A Reduced-Complexity Partial-Interference-Cancellation Group Decoder for STBCs

Rakshith Rajashekar, *Student Member, IEEE*, K. V. S. Hari, *Senior Member, IEEE*, and L. Hanzo, *Fellow, IEEE*

***Abstract*—In this paper, we propose a reduced-complexity implementation of partial interference cancellation group decoder with successive interference cancellation (PIC-GD-SIC) by employing the theory of displacement structures. The proposed algorithm exploits the block-Toeplitz structure of the effective matrix and chooses an ordering of the groups such that the zero-forcing matrices associated with the various groups are obtained through Schur recursions without any approximations. We show using an example that the proposed implementation offers a significantly reduced computational complexity compared to the direct approach without any loss in performance.**

***Index Terms*—Displacement structure, diversity and decoding complexity, partial interference cancellation group decoding, STBC.**

I. Introduction

A generalized class of linear space-time block codes (STBC) [1] was proposed in [2]–[5] which offers full transmit- and receive-diversity with the aid of the so-called partial interference cancellation group decoding (PIC-GD). The PIC-GD offers a significantly reduced computational complexity compared to that of the maximum-likelihood (ML) detector and hence has received significant attention in the recent past. The rate achieved by these STBCs is $R = Q/T = \gamma(T - N_t + 1)/T$ complex symbols per channel use (cspcu), where Q is the number of complex symbols encoded in the STBC, which spans T channel uses, while N_t is the number of transmit antennas and $1 \leq \gamma \leq N_t$. An extended version of the PIC-GD known as the partial interference cancellation group decoding with successive interference cancellation (PIC-GD-SIC) was also proposed in [2]. It was observed in [2]–[5] that for any given antenna configuration and spectral efficiency, the PIC-GD-SIC gives a better bit error ratio (BER) performance than that of the PIC-GD. Recently, a low-complexity algorithm was proposed for the implementation of the PIC-GD, which imposes on the order of $\mathcal{O}(N_t^3T^3)$ operations (refer to Table II, [6]). However, neither a low-complexity detector nor any beneficial ordering of the groups has been proposed for the PIC-GD-SIC in the existing literature. Hence we solve this open problem by proposing an ordering of the groups and a recursive algorithm for computing the zero-forcing matrices for PIC-GD-SIC that significantly reduces the computational complexity imposed.

Consider a linear STBC design given by $\mathcal{C}' = \sum_{i=1}^{Q} s_i\mathbf{M}_i$, where $\mathbf{M}_i \in \mathbb{C}^{N_t\times T}$ are the linearly independent weight matrices and s_i are the complex valued symbols from a signal set $\mathcal{A} \subset \mathbb{C}$. Considering a frequency-flat block Rayleigh-fading scenario, we have

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{N}, \tag{1}$$

where $\mathbf{X} \in \mathcal{C}'$, $\mathbf{Y} \in \mathbb{C}^{N_r\times T}$ is the received matrix, $\mathbf{H} \in \mathbb{C}^{N_r\times N_t}$ is the channel matrix and $\mathbf{N} \in \mathbb{C}^{N_r\times T}$ is the noise matrix. The entries of the channel and the noise matrices are from zero-mean complex Gaussian distributions $\mathcal{CN}(0,1)$ and $\mathcal{CN}(0,\sigma^2)$, respectively, where σ^2 represents the noise variance per complex dimension. Assuming perfect channel state information at the receiver, the ML detection yields $(\hat{s}_1, \hat{s}_2, \cdots, \hat{s}_Q)_{ML} = \arg\min \|\mathbf{Y} - \mathbf{H}\sum_{i=1}^{Q} s_i\mathbf{M}_i\|^2$, where the minimization is over $\{s_i\}_{i=1}^{Q} \in \mathcal{A}^Q$.

Upon vectorizing (1), we arrive at $\bar{\mathbf{y}} = \mathbf{G}\mathbf{s} + \bar{\mathbf{n}}$, where $\bar{\mathbf{y}} = \text{vec}(\mathbf{Y}) \in \mathbb{C}^{N_rT}$, $\bar{\mathbf{n}} = \text{vec}(\mathbf{N}) \in \mathbb{C}^{N_rT}$, $\mathbf{G} = [\mathbf{I}_T \otimes \mathbf{H}]\Phi \in \mathbb{C}^{N_rT\times Q}$ such that $\Phi = [\text{vec}(\mathbf{M}_1), \text{vec}(\mathbf{M}_2), \ldots, \text{vec}(\mathbf{M}_Q)] \in \mathbb{C}^{N_tT\times Q}$ and $\mathbf{s} = [s_1, s_2, \ldots, s_Q]^T \in \mathcal{A}^Q$. Note that the ML detection complexity order may remain excessive, even when using the sphere decoder [7].

A. PIC-GD-SIC Algorithm

The set of Q symbols $\{s_i\}_{i=1}^{Q}$ is divided into K groups, where the k^{th} group is represented by $\mathbf{s}_{\mathcal{I}_k}$, each group contains the symbols indexed by the set $\mathcal{I}_k$, such that each $\mathcal{I}_k \subset \{1, 2, \ldots, Q\}$, $\bigcup_{k=1}^{K} \mathcal{I}_k = \{1, 2, \ldots, Q\}$, and $\mathcal{I}_k \cap \mathcal{I}_l = \phi$ for $1 \leq k \neq l \leq K$. The K symbol groups are ordered as $\{\mathcal{I}_{j_1}, \mathcal{I}_{j_2}, \ldots, \mathcal{I}_{j_K}\}$, where j_k for $1 \leq k \leq K$ are the distinct elements from the set $\{1, 2, \cdots, K\}$.

Let $\mathbf{G}_{\mathcal{I}_k}$ represent a matrix hosting only those specific columns of $\mathbf{G}$ that are indexed by the set $\mathcal{I}_k$ and $\mathbf{G}^c_{\mathcal{I}_k} = \{\mathbf{G}_{\mathcal{I}_1}, \mathbf{G}_{\mathcal{I}_2}, \cdots, \mathbf{G}_{\mathcal{I}_{k-1}}, \mathbf{G}_{\mathcal{I}_{k+1}}, \ldots, \mathbf{G}_{\mathcal{I}_K}\}$. Furthermore, let $\mathbf{G}^c_{\mathcal{I}_{j_k}} = [\mathbf{G}_{\mathcal{I}_{j_{k+1}}}, \mathbf{G}_{\mathcal{I}_{j_{k+2}}}, \ldots, \mathbf{G}_{\mathcal{I}_{j_K}}]$.

For $1 \leq k < K$, obtain $\hat{\mathbf{s}}_{\mathcal{I}_{j_k}} = \arg\min_{\mathbf{x}\in\mathcal{A}^{|\mathcal{I}_{j_k}|}} \|\bar{\mathbf{z}}^{(k)} - \mathbf{P}_{\mathcal{I}_{j_k}}\mathbf{G}_{\mathcal{I}_{j_k}}\mathbf{x}\|^2$, where $\bar{\mathbf{z}}^{(k)}_{\mathcal{I}_{j_k}} = \mathbf{P}_{\mathcal{I}_{j_k}}\bar{\mathbf{y}}^{(k)}$ so that $\bar{\mathbf{y}}^{(k)} = \bar{\mathbf{y}}^{(k-1)} - \mathbf{G}_{\mathcal{I}_{j_{k-1}}}\hat{\mathbf{s}}_{\mathcal{I}_{j_{k-1}}}$, $\mathbf{P}_{\mathcal{I}_{j_k}} = \mathbf{I} - \mathbf{Q}_{\mathcal{I}_{j_k}}$ so that $\mathbf{Q}_{\mathcal{I}_{j_k}} = \mathbf{G}^c_{\mathcal{I}_{j_k}}(\mathbf{G}^{c^H}_{\mathcal{I}_{j_k}}\mathbf{G}^c_{\mathcal{I}_{j_k}})^{-1}\mathbf{G}^{c^H}_{\mathcal{I}_{j_k}}$. For $k = 1$, $\bar{\mathbf{y}}^{(k)}$ is taken to be $\bar{\mathbf{y}}$.

Manuscript received March 28, 2013; revised June 19, 2013; accepted July 12, 2013. The financial support of the DST, India and of the EPSRC, UK under the auspices of the India-UK Advanced Technology Center (IU-ATC) is gratefully acknowledged.

R. Rajashekar and K. V. S. Hari are with the Department of Electrical Communication Engineering, Indian Institute of Science, India (e-mail: {rakshithmr, hari}@ece.iisc.ernet.in).

L. Hanzo is with the School of ECS, University of Southampton, UK (e-mail: lh@ecs.soton.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2013.2273532

For example, when $|\mathcal{I}_k| = \lambda = Q/K$ for $1 \leq k \leq K$, it is straightforward to show that the PIC-GD/PIC-GD-SIC complexity order is only $|\mathcal{A}|^\lambda$, while the ML decoding complexity order is $|\mathcal{A}|^Q$.

B. PIC-GD-SIC Complexity Analysis

In this paper, we only focus our attention on the computational complexity involved in obtaining the set $\{\mathbf{Q}_{\mathcal{I}_{j_k}}\}_{k=1}^{K-1}$. For the ease of presentation, we assume that $|\mathcal{I}_{j_k}| = \lambda$ for all $1 \leq k \leq K$. It may then be readily seen that $\mathbf{G}^c_{\mathcal{I}_{j_k}}$ is of size $(N_r T \times Q_k)$, where $Q_k = Q - k\lambda$. Hence, it is straightforward to show that

- $(\mathbf{G}^{c^H}_{\mathcal{I}_{j_k}} \mathbf{G}^c_{\mathcal{I}_{j_k}})$ imposes $\mathcal{O}(N_r^2 T^2 Q_k)$ operations,[1]
- $(\mathbf{G}^{c^H}_{\mathcal{I}_{j_k}} \mathbf{G}^c_{\mathcal{I}_{j_k}})^{-1}$ requires $\mathcal{O}(Q_k^3)$ operations, and
- $\mathbf{G}^c_{\mathcal{I}_{j_k}} (\mathbf{G}^{c^H}_{\mathcal{I}_{j_k}} \mathbf{G}^c_{\mathcal{I}_{j_k}})^{-1} \mathbf{G}^{c^H}_{\mathcal{I}_{j_k}}$ takes $\mathcal{O}(Q_k^2 N_r T + Q_k N_r^2 T^2)$ operations.

For a given k, the direct computation of $\mathbf{G}^c_{\mathcal{I}_{j_k}} (\mathbf{G}^{c^H}_{\mathcal{I}_{j_k}} \mathbf{G}^c_{\mathcal{I}_{j_k}})^{-1} \mathbf{G}^{c^H}_{\mathcal{I}_{j_k}}$ imposes $\mathcal{O}(N_r^2 T^2 Q_k + Q_k^3 + Q_k^2 N_r T + Q_k N_r^2 T^2)$ operations.

Thus, the computation of $\{\mathbf{Q}_{\mathcal{I}_{j_k}}\}_{k=1}^{K-1}$ has a complexity order of

$$\sum_{k=1}^{K-1} \left(N_r^2 T^2 Q_k + Q_k^3 + Q_k^2 N_r T + Q_k N_r^2 T^2\right) \tag{2}$$

operations.

Recall that the rate achieved by the codes of [2]–[5] is $R = Q/T = \gamma(T - N_t + 1)/T$ cspcu, which reaches γ asymptotically with T. Hence a large T is desirable for achieving a high bandwidth efficiency. Since $Q = \gamma(T - N_t + 1)$, a large T leads to a large Q and in turn a large Q_k. Thus, operating at a high bandwidth efficiency requires a large T and Q_k, which leads to a high computational complexity. Assuming $\gamma = N_t$ and N_r to be fixed, the highest order term in (2) is $N_t^3 T^3$.

II. Proposed Reduced-Complexity Algorithm (R-PIC-GD-SIC)

By exploiting the block-Toeplitz nature of $\mathbf{G}$ and employing the theory of *displacement structures* [8]–[10], we propose a beneficial ordering of the groups that significantly reduces the complexity involved in computing the set $\{\mathbf{Q}_{\mathcal{I}_{j_k}}\}_{k=1}^{K-1}$.

A. Review of the Theory of Displacement Structures [8]–[10]

Consider $\mathbf{C} \in \mathbb{C}^{m \times n}$ and strictly lower triangular matrices $\mathbf{F}^f \in \mathbb{C}^{m \times m}$ and $\mathbf{F}^b \in \mathbb{C}^{n \times n}$. The *displacement* of $\mathbf{C}$ with respect to the *displacement operators* $\mathbf{F}^f$ and $\mathbf{F}^b$ is given by $\nabla_{(\mathbf{F}^f,\mathbf{F}^b)}\mathbf{C} = \mathbf{C} - \mathbf{F}^f \mathbf{C} (\mathbf{F}^b)^H$. The rank of $\nabla_{(\mathbf{F}^f,\mathbf{F}^b)}\mathbf{C}$ is referred to as the *displacement rank* of $\mathbf{C}$ with respect to $(\mathbf{F}^f, \mathbf{F}^b)$, and is denoted by α. The matrix pair $\mathbf{J} = [\mathbf{j}_1, \mathbf{j}_2, \ldots, \mathbf{j}_\alpha]$ and $\mathbf{K} = [\mathbf{k}_1, \mathbf{k}_2, \cdots, \mathbf{k}_\alpha]$ is said to be a *generator* of $\mathbf{C}$ if $\nabla_{(\mathbf{F}^f,\mathbf{F}^b)}\mathbf{C} = \mathbf{J}\mathbf{K}^H$, and α represents the *length* of the generator, while $(\mathbf{J}, \mathbf{K})$ having the minimal possible length is termed as the *minimal generator*. The generator $(\mathbf{J}, \mathbf{K})$ is said to be *proper* (with respect to some pivoting column i) if all the elements of the j^{th} row and above in both $\mathbf{J}$ and $\mathbf{K}$ are zero, except for the elements $\mathbf{J}_{j,i}$ and $\mathbf{K}_{j,i}$. If the generator is not proper, then it may be made proper with the aid of $\mathcal{O}(\alpha q)$ operations, where $q = \max(m, n)$ (see Section III, [9]). The *displacement representation* of $\mathbf{C}$ is given by $\mathbf{C} = \sum_{i=1}^{\alpha} \mathbf{L}_n(\mathbf{j}_i, \mathbf{F}^f) \mathbf{L}_n^H(\mathbf{k}_i, \mathbf{F}^b)$, where $\mathbf{L}_n(\mathbf{j}_i, \mathbf{F}^f) \in \mathbb{C}^{m \times n}$ and $\mathbf{L}_n(\mathbf{k}_i, \mathbf{F}^b) \in \mathbb{C}^{n \times n}$ are the lower triangular matrices given by $\mathbf{L}_n(\mathbf{j}_i, \mathbf{F}^f) = [\mathbf{j}_i, \mathbf{F}^f \mathbf{j}_i, \ldots, (\mathbf{F}^f)^{n-1} \mathbf{j}_i]$, and $\mathbf{L}_n(\mathbf{k}_i, \mathbf{F}^b) = [\mathbf{k}_i, \mathbf{F}^b \mathbf{k}_i, \ldots, (\mathbf{F}^b)^{n-1} \mathbf{k}_i]$. It is clear that the complete information about $\mathbf{C}$ is present in its generator. Note that for any non-singular matrix $\mathbf{S}$, we have $\nabla_{(\mathbf{F}^f,\mathbf{F}^b)}\mathbf{C} = \mathbf{J}\mathbf{K}^H = \mathbf{J}\mathbf{S}\mathbf{S}^{-1}\mathbf{K}^H$. Thus generators are not unique.

Generalized Schur Algorithm (GSA): Given $\mathbf{A}^{(1)} \in \mathbb{C}^{m \times n}$ relying on the proper generator pair $\mathbf{J}_p^{(1)} = [\mathbf{j}_1^{(1)}, \ldots, \mathbf{j}_{pvt_1}^{(1)}, \ldots, \mathbf{j}_\alpha^{(1)}]$, $\mathbf{K}_p^{(1)} = [\mathbf{k}_1^{(1)}, \ldots, \mathbf{k}_{pvt_1}^{(1)}, \ldots, \mathbf{k}_\alpha^{(1)}]$, we obtain $\mathbf{A}^{(2)} = \mathbf{A}^{(1)} - \mathbf{j}_{pvt_1}^{(1)} \mathbf{k}_{pvt_1}^{(1)^H}$, which gives the Schur complement of element (1,1) of $\mathbf{A}^{(1)}$. The generator of $\mathbf{A}^{(2)}$ can be obtained from that of $\mathbf{A}^{(1)}$ as $\mathbf{J}^{(2)} = [\mathbf{j}_1^{(1)}, \ldots, \mathbf{F}^f \mathbf{j}_{pvt_1}^{(1)}, \ldots, \mathbf{j}_\alpha^{(1)}]$, $\mathbf{K}^{(2)} = [\mathbf{k}_1^{(1)}, \ldots, \mathbf{F}^b \mathbf{k}_{pvt_1}^{(1)}, \ldots, \mathbf{k}_\alpha^{(1)}]$, which may not be proper. Converting this to a proper generator yields $\mathbf{J}_p^{(2)}, \mathbf{K}_p^{(2)}$. By repeating this process r times, we will arrive at the Schur complement of the (r, r) block of $\mathbf{A}^{(1)}$, which is given by $\mathbf{A}^{(r+1)} = \mathbf{A}^{(r)} - \mathbf{j}_{pvt_r}^{(r)} \mathbf{k}_{pvt_r}^{(r)^H}$, with the generator pair $[\mathbf{J}_p^{(r+1)}, \mathbf{K}_p^{(r+1)}]$. Note that $\mathbf{A}^{(1)} = \sum_{i=1}^{r} \mathbf{j}_{pvt_i}^{(i)} \mathbf{k}_{pvt_i}^{(i)^H} + \mathbf{A}^{(r+1)}$ has the form

$$\underbrace{[\mathbf{j}_{pvt_1}^{(1)}, \ldots, \mathbf{k}_{pvt_r}^{(r)}]}_{lower\ triangular} \underbrace{[\mathbf{k}_{pvt_1}^{(1)}, \ldots, \mathbf{k}_{pvt_r}^{(r)}]^H}_{upper\ triangular} + \begin{bmatrix} \mathbf{O}_{r \times r} & \mathbf{O} \\ \mathbf{O} & \mathbf{A}_{m-r \times n-r}^{(r+1)} \end{bmatrix}.$$

Thus, the recursive operations applied to the generator of a given matrix yields the generator of the Schur complement of its (r, r) block.

Thus, instead of operating directly on the matrix, one can operate on its generator to obtain the Schur complements, which takes significantly lesser number of operations compared to that of the direct approach. Following lemma quantifies the complexity order of the GSA.

Lemma 1 (Chun and Kailath): If $\mathbf{A} = \begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \end{bmatrix} \in \mathbb{C}^{m \times n}$, has a displacement rank α with respect to the displacement operators $\mathbf{F}^f$ and $\mathbf{F}^b$, then the generator of the Schur complement of $\mathbf{A}_{1,1} \in \mathbb{C}^{r \times r}$, i.e. $\mathbf{A}_{2,2} - \mathbf{A}_{2,1} \mathbf{A}_{1,1}^{-1} \mathbf{A}_{1,2}$, can be obtained by r recursions of the GSA that imposes $\mathcal{O}(\alpha^3 r \log^2 r)$ operations based on the divide-and-conquer approach (refer to Section 2 of [8]).

Proof: Proof can be found in Section 4 of [8]. ■

It is clear from Lemma 1 that the complexity order of the GSA is determined by the displacement rank of the matrix and the number of Schur recursions, rather than by the actual size of the matrix.

[1] By operations, we mean complex-valued multiplications/additions.

B. The R-PIC-GD-SIC Algorithm Proposed for STBCs Relying on a Block-Toeplitz $\mathbf{G}$

Consider a block-Toeplitz matrix $\mathbf{G} \in \mathbb{C}^{m\times n}$ of the form

$$\begin{bmatrix} \mathbf{G}_1 & \mathbf{O} & \dots & \mathbf{O} & \mathbf{O} \\ \mathbf{G}_2 & \mathbf{G}_1 & \dots & \mathbf{O} & \mathbf{O} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{G}_\ell & \mathbf{G}_{\ell-1} & \ddots & \mathbf{G}_1 & \mathbf{O} \\ \mathbf{O} & \mathbf{G}_\ell & \ddots & \mathbf{G}_2 & \mathbf{G}_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{O} & \mathbf{O} & \dots & \mathbf{G}_\ell & \mathbf{G}_{\ell-1} \\ \mathbf{O} & \mathbf{O} & \dots & \mathbf{O} & \mathbf{G}_\ell \end{bmatrix}, \quad (3)$$

where we have $\mathbf{G}_i \in \mathbb{C}^{r\times s}$ for $1 \le i \le \ell$ and each $\mathbf{O}$ is a $(r \times s)$-element null-matrix. Note that $\mathbf{G}$ can be viewed as a $(p \times q)$-element matrix, where each element is a matrix of size $(r \times s)$, so that $m = rp = N_rT$ and $n = sq = Q$. The structure of the effective matrix $\mathbf{G}$ in all the codes of [2], [3] is the same as that of (3). For example, please refer to (41) in [2] and (22), (24), (29) in [4]. Let $\mathcal{I}_j = \{1 + s(j-1), 2 + s(j-1), \dots, s + s(j-1)\}$ for $1 \le j \le K$, and consider the ordering of the groups given by $\{\mathcal{I}_K, \mathcal{I}_{K-1}, \dots, \mathcal{I}_2, \mathcal{I}_1\}$. With the aid of this ordering of the groups, we have $\mathbf{G}^c_{\mathcal{I}_j} = \{\mathbf{G}_{\mathcal{I}_1}, \mathbf{G}_{\mathcal{I}_2}, \dots, \mathbf{G}_{\mathcal{I}_{j-1}}\}$ for $j \in \{K, K-1, \dots, 2\}$. Note that we have to compute $\mathbf{P}_{\mathcal{I}_j} = \mathbf{I} - \mathbf{Q}_{\mathcal{I}_j}$, and hence we have $-\mathbf{Q}_{\mathcal{I}_j} = -\mathbf{G}^c_{\mathcal{I}_j}\left(\mathbf{G}^{c^H}_{\mathcal{I}_j}\mathbf{G}^c_{\mathcal{I}_j}\right)^{-1}\mathbf{G}^{c^H}_{\mathcal{I}_j}$, which is exactly the Schur complement of $\mathbf{G}^{c^H}_{\mathcal{I}_j}\mathbf{G}^c_{\mathcal{I}_j}$ in $\begin{bmatrix} \mathbf{G}^{c^H}_{\mathcal{I}_j}\mathbf{G}^c_{\mathcal{I}_j} & \mathbf{G}^{c^H}_{\mathcal{I}_j} \\ \mathbf{G}^c_{\mathcal{I}_j} & \mathbf{O} \end{bmatrix}$. In what follows, we show that while obtaining the generator of $-\mathbf{Q}_{\mathcal{I}_K}$ using the GSA, the generators of $\{-\mathbf{Q}_{\mathcal{I}_j}\}_{j=2}^{K-1}$ can also be obtained without any additional computational burden.

Let $\mathbf{W}_{u,v}$ be a $(u \times u)$-element all-zero matrix, except for $\mathbf{I}_v$ on the sub-diagonal blocks, where u is an integer multiple of v. Consider $\mathbf{R} = \mathbf{G}^{c^H}_{\mathcal{I}_K}\mathbf{G}^c_{\mathcal{I}_K} \in \mathbb{C}^{(n-s)\times(n-s)}$, which is of the form

$$\begin{bmatrix} \mathbf{R}_1 & \mathbf{R}_2^H & \dots & \mathbf{R}_{q-1}^H \\ \mathbf{R}_2 & \mathbf{R}_1 & \dots & \mathbf{R}_{q-2}^H \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{R}_{q-1} & \mathbf{R}_{q-2} & \dots & \mathbf{R}_1 \end{bmatrix}, \quad (4)$$

where $\mathbf{R}_j = \sum_{i=1}^{\ell} \mathbf{G}_i^H \mathbf{G}_{i+j-1}$ for $1 \le j \le q-1$, and $\mathbf{G}_k$ is taken as $\mathbf{O}$ for $k > \ell$. Let $\bar{\mathbf{Q}}^{(1)}_{\mathcal{I}_K} = \begin{bmatrix} \mathbf{R} & \mathbf{G}^{c^H}_{\mathcal{I}_K} \\ \mathbf{G}^c_{\mathcal{I}_K} & \mathbf{O} \end{bmatrix} \in \mathbb{C}^{(m+n-s)\times(m+n-s)}$ and $\mathbf{W} = \begin{bmatrix} \mathbf{W}_{n-s,s} & \mathbf{O} \\ \mathbf{O} & \mathbf{W}_{m,r} \end{bmatrix} \in \mathbb{R}^{(m+n-s)\times(m+n-s)}$. Then, it is easy to verify that $\bar{\mathbf{Q}}^{(1)}_{\mathcal{I}_K}$ has a displacement rank of $2s$ associated with $\mathbf{F}^f = \mathbf{F}^b = \mathbf{W}$ and has a proper generator pair[2] of $\mathbf{J}_p^{(1)} = \begin{bmatrix} \mathbf{I}_s & \mathbf{O} \\ \mathbf{R}_2\mathbf{R}_1^{-1} & \mathbf{R}_2\mathbf{R}_1^{-1} \\ \vdots & \vdots \\ \mathbf{R}_{q-1}\mathbf{R}_1^{-1} & \mathbf{R}_{q-1}\mathbf{R}_1^{-1} \\ \mathbf{G}_{\mathcal{I}_1}\mathbf{R}_1^{-1} & \mathbf{G}_{\mathcal{I}_1}\mathbf{R}_1^{-1} \end{bmatrix}$, $\mathbf{K}_p^{(1)} = \begin{bmatrix} \mathbf{R}_1 & \mathbf{O} \\ \mathbf{R}_2 & -\mathbf{R}_2 \\ \vdots & \vdots \\ \mathbf{R}_{q-1} & -\mathbf{R}_{q-1} \\ \mathbf{G}_{\mathcal{I}_1} & -\mathbf{G}_{\mathcal{I}_1} \end{bmatrix}$, with pivoting block-columns of $\mathbf{J}^{(1)}_{pvt_1} = \begin{bmatrix} \mathbf{I}_s \\ \mathbf{R}_2\mathbf{R}_1^{-1} \\ \vdots \\ \mathbf{R}_{q-1}\mathbf{R}_1^{-1} \\ \mathbf{G}_{\mathcal{I}_1}\mathbf{R}_1^{-1} \end{bmatrix}$ and $\mathbf{K}^{(1)}_{pvt_1} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \\ \vdots \\ \mathbf{R}_{q-1} \\ \mathbf{G}_{\mathcal{I}_1} \end{bmatrix}$.

Lemma 2: Let $\bar{\mathbf{Q}}^{(1)}_{\mathcal{I}_K}$, $\mathbf{W}$, $\mathbf{J}^{(1)}_{pvt_1}$ and $\mathbf{K}^{(1)}_{pvt_1}$ be defined as above. If $\bar{\mathbf{Q}}^{(2)}_{\mathcal{I}_K} = \bar{\mathbf{Q}}^{(1)}_{\mathcal{I}_K} - \mathbf{J}^{(1)}_{pvt_1}\mathbf{K}^{(1)^H}_{pvt_1}$, then the $(m \times m)$-element right-lower block matrix of $\bar{\mathbf{Q}}^{(2)}_{\mathcal{I}_K}$ is equal to $-\mathbf{Q}_{\mathcal{I}_2} = -\mathbf{G}^c_{\mathcal{I}_2}\left(\mathbf{G}^{c^H}_{\mathcal{I}_2}\mathbf{G}^c_{\mathcal{I}_2}\right)^{-1}\mathbf{G}^{c^H}_{\mathcal{I}_2}$.

Proof: From the GSA, we have $\bar{\mathbf{Q}}^{(2)}_{\mathcal{I}_K}$ equal to the Schur complement of the $(s \times s)$-element left-upper block matrix of $\bar{\mathbf{Q}}^{(1)}_{\mathcal{I}_K}$ (i.e. $\mathbf{R}_1$). Thus, we have

$$\bar{\mathbf{Q}}^{(2)}_{\mathcal{I}_K} = \begin{bmatrix} \mathbf{O} & \mathbf{O} & \dots & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{R}_1 & \dots & \mathbf{R}_{q-2}^H & \mathbf{G}^H_{\mathcal{I}_2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{O} & \mathbf{R}_{q-2} & \dots & \mathbf{R}_1 & \mathbf{G}^H_{\mathcal{I}_{K-1}} \\ \mathbf{O} & \mathbf{G}_{\mathcal{I}_2} & \dots & \mathbf{G}_{\mathcal{I}_{K-1}} & \mathbf{O} \end{bmatrix} - \begin{bmatrix} \mathbf{O} \\ \mathbf{R}_2 \\ \vdots \\ \mathbf{R}_{q-1} \\ \mathbf{G}_{\mathcal{I}_1} \end{bmatrix} \mathbf{R}_1^{-1}\left[\mathbf{O}, \mathbf{R}_2^H, \dots, \mathbf{R}_{q-1}^H, \mathbf{G}^H_{\mathcal{I}_1}\right], \quad (5)$$

which reduces to (6) (See equation at bottom of page) . It may be seen from (6) that the $(m\times m)$-element right-lower block matrix of $\bar{\mathbf{Q}}^{(2)}_{\mathcal{I}_K}$ is equal to $-\mathbf{G}^c_{\mathcal{I}_2}\left(\mathbf{G}^{c^H}_{\mathcal{I}_2}\mathbf{G}^c_{\mathcal{I}_2}\right)^{-1}\mathbf{G}^{c^H}_{\mathcal{I}_2}$. Note that we may also arrive at (6) by directly evaluating $\bar{\mathbf{Q}}^{(1)}_{\mathcal{I}_K} - \mathbf{J}^{(1)}_{pvt_1}\mathbf{K}^{(1)^H}_{pvt_1}$.

Proposition 1: Let $\bar{\mathbf{Q}}^{(1)}_{\mathcal{I}_K}$, $\mathbf{W}$, $\mathbf{J}^{(1)}_{pvt_1}$ and $\mathbf{K}^{(1)}_{pvt_1}$ be defined as before. If we have $\bar{\mathbf{Q}}^{(t+1)}_{\mathcal{I}_K} = \bar{\mathbf{Q}}^{(t)}_{\mathcal{I}_K} - \mathbf{J}^{(t)}_{pvt_t}\mathbf{K}^{(t)^H}_{pvt_t}$, then the $(m \times m)$-element right-lower block matrix of $\bar{\mathbf{Q}}^{(t+1)}_{\mathcal{I}_K}$ is equal to $-\mathbf{Q}_{\mathcal{I}_{t+1}} = -\mathbf{G}^c_{\mathcal{I}_{t+1}}\left(\mathbf{G}^{c^H}_{\mathcal{I}_{t+1}}\mathbf{G}^c_{\mathcal{I}_{t+1}}\right)^{-1}\mathbf{G}^{c^H}_{\mathcal{I}_{t+1}}$.

Proof: The proof directly follows from the GSA and by recursively following the steps of *Lemma 2*. ■

Algorithm 1 summarizes the steps involved in obtaining $\{-\mathbf{Q}_{\mathcal{I}_j}\}_{j=2}^{K-1}$.

[2] A generic procedure for finding the initial pair of generators can be found in the Appendix of [9].

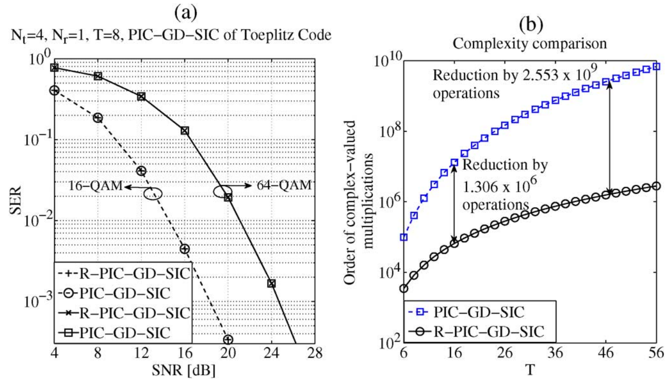

Fig. 1. Plot (a) compares the SER performance of the R-PIC-GD-SIC and the PIC-GD-SIC of Toeplitz code [11] in a system having $N_t = 4$, $N_r = 1$ and $T = 8$. Plot (b) compares the order of computational complexity of the two approaches in the aforementioned scenario.

Algorithm 1

Require: $t = 1$, the proper generator $(\mathbf{J}_p^{(t)}, \mathbf{K}_p^{(t)})$ of $\bar{\mathbf{Q}}_{\mathcal{I}_K}^{(t)}$ with pivoting column-blocks $\mathbf{J}_{pvt_1}^{(t)}$ and $\mathbf{K}_{pvt_1}^{(t)}$.

while $t \leq K - 1$ **do**

1. Compute $\bar{\mathbf{Q}}_{\mathcal{I}_K}^{(t+1)} = \bar{\mathbf{Q}}_{\mathcal{I}_K}^{(t)} - \mathbf{J}_{pvt_t}^{(t)} \mathbf{K}_{pvt_t}^{(t)^H}$.
2. The $(m \times m)$-element right-lower block of $\bar{\mathbf{Q}}_{\mathcal{I}_K}^{(t+1)}$ gives $-\mathbf{Q}_{\mathcal{I}_{t+1}}$.
3. Obtain the proper generator $(\mathbf{J}_p^{(t+1)}, \mathbf{K}_p^{(t+1)})$ and the pivoting column-blocks $\mathbf{J}_{pvt_{t+1}}^{(t+1)}$ and $\mathbf{K}_{pvt_{t+1}}^{(t+1)}$.
4. $t \to t + 1$.

end while

C. Performance and Complexity Analysis

Fig. 1(a) gives the symbol error rate (SER) performance achieved by the Toeplitz code ($N_t = 4$, $N_r = 1$, $T = 8$) [11] when employing PIC-GD-SIC and R-PIC-GD-SIC. It is clear that R-PIC-GD-SIC does not incur any performance loss compared to PIC-GD-SIC, since the set of projection matrices $\{\mathbf{Q}_{\mathcal{I}_j}\}_{j=2}^{K-1}$ obtained by R-PIC-GD-SIC are exactly same as that of PIC-GD-SIC. The difference in the numerical values of the entries of the projection matrices are of order 10^{-15}.

Obtaining the generator pair of $\mathbf{J}_p^{(1)}$ and $\mathbf{K}_p^{(1)}$ essentially involves the computation of

- $\{\mathbf{R}_i\}_{i=1}^{q-1}$ that imposes $\mathcal{O}(\ell^2 q^2 p)$ operations,
- $\mathbf{R}_1^{-1}$ that requires $\mathcal{O}(s^3)$ operations,
- $\{\mathbf{R}_i \mathbf{R}_1^{-1}\}_{i=1}^{q-1}$ that takes $\mathcal{O}(qs^3)$ operations and
- $\mathbf{G}_{\mathcal{I}_1} \mathbf{R}_1^{-1}$ that is associated with $\mathcal{O}(rs^3)$ operations.

It may be readily seen that Algorithm 1 imposes $K' = K - 1$ Schur recursions and hence requires $\mathcal{O}(s^3 K' \log^2 K')$ operations using the divide-and-conquer approach (refer to *Lemma 1*). A careful comparison with (2) reveals that the proposed algorithm gives a significant reduction in the complexity with respect to the direct computation. Considering the Toeplitz code [11] for $N_t = 4$, $N_r = 1$, we have $r = 1$, $s = 1$, $\ell = 4$, $p = m = T$, $q = K$ and $Q = n = K$. Fig. 1(b) provides a comparison of the computational complexity incurred by the R-PIC-GD-SIC to that of the PIC-GD-SIC. It is clear from Fig. 1(b) that R-PIC-GD-SIC offers a significant reduction in the complexity. Specifically, at $T = 16$ PIC-GD-SIC imposes on the order of 1313×10^4 operations, while as R-PIC-GD-SIC requires on the order of 6.578×10^4 operations. By contrast, at $T = 46$ PIC-GD-SIC has on the order of 2555×10^6 operations, while R-PIC-GD-SIC imposes on the order of 1.559×10^6 operations.

III. Conclusions

We proposed an ordering of the groups for the PIC-GD-SIC of STBCs relying on a block-Toeplitz effective matrix and showed that the associated zero-forcing matrices can be computed with the aid of Schur recursions by applying the theory of displacement structures without any approximations. Furthermore, we have demonstrated with the aid of an example that the proposed algorithm offers a significantly reduced complexity compared to the direct approach.

References

[1] V. Tarokh, N. Seshadri, and A. Calderbank, "Space-time codes for high data rate wireless communication: Performance criterion and code construction," *IEEE Trans. Inf. Theory*, vol. 44, no. 2, pp. 744–765, Mar. 1998.

[2] X. Guo and X.-G. Xia, "On full diversity space-time block codes with partial interference cancellation group decoding," *IEEE Trans. Inf. Theory*, vol. 55, no. 10, pp. 4366–4385, Oct. 2009.

[3] L. P. Natarajan and B. Sundar Rajan, "Collocated and distributed STBCs with partial interference cancellation decoding, part II: Code constructions," *IEEE Trans. Wireless Commun.*, vol. 10, no. 9, pp. 3042–3052, Sep. 2011.

[4] W. Zhang, T. Xu, and X.-G. Xia, "Two designs of space-time block codes achieving full diversity with partial interference cancellation group decoding," *IEEE Trans. Inf. Theory*, vol. 58, no. 2, pp. 747–764, Feb. 2012.

$$\left[\begin{array}{cccc|c}
\mathbf{O} & \mathbf{O} & \cdots & \mathbf{O} & \mathbf{O} \\
\mathbf{O} & \mathbf{R}_1 - \mathbf{R}_2\mathbf{R}_1^{-1}\mathbf{R}_2^H & \cdots & \mathbf{R}_{q-2}^H - \mathbf{R}_2\mathbf{R}_1^{-1}\mathbf{R}_{q-1}^H & \mathbf{G}_{\mathcal{I}_2}^H - \mathbf{R}_2\mathbf{R}_1^{-1}\mathbf{G}_{\mathcal{I}_1}^H \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
\mathbf{O} & \mathbf{R}_{q-2} - \mathbf{R}_{q-1}\mathbf{R}_1^{-1}\mathbf{R}_2^H & \cdots & \mathbf{R}_1 - \mathbf{R}_{q-1}\mathbf{R}_1^{-1}\mathbf{R}_{q-1}^H & \mathbf{G}_{\mathcal{I}_{K-1}}^H - \mathbf{R}_{q-1}\mathbf{R}_1^{-1}\mathbf{G}_{\mathcal{I}_1}^H \\
\hline
\mathbf{O} & \mathbf{G}_{\mathcal{I}_2} - \mathbf{G}_{\mathcal{I}_1}\mathbf{R}_1^{-1}\mathbf{R}_2^H & \cdots & \mathbf{G}_{\mathcal{I}_{K-1}} - \mathbf{G}_{\mathcal{I}_1}\mathbf{R}_1^{-1}\mathbf{R}_{q-1}^H & \underbrace{-\mathbf{G}_{\mathcal{I}_1}\mathbf{R}_1^{-1}\mathbf{G}_{\mathcal{I}_1}^H}_{m \times m}
\end{array}\right] \quad (6)$$

[5] L. Shi, W. Zhang, and X.-G. Xia, "High-rate and full-diversity space-time block codes with low complexity partial interference cancellation group decoding.," *IEEE Trans. Commun.*, vol. 59, no. 5, pp. 1201–1207, May 2011.

[6] X. Guo and K. Wu, "An efficient algorithm for partial interference cancellation group decoding," in *Proc. IEEE Wireless Communication & Networking Conference (WCNC)*, Mexico, Mar. 2011, pp. 1977–1982.

[7] J. Jalden and B. Ottersten, "On the complexity of sphere decoding in digital communications," *IEEE Trans. Signal Process.*, vol. 53, no. 4, pp. 1474–1484, Apr. 2005.

[8] J. Chun and T. Kailath, "Divide and conquer solutions of least squares problems for matrices with displacement structure," *SIAM J. Matrix Anal. Appl.*, vol. 12, no. 1, pp. 128–145, Jan. 1991.

[9] J. Chun and T. Kailath, "Generalized displacement structure for block-Toeplitz, Toeplitz-block, Toeplitz-derived matrices," *SIAM J. Matrix Anal. Appl.*, vol. 15, no. 1, pp. 114–128, Jan. 1994.

[10] T. Kailath and A. Sayed, "Displacement structure: Theory and applications," *SIAM Review*, vol. 37, no. 3, pp. 297–386, Sep. 1995.

[11] J. K. Zhang, J. Liu, and K. M. Wong, "Linear Toeplitz space time block codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT 05)*, Adelaide, Australia, Sep. 4–9, 2005, pp. 1942–1946.