TAMPERE UNIVERSITY OF TECHNOLOGY

LOW-COMPLEXITY ALGORITHM FOR INVERSION OF SPECIAL MATRICES IN SDR SYSTEMS

David Guevorkian
Tampere University of Technology, Finland
Currently with Nokia-Siemens Networks, Espoo, Finland

Kim Rounioja
Renesas Mobile Corporation, Oulu, Finland

Jarmo Takala
Tampere University of Technology, Finland

SDR-WInnComm 2013, Washington DC, Jan. 2013

Outline

- Introduction
- The problem
- Conventional method
- Proposed method
- Complexity comparison
- Conclusion

Introduction

- Inversion of complex-valued circulant Hermitian matrices is needed in SDR
  - e.g., in channel estimation of advanced OFDM systems such as HSDPA and WCDMA downlink receivers
- For such matrices the complexity of *O(NlogN)* may be reached by using complex-valued Fast Fourier Transform (FFT) algorithm
- In this work, we propose Type 1 Discrete Cosine and Sine Transforms (DCT-I and DST-I) based methods
  - Complexity is further reduced by a factor of four compared to the FFT method

The problem

- Our main target is to reduce the complexity of tap solvers for advanced HSDPA equalizer receivers
- The chip-level signal model for one transmitter and one receiver antenna system can be represented by

$$\mathbf{y} = H\mathbf{s} + \mathbf{n}$$

  - **y** is the received chip sequence (the input to the equalizer)
  - *H* models the transmission channel
  - **s** is the transmitted chip-sequence
  - **n** represents the noise as seen by the receiver
- Output of the chip-level equalizer receiver is a delayed estimate of the transmitted chip:

$$\hat{\mathbf{s}} = \mathbf{x}^H \mathbf{y}$$

- Array **x** represents the filter taps obtained from the equalizer tap solver

The problem

- The LMMSE equalizer solution for **x** is:

$$\mathbf{x} = \sigma_s^2 H^H \left(\sigma_s^2 H H^H + \sigma_n^2 I \right)^{-1} \delta_D = \sigma_s^2 H^H A^{-1} \delta_D$$

  - σ_s^2 and σ_n^2 denote variance of the chip and noise sequence, respectively.
  - δ_D is Kronecker Delta
  - A is a Hermitian Toeplitz matrix

The problem

- Thus, the essential task of the tap solver is to find a column of the inverse of an Hermitian Toeplitz matrix *A*:

$$A = \begin{bmatrix} r_0 & \cdots & r_{m-1} & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ r_1^* & r_0 & \cdots & r_{m-1} & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & & \ddots & \ddots & & \ddots & \ddots & \vdots \\ r_{m-2}^* & \cdots & r_1^* & r_0 & \cdots & r_{m-1} & 0 & \cdots & 0 & 0 \\ r_{m-1}^* & r_{m-2}^* & \cdots & r_1^* & r_0 & \cdots & r_{m-1} & 0 & \ddots & 0 \\ 0 & r_{m-1}^* & r_{m-2}^* & \ddots & r_1^* & r_0 & \ddots & r_{m-1} & \ddots & 0 \\ \vdots & \cdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & 0 & r_{m-1}^* & r_{m-2}^* & \ddots & r_1^* & r_0 & \ddots & r_{m-1} \\ 0 & 0 & \ddots & 0 & r_{m-1}^* & r_{m-2}^* & \ddots & r_1^* & r_0 & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & & & & \\ 0 & \ddots & 0 & 0 & \cdots & 0 & r_{m-1}^* & r_{m-2}^* & \ddots & r_0 \end{bmatrix}$$

- *A* is Hermitian Toeplitz but not circulant;
- Inversion of *A* was computationally simpler if it was circulant;

The problem

❖ Extensive simulations have shown that the performance of tap solvers is only a little lower if the matrix *A* is modified to become circulant :

$$C=\begin{bmatrix}
r_0 & \cdots & r_{m-1} & 0 & \cdots & 0 & r_{m-1}^* & \cdots & r_2^* & r_1^* \\
r_1^* & r_0 & \cdots & r_{m-1} & 0 & \cdots & 0 & r_{m-1}^* & \cdots & r_2^* \\
\vdots & \ddots & \ddots & & \ddots & \ddots & & \ddots & \ddots & \vdots \\
r_{m-2}^* & \cdots & r_1^* & r_0 & \cdots & r_{m-1} & 0 & \cdots & 0 & r_{m-1}^* \\
r_{m-1}^* & r_{m-2}^* & \cdots & r_1^* & r_0 & \cdots & r_{m-1} & 0 & \Box & 0 \\
0 & r_{m-1}^* & r_{m-2}^* & \Box & r_1^* & r_0 & \Box & r_{m-1} & \Box & 0 \\
\vdots & \cdots & \ddots & \ddots & \Box & \ddots & \ddots & \Box & \ddots & \vdots \\
0 & \Box & 0 & r_{m-1}^* & r_{m-2}^* & \Box & r_1^* & r_0 & \Box & r_{m-1} \\
r_{m-1} & 0 & \Box & 0 & r_{m-1}^* & r_{m-2}^* & \Box & r_1^* & r_0 & \Box \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & & & & \\
r_1 & \Box & r_{m-1} & 0 & \cdots & 0 & r_{m-1}^* & r_{m-2}^* & \Box & r_0
\end{bmatrix}$$

- Find inverse of a circulant Hermitian (*LxL*) matrix with as low computational complexity as possible

Example (L=8, m=3)

$$C = \begin{bmatrix} r_0 & r_1 & r_2 & 0 & 0 & 0 & r_2^* & r_1^* \\ r_1^* & r_0 & r_1 & r_2 & 0 & 0 & 0 & r_2^* \\ r_2^* & r_1^* & r_0 & r_1 & r_2 & 0 & 0 & 0 \\ 0 & r_2^* & r_1^* & r_0 & r_1 & r_2 & 0 & 0 \\ 0 & 0 & r_2^* & r_1^* & r_0 & r_1 & r_2 & 0 \\ 0 & 0 & 0 & r_2^* & r_1^* & r_0 & r_1 & r_2 \\ r_2^* & 0 & 0 & 0 & r_2^* & r_1^* & r_0 & r_1 \\ r_1^* & r_2^* & 0 & 0 & 0 & r_2^* & r_1^* & r_0 \end{bmatrix}$$

Conventional method

Every circulant matrix is diagonalized by 2-D Discrete Fourier Transform (DFT):

$$C = FGF^{H}$$

where F is the DFT matrix, F^H is conjugate transpose (inverse) of F,

$$G = diag(\mathbf{g}), \quad \mathbf{g} = F \cdot C(*,0)$$

and $C(*,0)$ is the 0th column of C.

Therefore, the Dth column of C^{-1}, $C^{-1}(*,D)$, is obtained as:

$$C^{-1}(*,D) = C^{-1}\delta_D = (FGF^{H})^{-1}\delta_D = FG^{-1}F^{H}\delta_D = FG^{-1}F^{H}(*,D)$$

where δ_D is Kronecker delta vector consisting of all zeros except one unity at position D.

Conventional method

Algorithm

1. Compute FFT over $C(*,0)$ to find vector $\boldsymbol{g}$
2. Point-wise invert $\boldsymbol{g}$ to obtain diagonal of G^{-1}, $\boldsymbol{g}'$
3. Point-wise multiply $\boldsymbol{g}'$ and $F^H(*,D)$ to obtain $\boldsymbol{g}''=G^{-1}F^H(*,D)$
4. Compute FFT over $\boldsymbol{g}''$ to obtain D^{th} column of the inverse matrix C^{-1}, $C^{-1}(*,D)=F\boldsymbol{g}''$

J. Zhang et al., "Efficient linear equalization for high data rate downlink CDMA signalling," *Conf. Rec. of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers, 2003*. Volume 1, 9-12 Nov. 2003, pp. 141 – 145.

Conventional method

Example (L=8, m=3, D=4)

$$C=\begin{bmatrix} 1 & 2+2i & 3+i & 0 & 0 & 0 & 3-i & 2-2i \\ 2-2i & 1 & 2+2i & 3+i & 0 & 0 & 0 & r_2^* \\ 3-i & 2-2i & 1 & 2+2i & 3+i & 0 & 0 & 0 \\ 0 & 3-i & 2-2i & 1 & 2+2i & 3+i & 0 & 0 \\ 0 & 0 & 3-i & 2-2i & 1 & 2+2i & 3+i & 0 \\ 0 & 0 & 0 & 3-i & 2-2i & 1 & 2+2i & 3+i \\ 3+i & 0 & 0 & 0 & 3-i & 2-2i & 1 & 2+2i \\ 2+2i & 3+i & 0 & 0 & 0 & 3-i & 2-2i & 1 \end{bmatrix}$$

$C^{-1}(*,4)$

$$C^{-1}\cong\begin{bmatrix} -0.37 & 0.01-0.07i & 0.19+0.22i & -0.07+0.15i & 0.2 & -0.07-0.15 & 0.19-0.22i & 0.01+0.07i \\ 0.01+0.07i & -0.37 & 0.01-0.07i & 0.19+0.22i & -0.07+0.15i & 0.2 & -0.07-0.15 & 0.19-0.22i \\ 0.19-0.22i & 0.01+0.07i & -0.37 & 0.01-0.07i & 0.19+0.22i & -0.07+0.15i & 0.2 & -0.07-0.15 \\ -0.07-0.15 & 0.19-0.22i & 0.01+0.07i & -0.37 & 0.01-0.07i & 0.19+0.22i & -0.07+0.15i & 0.2 \\ 0.2 & -0.07-0.15 & 0.19-0.22i & 0.01+0.07i & -0.37 & 0.01-0.07i & 0.19+0.22i & -0.07+0.15i \\ -0.07+0.15i & 0.2 & -0.07-0.15 & 0.19-0.22i & 0.01+0.07i & -0.37 & 0.01-0.07i & 0.19+0.22i \\ 0.19+0.22i & -0.07+0.15i & 0.2 & -0.07-0.15 & 0.19-0.22i & 0.01+0.07i & -0.37 & 0.01-0.07i \\ 0.01-0.07i & 0.19+0.22i & -0.07+0.15i & 0.2 & -0.07-0.15 & 0.19-0.22i & 0.01+0.07i & -0.37 \end{bmatrix}$$

Conventional method

Example (L=8, m=3, D=4)

$$C = \begin{bmatrix} 1 & 2+2i & 3+i & 0 & 0 & 0 & 3-i & 2-2i \\ 2-2i & 1 & 2+2i & 3+i & 0 & 0 & 0 & r_2^* \\ 3-i & 2-2i & 1 & 2+2i & 3+i & 0 & 0 & 0 \\ 0 & 3-i & 2-2i & 1 & 2+2i & 3+i & 0 & 0 \\ 0 & 0 & 3-i & 2-2i & 1 & 2+2i & 3+i & 0 \\ 0 & 0 & 0 & 3-i & 2-2i & 1 & 2+2i & 3+i \\ 3+i & 0 & 0 & 0\ {}^{r_0} & 3-i & 2-2i & 1 & 2+2i \\ 2+2i & 3+i & 0 & 0 & 0 & 3-i & 2-2i & 1 \end{bmatrix}$$

$$C(*,0) = \begin{bmatrix} 1 \\ 2-2i \\ 3-i \\ 0 \\ 0 \\ 0 \\ 3+i \\ 2+2i \end{bmatrix}$$

Step 1

FFT

$$\begin{bmatrix} 11 \\ -1 \\ -9 \\ -2.66 \\ 3 \\ -1 \\ -1 \\ 8.66 \end{bmatrix} = \mathbf{g}$$

Step 2

Point-wise invert

$$\begin{bmatrix} 0.09 \\ -1 \\ -0.11 \\ -0.38 \\ 0.33 \\ -1 \\ -1 \\ 0.12 \end{bmatrix} = 1./\mathbf{g}$$

Step 3

Point-wise multiply to $F^H(*,D)$

TAMPERE UNIVERSITY OF TECHNOLOGY

Conventional method

Step 3

Point-wise multiply to $F^H(*,D)$

$$(1./\mathbf{g}).*F(*,4)=\begin{bmatrix}0.09\\-1\\-0.11\\-0.38\\0.33\\-1\\-1\\0.12\end{bmatrix}.*\begin{bmatrix}0.35\\-0.35\\0.35\\-0.35\\0.35\\-0.35\\0.35\\-0.35\end{bmatrix}=\begin{bmatrix}0.09\\1\\-0.11\\0.38\\0.33\\1\\-1\\-0.12\end{bmatrix}$$

Step 4

FFT

$$\begin{bmatrix}0.2\\-0.07+0.15i\\-0.2+0.22i\\0.01-0.07\\-0.37\\0.01+0.07\\-0.2-0.22i\\-0.07-0.15i\end{bmatrix}=C^{-1}(*,4)$$

=

$$C^{-1}\cong\begin{bmatrix}-0.37 & 0.01-0.07i & 0.19+0.22i & -0.07+0.15i & 0.2 & -0.07-0.15 & 0.19-0.22i & 0.01+0.07i\\0.01+0.07i & -0.37 & 0.01-0.07i & 0.19+0.22i & -0.07+0.15i & 0.2 & -0.07-0.15 & 0.19-0.22i\\0.19-0.22i & 0.01+0.07i & -0.37 & 0.01-0.07i & 0.19+0.22i & -0.07+0.15i & 0.2 & -0.07-0.15\\-0.07-0.15 & 0.19-0.22i & 0.01+0.07i & -0.37 & 0.01-0.07i & 0.19+0.22i & -0.07+0.15i & 0.2\\0.2 & -0.07-0.15 & 0.19-0.22i & 0.01+0.07i & -0.37 & 0.01-0.07i & 0.19+0.22i & -0.07+0.15i\\-0.07+0.15i & 0.2 & -0.07-0.15 & 0.19-0.22i & 0.01+0.07i & -0.37 & 0.01-0.07i & 0.19+0.22i\\0.19+0.22i & -0.07+0.15i & 0.2 & -0.07-0.15 & 0.19-0.22i & 0.01+0.07i & -0.37 & 0.01-0.07i\\0.01-0.07i & 0.19+0.22i & -0.07+0.15i & 0.2 & -0.07-0.15 & 0.19-0.22i & 0.01+0.07i & -0.37\end{bmatrix}$$

Proposed method

- ✓ Since C is circulant Hermitian, C^{-1} is also circulant Hermitian:

$$C^{-1} = \begin{bmatrix}
q_0 & \cdots & q_{m-1} & 0 & \cdots & 0 & q_{m-1}^* & \cdots & q_1^* & q_1^* \\
q_1^* & q_0 & \cdots & q_{m-1} & 0 & \cdots & 0 & q_{m-1}^* & \cdots & q_2^* \\
\vdots & \ddots & \ddots & & \ddots & \ddots & & \ddots & \ddots & \vdots \\
q_{m-2}^* & \cdots & q_1^* & q_0 & \cdots & q_{m-1} & 0 & \cdots & 0 & q_{m-1}^* \\
q_{m-1}^* & q_{m-2}^* & \cdots & q_1^* & q_0 & \cdots & q_{m-1} & 0 & \cdots & 0 \\
0 & q_{m-1}^* & q_{m-2}^* & \cdots & q_1^* & q_0 & \cdots & q_{m-1} & \cdots & 0 \\
\vdots & \cdots & \ddots & \ddots & \cdots & \ddots & \ddots & \cdots & \ddots & \vdots \\
0 & \cdots & 0 & q_{m-1}^* & q_{m-2}^* & \cdots & q_1^* & q_0 & \cdots & q_{m-1} \\
q_{m-1} & 0 & \cdots & 0 & q_{m-1}^* & q_{m-2}^* & \cdots & q_1^* & q_0 & \cdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & & & & \\
q_1 & \cdots & q_{m-1} & 0 & \cdots & 0 & q_{m-1}^* & q_{m-2}^* & \cdots & q_0
\end{bmatrix}$$

- ✓ To find C^{-1} it is sufficient to find first $L/2+1$ entries of its 0^{th} column $C^{-1}(0{:}L/2,0)$.
- ✓ Other entries of 0^{th} column are obtained as conjugates.
- ✓ The D^{th} column column is obtained by D circular shifts.

Proposed method

The proposed method is based on two transforms:

1. Modified Discrete Cosine Transform of Type 1 (DCT-1) of order *L*/2+1:

$$C_1^{L/2+1} = [c_{n,m}], \quad c_{n,m} = b_m \cos(\pi nm / (L/2)), \quad n,m = 0,\ldots,L/2$$

$$b_m = \begin{cases} 1/2, & \text{for } \mathrm{m} = 0, L/2 \\ 1, & \text{otherwise} \end{cases}$$

2. Discrete Sine Transform of Type 1 (DST-1) of order *L*/2-1:

$$S_1^{L/2-1} = [s_{n,m}], \quad s_{n,m} = \sin(\pi nm / (L/2)), \quad n,m = 1,\ldots,L/2-1$$

For both transforms fast algorithms exist

Proposed method

Lemma. For an arbitrary LxL circulant Hermitian matrix C, the real and imaginary parts of the sub vector $C^{-1}(0{:}L/2,0)$ of the 0^{th} column of C^{-1} can be expressed as follows

$$\mathrm{Re}(\mathbf{c}) = C_1^{L/2+1} Q\mathbf{p}\ , \qquad \mathrm{Im}(\mathbf{c}) = -\left[0, S_1^{L/2-1} Q_1 \mathbf{s}, 0\right]$$

where

$$\mathbf{p} = \left[p_0, p_1, ..., p_{L/2}\right]^T = C_1^{L/2+1}\left(\mathrm{Re}\left(C(0 : L/2, 0)\right)\right)$$

$$\mathbf{s} = \left[s_1, s_2, ..., s_{L/2-1}\right]^T = S_1^{L/2-1}(\mathrm{Im}(C(1 : L/2-1, 0)))$$

$C(0{:}L\text{-}1,0)$ is the 0^{th} column of C, Q=diag(q_0,…, $q_{L/2}$),

Q_1=diag(q_1,…,$q_{L/2-1}$) such that:

$$q_0 = 1/p_0, \quad q_i = 1/(p_i^2 - s_i^2), \ i = 1,...,L/2-1, \ q_{L/2} = 1/p_{L/2}$$

Proposed method

Algorithm.

1.Apply fast DCT-1 and DST-1 to the vector $C(0{:}L/2,0)$ [first $L/2+1$ components of the 0^{th} column of C] to find

$$\mathbf{p} = C_1^{L/2+1}\left(\mathrm{Re}\left(C(0:L/2,0)\right)\right) \text{ and } \mathbf{s} = S_1^{L/2-1}(\mathrm{Im}(C(1:L/2-1,0)))$$

2.Find vector

$$\mathbf{q} = \left[q_0, q_1, \ldots, q_{L/2}\right]$$

$$q_0 = 1/p_0\ , \quad q_i = 1/(p_i^2 - s_i^2)\ , \ i = 1,\ldots,L/2-1\ , \ q_{L/2} = 1/p_{L/2}$$

2.Point-wise multiply **p** and **s** to **q** to obtain **p**’=**p**.***q** and **s**’=**s**.***q**(1:L/2-1).

3.Apply fast DCT-1 to **p**’ and DST-1 to **s**’ and find the real and imaginary parts of the vector $C^{-1}(0{:}L/2,0)$.

Proposed method

Example (L=8, m=3)

$$
C = \begin{bmatrix}
1 & 2+2i & 3+i & 0 & 0 & 0 & 3-i & 2-2i \\
2-2i & 1 & 2+2i & 3+i & 0 & 0 & 0 & r_2^* \\
3-i & 2-2i & 1 & 2+2i & 3+i & 0 & 0 & 0 \\
0 & 3-i & 2-2i & 1 & 2+2i & 3+i & 0 & 0 \\
0 & 0 & 3-i & 2-2i & 1 & 2+2i & 3+i & 0 \\
0 & 0 & 0 & 3-i & 2-2i & 1 & 2+2i & 3+i \\
3+i & 0 & 0 & 0 & 3-i & 2-2i & 1 & 2+2i \\
2+2i & 3+i & 0 & 0 & 0 & 3-i & 2-2i & 1
\end{bmatrix}
$$

C is circulant Hermitian → C^{-1} is circulant Hermitian.

To find C^{-1} it is enough to find only first L/2+1 entries of 0^{th} column of C^{-1}.

Other entries of 0^{th} column are obtained as conjugates.

The D^{th} column columns are obtained by D circular shifts

$C^{-1}(0{:}4,0)$

$$
C^{-1} \cong \begin{bmatrix}
-0.37 & 0.01-0.07i & 0.19+0.22i & -0.07+0.15i & 0.2 & -0.07-0.15 & 0.19-0.22i & 0.01+0.07i \\
0.01+0.07i & -0.37 & 0.01-0.07i & 0.19+0.22i & -0.07+0.15i & 0.2 & -0.07-0.15 & 0.19-0.22i \\
0.19-0.22i & 0.01+0.07i & -0.37 & 0.01-0.07i & 0.19+0.22i & -0.07+0.15i & 0.2 & -0.07-0.15 \\
-0.07-0.15 & 0.19-0.22i & 0.01+0.07i & -0.37 & 0.01-0.07i & 0.19+0.22i & -0.07+0.15i & 0.2 \\
0.2 & -0.07-0.15 & 0.19-0.22i & 0.01+0.07i & -0.37 & 0.01-0.07i & 0.19+0.22i & -0.07+0.15i \\
-0.07+0.15i & 0.2 & -0.07-0.15 & 0.19-0.22i & 0.01+0.07i & -0.37 & 0.01-0.07i & 0.19+0.22i \\
0.19+0.22i & -0.07+0.15i & 0.2 & -0.07-0.15 & 0.19-0.22i & 0.01+0.07i & -0.37 & 0.01-0.07i \\
0.01-0.07i & 0.19+0.22i & -0.07+0.15i & 0.2 & -0.07-0.15 & 0.19-0.22i & 0.01+0.07i & -0.37
\end{bmatrix}
$$

Proposed method

Example (L=8, m=3)

Step 1

$$\mathrm{Re}(C(0:4,0)) = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 0 \\ 0 \end{bmatrix} \xrightarrow{\text{mDCT-I}} \begin{bmatrix} 5.5 \\ 1.9 \\ -2.5 \\ -0.9 \\ 1.5 \end{bmatrix} = \mathbf{p}$$

$$C(1:L/2-1,0) = \begin{bmatrix} -2 \\ -1 \\ 0 \end{bmatrix} \xrightarrow{\text{DST-I}} \begin{bmatrix} -2.4 \\ -2 \\ -0.4 \end{bmatrix} = \mathbf{s}$$

Step 2

$$\xrightarrow{\text{Find } \mathbf{q}} \begin{bmatrix} 0.18 \\ -2.16 \\ 2.25 \\ 0.66 \\ 0.67 \end{bmatrix} = \mathbf{q}$$

Step 3

Point-wise multiply **p** and **s** to **q**

Proposed method

Example (L=8, m=3)

Step 3

Point-wise multiply **p** and **s** to **q**

$$\mathbf{p}.*\mathbf{q}=\begin{bmatrix}5.5\\1.9\\-2.5\\-0.9\\1.5\end{bmatrix}.*\begin{bmatrix}0.18\\-2.16\\2.25\\0.66\\0.67\end{bmatrix}=\begin{bmatrix}0.18\\-0.88\\-1.11\\-1.38\\0.67\end{bmatrix}=\mathbf{p}'$$

Step 4

mDCT-I

$$\begin{bmatrix}-0.37\\0.01\\0.19\\-0.07\\0.2\end{bmatrix}=\mathrm{Re}(C^{-1}(0:4,0)$$

$$\mathbf{s}.*\mathbf{q}(1:3)=\begin{bmatrix}-2.4\\-2\\-0.4\end{bmatrix}.*\begin{bmatrix}-2.16\\2.25\\0.66\end{bmatrix}=\begin{bmatrix}-1.15\\0.89\\0.62\end{bmatrix}=\mathbf{s}'$$

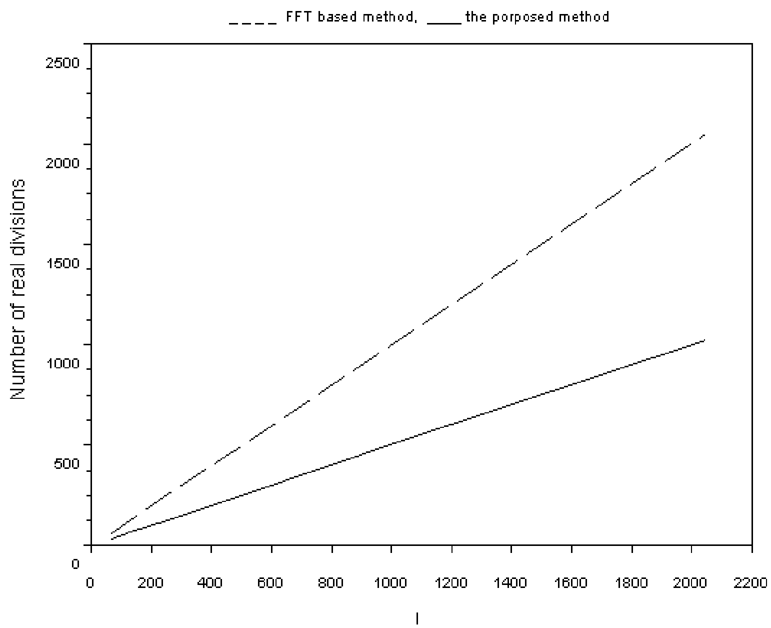
DST-I

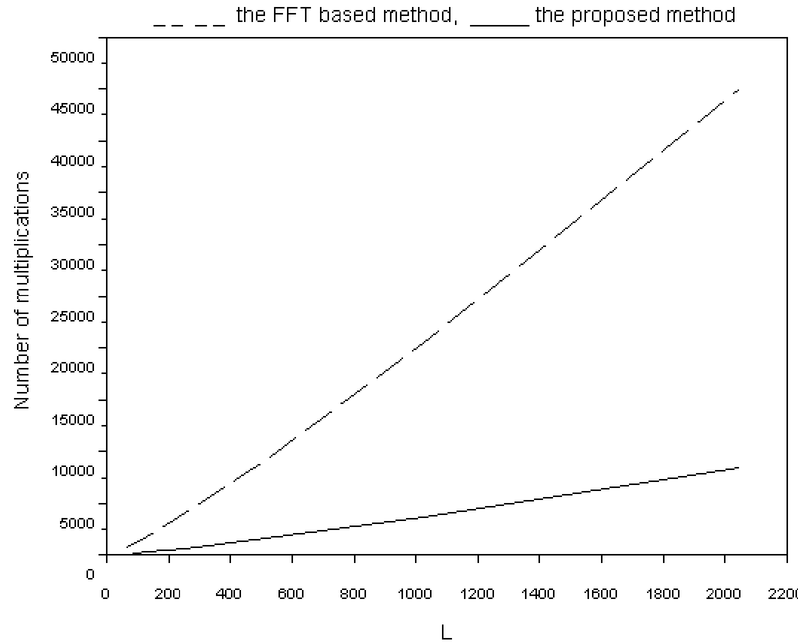
$$\begin{bmatrix}0.07\\-0.22\\-0.15\end{bmatrix}=\mathrm{Im}(C^{-1}(1:3,0))$$

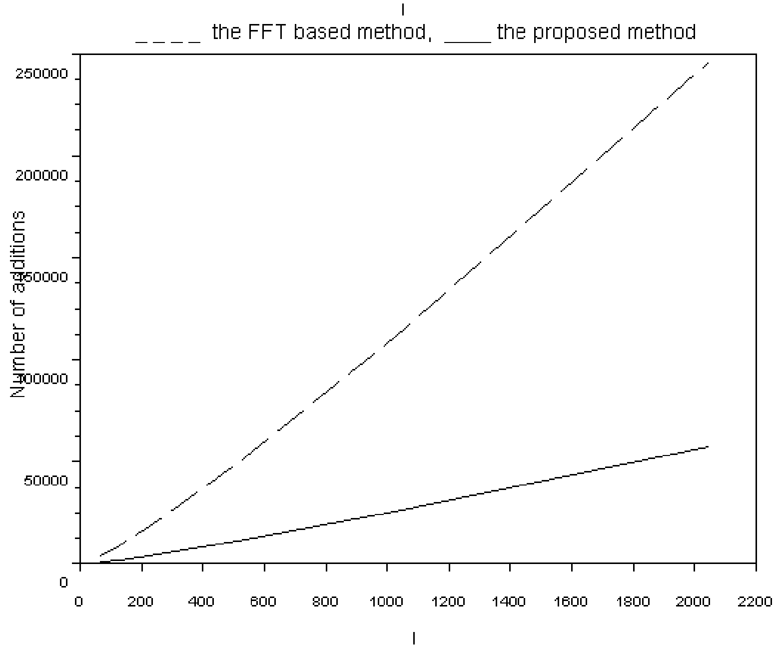
$C^{-1}(0:4,0)$

$$C^{-1}\cong\begin{bmatrix}
-0.37 & 0.01-0.07i & 0.19+0.22i & -0.07+0.15i & 0.2 & -0.07-0.15 & 0.19-0.22i & 0.01+0.07i\\
0.01+0.07i & -0.37 & 0.01-0.07i & 0.19+0.22i & -0.07+0.15i & 0.2 & -0.07-0.15 & 0.19-0.22i\\
0.19-0.22i & 0.01+0.07i & -0.37 & 0.01-0.07i & 0.19+0.22i & -0.07+0.15i & 0.2 & -0.07-0.15\\
-0.07-0.15 & 0.19-0.22i & 0.01+0.07i & -0.37 & 0.01-0.07i & 0.19+0.22i & -0.07+0.15i & 0.2\\
0.2 & -0.07-0.15 & 0.19-0.22i & 0.01+0.07i & -0.37 & 0.01-0.07i & 0.19+0.22i & -0.07+0.15i\\
-0.07+0.15i & 0.2 & -0.07-0.15 & 0.19-0.22i & 0.01+0.07i & -0.37 & 0.01-0.07i & 0.19+0.22i\\
0.19+0.22i & -0.07+0.15i & 0.2 & -0.07-0.15 & 0.19-0.22i & 0.01+0.07i & -0.37 & 0.01-0.07i\\
0.01-0.07i & 0.19+0.22i & -0.07+0.15i & 0.2 & -0.07-0.15 & 0.19-0.22i & 0.01+0.07i & -0.37
\end{bmatrix}$$

Complexity comparison







Approximately four times less arithmetic operations compared to the conventional method

Conclusion

- ✓ New fast DCT-1 and fast DST-1 based algorithm for inversion of circulant Hermitian matrices was proposed.
- ✓ This algorithm is approximately four times less costly compared to the conventional FFT based method.
- ✓ The proposed algorithm may reduce complexity of tap solvers for advanced HSDPA equalizer receivers