Optimized implementation of FMT modulation on DSP

Ondrej Krajsa, Pavel Silhavy, Martin Koutny and Petr Sysel

***Abstract*—This paper deals with description of multitone modulation technique called Filtered MultiTone. Main part of this article describes prototype filter for FMT modulation and implementation of filter bank. Direct form, efficient form with using FFT algorithm and optimized form of implementation are described. All of this possibilities of implementation are compared. Also implementation of efficient and optimized algorithms in C is shown.**

***Index Terms*—DSP, Filtered MultiTone, Implementation, Efficient, Prototype Filter**

I. Introduction

MULTICARRIER modulations (MCM) are very popular modulation techniques, used in modern transmittion systems like ADSL, VDSL, PLC, WiMAX etc. This modulation techniques enables optimum utilization of provided frequency band on used channel. The twisted pair copper used in ADSL and VDSL transmission was originally designed for voice transmission in the band up to 3400 Hz. The transmission channel in PLC systems is created by conductors primarily used for electricity distribution, where the insertion loss is strongly affected by topology. The last example is wireless technology using fading channel with multipath signal propagation and with high degree of interferences from other systems. Multicarrier modulation technique utilize the fact that when the transmission band is divided into a sufficient number of parallel subchannels, it is possible to consider that the transmission function is constant. It means that the equalization on the receiver's side is easier. In all the above mentioned systems the well known and well described modulation DMT (Discrete MultiTone) or OFDM (Orthogonal Frequency Division Multiplexing) is used. These modulations are relatively easy to implement, but the properties of DMT respectively OFDM modulation cannot completely allow effective using of transmission channels with specially shaped spectral characteristics with sharp edges. This is due the sync-function shape subchannel frequency characteristics. Therefore, the alternative modulation approaches with better characteristics are looked for. One of alternative modulations is FMT - Filtered MultiTone. FMT modulation represents a modulation technique using filter banks to divide the frequency spectrum, proposed by Giovanni Cherubini in 1999[1]. Another type of this modulation is the Half-overlapped FMT.

This work was supported in part by the Ministry of Education, Youth, and Sports of Czech Republic under research programme MSM 0021630513 and also by the Grant Agency of Czech Republic under Grant 102/09/1846.

O. Krajsa is with the Brno University of Technology, Faculty of Electrical Engineering and Communication, Purkynova 118, Brno, 602 00, Czech Republic; email: krajsao@feec.vutbr.cz..

P. Silhavy is with the Brno University of Technology, Faculty of Electrical Engineering and Communication, Purkynova 118, Brno, 602 00, Czech Republic; e-mail: silhavy@feec.vutbr.cz.

M. koutny is with the Brno University of Technology, Faculty of Electrical Engineering and Communication, Purkynova 118, Brno, 602 00, Czech Republic; e-mail: koutnym@feec.vutbr.cz.

P. Sysel is with the Brno University of Technology, Faculty of Electrical Engineering and Communication, Purkynova 118, Brno, 602 00, Czech Republic; e-mail: sysel@feec.vutbr.cz.

II. Filtered Multitone

FMT modulation uses filter banks for individual subchannels separating. The most used technique is uniform, critically sampled filter bank. It means that each subchannel has the same bandwidth and that they are equally distributed in the $< 0, Fs >$ interval, where Fs is sampling frequency. This modulation scheme was firstly introduced in [1].In this case, subchannels are perfectly separated. The spectrum of transmitted signal in this FMT system is shown in Fig. 2. This variant is called "Non-overlapped FMT". Another one is called "Half-overapped FMT"[4]. In this variant, individual subchannels overlap one half of each other. The spectrum of transmitted signal in Half-overlapped FMT system is shown in Fig. 3. A block diagram of FMT system is shown on Fig. 1. In both systems are the complex symbols on the input. They are obtained from the QAM modulation. The number of bits allocated to each carrier is determined during the initialization of transmission according to the level of interference and the attenuation of channel[5]. Filters in each branch are frequency-shifted versions of prototype filter - low pass FIR filter in the first branch. Transmitted signal can be described by eguation(1)

$$x(n) = \frac{1}{\sqrt{2N}} \sum_{k=-\infty}^{\infty} \sum_{i=0}^{2N-1} X_i^k h(t-kT) e^{j\frac{2\pi nk}{2N}}, \qquad (1)$$

In contrast to DMT, the inter channels interference (ICI) are strongly suppressed, up to the level comparable with other noise thanks to the separation of subchannel. On the other hand, in FMT modulation the inter-symbol interferences (ISI) occur even on ideal channel, which is given by FMT modulation system principle, adding filters to IFFT output and FFT input[6]. Therefore, it is necessary to perform equalizing of transmission channel and also equalization of filters influence. This equalization may be realized completely in the frequency domain. FMT also facilitates the application of frequency division duplex, because there is no power emission from one channel into another.

ISBN: 978-960-474-250-9

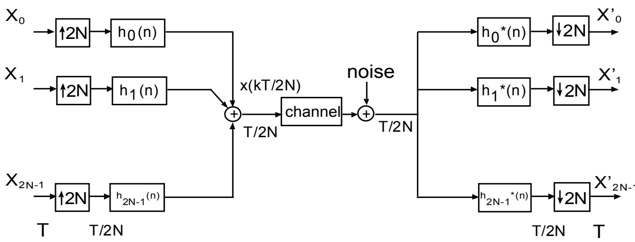

Fig. 1. Direct realization of FMT system

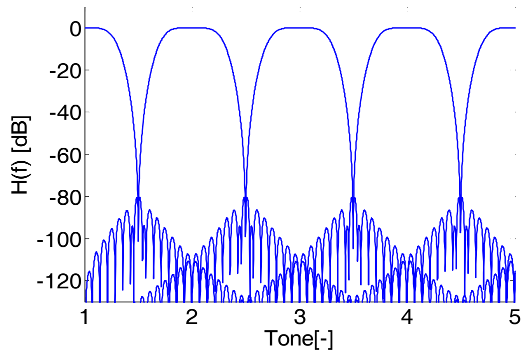

Fig. 2. Non-overlapped FMT modulation

III. Filtered MultiTone implementation

The first step in implementation is the prototype filter design. We typically attempt to design it to reach the best frequency characteristic. It mainly concerns the suppression of side lobes, the orthogonality of derived filters, and the frequency separation of particular subchannels in non-overlap variant of FMT. In this way we can obtain an ideal suppression of inter-channel interference ICI. Inter-symbol interference ISI, on the other hand, will not be limited by these actions as described above. The prototype filter can be designed using any FIR filter design method with the limitation that the filters derived must be orthogonal to each other. The following methods meeting the orthogonality condition appear the most convenient for the design. Approximation of IIR filter [1], where ρ is the control parameter. ρ controls the shape of filter transition bandwidth and is within the interval $(0, 1)$. For greater ρ the final filter is of a greater steepness but it decreases the stop-band attenuation. If we increase ρ above 0.6, the ripple in the pass-band will increase significantly. For $\rho = 0.9$ it can be up to $12dB$. The side lobes are suppressed to $-63dB$ for $\rho = 0.1$. The other option is to shape the filter transition bandwidth using the square-root raised cosine filter [2]. In this filter, the lobes are suppressed to $-38dB$ for $\alpha = 0.5$. The third method is the windows method. The final characteristic is then formed by the properties of the window used. The Blackman, Hamming or Hanning windows appear to be sufficient, possibly also some other window retaining the orthogonality with sufficient side lobe fall-off and broadness of the main lobe. Another method uses the modified Parks-McClellan algorithm [3].

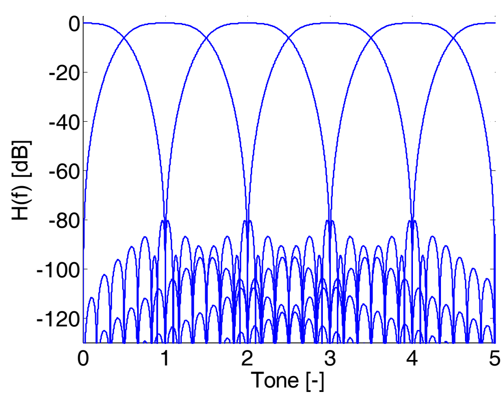

Fig. 3. Half-overlapped FMT

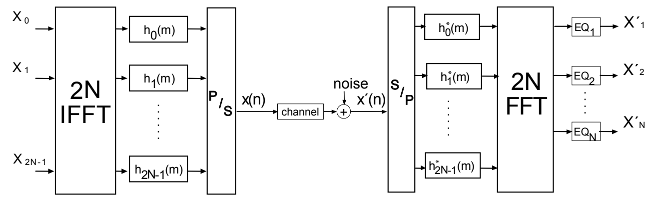

Fig. 4. Efficient realization of FMT system

Also the filterbank can be designed in two ways. The first way is described above. This direct realization of FMT is very unefficient. Filter with high order - $2\gamma N$ is in each branch. It means high delay and high computational complexity. In [2] was described efficient implementation using FFT algorithm. Block diagram of FMT with help of FFT algorithm is shown in Fig. 4. In comparison with direct realization, in each branch is low-order FIR filter, polyphase component of prototype filter obtained with helf of polyphase decomposition by equation(2)

$$h_i(m) = h(2mN + i) \tag{2}$$

Comparison of both realizations is in table I, where FFT is FFT dimension and also frame size, D-FMT is direct realization, E-FMT si effective realization and γ is overlap factor. This table compare direct and effective realization in light of number of MAC (multiply and accumulate) instructions, necessary to calculating one output frame.

Sub-optimal implementation of filterbank ob DSP is shown on Fig. 5, where h_n^m is n-th coefficient of m-th filter, X_n^m is n-th symbol in m-th branch and is o_m^n is n-th output sample in m-th branch (m-th sample of n-th frame). We have three buffers. One (a) for prototype coefficients, one circular buffer (b) for input symbols from IFFT, and the last one, circular buffer (c) for output frame. Samples in buffer b are written in frames of $2N$ samples, also output samples are read in frames of $2N$ samples.

Example 1. Filter bank function in Matlab

```
function [y]=bank_filt(x,h,M,g)
y=zeros(1,M*g);
  for i=0:M*g
     y(1,mod(i,M))=y(1,mod(i,M))+h(i)*x(i);
  end
```

This way of filtering is more effective, because we need only one **for** cycle for computing one output frame. Implementation

FFT (Frame size)	DMT	E-FMT $\gamma = 8$	E-FMT $\gamma = 14$	D-FMT $\gamma = 8$	D-FMT $\gamma = 14$
512	18432	38912	51200	8392704	14684160
1024	40960	81920	106496	33562624	58728448
2048	90112	172032	221184	34234112	234897408
4096	196608	360448	458752	536903680	939556864
8192	425984	753664	950272	2147549184	3758161920

TABLE I
NUMBER OF MAC INSTRUCTIONS PER FRAME

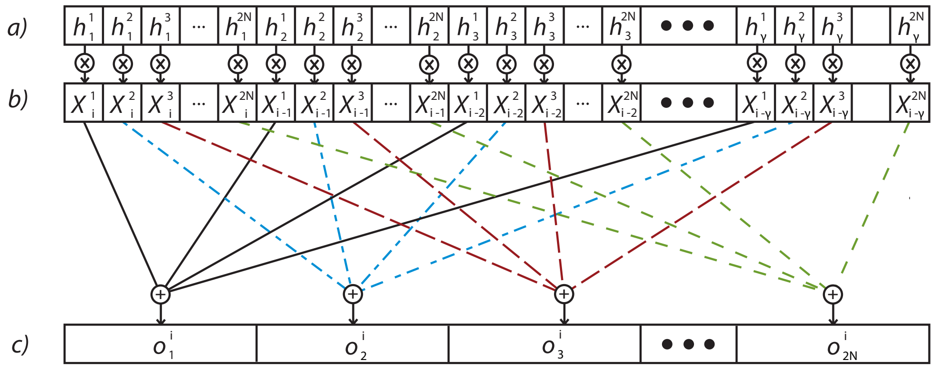

Fig. 5. Optimized algorithm for implementing FIR filter bank

of this algorithm in Matlab is shown in Example 1.

Next step is implementing this algorithm on DSP. Compilers designed for digital signal processors differs from the ANSI-C or C++ standard in a few details, which in the ultimate result have a considerable effect on the speed and stability of algorithm implementation. The basic difference lies in that the defined data types are fully adapted to the architecture of digital signal processor. The number of data bits and the format of storing numbers in a given code (mostly the two's complement) correspond to the actual storage of numbers in digital signal processor registers. When optimizing the source code it is convenient either to enter the instructions of digital signal processor assembler directly into the C-language source code or to use the intrinsic functions, which are assembled as a single instruction. In this way, the critical parts of source code that the assembler is not able to analyze correctly, can be optimized. In parallel processing the given algorithm can be realized simultaneously for several values of the input signal. Using parallel processing will greatly increase the speed of algorithm processing. Optimized C code for filter bank is shown in Example 2. The input sample pointer `x` and the pointer to the field of prototype filter coefficients `h` can be declared by the key word `const` since in the course of calculation the input sample value and the values of individual filter coefficients will not change. The output sample value and the values of state-space variables will, on the contrary, change during calculation and thus they cannot be declared by the key word `const`. It is obvious from the algorithm structure that the individual input arguments represent mutually independent data structures, which will be stored in separate memory locations. In that case it is of advantage to use the key word `restrict`, which informs the compiler about the memory-independence of the variables. In case the output sample was entered into the same memory field as the input samples (in-place processing), there would evidently be a dependence relation between the in and out pointers and the restrict keyword could not be used in declaring the two arguments.

Example 2. Filter bank for FMT in C

```
void bank_filt( const int x[restrict],
                const int h[restrict],
                      short y[restrict],
                  //frame length
                  int M,
                  //overlap factor
                      int g){
  int i;

  for (i=0 ; i < M*g ; i=i+2)
  {
      //sample in i-th position
      *(y+i%M) += _mpy( h[i], x[i]);
      //sample in i-th + 1 position
      *(y+i%M+1) += _mpyh( h[i], x[i]);
  }
}
```

The arguments are assumed to be of the int type (i.e. 32-bit arguments),where 16 most significant bits are one sample and least signicivant another one. The multiply operation performs only the product of two 16-bit arguments (16x16 bits). The two multiply functions will then differ in the way the 16-bit parts are aligned in a 32-bit argument. While the first function, `_mpy`, multiplies 16 least significant bits of both arguments, the second function, `_mpyh`, multiplies 16 most significant bits of both arguments. In this way, two samples are processed in parallel.

ISBN: 978-960-474-250-9

IV. Conclusion

In this article, the FMT modulation was described and possibilities of its implementation were shown. Efficient implementation of this modulation with using FFT algorithm was compared with direct implementation. Effectiveness of the implementation was compared to the number of multiply and accumulate (MAC) instructions, that are necessary for calculation one output frame. Also implementation of filter bank was shown for Matlab and for sub-optimal implementation on VLIW digital signal processors in C language.

References

[1] G. CHERUBINI, E. ELEFTHERIOUS, S. OLCER,*Filtered multitone modulation for VDSL*, Proc. IEEE Globecom'99, pp.1139-1144, Rio de Janeiro,Brazil, 1999.

[2] CHERUBINI, G., ELEFTERIOU, E., OELCER, S. *Filtered Multitone modulation for Very High-speed Digital Subscriber Lines*. In IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, vol 20, No. 5. [s.l.] : [s.n.], 2002. s. 1016-1028

[3] BERENGUER, I., WASSELL, I.J. *FMT Modulation: Receiver Filter Bank Definition for the derivation of an Efficient Implementation*,IEEE 7th International OFDM Workshop, Hamburg (Germany), September 2002.

[4] ŠILHAVÝ, P. *Half-overlap subchannel Filtered MultiTone Modulation with the small delay*. In The Seventh International Conference of Networking. 1. Cancun, Mexico: IARIA, LCN 2007941923, 2008. s. 1-5. ISBN: 978-0-7695-3106-9

[5] BINGHAM, J. A. C. *ADSL, VDSL, and Multicarrier Modulation*. [s.l.] : Wiley Interscience, 2000. 289 s. ISBN 0-471-29099-8

[6] BENVENUTO, N., TOMASIN, S., TOMBA, L. *Equalization methods in DMT and FMT Systems for Broadband Wireless Communications*. In IEEE Transactions on Communications, vol. 50, no. 9. [s.l.] : [s.n.], 2002. s. 1413-1418