Matrix decomposition of $N$-continuous OFDM suitable for FPGA implementation

Hikaru Kawasaki, Kumpei Torigoe, Masaya Ohta$^a$, and Katsumi Yamashita

Graduate School of Engineering, Osaka Prefecture University,
1–1 Gakuen-cho, Nakaku, Sakai, Osaka 599–8531, Japan

$^a$ota@eis.osakafu-u.ac.jp

Abstract: $N$-continuous orthogonal frequency division multiplexing is a precoding technique that achieves a lower-sidelobe transmission signal. However, the precoding method requires a wide dynamic range arithmetic. This study proposes a novel decomposition of the precoder matrix of $N$-continuous OFDM 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, $N$-continuous OFDM, FPGA

Classification: Wireless Communication Technologies

References


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, 2, 3, 4] including $N$-continuous OFDM [5], that is a precoding 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.

The size of the precoder matrix of $N$-continuous OFDM is very large, so it causes huge computational complexity. For its implementation, Ref. [5] has proposed a simple matrix decomposition from the definition of the precoder matrix and it reduce the computational complexity drastically. In the conventional decomposition, however, the matrix elements actually represent a wide dynamic range, and the precoding requires floating-point arithmetic, which consumes a large number of slice resources on a field-programmable gate array (FPGA). Although a DSP block in an FPGA performs the 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. So a narrow dynamic range and fixed-point arithmetic are suitable for a low-end FPGA.

This paper proposes a novel decomposition of the $N$-continuous OFDM precoder matrix and clarify the matrix decomposition suitable for a low-end FPGA without DSP blocks. In section 2, we explain $N$-continuous OFDM and its precoder matrix. In section 3, we mention the conventional decomposition and our proposed decomposition of the precoder matrix. In section 4, we analyse and evaluate the proposed decomposition on an FPGA implementation without DSP blocks. In section 5, we conclude this paper.

2 $N$-continuous OFDM

The key idea of $N$-continuous OFDM is precoding the set of transmit symbols in each OFDM symbols such that the baseband-equivalent OFDM signal becomes $N$-continuous. The $i$th $N$-continuous OFDM symbol $s_i(t)$ satisfies (1) and (2),

$$ s_i(t) = \sum_{k \in K} d_{i,k} e^{j2\pi k t/T_s}, \quad -T_g \leq t < T_s, $$

$$ \left. \frac{d^n}{dt^n} s_i(t) \right|_{t=-T_g} = \left. \frac{d^n}{dt^n} s_{i-1}(t) \right|_{t=T_g}, \quad \text{for } n = 0, \cdots, N, $$

where $d_{i,k}$ are the results of precoding information symbols $d_{i,k} \in C$ ($C$ is a symbol constellation), $K = \{k_0, \cdots, k_{K-1}\}$ is a set of data subcarrier indices, $K$ is the number of data subcarriers, $T_s$ is the OFDM symbol duration, $T_g$ is the guard interval length, and $N$ is the derivative order. The constraints can be expressed in matrix form as

$$ A \Phi \tilde{d}_i = A \tilde{d}_{i-1}, $$

where $A$ is a matrix of the precoding matrix and $\Phi$ is a matrix of the $N$-continuous OFDM symbols.
where $\tilde{d}_i = [d_{i,k_0}, \ldots, d_{i,k_{K-1}}]^T$ is the result of precoding transmit symbol $d_i = [d_{i,k_0}, \ldots, d_{i,k_{K-1}}]^T$,

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

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

Jaap van de Beek et al. proposed a method by which to precode the symbol $d_i$ to $\tilde{d}_i$ [5], such that

$$\tilde{d}_i = d_i + w_i,$$  
$$w_i = -P d_i + P \Phi^H \tilde{d}_{i-1}. $$

Here,

$P = (A \Phi)^+ A \Phi,$

$$= \Phi^H A^H (A A^H)^{-1} A \Phi,$$ (7)

where $(A \Phi)^+ = \Phi^H A^H (A A^H)^{-1}$ represents the Moore–Penrose pseudoinverse of $A \Phi$. This precoding satisfies (1) and (2), and the correction symbol $w_i = [w_{i,k_0}, \ldots, w_{i,k_{K-1}}]^T$ has the smallest power in a Euclidean sense. The $K \times K$ matrix $P$ is referred to here as precoder matrix.

### 3 Matrix decompositions

The computational complexity of (6) is $O(K^2)$ in multiplications due to the size of the precoder matrix $P$, which results in an enormous computational load. To solve this problem, Ref. [5] has proposed a matrix decomposition of the precoder based on the definition (7), such as

$$P_1 = \Phi^H A^H,$$ (8)

$$P_2 = (A A^H)^{-1} A,$$ (9)

and the calculation formula (6) is rewritten as

$$w_i = P_1 (P_2 x_i),$$ (10)

where $x_i = -\Phi d_i + \tilde{d}_{i-1}$. This decomposition is referred to here as $D1$. Using this decomposition, the complexity of (6) is reduced to $O(KN)$ from $O(K^2)$ because the sizes of $P_1$ and $P_2$ are $K \times (N + 1)$ and $(N + 1) \times K$, respectively. $D1$ can reduce the computational complexity drastically because $N \ll K$ is an enough choice to achieve sidelobe suppression. For example, the computational load reduces to 1.3% of that under the conditions in Ref. [5]: $K = 300$ and $N = 3$. However, $P_1$ and $P_2$ have wide dynamic ranges compared with $P$, as shown in Fig. 1 of the next section. $P_1$ and $P_2$ are constructed by $A$ whose elements are integers ranging from 1 to $(K/2)^N$, so these matrices represent wide dynamic ranges. The wide dynamic range requires floating-point arithmetic, which consumes a large number of slice resources on an FPGA implementation. So a narrow dynamic range and fixed-point arithmetic are suitable for a low-end FPGA.
Therefore, another decomposition of the precoder matrix should be considered for a narrow dynamic range. We here propose a novel decomposition by applying singular-value decomposition (SVD) [6] for a general precoder matrix. Firstly, \( P \) can be decomposed by SVD as

\[
P = U \Sigma V^H,
\]

(11)

where \( U \) and \( V \) are the \( K \times K \) matrices, \( \Sigma \) is the \( K \times K \) diagonal matrix containing the singular values of \( P \) in non-increasing order along its diagonal, and the first \( N + 1 \) diagonal elements are all equal to one and the remaining \( K - N - 1 \) diagonal elements are all equal to zero, that is,

\[
\Sigma = \begin{bmatrix}
I_{N+1} & O_{(N+1)\times(K-N-1)} \\
O_{(K-N-1)\times(N+1)} & O_{(K-N-1)}
\end{bmatrix}.
\]

(12)

Because \( P \) is idempotent (\( P = P^2 \)) and Hermitian (\( P = P^H \)), we can express the precoder (11) as

\[
P = P^2 = PP^H = U \Sigma V^H V \Sigma^H U^H.
\]

(13)

Since \( V \) is unitary (\( VV^H = I \)) and \( \Sigma \) is idempotent and Hermitian, we obtain

\[
P = U \Sigma V^H = U T T^H V^H = S S^H,
\]

(14)

where \( S = U T \) and \( T \) is defined as

\[
T = \begin{bmatrix}
I_{N+1} \\
O_{(K-N-1)\times(N+1)}
\end{bmatrix}.
\]

(15)

and \( T T^H = \Sigma \). Then we can rewrite the calculation formula (6) as

\[
w_i = S(S^H y_i).
\]

(16)

where \( y_i = -d_i + \Phi^H \tilde{d}_{i-1} \). This decomposition is referred to here as \( D2 \).

The complexity reduction of this decomposition is also same as \( D1 \) due to the size of \( S \). In contrast to \( D1 \) that requires two different matrices, \( D2 \) requires only a matrix \( S \) and can save memory space by a half.

4 Analysis and evaluation

We compared the two decompositions \( D1 \) and \( D2 \) in aspect of the dynamic ranges on an FPGA implementation. Fig. 1(a) and Fig. 1(b) show histograms of the power

![Fig. 1](image-url)
of all the elements of the matrices $P_1$, $P_2$ and $S$ along with $P$ under the condition described in Ref. [5]: 16-QAM modulation, $T_s = 1/15$ ms, $T_g = 9T_s/128$, $N = 3$. From these figures, we see that the dynamic ranges of $S$ are narrower than those of $P_1$ and $P_2$. This indicates that $D2$ is better than $D1$ for implementing the precoding using fixed-point arithmetic on an FPGA.

Next, we conducted numerical experiments and evaluated the performances of the proposed decomposition of $K = 180, 300$, respectively. Fig. 2 show the power spectral density (PSD) and the symbol error rate (SER) in the additive white Gaussian noise channel using floating-point and fixed-point. We considered 32-bit fixed-point (“Fix32”), 16-bit floating-point (“Float16”), and 16-bit fixed-point (“Fix16”) arithmetics for a multiply-and-accumulation (MAC) operation. The “Ideal” results are those of a normal numerical experiment of $N$-continuous OFDM obtained using MATLAB (double-precision floating-point arithmetic). Fig. 2(a) shows that when using “Fix32”, the SER of $D2$ is identical to that of $D1$, but Fig. 2(c) shows the PSD of $D2$ is superior to that of $D1$. For “Float16” and “Fix16”, $D2$ is superior to $D1$ in terms of both the PSD and SER. These results are influenced by the narrow dynamic range of $D2$. Fig. 2(b) and Fig. 2(d) shows the same results as Fig. 2 when using “Fix16” and “Fix32”. For “Float16”, the SER of $D2$ is identical to that of $D1$, but the PSD of $D2$ is superior to that of $D1$.

Lastly, we implemented precoding of $N$-continuous OFDM on an FPGA without DSP blocks and evaluated the circuit scale and the power consumption.
on Xilinx ISE 14.7. The target device was Xilinx Spartan-6 family. Table I(a) shows the report of the design implementation for \( K = 300 \) and \( N = 3 \). Memory including the value of the precoding matrix is placed outside the FPGA. The circuit scale for “Fix16” was smaller than other designs, for example, the occupied slices for “Fix16” was found to be 41.7% of that for “Float16” and 32.3% of that for “Fix32”. Table I(b) shows the power dissipation of the designs running at 50 MHz. The circuit scale for “Fix16” consume less power than other designs, for example, the total power consumption for “Fix16” was found to be 63.2% of that for “Float16” and 46.8% of that for “Fix32”. \( D2 \) is capable of implementing design for “Fix16”. These results verify that \( D1 \) is effective for implementing \( N \)-continuous OFDM on a low-end FPGA.

**Table I.** Evaluation of hardware implementation.

<table>
<thead>
<tr>
<th>Resource</th>
<th>Fix16</th>
<th>Float16</th>
<th>Fix32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slice Registers</td>
<td>4,472</td>
<td>8,185</td>
<td>14,631</td>
</tr>
<tr>
<td>Slice LUTs</td>
<td>3,932</td>
<td>8,089</td>
<td>13,325</td>
</tr>
<tr>
<td>Occupied Slices</td>
<td>1,241</td>
<td>2,973</td>
<td>3,847</td>
</tr>
<tr>
<td>Block RAMs</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>BUFG/BUFGMUXs</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Design</th>
<th>Fix16 (mW)</th>
<th>Float16 (mW)</th>
<th>Fix32 (mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quiescent</td>
<td>66</td>
<td>67</td>
<td>69</td>
</tr>
<tr>
<td>Dynamic</td>
<td>58</td>
<td>125</td>
<td>196</td>
</tr>
<tr>
<td>Total</td>
<td>124</td>
<td>196</td>
<td>265</td>
</tr>
</tbody>
</table>

**5 Conclusion**

We have proposed a novel decomposition of the precoder matrix of \( N \)-continuous 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.