



## **Filter Implementation for Power-Efficient Chromatic Dispersion Compensation**

Downloaded from: <https://research.chalmers.se>, 2026-04-04 18:17 UTC

Citation for the original published paper (version of record):

Fougstedt, C., Sheikh, A., Johannisson, P. et al (2018). Filter Implementation for Power-Efficient Chromatic Dispersion Compensation. IEEE Photonics Journal, 10(4).  
<http://dx.doi.org/10.1109/JPHOT.2018.2846799>

N.B. When citing this work, cite the original published paper.

© 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, or reuse of any copyrighted component of this work in other works.

# Filter Implementation for Power-Efficient Chromatic Dispersion Compensation

Christoffer Fougstedt <sup>1</sup>, *Student Member, IEEE*, Alireza Sheikh <sup>2</sup>,  
Pontus Johannisson,<sup>3</sup>  
and Per Larsson-Edefors <sup>1</sup>, *Senior Member, IEEE*

<sup>1</sup>Department of Computer Science and Engineering, Chalmers University of Technology,  
SE-41296 Gothenburg, Sweden

<sup>2</sup>Department of Electrical Engineering, Chalmers University of Technology, 412 58  
Gothenburg, Sweden

<sup>3</sup>RISE Acreo, SE-40014 Gothenburg, Sweden

DOI:10.1109/JPHOT.2018.2846799

1943-0655 © 2018 IEEE. Translations and content mining are permitted for academic research only.

Personal use is also permitted, but republication/redistribution requires IEEE permission.

See [http://www.ieee.org/publications\\_standards/publications/rights/index.html](http://www.ieee.org/publications_standards/publications/rights/index.html) for more information.

Manuscript received May 8, 2018; revised June 7, 2018; accepted June 8, 2018. Date of publication June 13, 2018; date of current version July 13, 2018. This work was financially supported by the Knut and Alice Wallenberg Foundation. This paper was presented in part at the OSA Signal Processing in Photonic Communications, Boston, MA, USA, June 2015. Corresponding author: Christoffer Fougstedt (e-mail: chrfo@chalmers.se).

**Abstract:** Chromatic dispersion (CD) compensation in coherent fiber-optic systems represents a very significant DSP block in terms of power dissipation. Since spectrally efficient coherent systems are expected to find a wider deployment in systems shorter than long haul, it becomes relevant to investigate filter implementation aspects of CD compensation in the context of systems with low-to-moderate amounts of accumulated dispersion. The investigation we perform in this paper has an emphasis on implementation aspects such as power dissipation and area usage, it deals with both time-domain and frequency-domain CD compensations, and it considers both A/D-conversion quantization and fixed-point filter design aspects. To enable an accurate analysis on power dissipation and chip area, the evaluated filters are implemented in a 28-nm fully depleted silicon-on-insulator (FD-SOI) process technology. We show that an optimization of the filter response that takes pulse shaping into account can significantly reduce power dissipation and area usage of time-domain implementations, making them a viable alternative to frequency-domain implementations.

**Index Terms:** Optical fiber communication, digital signal processing, application specific integrated circuits, chromatic dispersion.

## 1. Introduction

Coherent fiber-optic communication systems have enabled effective receiver-side DSP-based compensation of linear impairments, but the very high throughput requirements result in high power dissipation in the receiver application-specific integrated circuit (ASIC). Along with dynamic equalization and forward error correction, chromatic dispersion (CD) compensation represents one of the most power dissipating functions in coherent receiver ASICs [1], [2]. Since traffic in metro networks is expected to grow significantly [3], an increase in deployment of spectrally-efficient coherent systems for relatively short transmission distances is expected [4]. Thus, energy-efficient CD compensation will not only be important for long-haul systems.

While dynamic equalization also can handle CD compensation, this equalization is in practice limited by the number of taps in the adaptive filter. Here, it is important to recognize that an

increasing tap count of an adaptive filter rapidly increases complexity both in filtering and dynamic coefficient update calculation, which in turn increases DSP power dissipation [5]. In view of the quadratic scaling with symbol rate, the possible reach of systems with only dynamic equalization is severely limited by dispersion. Thus, low-complexity CD compensation modules optimized to specific distances can yield a reduced overall power dissipation in systems with limited accumulated chromatic dispersion.

Finite impulse response (FIR) filters are often preferred over infinite impulse response (IIR) filters because of their inherent stability. In addition, since FIR filters are strictly feed-forward, FIR-based circuits can be implemented to reach arbitrarily high throughput through parallelization and pipelining. Published FIR filter design methods include direct sampling of the continuous-time inverse of the CD impulse response [6] and least-squares optimization of the discrete impulse response [7].

With a focus on CD compensation for fiber-optic systems with low-to-moderate levels of accumulated chromatic dispersion, this paper evaluates fixed-point filters and their power dissipation in terms of parallel real-time time-domain and frequency-domain ASIC circuit implementations. We will consider three different filter designs: The two mentioned above and the least-squares constrained-optimization (LS-CO) filter design method that we have proposed [8], [9]. While the LS-CO method was shown to have an improved fixed-point performance over other filter design methods, our evaluation [9] was limited to the MATLAB environment and to discussions on algorithmic complexity. In contrast, we here consider in detail implementation aspects such as quantization and finite wordlengths, parallelism for strict throughput requirements, and different filter circuit structures. In comparison to our previous work [10], we here supply an in-depth comparison of the implemented time-domain filters, both for fixed and adjustable coefficients, and comparisons to full overlap-save implementations with identical in-system performance requirements. In addition, the design flow is migrated to a 28-nm fully-depleted silicon-on-insulator (FD-SOI) process technology and we describe in detail the considered filter design methods, the implemented filter structures, the system-level considerations, and the system model.

Section 2 reviews the three different design methods used to construct filter coefficients over a number of filter taps, whereas Section 3 presents the three different filter implementation structures used to implement the fixed-point FIR filters. The system assumptions and the evaluation methodology are described in Section 4. Circuit implementation results in terms of area and power dissipation are presented in Section 5, which is followed by a discussion and a conclusion.

## 2. Filter Design Methods

We will here review the following filter design methods: direct sampling, least-squares optimization, and least-squares constrained optimization (LS-CO). The latter is the design method we have already proposed [8] and which is elaborated on in our later work [9].

### 2.1 Preliminaries

In a fiber system, chromatic dispersion (CD) can be ideally compensated by an all-pass static filter with a frequency response of

$$H_{\text{CD}}^{-1}(e^{j\omega T}) = e^{jM(\omega T)^2}, \quad M = \frac{D\lambda^2 L}{4\pi c T^2}, \quad (1)$$

where  $D$  is the dispersion parameter ( $D = 16$  ps/(nm km) in this work),  $T$  is the sampling interval,  $L$  is the fiber length,  $\lambda$  is the transmitting laser's wavelength ( $\lambda = 1550$  nm in this work),  $c$  denotes the speed of light, and  $\omega T \in [-\pi, \pi]$  is the digital frequency.

To reduce inter-symbol interference over the fiber, the signal bandwidth is routinely reduced by way of pulse shaping of the transmitted pulse, e.g., by using root-raised-cosine (RRC) shaping. While the full RRC frequency response can be found in [9], we here reiterate the limited signal bandwidth:  $\frac{1-\beta}{G}\pi \leq |\omega T| \leq \frac{1+\beta}{G}\pi$ . Here,  $G$  denotes samples per symbol ( $G = 2$  in this work), whereas  $\beta$  is the roll-off factor ( $\beta = 0.25$  in this work). The in-band frequency range thus becomes  $[-0.625\pi, 0.625\pi]$  in this work, and the frequencies outside of this range are consequently out of band.

## 2.2 Direct Sampling [6]

By taking the inverse Fourier transform of (1), the ideal CD compensation impulse response can be derived. Savory uses direct sampling (DS) and truncation of the continuous CD compensation filter impulse response to derive the filter taps [6]:

$$h_n^{\text{DS}} = \sqrt{\frac{j}{4M\pi}} e^{-j\frac{n^2}{4M}} \quad (2a)$$

$$-\frac{N_{\text{max}} - 1}{2} \leq n \leq \frac{N_{\text{max}} - 1}{2}, \quad (2b)$$

where  $N_{\text{max}}$  ensures that aliasing does not occur.

For the DS method, however, the possible discrete impulse response length is limited and misbehaves for low levels of accumulated dispersion. Using simulations, we can confirm that the DS design method has a significant power penalty<sup>1</sup> when compensating for limited chromatic dispersion (<320 km single-mode fiber, given the setup in Section 4.1) [11], so this filter design will not be part of our evaluation.

## 2.3 Least-Squares Optimization [7]

Eghbali *et al.* proposed using least-squares optimization to minimize the deviation between the filter response and the ideal impulse response [7]. This optimization can be performed either on the full band of the signal or on a part of the band, utilizing that pulse shaping is employed. The latter band-limited optimization method, however, requires an adjustment factor to prevent numerical issues in the filter coefficient calculation [7]. The disadvantage is that this adjustment factor, which has to be arbitrarily set, turns out to have a significant impact on the out-of-band gain and the fixed-point performance. Since this work, in contrast to our previous paper [9], is considering an evaluation of fixed-point hardware, we need to avoid the arbitrary selection of the adjustment factor. Thus, in the following, we will consider least-squares optimization for only the full-band case and we call this method LS-FB.

## 2.4 Least-Squares Constrained Optimization [9]

Assume now a filter is implemented for a limited band. Here the designer has an obvious option to reduce hardware; use fewer filter taps than for the full band. However, a reduction in filter taps translates in an increasing out-of-band gain which in turn increases the quantization errors. The key idea behind our least-squares constrained optimization (LS-CO) [9] was to come up with a method that allowed us to put a constraint on the out-of-band gain to enable practical filter optimization for band-limited signals.

We can define the normalized average filter response error within a certain signal band, for which  $\Omega_1$  and  $\Omega_2$  are boundary frequencies, to assess how well a band-limited filter's frequency response matches the ideal CD compensation filter:

$$\xi_s = \frac{1}{\Omega_2 - \Omega_1} \int_{\Omega_1}^{\Omega_2} |H(e^{j\omega T}) - H_{\text{CD}}^{-1}(e^{j\omega T})|^2 d(\omega T). \quad (3)$$

Another important frequency response property has to do with the out-of-band frequencies. The normalized average out-of-band gain of the filter can be defined as

$$\xi_o = \frac{1}{2\pi + \Omega_1 - \Omega_2} \cdot \left[ \int_{-\pi}^{\Omega_1} |H(e^{j\omega T})|^2 d(\omega T) + \int_{\Omega_2}^{\pi} |H(e^{j\omega T})|^2 d(\omega T) \right]. \quad (4)$$

<sup>1</sup>Here, power penalty is defined as the increase in signal power to achieve the same bit-error rate (BER) performance as that of an ideal system.

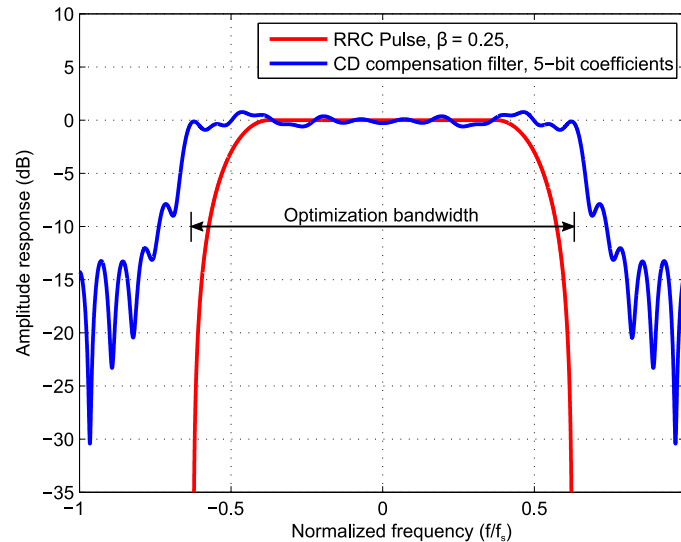


Fig. 1. Amplitude response of a quantized LS-CO filter for 100 km of fiber.

The main feature of the LS-CO design method is to make the in-band filter response as similar as possible to ideal compensation (i.e., minimize  $\xi_s$ ), while confining the out-of-band gain (i.e., limit  $\xi_o$ ). This method allows for a trade-off between the in-band error and the error due to the out-of-band gain when quantizing the coefficients. Fig. 1 shows the amplitude response of a 5-bit quantized LS-CO filter assuming an input RRC pulse spectrum ( $\beta = 0.25$ ). For details on the LS-CO method, we refer the reader to the complete description [9].

### 3. FIR Filter Implementation Structures

Since DSP-based coherent fiber optic systems have very high throughput requirements, extensive parallelization is necessary. The following section first reviews the used filter structures; here, the assumed 64-way parallelization makes for a good trade-off with the default clock frequency of 875 MHz for the assumed 28-GBd (56-Gsample/s) system requirement. Finally in this section, we will address fixed-point implementation aspects for the different filter structures.

#### 3.1 Parallel Polyphase FIR

Parallel FIR structures can be implemented by performing polyphase decomposition, exposing the input-output dependence of each parallel filter lane, to generate a block-processing formulation of the FIR filter [12]. An  $L$ -parallel filter is implemented by decomposing the impulse response into  $L$  sub-filter impulse responses, each containing every  $L$ -th component of the CD compensation impulse response, starting at the corresponding sub-filter index,  $H_0-H_L$ . The general principle is shown in Fig. 2, and can be represented in matrix form as [12]

$$\begin{bmatrix} Y_0 \\ Y_1 \\ \vdots \\ Y_{L-1} \end{bmatrix} = \begin{bmatrix} H_0 & z^{-L}H_{L-1} & \cdots & z^{-L}H_1 \\ H_1 & H_0 & \cdots & z^{-L}H_2 \\ \vdots & \vdots & \ddots & \vdots \\ H_{L-1} & H_{L-2} & \cdots & H_0 \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \\ \vdots \\ X_{L-1} \end{bmatrix} \quad (5)$$

where multiplication represents filtering operations,  $X$  and  $Y$  represent each input and output sample, respectively, in the current block, and  $z^{-L}$  represents a delay of  $L$ . Note that each column

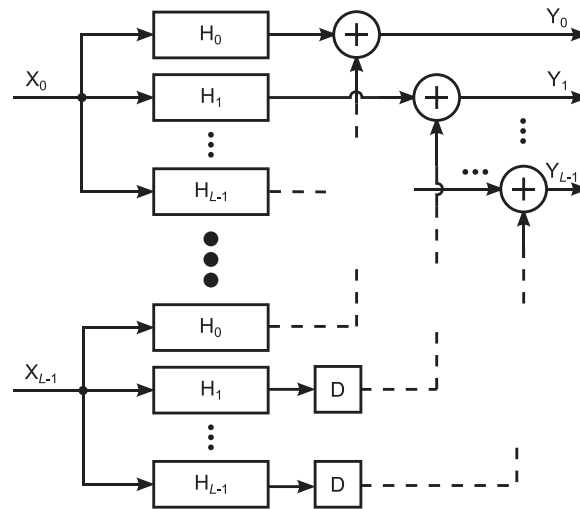


Fig. 2. Polyphase FIR structure where delay registers are denoted D [10].

except the leftmost is a one-step circularly downwards-shifted version of the left-adjacent column, with circularly-wrapped elements multiplied by  $z^{-L}$ .

The filter complexity increases linearly with parallelization as an  $N$ -tap filter is decomposed into  $L$  sub-filters containing every  $L$ -th tap; each parallel lane requires  $L$  filters of length  $N/L$ . Polyphase parallel filters, thus, yield the same computational complexity per output sample as standard serial input-output FIR filter implementations, while increasing the possible throughput.

The internal wordlengths are similar to standard FIR filters; filter multiplier input widths are the same as the input and coefficient wordlength for the respective inputs, and the wordlength of the arithmetic is set to prevent input-signal dependent overflow. The same adder-tree wordlengths are used for all compensation lengths.

### 3.2 Fast-FIR

Parallelization of FIR filters allows for exploitation of common sub-expressions to reduce the multiplicative complexity of the filters through the use of Fast-FIR algorithms [12], which are based on efficient algorithms for polynomial multiplication applied to the polynomial formulation of FIR filters. Iterated application of Fast-FIR algorithms can be used to build filter structures with wider parallelism. Iterated application and delay element unfolding for the implementation of  $2^N$ -parallel filters is illustrated in Fig. 3.

In this work, a 2-parallel Fast-FIR algorithm is applied 6 times to obtain a 64-parallel filter structure. The basic 2-parallel Fast-FIR algorithm used here can be represented in matrix form as [12]

$$\begin{bmatrix} Y_0 \\ Y_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & z^{-2} \\ -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} H_0 & 0 & 0 \\ 0 & H_0 + H_1 & 0 \\ 0 & 0 & H_1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \end{bmatrix}, \quad (6)$$

which can be written as

$$\mathbf{Y}_2 = \mathbf{Q}_2 \mathbf{F}_2 \mathbf{P}_2 \mathbf{X}_2, \quad (7)$$

where  $H_i$  represents the sub-filter impulse response, and multiplication represents filtering operations. In this case,  $H_0$  and  $H_1$  contains every second component of the impulse response, starting at the respective index. Each sub-filter response is therefore half the length of the initial filter response.

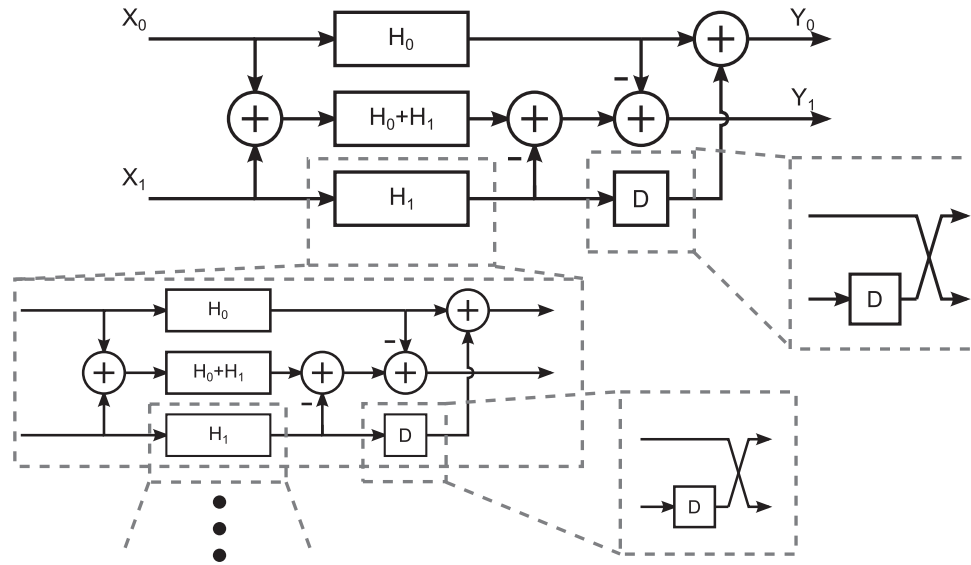


Fig. 3. Iterated application of the Fast-FIR algorithm [10].

An iterative application of the filters requires delay unfolding and parallelization, which is performed by replacing each delay element in the matrix as

$$Y = z^{-2}X \rightarrow \begin{bmatrix} Y_0 \\ Y_1 \end{bmatrix} = \begin{bmatrix} 0 & z^{-4} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} X_0 \\ X_1 \end{bmatrix}, \quad (8)$$

and by replacing each 1,  $-1$ , and 0 with  $\mathbf{I}_{2 \times 2}$ ,  $-\mathbf{I}_{2 \times 2}$ , and  $\mathbf{0}_{2 \times 2}$ , respectively. Applying the unfolding procedure once to  $\mathbf{Q}_2$  thus yields

$$\mathbf{Q}_{2_u} = \text{unfold}(\mathbf{Q}_2) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & z^{-4} \\ 0 & 1 & 0 & 0 & 1 & 0 \\ -1 & 0 & 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 & 0 & -1 \end{bmatrix}. \quad (9)$$

The order of the parallel-filter inputs is established by applying bit-reversal permutation [13], i.e., the order obtained by reversing the binary representation of each index, to the  $2^N$  long input vector.

The pre-processing matrix is generated by taking the Kronecker product [14] of the pre-processing matrix to itself  $N - 1$  times as

$$\mathbf{P}_N = \overbrace{\mathbf{P}_2 \otimes \cdots \otimes \mathbf{P}_2 \otimes \mathbf{P}_2}^{N-1 \text{ times}}. \quad (10)$$

The diagonal filtering matrix is generated as

$$\mathbf{F}_N = \text{diag}(\mathbf{P}_N \mathbf{H}_{Np}), \quad (11)$$

where  $\mathbf{H}_{Np}$  is the sub-filter vector permuted in the same order as the input vector. The  $N$  post-processing matrices are generated by unfolding the previous matrix as

$$\mathbf{q}_i = \text{unfold}(\mathbf{q}_{i-1}), \quad (12)$$

starting with

$$\mathbf{q}_0 = \begin{bmatrix} 1 & 0 & z^{-2} \\ -1 & 1 & -1 \end{bmatrix}, \quad (13)$$

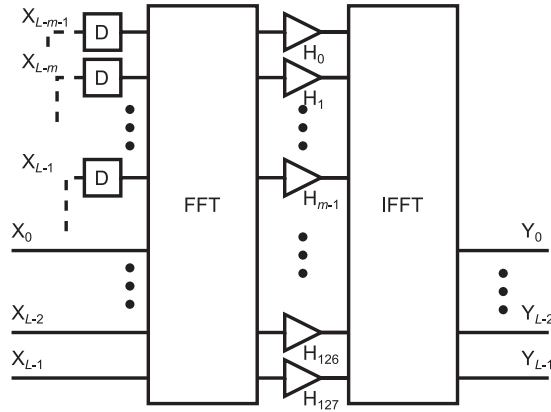


Fig. 4. Block diagram for the overlap-save structure.

and finally parallelizing the matrices as

$$\mathbf{Q}_{P_i} = \mathbf{q}_i \otimes \overbrace{\mathbf{I}_{3 \times 3} \otimes \mathbf{I}_{3 \times 3} \cdots \otimes \mathbf{I}_{3 \times 3}}^{N-i-1 \text{ times}}. \quad (14)$$

The resulting  $N$ -parallel structure can be written as

$$\mathbf{Y}_N = \mathbf{Q}_{P_{N-1}} \mathbf{Q}_{P_{N-2}} \cdots \mathbf{Q}_{P_{N_0}} \mathbf{F}_N \mathbf{P}_N \mathbf{X}_N. \quad (15)$$

Fast-FIR filters reduce multiplicative complexity at the expense of introducing pre- and post-processing addition trees. Multiplier wordlength is also increased as summation of both coefficients and inputs is performed, thus increasing the wordlength requirement on both the sample and coefficient input of the filter multipliers.

### 3.3 Overlap Save

FIR filtering can be performed in the frequency domain using the overlap-save (OS) algorithm to perform linear convolution using circular convolution, which is efficiently implemented using FFT and IFFT. Each block of input samples is concatenated with the previous block, Fourier transformed, element-wise multiplied with the transformed zero-padded impulse response, and transformed back. Samples affected by artifacts due to circular convolution are discarded. The  $L$ -parallel OS implementation can be represented as

$$\begin{bmatrix} O_0 \\ \vdots \\ O_{m-1} \\ Y_0 \\ \vdots \\ Y_{L-1} \end{bmatrix} = \mathcal{F}^{-1} \left( \hat{\mathbf{H}} \odot \mathcal{F} \left( \begin{bmatrix} z^{-L} X_{L-m-1} \\ \vdots \\ z^{-L} X_{L-1} \\ X_0 \\ \vdots \\ X_{L-1} \end{bmatrix} \right) \right) \quad (16)$$

where  $\odot$  denotes element-wise multiplication of the vectors,  $\hat{\mathbf{H}}$  is the frequency response vector, and  $O_0$ – $O_{m-1}$  are the  $m$  discarded samples. The required number of discarded samples and samples from the previous block are the same as the length of the impulse response, and  $L + m$  must be equal to the FFT and IFFT length. Fixed-point implementation of overlap-save filters entails limited resolution in both coefficient multiplication and the FFT/IFFT-pair. In the interest of reproducibility, and as the focus of this paper is on filter implementation, the Spiral tool [15] has been used to generate the FFT/IFFT-pair. Fig. 4 shows a block diagram of the implemented overlap-save structure.

The power dissipation for an OS structure is dependent on the required overlap and subject to implementation considerations, since the block-processing width is set by the FFT length and overlap factor. Here, we will use a 128-point FFT/IFFT-pair, to build a 50%-overlap 64-parallel structure, a 25%-overlap 96-parallel structure, and a 12.5%-overlap 112-parallel structure. The 96- and 112-parallel structures require either a change of parallelism in other sub-systems or block-width conversion; the latter increases area and power dissipation.

### 3.4 Fixed-Point Filter Aspects

The BER performance of the time-domain filters will be evaluated using MATLAB implementations with quantized coefficients and inputs. This yields equivalent results to time-domain hardware description language (HDL) implementations as, first, no internal rounding is performed and, second, internal wordlength is set to avoid overflow. The polyphase (Section 3.1) and Fast-FIR (Section 3.2) filters are bit-equivalent<sup>2</sup> and, thus, have identical BER performance. The OS implementation (Section 3.3), however, is evaluated by simulating the implemented HDL code (Section 4.3), since internal truncation and twiddle-factor rounding are performed in the generated FFT and IFFT blocks.

**3.4.1 Wordlength Requirements:** The shortest possible wordlength for fulfilling the power-penalty requirement of each filter design is determined by sweeping the coefficient wordlength. The power penalty of the OS implementation is significantly impacted by the resolution in the FFT and IFFT, which are the most complex and power-dissipating blocks. The resolution requirements for the OS configuration are therefore determined by finding the minimal-resolution FFT/IFFT-pair that fulfills the power-penalty requirement.

The filters evaluated are implemented to achieve a power penalty of at most 0.25 dB, including the A/D conversion quantization penalty (see Section 4.2). In the time-domain implementations, the LS-FB filter fulfills this requirement using 5- and 6-bit coefficients for PM-QPSK and PM-16-QAM, respectively, while the LS-CO filter allows for a reduction of one bit in coefficient wordlength. The wordlength requirements are different for the FFT/IFFT-pair compared to general FIR filtering as twiddle-factor rounding and internal scaling are required in a fixed-point implementation.

Spiral [15], which is available online [16], is here used to generate fully streaming fixed-point FFT and IFFT HDL implementations with internal scaling to prevent overflow. Spiral is configured to generate 128-point radix-4 [13] FFTs/IFFTs, yielding transforms consisting of radix-4 stages and one radix-2 stage. The data are left-shifted to the most significant bits of the FFT, after which stepwise internal truncation is performed after each twiddle factor multiplication to reduce the impact of twiddle-factor rounding and prevent internal overflow in the FFT/IFFT-pair. A 12- and 13-bit wordlength is required in the FFT/IFFT-pair for PM-QPSK and PM-16-QAM, respectively. Element-wise multiplication with the frequency-domain response is performed, and truncation is performed to prevent overflow. The truncation factor is tested empirically to utilize the IFFT dynamic range while reducing the risk of overflow. Overly optimistic scaling would have resulted in overflow that depends on the temporal spectral density; some data patterns cause peaks in the spectrum, which can result in burst errors. The power-penalty goal of  $\leq 0.25$  dB is met with 5- and 6-bit frequency-response multipliers when using LS-CO, given the required FFT/IFFT wordlength.

**3.4.2 Required Number of Taps:** Once the coefficients are fixed, the required number of taps can be determined by sweeping this parameter to find the lowest number fulfilling the power-penalty requirement. Table 1 shows the required number of taps given the system parameters listed in Section 4.1.

The benefit of reducing wordlength and tap count is diminishing in the case of an OS implementation, since the wordlength in the FFT and IFFT is unaffected by the filter design and since the number of taps is set by implementation considerations. Therefore, the LS-FB variant of the OS structure is not implemented as this differs from LS-CO only for the 128 frequency-response multipliers. It is, however, important to note that the reach for any FFT/IFFT-pair with a fixed number of points is increased due to the relaxed tap requirement of LS-CO as compared to LS-FB.

<sup>2</sup>Given an arbitrary input sequence, two bit-equivalent implementations will yield the same output sequence.

TABLE 1  
Required Number of Taps for  $\leq 0.25$  dB Power Penalty

Dist. (km)	PM-QPSK		PM-16-QAM	
	LS-CO	LS-FB	LS-CO	LS-FB
33.5	13	19	13	21
50	15	25	17	29
75	21	33	25	41
100	29	45	33	51
150	39	63	49	73

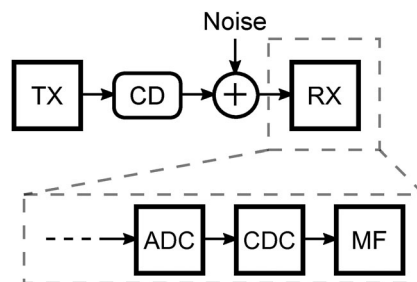


Fig. 5. Block diagram of the system model. The receiver consists of ADC, CD compensation (CDC), and matched filtering (MF). The order in which CD and Noise are applied is arbitrary as CD does not affect the noise distribution.

## 4. Evaluation Setup

Implementation of real-time coherent receivers imposes stringent requirements on both power dissipation and throughput, and fixed-point implementation is thus necessary. The main evaluation metrics of the implemented filters are BER performance (quantified in terms of power penalty at a BER of  $10^{-3}$ ) together with area usage and power dissipation of the circuits. To make the evaluation effective and accurate, we have to mix MATLAB and ASIC circuit implementation evaluations: While the power penalty from A/D conversion (ADC) can be evaluated by performing simulations in MATLAB (Section 4.2), different electronic design automation software suites for ASICs are required for circuit implementation and accurate estimation of circuit performance, power dissipation and area (Section 4.3).

In the ensuing evaluations, we consider CD compensation filter implementations in the context of the coherent fiber system model in Section 4.1. An overriding constraint that is used for all filter evaluations is that the resulting power penalty, when evaluated in the system model, is at most 0.25 dB. This constraint is a result of ADC quantization considerations described in Section 4.2. Finally, the ASIC circuit implementation and verification flow used to obtain power dissipation estimates is presented in Section 4.3.

### 4.1 System Model

The fixed-point filter implementations are evaluated using a MATLAB-based coherent fiber-optic system model that assumes two samples-per-symbol at 28 GBd, polarization-multiplexed quadrature phase-shift keying (PM-QPSK) or polarization-multiplexed 16-quadrature amplitude modulation (PM-16-QAM), a CD parameter of  $D = 16$  ps/(nm km), at the wavelength  $\lambda = 1550$  nm, and RRC pulses with a roll-off factor  $\beta = 0.25$ . The transmitter is assumed to be ideal. The channel is modeled as an additive white Gaussian noise channel with chromatic dispersion. Fig. 5 shows a block diagram of the simulation setup.

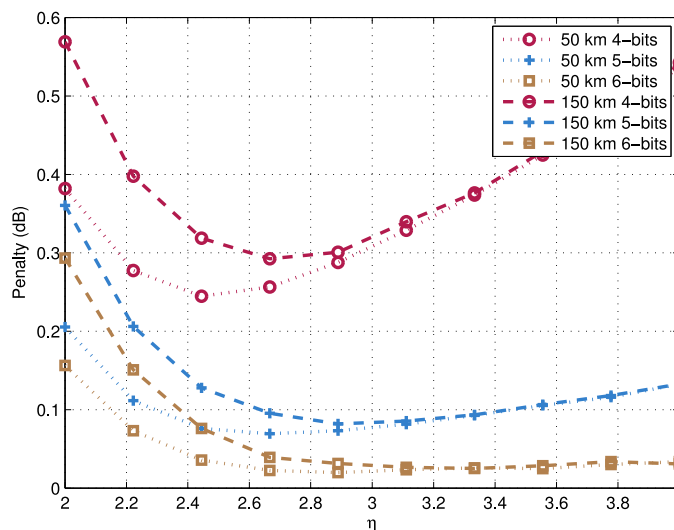


Fig. 6. Power penalty as a function of  $\eta$  for different ADC resolutions and fiber system lengths, using PM-QPSK.

#### 4.2 A/D-Conversion Considerations

Since fixed-point aspects are at the core of this work, ADC assumptions become important. ADC quantization introduces quantizer resolution and signal-dependent noise, which result in a power penalty. The resulting input signal distribution tends to become Gaussian-like due to the inter-symbol interference (ISI) caused by the chromatic dispersion. This needs to be taken into account when selecting the quantization range.

We investigated the impact of ADC quantization by simulating the system in Fig. 5 with chromatic dispersion, ADC quantization, and ideal penalty-free floating-point CD compensation. This allowed us to isolate the impact of linear-step quantization on a realistic input signal distribution. An ADC resolution of 4–6 bits and 5–7 bits was considered for PM-QPSK and PM-16-QAM, respectively. The signal was represented in signed binary-fraction form; the sign bit as the most significant bit and the others representing the binary fraction. The input signal was multiplied with a pre-quantization signal scaling factor,  $S = 1/(\eta\sigma)$ , where  $\sigma$  is the standard deviation of the input distribution of each ADC and  $\eta$  was set to adjust the quantization range.

Fig. 6 and Fig. 7 show the simulated power penalty as a function of  $\eta$  for different quantizer resolutions, for PM-QPSK and PM-16-QAM, respectively. For PM-16-QAM to attain similar performance to PM-QPSK, an increase of 1 bit in resolution is required. In summary, 5- and 6-bit ADC resolution is required to fulfill the  $\leq 0.25$  dB power penalty constraint for PM-QPSK and PM-16-QAM, respectively.

#### 4.3 Circuit Implementation Flow

Since this work considers highly complex FIR structures, we created a customized MATLAB-based tool to generate HDL descriptions from matrix-based filter structure specifications. The tool calculates the pre- and post-processing matrices, which are translated to a behavioral VHDL description, connects the sub-filter components, and calculates the corresponding coefficients. The generated code is structured for further optimization by the synthesis tool. The generator was verified by simulating generated polyphase and Fast-FIR HDL implementations in the system model (Fig. 5), and by confirming bit-equivalent outputs in pre- and post-synthesis simulation. Pre- and post-processing trees, sub-filter implementation and wrapper files for synthesis were generated based on the specified filter structure, parallelization width and coefficients. Pipelining was implemented to reduce critical-path length and reduce adder-tree switching activity. To obtain power-efficient implementa-

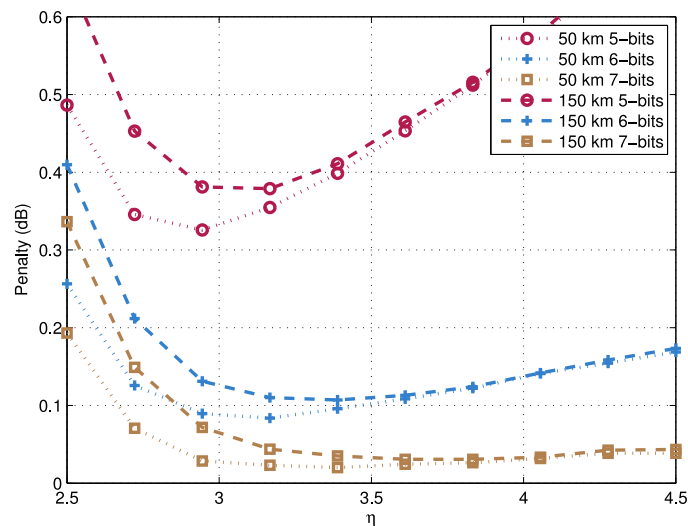


Fig. 7. Power penalty as a function of  $\eta$  for different ADC resolutions and fiber system lengths, using PM-16-QAM.

tions, extensive parallelization was necessary. Here, 56 Gsample/s requires a 64-way parallelization at a clock rate of 875 MHz.

The FFT/IFFT-pair for the OS structure (Section 3.3) was generated using Spiral [15]. The structure is similar to the CD compensation used in [1], with shorter FFT/IFFT and hard-wired twiddle factor multipliers being the main differences. Zero-padding of the impulse response was used when the impulse response was shorter than 64, 32 or 16 for 50%, 25% and 12.5% overlap, respectively. The 64-parallel implementation can run at a default clock rate of 875 MHz, while the 96- and 112-parallel variants only require a clock rate of 583 and 500 MHz, respectively.

The filters were synthesized using Cadence Encounter RTL Compiler [17] and a 28-nm 1.0-V fully-depleted silicon-on-insulator (FD-SOI) process technology, with standard threshold voltage cell libraries characterized at slow-slow corner and 125 °C, in order to ensure reasonable results taking manufacturing yield into account. This should be contrasted to our previous work which was based on a 65-nm design flow [10]. The tool's retiming feature was used to optimize the pipelining register distribution in the implementations. No carry-save optimization was used, as the quality of results of the automatic optimization in RTL Compiler depends on code structure and register placement. Furthermore, the physical layout estimation feature of RTL Compiler was used to give more accurate wire-load estimations.

Once the filters were synthesized, the netlists were simulated with an input signal affected by chromatic dispersion in order to log the switching activity of each circuit net. The switching activity obtained was back-annotated to the netlist and power estimation was performed using RTL Compiler for the typical 28-nm cell library (nominal corner and 25 °C). Since switching power represents 99% of the total power, the power dissipation's dependence on temperature is negligible.

#### 4.4 Polyphase vs Fast-FIR for Different Tap Counts

Implementation of algorithms requires a significant engineering effort. In DSP research, a common approach to approximate hardware parameters like area and power dissipation, is to evaluate algorithms from a high-level complexity perspective, often in terms of required number of multiplications. (In fact, to the best of our knowledge, no circuit-level evaluation has been published on highly-parallel FIR filters.) However, complexity-based assessment of algorithms often fail to properly capture many factors that contribute to overall power dissipation, such as circuit structure, pipelining, switching activity, and resolution.

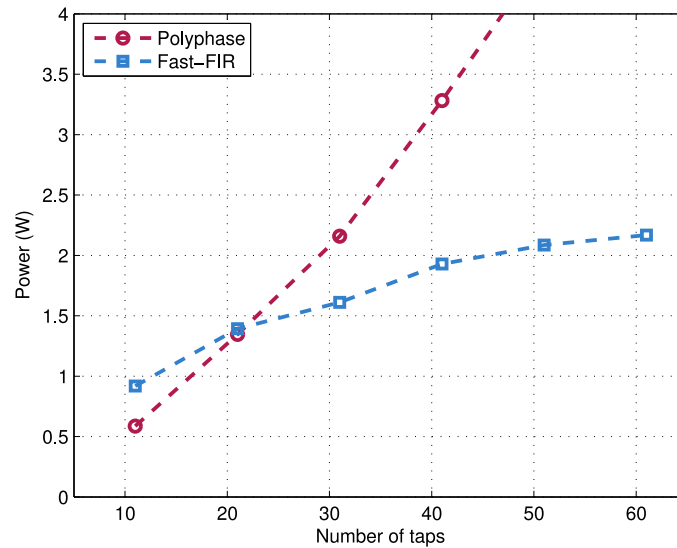


Fig. 8. Power dissipation of the parallel random-coefficient filters.

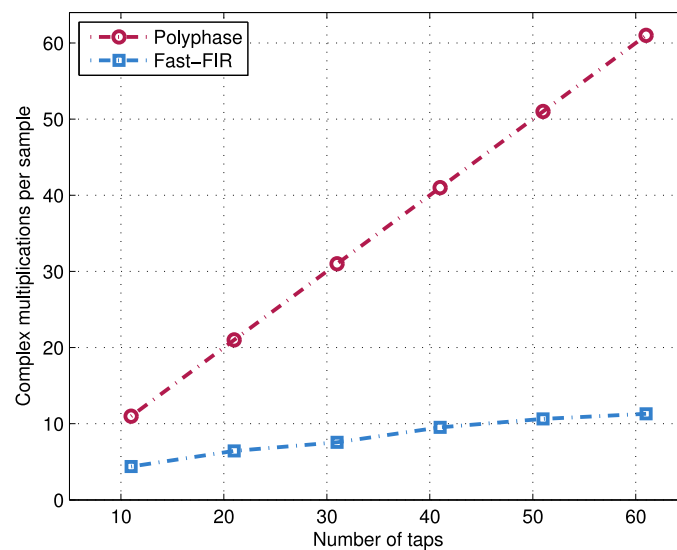


Fig. 9. Complexity #multiplications of the parallel filter implementations.

Prior to the evaluation of filters deployed in fiber systems of different lengths, we here present power dissipation for the more general case when parallel polyphase FIR and Fast-FIR filters are evaluated as function of tap count. Fig. 8 shows the power dissipation (obtained as described in Section 4.3) of 64-parallel filter implementations with uniform random complex coefficients and input data. Clearly the Fast-FIR structure scales better with tap count than the polyphase structure does. While power dissipation and complexity correlate relatively well when comparing two different implementations of the same structure, Fig. 9 that shows the corresponding complexity metric based on number of multiplications per filter illustrates that complexity-based comparison of *different* structures can be misleading.

## 5. Implementation Results

While adjustable filter responses are desirable in communication systems, any kind of system flexibility brings a significant overhead in terms of ASIC area and power. In a fixed-coefficient filter, where multiplications with constant numbers are carried out, a gate-level implementation based on shifts and additions is possible. In comparison to an adjustable-coefficient filter, where multiplications with variable numbers are required, the fixed-coefficient filter's circuit complexity is lower, thus, reducing power dissipation and area usage.

Fixed-coefficient filters offer an important trade-off in the design of a practical system, especially in coherent fiber optic communication where there are very stringent requirements on throughput and power dissipation. For example, in an implementation of a cascaded filter, a mix of fixed- and adjustable-coefficient filter can provide a trade-off between power dissipation and flexibility.

In this work, both adjustable- and fixed-coefficient filters are implemented and evaluated. This is in contrast to our previous contribution [10] where only the latter types were considered. As far as design flow (Section 4.3), we employ a method where we first synthesize the filters with full complex multipliers for each tap to yield adjustable coefficients. The resulting netlist is then incrementally resynthesized with boundary constant propagation to let RTL Compiler reduce the multipliers to shift-and-add trees in order to yield fixed-coefficient filters with the same overall structure as the adjustable filters.

All implemented filters use single-tap sub-filters, and can thus support a tap number of up to the chosen parallelization factor (64 taps in this work). The longest LS-FB filter (73 taps for PM-16-QAM) has thus been excluded from the evaluation.

It should be noted that the area and power dissipation ratios between adjustable- and fixed-coefficient filters depend on the system timing constraint. Since they use multipliers with higher circuit complexity, stricter timing constraints affect adjustable-coefficient filter circuits more than their fixed-coefficient counterparts.

### 5.1 Adjustable-Coefficient Filters

The LS-CO filter gives a significant power dissipation reduction in comparison to the LS-FB filter due to a reduction in both the required wordlength and the number of required taps. The Fast-FIR filter scales significantly better with tap count compared to the polyphase filter, but the pre- and post-processing trees along with increased internal wordlength requirements give a high overhead for short fiber lengths.

The resulting power dissipation *per polarization channel* is shown in Fig. 10 and Fig. 11, for PM-QPSK and PM-16-QAM, respectively, with the corresponding number of taps shown in Table 1. The reduction in power dissipation for implementing an LS-CO filter is more significant for the polyphase implementation as compared to Fast-FIR; the crossover point between the implementations is shifted significantly towards longer transmission distances.

The power dissipation of the OS implementation is dominated by the FFT and IFFT; the FFT/IFFT-pair dissipates over 85% of the total power. The maximum number of taps is limited by the FFT length and overlap factor, and the LS-CO filter does not give a significant power-dissipation reduction in the case of a frequency-domain implementation if it is not possible to change the overlap. The reach for a given FFT length is increased as the required number of taps is reduced, allowing for use of an FFT with fewer points or lower overlap for the same reach.

Fig. 12 and Fig. 13 show the total area of the cells needed to implement the adjustable-coefficient QPSK and 16-QAM filters, respectively. The time-domain implementations give a significant area reduction in comparison to the frequency-domain implementations, since the same FFT/IFFT pair is required in all FD implementations with overlap and clock rate changed. Both Fast-FIR and polyphase implementations give a significant reduction in area usage compared to the overlap-save implementation.

As shown in Fig. 12 and Fig. 13, the area for the filters with 64-parallel 50% overlap and 96-parallel 25% overlap are very similar since the same FFT/IFFT-pair is used in both cases (minor variations are likely caused by the heuristic synthesis algorithms). The power dissipation of the OS implementation is however significantly reduced as the clock rate used for the 96- and the

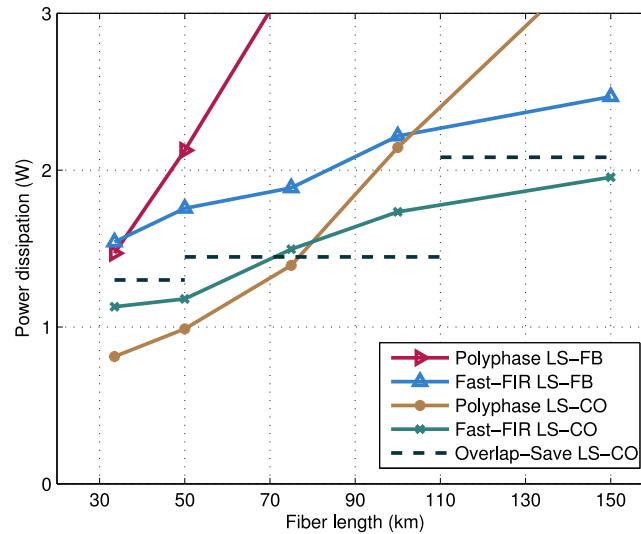


Fig. 10. Power dissipation of adjustable-coefficient PM-QPSK filters.

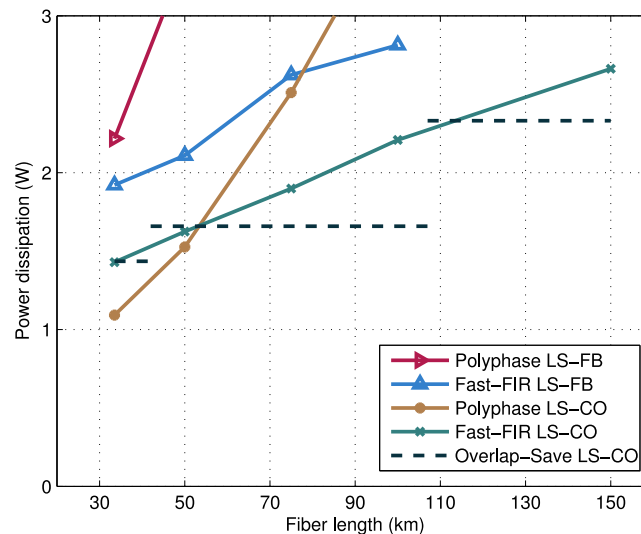


Fig. 11. Power dissipation of adjustable-coefficient PM-16-QAM filters.

112-parallel implementation is  $2/3$  and  $4/7$ , respectively, of the 64-parallel implementation. Overall, LS-CO significantly reduces the area needed to implement time-domain filters; a time-domain filter for compensation of 50 km of dispersive propagation requires less than 50% area for PM-QPSK and approximately 50% area for PM-16-QAM in comparison to a frequency-domain implementation, thus making it viable to include fine-tuned filters for low-power operation in coherent systems with a limited amount of chromatic dispersion.

## 5.2 Fixed-Coefficient Filters

Fig. 14 and Fig. 15 show the power dissipation of the fixed-coefficient implementations. Comparing the filters to the case of adjustable coefficients, time-domain implementations give a further reduction in power dissipation compared to the fixed-coefficient frequency-domain implementations, since a higher proportion of multipliers are reduced to shift-and-add units. Compared to

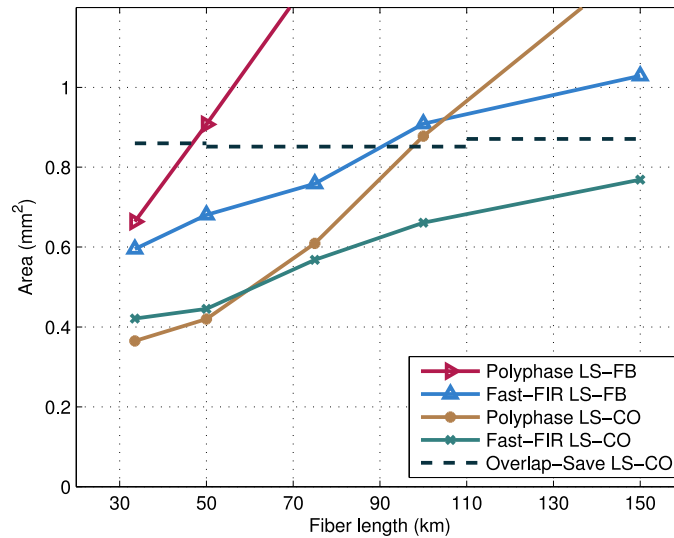


Fig. 12. Area usage of adjustable-coefficient PM-QPSK filters.

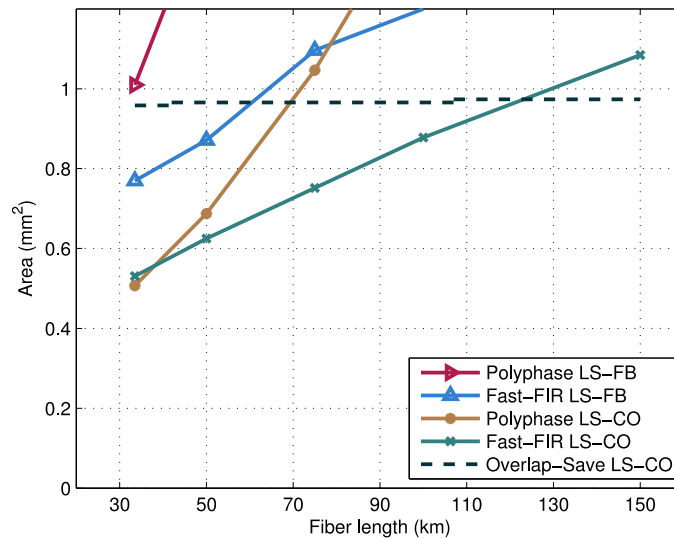


Fig. 13. Area usage of adjustable-coefficient PM-16-QAM filters.

Fast-FIR, the polyphase implementations give an additional improvement for the short transmission distances, since the pre- and post-processing trees in the Fast-FIR filters are not changed.

Fig. 16 and Fig. 17 show the total area of the cells needed to implement the fixed-coefficient QPSK and 16-QAM filters, respectively. The Fast-FIR filters occupy less area than polyphase filters for the longer distances; for shorter distances, the area needs are similar. In comparison to the adjustable-coefficient case, the polyphase filters are slightly more area efficient than Fast-FIR. A significant reduction in area usage is possible, in comparison to the frequency-domain implementation, since the change in overlap factor in the frequency-domain case only allows for a reduction in power dissipation due to a reduced clock rate. The time-domain implementations allow for both area reduction and power-dissipation reduction when reducing the number of taps.

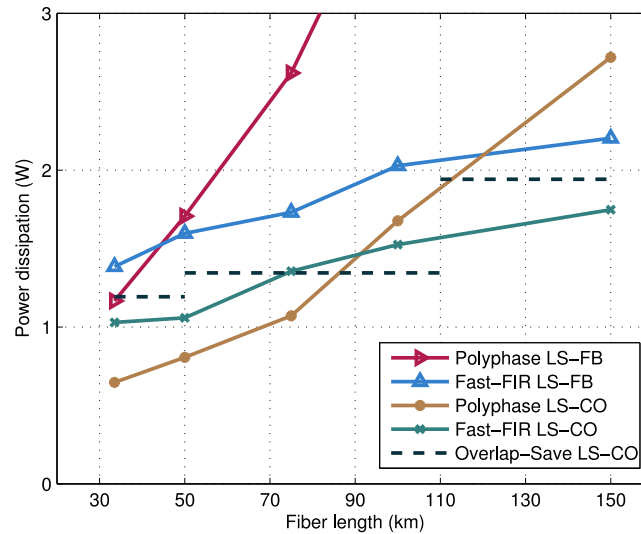


Fig. 14. Power dissipation of fixed-coefficient PM-QPSK filters.

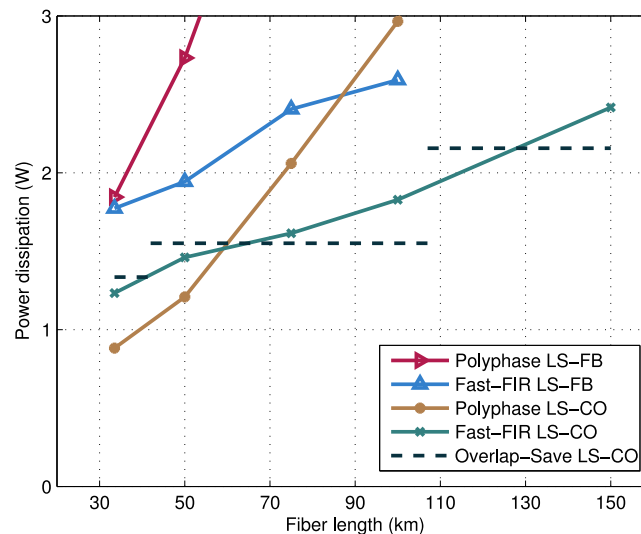


Fig. 15. Power dissipation of fixed-coefficient PM-16-QAM filters.

## 6. Discussion

Although a time-domain implementation of CD compensation for short transmission links can give power dissipation advantages over their frequency-domain counterparts, the greatest gain is in terms of area. An overall area reduction is possible for PM-QPSK links in all cases up to the considered maximum transmission distance of 150 km, in comparison to the reference frequency-domain implementation. While it is possible to reduce power dissipation for the considered transmission distances, this reduction is more pronounced for PM-QPSK than for PM-16-QAM, since the latter requires larger wordlengths. Large reductions in area usage are possible in all but the longest of the considered PM-16-QAM transmission distances, for the adjustable-coefficient filter.

Time-domain implementations allow fine tuning of the filter to the actual transmission distance. Consider, for example, fiber non-linearity mitigation: A recent DSP implementation technique (TD-DBP) for digital back propagation is based on a distributed, time-domain approach [18]. In TD-DBP, a large number of shorter CD compensation steps are cascaded, making each filter section

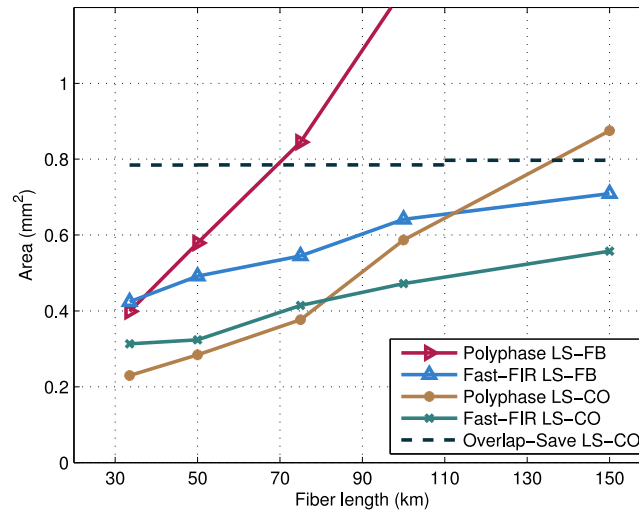


Fig. 16. Area usage of fixed-coefficient PM-QPSK filters.

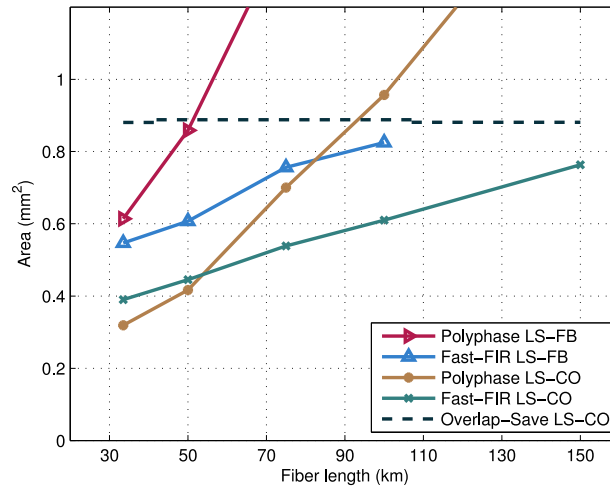


Fig. 17. Area usage of fixed-coefficient PM-16-QAM filters.

handle a limited amount of dispersion. Since the TD-DBP technique employs many filter sections, it becomes essential to carefully optimize each section. Here, LS-CO time-domain filters can give a large reduction in area usage compared to the frequency-domain implementation for the distances of interest. In addition, the power dissipation can be reduced but as suggested by our results, this will depend on the choice of dispersive step distance. As an example, using LS-CO time-domain dispersive steps of 33.5 km and 50 km for PM-QPSK leads to a power reduction of 38% and 24%, respectively (as illustrated in Fig. 18), compared to frequency-domain implementation of the steps. The area usage reduction for the same cases are 58% and 51%, respectively. Since the circuits that handle compensation of linear impairments dominate the circuits for non-linear mitigation, the power reduction in each step of the cascaded filter will translate into a significant system-wide power reduction.

Our evaluations consistently show that the LS-CO filter design method makes time-domain implementations competitive in terms of power dissipation to those in the frequency domain. Across all evaluations, LS-FB filters perform significantly worse than LS-CO and this is due to their need for longer wordlengths and longer impulse responses, showing the effectiveness of taking the pulse-shaped spectrum into account when designing CD filters. The receiver in this work operates

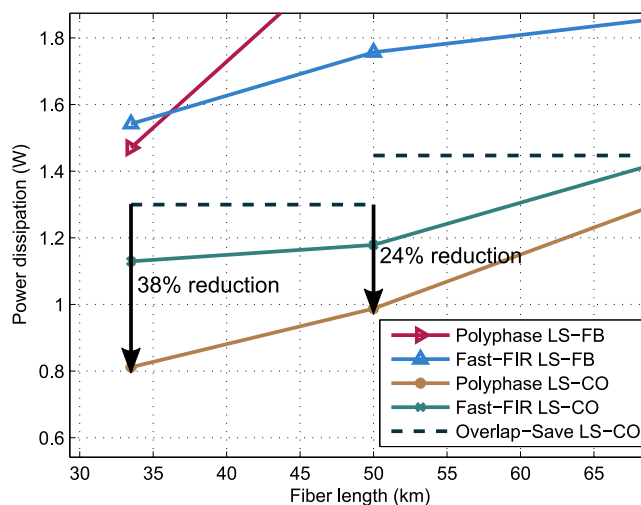


Fig. 18. Power dissipation reduction in the case of adjustable coefficients and PM-QPSK, also shown in Fig. 10.

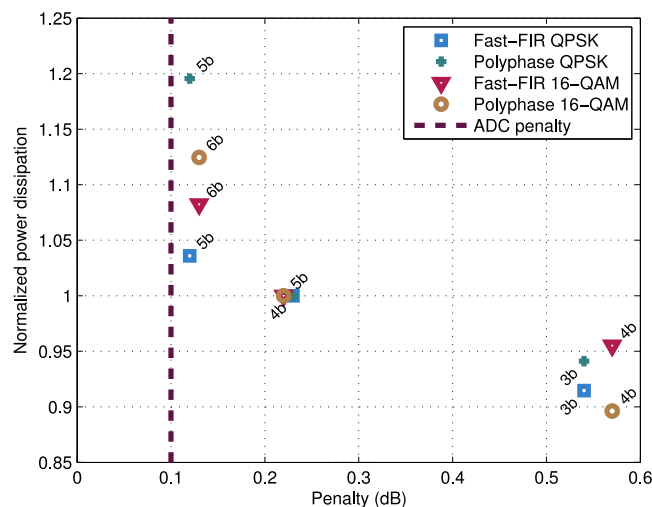


Fig. 19. Normalized power dissipation and corresponding power penalty for different coefficient resolution configurations. The dashed, vertical line shows the approximate A/D-converter penalty.

at 2 samples per symbol, and a reduction in oversampling ratio will give a lower impulse-response reduction in comparison to full-band optimizations. However, a reduced sampling rate also reduces the impulse-response length in all cases, which is advantageous for time-domain implementations. All in all, time-domain filters designed with the pulse shape in mind give significant circuit implementation improvements for low-to-moderate accumulated dispersion.

### 6.1 Power Dissipation and BER Performance

Time-domain implementations for compensation of 100-km fibers were synthesized with 3–5-bit and 4–6-bit coefficients for PM-QPSK and PM-16-QAM, respectively, to investigate the power-performance trade-off. Fig. 19 shows the power penalty and normalized power dissipation for the different wordlength configurations. The power dissipation was normalized to the power dissipation of 4- and 5-bit coefficient filters for PM-QPSK and PM-16-QAM, respectively. As shown in Fig. 19, 4-bit and 6-bit coefficients provide a good trade-off between power dissipation and power penalty.

Furthermore, the figures show that polyphase filters are more sensitive to coefficient wordlength than Fast-FIR filters, which can be attributed to an increase in internal wordlength requirements and the large pre- and post-processing trees.

## 7. Conclusion

We have shown that time-domain filter implementations optimized for the pulse shape (LS-CO filters [8], [9]) reduce the area usage and power dissipation of CD compensation for coherent systems with dispersion corresponding to up to 70 km, compared to a frequency-domain overlap-save implementation. Furthermore, Fast-FIR filters can extend the viable distance for time-domain CD compensation. We have also shown that optimizing the filter with the spectrum of the pulse-shaped signal provides large benefits for time-domain implementations; the LS-CO filter is less sensitive to coefficient quantization, achieving similar BER performance with one bit shorter coefficient wordlength and fewer taps, which leads to a significant power dissipation reduction.

## Acknowledgment

The authors would like to thank Dr. L. Svensson and Prof. A. Graell i Amat for fruitful discussions.

## References

- [1] D. Crivelli *et al.*, "Architecture of a single-chip 50 Gb/s DP-QPSK/BPSK transceiver with electronic dispersion compensation for coherent optical channels," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 4, pp. 1012–1025, Apr. 2014.
- [2] B. Pillai *et al.*, "End-to-end energy modeling and analysis of long-haul coherent transmission systems," *IEEE J. Lightw. Technol.*, vol. 32, no. 18, pp. 3093–3111, Sep. 2014.
- [3] G. Wellbrock and T. Xia, "How will optical transport deal with future network traffic growth?" in *Proc. Eur. Conf. Opt. Commun.*, Sep. 2014, Paper Th.1.2.1.
- [4] J. Pedro and B. Gangopadhyay, "Prospects for transparent handover between the metro and core segments of next-generation transport networks," in *Proc. Int. Conf. Transparent Opt. Netw.*, Jul. 2015, Paper Tu.D1.4.
- [5] C. Fougstedt, P. Johannisson, L. Svensson, and P. Larsson-Edefors, "Dynamic equalizer power dissipation optimization," in *Proc. Opt. Fiber Commun. Conf.*, Mar. 2016, Paper W4A.2.
- [6] S. J. Savory, "Digital filters for coherent optical receivers," *Opt. Exp.*, vol. 16, no. 2, pp. 804–817, Jan. 2008.
- [7] A. Eghbali, H. Johannsson, O. Gustafsson, and S. J. Savory, "Optimal least-squares FIR digital filters for compensation of chromatic dispersion in digital coherent optical receivers," *IEEE J. Lightw. Technol.*, vol. 32, no. 8, pp. 1449–1456, Apr. 2014.
- [8] A. Sheikh, C. Fougstedt, A. Graell i Amat, P. Johannisson, P. Larsson-Edefors, and M. Karlsson, "Dispersion compensation filter design optimized for robustness and power efficiency," in *Proc. Signal Process. Photon. Commun.*, Jun. 2015, Paper SpT3D.2.
- [9] A. Sheikh, C. Fougstedt, A. Graell i Amat, P. Johannisson, P. Larsson-Edefors, and M. Karlsson, "Dispersion compensation FIR filter with improved robustness to coefficient quantization errors," *IEEE J. Lightw. Technol.*, vol. 34, no. 22, pp. 5110–5117, Nov. 2016.
- [10] C. Fougstedt, A. Sheikh, A. Graell i Amat, P. Johannisson, and P. Larsson-Edefors, "Power-efficient time-domain dispersion compensation using optimized FIR filter implementation," in *Proc. Signal Process. Photon. Commun.*, Jun. 2015, Paper SpT3D.3.
- [11] T. Xu *et al.*, "Chromatic dispersion compensation in coherent transmission system using digital filters," *Opt. Exp.*, vol. 18, no. 15, pp. 16 243–16 257, Jul. 2010.
- [12] K. K. Parhi, *VLSI Digital Signal Processing*. Hoboken, NJ, USA: Wiley-Interscience, 1999.
- [13] R. Chassaing and D. Reay, *Digital Signal Processing and Applications with the TMS320C6713 and TMS320C6416 DSK*. Hoboken, NJ, USA: Wiley, 2008, ch. 6.
- [14] P. R. Halmos, *Finite-Dimensional Vector Spaces*. New York, NY, USA: Springer, 1958.
- [15] G. Nordin, P. Milder, J. Hoe, and M. Puschel, "Automatic generation of customized discrete Fourier transform IPs," in *Proc. Des. Autom. Conf.*, Jun. 2005, pp. 471–474.
- [16] Spiral DFT/FFT IP Core Generator. 2014. [Online]. Available: <http://www.spiral.net/hardware/dftgen.html>
- [17] *Cadence Encounter RTL Compiler, v. 14.11*, Cadence Design Systems, Inc., 2014.
- [18] C. Fougstedt, M. Mazur, L. Svensson, H. Eliasson, M. Karlsson, and P. Larsson-Edefors, "Time-domain digital back propagation: Algorithm and finite-precision implementation aspects," in *Proc. Opt. Fiber Commun. Conf.*, Mar. 2017, Paper W1G.4.