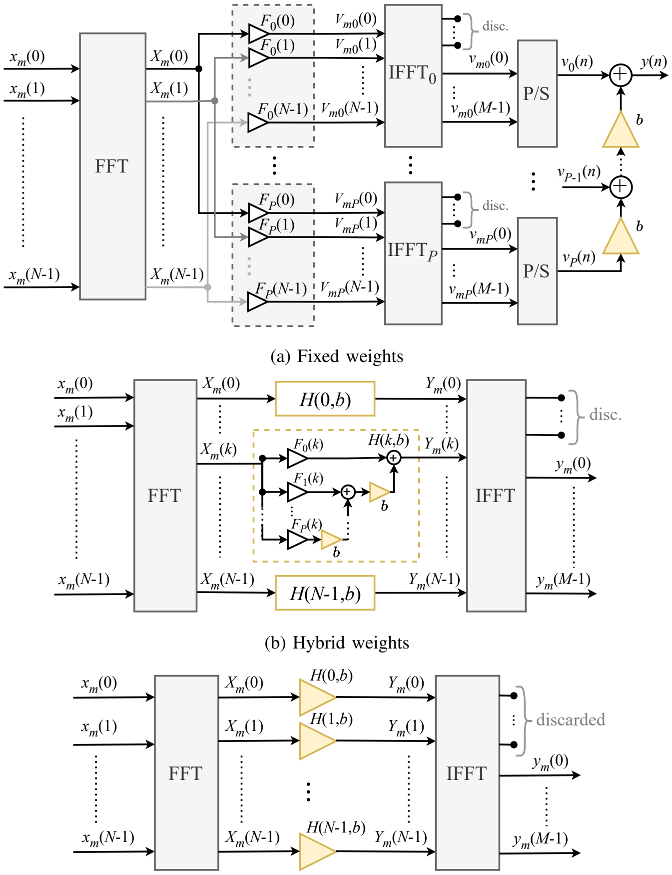
The length-N DFT coefficients of $h_z(n)$ are denoted as $H(k)$ for $k = 0, 1, ..., N-1$, where $N = M + L - 1$. Each corresponding output segment $y_m(n)$ can be computed by multiplying $H(k)$ by the length-N FFT of $x_m(n)$, then computing the length-N IFFT of the so obtained result and finally discarding the first $L-1$ samples. Therefore, the output segments are of length M and not overlapping. The output sequence can be obtained as $y(n) = \sum_{m=0}^{\infty} y_m(n - mM)$.

III. Proposed Frequency-Domain Implementations of Variable Filters

Previous publications on VDFs have considered time-domain implementations [1]–[5], [7]–[10]. One of the most common is a structure based on a weighted linear combination of fixed subfilters. It means that the frequency response is expressed as

$$F(e^{j\omega T}, b) = \sum_{p=0}^{P} b^p F_p(e^{j\omega T}), \tag{2}$$

[1]The overlap-add method can also be used. However, the overlap-save method requires slightly less additions per sample than the overlap-add technique while the number of multiplications is the same for both methods.



(a) Fixed weights

(b) Hybrid weights

(c) Variable weights

Fig. 1: Proposed frequency-domain implementations of a VBW filter.

where $F_p(e^{j\omega T})$ are typically linear-phase FIR filters of order $L-1$. For example, for a VBW filter, $b \in [b_l, b_u]$ is a variable parameter determining the center of the transition band. For obtaining a more balanced structure with smaller filter coefficients, b is replaced by $b - b_0$, where $b_0 = (b_l + b_u)/2$ [7]. For a VFD filter, the variable parameter $b \in [-1/2, 1/2]$ is a fractional delay [10].

Based on (2), we introduce three frequency-domain VDF implementation structures and derive complexity expressions for each of them. All proposals assume that the impulse responses $f_p(n)$ of the subfilters are designed only once for a real-valued input sequence [7]. Therefore, $f_p(n)$ are assumed to be real-valued.

A. Proposed VDF Structures

1) Fixed Weights: The first frequency-domain implementation of the VBW filter is shown in Fig. 1a. Compared to the time-domain approach in (2), here, fixed filter DFT coefficients $F_p(k)$ are multiplied by DFT coefficients $X_m(k)$ of the input segments while the variable coefficients b remain being implemented in the time domain. It means that the number of IFFTs required for this implementation is $P+1$. The input of the p-th IFFT can be computed as $V_{mp}(k) = X_m(k)F_p(k)$, $p = 0, 1, ..., P$, $k = 0, 1, ..., N-1$, $N \geq L$. The segments $x_m(n)$ are of length N and $v_{mp}(n)$ are of length M since the first $L-1$ samples of the length-N IFFT are discarded. The output of the VBW filter is computed as

$$y(n) = \sum_{p=0}^{P} b^p v_p(n), \tag{3}$$

where $v_p(n)$ is the output of the corresponding parallel-to-serial (P/S) converter.

2) Hybrid Weights: The hybrid-weight-based structure shown in Fig. 1b allows all computations to be implemented in the frequency domain. Therefore, weighting of the DFT coefficient $X_m(k)$ in each branch is implemented as a linear combination of $P+1$ fixed coefficients $F_p(k)$ and P variable coefficients b, i.e.,

$$H(k,b) = \sum_{p=0}^{P} b^p F_p(k). \tag{4}$$

This implementation allows to use only one IFFT per segment by placing variable multipliers in the frequency domain that can significantly reduce the implementation complexity.

3) Variable Weights: The third structure, which is shown in Fig. 1c, utilizes only N adjustable DFT coefficients $H(k,b)$ placed in the frequency domain. Here, each multiplier is recomputed as in (4) and stored in memory whenever b is changed. This approach enables to reduce the number of computations per sample even more compared to the hybrid weighting approach.

B. Implementation Complexity

1) Implementation Complexity of FFT/IFFT: The complexity of FFT/IFFT algorithms depends on whether the complex multiplications are implemented with three real multiplications and three real additions (a 3/3 algorithm) or with four real multiplications and two real additions (a 4/2 algorithm) [20]. Here, we consider the 3/3 algorithm for complex multiplications inside FFT and IFFT operations since twiddle factors are fixed and can be stored [21]. A complex addition requires two real additions.

A common FFT algorithm is split-radix as it uses the lowest total number of operations [20], [22]. In this paper, we assume that the segments $x_m(n)$ and $y_m(n)$ are real-valued and $N = 2^Q$. Therefore, both the FFT and IFFT require $C_{m,F} = (1/2)N\log_2 N - (3/2)N + 2$ multiplications and $C_{a,F} = (3/2)N\log_2 N - (5/2)N + 4$ additions [20].

2) Implementation Complexity of Weighting: For operations outside the FFT and IFFT algorithms, complex multiplications can be realized as the 3/3 algorithm for fixed multiplications[2] and as the 3/5 algorithm for variable multiplications. Moreover, since the DFT coefficients of $x_m(n)$ and $f_p(n)$ are conjugate symmetric, only half of the multiplications $X_m(k)H(k,b)$ (or $X_m(k)F_p(k)$ for the fixed weighting approach) needs to be carried out.

[2]Fixed multiplication here means multiplication of any value by a fixed coefficient.

Multiplication and addition rate	Fixed weights	Hybrid weights	Variable weights
R_{mv}	P	$\frac{PN}{N-L+1}$	$\frac{3N}{2(N-L+1)}$
R_{mf}	$\frac{(P+2)(N\log_2 N+4)-3N}{2(N-L+1)}$	$\frac{N\log_2 N+3N(P-1)/2+4}{N-L+1}$	$\frac{N\log_2 N-3N+4}{N-L+1}$
R_a	$\frac{(P+2)(3N\log_2 N+8)-(2P+7)N}{2(N-L+1)}+P$	$\frac{3N\log_2 N+(5P-7)N/2+8}{N-L+1}$	$\frac{3N\log_2 N-5N/2+8}{N-L+1}$

TABLE I: Multiplication and addition rates for the proposed structures.

The structure based on the fixed weighting requires only fixed complex multiplications in the frequency domain. Therefore, the number of real fixed multiplications and real additions is computed as $C_{mf,W} = C_{a,W} = (3/2)(P+1)N$.

The hybrid weighting approach consists of $P+1$ fixed complex coefficients and P adjustable real ones. Considering that a multiplication of a complex number by a real number requires two real multiplications, this structure requires $C_{mf,W} = (3/2)(P+1)N$ fixed multiplications, $C_{mv,W} = PN$ general multiplications, and $C_{a,W} = (5P+3)N/2$ additions for N branches.

The structure based on the variable weighting includes only one variable complex multiplication per each branch k. Therefore, the total number of real general multiplications and real additions is obtained as $C_{mv,W} = (3/2)N$ and $C_{a,W} = (5/2)N$, respectively.

3) Total Implementation Complexity: The total number of real fixed multiplications, real general multiplications and real additions per sample can be expressed as a fixed multiplication rate R_{mf}, variable multiplication rate R_{mv}, and addition rate R_a. Table I summarizes the expressions for the proposed implementation structures.

Since L and P are fixed, once the filter is designed, the total implementation complexity depends only on N. To minimize this, it is therefore necessary to obtain the optimal $N = N_{opt}$ for which R_{mf} and R_{mv} are minimum since a multiplication is considerably more complex than an addition in hardware [21]. Thus, $N_{opt} = \lceil \text{argmin}_N R_{mf} \rceil$ for the fixed weighting approach and $N_{opt} = \lceil \text{argmin}_N (R_{mf} + R_{mv}) \rceil$ otherwise. For the structure based on variable weighting, one can use the expression $N_{opt} = 0.9L\log_2 L$ that has been derived in [16] for a general FIR filter. As will be demonstrated in Section IV, the values of M are the same for all the structures after rounding to the nearest power of two number. Therefore, the same N_{opt} can be used for all the proposed structures.

C. Design Complexity

Design complexity includes the cost of variable coefficient calculations within a block of M samples for which b is altered.

The main feature of the structures shown in Figs. 1a and 1b is that they utilize the adjustable parameter b as a separate multiplier. Therefore, there is no computations for these structures when b is changed.

The variable-weight-based structure consists of the adjustable coefficients $H(k,b)$ which need to be recomputed for each new b-value, i.e., some values are required to be stored in a look-up table (LUT). It can be implemented in two ways:

(a) Store the precomputed $F_p(k)$ in a LUT and use them to compute $H(k,b)$ whenever b is changed. In this case, the design complexity consists of only computations in (4), where only half of the general multiplications need to be carried out because of conjugate symmetry of the DFT coefficients. To compute $H(k,b)$ for $k = 0,1,...,N-1$, the computational rates are obtained as $R_{m,d} = (PN + P-1)/M$ and $R_{a,d} = PN/M$. The number of real values to place in a LUT is $(P+1)N$.

(b) Store the filter coefficients $f_p(n)$ and compute $F_p(k)$ when b is changed. This approach allows to reduce memory requirements but involves computations in the design process. The DFT coefficients $F_p(k)$ can be efficiently computed using a length-N FFT of $f_{zp}(n)$, where $f_{zp}(n)$ is the zero-padded impulse response obtained from $f_p(n)$ as in (1). The design complexity rates are then given as $R_{m,d} = (C_{m,F} + PN + P - 1)/M$ and $R_{a,d} = (C_{a,F} + PN)/M$. In this paper, we consider that $f_p(n)$ are Type I linear-phase FIR filters of even order $L-1$ [7]. Therefore, only $(L+1)(P+1)/2$ real values need to be placed in a LUT.

IV. Design Examples

A. Example A: VBW Filter

This example applies the proposed frequency-domain implementations to a VBW filter and compares them with its time-domain implementation [7]. The specifications of the VBW filter are given as follows: $b = [0.8\pi, 0.9\pi]$, $\Delta = 0.1\pi$, $\delta_c = 0.01$, $\delta_s = 0.00316$. According to the filter design procedure in [7], the specifications are met for $P = 4$ and $L = 51$. Figure 2 plots the magnitude response of the VBW filter that is the same for all the proposed implementations. Tables IIa and IIb compares the implementation and design complexities, respectively. It is worth noting that the time domain implementation in [7] uses $b - b_0$ as a variable coefficient. Therefore, for all the proposed structures, the additional design complexity per sample is counted as $1/M$ additions that is approximately zero in this example. The value of b_0 needs to be stored in memory.

It is seen that the fixed multiplication rate R_{mf} and addition rate R_a are less for the proposed frequency-domain implementations compared to the time-domain implementation. The variable multiplication rate R_{mv} is almost the same for all structures except the structure based on the variable weights which offers the smallest number and allows to decrease the total implementation complexity significantly. However, in the design complexity, this implementation requires multiplications, additions, and memory for storing coefficients. Depend-

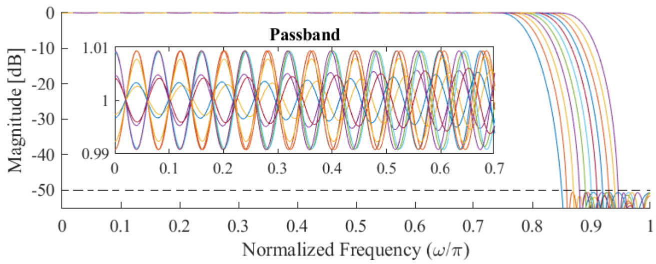


Fig. 2: Modulus of the frequency response (upper) and weighted error $E(j\omega T, b)$ (lower) for Example A filter.

Technique	L	N	M	R_{mf}	R_{mv}	R_a
Time-domain [7]	51	-	-	130	4	255
Fixed weighting	51	256	206	28	4	84.3
Saving to [7]	-	-	-	78%	0%	67%
Hybrid weighting	51	256	206	15.6	5	37.9
Saving to [7]	-	-	-	88%	-25%	85%
Variable weighting	51	256	206	6.2	1.9	26.8
Saving to [7]	-	-	-	95%	53%	89%

(a) Implementation complexity

Technique	L	N	M	R_{mv}	R_a	Memory
Time-domain [7]	51	-	-	0	1	1
Fixed (Prop.)	51	256	206	0	0	1
Hybrid (Prop.)	51	256	206	0	0	1
Variable:a (Prop.)	51	256	206	5	5	1281
Variable:b (Prop.)	51	256	206	8.1	16.8	131

(b) Design complexity

TABLE II: Complexity per sample for Example A: VBW filter.

ing on the application, either (a) the number of computations or (b) memory requirements can be reduced.

Despite the relatively large percentage of savings for all the proposed implementations, two of them (hybrid-weight-based and variable-weight-based) have a potential drawback, namely, the presence of transients that will distort the signal whenever the variable parameter b is altered within a block of M samples. Such structures are suitable in applications where the parameters do not change from sample to sample, for instance, in communication applications where the systems are time invariant during a coherence interval and allows for parameter changes in the time-domain guard intervals. The fixed weighing approach allows to avoid the limitation caused by the transients since all variable coefficients are located after delay elements.

B. Example B: VBWFD Filter

This example compares implementations of a variable bandwidth and fractional delay (VBWFD) filter using the proposed structures, a time-domain implementation in [9], and an improved fast filter bank (FBB) approach in [23].

For the VBWFD filter, a variable DFT coefficient can be obtained as $H(k, b, d) = \sum_{p=0}^{P_F} (b - b_0)^p F_p(k) \sum_{l=0}^{P_G} d^l G_l(k)$ [9] as shown in Fig. 3. In terms of notations discussed above, here $P = P_F$ for the VBW part and the total filter order is $L = L_F + L_G - 1$ for $L_F - 1$ and $L_G - 1$ being the filter order of a VBW filter and a VFD filter, respectively. For the hybrid-

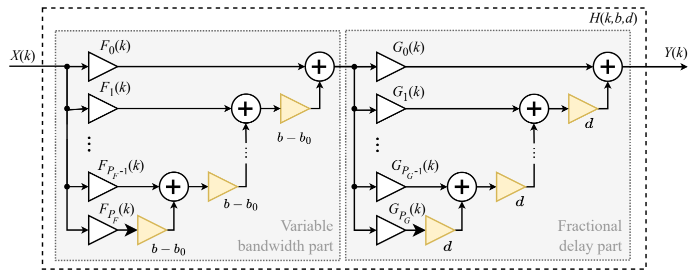


Fig. 3: Polynomial realization of the variable coefficient $H(k, b, d)$ for a VBWFD filter.

Technique	L	N	M	R_{mf}	R_{mv}	R_a
Time-domain [9]	43	-	-	174	23	335
FBB [23]	-	-	-	91	26	283
Fixed weighting	43	128	86	142	23	366.1
Saving to [9]	-	-	-	18%	0%	-9%
Saving to [23]	-	-	-	-55%	12%	-29%
Hybrid weighting	43	128	86	28.3	11.9	58.1
Saving to [9]	-	-	-	84%	48%	83%
Saving to [23]	-	-	-	69%	54%	79%
Variable weighting	43	128	86	6	2.2	27.6
Saving to [9]	-	-	-	97%	90%	92%
Saving to [23]	-	-	-	93%	92%	89%

TABLE III: Implementation complexity per sample for Example B: VBWFD filter.

weight-based implementation, the fractional delay part brings in additional complexity, specifically, $3N(P_G + 1)/(2M)$ to the fixed multiplication rate, $P_G N/M$ to the variable multiplication rate, and $(5P_G + 3)N/(2M)$ to the addition rate. Regarding the fixed-weight-based structure, the number of required IFFTs is $(P_F + 1)(P_G + 1)$. This obviously causes large implementation complexity. Additionally, this structure requires $3N(P_F+1)(P_G+2)/(2M)$ fixed multiplications and general additions per sample in the frequency domain and $P_G(P_F+1)+P_F$ general additions and multiplications in the time domain. For the variable-weight-based structure, only the number of computations in the design complexity increases.

The specifications as in [9] and [23] are considered for which $P_F = 5$, $P_G = 3$, $L_F = 37$, and $L_G = 7$. Table III provides implementation complexity comparison. It is seen that the proposed hybrid and variable weighting approaches employ the smallest number of computations and achieve significant savings compared to both [9] and [23]. The implementation based on the fixed weighting is less efficient here. Therefore, this structure is mainly attractive for filters with variability of only one parameter.

V. Conclusions

This paper proposed frequency-domain implementations of variable filters using the overlap-save method. Three structures of a VDF were introduced and expressions for implementation and design complexities were derived. The design examples for a VBW filter and a VBWFD filter demonstrated that the proposed implementations can achieve up to 95% and 89% saving regarding multiplications and additions, respectively, in the implementation complexity.

References

[1] S. Dhabu, A. Ambede, N. Agrawal, K. G. Smitha, S. Darak, and A.P. Vinod, "Variable cutoff frequency FIR filters: A survey," *SN Applied Sciences*, vol. 2, no. 3, Feb. 2020.

[2] N. Haridas and E. Elias, "Efficient Farrow structure based bank of variable bandwidth filters for digital hearing aids," in *Proc. 2015 IEEE Int. Conf. Signal Process., Informat., Communic., Energy Syst. (SPICES)*, Feb. 19-21, 2015, pp. 1–5.

[3] T. Ma, C. Shen, and Y. Wei, "Adjustable filter bank design for hearing aids system," in *Proc. 2019 IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 26-29, 2019, pp. 1–5.

[4] I. Raghu and E. Elias, "Low complexity spectrum sensing technique for cognitive radio using Farrow structure digital filters," *Eng. Science Tech., Int. Journal*, vol. 22, no. 1, pp. 131–142, Feb. 2019.

[5] N. Haridas and E. Elias, "Reconfigurable Farrow structure-based FRM filters for wireless communication systems," *Circuits Syst. Signal Process.*, vol. 36, no. 1, pp. 315–338, Jan. 2017.

[6] C. W. Farrow, "A continuously variable delay element," in *Proc. IEEE Int. Symp. Circuits Syst.*, June 7–9, 1988, vol. 3, pp. 2641–2645.

[7] P. Löwenborg and H. Johansson, "Minimax design of adjustable-bandwidth linear-phase FIR filters," *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 53, no. 2, pp. 431–439, Feb. 2006.

[8] T. O. Otunniyi and H. C. Myburgh, "Low-complexity filter for software-defined radio by modulated interpolated coefficient decimated filter in a hybrid Farrow," *Sensors*, vol. 22, no. 3, Feb. 2022.

[9] H. Johansson and A. Eghbali, "A realization of FIR filters with simultaneously variable bandwidth and fractional delay," *Proc. 20th European Signal Processing Conference (EUSIPCO)*, pp. 2178–2182, Aug. 27-31, 2012.

[10] H. Johansson and P. Löwenborg, "On the design of adjustable fractional delay FIR filters," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 50, no. 4, pp. 164–169, Apr. 2003.

[11] R. Zhao and D. Tay, "A complex exponential structure for low-complexity variable fractional delay FIR filters," *Circuits Syst. Signal Process.*, vol. 42, pp. 1105–1141, Feb. 2023.

[12] S. K. Mitra, *Digital Signal Processing: A Computer-Based Approach*, McGraw-Hill, 3rd edition, 2006.

[13] J. J. Shynk, "Frequency-domain and multirate adaptive filtering," *IEEE Signal Process. Magazine*, vol. 9, no. 1, pp. 14–37, Jan. 1992.

[14] K. Ishihara, R. Kudo, T. Kobayashi, A. Sano, Y. Takatori, T. Nakagawa, and Y. Miyamoto, "Frequency-domain equalization for coherent optical transmission systems," in *Proc. Optical Fiber Commun. Conf. and Exposition and the National Fiber Optic Engineers Conf.*, Mar. 6-10, 2011, pp. 1–3.

[15] A. Kovalev, O. Gustafsson, and M. Garrido, "Implementation approaches for 512-tap 60 gsa/s chromatic dispersion FIR filters," in *Proc. 51st Asilomar Conf. Signals, Systems, and Computers*, Oct. 29–Nov. 1, 2017.

[16] H. Johansson and O. Gustafsson, "On frequency-domain implementation of digital FIR filters using overlap-add and overlap-save techniques," Feb. 2023, doi: 10.48550/ARXIV.2302.08845.

[17] I. Slim, A. Mezghani, L. G. Baltar, J. Qi, F. N. Hauske, and J. A. Nossek, "Delayed single-tap frequency-domain chromatic-dispersion compensation," *IEEE Photon. Tech. Lett.*, vol. 25, no. 2, pp. 167–170, Jan. 2013.

[18] J. Nadal, F. Leduc-Primeau, C. A. Nour, and A. Baghdadi, "Overlap-save FBMC receivers," *IEEE Trans. Wireless Commun.*, vol. 19, no. 8, pp. 5307–5320, Aug. 2020.

[19] S. Ruiz, T. Dietzen, T. van Waterschoot, and M. Moonen, "A comparison between overlap-save and weighted overlap-add filter banks for multi-channel Wiener filter based noise reduction," in *Proc. 2021 29th Europ. Signal Process. Conf. (EUSIPCO)*, Aug. 23-27, 2021, pp. 336–340.

[20] H. Sorensen, D. Jones, M. Heideman, and C. Burrus, "Real-valued fast Fourier transform algorithms," *IEEE Trans. Acoustics, Speech, and Signal Process.*, vol. 35, no. 6, pp. 849–863, June 1987.

[21] C. S. Burrus, M. Frigo, Johnson G. S., M. Pueschel, and I. Selesnik, *Fast Fourier Transforms*, Samurai Media Limited, 2018.

[22] D. Takahashi, "An extended split-radix FFT algorithm," *IEEE Signal Process. Letters*, vol. 8, no. 5, pp. 145–147, 2001.

[23] W. J. Xu, Y. J. Yu, and H. Johansson, "Improved filter bank approach for the design of variable bandedge and fractional delay filters," *IEEE Trans. Circuits Syst. I: Reg. Pap.*, vol. 61, no. 3, pp. 764–777, Mar. 2014.