Progress In Electromagnetics Research C, Vol. 5, 93–117, 2008

COMPRESSION OF MULTICARRIER PHASE-CODED RADAR SIGNALS BASED ON DISCRETE FOURIER TRANSFORM (DFT)

R. Mohseni, A. Sheikhi, and M. A. Masnadi-Shirazi

Department of Electrical Engineering
School of Engineering
Shiraz University
Shiraz, Iran

Abstract—Multicarrier Phase-Coded signals have been recently introduced to achieve high range resolution in radar systems. As in single carrier phase coded radars, the conventional method for compression of these signals is based on using matched filter or direct computation of autocorrelation function. In this paper we propose a new method based on Discrete Fourier Transform (DFT) that has lower computational complexity compared to the conventional approach. It has been proved that the proposed method is mathematically equivalent to matched filtering, so there is no processing loss. Also the effect of sampling frequency on compression loss has been investigated and for the oversampled matched filter of MCPC signals, a computational efficient algorithm based on polyphase implementation has been proposed.

1. INTRODUCTION

Nowadays, there is an increasing demand for high range resolution radar systems in many applications such as target recognition and classification [1, 2]. High range resolution is also achieved by increasing the bandwidth of the radar signal. Among the techniques that can be used for this purpose are phase and frequency modulation of transmitted pulse [3]. However increasing bandwidth in the phase code modulated signal (PCM) is equivalent to decreasing chip width which faces the technological limitations if it is beyond more than a specific extent. On the other hand, in linear frequency modulation (LFM) creating a signal with high frequency slope and adequate linearity in frequency is not easily possible. To overcome this problem one can use multicarrier signals.

Multicarrier signals for radars have been presented for the first time in 1998 by Jankiraman et al. [4–6]. They designed PANDORA radar with a signal consisting of several narrowband LFM channels where the channels are separated by frequency guard bands. Output signals of different channels are combined in receiver in such a way that range data with high resolution are produced. An appropriate replacement of the analog LFM signal is a digital phase coded modulation signal or PCM, particularly the P3 and P4 codes, in which the phase sequences are samples from the phase trajectory of a LFM signal [7]. In other words, similar to multicarrier LFM signal, PCM sequences can be modulated on N subcarriers and transmitted simultaneously. Using this signal in radar systems has been proposed by Levanon in 2000 [7]. In the proposed idea, minimum frequency space between subcarriers is used to preserve orthogonality.

The main advantage of multicarrier signal based on PCM (MCPC) compared with multicarrier signal based on LFM, is its high spectral efficiency due to its property of orthogonal frequency division multiplexing (OFDM). Because the LFM based multicarrier signals need to use guard band to separate each frequency channel, but in MCPC signals based on using OFDM concept it is not necessary to use guard bands. On the other hand by choosing appropriate codes, the side lobes of ambiguity function can also be reduced significantly [7–15].

The other good point about this signal which is the main issue of this paper is to achieve compression of the signal by introducing efficient computational algorithms based on interesting inherent mathematical property of the signal.

In this paper, first, the MCPC signal is introduced and the conventional method for its compression is presented. Then the new compression method is proposed and its advantages are discussed. Also, some necessary discussions are presented on choosing the appropriate sampling frequency for realization of digital matched filter, and a low frequency sampling rate method based on polyphase implementation of the oversampled matched filter is suggested for compressing these signals.

2. MCPC SIGNAL AND ITS CHARACTERISTICS

A multicarrier phase-coded (MCPC) pulse consists of P carriers transmitted simultaneously where each carrier is phase modulated using a sequence of M chips. The carrier frequencies are equally spaced with frequency separation equal to the inverse of the chip duration, forming orthogonal frequency division multiplexing (OFDM) [16–19]. The complex envelope of the MCPC signal is given by:

$$x(t) = \sum_{m=1}^{M} \left[\sum_{p=0}^{P-1} a_{p,m} \exp(j2\pi f_p t) \right] s\left(t - (m-1)t_c\right) \tag{1}$$

where

$$s(t) = \begin{cases} 1, & 0 \le t < t_c \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

and

$$f_p = \frac{p}{t_c}, \quad p = 0, 1, \ldots, P-1 \tag{3}$$

The string $\{a_{p,m}\}_{m=1}^{M}$ in Equation (1) is the code sequence on p-th carrier and t_c is the chip width.

Ambiguity function of MCPC signal depends on modulated codes on each carrier signal. It has been shown that by choosing proper codes for each subcarrier in MCPC signals, ambiguity function sidelobes in both range and Doppler domain can be reduced to below that of the single-carrier signals with similar resolution, [15]. Several works on designing appropriate code for MCPC signal can be found in [9, 10].

One of the obvious characteristics of the MCPC signals is their frequency spectrum which is almost bounded and flat, with effective band width of P/t_c and its frequency sidelobes less than that of the single-carrier coded signal that has a shape of the Sinc function [8].

A major drawback of the multicarrier signal is its varying envelope during a pulse. For transmitter power amplifier it is desirable to reduce the Peak to Mean Envelope Power Ratio (PMPER) as much as possible. Also, several works have been reported to decrease the PMPER that are based on the appropriate code designing, amplitude and initial phase of each subcarrier setting [7, 15].

3. CONVENTIONAL METHOD FOR THE MCPC PULSE COMPRESSION

The conventional method for compressing phase coded pulse is to use discrete time version of matched filter. In this method, after sampling the received signal, correlation of the produced samples and samples of the reference signal are calculated. For MCPC signal, the first choice for sampling rate, based on the signal bandwidth, is $f_s = P/t_c$. Of course it can be shown that using this sampling rate may cause much compression loss. So to generalize the following discussion, sampling rate of $f_s = NP/t_c$ is used where N is an integer number.

Now if we sample the MCPC signal defined in Equation (1) with $f_s = NP/t_c$ rate, we have:

$$\begin{aligned} x[n] &= x\left(nt_s\right) = x\left(n\frac{t_c}{NP}\right) \\ &= \sum_{p=0}^{P-1}\sum_{m=1}^{M} a_{p,m}\exp\left(j2\pi\left(\frac{p}{t_c}\right)\frac{nt_c}{NP}\right)s\left(\frac{nt_c}{NP}-(m-1)t_c\right) \\ &= \sum_{p=0}^{P-1}\sum_{m=1}^{M} a_{p,m}\exp\left(j2\pi\frac{np}{NP}\right)s\left(\frac{nt_c}{NP}-(m-1)t_c\right), \\ & \quad n = 0,\, 1, \ldots,\, NPM-1 \end{aligned} \tag{4}$$

By choosing window function $s[n] = s\left(\frac{nt_c}{NP}\right)$ then:

$$s\,[n] = \begin{cases} 1, & 0 \le n \le NP-1 \\ 0, & \text{else} \end{cases} \tag{5}$$

Also

$$s\left(\frac{nt_c}{NP}-(m-1)t_c\right) = s\left[n-(m-1)NP\right] \tag{6}$$

Hence Equation (4) could be rewritten as:

$$\begin{aligned} x[n] &= \sum_{p=0}^{P-1}\sum_{m=1}^{M} a_{p,m}\exp\left(j2\pi\frac{np}{NP}\right)s\left[n-(m-1)NP\right], \\ & \quad n = 0,\, 1, \ldots,\, NPM-1 \end{aligned} \tag{7}$$

In the conventional method, matched filter is used for pulse compression, where the filter impulse response is the sampled version of the analog filter impulse response given by

$$h(t) = x^*\left(Mt_c - t\right) \tag{8}$$

where superscript $*$ denotes complex conjugate.

If we sample this function with $f_s = NP/t_c$ rate, we have:

$$\begin{aligned} h\left[n\right] &= x^*\left(Mt_c - \frac{nt_c}{NP}\right) = x^*\left((NPM-n)\frac{t_c}{NP}\right) = x^*\left[NPM-n\right] \\ &= \sum_{p=0}^{P-1}\sum_{m=1}^{M} a^*_{p,m}\exp\left(-j2\pi\frac{(NPM-n)p}{NP}\right) \\ &\quad s\left(\frac{(NPM-n)t_c}{NP} - (m-1)t_c\right), \\ &\quad n = 1, \ldots, NPM \end{aligned} \tag{9}$$

Since the impulse response of the matched filter can be shifted in time domain, so the impulse response of the matched filter can be chosen as:

$$\begin{aligned} h[n] &= x^*[(NPM-1)-n] \\ &= \sum_{p=0}^{P-1}\sum_{m=1}^{M} a^*_{p,m}\exp\left(-j2\pi\frac{[(NPM-1)-n]p}{NP}\right) \\ &\quad s\left(\frac{[(NPM-1)-n]t_c}{NP} - (m-1)t_c\right), \\ &\quad n = 0, 1, \ldots, NPM-1 \end{aligned} \tag{10}$$

By this choice, non-zero samples of the matched filter will be at samples $n = 0$ to $n = NPM - 1$.

Based on the above definition for discrete window function $s[n]$, Equation (10) is equal to:

$$\begin{aligned} h[n] &= \sum_{p=0}^{P-1}\sum_{m=1}^{M} a^*_{p,m}\exp\left(-j2\pi\frac{[(NPM-1)-n]p}{NP}\right) \\ &\quad s\left[(NPM-1)-n-(m-1)NP\right], \\ &\quad n = 0, 1, \ldots, NPM-1 \end{aligned} \tag{11}$$

The matched filter output is calculated as follows:

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{+\infty} x[k]h[n-k] \tag{12}$$

where $*$ denotes linear convolution. For example if 5 carriers are used in the MCPC signal, and the consecutive ordered cyclic shift of a P4 code with length 5 has been used to modulate these 5 carriers [8], its matched filter output for a fixed target is shown in Figure 1. The phase vector of P4 code that is used is given by:

$$\mathbf{a} = [0^\circ \quad -144^\circ \quad -216^\circ \quad -216^\circ \quad -144^\circ] \tag{13}$$

In this example the sampling rate $f_s = P/t_c$ is used for the matched filter implementation and the target delay is assumed to be an integer number of the sampling interval.

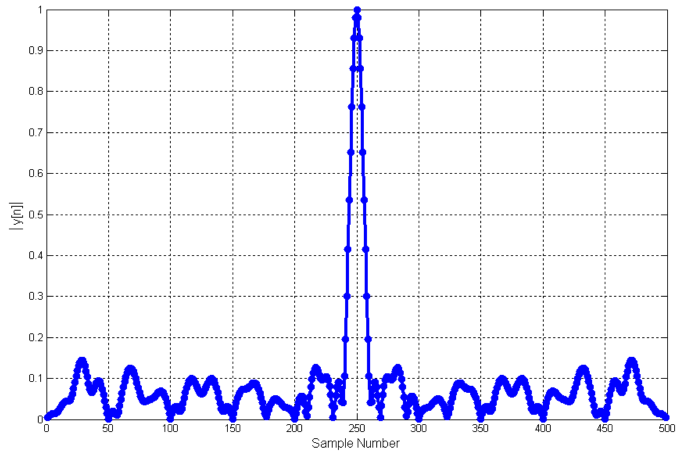


Figure 1. The matched filter output (conventional method of pulse compression) for $M = P = 5$, Target delay=$10t_c$, and $f_s = P/t_c$.

4. THE SUGGESTED METHOD FOR COMPRESSING THE MCPC SIGNAL

In this section another method for realization of the matched filter or Equation (12) is presented to compress the MCPC signal. For this purpose, Equation (11) can be written in the following form by swapping the two summations:

$$\begin{aligned} h[n] \;=\; & \sum_{m=1}^{M} \sum_{p=0}^{P-1} a^*_{p,m} \exp\left(-j2\pi \frac{[(NPM-1)-n]p}{NP}\right) \\ & s\left[(NPM-1)-n-(m-1)NP\right], \\ & n = 0, 1, \ldots, NPM-1 \end{aligned} \tag{14}$$

Using this equation, function $h[n]$ can be considered as a summation of M functions given by:

$$h[n] = \sum_{i=1}^{M} h_i[n] \tag{15}$$

where:

$$\begin{aligned} h_i[n] &= \sum_{p=0}^{P-1} a_{p,i}^* \exp\left(j2\pi \frac{[n-(NPM-1)]p}{NP}\right) \\ &\quad s\left[(NPM-1)-n-(i-1)NP\right] \\ &= \begin{cases} \sum\limits_{p=0}^{P-1} a_{p,i}^* \exp\left(j2\pi \frac{[n-(NPM-1)]p}{NP}\right), & \\ \qquad\qquad n=(M-i)NP,\,\dots\,,(M-i+1)NP-1 & \\ 0, \qquad \text{otherwise} & \end{cases} \end{aligned} \tag{16}$$

and also the matched filter output in response to input $x[n]$ is equal to:

$$\begin{aligned} y[n] &= x[n]*h[n] = \sum_{i=1}^{M} x[k]*h_i[n] \\ &= \sum_{k=n+1-NPM}^{n-NP(M-1)} x[k]h_1[n-k] + \sum_{k=n-NP(M-1)+1}^{n-NP(M-2)} x[k]h_2[n-k] \\ &\quad + \ldots + \sum_{k=n-NP+1}^{n} x[k]h_M[n-k] \\ &= \sum_{i=1}^{M} y_i[n] \end{aligned} \tag{17}$$

where $y_i[n]s$ are defined as:

$$y_i[n] = \sum_{k=n+1-NP(M-i+1)}^{n-NP(M-i)} x[k]h_i[n-k] = \mathbf{X}_i^T \cdot \mathbf{h}_i^* \tag{18}$$

where

$$\begin{aligned} \mathbf{X}_i &= [x[n-(M-i+1)NP+\mathbf{1}] \quad x[n-(M-i+1)NP+\mathbf{2}] \\ &\quad \ldots \quad x[n-(M-i+1)NP+\mathbf{NP}]]^T \\ \mathbf{h}_i^* &= [h_i[(M-i+1)NP-\mathbf{1}] \quad h_i[(M-i+1)NP-\mathbf{2}] \\ &\quad \ldots \quad h_i[(M-i+1)NP-\mathbf{NP}]]^T \end{aligned} \tag{19}$$

We know that if $\mathbf{A}$, and $\mathbf{B}$ are two vectors and $\mathbf{U}$ is a unitary matrix, we have:

$$\mathbf{A}^T\mathbf{B}^* = \hat{\mathbf{A}}^T\hat{\mathbf{B}}^* \tag{20}$$

where

$$\begin{aligned}\hat{\mathbf{A}} &= \mathbf{UA}\\ \hat{\mathbf{B}} &= \mathbf{UB}\end{aligned} \tag{21}$$

Now if we define the matrix $\mathbf{U}$ according to Equation (22), then calculation of $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ are equal to the DFT calculation of the elements of $\mathbf{A}$ and $\mathbf{B}$ respectively.

$$\mathbf{U} = \frac{1}{\sqrt{NP}} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \exp\left(\frac{-j2\pi}{NP}\right) & \exp\left(\frac{-j2\pi\times 2}{NP}\right) & \dots & \exp\left(\frac{-j2\pi(NP-1)}{NP}\right) \\ 1 & \exp\left(\frac{-j2\pi\times 2}{NP}\right) & \exp\left(\frac{-j2\pi\times 2\times 2}{NP}\right) & \dots & \exp\left(\frac{-j2\pi\times 2\times(NP-1)}{NP}\right) \\ \dots & \dots & \dots & \dots & \dots \\ 1 & \exp\left(\frac{-j2\pi(NP-1)}{NP}\right) & \exp\left(\frac{-j2\pi(NP-1)\times 2}{NP}\right) & \dots & \exp\left(\frac{-j2\pi(NP-1)(NP-1)}{NP}\right) \end{bmatrix}_{NP\times NP} \tag{22}$$

By using the above definition for matrix $\mathbf{U}$ and using Equation (20), the inner product of Equation (18) can be calculated as follows (this equation can also be obtained using the Parseval's theorem):

$$y_i[n] = \mathbf{X}_i^T \cdot \mathbf{h}_i^* = \hat{\mathbf{X}}_i^T \cdot \hat{\mathbf{h}}_i^* \tag{23}$$

where:

$$\hat{\mathbf{X}}_i = \frac{1}{\sqrt{NP}}DFT\{\mathbf{X}_i\}, \quad \hat{\mathbf{h}}_i^{=}\frac{1}{\sqrt{NP}}DFT\{\mathbf{h}_i\} \tag{24}$$

On the other hand the vector $\hat{\mathbf{h}}_i$ can be calculated as:

$$\begin{aligned}\hat{\mathbf{h}}_i(K) \;=\; & \frac{1}{\sqrt{NP}}\sum_{n=0}^{NP-1}\mathbf{h}_i(n+1)\exp\left(-j2\pi\frac{nK}{NP}\right),\\ & K = 0,\, 1,\, \dots\,,\, NP-1\end{aligned} \tag{25}$$

where $\mathbf{h}_i(K)$ and $\hat{\mathbf{h}}_i(K)$ denotes the k-th element of the vectors $\mathbf{h}_i$ and $\hat{\mathbf{h}}_i$ respectively. Hence $\hat{\mathbf{h}}_i^*$ is given by:

$$\begin{aligned}\hat{\mathbf{h}}_i^*(K) &= \frac{1}{\sqrt{NP}}\sum_{n=0}^{NP-1}\mathbf{h}_i^*(n+1)\exp\left(j2\pi\frac{nK}{NP}\right),\\ &\quad K=0,\,1,\,\ldots\,,\,NP-1 \qquad (26)\\ \hat{\mathbf{h}}_i^*(K) &= \frac{1}{\sqrt{NP}}\sum_{n=0}^{NP-1}h_i[(M-i+1)NP-n-1]\exp\left(j2\pi\frac{nK}{NP}\right) \qquad (27)\end{aligned}$$

Therefore:

$$\begin{aligned}\hat{\mathbf{h}}_i^*(K) &= \frac{1}{\sqrt{NP}}\sum_{n=0}^{NP-1}\sum_{p=1}^{P}a_{p,i}^*\\ &\quad \exp\left(j2\pi\frac{[(M-i+1)NP-n-1-(NPM-1)]p}{NP}\right)\exp\left(j2\pi\frac{nK}{NP}\right)\\ &= \frac{1}{\sqrt{NP}}\sum_{n=0}^{NP-1}\sum_{p=1}^{P}a_{p,i}^*\exp\left(j2\pi\frac{[(1-i)NP-n]p}{NP}\right)\exp\left(j2\pi\frac{nK}{NP}\right)\\ &= \frac{1}{NP}\sum_{p=1}^{P}a_{p,i}^*\sum_{n=0}^{NP-1}\exp\left(j2\pi\frac{n(K-p)}{NP}\right) \qquad (28)\end{aligned}$$

On the other hand:

$$\sum_{n=0}^{NP-1}\exp\left(j2\pi\frac{n(K-p)}{NP}\right)=\begin{cases}NP, & p=K\\ 0, & p\neq K\end{cases} \qquad (29)$$

So we have:

$$\hat{\mathbf{h}}_i^*(K)=\begin{cases}\sqrt{NP}a_{K,i}^*, & K<P\\ 0, & K\geq P\end{cases} \qquad (30)$$

The Fourier transform of the vector $\mathbf{X}_i$ is also equal to:

$$\begin{aligned}\mathbf{F}_{X_i}(K) &= \sqrt{NP}\hat{\mathbf{X}}_i(K)\\ &= \sum_{b=0}^{NP-1}x[n+1-(M-i+1)NP+b]\exp\left(-j2\pi\frac{Kb}{NP}\right),\\ &\quad K=0,\,1,\,\ldots\,,\,NP-1 \qquad (31)\end{aligned}$$

So the inner product of Equation (18) can be calculated by:

$$y_i[n] = \sum_{K=0}^{NP-1} \hat{\mathbf{X}}_i(K)\hat{\mathbf{h}}_i(K) = \sum_{K=0}^{P-1} \mathbf{F}_{X_i}(K)a^*_{K,i} \tag{32}$$

and the matched filter output is equal to:

$$y[n] = \sum_{i=1}^{M} y_i[n] = \sum_{i=1}^{M}\sum_{K=0}^{P-1} \mathbf{F}_{X_i}(K)a^*_{K,i} \tag{33}$$

By interchanging the order of summation in Equation (33), we have:

$$y[n] = \sum_{K=0}^{P-1}\left[\sum_{i=1}^{M} \mathbf{F}_{X_i}(K)a^*_{K,i}\right] = \sum_{K=0}^{P-1} S_K \tag{34}$$

The realization of this equation which is shown in Figure 2, presents a new method for compressing the MCPC signals. Based on this equation, to calculate the matched filter output in every sample time, first, the last received NPM samples are divided into M segments where each one contains NP samples and the DFT of length NP is computed for each segment. The first P samples of each DFT are demultiplexed and the resulting P different sequences are filtered by P conventional single carrier pulse compression filters that are matched to the corresponding codes modulated on each subcarrier.

At the end different channels data are added in order to compute the final output for the given sample time. This operation is performed sequentially by sliding on all of the sample times.

If we apply the above algorithm to the example of Section 3 with the same sampling rate, the result is exactly the same as the one obtained in Figure 1.

The important characteristic of the suggested method to compress the MCPC signal for realization of matched filter is that its computational complexity is lower than the conventional method, as shown in Section 7. Also the proposed method has been proved that is mathematically equivalent to the conventional matched filter, so there is no processing loss compared to the conventional approach.

As another important characteristic of the suggested method is the possibility of phase and amplitude estimation for different subcarriers frequencies based on this method. These data can be used in radar systems in various scenarios, especially for appropriate power management in the tracking mode. This is a new concept in radar systems and would deserve a separate research by itself.

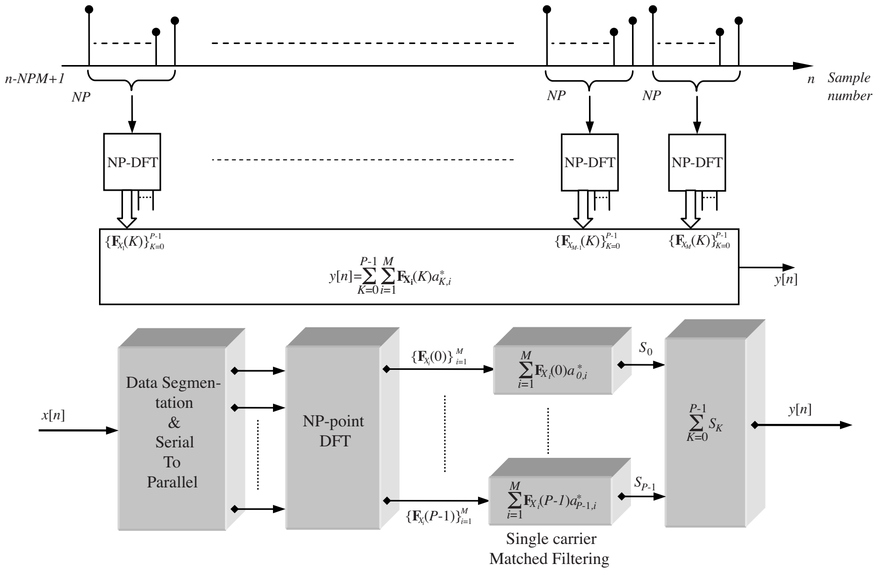
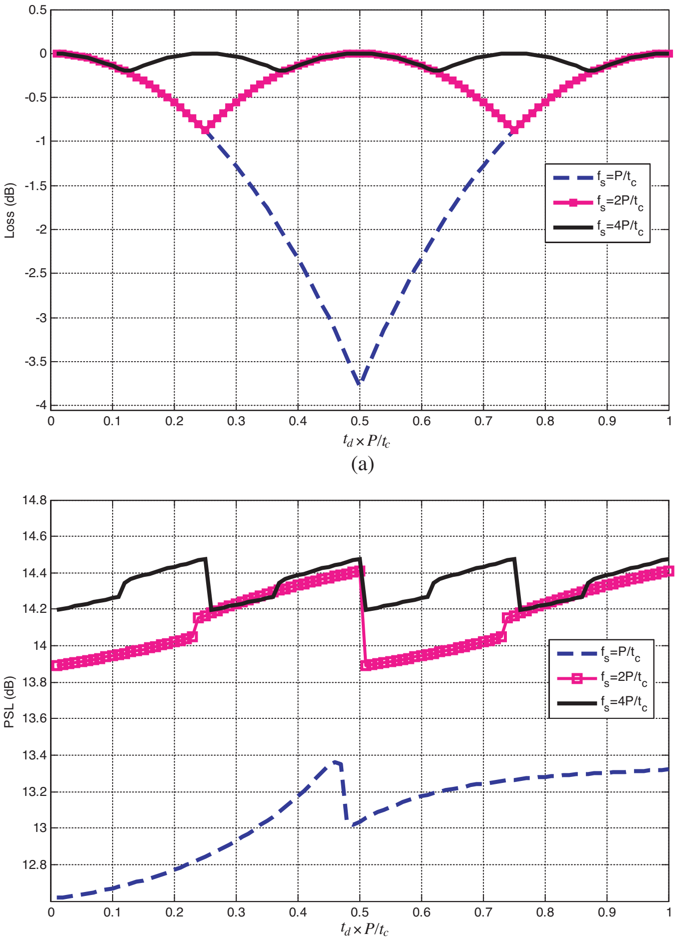

n-NPM+1
NP
NP
NP
n
Sample number
NP-DFT
NP-DFT
NP-DFT
$\{\mathbf{F}_{X_1}(K)\}_{K=0}^{P-1}$
$\{\mathbf{F}_{X_{M-1}}(K)\}_{K=0}^{P-1}$
$\{\mathbf{F}_{X_M}(K)\}_{K=0}^{P-1}$
$y[n]=\sum_{K=0}^{P-1}\sum_{i=1}^{M}\mathbf{F}_{\mathbf{X}_\mathbf{i}}(K)a_{K,i}^*$
$y[n]$
$x[n]$
Data Segmentation & Serial To Parallel
NP-point DFT
$\{\mathbf{F}_{X_i}(0)\}_{i=1}^{M}$
$\{\mathbf{F}_{X_i}(P-1)\}_{i=1}^{M}$
$\sum_{i=1}^{M}\mathbf{F}_{X_i}(0)a_{0,i}^*$
$\sum_{i=1}^{M}\mathbf{F}_{X_i}(P-1)a_{P-1,i}^*$
S_0
S_{P-1}
$\sum_{K=0}^{P-1}S_K$
$y[n]$
Single carrier Matched Filtering

Figure 2. Block diagram of the proposed method to compress the MCPC signals.

5. THE EFFECT OF SAMPLING FREQUENCY ON THE COMPRESSION QUALITY

If we study the matched filter output, at the sampling frequency P/t_c, we find that when the target signal delay is not exactly a multiple of sampling rate, the pulse compression faces some problems. In this case, the peak output power of the matched filter is reduced, and autocorrelation sidelobes also increase. For example, if the barker code of length 5 is used to modulate different 5 carriers, the peak power loss versus target delay will be as in Figure 3(a) and the ratio of the peak power to maximum sidelobe level is also similar to Figure 3(b).

Figure 3(a) shows a compression loss of 3.7 dB in some target delays which is drastically a large amount. To avoid the peak power loss by changing the target delay, we should increase the sampling rate. As this figure shows, increasing the sampling rate by a factor of 2 or 4 can reduce the compression loss considerably. Increasing the sampling frequency, makes the ratio of the peak power to maximum sidelobe level almost independent of the target delay, as it is shown in Figure 3(b). According to these figures the appropriate sampling rate for compressing the MCPC signal would be more than $f_s = P/t_c$ and $f_s = 4P/t_c$ seems to be enough for this purpose.



(a)

(b)

Figure 3. (a) Compression loss versus normalized target delay (t_d/t_s) for 3 sampling frequencies ($M = P = 5$, Target delay$= 9.1t_c$, and $t_s = t_c/P$). (b) The ratio of the peak power to maximum sidelobe level versus normalized target delay (t_d/t_s) for 3 sampling frequencies ($M = P = 5$, Target delay$= 9.1t_c$, and $t_s = t_c/P$).

6. POLYPHASE IMPLEMENTATION OF THE OVERSAMPLED MCPC PULSE COMPRESSION FILTER

As it has been shown in Section 5, if the sampling frequency is P/t_c and the target signal delay is not exactly a multiple of sampling interval, the matched filter output signal amplitude will reduce and also sidelobe levels will increase slightly. This is mainly due to the fact that in this case the discrete-time pulse compression filter is not exactly the matched filter. So mismatch loss can be expected. A method to avoid these mismatch losses that depend on the target delay is to increase the sampling rate. It has been shown that increasing the sampling rate by a factor of 4 can dramatically reduce the mismatch loss and certainly this causes the increase of the computational complexity.

For MCPC signals the lowest possible sampling rate according to Nyquist criteria is approximately P/t_c. So to reduce the mismatch loss in discrete time pulse compression we should implement an oversampled matched filter.

In this section we derive polyphase representation of the oversampled matched filter and show that the polyphase representation yields a more efficient structure for compression of MCPC signals.

Consider MCPC signal $x[n]$ as the sampled received signal $x(t)$ by the Nyquist sampling rate (P/t_c). According to above discussion it should be oversampled by a factor of L, and then applied to a matched filtering at sampling rate of LP/t_c as shown in Figure 4.

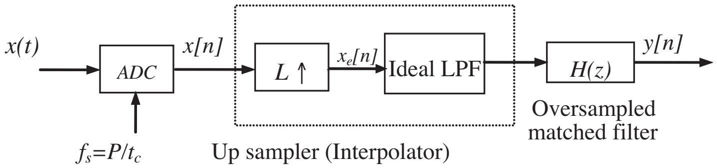


Figure 4. The oversampled matched filtering after interpolation to avoid compression loss.

In Figure 4 the first block in the up-sampler is called "sampling rate expander" or "L-fold expander" in literature [20, 21] and its output is given by:

$$x_e[n] = \begin{cases} x\left[\frac{n}{L}\right], & n = 0, \pm L, \pm 2L, \ldots \\ 0, & \text{otherwise} \end{cases} \tag{35}$$

or equivalently:

$$x_e[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n - kL] \tag{36}$$

The second block in the up-sampler shown in Figure 4 is the interpolator filter which is an ideal low pass filter with cut off frequency $\frac{\pi}{L}$. $H(z)$ in Figure 4 is the oversampled matched filter or pulse compression filter, given by:

$$H(z)=\sum_n h[n]z^{-n} \tag{37}$$

where $h[n]$ is given by Equation (11) with N replaced by L:

$$\begin{aligned} h[n] \;=\; & \sum_{p=0}^{P-1}\sum_{m=1}^{M} a^*_{p,m}\exp\left(-j2\pi\frac{[(LPM-1)-n]p}{LP}\right) \\ & s\left[(LPM-1)-n-(m-1)LP\right], \\ & n=0,\,1,\,\ldots\,,\,LPM-1 \end{aligned} \tag{38}$$

and

$$s\left[n\right]=\begin{cases} 1, & 0\leq n\leq LP-1 \\ 0, & \text{else} \end{cases} \tag{39}$$

or

$$\begin{aligned} & s\left[(LPM-1)-n-(m-1)LP\right] \\ = & \begin{cases} 1, & (M-m)LP\leq n\leq(M-m+1)LP-1 \\ 0, & \text{else} \end{cases} \end{aligned} \tag{40}$$

Since in Figure 4 the interpolator filter is an ideal low pass filter which has approximately the same bandwidth as the pulse compression filter $H(z)$, so it can be omitted as shown in Figure 5. Thus, oversampled matched filter $H(z)$ can be used as the interpolator filter in this figure too.

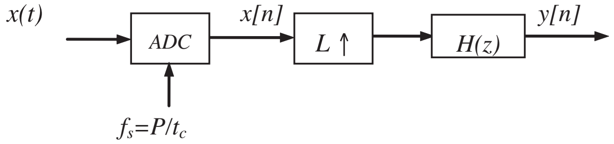


Figure 5. Using the oversampled matched filtering as interpolation filter in Figure 4.

As it is well-known in multirate signal processing, we can decompose $H(z)$ as: [21]

$$\begin{aligned} H(z) \;=\; & \sum_n h[nL]z^{-nL}+z^{-1}\sum_n h[nL+1]z^{-nL} \\ & +\ldots+z^{-(L-1)}\sum_n h[nL+L-1]z^{-nL} \end{aligned} \tag{41}$$

This can compactly be rewritten as:

$$H(z) = \sum_{l=0}^{L-1} z^{-l} E_l(z^L) \tag{42}$$

where:

$$E_l(z) = \sum_n e_l[n] z^{-n} \tag{43}$$

with:

$$e_l[n] = h[Ln + l], \quad 0 \leq l \leq L-1 \tag{44}$$

Equation (42) which is depicted in Figure 6 is called the polyphase representation of the oversampled matched filter $H(z)$ of Figure 4 (with respect to L). In Figure 6, $E_l(z)$, $l = 0, \ldots, L-1$ are called the polyphase components.

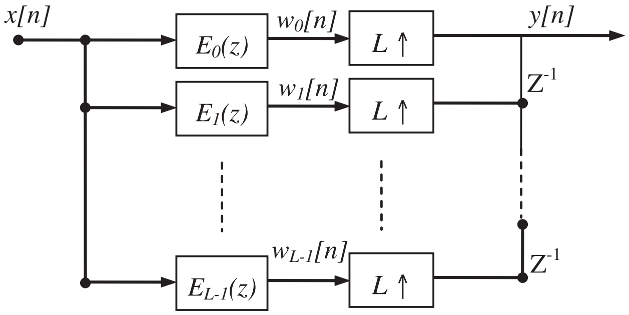


Figure 6. Polyphase implementation of the oversampled matched filter, $H(z)$.

Using Equations (38) and (44), the l-th polyphase component of $H(z)$ is given by:

$$\begin{aligned} e_l[n] &= h[nL + l] \\ &= \sum_{p=0}^{P-1} \sum_{m=1}^{M} a_{p,m}^{*} \exp\left(-j2\pi \frac{[(LPM-1)-(nL+l)]p}{LP}\right) \\ &\quad\; s\left[(LPM-1)-(Ln+l)-(m-1)LP\right], \\ &\quad\; n = 0, 1, \ldots, PM-1 \end{aligned} \tag{45}$$

where:

$$\begin{aligned} & s\left[(LPM-1)-(Ln+l)-(m-1)LP\right] \\ = \; & s\left[(PM-1)-n-(m-1)P\right] \\ = \; & \begin{cases} 1, & (M-m)P \leq n \leq (M-m+1)P-1 \\ 0, & \text{else} \end{cases} \end{aligned} \tag{46}$$

Hence Equation (45) could be expressed as:

$$\begin{aligned} e_l[n] \; = \; & \sum_{p=0}^{P-1} \sum_{m=1}^{M} a^*_{p,m} \exp\left(-j2\pi \frac{[L-l-1]p}{LP}\right) \\ & \exp\left(-j2\pi \frac{[(PM-1)-n]p}{P}\right) s\left[(PM-1)-n-(m-1)P\right] \\ & n=0,\, 1,\, \ldots,\, PM-1, \quad l=0,\, 1,\, \ldots,\, L-1 \end{aligned} \tag{47}$$

or

$$\begin{aligned} e_l[n] \; = \; & \sum_{p=0}^{P-1} \sum_{m=1}^{M} \tilde{a}^*_{p,m} \exp\left(-j2\pi \frac{[(PM-1)-n]p}{P}\right) \\ & s\left[(PM-1)-n-(m-1)P\right] \\ & n=0,\, 1,\, \ldots,\, PM-1, \quad l=0,\, 1,\, \ldots,\, L-1 \end{aligned} \tag{48}$$

where:

$$\tilde{a}^*_{p,m} = a^*_{p,m} \exp\left(-j2\pi \frac{[L-l-1]p}{LP}\right) = a^*_{p,m} \times \alpha_{p,l} \tag{49}$$

This equation is exactly the same as Equation (11) for $N = 1$ where $a^*_{p,m}$ is replaced by $\tilde{a}^*_{p,m}$. Thus, $e_l[n]$ can be implemented as Equation (11) based on proposed method which has been introduced in Section 4 and shown in Figure 2. The only difference is that the subcarrier code words are different. Using the fact that $\tilde{a}^*_{p,m} = a^*_{p,m} \times \alpha_{p,l}$, $e_l[n]$ can be implemented as shown in Figure 7.

According to Figure 7 the proposed structure for MCPC pulse compression is based on the implementation of polyphase components which has the same structure with different coefficients at the last stage. Therefore, to implement all of the polyphase components, *we can implement the first three blocks of Figure 7 only one time*, and then combine its outputs with different coefficients as shown in Figure 8. It is obvious that this approach is more efficient. In the next section we will show that based on this representation the computational complexity is significantly reduced.

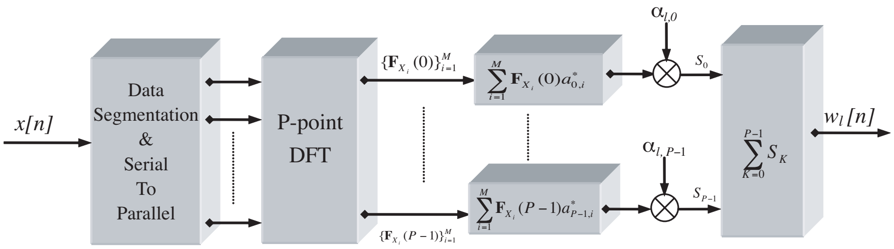

x[n]
Data Segmentation & Serial To Parallel
P-point DFT
$\{\mathbf{F}_{X_i}(0)\}_{i=1}^{M}$
$\{\mathbf{F}_{X_i}(P-1)\}_{i=1}^{M}$
$\sum_{i=1}^{M}\mathbf{F}_{X_i}(0)a_{0,i}^{*}$
$\sum_{i=1}^{M}\mathbf{F}_{X_i}(P-1)a_{P-1,i}^{*}$
$\alpha_{l,0}$
S_0
$\alpha_{l,P-1}$
S_{P-1}
$\sum_{K=0}^{P-1}S_K$
$w_l[n]$

Figure 7. Computation of the l-th polyphase component of the oversampled matched filter, $H(z)$.

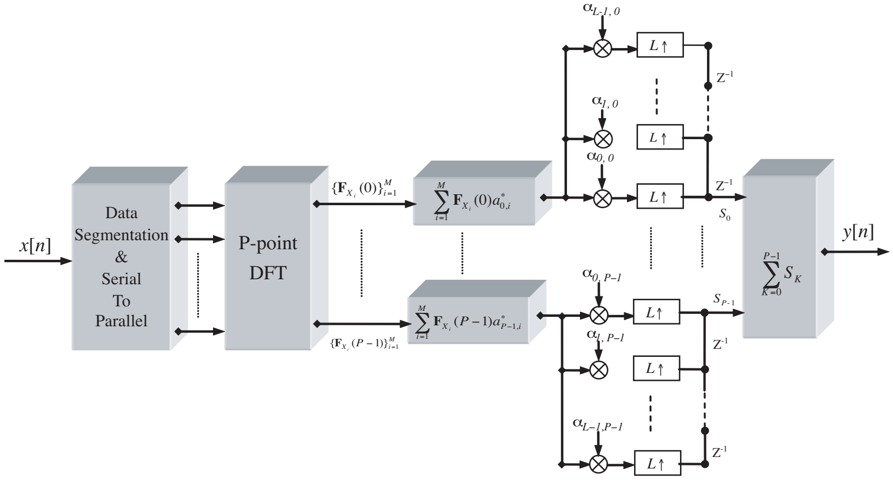

$\alpha_{L-1,0}$
$\alpha_{l,0}$
$\alpha_{0,0}$
$L\uparrow$
Z^{-1}
S_0
x[n]
Data Segmentation & Serial To Parallel
P-point DFT
$\{\mathbf{F}_{X_i}(0)\}_{i=1}^{M}$
$\{\mathbf{F}_{X_i}(P-1)\}_{i=1}^{M}$
$\sum_{i=1}^{M}\mathbf{F}_{X_i}(0)a_{0,i}^{*}$
$\sum_{i=1}^{M}\mathbf{F}_{X_i}(P-1)a_{P-1,i}^{*}$
$\sum_{K=0}^{P-1}S_K$
y[n]
$\alpha_{0,P-1}$
$\alpha_{l,P-1}$
$\alpha_{L-1,P-1}$
S_{P-1}

Figure 8. Polyphase implementation of the oversampled MCPC pulse compression filter.

7. COMPARING DIFFERENT METHODS FROM COMPUTATIONAL COMPLEXITY POINT OF VIEW

Two substituting methods were given for the realization of the matched filter to compress the MCPC signals in Sections 4 and 6. The first method that is based on the Equation (34), has been proved in this paper that is exactly equivalent to the matched filter, and the second method or the structure of Figure 8 is the polyphase implementation of the oversampled matched filter based on the first proposed structure. One of the advantages of the above two suggested methods is the lower computational complexity compared to the straight realization of the matched filter. In this section, these approaches are compared to each other from the computational complexity point of view.

For this purpose assume that the MCPC signal contains P subcarriers and each subcarrier is modulated by a code of length M. In this case the number of required basic computational operations for different methods can be calculated as follows:

7.1. Conventional Implementation of the Matched Filter

In this method, and with the assumption of using the sampling rate $f_s = NP/t_c$, the matched filter is implemented as an FIR filter with NPM complex coefficients. To calculate one output sample, NPM complex multiplications and NPM-1 complex additions are required. On the other hand, we know that for calculation of one complex multiplication, 4 real multiplications and 2 real additions are needed, and for the calculation of one complex addition, 2 real additions are required. So the required total number of calculations is $4NPM$ real multiplications and $4NPM$-2 real additions. If the total number of range cells is equal to N_R, these calculations are also multiplied by this number. On the average the required number of computational operation for compressing signal in each range cells is equal to:

- *Number of real multiplications: 4MNP.*
- *Number of real additions: 4MNP-2.*

7.2. First Proposed Method of Compression (Direct Implementation of DFT-Based Method)

If the total number of range cells is equal to N_R, for calculation of all of the output samples, N_R DFT of length NP is needed (for the cells with NP cells apart, only computation of one new DFT is needed and other DFTs are the same).Also P_{NR} compressing operations with FIR filter of length M are required, and finally, N_R additions of length P are also needed. On the other hand if we use the fact that only P out of NP DFT points need to be computed, then computational complexity could be reduced via efficient algorithms such as Goertzel, Boncelet, Prunning and transform decomposition algorithms [20, 22]. For example to compute P out of NP-DFT points, the Goertzel algorithm requires $4P(NP + 1)$ real additions and $2P(NP + 2)$ real multiplications (with no assumption on the length of the data) [20]. If the modulated codes on each of the subcarriers are complex, then FIR filters multiplications are complex and so M complex multiplications and $M - 1$ complex additions are needed.

Thus by using Goertzel algorithm for DFT calculation, the average number of computational operations for compressing signal in each of the range cells is equal to:

- *Number of real multiplications:* $2P(NP+2)+4MP$.
- *Number of real additions:* $4P(NP+1)+4MP-2$.

Also if the modulated codes on the subcarriers are real, FIR filters multiplications are real and the input signal is complex, so M complex to real multiplications and $M-1$ complex additions are required. In this case the average number of computations for compressing signal in each of the range cells is equal to:

- *Number of real multiplications:* $2P(NP+2)+2MP$.
- *Number of real additions:* $4P(NP+1)+3MP-2$.

On the other hand if NP is a power of 2 then computation of NP-point DFT can be calculated via a fast Fourier transform algorithm (FFT). In this case there are different algorithms with different computational complexity that are more efficient than Goertzel algorithm [22]. For example, for computation of NP point FFT by 3-butterfly split-radix algorithm, $3NP \times \log_2(NP) - 3NP + 4$ real additions and $NP \times \log_2(NP) - 3NP + 4$ real multiplications are required (to compute all NP outputs) [22]. Based on using this algorithm for FFT calculation, the average number of computational operations for compressing signal in each of the range cells in the case of complex modulated codes is equal to:

- *Number of real multiplications:* $NP\log_2(NP)-3NP+4MP+4$.
- *Number of real additions:* $3NP\log_2(NP)-3NP+4MP+2$.

and in the case of real codes it is equal to:

- *Number of real multiplications:* $NP\log_2(NP)-3NP+2MP+4$.
- *Number of real additions:* $3NP\log_2(NP)-3NP+3MP+2$.

7.3. Second Proposed Method of Compression (Polyphase Implementation of the DFT-based Method

For this method that uses the sampling rate $f_s = P/t_c$ and implement the oversampled matched filter using polyphase decomposition, if the oversampling ratio is equal to L and the modulated codes on each of the subcarriers are complex, the required average number of calculations for compressing signal in each of the range cells is:

- *Number of real multiplications:* $2P(P+2)+4(M+1)P$.
- *Number of real additions:* $4P(P+1)+(4M+2)P-2$.

In the above calculations extra adders following the expanders are not counted, because the signal $S_K\ [n]$, $K = 0, 1, \dots , P-1$ which has been shown in Figure 8 is obtained merely by interlacing the outputs of these expanders consecutively. So this method requires only P extra complex multiplications than the direct implementation of DFT based method with $N = 1$ for compression of signal in each of the range cells.

In this method if the modulated codes on the subcarriers are real, the number of calculations for compressing signal in each of the range cells decreases to:

- *Number of real multiplications:* $2P(P + 2) + (2M + 4)P$.
- *Number of real additions:* $4P(P + 1) + (3M + 2)P - 2$.

Also in this method if P is a power of 2 then computation of P-point DFT can be done by FFT algorithms such as split-radix algorithm. If we use 3-butterfly split-radix algorithm, the average number of computational operations for compressing signal in each of the range cells in the case of complex modulated codes is equal to:

- *Number of real multiplications:* $P\log_2(P) - 3P + 4(M + 1)P + 4$.
- *Number of real additions:* $3P\log_2(P) - 3P + (4M + 2)P + 2$.

and in the case of real codes it is equal to:

- *Number of real multiplications:* $P\log_2(P) - 3P + (2M + 4)P + 4$.
- *Number of real additions:* $3P\log_2(P) - 3P + (3M + 2)P + 2$.

Tables 1 and 2 summarize the above calculations for computational complexity of different methods.

Table 1. Number of real multiplications required for compressing signal in each of the range cells.

	conventional	DFT Based Method		Polyphase implementation of the DFT based method	
		NP≠2^k	NP=2^k	NP≠2^k	NP=2^k
Complex codes	4MNP	$2P(NP+2)$ $+4MP$	$NP\log_2(NP)-3NP$ $+4MP+4$	$2P(P+2)+$ $4(M+1)P$	$P\log_2(P)+$ $(4M+1)P+4$
Real codes	4MNP	$2P(NP+2)+$ $2MP$	$NP\log_2(NP)-3NP$ $+2MP+4$	$2P(P+2)+$ $(2M+4)P$	$P\log_2(P)+$ $(2M+1)P+4$

Figures 9 and 10 show the ratio of computational complexity of the conventional method to the first proposed method with sampling frequency $4P/t_c$. In Figure 9 the computational load ratio of the conventional implementation of the matched filter and the DFT based method are shown versus P, for $M = 50$ (it is assumed that Goertzel algorithm used for DFT calculation). As this figure shows when

Table 2. Number of real additions required for compressing signal in each of the range cells.

	conventional	DFT Based Method		Polyphase implementation of the DFT based method	
		NP≠2^k	NP=2^k	NP≠2^k	NP=2^k
Complex codes	4MNP-2	$4P(NP+1)+$ $4MP-2$	$3NP\log_2(NP)-3NP$ $+4MP+2$	$4P(P+1)+$ $(4M+2)P-2$	$3P\log_2(P)$ $+(4M-1)P+2$
Real codes	4MNP-2	$4P(NP+1)+$ $3MP-2$	$3NP\log_2(NP)-3NP$ $+3MP+2$	$4P(P+1)+$ $(3M+2)P-2$	$3P\log_2(P)$ $+(3M-1)P+2$

the codes on the subcarriers are complex, computational load of the proposed method is lower (up to 3.7 times) than that of the conventional method and if the codes are real, the number of required additions are up to 4.75 times less and the number of multiplications are up to 7.1 times less than that of the conventional method. Thus, the DFT based method gives significant saving especially for real modulated codes. Furthermore the complexity reduction is more than that of depicted in Figure 9 when P is power of 2 and DFT can be evaluated using FFT algorithms.

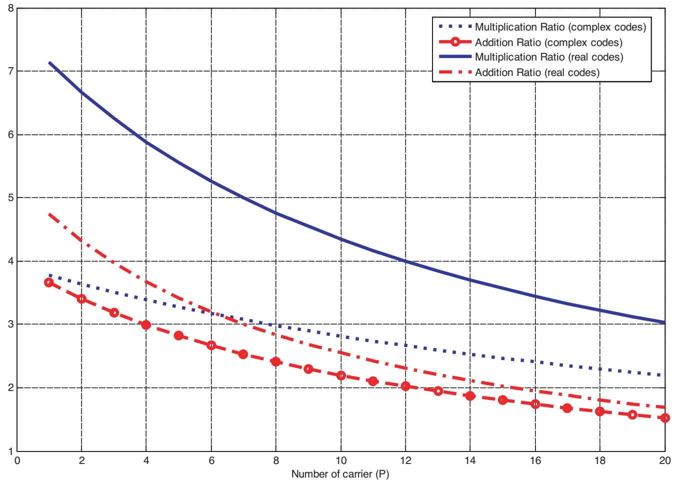


Figure 9. Computational load ratio of conventional and DFT based methods versus number of subcarriers ($f_s = 4P/t_c$, $M = 50$).

In Figure 10 the effect of code length on computational complexity (the number of required multiplications) is studied for both methods (conventional and DFT based) in the case of $P = 5$. This figure

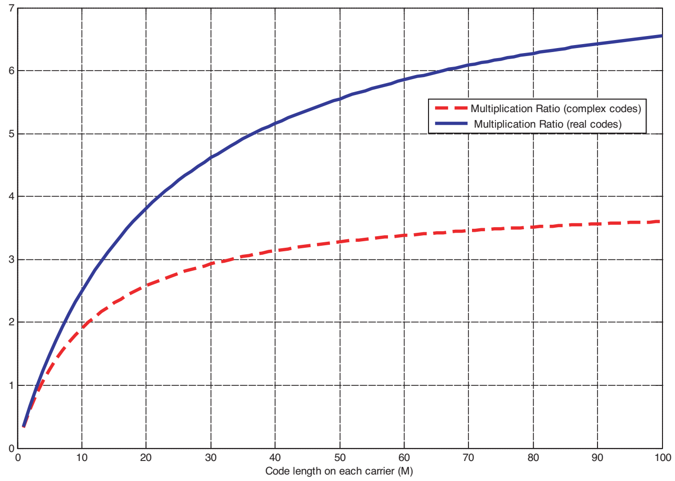


Figure 10. Computational load ratio of conventional and DFT based methods versus code length ($f_s = 4P/t_c,\ P = 5$).

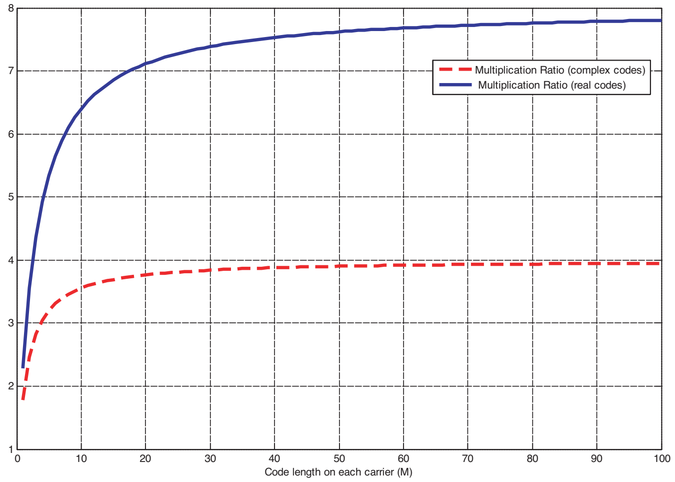


Figure 11. Computational load ratio of conventional and FFT based methods versus code length ($f_s = 4P/t_c,\ P = 4$).

shows that as the code length increases, the computational complexity difference between the two methods increases. If the code length is equal to 50, the computational complexity of the DFT based method is 5.5 times less for real codes and for the complex codes is approximately

3.2 times less than that of the conventional method. Figure 11 is the same as Figure 10 but for $P = 4$ where the FFT algorithms can be used for calculation of DFT. As this figure show if the code length is equal to 50, the computational complexity of the proposed method is 7.4 times less for real codes and for the complex codes is approximately 3.85 times less than that of the conventional method.

Figure 12 shows the ratio of computational load (total number of additions and multiplications) of the first proposed method to the second suggested one versus P, for $M = 10$. The sampling rate for the first method or direct implementation of the DFT based method is assumed to be $4P/t_c$ and the oversampling factor for the second method or polyphase implementation of the DFT based method is $L = 4$. According to this figure increasing the number of subcarriers results in increase of difference in computational load and for large value of P the polyphase implementation of the DFT based method is 4 times more efficient than the direct implementation of DFT based method.

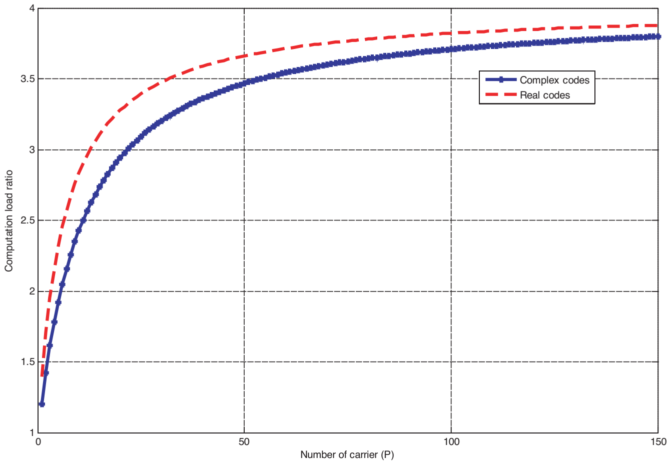


Figure 12. The ratio of computational load of the first proposed method ($f_s = 4P/t_c$) to the second one ($f_s = P/t_c$) versus number of subcarriers for $M = 10$.

8. CONCLUSIONS

In this paper a new approach for compressing the MCPC signals has been proposed. The proposed method that is based on DFT and is mathematically equivalent to the conventional method has two main advantages in comparison with the conventional approach. The first advantage is the significant reduction in computational complexity that the amount of reduction depends on signal parameters. The other one is the possibility of phase and amplitude estimation for different subcarriers frequencies that could be used for power management in radar systems in various scenarios. This is a new concept in radar systems and comprises our future research.

REFERENCES

1. Lee, K.-C., C.-W. Huang, and M.-C. Fang, "Radar target recognition by projected features of frequency diversity RCS," *Progress In Electromagnetics Research*, PIER 81, 121–133, 2008.
2. Jung, J.-H. and H.-T. Kim, "Comparisons of for feature extraction approaches based on Fisher's linear discriminant criterion in radar target recognition," *Journal of Electromagnetic Waves and Applications*, Vol. 21, No. 2, 251–265, 2007.
3. Shi, Z. G., S. Qiao, and K. S. Chen, "Ambiguity functions of direct chaotic radar employing microwave chaotic Colpitts oscillator," *Progress In Electromagnetics Research*, PIER 77, 1–14, 2007.
4. Jankiraman, M., B. J. Wessels, and P. V. Genderen, "Design of a multi-frequency FMCW radar," *Proc. of 28th European Microwave Conference*, Vol. 1, 584–589, Amsterdam, October 1998.
5. Jankiraman, M., B. J. Wessels, and P. V. Genderen, "System design and verification of the PANDORA multifrequency radar," *Proc. of 5th Int. Conference on Radar Systems*, Brest, France, 1999.
6. Jankiraman, M., B. J. Wessels, and P. V. Genderen, "Ambiguity analysis of pandora multifrequency FMCW/SFCW radar," *Proc. of the IEEE 2000 International Radar Conference*, 35–41, May 2000.
7. Levenon, N., "Multifrequency radar signals," *Proc. of the IEEE 2000 International Radar Conference*, 683–688, May 2000.
8. Levenon, N., "Multifrequency complementary phase-coded radar signal," *IEE Proc. Radar, Sonar Navig.*, Vol. 147, No. 6, 276–284, December 2000.
9. Mozeson, E. and N. Levenon, "Multicarrier radar signals with low peak-to-mean envelope ratio," *IEE Proc. Radar, Sonar Navig.*, Vol. 150, No. 2, 71–77, April 2003.
10. Levenon, N. and E. Mozeson, "Multicarrier radar signal — Pulse train and CW," *IEEE Trans. AES*, Vol. 38, No. 2, 707–720, April 2002.

11. Levenon, N., "Train of diverse multifrequency radar pulses," *Proc. of the 2001 IEEE Radar Conf.*, 93–98, May 2001.

12. Prasad, N. N. S. S. R. K., V. Shameem, U. B. Desai, and S. N. Merchant, "Improvement in target detection performance of pulse coded Doppler radar based on multicarrier modulation with fast Fourier transform (FFT)," *IEE Proc. Radar, Sonar Navig.*, Vol. 151, No. 1, 11–17, February 2004.

13. Singh, S. P. and K. S. Rao, "Pulse train of multicarrier complementary phase coded radar signal for favourable autocorrelation and ambiguity function," *Proc. International Conference on Systemics, Cybernetics, and Informatics (ICSCI 2005)*, 81–86, India, January 2005.

14. Franken, G. E. A., H. Nikookar, and P. Van Gendern, "Doppler tolerance of OFDM-coded radar signals," *Proc. of the 3rd European Radar Conference*, 108–111, September 2006.

15. Levenon, N. and E. Mozeson, *Radar Signals*, John Wiley & Sons, Inc., New York, 2004.

16. Kim, J.-H. and Y.-H. You, "PILOT-frequency tracking method for ultra-wideband receivers," *Progress In Electromagnetics Research*, PIER 82, 65–75, 2008.

17. Shim, E.-S. and Y.-H. You, "Parameter estimation and error reduction in multicarrier system by time-domain spreading," *Progress In Electromagnetics Research B*, Vol. 7, 1–12, 2008.

18. Nam, S.-H., J.-S. Yoon, and H.-K. Song "An elaborate frequency offset estimation for OFDM systems," *Progress In Electromagnetics Research Letters*, Vol. 4, 33–41, 2008.

19. Park, S. H. and H. T. Kim, "Stepped-frequency ISAR motion compensation using particle SWARM optimization with an island model," *Progress In Electromagnetics Research*, PIER 85, 25–37, 2008.

20. Oppenheim, A. V. and R. W. Schafer, *Discrete Time Signal Processing*, Prentice-Hall, 1999.

21. Vaidyanathan, P. P., *Multirate Systems and Filter Banks*, Prentice-Hall, Englewood Cliffs, NJ, 1993.

22. Sorensen, H. V. and C. S. Burms, "Efficient computation of the DFT with only a subset of input or output points," *IEEE Trans. on Signal Processing*, Vol. 41, No. 3, 1184–1200, March 1993.