Comparison of time and frequency domain interpolation implementations for MB-OFDM UWB transmitters

Eleni Fotopoulou
Department of Electrical and Computer Engineering, University of Patras, Greece
helenf@ece.upatras.gr

Dorina Thanou
School of Computer and Communications Sciences Swiss Federal Institute of Technology, Lausanne (EPFL)
dorina.thanou@epfl.ch

Thanos Stouraitis
Department of Electrical and Computer Engineering, University of Patras, Greece
thanos@ece.upatras.gr

***Abstract*—This paper investigates the effect of time-domain (TD) and frequency-domain (FD) interpolation on the performance of a Multi-Band (MB) Orthogonal Frequency Division Multiplexing (OFDM) Ultra-Wideband (UWB) system. We introduce a FD interpolator implemented by a radix-8 512-point IFFT architecture for applications on MB-OFDM UWB transmitters. For the specific application where the interpolation factor is fixed to four, the FD interpolator outperforms the TD interpolator implemented with digital low-pass FIR filters in terms of computational complexity. On the other hand simulation results show that FD implementation degrades the overall system performance for certain UWB channels.**

I. Introduction

Orthogonal Frequency Division Multiplexing (OFDM) has been selected as an efficient solution for high-speed digital wireless communications and is employed in current wireless communication standards. Currently, the WiMedia Alliance is supporting the use of Ultra-Wide Band implementing Multi-Band OFDM (MB-OFDM) for high-speed wireless, multimedia-capable personal-area interconnectivity [1].

In practical implementations typical DACs use digital interpolation filtering before implementing the D-to-A conversion. By interpolating the data samples, the performance of the reconstruction filter is improved, as increasing the sampling frequency can be seen to directly reduce the analog filters requirements for alias suppression. However, as discussed in [2], [3], filters in the digital and analog part both on the transmitter and the receiver end dominate the total systems impulse response in an indoor short-range wireless system. The impulse response affects the Inter-Symbol-Interference (ISI) degrading the system performance. In those terms it is crucial to design the transmitter with a digital interpolator taking into account both ISI and computational complexity. Reducing the ISI caused by the systems impulse response allows minimization of the guard interval and increase in the effective link capacity [4].

Interpolation in OFDM transmitters has been previously discussed in literature. In particular Leyonhjelm and Faulkner introduce three possible architectures to meet the requirements of a WLAN 802.11 OFDM system [2]. Conventional TD and circular time domain (CTD) interpolation are compared with FD interpolation in terms of ISI and computational complexity. The same authors in [3] show that the impulse response length of the transmitter is reduced by increasing the oversampling ratio and by using FD interpolation. In [5] a FD interpolator architecture for WLAN transmitters was introduced, where for an upsampling factor of eight, the FD interpolator outperforms the TD interpolator implemented with digital low-pass FIR filters in terms of computational complexity.

In this paper we investigate TD and FD interpolation in MB-OFDM UWB systems. The main disadvantage of the FD interpolation technique is the increase of the IFFT size. For this reason the oversampling rate is kept relatively low and at a factor of four. We examine the implementation of the IFFT with the advantages of oversampling in a higher rate by using a radix-8 IFFT implementation. For comparison reasons, we provide also the results of the same implementation when radix-2 is used. For the particular case of a 802.15.3a WPAN implementation a FD interpolator requires approximately 70% less multiplications per OFDM symbol when compared to the conventional filter interpolation technique for the same sampling rate.

The remainder of the paper is organized as follows: In Section II, a brief overview of the OFDM transmitter is presented. We present the proposed architecture of the implementation for both radix-2 and radix-8 in Section IV, while in Section V the computational complexity of the equivalent interpolation filter implementation is analyzed and comparisons are made. Conclusions are drawn in Section VI.

II. Overview of a MB-OFDM UWB System

The system model of a transmitter specified by the IEEE 802.15.3a standard proposal [6] is shown in Fig. 1. The binary input data after being scrambled is encoded by a convolutional encoder. The encoded data are subsequently interleaved and mapped onto QPSK values. Before the IFFT modulation is performed, 12 tone pilots are inserted in each symbol. A guard interval, which consists of zero samples, is introduced in each OFDM symbol, in order to maintain subcarrier orthogonality

978-1-4244-5309-2/10/$26.00 ©2010 IEEE

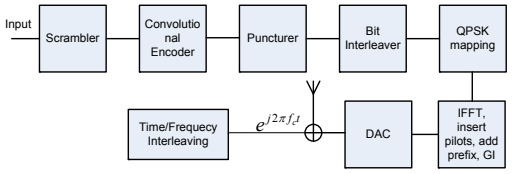


Fig. 1. Block diagram of the transmitter.

and to almost completely eliminate intersymbol interference. The OFDM symbol after being filtered is sent to the A-D converter and finally to the receiver which implements the reverse procedure. Inter-sub-band frequency diversity is introduced by repeating the same OFDM symbol over two different sub-bands while frequency domain spreading is obtained by choosing conjugate symmetric inputs to the IFFT.

III. Time and Frequency-Domain Interpolation

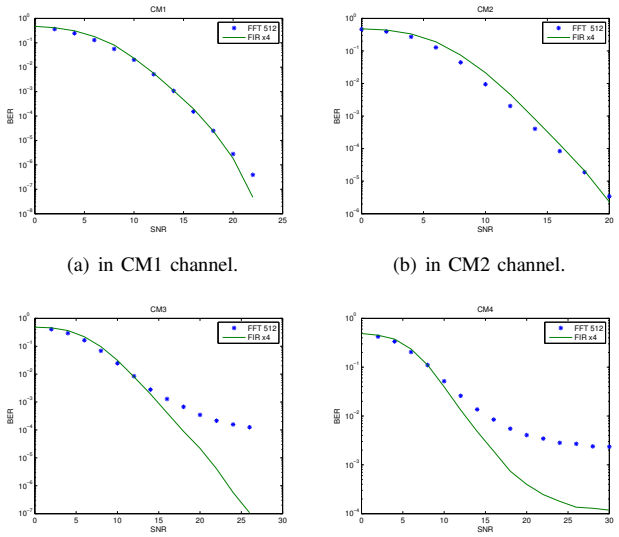
As described previously in the introduction section, two techniques for increasing the digital interpolation ratio may be used to reduce the impulse response and ISI: TD or FD interpolation. In order to perform TD interpolation we upsample the input sequence after the IFFT processing by using digital low pass FIR filters and then we downsample the received sequence by the same factor at the receivers end.

In order to perform FD interpolation we upsample the input data prior the FFT processing. The upsampling is performed by zero-padding the middle samples of the input sequence [7]. Hence in order to achieve an oversampling of a factor of 4, the 122 subcarriers of a typical OFDM UWB system should be padded with 390 zeros in the middle samples.

Below we investigate how interpolation in time and frequency domain affects the performance of multiband-OFDM by utilizing the inherent frequency diversity of the IEEE 802.15 UWB channel models [8]. CM1 describes a line of sight scenario with a separation between transmitter and receiver of less than 4m. CM2 describes the same range, but for a non-LOS situation.CM3 describes a non-LOS scenario for distances 4-10m and CM4 describes an environment with strong delay dispersion, resulting in a delay spread of 25ns.

The interpolation factor of the particular implementation is chosen to be four. The corresponding parameters applied for the simulations are as follows: the data rate is 80 Mb/s, the convolutional coding rate is $1/3$, the IFFT length is 128, the length of guard interval is 5 zero samples and the length of the cyclic prefix is 32 zero samples. Ideal synchronization is assumed and time domain spreading is employed to generate data rate of 80 Mb/s. The simulated BER performance is presented in SNR per bit for all channel modes and the final plots correspond to the performance of the 90th channel realization. A comparison of the two interpolation techniques has been made.

These results show that the performance for both interpolation techniques is found to be better in CM1 mode, which is absolutely expected as CM1 mode is less dispersive. The worst results appear in the CM4 mode as the BER does not even approach the acceptable value of 10^{-5}. The type of behavior



(a) in CM1 channel. (b) in CM2 channel.

(c) in CM3 channel. (d) in CM4 channel.

Fig. 2. Comparison of both interpolation techniques for an interpolation factor of 4

shown in these plots also indicates that the interpolation in the frequency domain can be implemented only in the first 2 channel modes. The performance in CM3, CM4 modes is optimum by using interpolation filters. In the following section we will present the architecture for the implementation of a 512 FFT using radix 2 and radix 8 butterflies.

IV. Proposed Frequency-Domain Architectures

A. Using a radix-8 512-point FFT

The Radix-8 FFT algorithm derives from the Cooley-Tukey algorithm described in [9]. The algorithm assumes that the sequence length can be expressed as product $N = N_1 N_2 ... N_\nu$. In the implementation studied in this paper, the sequence length is $N = 512 = 8^3$, therefore $N_1 = N_2 = N_3 = 8$. Three-dimensional index maps are defined according to the Decimation-in-Time (DIT) FFT, where $0 \leq n_1, n_2, n_3 \leq 7$ and $0 \leq k_1, k_2, k_3 \leq 7$. Eq. (1) derives by simplifying and rearranging the W_N coefficients.

$$X[k] = X[k_1 + 8k_2 + 64k_3]$$
$$= \sum_{n_3=0}^{N_3-1} \left[\left(\sum_{n_2=0}^{N_2-1} \left[\left(\sum_{n_1=0}^{N_1-1} x[64n_1 + 8n_2 + n_3] W_8^{n_1 k_1} \right) W_{64}^{n_2 k_1} \right] W_8^{n_2 k_2} \right) W_{512}^{n_3(k_1+8k_2)} \right] W_8^{n_3 k_3}. \quad (1)$$

From (1) the expression of the 512-point FFT is derived in terms of a three-dimensional structure of 8-point FFTs with complex interdimensional constant multiplications of the twiddle factors.

The block diagram of the radix-8 512-point FFT that is based on (1) is given in Fig. 3. It consists of the input unit that reorders the input data in an appropriate fashion, the first 8-point FFT unit, two full 8-point FFT units, two multiplier units, two storage reordering units (RU) for the intermediate data and finally the output unit. The basic operating schemes

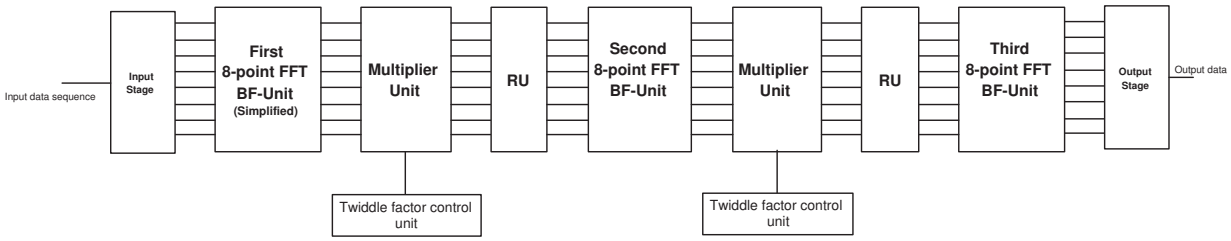


Fig. 3. Block diagram of the 512-point FFT.

of the input unit and multiplier units are described in [10]. The interesting feature of the realization of a radix-8 butterfly is the limited need for explicit multiplication operations [11]. The IFFT implementation, which is of special interest in an OFDM transmitter, is carried out by simply swapping the real and imaginary parts of the incoming data, performing the forward FFT and finally by swapping once again the real and imaginary parts of the data output. In this work we focus on the particularities of the interpolation implementation and exploit them to simplify the leading FFT unit.

In order to perform FD interpolation we upsample the input sequence prior the FFT processing [2] as described in Section III. By doing so, and by reordering the input data in groups of octets as the applied algorithm requires, we obtain an input, where in each octet only two samples are at the most potentially non-zero. The non-zero samples are located at the top and/or at the bottom of each octet as can be seen in Fig. 4. The particular organization of the input simplifies the radix-8 butterfly of the first stage, as shown in Fig. 5. The simplified butterfly has two inputs, one for each set of "top" and "bottom" samples of each octet. Compared to the radix-8 butterfly, that is essentially a radix-2 8-point FFT, the simplified butterfly is much more efficient in terms of number of calculations and processing time as we can obtain the first stage intermediate results after one stage as opposed to three. The intermediate data produced after the first stage do not necessarily contain zero elements and thus the remainder of the implementation cannot be further simplified

Incoming octets per clock cycle							
8	**7**	**6**	**5**	**4**	**3**	**2**	**1**
x(56)	x(48)	x(40)	x(32)	x(24)	x(16)	x(8)	x(0)
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
x(114)	x(106)	x(98)	x(90)	x(82)	x(74)	x(66)	0

Fig. 4. Pattern of the incoming oversampled data.

Considering that the input and the coefficients are complex and taking into account that a complex multiplication can be performed with three real multiplications, the total number of multiplications occurs to be 2499. The theoretical number of multiplications for a conventional radix-8 512 point FFT is 3204 [12], there is a 22% savings in multiplication operations for the interpolation implementation.

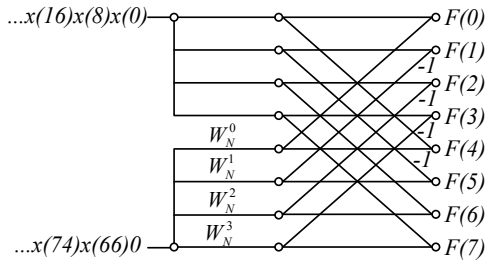


Fig. 5. Simplified butterfly scheme.

B. Using a radix-2 512-point FFT

Another way to implement a 512-point FFT is by using radix-2 butterfly. In this case the sequence length can be expressed as 2^9, which essentially results in a nine stage structure. In this case we also take advantage of the fact that the upsampling of the input sequence is performed by zero padding the middle samples. By reordering the input data in groups of dyads in the first stage of the block diagram we obtain an input where in each dyad at most one sample can be non zero. The non zero sample for the first 16 cycles is located either at the top or at the bottom as can be seen in Fig. 6 and the pattern is repeated periodically for the remaining cycles. As a result the butterfly of the first stage can be simplified.

Incoming dyads per clock cycle															
16	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**
0	0	0	x[48]	0	0	0	x[16]	0	0	0	x[32]	0	0	0	x[0]
x[106]	0	0	0	x[74]	0	0	0	x[90]	0	0	0	0	0	0	0

Fig. 6. Pattern of the incoming oversampled data.

In the case that only the "top" sample is non zero the two outputs are a replica of the input sample. Otherwise, if only the "bottom" sample is non zero the first output is also a replica of the input while the second output is the replica of the input multiplied by -1. The simplified butterfly is given in Fig. 7

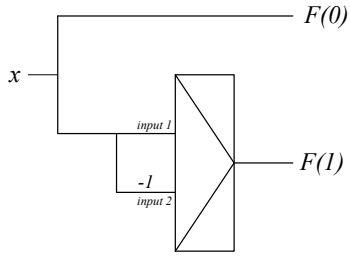


Fig. 7. Simplified butterfly scheme of the first stage.

The control of the multiplexer is given by the scheme in Fig. 8 Due to the zero samples in the input of the second stage we can

Clock cycle	1-4	5-64	65-68	69-124	125-128	129-132	133-188	189-192	193-252	253-256
Input1 enable	1	1	1	1	0	1	1	0	1	0
Hold	3	2	3	2	3	3	2	3	2	3
Input2 enable	0	1	0	1	1	0	1	1	1	1

Fig. 8. Control of the multiplexer of the simplified butterfly (first stage).

do further simplifications on the butterfly of the second stage. After the appropriate reordering, the input samples present a similar behavior with that of the first stage. As a result we can use the simplified butterfly scheme of the first stage shown in Fig. 7 once again, by controlling the multiplexer to fit the pattern of the intermediate results entering the second stage. No further simplifications can be done after the second stage since no zero samples are observed.

The computational complexity analysis reveals that there is a total of 3852 real multiplications per OFDM symbol. The cost in multiplications of an FD interpolator is 12% less than a typical 512-FFT implementation.

V. Hardware complexity and comparisons

Comparing the two proposed architectures with radix-8 and radix-2 FFT implementations the choice depends on the optimization criteria. An FD interpolator with radix-8 butterfly provides a way to realize a 512-point FFT at a reduced hardware cost, as it utilizes 35% less real multiplications compared to the radix-2 512-point FFT implementation. Moreover using the first implementation, a gain in processing time is achieved as it requires only three stages while the implementation with radix-2 butterfly requires nine stages. However, the use of a radix-2 butterfly is more efficient in applications where the implementation area is considered crucial.

TD interpolation by a factor of 4 is implemented by adding a cyclic prefix and upsampling the sequence by zero-padding between the samples. The upsampled sequence is filtered by an FIR low pass filter and it is sent to the channel. In practice the TD filters that implement the interpolation by a factor of 4 consist of two stages of upsampling by a factor of 2 followed by a half-band filter (HBF)[13]. A FIR low-pass filter design to meet the 802.15.3a spectral mask [6] requirements is a 23-tap HBF followed by a 15-tap HBF and is used for simulations. Taking into account that the input is complex the total number of real multiplications required for filtering the oversampled data is 9272.

In the following table we summarize the total number of multiplications per OFDM symbol needed for all the above mentioned cases. For the TD implementation the number of multiplications per OFDM symbol needed for the 128-point IFFT of the MB-OFDM system is included realized with radix-2 and one butterfly, radix-2 and five butterflies, and split-radix [12], to compare all possible implementations. The savings for each case compared to a typical implementation of TD interpolation with a radix-2 FFT implementation is given. As can be seen in Table I there is a tremendous gain in computational complexity with the FD interpolation implementation. Another important drawback of using digital

Interpolation method	Mult/OFDM symbol	Savings vs typical impl.
FD (radix-2 512-FFT)	3.852	63%
FD (radix-8 512-FFT)	2.499	76%
TD (FIR(x4)+ radix-2 (1BF) 128-FFT)	10.626	-
TD (FIR(x4)+ radix-2 (5BF) 128-FFT)	9.984	6%
TD (FIR(x4)+S-R 128-FFT)	9.788	8 %

TABLE I
NUMBER OF MULTIPLICATIONS PER OFDM SYMBOL NEEDED FOR DIFFERENT INTERPOLATION METHODS

FIR filters is the processing delay compared to the FFT implementation. This delay can be decreased by using polyphase filters but even in this case the delay is larger than the three-stage FFT implementation proposed using radix-8. The number of multiplications is not affected by the use of polyphase interpolators.

VI. Conclusions

In this work a comparison between TD and FD interpolators for MB-OFDM UWB transmitters was performed. Two FD interpolation architectures using a radix-8 FFT and a radix-2 FFT implementation were proposed. FD interpolation was found to reduce the computational complexity by **60%-70%**, while there is a gain in processing time mainly due to the advantages of the radix-8 butterfly. However, as found in simulation results FD interpolation is acceptable only for short range wireless applications.

References

[1] "How it works:UWB, WPAN and WiMedia Radio Space," WiMedia Alliance, Tech. Rep., 2008.

[2] S. Leyonhjelm and M. Faulkner, "Comparison of three interpolator designs for use in OFDM transmitters," *Electronics Letters*, vol. 42, no. 5, pp. 289– 291, Mar. 2006.

[3] ——, "Designing for low ISI in an OFDM modem," *TENCON 2005, 2005 IEEE Region 10*, pp. 1–5, Nov. 2005.

[4] M. Faulkner, "The effect of filtering on the performance of OFDM systems," *IEEE Transactions on Vehicular Technology*, vol. 49, no. 5, pp. 1877–1884, Sep. 2000.

[5] E. Fotopoulou, V. Paliouras, and T. Stouraitis, "A frequency-domain interpolation implementation for OFDM transmitters," *Wireless Pervasive Computing, 2008. ISWPC 2008. 3rd International Symposium on*, pp. 628–632, May 2008.

[6] "Multi-band OFDM Physical Layer Proposal for IEEE 802.15 Task Group 3a," IEEE, Tech. Rep., 2003.

[7] R. Van Nee and R. Prasad, *OFDM Wireless Multimedia Communications*. Artech House Publishers, 2000.

[8] E. J. Foerster, "Channel modeling sub-committee report final," IEEE, Tech. Rep. IEEE P802.15-02/490r1-SG3a, 2003.

[9] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Prentice Hall Inc., 1999.

[10] K. Maharatna, E. Grass, and U. Jagdhold, "A 64-point Fourier transform chip for high-speed wireless LAN application using OFDM," *Solid-State Circuits, IEEE Journal of*, vol. 39, no. 3, pp. 484–493, March 2004.

[11] L. Jia, Y. Gao, J. Isoaho, and H. Tenhunen, "A new VLSI-oriented FFT algorithm and implementation," *ASIC Conference 1998. Proceedings. Eleventh Annual IEEE International*, pp. 337–341, 13-16 Sep 1998.

[12] S. K. Mitra and J. F. Kaiser, *Handbook for Digital Signal Processing*. John Wiley & Sons, 1993.

[13] P. Vaidyanathan, *Multirate Systems and Filter Banks*. Prentice Hall Inc., 1993.