Frequency Domain Equalization with Minimum Complexity in Coherent Optical Transmission Systems

Jochen Leibrich and Werner Rosenkranz
Chair for Communications, Christian-Albrechts-Universität zu Kiel, Kaiserstraße 2, D-24143 Kiel
E-mail: jol@tf.uni-kiel.de

Abstract: For chip-design of a frequency-domain equalizer, a simple procedure to minimize the number of multiplications is shown. Considerations include recent improvements of FFT-algorithms.

OCIS-codes: (060.2360) Fiber optics links and subsystems; (060.4510) Optical communications

1. Introduction and intention

The amount of chromatic dispersion to be compensated for by means of digital electronic equalizers has changed significantly in the recent past. While for direct-detection transmission systems, the purpose of electrical equalization was to fine-tune dispersion compensation provided mainly by dispersion compensating fibers (DCF), for today's coherent transmission systems the aim is to omit DCF modules as far as possible to improve OSNR performance and to provide a high amount of dispersion compensation by means of electronic equalization [1].

As a consequence, the length M of the discrete impulse response of the equalizer increases by one order of magnitude at least. Hence, efficient implementation of the filtering algorithm is mandatory to limit chip size and power consumption. According to what is known in DSP theory, as soon as M goes beyond a threshold of approximately $M > 10$, fast convolution, which consists of one forward fast Fourier transform (FFT) followed by frequency domain multiplication with the transfer function of the equalizer and one final inverse FFT, shows much lower complexity than a time-domain FIR structure [2]. In equalization technology, the two methods are referred to as frequency-domain equalization (FDE) and time-domain equalization (TDE), respectively [2,3].

For FDE, the length N_{FFT} of the fast Fourier transforms provides an additional degree of freedom that may be utilized to minimize complexity. It is the intention of this contribution to provide a simple procedure that gives the optimum value for N_{FFT} depending on the amount of dispersion to be compensated for.

The paper is organized as follows: Section 2 provides a brief explanation of how algorithm complexity is evaluated here. Section 3 reviews recent developments in the field of FFT-algorithms and selects two of them for consideration in section 4, where the actual optimization is developed and its outcomes are discussed.

2. Quantification of algorithm complexity

The way the complexity of an algorithm is to be quantified depends heavily on the platform it is implemented on. In literature, often the costs for addition and for multiplication are considered to be equal [4]. This originates from the fact that today's computers provide hardware adders and multipliers. Both introduce the same latency, which is the relevant figure of merit although hardware effort is much higher for a multiplier compared to an adder.

For the implementation of a digital signal processing algorithm on an ASIC or an FPGA, however, power consumption and chip space are important. Here, the effort for a multiplier is much higher than for an adder. Therefore, the complexity of an algorithm needs to be measured in terms of the required number of multiplications, which is the figure of merit within this paper, too.

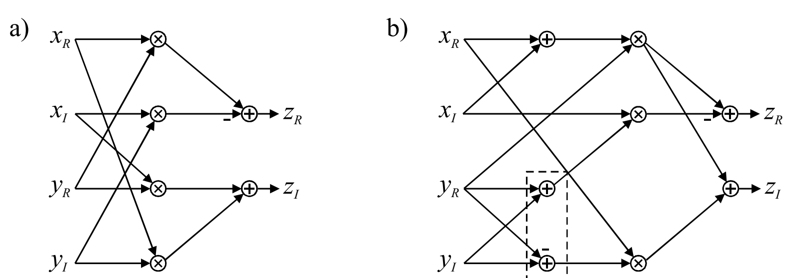

Fig.1. Complex multiplication $x{\cdot}y{=}z$ implemented by means of a) four real multiplications and two real additions, and b) by means of three real multiplications and five real additions. In case one of the two complex factors does not depend on the signal (here e.g. the factor y), which is the case for all multiplications that occur in FDE, two of the five additions marked by the dashed rectangle may be computed a priori.

To reduce the number of multipliers, it makes sense to implement the required complex multiplications not by means of the well-known procedure using four real multiplications and two real additions but by means of a modified procedure requiring only three real multiplications and five real additions. The two procedures are compared in Fig.1, where also the strategy for further reduction by two additions is explained.

3. Recent developments in the design of FFT algorithms

There is a large pool of different FFT algorithms. Clearly, the most widespread one is the classical radix-2 algorithm, for which the length of the vector to be transformed needs to be equal to a power of 2.

Theoretically, any integer number may be selected as basis for an FFT-algorithm, but only few of them are advantageous in terms of complexity. In all cases the number of complex multiplications for an FFT algorithm may be expressed by the form $C \cdot N_{FFT} \cdot \log_2(N_{FFT})$ [5]. For the radix-2 algorithm, C=1/2. In case of more advanced strategies, C may be decreased below this value. The radix-4 FFT for example reduces the number of complex multiplications by 25% to result in C=3/8, but it requires the length of the FFT to be a power of 4. An even more sophisticated algorithm, called split-radix FFT, results in C=1/3 and has been improved in 2007 to reach C<1/3 [4].

Within this paper, two FFT algorithms are taken into account: Firstly, due to its popularity and for comparison we consider the classical radix-2 algorithm. Secondly, since for the radix-4 algorithm the number of complex multiplications is reduced significantly at the cost of the constraint of N_{FFT} being equal to a power of 4, we consider this algorithm as an interesting alternative. For the realization of complex multiplications, we choose the method requiring three real multiplications such that the two FFT-algorithms require $3/2 \cdot N_{FFT} \cdot \log_2(N_{FFT})$ and $9/8 \cdot N_{FFT} \cdot \log_2(N_{FFT})$ real multiplications, respectively.

4. Optimization of length of FFT

One important difference between TDE and FDE is the fact that TDE is a continuous process while FDE is a block-wise algorithm. For FDE, a certain number K of input samples is collected [see Fig. 2a)], an FFT of length N_{FFT} is carried out, the spectrum is multiplied with the N_{FFT} -point transfer function of the equalizer, and finally an inverse FFT of the same length is carried out. Neighboring blocks are composed e.g. by means of the overlap-add algorithm [3]. For the optimization, the length M of the discrete impulse response of the equalizer is a central parameter. It may be related roughly to the amount of dispersion d to compensate for as follows [3], where [d]=ps/nm:

$$M \approx d \cdot \frac{c}{f_c^2} \cdot f_{symb}^2 \quad ,$$

where $f_c \approx 193$ THz is the optical carrier frequency, f_{symb} is the symbol frequency, and c is the speed of light. In case of f_{symb}=28 Gbaud that is required for 112 Gb/s polarization multiplexed QPSK [2], $M \approx d/160$ ps/nm.

Except for deviations due to limited numerical accuracy, the output signal of the equalizer is independent of N_{FFT} as long as it is larger than M. However, the complexity of the algorithm depends on N_{FFT}, so it makes sense to optimize its value. The basic approach for optimization is explained as follows: The length of M of the impulse response of the equalizer is a fixed parameter determined by the constraints according to the equation above. The length of the response of the equalizer to an input signal of length K is equal to K+M-1. Thus, N_{FFT} needs to be greater or equal to K+M-1. Best efficiency is achieved in case of equality.

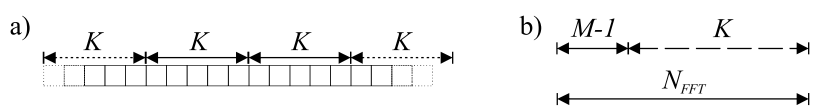

Fig.2. a) Separation of input data stream into blocks of length K; b) construction of FFT-length N_{FFT} from block length K and number of taps M.

Depending on the optimization parameter N_{FFT}, according to Fig. 2b) a number of K=N_{FFT} –M+1 samples per FFT may be processed. The objection of optimization is clarified as follows: On one hand, for a low value of N_{FFT} that is only slightly larger than M, K is very low and only a few samples may be processed per block. A high number of blocks is required yielding high computational effort. On the other hand, the complexity of the FFT grows faster than linearly in N_{FFT}. Thus, a value of N_{FFT} that is very large again increases computational effort. Minimum effort is obtained for a compromise in between, which is derived in the following: For the two FFTs and the frequency-domain multiplication, $N_{FFT} \cdot (6C \cdot \log_2[N_{FFT}]+3)$ real multiplications are required, where C depends on the specific FFT-algorithm. Per block a number of K samples is processed, so the number N_{MUL} of multiplications per sample is given by

$$N_{MUL} = \frac{N_{FFT} \cdot \left(6C \cdot \log_2\left[N_{FFT}\right] + 3\right)}{K} = \frac{N_{FFT} \cdot \left(6C \cdot \log_2\left[N_{FFT}\right] + 3\right)}{N_{FFT} - M + 1}.$$

Depending on whether the radix-2 or the radix-4 FFT is considered, N_{FFT} either needs to be equal to a power of two or to a power of four. In addition, it needs to be larger than M to ensure $K > 0$. Thus, only a few discrete values are allowed for N_{FFT}, and the minimization of N_{MUL} in N_{FFT} can be carried out simply by computation of N_{MUL} for those values that are allowed and determination of the minimum value.

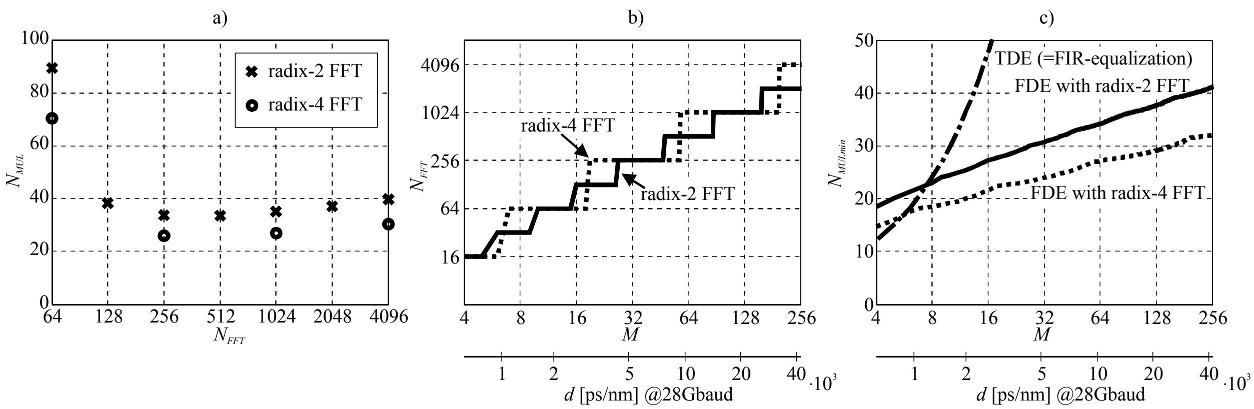

Fig. 3. a) Minimization of number N_{MUL} of multiplications for length M=50 of the impulse response of the equalizer (≈8000 ps/nm chromatic dispersion for f_{symb}=28 Gbaud); b) Optimum value for N_{FFT} and c) minimized number N_{MULmin} of real multiplications per sample for different filtering strategies; both depending on M resp. the amount of chromatic dispersion to compensate for in case of 28 Gbaud transmission.

The optimization process is shown in Fig. 3a) for an example with M=50, which is required to compensate for d≈8000 ps/nm in case f_{symb}=28 Gbaud. For the radix-2 FFT, for any power of two for N_{FFT} a value N_{MUL} can be computed. The lowest possible value is N_{FFT}=64>M=50. However, as a low number K=64-50+1=15 of samples are processed per block, a high number of blocks is required resulting in a high value of N_{MUL}≈90. Higher values for N_{FFT} result in significant improvement. The optimum value is N_{MUL}≈33 for N_{FFT}=512. For even longer FFTs, the effort again increases.

For the radix-4 FFT, only values that are a power of four are allowed. Basically, this is a slight drawback as the resolution for the optimization is lower. According to Fig. 3a), this drawback appears to be small compared to the fact that inherently the complexity is lower resulting in the minimum value of N_{MUL}≈26 for N_{FFT}=256.

This optimization procedure was carried out for arbitrary values for M yielding the results in Figs. 3b) and 3c). Fig. 3b) shows the optimum values for N_{FFT} for the two FFT strategies as function of M and of dispersion, respectively. Again, it is observable that for the radix-4 FFT only powers of four are allowed for N_{FFT}. Fig. 3c) shows the minimized complexity N_{MULmin}, which is the main result of this paper. The radix-4 FFT in any case shows lower complexity. Comparison of the complexity with TDE is given also, where the number of real multiplications per sample is 3M (i.e. three times the number M of complex multiplications). The point of intersection where FDE requires fewer multiplications than TDE is M≈8 in case of radix-2 FFT and M≈6 in case of radix-4 FFT.

5. Conclusion

Taking into account recent developments in the design of FFT-algorithms, the number of multiplications required to perform electronic dispersion compensation of coherent optical transmission systems is minimized. It is shown that implementation of the radix-4 FFT reduces the number of multiplications significantly. Optimization of N_{FFT} is possible such that even for low-order equalizers the number of multiplications required is less than for a time-domain FIR-filter.

References

[1] M. Kuschnerov et al., "DSP for coherent single-carrier receivers," IEEE J. Lightw. Technol. **27**, 3614-3622 (2009).

[2] B. Spinnler, F.N. Hauske, and M. Kuschnerov, "Adaptive equalizer complexity in coherent optical receivers," ECOC 2008, paper We.2.E.4.

[3] R. Kudo et al., "Coherent optical single carrier transmission using overlap frequency domain equalization for long-haul optical systems," IEEE J. Lightw. Technol. **27**, 3721-3728 (2009).

[4] S.G. Johnson and M.Frigo, "A modified split-radix FFT with fewer arithmetic operations," IEEE Trans. Sig. Proc. **55**, 111-119 (2007).

[5] A. Oppenheim et al., *Discrete-time signal processing* (Prentice Hall, 2009), Chap. 9.