Reducing Computational Complexity for DBZF Precoding in xDSL Downlinks

I V Semushin and J V Tsyganova

Ulyanovsk State University, Lev Tolstoy street 42, Ulyanovsk, Russia, 432017

e-mail: kentvsem@gmail.com, tsyganovajv@gmail.com

Abstract. However broad the Decomposition-based Zero-forcing (DBZF) precoder acceptance may be, reducing the computational complexity of its implementation is an absolute necessity for the VDSL networking professionals. The paper digs deeper into this problem from the perspective of matrix inversion which is inherent in the very nature of the DBZF. Eight strategies considered here differ in mode of action: four of them include matrix inversion, and four others drop implementing the procedure. While the baseline strategy itemized under No. 1 acts with the Gaussian $LU$-decomposition, strategy No. 2 deals with the Jordanian $LU$-decomposition thereby enabling mild reduction of the operation count. Strategy No. 3 works for more significant reduction as it operates with the elimination form of the inverse matrix. The most cost-cutting are strategies excluding the question of matrix inversion and replacing it by far more straightforward linear system solution, as it is in Strategy No. 4. Both cases No. 3 and No. 4 are examined in their two subcases marked (1) and (2) differing in Gaussian (1) or Jordanian (2) subversions of the same $LU$-decomposition. An alternative strategy No. 5 uses the least squares-based square-root-type sequential system solution, and it is the most accurate computational procedure when compared with other strategies. Strategy No. 6 is different in that it implements the $QR$-decomposition of the channel transition matrix $H$, and it is the most numerically robust one against the ill-conditioned matrix $H$.

1. Introduction

In the last two decades, communication technologies and transmission equipment are developing at an unprecedented pace to provide for both the residential and mobile access. Broadband wireline access networks offer promising and stable bandwidth to resident user premises. Their services include many modern network applications such as video-streaming, file sharing, telecommuting, online gaming, video-conferencing, and others. It became possible in the early 2000s with the invention of Very High-Speed Digital Subscriber Line (VDSL).

Although passive optical networks (PON) is the most popular access network worldwide, countries with abundant copper line resources make good use of xDSL (such as VDSL) by taking advantage of existing telephone lines. The combined technology called ‘fiber-to-the-x’ (FTTx), where ‘x’ may stand for ‘N=node,’ or ‘C=curb,’ or ‘Cab=cabinet,’ or ‘B=building,’ delivers both low deployment cost and better performance. In densely populated areas or cities, many customers are within 1.5 km of the central office (CO) or Local Exchange (LEx). In such cases, VDSL can be deployed directly from the CO. When fiber extends deeper into the network, public service carriers deploy VDSL from the optical network unit (ONU) in a configuration known as ‘fiber-to-the-cabinet’ (FTTCab) [1, p. 6]. In this case, PON connects the optical signal to
the cabinet ONU, and then, telephone line or twisted pair (TP) will carry the signal to user premises using xDSL. That way, hybrid fiber-copper systems deploying VDSL in the last mile to the customer premises (CPs) make the optical network core closer to the customer, and so offer sufficient bandwidth retaining the edge over pure fiber networks as a more economical solution [2, p. 15], [3, p. 55].

The Organization for Economic Co-operation and Development publishes broadband-related information of its members every year (Figure 1). As China is not an OECD member but owns the most broadband users in the world, China’s broadband information is also put together in some charts with OECD information for reference. This information attests to the fact that in our age of high technology betting on fiber, the DSL remains an attractive choice to transport the data of high speed real time processing systems by using Public Switched Telephone Network (PSTN) with Public Old Telephone Service (POTS).

![Figure 1. OECD Broadband Portal, Dec. 2013: Number of broadband subscribers...](https://example.com/broadband-chart.png)

However, a crucial problem in VDSL networks limiting both the data rate and reach of service is the phenomenon known as near-end crosstalk (NEXT) and far-end (FEXT) crosstalk. It is known that FEXT has more huge consequences on shorter VDSL lines: “it can dominate noise profiles” [1, p. 11]. Once the modems transmitting signals in the downstream direction are collocated at the CO, a technique of crosstalk precoding can be applied to each modem’s signal before transmission. A comprehensive review of precoding techniques for digital communication systems is recently given in [4]. A near-optimal linear crosstalk precoder [5] helps to cope with this problem for wireline downstream VDSL. This precoder dates back to 2004 as can be seen from a broadcast draft paper version [5]. Thanks to a low complexity and no-need-for-additional-arithmetic at the receiver-side, it gained broader acceptance and further consideration as in [6, p. 34–35].
This solution termed in work last cited as Decomposition-based Zero-forcing Precoder is producing the desired effect in crosstalk cancellation, but along with this, it poses a problem of computation complexity. The point is that the precoder computations include the operation of matrix inversion and so may have a meaningful effect on VDSL power consumption as stressed in [1, p. 20]. If VDSL modems are deployed from the ONU, which is typically located in a small curbside cabinet with no cooling or temperature control, VDSL power consumption must be very low. Besides, to fit in the ONU, VDSL line cards must also be small. For the preceding reasons, networking professionals refer to reducing the complexity of big matrices inversion as a ‘perennial, evergreen topic’ [7, pp. 39, 55–70].

The paper investigates the problem of computational complexity of different algorithms related to precoder design—both including and avoiding matrix inversion. Section 2 recalls mind to the subject. Section 3 gives two formal answers to the question of how to obtain the sought-for solution for the problem above formulated regarding inverse matrix \( A^{-1} \) intended to be a precoder \( P \). Section 4 presents the baseline solution to the system \( Ax' = (Gx) \) using any one of three forms of \( LU \)-decomposition for matrix \( A \). Section 5 describes the Jordanian version of \( LU \)-decomposition for \( A \) with the intention of computing \( A^{-1} \) as precoder \( P \). Section 6 uses the baseline \( LU \)-decomposition to obtain the elimination form of \( A^{-1} \). Section 7 proposes an off-the-beaten-path \( LU \)-decomposition based solution for precoding without direct computing \( A^{-1} \). Section 8 introduces another non-mainstream alternative in the form of least squares-based square root sequential system solution to avoid \( A^{-1} \). Section 9 comes up with solution to the system \( Ax' = (Gx) \) by way of \( QR \)-decomposition for channel matrix \( H \) to prevent explicit calculation of \( A^{-1} \) formally meant to be the precoder \( P \). Section 10 makes tradeoff study from the perspective of matrix inversion in precoder design. Section 11 contains the obtained results of numerical experiments conducted in MATLAB. The last section concludes the paper with some recommendation for the network service management.

Throughout this paper we use the following notations:
- the complex \( n \) vector space by \( \mathbb{C}^n \) and the complex \( m \times n \) matrix space by \( \mathbb{C}^{m \times n} \),
- the real \( n \) vector space by \( \mathbb{R}^n \) and the real \( m \times n \) matrix space by \( \mathbb{R}^{m \times n} \),
- \( n \)-vectors of the same dimension \( n \) by lowercase latin letters,
- \( n \times n \) matrices by uppercase latin letters,
- scalars by small Greek letters.

2. Channel Model and DMT Transmission

Because of the below reasons VDSL systems are treated as Multiple Input Multiple Output (MIMO) systems.

(i) VDSL system is DMT (Discrete Multi-tone Transmission)-based. In this method, the allocated frequency band (or channel) is separated into many frequency subbands (or subchannels as they are specified at times). DMT uses the fast Fourier transform (FFT) algorithm for signal modulation (before transmission) and demodulation (at receiving side). The transmission process runs on each tone \( k \), i.e., on the \( k \)-th carrier frequency \( f \) in the \( k \)-th subchannel, \( k = 1, \ldots, K \) (Figure 2).

(ii) Modulated data are passed to \( N \) users in parallel through \( N \) twisted pairs. As a general rule, \( N \) individual TPs are grouped in binder-groups of 4 to 10 cables, and 50 to 100 TPs are bundled together into a cable. So, \( N \) may be from two to ten hundred [8, p. 5], and thus the FEXT is rendered the most dangerous phenomenon (Figure 3).

The \( k \)-th subchannel is modeled by the \( k \)-th complex valued channel transfer matrix \( H_k \triangleq H_k(f) \in \mathbb{C}^{N \times N} \) on each tone \( k \) (see Figure 2). The \( n \)-th diagonal elements of \( H_k \) correspond to the direct channel coefficients of the different TPs and describe the impact of
The channel transfer matrices $H_k(f)$ are complex-valued $N \times N$ matrices; $k = 1, \ldots, K$, the number of subchannels (or tones).

$y_k = H_k x_k + v_k$ \hspace{1cm} (1)

where $v_k$ is the additive noise on tone $k$. It is comprised of thermal noise, alien crosstalk, radio frequency interference (RFI), etc. [5], and is frequently modeled as an additive white Gaussian noise (AWGN).

3. Signal Transmission Modification with Gain $G$ and Precoder $P$

To control the transmit power spectral density (PSD) of $n$ user on tone $k$, which is denoted $s_k^n \triangleq \mathbb{E}\{|x_k^n|^2\}$, the transmit signal $x_k \triangleq [x_k^1, \ldots, x_k^N]^T \in \mathbb{C}^N$ is pre-multiplied by a diagonal matrix $G_k \triangleq \text{diag} \{g_k^1, \ldots, g_k^N\} \in \mathbb{R}^{N \times N}$. Then the resulting vector $(G_k x_k)$ is pre-disturbed by the precoder matrix $P_k \in \mathbb{C}^{N \times N}$ to obtain the channel input signal $x'_k = P_k(G_k x_k)$. Matrix $P_k$
is to be specific for every tone \( k \), while the tones number into thousands \( K = 2048 \) as is in a typical case or may reach 4096). So, relation (1) is replaced (Figure 4, b) by

\[
y'_k = H_k x'_k + v_k = H_k P_k (G_k x_k) + v_k.
\]  

(2)

**Question:**

- How to design Precoder \( P_k \) in order to cancel crosstalk in \( y'_k \), i.e., to ensure an element-wise relation between \( y'_k \) (the received vector) and \( x_k \) (the transmit vector)? In other words, we need:

\[
y'_k = \alpha_k n x'_k + v'_k \quad \text{with a scalar } \alpha_k, \ n = 1, \ldots, N.
\]  

(3)

**One formal ("the-beaten-path") solution:**

1. Premultiplying \( x_k \) by \( G_k \) yields \( (G_k x_k) = [g'_k x_k^1, \ldots, g'_k x_k^N]^T \).
2. Denote \( F_k \triangleq \operatorname{diag} \left[ \left( h_k^{1,1} \right)^{-1} \right. \left| \cdots \left| \left( h_k^{N,N} \right)^{-1} \right] \right. \); therefore \( F_k^{-1} \triangleq \operatorname{diag} \left[ h_k^{1,1} \right. \left| \cdots \left| h_k^{N,N} \right] \right. \).
3. Denote \( A_k \triangleq F_k H_k = \begin{bmatrix} 1 & \left( h_k^{1,1} \right)^{-1} & h_k^{1,2} & \cdots & \left( h_k^{1,N} \right)^{-1} & h_k^{1,N} \\ \\
\left( h_k^{2,1} \right)^{-1} & 1 & \cdots & \left( h_k^{2,N} \right)^{-1} & h_k^{2,N} \\ \\
\left( h_k^{3,1} \right)^{-1} & \cdots & \left( h_k^{3,N} \right)^{-1} & \cdots & \left( h_k^{3,N} \right)^{-1} \\ \\
\vdots & \cdots & \cdots & \cdots & \cdots \\ \\
\left( h_k^{N,1} \right)^{-1} & \left( h_k^{N,N} \right)^{-1} & \cdots & \left( h_k^{N,N} \right)^{-1} & 1 \end{bmatrix} \).
4. Denote \( P_k \triangleq A_k^{-1} \), so that \( P_k = H_k^{-1} F_k^{-1} \). This relation means the decomposition:

\[ H_k = F_k^{-1} A_k \].

5. With such \( P_k \), form \( x'_k = P_k (G_k x_k) \), as it is shown in the above Figure 4, b to obtain at the receiving side

\[
y'_k = H_k x'_k + v_k = H_k P_k (G_k x_k) + v_k = H_k H_k^{-1} F_k^{-1} (G_k x_k) + v_k = F_k^{-1} (G_k x_k) + v_k
\]

\[
y'_k = \left[ h_k^{1,1} g'_k x_k^1 \mid \cdots \mid h_k^{N,N} g'_k x_k^N \right]^T + v_k; \quad y'_k = \alpha_k n x'_k + v'_k, \ \alpha_k = h_k^{n,n} g'_k, \ n = 1, \ldots, N.
\]

Steps from 1 to 5 in the above list are presented to substantiate the known answer to the above question:

\[ P_k \text{ should be defined by formula } P_k \triangleq A_k^{-1} \]
Thus, decomposition-based zero-forcing precoder $P_k$ is the inverse of the normalized (i.e., unit diagonal) channel matrix $A_k$ [6, p. 34–35]. Formally, $A_k$ equals the channel matrix $H_k = [h_{n,m}^k] \in \mathbb{C}^{N \times N}$ whose row and column indices $n$ and $m$ for entries $h_{n,m}^k$ run the range $1, \ldots, N$, premultiplied by matrix $F_k$. Such decomposing matrix $H_k$ into $F_k^{-1} \times A_k$ and precoding only with $P_k$ leads to a high transmission overhead of the CO/ONU due to the increased computational complexity if only matrix $P_k$ is to be precomputed in explicit form by inverting matrix $A_k$.

Another formal (slightly modified) solution:

1. Premultiplying $x_k$ by $G_k$ yields $(G_kx_k) = [g_k^1 x_k^1, \ldots, g_k^N x_k^N]^T$.

2. Denote $F_k \triangleq \text{diag} \left[ \left( h_{1,1}^k \right)^{-1}, \ldots, \left( h_{N,N}^k \right)^{-1} \right]$; therefore $F_k^{-1} \triangleq \text{diag} \left[ h_{1,1}^k, \ldots, h_{N,N}^k \right]$.

3. Find $Q_k$ and $R_k$ resulting in $H_k = Q_kR_k$ by one of the left-side orthogonal transforms $T_k \triangleq Q_k^T$ to obtain $R_k = T_kH_k$ (Householder or Givens or GramSchmidt) [10, pp. 107–140].

4. Find $R_k^{-1}$ and define $P_k \triangleq R_k^{-1}Q_k^TF_k^{-1}$.

5. With such $P_k$, produce $x'_k = P_k(G_kx_k)$ on entering the channel $H_k$—as it is shown in the above Figure 4, $b$—to yield the receiving side vector: $y'_k = H_kx'_k + v_k = H_kP_k(G_kx_k) + v_k = H_kH_k^{-1}F_k^{-1}(G_kx_k) + v_k = F_k^{-1}(G_kx_k) + v_k$.

What is different in the second solution is that steps $\text{(3)}$ and $\text{(4)}$ lean heavily on the orthogonal $Q_kR_k$ technique, and what makes it formally equal to the first one is that $P_k \triangleq H_k^{-1}F_k^{-1}$ and $A_k \triangleq F_kQ_kR_k$ are the same, whilst calculated differently.

Below for the sake of simplicity, we omit index $k$ and compare different execution strategies for obtaining the channel input signal $x'_k$ satisfying $A_kx'_k = (G_kx_k)$.

Note: We use the same symbol $\Sigma_i$ with subindex $i$ to designate the $i$-th strategy and its complexity understood as the total multiplication/division count in it. Calculations for Section 4 to Section 9 are rigorously substantiated in [9] by summing finite series of positive numbers.

4. Baseline solution $\Sigma_1$: Gaussian $LU$-decomposition followed by the forward and backward substitutions to designate $A^{-1}$ as precoder $P$

The $LU$-decomposition of $N \times N$ matrix $A$ is well known and may be performed by a variety of ways [10, pp. 27–81, 117–120, 137]. Take for consideraton: (a) Gauss column sweep algorithm; (b) Crout’s reduction algorithm; and (c) bordering algorithm, in each case matrix $L$ being lower triangular, matrix $U$ unit diagonal upper triangular, and non-trivial elements of $L$ and $U$ overwrite the given matrix $A$.

Step 1: $A = LU$. Irrespective of differences in the hierarchy of actions, all the algorithms have the same complexity: $\Sigma_1^{\text{Step } 1} = (N - 1)N(N + 1)/3$.

Step 2: (a) From system $LW = I$, we find $W$. At this point, $\Sigma_1^{\text{Step } 2(a)} = N(N + 1)(N + 2)/6$. (b) From system $UX = W$, we obtain $X \triangleq A^{-1}$, $P := X$. At this point, $\Sigma_1^{\text{Step } 2(b)} = (N - 1)N^2/2$. Totalling: $\Sigma_1^{\text{Step } 2} = \Sigma_1^{\text{Step } 2(a)} + \Sigma_1^{\text{Step } 2(b)} = N(2N^2 + 1)/3$.

Step 3: $x' = Px$. At this point, $\Sigma_1^{\text{Step } 3} = N^2$.

Consequently, strategy $\Sigma_1$ has the following complexity:

$$\Sigma_1 = \Sigma_1^{\text{Step } 1} + \Sigma_1^{\text{Step } 2} + \Sigma_1^{\text{Step } 3} = N^2(N + 1).$$ (4)
5. Alternative $\Sigma_2$: Jordanian $LU$-decomposition followed by the forward substitution only to designate $A^{-1}$ as precoder $P$

Jordan’s method provides for complete elimination procedure. Operation count shows:

- Step 1: $A = LU$ with resulting in $L$ and $U^{-1}$: $\Sigma_2^{\text{Step 1}} = (N - 1)N^2/2$.
- Step 2: $L^{-1}$ by the instrumentality of elementary matrices: $\Sigma_2^{\text{Step 2}} = (N - 1)N(N + 1)/6$.
- Step 3: $P := A^{-1} = U^{-1}L^{-1}$: $\Sigma_2^{\text{Step 3}} = (N - 1)N(N + 1)/3$.
- Step 4: $x' = Px$. At this point, $\Sigma_2^{\text{Step 4}} = N^2$.

Consequently, strategy $\Sigma_2$ has the following complexity:

$$\Sigma_2 = \Sigma_2^{\text{Step 1}} + \Sigma_2^{\text{Step 2}} + \Sigma_2^{\text{Step 3}} + \Sigma_2^{\text{Step 4}} = N(N + 1)(2N - 1)/2. \quad (5)$$

6. Alternative $\Sigma_3$: Gaussian $LU$-decomposition followed by finding the elimination form of $A^{-1}$ to designate it as precoder $P$

This operational scheme differs significantly in character from that used in the above. Namely, $L^{-1}$ and $U^{-1}$ are found after the $LU$-decomposition by means of step-by-step elementary matrix multiplying [10, pp. 42–46]. The algorithm is comprised of the following five steps:

- Step 1: $A = LU$, as in strategy $\Sigma_1$: $\Sigma_3^{\text{Step 1}} = (N - 1)N(N + 1)/3$.
- Step 2: $L^{-1}$ by the instrumentality of elementary matrices, as in strategy $\Sigma_2$: $\Sigma_3^{\text{Step 2}} = (N - 1)N(N + 1)/6$.
- Step 3: $U^{-1}$ by the instrumentality of elementary matrices: $\Sigma_3^{\text{Step 3}} = (N - 2)(N - 1)N/6$.
- Step 4: $P := A^{-1} = U^{-1}L^{-1}$, as in strategy $\Sigma_2$: $\Sigma_3^{\text{Step 4}} = (N - 1)N(N + 1)/3$.
- Step 5: $x' = Px$. At this point, $\Sigma_3^{\text{Step 5}} = N^2$.

Consequently, strategy $\Sigma_3$ has the following complexity:

$$\Sigma_3 = \Sigma_3^{\text{Step 1}} + \Sigma_3^{\text{Step 2}} + \Sigma_3^{\text{Step 3}} + \Sigma_3^{\text{Step 4}} + \Sigma_3^{\text{Step 5}} = N(N + 1)(4N - 1)/6. \quad (6)$$

7. Alternative $\Sigma_4$: Gaussian $LU$-decomposition followed by system solution to avoid finding and designating $A^{-1}$ as precoder $P$

We now proceed to avoid matrix inversion in precoding.

- Step 1: $A = LU$, as in strategy $\Sigma_1$: $\Sigma_4^{\text{Step 1}} = (N - 1)N(N + 1)/3$.
- Step 2: (a) By solving $Lw = I$, we obtain $w$. At this point, $\Sigma_4^{\text{Step 2(a)}} = N(N + 1)/2$.
  (b) By solving $Ux' = w$, we obtain $x'$. At this point, $\Sigma_4^{\text{Step 2(b)}} = N(N - 1)/2$. Totalling:
  $$\Sigma_4^{\text{Step 2}} = \Sigma_4^{\text{Step 2(a)}} + \Sigma_4^{\text{Step 2(b)}} = N^2. \quad (7)$$

Consequently, strategy $\Sigma_4$ has the following complexity:

$$\Sigma_4 = \Sigma_4^{\text{Step 1}} + \Sigma_4^{\text{Step 2}} = N(N^2 + 3N - 1)/3. \quad (7)$$

8. Alternative $\Sigma_5$: Least squares–based square root sequential system solution to avoid finding and designating $A^{-1}$ as precoder $P$

As a test specimen, let us take Potter’s square root least squares (LS) algorithm [10, pp. 250–251]. It avoids matrix inversion by means of row-by-row matrix processing while solving $Ax' = (Gx)$. Modelling on the LS-algorithm [10, p. 250]) and utilizing complex conjugate matrix transposition (where needed), we obtain:

I. Initialization. Initial values: LS-estimator $x_0$ and its covariance $P_0$. A’priori data is lacking, it means $x_0 = 0$ and $P_0 = \varepsilon^{-2} \text{diag } [1 \mid \cdots \mid 1]$ with as possible small $\varepsilon$, theoretically $\varepsilon \to 0$.

Let $S_0$ be a matrix square root of $P_0$, i.e. $S_0 = P_0^{1/2} = \varepsilon^{-1} \text{diag } [1 \mid \cdots \mid 1]$. Set $\hat{x} := x_0$ and $\hat{S} := S_0$. 

7
II. Processing the n-th matrix row \( [A | (Gx)]_n \) of data \( [A | (Gx)] : a \triangleq a_n \) is the n-th row of \( A \) (viewed as a column) and \( z \triangleq z_n \) is the n-th item of \( (Gx) \) in system \( Ax' = (Gx) \). Cycle on \( n = 1, \ldots, N \):

\[
f = \tilde{S}^H a; \quad \alpha = f^H f; \quad K = \tilde{S} f / \alpha; \quad \tilde{S} = \tilde{S} - K f^H; \quad \tilde{x} = \tilde{x} + K(z - A\tilde{x}).
\]

III. Propagating the instantaneous solution estimator to the next matrix row for Item II repetition:

\[
\tilde{S} := \tilde{S}, \quad \tilde{x} := \tilde{x}.
\]

On exit from Item III after having \( n = N \) at Item II, one obtains \( \tilde{x} \) as the desired solution \( x' \) for system \( Ax' = (Gx) \).

Counting shows that strategy \( \Sigma_5 \) has the following complexity:

\[
\Sigma_5 = N(3N^2 + 4N).
\]  

9. Alternative \( \Sigma_6 \): Channel QR-decomposition based system solution to avoid finding and designating \( A^{-1} \) as precoder \( P \)

This algorithm comprises, in its most streamlined layout, three steps as follows.

Step 1: \( x'' := F^{-1}(Gx) \). It requires \( \Sigma_6^{\text{Step 1}} = N \) multiplications/divisions (m/ds).

Step 2: \( \tilde{Y} := T [H | x''] \). The \( N \times (N+1) \) matrix \( \tilde{Y} \triangleq \{\tilde{y}_{i,j}\}_{i=1}^{N} \) is a scratch array. Prior to action, \( \tilde{Y} := [H | x''] \). Let \( T \) be the Householder transformation. To present its pseudocode more compactly, use notation \( \tilde{O}_{N-k} \) for any zero-valued vector of dimension \( N - k \):

```plaintext
for k = 1 to N - 1
begin
    y := \tilde{y}_k \triangleq [\tilde{y}_{k,k} | \tilde{y}_{k+1,k} | \cdots | \tilde{y}_{N,k}]^T; \quad s := -\text{sign} (\tilde{y}_{k,k}) \sqrt{y^T y};
    u := [\tilde{y}_{k,k} - s | \tilde{y}_{k+1,k} | \cdots | \tilde{y}_{N,k}]^T; \quad ||u||^2 = u^T u; \quad \tilde{y}_k := [s \mid \tilde{O}_{N-k}]^T
end
for j = k + 1 to N + 1
begin
    y := \tilde{y}_j \triangleq [\tilde{y}_{j,k} | \tilde{y}_{j,k+1} | \cdots | \tilde{y}_{j,N}]^T; \quad \gamma = 2 (y^T u) / ||u||^2; \quad \tilde{y}_j := y - \gamma u
end
```

As such, Step 2 produces the upper right triangular matrix \( R = TH \) and column \( b = Tx'' \). It requires \( \Sigma_6^{\text{Step 2}} = \left[ N(N+1)(2N+7)/3 \right] \) m/ds and \( (N - 1) \) square roots on top of it.

Step 3: Solving system \( Rx' = b \) requires \( \Sigma_6^{\text{Step 3}} = \frac{N(N+1)}{2} \) m/ds.

As measured in m/ds, strategy \( \Sigma_6 \) has the following complexity:

\[
\Sigma_6 = \Sigma_6^{\text{Step 1}} + \Sigma_6^{\text{Step 2}} + \Sigma_6^{\text{Step 3}} = \left[ 4N^3 + 21N^2 + 23N - 36 \right] / 6.
\]

10. Complexity: Implementation tradeoff analysis in precoder design from the perspective of matrix inversion

For the six strategies considered, we have relations (4), (5), (6), (7), (8), and (9) to characterize their complexity. To intercompare them, we introduce a complexity trend index associated with passing from \( \Sigma_i \) to \( \Sigma_j \): \( \Delta_{ij} \triangleq \Sigma_i - \Sigma_j \). We also propose the limit relative indices \( \delta_{ij} \triangleq \lim_{N \to \infty} \Sigma_i / \Sigma_j \) obtained for \( N \to \infty \) when moving from \( \Sigma_i \) to \( \Sigma_j \), and sum up the exact results in Table 1.
Keeping in mind the necessity of expanded analysis, we record a fact to be used later, namely

\begin{align*}
\Delta_{14} &= \frac{N}{2} (2N^2 + 1) \\
\Delta_{24} &= \frac{N}{6} (N - 1)(4N + 1) \\
\Delta_{44} &= \frac{N}{6} (N - 1)(2N - 1) \\
\Delta_{12} &= \frac{N}{2} (N + 1) \\
\Delta_{23} &= \frac{N}{3} (N - 1)(N + 1) \\
\Delta_{13} &= \frac{N}{6} (N + 1)(2N + 1) \\
\Delta_{15} &= -N^2(2N + 3) \\
\Delta_{35} &= -\frac{N}{2}(14N^2 + 21N + 1) \\
\Delta_{45} &= -\frac{N}{3}(8N^2 + 9N + 1) \\
\Delta_{16} &= \left[2N^3 - 15N^2 - 23N + 36\right]/6 \\
\Delta_{64} &= \left[2N^3 + 15N^2 + 19N - 36\right]/6
\end{align*}

The limit relative indices for processor off-loading prove to be as follows:
\begin{align*}
\delta_{14} &= 3, \ \delta_{24} = 3, \ \delta_{34} = 2 \Rightarrow \text{processor load dropping.} \\
\delta_{12} &= 1, \ \delta_{23} = 3/2, \ \delta_{13} = 3/2 \Rightarrow \text{processor load keeping or dropping.} \\
\delta_{15} &= 1/3, \ \delta_{35} = 2/9, \ \delta_{45} = 1/9 \Rightarrow \text{processor load increasing.} \\
\delta_{16} &= 3/2 \Rightarrow \text{processor load dropping.} \\
\delta_{64} &= 2 \Rightarrow \text{processor load dropping.}
\end{align*}

### Table 1. Processor off-loading \( \Delta_{ij} \) at passing from strategy \( \Sigma_i \) to \( \Sigma_j \).

| \( \Delta_{ij} \) | \( \Delta_{14} \) | \( \Delta_{24} \) | \( \Delta_{44} \) | \( \Delta_{12} \) | \( \Delta_{23} \) | \( \Delta_{13} \) | \( \Delta_{15} \) | \( \Delta_{35} \) | \( \Delta_{45} \) | \( \Delta_{16} \) | \( \Delta_{64} \) |
|-------------------|-----------------|--------------|----------------|-----------------|-----------------|-----------------|-----------------|----------------|----------------|-----------------|-----------------|
| \( \frac{N}{2} (2N^2 + 1) \) | \( \frac{N}{6} (N - 1)(4N + 1) \) | \( \frac{N}{6} (N - 1)(2N - 1) \) | \( \frac{N}{2} (N + 1) \) | \( \frac{N}{3} (N - 1)(N + 1) \) | \( \frac{N}{6} (N + 1)(2N + 1) \) | \( -N^2(2N + 3) \) | \( -\frac{N}{2}(14N^2 + 21N + 1) \) | \( -\frac{N}{3}(8N^2 + 9N + 1) \) | \( \left[2N^3 - 15N^2 - 23N + 36\right]/6 \) | \( \left[2N^3 + 15N^2 + 19N - 36\right]/6 \) |

11. Numerical Experiments

Keeping in mind the necessity of expanded analysis, we record a fact to be used later, namely that we rename \( \Sigma_3 \) as \( \Sigma_{3(1)} \) and \( \Sigma_4 \) as \( \Sigma_{4(1)} \). In parallel with them, we will test two more versions appropriately labeled as \( \Sigma_{3(2)} \) and \( \Sigma_{4(2)} \) which are different in that they make use of Jordanian \( LU \)-decomposition instead of Gaussian one. Thus, we test the following seven strategies for finding the desired solution \( x' \):

- \( \Sigma_1 \): Gaussian \( LU \)-decomposition followed by the forward and backward substitutions to designate \( A^{-1} \) as precoder \( P \);
- \( \Sigma_2 \): Jordanian \( LU \)-decomposition followed by the forward substitution only to designate \( A^{-1} \) as precoder \( P \);
- \( \Sigma_{3(1)} \): Gaussian \( LU \)-decomposition followed by finding the elimination form of \( A^{-1} \) to designate it as precoder \( P \);
- \( \Sigma_{3(2)} \): Jordanian \( LU \)-decomposition followed by finding the elimination form of \( A^{-1} \) to designate it as precoder \( P \);
- \( \Sigma_{4(1)} \): Gaussian \( LU \)-decomposition followed by system solution to avoid finding and designating \( A^{-1} \) as precoder \( P \);
- \( \Sigma_{4(2)} \): Jordanian \( LU \)-decomposition followed by system solution to avoid finding and designating \( A^{-1} \) as precoder \( P \);
- \( \Sigma_5 \): Least squares-based square root sequential system solution to avoid finding and designating \( A^{-1} \) as precoder \( P \);
- \( \Sigma_6 \): Channel \( QR \)-decomposition based system solution to avoid finding and designating \( A^{-1} \) as precoder \( P \).

The first four strategies suppose that having \( A^{-1} \) found, we designate the product \( \hat{x}' = P(Gx) \) with \( P := A^{-1} \) as the desired solution \( x' \). In the four last-named strategies, \( \hat{x}' \) is the result of the system \( Ax' = (Gx) \) solving phase. Apart from method delivering the solution \( \hat{x}' \), it is essential to know the estimated accuracy \( e = \hat{x}' - x' \). Because the precise meaning \( x' \) is to be supposed unknown, all that remains is to verify the residual \( r = Ax' - (Gx) \). As is evident, \( A^{-1}r = e \). Considering that matrices \( A \) for the downstream VDSL channels would
possess the property of row-wise diagonal dominance [5, p. 860]—and so they are meant to be well conditioned—one may with good reason estimate accuracy by $||r||_\infty = ||A\vec{x}' - (Gx)||_\infty$ for all afore-mentioned strategies.

We have implemented eight m-functions in MATLAB. Using these implementations, we conducted computational experiments with the set of test matrices $H_k \in \mathbb{C}^{N \times N}$ on one arbitrarily selected tone $k$ with $N = 20, 40, 80, \ldots, 800$. For each matrix $H_k$ and given complex vector $(Gx)$, we saved the computed solution $\vec{x}'$ and the execution time (in the sec) and the accuracy of computations $||r||_\infty$. Figure 5 and Table 2 show the obtained results. One can see from there that the new proposed strategy $\Sigma_5$ has the almost minimal execution time and the best accuracy of computations for all test matrices. As for $\Sigma_6$, its accuracy did not go into Table 2 because it is in the mould of $\Sigma_5$’s accuracy. That gives rise to a suggestion that $\Sigma_5$ is a numerically efficient method for practical applications concerning both of computational complexity and accuracy.

![Figure 5. Execution time (sec) according to strategy $\Sigma_i$.](image)

12. Conclusion
In this paper we considered eight strategies $\Sigma_i$ involving ($\Sigma_1$, $\Sigma_2$, $\Sigma_3(1)$, $\Sigma_3(2)$) or avoiding ($\Sigma_4(1)$, $\Sigma_4(2)$, $\Sigma_5$, $\Sigma_6$) explicit matrix inversion from the perspective of reducing computational complexity for DBZF precoding in xDSL downlinks.

- The smallest effect of descending complexity holds for moving from $\Sigma_1$ to $\Sigma_2$, $\Sigma_1 \rightarrow \Sigma_2$.
- Making the transition from $\Sigma_2$ to $\Sigma_3$ lowers the limit complexity (for big matrices) by 50%.
Table 2. Accuracy of computations according to strategy $\Sigma_i$.

| $N$  | $\Sigma_1$  | $\Sigma_2$  | $\Sigma_{3(1)}$ | $\Sigma_{3(2)}$ | $\Sigma_{4(1)}$ | $\Sigma_{4(2)}$ | $\Sigma_5$  |
|------|-------------|-------------|----------------|----------------|----------------|----------------|-------------|
| 20   | 2.2225e-16  | 2.2225e-16  | 2.2225e-16     | 3.3309e-16     | 3.3306e-16     | 6.6613e-16     |
| 40   | 1.5142e-15  | 1.5143e-15  | 1.5143e-15     | 1.3528e-15     | 7.8773e-16     |
| 100  | 1.1770e-12  | 2.7423e-12  | 4.6210e-12     | 3.0986e-12     | 1.0900e-12     |
| 200  | 1.1803e-10  | 9.2966e-10  | 2.7933e-10     | 2.7968e-10     | 4.5409e-12     |
| 400  | 1.3830e-10  | 5.5337e-09  | 1.8709e-09     | 1.6365e-09     | 5.1123e-12     |
| 800  | 4.3237e-08  | 5.6475e-07  | 8.6912e-09     | 8.6935e-09     | 3.7705e-11     |

- Thanks to $\Sigma_1 \rightarrow \Sigma_4$, the maximum reduction effect $\delta_{14} = 3$ can be achieved as $N$ increases. It is of the primary interest for network carriers management.
- The fifth strategy $\Sigma_5$ seems a bit unusual or even ‘exotic.’ As $N$ tends to infinity, it imposes a thrice as much computational load as compared to $\Sigma_1$. Nevertheless, our MATLAB experiments show that it is the most accurate and fast computational procedure when compared with other strategies. While $\Sigma_5$ is affording good accuracy due to operating with square roots of matrix $A^TA$ that appears in the LS normal equations, the MATLAB quickness indicates that $\Sigma_5$ has coding tricks in store for programmers. In that context, this new proposed strategy $\Sigma_5$ can be useful when hardware implemented.
- Although $\Sigma_6$ requires two times more computations than $\Sigma_4$, it is attractive due to its high numerical robustness against the ill-conditioned channel transfer matrix $H$.  

13. References
[1] Jacobsen K S 1999 Providing the Right Solution for VDSL Texas Instruments White Paper Version 1.05 (San Jose, CA).
[2] 2001 Aspects of VDSL Evolution Technical Report TR-040 Digital Subscriber Line Forum (Formerly WT-047v8).
[3] Ma Y and Jia Z 2017 Fiber-Wireless Convergence in Next-Generation Communication Networks – Systems, Architectures, and Management 1st ed M Tornatore, G K Chang and G Ellinas (Switzerland: Springer International Publishing).
[4] Kuo C, Tsai S, Tadjpour L and Chang Y 2008 Precoding Techniques for Digital Communication Systems (Springer, US).
[5] Cendrillon R, Ginis G, Van den Bogaert E and Moonen M 2007 A Near-Optimal Linear Crosstalk Precoder for Downstream VDSL IEEE Transactions on Communications 55 pp 860–863.
[6] Dungen M 2016 Crosstalk Mitigation Techniques for Digital Subscriber Line Systems (Doktor-Ingenieurin (Dr.-Ing.) genehmigte Dissertation) Ph.D. thesis (Technischen Universität Hamburg-Harburg (TUHH), Germany).
[7] Semushin I V 2016 Adaptive State Prediction Methods for Big Data Modeling and Design The 3rd Professor Day Academic Conference (Huawei Russian Research Center, Moscow).
[8] Holte N 2002 Transmission properties of pair cables Digital Kommunikasjon (Norway: NTNU).
[9] Semushin I V 2017 Big Matrix Inverses Revisited: from Formalities to Implementation Automation of Control Processes 4(50) 36–41 (in Russian).
[10] Semushin I V 2011 Computational Methods of Linear Algebra and Estimation (Ulyanovsk: ULISTU Publishers).

Acknowledgments
We are grateful to Dr. Li Jiang, Pavel Lanchikov and Dr. Victor Dzhigan of Huawei Russian Research Centre, for their keen interest to the topic and providing us with the VDSL channel characteristics that had proved to be useful in our above numerical experiments. This research was funded by Russian Foundation for Basic Research and the government of the Ulyanovsk region of the Russian Federation, grant 18-47-730001 p_a.