New combined array information UD algorithm of the Kalman filter based channel estimation for OFDM data transmission

I V Semushin[1], **Yu V Tsyganova**[1,2], **V V Ugarov**[1] **and A V Tsyganov**[2]

[1] Department of Mathematics, Information and Aviation Technology, Ulyanovsk State University, Ulyanovsk, Russian Federation
[2] Department of Mathematics, Physics and Technology Education, Ulyanovsk State Pedagogical University named after I.N. Ulyanov, Ulyanovsk, Russian Federation

Abstract. The paper develops a new channel estimation algorithm for use in broadband OFDM data transmission over non-ideal channels. The channel is described by Gauss–Markov AR model of a given order in state-space form. One of the existing solutions to channel estimation is based on the well-known Kalman filter (KF). Another approach is to use the information formulation of KF, the so-called Information filter (IF). To improve the numerical properties of the IF implementation, we propose a new numerically efficient channel estimation algorithm, the so-called combined array UD Information Filter (caUD-IF). The algebraic equivalence between IF and new caUD-IF is proved. The aspects of a parallel implementation of the suggested algorithm are also considered.

1. Introduction

Orthogonal Frequency Division Multiplexing (OFDM) is a combination of modulation and multiplexing. In modulations, information is mapped on to changes in frequency, phase or amplitude (or a combination of them) of a carrier signal. Multiplexing deals with allocation/accommodation of users in a given bandwidth. In other words, it deals with the allocation of an available resource.

Following [1], an OFDM signal can be represented as

$$OFDM\ signal = \sum_{n=0}^{N-1} s_n\,(t)\sin\,(2\pi f_n t) \qquad (1)$$

where $s(t)$ represents the symbols mapped to the chosen constellation and f_n represents the orthogonal frequencies. The OFDM architecture is illustrated in figure 1.

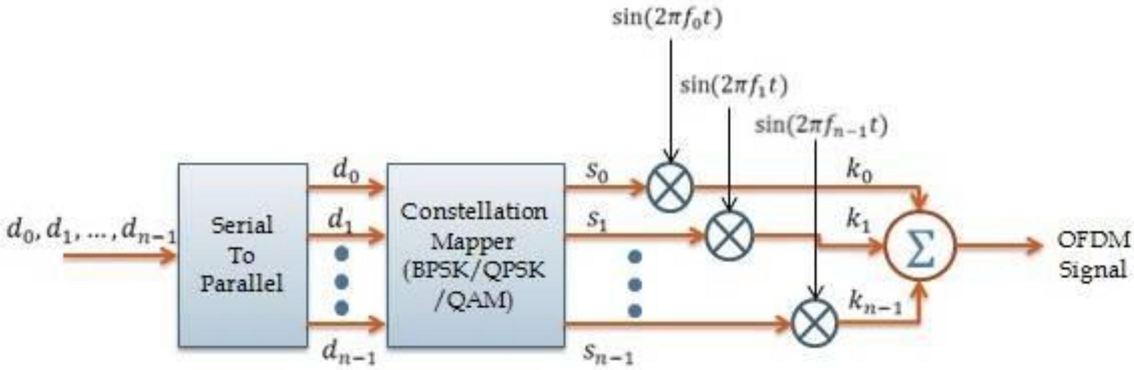

Figure 1. The scheme from page 233 of Viswanathan, M. *Simulation of Digital Communication Systems Using Matlab* [eBook], 2013.

The data bits $D = [d_0, d_1, \dots, d_{n-1}]$ are first converted from serial stream to parallel stream depending on the number of sub-carriers (n) named from 0 to $n-1$. The Serial to Parallel converter takes the serial stream of input bits and outputs n parallel streams. They are individually converted into the required digital modulation format (BPSK, QPSK, QAM, etc.).

Let us call this output $S = [s_0, s_1, \dots, s_{n-1}]$. The conversion of parallel data D into the digitally modulated data S is usually achieved by a constellation mapper, which is essentially a look-up table (LUT). Once the data bits are converted to required modulation format, they are superimposed on the required orthogonal subcarriers for transmission through the channel. This is achieved by a series of n parallel sinusoidal oscillators tuned to n orthogonal frequencies $f_0, f_1, \dots, f_{n-1}$. The resultant output from the n parallel arms is summed up together to produce the OFDM signal. Figure 2 shows the entire architecture of a basic OFDM system with both transmitter and receiver.

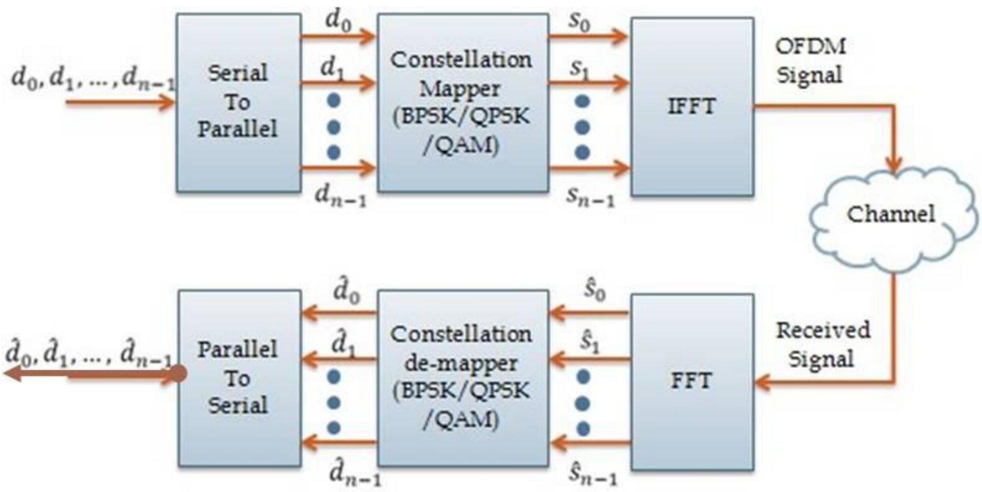

Figure 2. The scheme from page 233 of Viswanathan, M. *Simulation of Digital Communication Systems Using Matlab* [eBook], 2013.

The FFT/IFFT (Fast Fourier Transform / Inverse Fast Fourier Transform) length n defines the number of total subcarriers present in the OFDM system. For example, an OFDM system with $n = 64$ provides 64 subcarriers. In reality, not all the subcarriers are utilized for data transmission. Some subcarriers are reserved for pilot carriers (used for channel estimation/equalization and to combat magnitude and phase errors in the receiver) and some are left unused to act as a guard band.

In the simplest case, the channel is modeled as a simple AWGN (additive white Gaussian noise) channel. In a more realistic case, the channel is modeled as a first rank Markov process. In our research work, we model the channel by the Gauss–Markov AR model of rank n_k written in the state space for each k-th sub-channel allocated for the k-th user.

2. Channel model for OFDM data transmission

Consider the Gauss–Markov AR model written in the state space

$$
\begin{aligned}
x_{t+1} &= \begin{bmatrix} \Phi_t^{(1)} & 0 & \cdots & 0 \\ 0 & \Phi_t^{(2)} & \cdots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & \cdots & 0 & \Phi_t^{(m)} \end{bmatrix} x_t + \begin{bmatrix} \Gamma^{(1)} & 0 & \cdots & 0 \\ 0 & \Gamma^{(2)} & \cdots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & \cdots & 0 & \Gamma^{(m)} \end{bmatrix} w_t, \quad t = 0,1,\dots; \\
z_t &= \begin{bmatrix} \mathrm{H}_t^{(1)} & 0 & \cdots & 0 \\ 0 & \mathrm{H}_t^{(2)} & \cdots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & \cdots & 0 & \mathrm{H}_t^{(m)} \end{bmatrix} x_t + v_t, \quad t = 1,2,\dots
\end{aligned}
\tag{2}
$$

where combined state vector $x_t \triangleq \left[x_t^{(1)T} | \dots | x_t^{(m)T}\right]^T$ consists of m sub-vectors $x_t^{(k)T} \triangleq \left[x_{t-n_k+1}^{(k)} | \dots | x_{t-1}^{(k)} | x_t^{(k)}\right]^T$, $k = 1,\dots,m$, corresponding to m different sub-channels; z_t is the measurement vector; the process noise w_t and the measurement noise v_t are mutually independent random (Gaussian) sequences, i.e., $w_t \sim \mathcal{N}(0, Q_t)$ and $v_t \sim \mathcal{N}(0, R_t)$. Let these noises be independent of some random initial state $x_0 \sim \mathcal{N}(\bar{x}_0, P_0)$. Covariance matrices Q_t and R_t have the following block (array) forms: $Q_t = \operatorname{diag}\left[q_t^{(1)}, q_t^{(2)}, \dots, q_t^{(m)}\right]$ and $R_t = \operatorname{diag}\left[r_t^{(1)}, r_t^{(2)}, \dots, r_t^{(m)}\right]$. The system matrices in (2) are determined as follows:

$$
\left.
\begin{aligned}
&\Phi_t^{(k)} = \begin{bmatrix} 0 & I \\ a_{n_{k-1,t}}^{(k)} & \dots a_{n_{1,t}}^{(k)} a_{n_{0,t}}^{(k)} \end{bmatrix} \\
&\Gamma^{(k)} = [0,\dots,0,1]^T,\ n_k - \text{dimensional} \\
&H_t^{(k)} = \left[0,\dots,0,s_t^{(k)}\right]^T,\ n_k - \text{dimensional}
\end{aligned}
\right\}
\begin{aligned}
&k = 1,\dots,m \\
&m = \text{the number of sub} - \text{channels} \\
&n_k = 1,\dots,N \\
&n_k = \text{the } k^{\text{th}} \text{ sub} - \text{channel dimension} \\
&N = \text{the greatest possible } n_k
\end{aligned}
\tag{3}
$$

The following nomenclature (see table 1) gives an insight into variables and parameters used in the channel model:

Table 1. Model nomenclature.

Name	Meaning	Range	Dimension
x_t	channel (state of)	$\mathbb{C}^n$	$n = \sum_{k=1}^{m} n_k$
Φ_t	channel transition	$\mathbb{C}^{n \times n}$	$n \times n$
Γ	channel noise input matrix	$\mathbb{R}^{n \times m}$	$n \times m$
w_t	channel noise	$\mathbb{C}^m$	m
z_t	observed signal	$\mathbb{C}^m$	m
H_t	channel pilot subcarriers	$\mathbb{C}^{m \times n}$	$m \times n$
v_t	channel observation noise	$\mathbb{C}^m$	m
Q_t	channel noise covariance	$\mathbb{C}^{m \times m}$	$m \times m$
R_t	channel noise covariance	$\mathbb{C}^{m \times m}$	$m \times m$

Our goal is to estimate unknown state vector x_t, i.e., to calculate, at each discrete time instant t, the one-step predicted estimate $\hat{x}_{t|t-1}$ minimizing the MSE criterion $\mathbb{E}\left[(x_t - \hat{x}_{t|t-1})^H (x_t - \hat{x}_{t|t-1})\right]$, given the available measurements $Z_1^{t-1} = [z_1^T | \dots | z_{t-1}^T]^T$. The well-known Kalman filter algorithm [2, 3] is an ideal theoretical tool for solving the linear estimation problem. At each discrete-time moment KF yields to compute the linear least-square predicted estimate $\hat{x}_{t|t-1}$, of the state vector x_t and the predicted error covariance matrix $P_{t|t-1} = \mathbb{E}\left[(x_t - \hat{x}_{t|t-1})(x_t - \hat{x}_{t|t-1})^H\right]$, and also the filtered estimate $\hat{x}_{t|t}$ given the measurements Z_1^t and the filtered error covariance matrix $P_{t|t} = \mathbb{E}\left[(x_t - \hat{x}_{t|t})(x_t - \hat{x}_{t|t})^H\right]$.

There is no doubt that the main computational effort in Kalman filtering is spent in solving the Riccati equation. This effort is needed to calculate the Kalman gains. However, only the value of the predicted error covariance matrix $P_{t|t-1}$ is required for this purpose. A value of the filtered error covariance matrix $P_{t|t}$ is only used as an intermediate result on the way to compute the next value of $P_{t+1|t}$.

Actually, it is not necessary to compute matrix $P_{t|t}$ explicitly. It is possible to compute the predicted error covariance matrix $P_{t|t-1}$ from one temporal epoch to the next, without going through the intermediate values of the filtered error covariance. This approach, and the algorithms for doing it, are called the *combined* measurement/time update or *one-stage* filters [3, 4]. The equations of the one-stage conventional KF can be found in [3, p. 317]. Thus, a combined (one-stage) KF formulation requires less computation than a two-stage one.

3. Channel Estimation Algorithm Based on Information Filter

The well-known *Information filter* (IF) is an alternative formulation of the Kalman filter, where the covariance matrix P is replaced by its inverse matrix Y, which is called the *information matrix*. The information formulation is particularly useful when there is no prior information, i.e. the initial covariance matrix $P_0 = \infty$. In this case, the covariance formulation of the KF is not defined, while the information formulation is, and can start from $Y_0 = P_0^{-1} = 0$.

Information filter does not use the same state vector representation as the conventional KF. Those that use the information matrix in the filter implementation use the *information state* $d \triangleq Yx$. The implementation equations for the "straight" information filter (i.e., using Y) one can find in [2, p. 263]. This algorithm has the two-stage formulation, i.e. it consists of the time update step and the measurement update step. It is easy to reformulate it in a combined (one-stage) form.

Here we suggest using the information filter for solving the channel estimation problem. Let us formulate the combined (one-stage) information algorithm (cIF) for OFDM channel estimation with the above mathematical model (2), (3). Taking into account the block diagonal structure of system matrices, we design the estimation algorithm in the form of m concurrent cIFs as it is described in the below.

For $k = 1, \dots, m$ we implement the combined Information Filter $c\mathcal{IF}_k$ to estimate the unobservable state vector $x_t^{(k)}$ from the noise-corrupted measurements $\left(Z^{(k)}\right)_1^{t-1}$ as it is shown in table 2.

Table 2. Summary of combined Information Filter (cIF).

Initial data	
Information matrix:	$Y_0 = P_0^{-1}$
Information estimate:	$d_0 = Y_0 \bar{x}_0$
Recursively update ($t = 0,1, \dots$)	
Information matrix:	$Y_{t+1}^{(k)} = A_t^{(k)} - L_t^{(k)} C_t^{(k)} \left(L_t^{(k)}\right)^H$ (4) where $A_t^{(k)} = \left(\Phi_t^{(k)}\right)^{-H} \left[Y_t^{(k)} + \left(H_t^{(k)}\right)^H \left(R_t^{(k)}\right)^{-1} H_t^{(k)}\right] \left(\Phi_t^{(k)}\right)^{-1}$, (5) $C_t^{(k)} = \left(\Gamma^{(k)}\right)^T A_t^{(k)} \Gamma^{(k)} + \left(Q_t^{(k)}\right)^{-1}$, (6) $L_t^{(k)} = A_t^{(k)} \Gamma^{(k)} \left(C_t^{(k)}\right)^{-1}$ (7)
Information estimate:	$\hat{d}_{t+1}^{(k)} = \left[I - L_t^{(k)} \left(\Gamma^{(k)}\right)^T\right] \left(\Phi_t^{(k)}\right)^{-H} \left(\hat{d}_t^{(k)} + \left(H_t^{(k)}\right)^H \left(R_t^{(k)}\right)^{-1} z_t^{(k)}\right)$ (8)

Remark 1. At any discrete-time moment t, one can easily obtain the predicted state estimate $\hat{x}_{t|t-1} = Y_t^{-1} \hat{d}_t$, where $Y_t = \mathrm{diag}\left[Y_t^{(1)}, Y_t^{(2)}, \dots, Y_t^{(m)}\right]$ and $\hat{d}_t = \left[\left(\hat{d}_t^{(1)}\right)^T, \left(\hat{d}_t^{(1)}\right)^T, \dots, \left(\hat{d}_t^{(1)}\right)^T\right]^T$.

Remark 2. It is obvious that the bank of cIFs $\{c\mathcal{IF}_k | k = 1, \dots, m\}$ is ideal for the organization of a parallel computing scheme that can significantly accelerate the process of channel estimation.

Recently, different solutions of various channel estimation problems with the use of the Kalman filtering technique were considered in [5–8]. However, it is well-known that the conventional KF algorithm is numerically unstable due to the Riccati computational procedure (see, for instance, the discussion in [9]), and the same is true for the information filter (4)–(8). The problem of machine round-off errors is unavoidable due to the limited machine precision of real floating-point numbers. Unfortunately, it is impossible to solve the problem completely. However, one can significantly reduce the effect of machine round-off errors by designing some *algebraically equivalent* Kalman filter implementations, which may become the desired numerically efficient algorithms. Such solutions are based on a variety of matrix factorization methods as applied to the error covariance matrices involved in the covariance filter equations, and also to the information matrices involved in the information filter equations.

Since the invention of the KF in the 1960s, there has been a great interest in the development of numerically stable and efficient KF implementation methods [2]. In this paper, we study KF implementation algorithms of information type and propose the novel *numerically efficient combined array UD information filter* (caUD-IF) for solving the channel estimation problem. This new estimation algorithm was constructed for the first time. Although the UD-based information-type KF implementation was recently proposed in [10], it does not have a convenient array form for processing homogeneous large-scale data.

4. Numerically stable channel estimation algorithm based on the new combined array UD Information Filter

The main idea of the KF array formulation is that the required quantities of the discrete filter are updated in an array form using the orthogonal matrix transforms. It means that numerically stable orthogonal transforms are used for updating the corresponding factors of the state error covariance matrix (and also state estimate) at each iteration step. In other words, orthogonal operators are applied to the *pre-array* (which contains the filter quantities available at the current step) to get the *post-array* in some special form. Then, the required updated filter quantities are simply read out from the post-array. This feature makes the array algorithms better suited to the parallel computations and to the very large scale integration (VLSI) implementations [11, 12]. The first information-type array filter was built in [13]. The covariance-type array square-root algorithms were constructed in [12].

Another important class of array algorithms is the UD KF methods. The special feature of it is that these algorithms are based on the modified Cholesky factorization of the state prediction error covariance matrix $= UDU^H$, where U is a unit upper triangular matrix, $D > 0$ is a diagonal matrix. The first UD filter was developed by G.J. Bierman; see [14] for more details. J.M. Jover and T. Kailath proposed in [15] the array form of UD based measurement update algorithm. Recently, a new extended array UD covariance filter (eUD-CF) was proposed in [16] and then apply to the channel estimation problem [17].

The modified Cholesky decomposition implies the factorization of a symmetric positive definite complex matrix A in the form $A = U_A D_A U_A^H$, where D_A denotes a diagonal matrix and U_A is an upper triangular matrix with 1's on the main diagonal. We need to note that the Cholesky decomposition (and its modified version) exists and is unique when the matrix to be decomposed is positive definite; see [18]. If the matrix is a positive semi-definite, then the Cholesky decomposition still exists, however, it is not unique.

Concerning the Information formulation of the KF, the matrix factorization must perform for the initial information matrix Y_0 and the covariance matrices Q_t, R_t. For the examined discrete-time stochastic model (2), (3) with the associated Information filter (4)–(9) we assume that $Y_0 > 0$, $Q_t > 0$, $R_t > 0$ and matrix Φ_t is invertible. Hence, the modified Cholesky decomposition exists and is unique for the mentioned quantities, i.e., we have $Y_0 = U_{Y_0} D_{Y_0} U_{Y_0}^H$; $Q_t = U_{Q_t} D_{Q_t} U_{Q_t}^H$ and $R_t = U_{R_t} D_{R_t} U_{R_t}^H$, $t = 0,1, \dots$.

For the UD-based KF implementations, the modified weighted Gram-Schmidt (MWGS) orthogonalization is used for the recursive update of the error covariance UD-factors. It was shown that the MWGS outperforms the usual Gram-Schmidt orthogonalization for accuracy; see [19].

In this paper, we construct the UD-based formulations of the newly developed combined array Information Filter (caUD-IF) by using MWGS orthogonalization. More precisely, each iteration of the new caUD-IF algorithm should have the following form: given a pair of the pre-arrays $\{\mathcal{A}, \mathcal{D}_{\mathcal{A}}\}$, compute a pair of the post-arrays $\{\mathcal{U}, \mathcal{D}_{\mathcal{U}}\}$ by means of the MWGS orthogonalization, $\mathfrak{W}$, i.e.,

$$\mathcal{A}^H = \mathcal{U}\mathfrak{W}^H \tag{10}$$

where $\mathcal{A} \in \mathbb{C}^{r\times s}$, $r \geq s$, and $\mathfrak{W} \in \mathbb{C}^{r\times s}$ is the MWGS transformation that produces the block upper triangular matrix with 1's on the main diagonal, $\mathcal{U} \in \mathbb{C}^{s\times s}$, such that

$$\mathcal{A}^H\mathcal{D}_{\mathcal{A}}\mathcal{A} = \mathcal{U}\mathcal{D}_{\mathcal{U}}\mathcal{U}^H \text{ and } \mathfrak{W}^H\mathcal{D}_{\mathcal{A}}\mathfrak{W} = \mathcal{D}_{\mathcal{U}} \tag{11}$$

where the diagonal matrices are $\mathcal{D}_{\mathcal{A}} \in \mathbb{C}^{r\times r}$, $\mathcal{D}_{\mathcal{U}} \in \mathbb{C}^{s\times s}$ and $\mathcal{D}_{\mathcal{A}} > 0$; see [14, Lemma VI.4.1] for an extended explanation.

Further, taking into account the block structure of the system matrices in (2), (3), for convenience we omit the superscript (k) in filter equations. Hence, the new combined array UD-based Information Filter can be written as follows.

Table 3. Combined array UD based Information Filter (caUD-IF).

I. Initial data	
I.1. Set: I.2. Calculate the modified Cholesky factors:	$Y_0 = P_0^{-1}$, $d_0 = Y_0\bar{x}_0$. $\{U_{Y_0}, D_{Y_0}\}$
I.3. Set:	$\{U_{Q_0} = I, D_{Q_0} = Q_0\}, \{U_{R_0} = I, D_{R_0} = R_0\}$
II. Recursively update $\{U_{Y_{t+1}}, D_{Y_{t+1}}\}$ and $\hat{d}_{t+1}$ as follows ($t > 0$):	
II.1. For $\{U_{Y_t}, D_{Y_t}\}$, $\{U_{R_t} = I, D_{R_t} = R_t\}$ construct a pair of the pre-arrays $\{\mathcal{A}, \mathcal{D}_{\mathcal{A}}\}$:	$\mathcal{D}_{\mathcal{A}} = \text{diag}\begin{bmatrix} D_{Y_t} & D_{R_t}^{-1}\end{bmatrix}$, $\mathcal{A}^H = \begin{bmatrix}\Phi_t^{-H}U_{Y_t} & \Phi_t^{-H}H_t^H U_{R_t}^{-H}\end{bmatrix}$ (12)
II.2. Apply the MWGS transforms of the columns of $\mathcal{A}$ with respect to the weighting matrix $\mathcal{D}_{\mathcal{A}}$ to obtain a pair of the post-arrays $\{\mathcal{U}, \mathcal{D}_{\mathcal{U}}\}$:	$\mathcal{U} = U_{A_t}$, $\mathcal{D}_{\mathcal{U}} = D_{A_t}$ (13)
II.3. For $\{U_{Y_t}, D_{Y_t}\}$, $\{U_{Q_t} = I, D_{Q_t} = Q_t\}$ construct a pair of the pre-arrays $\{\mathcal{A}^*, \mathcal{D}_{\mathcal{A}}^*\}$:	$\mathcal{D}_{\mathcal{A}}^* = \text{diag}\begin{bmatrix} D_{A_t} & D_{Q_t}^{-1}\end{bmatrix}$, $(\mathcal{A}^*)^H = \begin{bmatrix} U_{A_t} & 0 \\ \Gamma^T U_{A_t} & U_{Q_t}^{-H}\end{bmatrix}$ (15)
II.4. Apply the MWGS transforms of the columns of $\mathcal{A}^*$ with respect to the weighting matrix $\mathcal{D}_{\mathcal{A}}^*$ to obtain a pair of the post-arrays $\{\mathcal{U}^*, \mathcal{D}_{\mathcal{U}}^*\}$:	$\mathcal{U}^* = \begin{bmatrix} U_{Y_{t+1}} & (L_t U_{C_t}) \\ 0 & U_{C_t}\end{bmatrix}$, $\mathcal{D}_{\mathcal{U}}^* = \text{diag}[D_{Y_{t+1}} \quad D_{C_t}]$ (16)
II.5. Compute information estimate:	$\hat{d}_{t+1} = \left[I - (L_t U_{C_t})U_{C_t}^{-1}\Gamma^T\right](\Phi_t)^{-H}\left(\hat{d}_t + H_t^H R_t^{-1} z_t\right)$ (17)

Remark 3. At any discrete-time moment t, one can easily obtain the predicted state estimate $\hat{x}_{t|t-1} = U_{Y_t}^{-H} D_{Y_t}^{-1} U_{Y_t}^{-1}\hat{d}_t$. Another way is to solve linear system $U_{Y_t} D_{Y_t} U_{Y_t}^H \hat{x}_{t|t-1} = \hat{d}_t$ by backward and forward substitutions.

The caUD-IF implementation scheme of main steps II.1–II.4 is shown in figure 3.

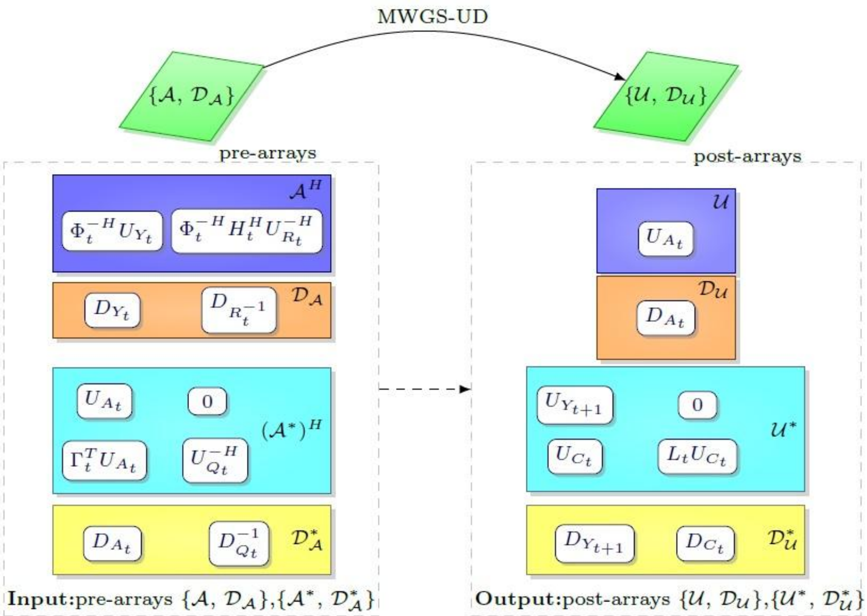

Figure 3. The caUD-IF implementation scheme.

It is very simple because of consists of only three steps:

1. Fill in the block pre-arrays with the input data.
2. Perform the MWGS-UD transforms.
3. Extract the required output data from the block post-arrays.

The designed eUD-CF estimator presented in table 2 has the main property to improve the accuracy and robustness of the computations for a finite-precision arithmetics. Now, we can formulate and prove our main result.

Statement 1. Algorithm caUD-IF is algebraically equivalent to the combined implementation of the Information Filter given by equations (4)–(8).

Proof: First, we can show the algebraic equivalence between equations (12), (13) and formulae (5) of the combined IF implementation. Indeed, taking into account the properties of orthogonal matrices, the first equation in (11) in terms of (12), (13) can be written as follows:

$$\Phi_t^{-H} U_{Y_t} D_{Y_t} U_{Y_t}^H \Phi_t^{-1} + \Phi_t^{-H} H_t^H U_{R_t}^{-H} D_{R_t}^{-1} U_{R_t}^{-1} H_t \Phi_t^{-1} = U_{A_t} D_{A_t} U_{A_t}^H \tag{18}$$

It is clear that expression (18) is equation (5).

Next, we can show that equations (15), (16) imply formulas (4)–(7). Indeed, again taking into account equations (10), (11) and the properties of orthogonal matrices, from (11) we obtain

$$(\mathcal{A}^*)^H \mathcal{D}_{\mathcal{A}}^* \mathcal{A}^* = \mathcal{U}^* \mathcal{D}_{\mathcal{U}}^* (\mathcal{U}^*)^H,$$

that is in terms of equations (15), (16) can be written as follows:

$$\begin{bmatrix} U_{A_t} D_{A_t} U_{A_t}^H & U_{A_t} D_{A_t} U_{A_t}^H \Gamma \\ \Gamma^T U_{A_t} D_{A_t} U_{A_t}^H & \Gamma^T U_{A_t} D_{A_t} U_{A_t}^H \Gamma + U_{Q_t}^{-H} D_{Q_t}^{-1} U_{Q_t}^{-1} \end{bmatrix} = \begin{bmatrix} U_{Y_{t+1}} D_{Y_{t+1}} U_{Y_{t+1}}^H + L_t U_{C_t} D_{C_t} U_{C_t}^H L_t^H & L_t U_{C_t} D_{C_t} U_{C_t}^H \\ U_{C_t} D_{C_t} U_{C_t}^H L_t^H & U_{C_t} D_{C_t} U_{C_t}^H \end{bmatrix} \tag{19}$$

Equating the corresponding (i, j)-th submatrices in (19), we obtain

$$(2{,}2) \Rightarrow \Gamma^T U_{A_t} D_{A_t} U_{A_t}^H \Gamma + U_{Q_t}^{-H} D_{Q_t}^{-1} U_{Q_t}^{-1} = U_{C_t} D_{C_t} U_{C_t}^H \Rightarrow \Gamma^T A_t \Gamma + Q_t^{-1} = C_t \tag{20}$$

$$(2{,}1) \Rightarrow \Gamma^T U_{A_t} D_{A_t} U_{A_t}^H = U_{C_t} D_{C_t} U_{C_t}^H L_t^H \Rightarrow \Gamma^T A_t = C_t L_t^H \tag{21}$$

$$(1{,}1) \Rightarrow U_{A_t} D_{A_t} U_{A_t}^H = U_{Y_{t+1}} D_{Y_{t+1}} U_{Y_{t+1}}^H + L_t U_{C_t} D_{C_t} U_{C_t}^H L_t^H \Rightarrow A_t = Y_{t+1} + L_t C_t L_t^H \tag{22}$$

It is obvious that expression (20) is equation (6). It is easy to show that (21) is equation (7). Expression (22) is (4), if one will take into account the last equality in (21). Hence, we proved that equations (15), (16) are equivalent to equations (4)–(7).

Finally, there is no need to prove the equivalence of (8) and (17). This completes the proof.

5. Parallel implementation of the proposed schemes

In this section, we consider a new channel estimation framework with m concurrent caUD-IF algorithms. This general scheme (see figure 4) is composed of a bank of blocks with MWGS-UD transforms, each filled in with its own subsystem matrices $\Phi_t^{(k)}$, $\Gamma^{(k)}$, $Q_t^{(k)}$, $H_t^{(k)}$, $R_t^{(k)}$. Each k-th caUD-IF algorithm, $k = 1, \ldots, m$, calculates its own information state estimate $\hat{d}_t^{(k)}$ independently of the other filters. This framework can be naturally implemented on a set of parallel processors using the concept of the *coarse-grained parallelism*.

On the other hand, each caUD-IF block in the proposed channel estimation framework is based on the time-consuming MWGS-UD transform and its effective implementation is important to overall performance. Since MWGS-UD transform consists of two computationally intensive nested loops they can be parallelized using the concept of the *fine-grained parallelism* and OpenMP technology. In [17] we have implemented MWGS-UD algorithm using Armadillo library [20] and parallelized the inner loop using the #pragma omp parallel for directive with num_threads clause. Using this implementation we have conducted computational experiments with different number of threads 1, 2, 4 and 8 on the set of randomly generated test matrices of sizes 100x100, 200x100, 200x200, 500x500, 1000x500, 1000x1000, 1500x1000, 1500x1500, 2000x1500 and 2000x2000.

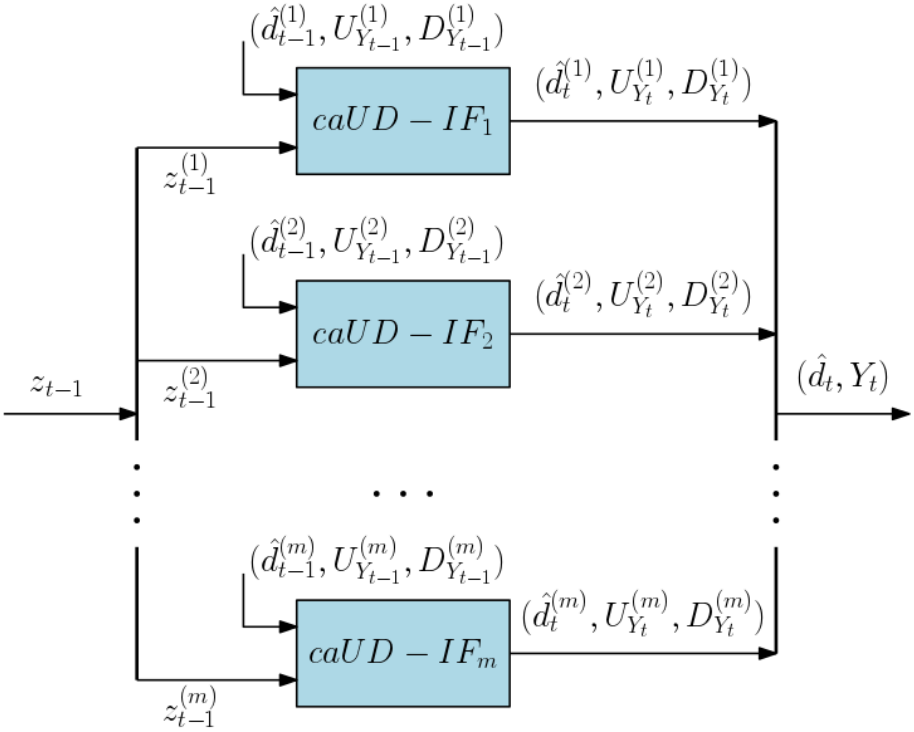

Figure 4. The caUD-IF based multi-subchannel estimation framework.

Numerical experiments were conducted at Scientific Research Laboratory of Mathematical Modeling, Ulyanovsk State Pedagogical University named after I. N. Ulyanov. Figure 5 illustrates the obtained results for each number of threads averaged over 20 runs.

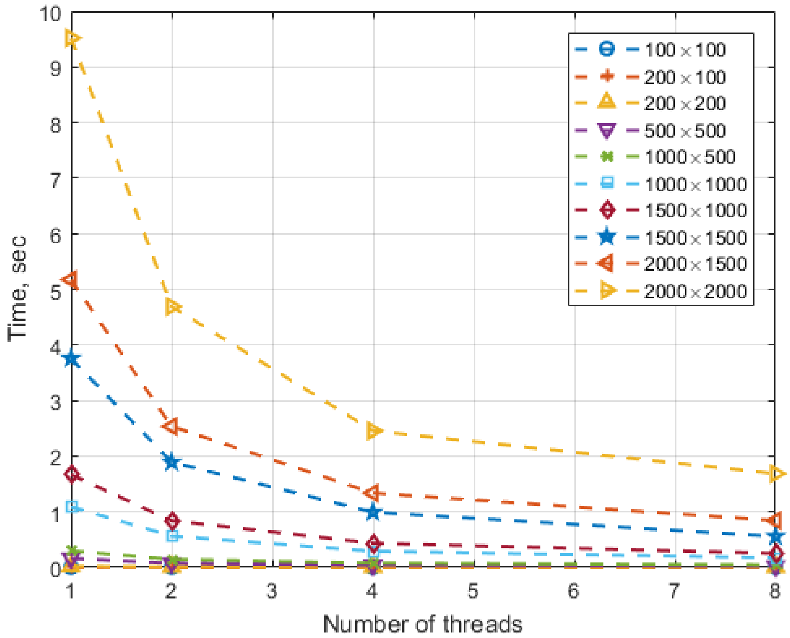

Figure 5. MWGS-UD average execution time (sec).

Figure 6 demonstrates parallel speedup for large matrices.

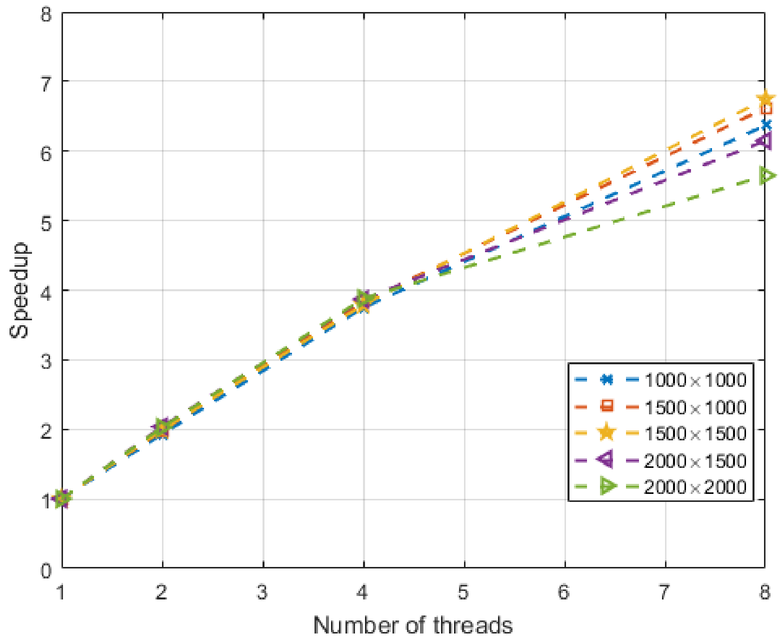

Figure 6. Parallel speedup.

5. Conclusions

The result of our work is twofold. Firstly, we have proposed the new numerically favored and convenient combined array UD Information Filter (caUD-IF algorithm). Secondly, we have demonstrated how network practitioners can efficiently use and implement this newest array algorithm based on MWGS transforms for coping with the difficulties caused by the numerical inefficiency of the standard Kalman Filter in attempting to use the latter for the OFDM multi-channel impulse response estimation. The advantages of the suggested solution are as follows:

- The channel estimating results are robust against the round-off errors.
- The computations do not contain the most time-consuming square-root operation.
- The compact and regular orthogonal array form of algorithm poses the best option for parallel computations in the Orthogonal Frequency-Division Multiple Access (OFDMA) multi-sub-channel organization.

6. References

[1] Viswanathan M 2013 *Simulation of Digital Communication Systems Using MATLAB* ed 2 (Los Gatos, CA: Smashwords, Inc.,) [eBook] p 292

[2] Grewal M S and Andrews A P 2001 *Kalman Filtering: Theory and Practice Using MATLAB* ed 2 (New Jersey: Prentice Hall) p 410

[3] Kailath T, Sayed A H and Hassibi B 2000 *Linear Estimation* (New Jersey: Prentice Hall) p 856

[4] Morf M and Kailath T 1975 Square-root algorithms for least-squares estimation *IEEE Trans. on Automatic Control* **AC-20** 4 pp 487–497

[5] Zhang S, Wang D and Zhao J 2014 A Kalman filtering channel estimation method based on state transfer coefficient using threshold correction for UWB systems *International Journal of Future Generation Communication and Networking* 7 pp 117–124

[6] Zhou J, Xia G and Wang J 2013 OFDM system channel estimation algorithm research based on Kalman filter compressed sensing *Journal of Theoretical and Applied Information Technology* **49** 1 pp 119–125

[7] Shi L, Zhou Z and Tang L 2012 Ultra wideband channel estimation based on Kalman filter compressed sensing *Transaction of Beijing Institute of Technology* **32** 2 pp 170–173

[8] Sternad M and Aronsson D 2003 Channel estimation and prediction for adaptive OFDM down-links *In Proceedings of the IEEE 58th Vehicular Technology Conference. VTC 2003-Fall (IEEE Cat. No.03CH37484)* (6–9 Oct. 2003) **2** pp 1283–1287

[9] Verhaegen M and Van Dooren P 1986 Numerical aspects of different Kalman filter implementations *IEEE Trans. on Automat. Contr.* **AC-31** pp 907–917

[10] D'Souza C and Zanetti R 2018 Information formulation of the UDU Kalman filter *IEEE Transactions on Aerospace and Electronic Systems* [Early Access]

[11] Hotop H-J 1989 New Kalman filter algorithms based on orthogonal transformations for serial and vector computers *Parallel Computing* **12** pp 233–247

[12] Park P and Kailath T 1995 New square-root algorithms for Kalman filtering *IEEE Trans. on Automatic Control* **40** 5 pp 895–899

[13] Dyer P and McReynolds S 1969 Extension of square-root filtering to include process noise *J. Optimiz. Theory Appl.* **3** pp 444–459

[14] Bierman G J 1977 *Factorization Methods for Discrete Sequential Estimation* (New York: Academic Press) p 241

[15] Jover J M and Kailath T 1986 A parallel architecture for Kalman filter measurement update and parameter estimation *Automatica* 22 1 pp 43–57

[16] Tsyganova Yu V 2013 On the UD filter implementation methods *University Proceedings. Volga Region* (*Physical and Mathematical Sciences*) 3(27) pp 84–104 [In Russian]

[17] Semushin I V, Tsyganova Yu V, Tsyganov A V and Prokhorova E F 2017 Numerically Efficient UD Filter Based Channel Estimation for OFDM Wireless Communication Technology *Procedia Engineering* **201** pp 726–735

[18] Golub G H and Van Loan C F 1983 *Matrix computations* (Baltimore, Maryland: Johns Hopkins University Press) p 694

[19] Björck A 1967 Solving least squares problems by orthogonalization *BIT* **7** pp 1–21

[20] Sanderson C and Curtin R 2016 Armadillo: a template-based C++ library for linear algebra *Journal of Open Source Software* **1** 26 p 7

Acknowledgments

The first three authors thank the support of Russian Foundation for Basic Research and the government of the Ulyanovsk region of the Russian Federation within the scope of project 18-47-730001 p_a.