Optical Fiber Technology 30 (2016) 72–80

Contents lists available at ScienceDirect

Optical Fiber Technology

www.elsevier.com/locate/yofte

Generalized optical code construction for enhanced and Modified Double Weight like codes without mapping for SAC–OCDMA systems

Soma Kumawat *, M. Ravi Kumar

MNIT Jaipur, Rajasthan, India

ARTICLE INFO

Article history:
Received 14 December 2015
Revised 19 February 2016
Accepted 13 March 2016

Keywords:
SAC–OCDMA
EDW
MDW
Mapping
Correlation
Balanced detection
Direct detection
BER
PIIN

ABSTRACT

Double Weight (DW) code family is one of the coding schemes proposed for Spectral Amplitude Coding–Optical Code Division Multiple Access (SAC–OCDMA) systems. Modified Double Weight (MDW) code for even weights and Enhanced Double Weight (EDW) code for odd weights are two algorithms extending the use of DW code for SAC–OCDMA systems. The above mentioned codes use mapping technique to provide codes for higher number of users. A new generalized algorithm to construct EDW and MDW like codes without mapping for any weight greater than 2 is proposed. A single code construction algorithm gives same length increment, Bit Error Rate (*BER*) calculation and other properties for all weights greater than 2. Algorithm first constructs a generalized basic matrix which is repeated in a different way to produce the codes for all users (different from mapping). The generalized code is analysed for *BER* using balanced detection and direct detection techniques.

1. Introduction

Optical Code Division Multiple Access (OCDMA) systems share a transmission media between different users at the same time by providing unique codes to each user [1]. The concept behind this is to transmit the code in place of sending single one and same length zero sequence in place of sending single zero. All the codes have same length but unique pattern for all users. The patterns are defined according to chosen coding scheme. The autocorrelation function of a code gives a high peak. A high peak may also be generated due to Multiple Access Interference (MAI). At receiver, a high peak is generated due to detection of desired code. On detection of a high peak, the receiver assumes the code was transmitted.

Spectral Amplitude Coding–Optical Code Division Multiple Access (SAC–OCDMA) systems are a type of the OCDMA technique which use the wavelength domain for its coding [2]. SAC–OCDMA systems completely eliminate the effect of MAI. The effect of MAI is eliminated due to spectral coding with ideal in-phase cross correlation and its detection techniques [3,4].

There are various coding schemes that have been developed for SAC–OCDMA systems such as, Modified quadratic congruence code for prime weights [4], extended perfect difference code family for limited code weights [5], Multi-Service code design for fixed weight code with variable number of users in a basic matrix and uses mapping to obtain codes for increasing number of users [6], vector combinatorial code using XOR subtraction detection and mapping [7], random diagonal code which uses code segment and data segment to generate the code, with a trade-off between crosscorrelation and weight at the code segment group [8]. DW code [9] is designed for a weight of 2 only. There are further extensions of DW code which are known as Modified Double Weight (MDW) code [9] and Enhanced Double Weight (EDW) code [10]. MDW code and EDW code have only even weights and only odd weights greater than two respectively. Finally, some of these codes are generated for some specific integer numbers, and some other codes have crosscorrelation and mapping limitations.

In DW code family to generate a code, first step is to construct a basic matrix. Depending on the number of users required in the code family, the basic matrix is repeated diagonally. Repetition of basic matrix to construct the code for higher number of users is known as Mapping technique [6,7,9–11]. Due to mapping, increment of code length is not constant. Its increment is dependent on the size of basic matrix and required number of additional users. For example, if basic matrix has size (3×9) where users are 3 and code length is 9. For 3 additional users, code length becomes 18. All the three coding techniques have maintained crosscorrelation value of atmost 1 ($\lambda_c \leqslant 1$) between all the N users.

* Corresponding author.
E-mail address: somakumawat@yahoo.com (S. Kumawat).

http://dx.doi.org/10.1016/j.yofte.2016.03.004
1068-5200/

Crosscorrelation value with-in basic matrix is $\lambda_c = 1$ and across basic matrices is $\lambda_c = 0$. Even though mapping and crosscorrelation constraints are similar for all three codes, they have different code construction algorithms, length and other properties.

Reconstruction of EDW code for higher weights in [12], 2D EDW code [13] and 2D MDW code [14] are reported for spectral/spatial domains for SAC–OCDMA systems. Performance improvement of SAC–OCDMA systems using different detection techniques is given in [11,15–17].

A new generalized algorithm to construct EDW and MDW like codes without mapping for any weight greater than 2 is proposed. The code construction is independent of mapping technique. It maintains crosscorrelation value of atmost 1 ($\lambda_c \leqslant 1$) between all the N users. A single code construction algorithm is designed for all weights greater than 2. Length and other properties are same as that of EDW and MDW. For each additional user, code length increment is constant. For example, if code length for 3 users is 9 then for 4 users it is 12, for 5 users it is 15 and so on (increment of 3 for each user).

The rest of the paper is organized as follows. Section 2 proposes a generalized algorithm to construct MDW and EDW like codes without mapping. Construction of odd and even weight codes using the algorithm is also explained with examples in Section 2. System setup for the constructed code is described in Section 3. Setup involves the designing of encoder and decoder for balanced detection. Bit Error Rate (BER) for balanced detection and Direct detection (DD) are analysed in Sections 4 and 5 respectively. Numerical results are discussed in Section 6, and the paper is concluded in Section 7.

2. Code construction

The Generalized code construction algorithm is explained below. The section is divided into 2 subSections 2.1 explains the code construction algorithm. 2.2 contains the code construction examples for odd and even weights.

2.1. Algorithm

The value of weight (W) and number of users (N) are chosen. Code length is given as $L = N * (W - 1)$ for W and N.

Basic Matrix (M) of size $2 \times (W - 1)$ is constructed as follows.

$$M = \begin{bmatrix} R_1 \\ R_2 \end{bmatrix} = \begin{bmatrix} \lfloor \frac{W-2}{2} \rfloor 0s & \lfloor \frac{W+1}{2} \rfloor 1s \\ \lfloor \frac{W}{2} \rfloor 1s & \lfloor \frac{W-1}{2} \rfloor 0s \end{bmatrix}_{2\times(W-1)} \tag{1}$$

The complete code set is represented by matrix U of size $N \times L$ for N users. The construction of U involves 3 steps in which an intermediary matrix U^* is first constructed. M is repeated $N - 1$ times in U^* as shown below.

$$U^* = \begin{bmatrix} R_1 & .. & .. & .. & .. & .. \\ R_2 & R_1 & .. & .. & .. & \vdots \\ \vdots & R_2 & R_1 & .. & .. & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & R_1 & \vdots \\ .. & .. & .. & .. & R_2 & .. \end{bmatrix} \tag{2}$$

To completely fill all columns, basic matrix rows R_1 and R_2 are added to last row and first row of last column of matrix U^* respectively as shown below.

$$U^{**} = \begin{bmatrix} R_1 & .. & .. & .. & .. & & R_2 \\ R_2 & R_1 & .. & .. & .. & .. & \vdots \\ \vdots & R_2 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & R_1 & \vdots \\ .. & .. & .. & .. & .. & R_2 & R_1 \end{bmatrix}_{N\times L} \tag{3}$$

The complete code set is obtained by filling up empty places in U^{**} with zeros.

$$U = \begin{bmatrix} R_1 & 0 & 0 & .. & .. & .. & R_2 \\ R_2 & R_1 & 0 & .. & .. & .. & 0 \\ 0 & R_2 & 0 & .. & .. & .. & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & R_1 & 0 \\ 0 & 0 & 0 & .. & .. & R_2 & R_1 \end{bmatrix}_{N\times L} \tag{4}$$

Algorithm is stated as:

- Choose W, N and calculate the code length L.
- Construct M as per Eq. (1).
- Repeat M in U^* as per Eq. (2).
- R_1 and R_2 are added to U^{**} as per Eq. (3) and empty places in U^{**} are filled with zeros to complete code construction for all users.

Flowchart of algorithm is shown in Fig. 1.

2.2. Code construction examples

Let us consider the following cases for different weights.

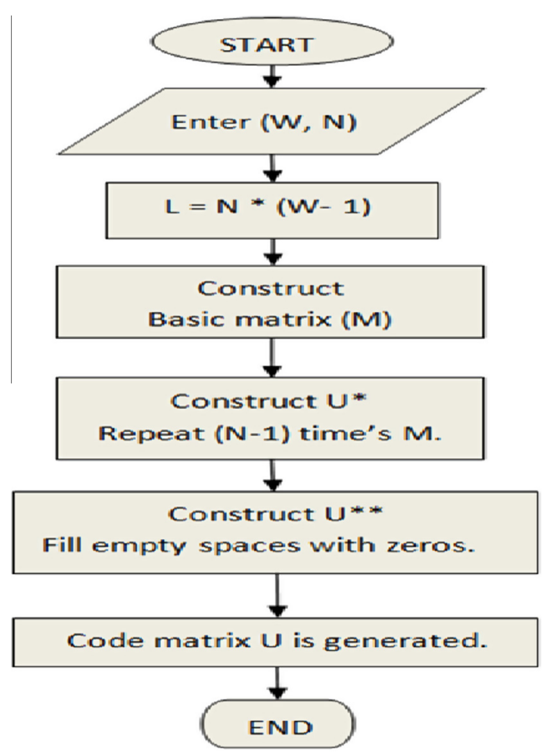

Fig. 1. Flow chart for proposed code construction.