Matrix Decomposition Suitable for FPGA Implementation of NCSP-OFDM

Hikaru Kawasaki, Zhiheng Zhou, Masaya Ohta and Katsumi Yamashita
Department of Electrical and Information Systems,
Graduate School of Engineering, Osaka Prefecture University, Japan
Email: {ota,yamashita}@eis.osakafu-u.ac.jp

Abstract: N-continuous symbol padding orthogonal frequency division multiplexing (NCSP-OFDM) is a modulation technique that achieves a lower-sidelobe transmission signal by precoding data symbols. However, the precoding requires a wide dynamic range arithmetic. This study considers two decompositions of the precoder matrix for a low-end field-programmable gate array (FPGA) without DSP blocks. Results of numerical experiments and a test design confirm that the singular value decomposition is suitable for FPGA implementation.

Keywords— **OFDM, sidelobe suppression, precoding, FPGA implementation, singular value decomposition**

1. Introduction

Orthogonal frequency division multiplexing (OFDM) has been adopted in several telecommunications technologies owing to the advantages of fast data transmission and robustness against multipath fading. However, one problem associated with OFDM is that high sidelobes arise from the discontinuity between adjacent OFDM symbols. Various methods of sidelobe suppression have been proposed [1]–[4] including N-continuous OFDM [5] that is a modulation technique in which the OFDM symbols are continuously connected with higher-order derivatives. This technique does not require an extended guard interval nor the insertion of a wide guard band or cancellation carriers. However a disadvantage of N-continuous OFDM is that it forces the receiver to use an iterative algorithm that increases the computational complexity, in order to eliminate the correction symbol inserted for the continuous connection of OFDM symbols. We have proposed N-continuous symbol padding OFDM (NCSP-OFDM) [6], in which the correction symbol is added only into the guard interval to enable the seamless connection of OFDM symbols. The receiver for NCSP-OFDM does not require the iterative algorithm because the correction symbol does not leak to the data block following the guard interval.

The precoder matrix used in this method is derived from a matrix whose elements are integers ranging from 1 to $(K/2)^N$ ($K = 300$ and $N = 3$ in Ref.[5]), which represents a wide dynamic range. Precoding using this matrix requires floating-point arithmetic, which consumes a large number of slices on a field-programmable gate array (FPGA). Although a DSP block in an FPGA performs floating-point arithmetic without other resource consumption, there are only a few DSP blocks in a low-end FPGA device, and these blocks should be available for other functional blocks such as the fast Fourier transform and finite impulse response filters.

We consider here two decompositions of the precoder matrix of NCSP-OFDM for FPGA implementation and clarify the matrix decomposition suitable for a low-end FPGA without DSP blocks.

2. NCSP-OFDM

The NCSP-OFDM signal is written as

$$s(t) = \sum_{i=0}^{\infty} s_i(t - i(T_s + T_g)), \tag{1}$$

where T_s is the OFDM symbol duration and T_g is the guard interval length. The i-th NCSP-OFDM symbol $s_i(t)$ can be expressed with the following formula:

$$s_i(t) = \begin{cases} \sum_{k\in\mathcal{K}} \bar{d}_{i,k} e^{j2\pi \frac{k}{T_s} t}, & -T_g \le t < 0, \\ \sum_{k\in\mathcal{K}} d_{i,k} e^{j2\pi \frac{k}{T_s} t}, & 0 \le t < T_s, \end{cases} \tag{2}$$

where $\bar{\mathbf{d}}_i = [\bar{d}_{i,k_0}, \ldots, \bar{d}_{i,k_{K-1}}]^T$ is the result of precoding data symbol $\mathbf{d}_i = [d_{i,k_0}, \ldots, d_{i,k_{K-1}}]^T$, $\mathcal{K} = \{k_0, \ldots, k_{K-1}\}$ is a set of data subcarrier indices, and K is the number of subcarriers.

The symbol $s_i(t)$ satisfies the following constraints so that the symbols in the guard interval are continuous with both the preceding and following symbols at the connecting boundaries until the N-th order derivative.

$$\frac{d^n}{dt^n} s_i(t) \bigg|_{t=-T_g} = \frac{d^n}{dt^n} s_{i-1}(t) \bigg|_{t=T_s}, \tag{3}$$

$$\frac{d^n}{dt^n} s_i(t) \bigg|_{t=-0} = \frac{d^n}{dt^n} s_i(t) \bigg|_{t=+0}, \tag{4}$$

where $n \in \{0, \ldots, N\}$. The constraints can be expressed in matrix form as

$$\begin{bmatrix} \mathbf{A\Phi} \\ \mathbf{A} \end{bmatrix} \bar{\mathbf{d}}_i = \begin{bmatrix} \mathbf{O}_{(N+1)\times K} \\ \mathbf{A} \end{bmatrix} \mathbf{d}_i + \begin{bmatrix} \mathbf{A} \\ \mathbf{O}_{(N+1)\times K} \end{bmatrix} \mathbf{d}_{i-1}, \tag{5}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ k_0 & k_1 & \ldots & k_{K-1} \\ \vdots & \vdots & \ddots & \vdots \\ k_0^N & k_1^N & \ldots & k_{K-1}^N \end{bmatrix}, \tag{6}$$

$\mathbf{\Phi} = \mathrm{diag}(e^{j\phi k_0}, \ldots, e^{j\phi k_{K-1}})$, and $\phi = -2\pi T_g/T_s$.

We have proposed a method to precode the symbol $\mathbf{d}_i$ to $\bar{\mathbf{d}}_i$ such that

$$\bar{\mathbf{d}}_i = \mathbf{d}_i + \mathbf{w}_i, \tag{7}$$

$$\mathbf{w}_i = -\mathbf{P}\mathbf{d}_i + \mathbf{P}\mathbf{\Phi}^H \bar{\mathbf{d}}_{i-1}. \tag{8}$$

Here,

$$\mathbf{P} = \begin{bmatrix} \mathbf{A\Phi} \\ \mathbf{A} \end{bmatrix}^{\dagger} \begin{bmatrix} \mathbf{I}_{N+1} \\ \mathbf{O}_{N+1} \end{bmatrix} \mathbf{A\Phi}\,, \tag{9}$$

where $\mathbf{X}^{\dagger}$ is the Moore–Penrose pseudoinverse of $\mathbf{X}$. This precoding satisfies (3) and (4), and the inserted symbol $\mathbf{w}_i$ has the smallest power in a Euclidean sense. The matrix $\mathbf{P}$ is referred to here as the precoder matrix.

3. Matrix Decompositions

The computational complexity of (8) is $O(K^2)$ in multiplications, which results in an enormous computational load. To reduce it for FPGA implementation, we consider two decompositions of the precoder matrix of NCSP-OFDM.

First, from (9), letting

$$\mathbf{P}_1 = \begin{bmatrix} \mathbf{A\Phi} \\ \mathbf{A} \end{bmatrix}^{\dagger} \begin{bmatrix} \mathbf{I}_{N+1} \\ \mathbf{O}_{N+1} \end{bmatrix}, \tag{10}$$

$$\mathbf{P}_2 = \mathbf{A\Phi}, \tag{11}$$

we can decompose $\mathbf{P}$ as

$$\mathbf{P} = \mathbf{P}_1\mathbf{P}_2. \tag{12}$$

Therefore we can rewrite (8) as

$$\mathbf{w}_i = \mathbf{P}_1(\mathbf{P}_2\mathbf{x}_i), \tag{13}$$

where $\mathbf{x}_i = -\mathbf{d}_i + \mathbf{\Phi}^H\bar{\mathbf{d}}_{i-1}$. This decomposition is referred to here as *D1*. Using this decomposition, the complexity of (8) reduces to $O(KN)$, because the sizes of $\mathbf{P}_1$ and $\mathbf{P}_2$ are $K \times (N+1)$ and $(N+1) \times K$.

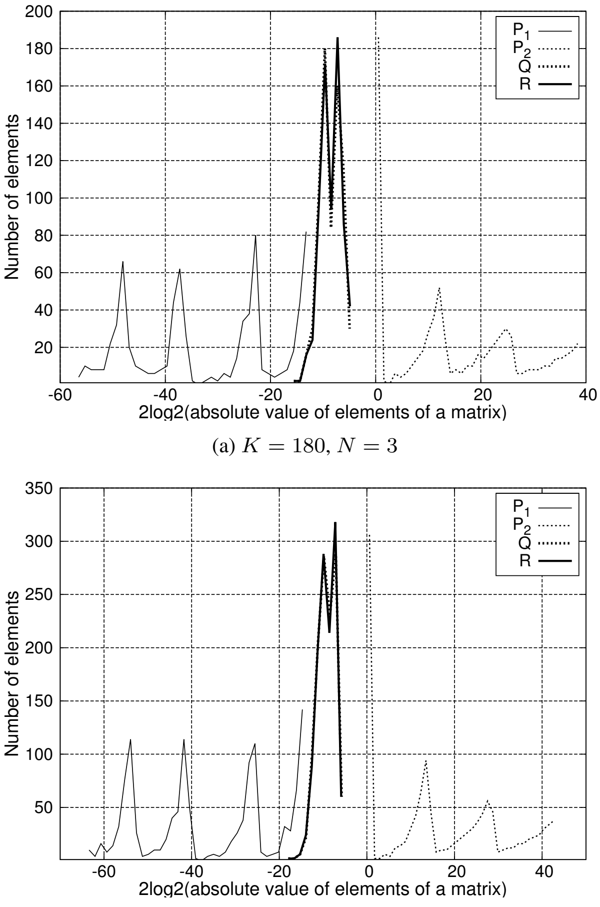
Second, we can apply singular value decomposition (SVD). Using SVD, $\mathbf{P}$ can be decomposed as $\mathbf{P} = \mathbf{USV}^H$, where $\mathbf{U}$ and $\mathbf{V}$ are $K \times K$ unitary matrices, and $\mathbf{S}$ is a $K \times K$ diagonal matrix containing the singular value of $\mathbf{P}$ in non-increasing order along its diagonal. The rank of $\mathbf{P}$ is less than or equal to $N+1$ (see Appendix), and the last $K-(N+1)$ diagonal elements of $\mathbf{S}$ are zero. Letting $\mathbf{Q}$ be a $K \times (N+1)$ matrix composed of the first $N+1$ columns of a matrix $\mathbf{US}$, and $\mathbf{R}$ be a $(N+1) \times K$ matrix composed of the first $N+1$ rows of $\mathbf{V}^H$, we can decompose $\mathbf{P}$ as

$$\mathbf{P} = \mathbf{QR}. \tag{14}$$

Therefore we can rewrite (8) as

$$\mathbf{w}_i = \mathbf{Q}(\mathbf{R}\mathbf{x}_i). \tag{15}$$

This decomposition from SVD is referred to as *D2*. Using this decomposition, the complexity of (8) also reduces to the same as that of *D1*, because the sizes of $\mathbf{Q}$ and $\mathbf{R}$ are same as those of $\mathbf{P}_1$ and $\mathbf{P}_2$. For example, the computational load reduces to 1.3% of that under the conditions in Ref.[5]: $K = 300$ and $N = 3$.



(a) $K = 180, N = 3$

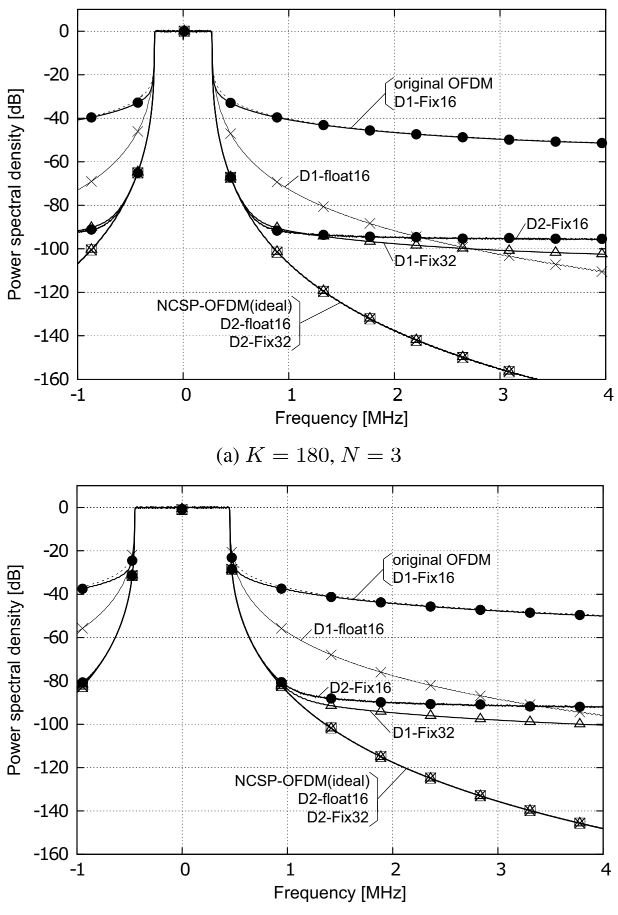
(a) $K = 300, N = 3$

Figure 1. Histogram of the powers of all the elements of the matrices $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{Q}$, and $\mathbf{R}$.

4. Analysis and Evaluation

In this section, the two decompositions *D1* and *D2* are compared in an implementation on an FPGA. First of all, we investigated the dynamic ranges of the elements in the matrices for *D1* and *D2* ($K = 180$, $N = 3$ and $K = 300$, $N = 3$). Figure 1 shows the histogram of the powers of all the elements of the matrices $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{Q}$, and $\mathbf{R}$. From these figures, we see that the dynamic ranges of $\mathbf{Q}$ and $\mathbf{R}$ are narrower than those of $\mathbf{P}_1$ and $\mathbf{P}_2$. This indicates that *D2* is better than *D1* for implementing the precoding using fixed-point arithmetic on an FPGA.

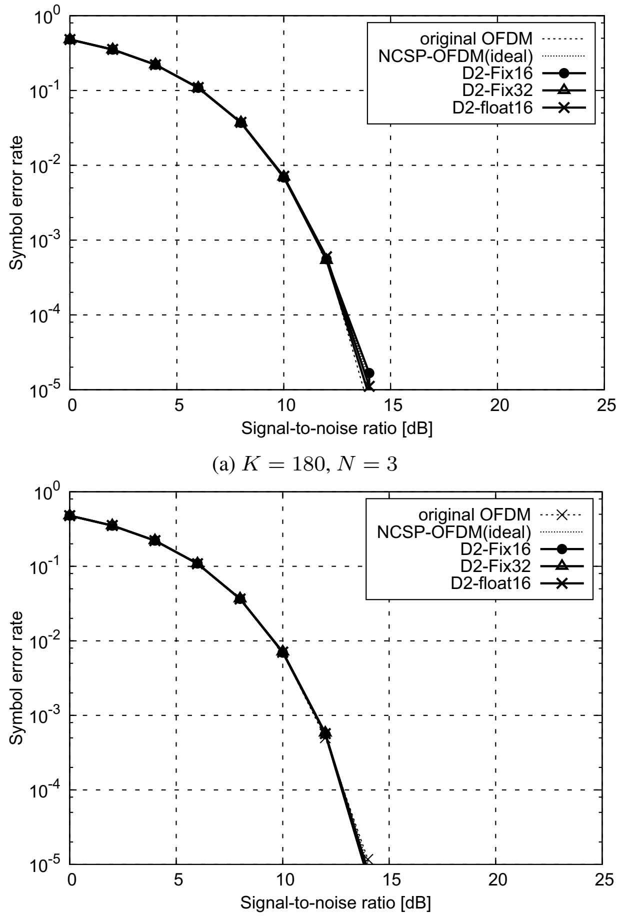
Next, we measured the power spectral density (PSD) and the symbol error rate (SER) by using floating-point and fixed-point arithmetic for a multiply-and-accumulation (MAC) operation. In calculation, we considered 16-bit fixed-point ("Fix16") MAC operations, 32-bit fixed-point ("Fix32"), and 16-bit floating-point ("float16"). The "Ideal" results are those of normal numerical experiments of NCSP-OFDM obtained using MATLAB (double-precision floating-point arithmetic). Figure 2 shows that the PSD of *D2* is superior to that of *D1*. The result is caused by the narrow dynamic range of *D2*. Figure 3 shows the SER in the additive white Gaussian noise channel. The result shows that the *D2* does not degrade the



(a) $K = 180$, $N = 3$

(b) $K = 300$, $N = 3$

Figure 2. Power spectral density using *D1* and *D2*.



(a) $K = 180$, $N = 3$

(b) $K = 300$, $N = 3$

Figure 3. Symbol error rate using *D1* and *D2*.

SER.

Finally, *D2* was implemented on an FPGA without DSP blocks, and the circuit scale was evaluated with $K = 180$ and $N = 3$. The circuit configuration of NCSP-OFDM was designed using VHDL on Xilinx ISE 14.7. The target device was a Xilinx FPGA XC6SLX45. Table 1 shows the report of the design implementation. We also present in this table the size of an IFFT for reference. For example, the occupied slices for “Fix16” was found to be 39.0% of that for “float16” and 35.3% of that for “Fix32”. This verifies that *D2* with the SVD is effective for implementing NCSP-OFDM on a low-end FPGA.

TABLE I.
Synthesis report of each block ($K = 180$, $N = 3$).

Resource	D2-Fix16	D2-float16	D2-Fix32	IFFT
Slice Registers	3,528	6,394	11,648	2,409
Slice LUTs	2,977	7,915	10,735	1,668
Occupied Slices	1,121	2,897	3,276	572
Block RAMs	0	0	0	1
BUFG/BUFGMUXs	1	4	1	1
DSP48A1s	0	0	0	12

5. Conclusions

We have considered the two decompositions of the precoder matrix of NCSP-OFDM for FPGA implementation. The SVD method was confirmed to allow precoding with a narrow dynamic range and is suitable for implementation on a low-end FPGA.

Appendix

Here we prove

$$\mathrm{rank}(\mathbf{P}) \leq N + 1. \tag{16}$$

Firstly, from the theorem of the rank of matrix product, we can obtain about the rank of $\mathbf{P} = \mathbf{P}_1\mathbf{P}_2$ such as

$$\mathrm{rank}(\mathbf{P}) \leq \min\left\{\mathrm{rank}(\mathbf{P}_1), \mathrm{rank}(\mathbf{P}_2)\right\}. \tag{17}$$

Generally, the rank of matrix is less than or equal to the number of rows or columns. So we can obtain

$$\mathrm{rank}(\mathbf{P}_1) \leq N + 1, \tag{18}$$

$$\mathrm{rank}(\mathbf{P}_2) \leq N + 1. \tag{19}$$

From (17), (18) and (19), we obtain (16).

References

[1] S. Brandes, I. Cosovic, and M. Schnell, “Reduction of out-of-band radiation in OFDM systems by insertion of cancellation carriers,” *IEEE Commun. Lett.*, vol. 10, no. 6, pp. 420–422, June 2006.

[2] I. Cosovic, S. Brandes, and M. Schnell, “Subcarrier weighting: a method for sidelobe suppression in OFDM systems,” *IEEE Commun. Lett.*, vol. 10, no. 6, pp. 444–446, June 2006.

[3] T. Weiss, J. Hillenbrand, A. Krohn, and F. K. Jondral, “Mutual interference in OFDM-based spectrum pooling system,” *Proc. IEEE Veh. Technol. Conf.,*, vol. 4, pp. 1873–1877, May. 2004.

[4] H. A. Mahmoud, and H. Arslan, “Sidelobe Suppression in OFDM-Based Spectrum Sharing Systems Using Adoptive Symbol Transition,” *IEEE Commun. Lett.*, vol. 12, no. 2, pp. 133–135, Feb. 2008.

[5] J. van de Beek, and F. Berggren, “N-continuous OFDM,” *IEEE Commun. Lett.*, vol. 13, no. 1, pp. 1–3, Jan. 2009.

[6] H. Kawasaki, M. Ohta, and K. Yamashita, “N-continuous Symbol Padding OFDM for Sidelobe Suppression,” Proc. of *IEEE ICC 2014*, pp. 5901–5906, June, 2014

[7] M. Ohta, K. Torigoe, H. Kawasaki, and Katsumi Yamashita, “Matrix Decomposition Suitable for FPGA Implementation of N-continuous OFDM,” Proc. of Int. Conf. on Information and Communication Technology Convergence *ICTC 2014*, pp. 413–415, Oct., 2014