Circuits Syst Signal Process (2009) 28: 845–867
DOI 10.1007/s00034-009-9131-6

A Stabilized Multichannel Fast RLS Algorithm for Adaptive Transmultiplexer Receivers

Dah-Chung Chang · Hsien-Cheng Chiu

Received: 21 September 2007 / Revised: 13 May 2009 / Published online: 11 September 2009

Abstract The transmultiplexer (TMUX) system has been studied for its application to multicarrier communications. The channel impairments including noise, interference, and distortion draw the need for adaptive reconstruction at the TMUX receiver. Among possible adaptive methods, the recursive least squares (RLS) algorithm is appealing for its good convergence rate and steady state performance. However, higher computational complexity due to the matrix operation is the drawback of utilizing RLS. A fast RLS algorithm used for adaptive signal reconstruction in the TMUX system is developed in this paper. By using the polyphase decomposition method, the adaptive receiver in the TMUX system can be formulated as a multichannel filtering problem, and the fast algorithm is obtained through the block Toeplitz matrix structure of received signals. In addition to the reduction of complexity, simulation results show that the adaptive TMUX receiver has a convergence rate close to that of the standard RLS algorithm and the performance approaches the minimum mean square error solution.

Keywords Adaptive signal processing · Fast recursive least squares (FRLS) · Polyphase decomposition · Toeplitz matrix · Transmultiplexer · Multicarrier communications · Minimum mean square error

This work was supported in part by the National Science Council of Taiwan under grants NSC92-2213-E-008-036) and NSC96-2219-E-008-003.

D.-C. Chang (✉)
Department of Communication Engineering, National Central University, 300 Jhongda Rd., Jhongli City, Taoyuan 320, Taiwan
e-mail: dcchang@ce.ncu.edu.tw

H.-C. Chiu
Metanoia Communications Inc., 5F, No. 12, Innovations Rd. 1,. Science-Based Industrial Park, Hsinchu 300, Taiwan
e-mail: tank_chiu@metanoia-comm.com

1 Introduction

The transmultiplexer (TMUX) is a bandwidth-efficient communication system that can simultaneously transmit multiple narrowband signals through a single wideband channel. The conventional implementation of TMUXs used the discrete Fourier transform (DFT) [12] for sub-channel allocation. Since the filterbank theory has been well developed in signal processing, the TMUX can use modulated filterbanks to digitally modulate/demodulate transmitted signals and it allows spectrum aliasing among sub-channels [1–3, 5, 9, 13–16, 20–23, 25, 26] in order to improve spectrum efficiency.

The filterbank of a TMUX system can be obtained from that of a subband system. Koilpillai, Nguyen, and Vaidyanathan have shown elaborate theories to obtain a crosstalk-free TMUX from an aliasing-free quadrature mirror filterbank (QMF) [15]. Although we can derive the perfect reconstruction (PR) property for TMUX systems based on the filterbank theory, the requirement of PR assumes an ideal transmission channel without noise and distortion between the transmitter and the receiver. However, noise, interference, and channel distortion always exist in a communication system, which draws the need for adaptive reconstruction at the TMUX receiver. In fact, we know that some popular adaptive filtering algorithms, e.g., least mean squares (LMS) [18, 19] and recursive least squares (RLS) [7], have been applied for signal reconstruction in a subband system. LMS algorithms are simple in architecture, but slow convergence rate may be a drawback in their applications with a long filter length [18, 19]. In [18], it was noted that the filterbank system consists of decimation and interpolation operations and thus fast RLS (FRLS) algorithms cannot be applied since the synthesis bank does not have a Toeplitz structure.

Some research proposed diverse approaches for TMUX applications based on the filterbank method are in [1–3, 5, 9, 13, 14, 16, 20–23, 25, 26] and the adaptive algorithm approaches for TMUX can be found in [2, 3, 5, 9, 25, 26]. The research of [2, 3, 13] recently proposed an adaptive equalizer for TMUX. However, their systems were established relying on the assumption of an oversampled filterbank in order to avoid the problem caused by inter-channel interference, but with the penalty of higher interpolation/decimation rate. For the maximally decimated TMUX systems, channel equalization was usually implemented at the filterbank output with post-combiner equalizers after the analysis bank. In [9], the post-combiner equalizers are simplified by taking into consideration only a few sub-channels spanned over the objective output for reducing the complexity of equalization to solve inter-channel interference. In [26], the standard minimum mean square error (MMSE) solution was derived for the post-equalizers applied to the output at the filterbank receiver in multicarrier communication applications; however, the interpolation and decimation operations involved in the filterbank system are modeled as matrices of large dimensions containing padded zeroes such that the MMSE solution becomes unfeasible. The MMSE method was also applied to a modified DFT-TMUX by utilizing polyphase decomposition for a generalized prototype transfer function [25]. However, the zero padded interpolation and decimation matrices still inhibit the development of fast adaptive algorithms which are more feasible than the MMSE solution.

Although the MMSE formulation has been successfully developed for signal reconstruction in maximally decimated TMUX systems, a recursive solution to adaptive

TMUX receivers for communication applications can be attractive. Since the RLS algorithm has better convergence rate and steady state performance than the LMS algorithm, the fast RLS algorithm for an adaptive TMUX receiver is developed in this paper in order to reduce the computational complexity. By exploring the polyphase structure of the analysis bank in a TMUX system, we found that signal reconstruction at the receiver can be viewed as a multichannel filtering problem. Using this formulation, a multichannel FRLS algorithm is derived. In addition, the method exploiting the convex combination scheme [24] for solving the stability problem connaturally residing in fast RLS algorithms is adapted. However, further analysis of the stability method from the theoretical aspect is another special issue and is beyond the scope of this paper, and hence, not a lot of work was extended to FRLS stability methods here. In simulation results, we show that the convergence rate of the multichannel FRLS algorithm is close to that of the standard RLS algorithm. The reconstruction performance approaches the bound of the Wiener solution. The bit error rate (BER) performance of the adaptive FRLS TMUX receiver is close to that of the orthogonal frequency division multiplexing (OFDM) technique in case of known channel state information (CSI).

2 System Model and Polyphase Decomposition

In Fig. 1, we depict an M-band TMUX system with the additive noise and channel distortion. Suppose the original signals present on M independent sub-channels to be $x_0(n), x_1(n), \ldots, x_{M-1}(n)$. Since the whole transmission band is divided by M sub-channels, the signal $x_i(n)$, $i = 0, 1, \ldots, M-1$ is first upsampled with the ratio of M and then passed through the bandpass synthesis filter f_i to shrink its spectrum within the subband of the corresponding sub-channel. The time index before upsampling is denoted as n while the index after upsampling is m, where we can write the relationship between m and n as $m = Mn - l$ and $l = 0, 1, \ldots, M-1$. The transmitted signal $y(m)$ is the sum of the M sub-channel signals. The TMUX receiver applies the analysis filter h_i to the received signal $z(m)$ for yielding the spectrum of the signals for the ith sub-channel. The signal with shrunk spectrum in the sub-channel is then transformed back to the full-band signal $\hat{x}_i(n)$ as a decimator with the decimation rate M performed after the analysis filter. Assume that the length of the analysis filter

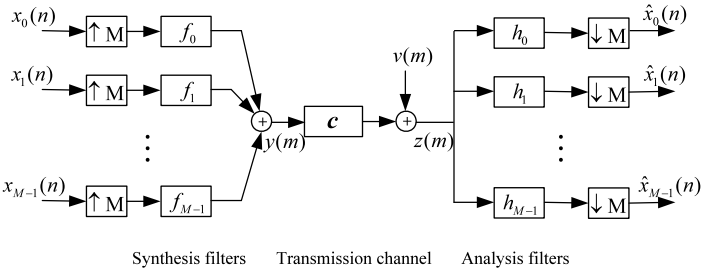

Fig. 1 An M-band TMUX system

is L; the reconstructed signal $\hat{x}_i(n)$ as shown in Fig. 1 can be expressed as

$$\hat{x}_i(n) = \sum_{\tau=0}^{L-1} h_i(\tau) z(Mn-\tau) \tag{1}$$

where $z(Mn-\tau)$ is the received signal and $h_i(\tau)$ denotes the τth coefficient of the ith analysis filter. In (1), the optimum Wiener solution of $h_i(z)$ can be obtained by calculating the statistics of the covariance matrix of $z(Mn-\tau)$ and the cross-correlation vector of the desired signal and $z(Mn-\tau)$. The suboptimal recursive algorithms, e.g. least mean squares (LMS) and recursive least squares (RLS) algorithms, can also be applied. However, we can see from (1) that the covariance matrix of the input vector $z(Mn-\tau)$ does not have the Toeplitz matrix property such that fast RLS algorithms cannot be developed [18] to solve $h_i(\tau)$ based on $z(Mn-\tau)$ and a training sequence.

To construct the Toeplitz property, we reformulate (1) by its polyphase decomposition as follows: Let $\tau = Mk+l$ and $P = L/M$ be an integer. Then, we have

$$\begin{aligned}\hat{x}_i(n) &= \sum_{l=0}^{M-1}\sum_{k=0}^{P-1} h_i(Mk+l) z(Mn-Mk-l) \\ &= \sum_{l=0}^{M-1}\left[\sum_{k=0}^{P-1} h_{l,i}(k) z_l(n-k)\right] \\ &= \sum_{l=0}^{M-1} h_{l,i}(n) * z_l(n)\end{aligned} \tag{2}$$

where $*$ denotes the operation of linear convolution, $h_{l,i}(n) = h_i(Mn+l)$ are the polyphase components of $h_i(m)$, and $z_l(n) = z(Mn-l)$ are the polyphase components of $z(m)$. Equation (2) can be interpreted as a result of multichannel filtering that has M-channel input signals $z_l(n)$ and M filters $h_{l,i}(n)$, which can lead to the reconstruction structure as depicted in Fig. 2. The functional block marked as "MMSE" can be some adaptive algorithm used for adjusting filter coefficients by the MMSE criterion. Define the coefficient vector of the analysis filter as

$$\mathbf{h}_i = \begin{bmatrix}\mathbf{h}_{0,i}^T & \mathbf{h}_{1,i}^T & \cdots & \mathbf{h}_{M-1,i}^T\end{bmatrix}_{L\times 1}^T \tag{3}$$

where

$$\mathbf{h}_{l,i} = \begin{bmatrix}h_{l,i}(0) & h_{l,i}(1) & \cdots & h_{l,i}(P-1)\end{bmatrix}^T, \tag{4}$$

and the input vector of the analysis filter as

$$\mathfrak{Z}(n) = \begin{bmatrix}Z_0^T(n) & Z_1^T(n) & \cdots & Z_{M-1}^T(n)\end{bmatrix}_{L\times 1}^T \tag{5}$$

where

$$Z_l(n) = \begin{bmatrix}z_l(n) & z_l(n-1) & \cdots & z_l(n-P+1)\end{bmatrix}_{K\times 1}^T. \tag{6}$$

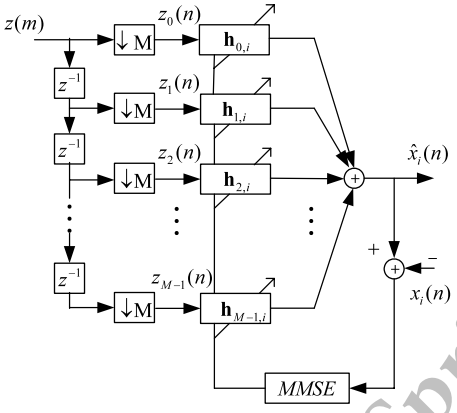

Fig. 2 The adaptive TMUX receiver structure

Note that $\mathfrak{Z}(n)$ is a column vector of L elements, containing all input signals from time index $n - P + 1$ to n for the M sub-channels. In deriving the algorithm as presented in Sect. 3, the number of elements will be augmented by M new input signals for the M sub-channels. From (3) and (5), (2) can be written as a product of two vectors:

$$\hat{x}_i(n) = \mathbf{h}_i^T \mathfrak{Z}(n) \tag{7}$$

Note that if T is a so-called $Mm \times Nn$ block Toeplitz matrix with $M \times N$ Toeplitz blocks of dimension $m \times n$, then T has the form

$$T = \begin{bmatrix} T_M & T_{M+1} & & \cdots & T_{M+N-1} \\ T_{M-1} & T_M & T_{M+1} & & \vdots \\ \vdots & T_{M-1} & T_M & \ddots & \vdots \\ & & T_{M-1} & \ddots & T_{M+1} \\ & & & \ddots & T_M \\ \vdots & & & & T_{M-1} \\ T_2 & & & & \vdots \\ T_1 & T_2 & \cdots & & T_N \end{bmatrix} \tag{8}$$

where T_j, $j = 1, 2, \ldots, M + N - 1$ are Toeplitz matrices. In general application, the covariance matrix of $Z_i(n)$ is usually treated as a Toeplitz matrix. Hence, the covariance matrix of the input vector $\mathfrak{Z}(n)$ becomes a block Toeplitz matrix.

3 Adaptive Reconstruction by Multichannel Fast RLS Algorithms

3.1 The Multichannel Fast RLS TMUX Receiver

In a communication signal processing system, the channel usually encounters distortion and narrowband interference. Although the TMUX system has been proved to

have better performance than the OFDM system in a narrowband interference channel [6], it needs an adaptive receiver to be implemented at the analysis filterbank terminal. The RLS algorithm has good convergence rate and steady state performance, but it requires the property of a Toeplitz covariance matrix for $\mathfrak{Z}(n)$ to achieve a better complexity reduction for using the fast RLS approach. Although the covariance matrix of $\mathfrak{Z}(n)$ is not Toeplitz, by using (7) we can develop a multichannel fast RLS algorithm for the adaptive TMUX receiver in light of a block step-up step-down approach which can be seen as a multichannel extension of the FAEST algorithm [4, 10]. A stabilized procedure is also considered in our algorithm.

We derive the multichannel fast RLS algorithm with the following steps:

Step 1: Define a priori and a posteriori errors.

Let $d_i(k), k = 0, 1, \ldots, n$ be desired signals for reconstruction. The a priori error is defined as

$$e_i(k) = d_i(k) - \mathbf{h}_i^T(n-1)\mathfrak{Z}(k) \tag{9}$$

and the a posteriori error is defined as

$$\varepsilon_i(k) = d_i(k) - \mathbf{h}_i^T(n)\mathfrak{Z}(k). \tag{10}$$

Using the exponentially weighted least squares method, we can find the estimate of $\mathbf{h}_i(n)$ by minimizing the following cost function:

$$\Phi(n) = \sum_{k=0}^{n} \lambda^{n-k} \varepsilon_i^2(k). \tag{11}$$

The parameter λ is an exponentially weighting factor that should be chosen in the range $0 \leq \lambda < 1$. Through the forgetting factor in (11), the information of the distant past has an increasingly negligible effect on the coefficient updating. Since $\Phi(n)$ is a quadratic form of $\mathbf{h}_i(n)$, the minimization can be obtained by setting the result to zero from differentiating $\Phi(n)$ with respect to $\mathbf{h}_i(n)$, and it follows that $\partial\Phi(n)/\partial\mathbf{h}_i(n) = 0$. After some mathematical manipulation, we have

$$R(n)\mathbf{h}_i(n) = \mathbf{r}_i(n) \tag{12}$$

where

$$R(n) = \sum_{k=0}^{n} \lambda^{n-k} \mathfrak{Z}(k)\mathfrak{Z}^T(k) = \lambda R(n-1) + \mathfrak{Z}(n)\mathfrak{Z}^T(n), \tag{13}$$

$$\mathbf{r}_i(n) = \sum_{k=0}^{n} \lambda^{n-k} \mathfrak{Z}(k) d_i(k) = \lambda \mathbf{r}_i(n-1) + \mathfrak{Z}(n) d_i(n). \tag{14}$$

Step 2: Find the recursive equation for $\mathbf{h}_i(n)$.

The time update equation of (12) is $R(n+1)\mathbf{h}_i(n+1) = \mathbf{r}_i(n+1)$. Substituted by recursive relations of (13) and (14) and manipulated by some simple algebraic operations, the time update equation yields

$$\mathbf{h}_i(n+1) = \mathbf{h}_i(n) + \mathbf{w}(n+1)\varepsilon_i(n+1) \tag{15}$$

where the Kalman gain $\mathbf{w}(n+1)$ is

$$\mathbf{w}(n+1) = \lambda^{-1} R^{-1}(n) \mathfrak{Z}(n+1). \tag{16}$$

Note that $\mathbf{w}(n+1)$ is a column vector of L elements and is independent of the polyphase index i. Hence the Kalman gain is the same for M decomposed reconstruction operations.

Step 3: Define forward and backward linear prediction (LP) errors based on a priori errors and a posteriori errors.

Now consider the time index to be k. Note that from (5) and (6), $\mathfrak{Z}(k)$ contains L input signals from time index $k-P+1$ to k, i.e., P signals for each one of the M channels. In order to derive a recursive formulation, we define a new column vector $\tilde{\mathfrak{Z}}(k)$ which contains $L+M$ input signals from time index $k-P$ to k, i.e., $P+1$ signals are invoked for each channel. Denote the new input signal for the jth channel as $z_j(k)$, $j=0,1,\ldots,M-1$. Since $Z_j(k-1)$ contains P input signals from time index $k-P$ to $k-1$ for the jth channel, we can express $\tilde{\mathfrak{Z}}(k)$ as

$$\tilde{\mathfrak{Z}}(k) = \begin{bmatrix} z_0(k) & Z_0^T(k-1) & \cdots & z_j(k) & Z_j^T(k-1) & \cdots & z_{M-1}(k) & Z_{M-1}^T(k-1) \end{bmatrix}^T \tag{17}$$

which can be treated as an augmented version of $\mathfrak{Z}(k)$ by extending $Z_j^T(k)$ to $[z_j(k)\ Z_j^T(k-1)]$ in (5). Let the vector $\tilde{Z}(k) = [z_0(k)\ z_1(k)\ \cdots\ z_{M-1}(k)]$ contain all of the new elements. The augmented signal vector can be partitioned by the following two means:

$$\tilde{\mathfrak{Z}}(k) = \mathfrak{T}^T \begin{bmatrix} \tilde{Z}(k) \\ \mathfrak{Z}(k-1) \end{bmatrix} = \mathfrak{S}^T \begin{bmatrix} \mathfrak{Z}(k) \\ \tilde{Z}(k-P) \end{bmatrix} \tag{18}$$

where $\mathfrak{T}^T$ and $\mathfrak{S}^T$ are permutation matrices. A permutation matrix is a square (0, 1)-matrix that has exactly one entry 1 in each row and each column and 0's elsewhere. The operation of the permutation matrices is described as follows: For example, we explain the operation of $\mathfrak{T}^T$, and the operation of $\mathfrak{S}^T$ has a similar result. As we augment L elements in $\mathfrak{Z}(k-1)$ to $L+M$ elements by adding the innovation vector $\tilde{Z}(k)$ of M elements, the order of the new elements can be rearranged by $\mathfrak{T}^T$ to the form as in (17) in order to remain the block Toeplitz property for $\tilde{\mathfrak{Z}}(k)$. We can say that $\mathfrak{T}\tilde{\mathfrak{Z}}(k)$ performs the forward partition of $\tilde{\mathfrak{Z}}(k)$ while $\mathfrak{S}\tilde{\mathfrak{Z}}(k)$ performs the backward partition. Besides, in matrix theory [11] we obtain that the permutation matrices are nonsingular and orthogonal, and have the property of $\mathfrak{T}^T = \mathfrak{T}^{-1}$ and $\mathfrak{S}^T = \mathfrak{S}^{-1}$.

Utilizing the above partitions, we can define two kinds of LP errors in the following. Forward LP errors based on a priori error and a posteriori error are

$$\mathbf{e}^f(k) = \tilde{Z}(k) - A^T(n-1)\mathfrak{Z}(k-1) \tag{19}$$

and

$$\boldsymbol{\varepsilon}^f(k) = \tilde{Z}(k) - A^T(n)\mathfrak{Z}(k-1), \tag{20}$$

respectively, and backward LP errors based on a priori error and a posteriori error are

$$\mathbf{e}^b(k) = \tilde{Z}(k-P) - B^T(n-1)\mathfrak{Z}(k) \tag{21}$$

and

$$\boldsymbol{\varepsilon}^b(k) = \tilde{Z}(k-P) - B^T(n)\mathfrak{Z}(k), \tag{22}$$

respectively, where $A(n)$ and $B(n)$ are $L \times M$ matrices. $A(n)$ is the forward predictor and $B(n)$ is the backward predictor. Using a posteriori LP equations, we have forward and backward normal equations

$$R(n-1)A(n) = R^f(n), \tag{23}$$

$$R(n)B(n) = R^b(n), \tag{24}$$

where

$$R^f(n) = \lambda R^f(n-1) + \mathfrak{Z}(n-1)\tilde{Z}^T(n), \tag{25}$$

$$R^b(n) = \lambda R^b(n-1) + \mathfrak{Z}(n)\tilde{Z}^T(n-P). \tag{26}$$

Step 4: Find recursive equations for $A(n)$ and $B(n)$.

Taking the same approach as in Step 2, the time update equations of (23) and (24) are substituted by the recursive relations of (25) and (26), respectively. After some simple algebraic operations, we obtain

$$A(n+1) = A(n) + \mathbf{w}(n)\boldsymbol{\varepsilon}^{fT}(n+1), \tag{27}$$

$$B(n+1) = B(n) + \mathbf{w}(n+1)\boldsymbol{\varepsilon}^{bT}(n+1). \tag{28}$$

Step 5: Find recursive equations for the Kalman gain.

Similar to the order update equation of (16), we augment the Kalman gain from L elements to $L+M$ elements in the following new order update equation:

$$\tilde{\mathbf{w}}(n+1) = \lambda^{-1}\tilde{R}^{-1}(n)\tilde{\mathfrak{Z}}(n+1), \tag{29}$$

where we define $\tilde{R}(n) = \sum_{k=0}^{n}\lambda^{n-k}\tilde{\mathfrak{Z}}(k)\tilde{\mathfrak{Z}}^T(k)$. Using (18), the correlation matrix $\tilde{R}(n)$ can be partitioned to

$$\tilde{R}(n) = \mathfrak{T}^T\begin{bmatrix}\mathbf{r}(n) & R^{fT}(n)\\ R^f(n) & R(n-1)\end{bmatrix}\mathfrak{T} = \mathfrak{S}^T\begin{bmatrix}R(n) & R^b(n)\\ R^{bT}(n) & \mathbf{r}(n-P)\end{bmatrix}\mathfrak{S} \tag{30}$$

where

$$\mathbf{r}(n-P) = \sum_{k=0}^{n}\lambda^{n-k}\tilde{Z}(k-P)\tilde{Z}^T(k-P). \tag{31}$$

The proof of (30) is referred to Appendix A. Then applying the matrix inversion lemma, $(\mathbf{F}^{-1} + \mathbf{C}\mathbf{D}^{-1}\mathbf{C}^H)^{-1} = \mathbf{F} - \mathbf{F}\mathbf{C}(\mathbf{D} + \mathbf{C}^H\mathbf{F}\mathbf{C})^{-1}\mathbf{C}^H\mathbf{F}$ [8], to (29), we have two recursive equations for $\tilde{\mathbf{w}}(n+1)$

$$\mathfrak{T}\tilde{\mathbf{w}}(n+1) = \begin{bmatrix}\mathbf{0}\\ \mathbf{w}(n)\end{bmatrix} + \begin{bmatrix}\mathbf{I}\\ -A(n)\end{bmatrix}\boldsymbol{\beta}^f(n+1), \tag{32}$$

$$\mathfrak{S}\tilde{\mathbf{w}}(n+1) = \begin{bmatrix}\mathbf{w}(n+1)\\ \mathbf{0}\end{bmatrix} + \lambda^{-1}\begin{bmatrix}-B(n)\\ \mathbf{I}\end{bmatrix}\boldsymbol{\alpha}^{b,-1}(n)\mathbf{e}^b(n+1) \tag{33}$$

where three variables are defined as

$$\boldsymbol{\beta}^f(n+1) = \lambda^{-1}\boldsymbol{\alpha}^{f,-1}(n)\mathbf{e}^f(n+1), \tag{34}$$

$$\boldsymbol{\alpha}^f(n) = \mathbf{r}(n) - R^{fT}(n)A(n), \tag{35}$$

$$\boldsymbol{\alpha}^b(n) = \mathbf{r}(n-P) - R^{bT}(n)B(n). \tag{36}$$

The backward equation (33) lets us have the following result. If we partition the backward equation as

$$\begin{bmatrix}\mathbf{c}(n+1)\\ \boldsymbol{\delta}(n+1)\end{bmatrix} = \mathfrak{S}\tilde{\mathbf{w}}(n+1), \tag{37}$$

where $\mathbf{c}(n+1)$ is a column vector of L elements and $\boldsymbol{\delta}(n+1)$ is a column vector of M elements, then we have

$$\mathbf{e}^b(n+1) = \lambda\boldsymbol{\alpha}^b(n)\boldsymbol{\delta}(n+1), \tag{38}$$

$$\mathbf{w}(n+1) = \mathbf{c}(n+1) + B(n)\boldsymbol{\delta}(n+1). \tag{39}$$

Step 6: Find recursive equations for $\boldsymbol{\alpha}^f(n)$ and $\boldsymbol{\alpha}^b(n)$.

Substituting the time update equations of (35) and (36) by the recursive relations of (27) and (28), respectively, after some simple algebraic operations, we find recursive equations for $\boldsymbol{\alpha}^f(n)$ and $\boldsymbol{\alpha}^b(n)$as follows:

$$\boldsymbol{\alpha}^f(n+1) = \lambda\boldsymbol{\alpha}^f(n) + e^f(n+1)\boldsymbol{\varepsilon}^{fT}(n+1), \tag{40}$$

$$\boldsymbol{\alpha}^b(n+1) = \lambda\boldsymbol{\alpha}^b(n) + \mathbf{e}^b(n+1)\boldsymbol{\varepsilon}^{bT}(n+1). \tag{41}$$

Step 7: Find relationships between $e_i(n)$ and $\varepsilon_i(n)$, $\mathbf{e}^f(n)$ and $\boldsymbol{\varepsilon}^f(n)$, and $\mathbf{e}^b(n)$ and $\boldsymbol{\varepsilon}^b(n)$.

From the time update equations of (10), (20), and (22), and the substitution of recursive equations (15), (27), and (28), the relationship between a priori errors and a posterior errors can be equated by

$$\varepsilon_i(n+1) = e_i(n+1)/\alpha(n+1), \tag{42}$$

$$\boldsymbol{\varepsilon}^f(n+1) = \mathbf{e}^f(n+1)/\alpha(n), \tag{43}$$

$$\boldsymbol{\varepsilon}^b(n+1) = \mathbf{e}^b(n+1)/\alpha(n+1) \tag{44}$$

where we define the variable

$$\alpha(n) = 1 + \mathfrak{Z}^T(n)\mathbf{w}(n). \tag{45}$$

Step 8: Find the recursive equation for $\alpha(n)$.

Consider the augmented order update equation of (45)

$$\tilde{\alpha}(n) = 1 + \tilde{\mathfrak{Z}}^T(n)\tilde{\mathbf{w}}(n). \tag{46}$$

Applying the forward partition to the right-hand side of (46), we obtain

$$\tilde{\alpha}(n+1) = \alpha(n) + \mathbf{e}^{fT}(n+1)\boldsymbol{\beta}^f(n+1). \tag{47}$$

Applying the forward partition to the right-hand side of (46), after some manipulations we have

$$\alpha(n+1) = \tilde{\alpha}(n+1) - \mathbf{e}^{bT}(n+1)\boldsymbol{\delta}(n+1). \tag{48}$$

For the purpose of simplifying the presentation of the proposed algorithm, the detailed derivation of (47) and (48) is stated in Appendix B.

The main idea of the fast algorithm is to update the Kalman gain $\mathbf{w}(n)$ by a block-wise step-up step-down procedure, i.e.,

$$\mathbf{w}(n) \rightarrow \tilde{\mathbf{w}}(n+1) \rightarrow \mathbf{w}(n+1). \tag{49}$$

The proposed TMUX reconstruction algorithm is listed in Table 1.

Table 1 The multichannel fast RLS TMUX algorithm

Initials:
$\mathbf{w}(0) = \mathbf{h}_i(0) = \mathbf{0}_{L\times 1},\ A(0) = B(0) = \mathbf{0}_{L\times M},$
$\alpha(0) = 1,\ \boldsymbol{\alpha}^f(0) = \boldsymbol{\alpha}^b(0) = \mathbf{I}_{M\times M},$
$0 < \lambda \leq 1$

Recursions	Count of MADS
The time updating of the Kalman gain:	
$\mathbf{e}^f(n+1) = \tilde{Z}(n+1) - A^T(n)\mathfrak{Z}(n)$	ML
$\boldsymbol{\beta}^f(n+1) = \lambda^{-1}\boldsymbol{\alpha}^{f,-1}(n)\mathbf{e}^f(n+1)$	M^2
$\mathfrak{T}\tilde{\mathbf{w}}(n+1) = \begin{bmatrix}\mathbf{0}\\ \mathbf{w}(n)\end{bmatrix} + \begin{bmatrix}\mathbf{I}_{M\times M}\\ -A(n)\end{bmatrix}\boldsymbol{\beta}^f(n+1)$	
Partition: $\begin{bmatrix}\mathbf{c}(n+1)\\ \boldsymbol{\delta}(n+1)\end{bmatrix} = \mathfrak{S}\tilde{\mathbf{w}}(n+1)$	
$\boldsymbol{\varepsilon}^f(n+1) = \mathbf{e}^f(n+1)/\alpha(n)$	M
$A(n+1) = A(n) + \mathbf{w}(n)\boldsymbol{\varepsilon}^{fT}(n+1)$	ML
$\mathbf{w}(n+1) = \mathbf{c}(n+1) + B(n)\boldsymbol{\delta}(n+1)$	ML
$\hat{\mathbf{e}}^b(n+1) = \tilde{Z}(n+1) - P) - B^T(n)\mathfrak{Z}(n+1)$	ML
$\mathbf{e}^b(n+1) = \lambda\boldsymbol{\alpha}^b(n)\boldsymbol{\delta}(n+1)$	M^2
$\mathbf{e}^{b,i}(n+1) = \tilde{\mathbf{e}}^b(n+1) + K_i(\hat{\mathbf{e}}^b(n+1) - \tilde{\mathbf{e}}^b(n+1)),\ i = 1,2,3$	$3M$
$\tilde{\alpha}(n+1) = \alpha(n) + \mathbf{e}^{fT}(n+1)\boldsymbol{\beta}^f(n+1)$	M
$\alpha(n+1) = \tilde{\alpha}(n+1) - \mathbf{e}^{bT}(n+1)\boldsymbol{\delta}(n+1)$	M
$\boldsymbol{\varepsilon}^b(n+1) = \mathbf{e}^b(n+1)/\alpha(n+1)$	M
$B(n+1) = B(n) + \mathbf{w}(n+1)\boldsymbol{\varepsilon}^{bT}(n+1)$	ML
$\boldsymbol{\alpha}^f(n+1) = \lambda\boldsymbol{\alpha}^f(n) + \mathbf{e}^f(n+1)\boldsymbol{\varepsilon}^{fT}(n+1)$	M^2
$\boldsymbol{\alpha}^b(n+1) = \lambda\boldsymbol{\alpha}^b(n) + \mathbf{e}^b(n+1)\boldsymbol{\varepsilon}^{bT}(n+1)$	M^2
The time updating of the analysis filters:	
For $i = 0$ to $M-1$ Do	
$e_i(n+1) = d_i(n+1) - \mathfrak{Z}^T(n+1)\mathbf{h}_i(n)$	L
$\varepsilon_i(n+1) = e_i(n+1)/\alpha(n+1)$	M
$\mathbf{h}_i(n+1) = \mathbf{h}_i(n) + \mathbf{w}(n+1)\varepsilon_i(n+1)$	L
End For	

3.2 The Complexity Issue

The proposed FRLS algorithm operates on M steps for each update of the Kalman gain. Each step performs a step-up step-down recursion for a subband channel. Hence only the scalar operation is required by this approach. Although the proposed algorithm needs M steps to complete one update, it can reduce computations near $5ML$ multiplications and divisions (MADS). The detailed count of MADS complexity for the proposed FRLS is also listed in Table 1 where a more accurate count of MADS for the proposed algorithm is $5ML + 4M^2 + 2L + 8M$ in total [6, 10, 24]. Note that $2M$ divisions are involved. Although the computational weight for a division operation could be heavier than a multiplication operation, the index of MADS is simple to use as an initial concept about the comparison of complexity [10]. Since the value of L is usually larger than that of M, the complexity of the proposed algorithm for updating the Kalman gain can be expressed as $O(5ML)$. As we use a similar method to calculate the MADS complexity for the standard RLS algorithm, the count of MADS is $2L^2 + 3L$ and the complexity of the RLS algorithm can be expressed as $O(2L^2)$. In Fig. 3, we show the complexity saving in percentage for different ratios of L to M by using the proposed FRLS algorithm compared with the standard RLS algorithm when $M = 64$. The complexity saving is defined as the ratio of the difference of the MADS complexity between the standard RLS algorithm and the proposed FRLS algorithm to the MADS complexity of the standard RLS algorithm. It can be noticed that the result is almost independent of the value of M for the chosen L/M ratio ≥ 4. Through an approximate observation, note that as we choose the ratio of L/M as P, the counts of MADS for the standard RLS algorithm and the proposed FRLS algorithm are about $O(2P^2M^2)$ and $O(5PM^2)$, respectively. The complexity saving in percentage can be measured as $(1 - \frac{2.5}{P}) \times 100\%$ for simplicity. For example, when $P = 8$, the complexity saving computed by the above simple rule is 68.75%, while a more accurate result as shown in Fig. 3 is 65.4%.

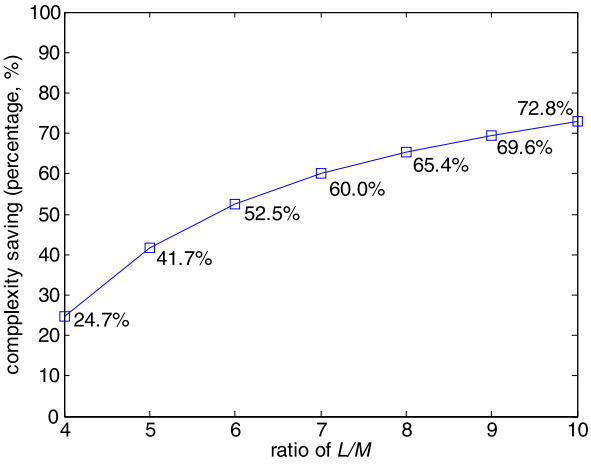

Fig. 3 Comparison of MADS complexity saving in percentage (%) by using the proposed FRLS instead of the standard RLS algorithm when $M = 64$

3.3 The Stabilized Algorithm

The problem of numerical instability is noticed when the fast algorithm is implemented with finite precision [24]. The numerically unstable behavior is related to λ, L, and the subband signal characteristics. It is found that the numerical problem is crucial to our filterbank reconstruction in the adaptive TMUX system. The problem of instability arises by the propagation of numerical errors when the fast RLS algorithm uses the backward prediction error equation (38). To stabilize fast algorithms, the backward prediction error feedback signals $\mathbf{e}^b(n+1)$ are replaced by new quantities. We notice that two equations (21) and (38) can be used to calculate $\mathbf{e}^b(n+1)$. However, they may have different behaviors that affect the stability in the feedback loop. If we redefine the following two computational methods for the backward prediction error:

$$\hat{\mathbf{e}}^b(n+1) = \tilde{Z}(n+1-P) - B^T(n)\mathfrak{Z}(n+1), \tag{50}$$

$$\tilde{\mathbf{e}}^b(n+1) = \lambda\boldsymbol{\alpha}^b(n)\boldsymbol{\delta}(n+1), \tag{51}$$

as a suggested thought in [24] the convex combination of (50) and (51) can be used to replace $\mathbf{e}^b(n+1)$:

$$\begin{aligned}\mathbf{e}^b(n+1) &= K\hat{\mathbf{e}}^b(n+1) + (1-K)\tilde{\mathbf{e}}^b(n+1)\\ &= \tilde{\mathbf{e}}^b(n+1) + K\big(\hat{\mathbf{e}}^b(n+1) - \tilde{\mathbf{e}}^b(n+1)\big)\end{aligned} \tag{52}$$

where K is a new empirical parameter to control the stabilizing behavior and $0 \le K \le 1$. As we explore the application of the term $\mathbf{e}^b(n+1)$ in the developed algorithm in Sect. 3.1, there are three equations (48), (28), and (41) using the quantities related to $\mathbf{e}^b(n+1)$. Although we can use (52) throughout the use for stabilization, it is a better method to use different K for those different equations in order to have more freedom in affecting the stability. Hence we choose three possible quantities for $\mathbf{e}^b(n+1)$ by changing different values of K, i.e., (52) becomes

$$\mathbf{e}^{b,i}(n+1) = \tilde{\mathbf{e}}(n+1) + K_i\big(\hat{\mathbf{e}}^b(n+1) - \tilde{\mathbf{e}}^b(n+1)\big), \quad i = 1,2,3. \tag{53}$$

By the order of the equations used in the proposed algorithm, we modified (48) as

$$\alpha(n+1) = \tilde{\alpha}(n+1) - \mathbf{e}^{b,1,T}(n+1)\boldsymbol{\delta}(n+1). \tag{54}$$

For (28) and (41), the value of $\mathbf{e}^b(n+1)$ contributes to the calculation of the quantity $\boldsymbol{\varepsilon}^b(n+1)$. Thus, we modify (44) in advance of (28) and (41) as

$$\boldsymbol{\varepsilon}^{b,i}(n+1) = \mathbf{e}^{b,i}(n+1)/\alpha(n+1), \quad i = 2,3. \tag{55}$$

And then (28) and (41) are modified according to (55) as

$$B(n+1) = B(n) + \mathbf{w}(n+1)\boldsymbol{\varepsilon}^{b,2,T}(n+1) \tag{56}$$

and

$$\boldsymbol{\alpha}^b(n+1) = \lambda\boldsymbol{\alpha}^b(n) + \mathbf{e}^{b,3}(n+1)\boldsymbol{\varepsilon}^{b,3,T}(n+1), \tag{57}$$

Table 2 The stabilized multichannel fast RLS TMUX algorithm

Initials:

$\mathbf{w}(0) = \mathbf{h}_i(0) = \mathbf{0}_{L\times 1},\ A(0) = B(0) = \mathbf{0}_{L\times M},$

$\alpha(0) = 1,\ \boldsymbol{\alpha}^f(0) = \boldsymbol{\alpha}^b(0) = \mathbf{I}_{M\times M},$

$0 \le K_i \le 1_i, i = 1, 2, 3$

$0 < \lambda \le 1$

Recursions:

The time updating of the Kalman gain:

$\mathbf{e}^f(n+1) = \tilde{Z}(n+1) - A^T(n)\mathfrak{Z}(n)$

$\boldsymbol{\beta}^f(n+1) = \lambda^{-1}\boldsymbol{\alpha}^{f,-1}(n)\mathbf{e}^f(n+1)$

$\mathfrak{T}\tilde{\mathbf{w}}(n+1) = \begin{bmatrix}\mathbf{0}\\ \mathbf{w}(n)\end{bmatrix} + \begin{bmatrix}\mathbf{I}_{M\times M}\\ -A(n)\end{bmatrix}\boldsymbol{\beta}^f(n+1)$

Partition: $\begin{bmatrix}\mathbf{c}(n+1)\\ \boldsymbol{\delta}(n+1)\end{bmatrix} = \mathfrak{S}\tilde{\mathbf{w}}(n+1)$

$\boldsymbol{\varepsilon}^f(n+1) = \mathbf{e}^f(n+1)/\alpha(n)$

$A(n+1) = A(n) + \mathbf{w}(n)\boldsymbol{\varepsilon}^{fT}(n+1)$

$\mathbf{w}(n+1) = \mathbf{c}(n+1) + B(n)\boldsymbol{\delta}(n+1)$

$\hat{\mathbf{e}}^b(n+1) = \tilde{Z}(n+1-P) - B^T(n)\mathfrak{Z}(n+1)$

$\tilde{\mathbf{e}}^b(n+1) = \lambda\boldsymbol{\alpha}^b(n)\boldsymbol{\delta}(n+1)$

$\mathbf{e}^{b,i}(n+1) = \tilde{\mathbf{e}}^b(n+1) + K_i(\hat{\mathbf{e}}^b(n+1) - \tilde{\mathbf{e}}^b(n+1)),\ i = 1, 2, 3$

$\tilde{\alpha}(n+1) = \alpha(n) + \mathbf{e}^{fT}(n+1)\boldsymbol{\beta}^f(n+1)$

$\alpha(n+1) = \tilde{\alpha}(n+1) - \mathbf{e}^{b,1,T}(n+1)\boldsymbol{\delta}(n+1)$

$\boldsymbol{\varepsilon}^{b,i}(n+1) = \mathbf{e}^{b,i}(n+1)/\alpha(n+1), i = 2, 3$

$B(n+1) = B(n) + \mathbf{w}(n+1)\boldsymbol{\varepsilon}^{b,2,T}(n+1)$

$\boldsymbol{\alpha}^f(n+1) = \lambda\boldsymbol{\alpha}^f(n) + \mathbf{e}^f(n+1)\boldsymbol{\varepsilon}^{fT}(n+1)$

$\boldsymbol{\alpha}^b(n+1) = \lambda\boldsymbol{\alpha}^b(n) + \mathbf{e}^{b,3}(n+1)\boldsymbol{\varepsilon}^{b,3,T}(n+1)$

The time updating of the analysis filters:

For $i = 0$ to $M-1$ Do

$e_i(n+1) = d_i(n+1) - \mathfrak{Z}^T(n+1)\mathbf{h}_i(n)$

$\varepsilon_i(n+1) = e_i(n+1)/\alpha(n+1)$

$\mathbf{h}_i(n+1) = \mathbf{h}_i(n) + \mathbf{w}(n+1)\varepsilon_i(n+1)$

End For

respectively. Based on the new convex combination scheme for $\mathbf{e}^b(n+1)$, the stabilized algorithm can be obtained by replacing three feedback quantities in (48), (28), and (41) from new values $\mathbf{e}^{b,i}(n+1)$, $i = 1, 2, 3$, respectively. Note that the new backward prediction error signals are applied three times for feedback in the proposed algorithm, where the three different quantities can be controlled by K_i with different characteristics of stability. The new stabilized TMUX reconstruction algorithm is listed in Table 2.

3.4 The MSE Performance Bound

Let the N-tap channel impulse response be $\{c_0\ c_1\ \cdots\ c_{N-1}\}$, the polyphase components of $y(m)$ be $y_l(n) = y(Mn - l)$, and the polyphase components of $v(m)$ be

$v_l(Mn-l)$. As shown in Fig. 1, $y(m)$ is the transmitted signal and $v(m)$ is the channel noise. For simplicity of representation, we assume that $Q=(N+L-1)/M$ is an integer. Define

$$\mathbf{y}(n)=\begin{bmatrix}\mathbf{y}_S^T(n) & \mathbf{y}_S^T(n-1) & \cdots & \mathbf{y}_S^T(n-Q+1)\end{bmatrix}_{(N+L-1)\times 1}^T, \tag{58}$$

$$\mathbf{v}(n)=\begin{bmatrix}\mathbf{v}_S^T(n) & \mathbf{v}_S^T(n-1) & \cdots & \mathbf{v}_S^T(n-P+1)\end{bmatrix}_{L\times 1}^T \tag{59}$$

where

$$\mathbf{y}_S(n)=\begin{bmatrix}y_0(n) & y_1(n) & \cdots & y_{M-1}(n)\end{bmatrix}_{M\times 1}^T, \tag{60}$$

$$\mathbf{v}_S(n)=\begin{bmatrix}v_0(n) & v_1(n) & \cdots & v_{M-1}(n)\end{bmatrix}_{M\times 1}^T. \tag{61}$$

Then, we have the following equation:

$$\mathfrak{Z}(n)=C\mathbf{y}(n)+\mathbf{v}(n) \tag{62}$$

where

$$C=\begin{bmatrix} c_0 & c_1 & \cdots & c_{N-1} & 0 & \cdots & 0\\ 0 & c_0 & c_1 & \cdots & c_{N-1} & \ddots & 0\\ \vdots & & & & & \ddots & \vdots\\ 0 & \ldots & 0 & c_0 & c_1 & \cdots & c_{N-1}\end{bmatrix}. \tag{63}$$

We next relate $y_l(n)$ to the input signal $x(n)$. The signal $y(m)$ can be expressed as follows:

$$y(m)=\sum_{i=0}^{M-1}\sum_{k=0}^{P-1} f_i(m-Mk)x_i(k) \tag{64}$$

where $f_i(m)$ denotes the mth coefficient of the ith synthesis filter. Using the polyphase representation, we can rewrite (64) as

$$\begin{aligned} y_l(n)&=\sum_{i=0}^{M-1}\sum_{k=0}^{P-1} f_i(Mn-Mk-l)x_i(k)\\ &=\sum_{i=0}^{M-1}\sum_{k=0}^{P-1} f_{l,i}(n-k)x_i(k)\\ &=\sum_{i=0}^{M-1}\left[\sum_{k=0}^{P-1} f_{l,i}(k)x_i(n-k)\right] \end{aligned} \tag{65}$$

where $f_{l,i}(n)=f_i(Mn-l)$ are the polyphase components of $f_i(m)$. To have a vector representation of (65), we first define

$$\mathbf{f}_l=\begin{bmatrix}\mathbf{f}_{l,0}^T & \mathbf{f}_{l,1}^T & \cdots & \mathbf{f}_{l,M-1}^T\end{bmatrix}_{L\times 1}^T, \tag{66}$$

$$\mathbf{x}_S(n)=\begin{bmatrix}\mathbf{x}_0^T(n) & \mathbf{x}_1^T(n) & \cdots & \mathbf{x}_{M-1}^T(n)\end{bmatrix}_{L\times 1}^T \tag{67}$$

where

$$\mathbf{f}_{l,i} = \begin{bmatrix} f_{l,i}(0) & f_{l,i}(1) & \cdots & f_{l,i}(P-1) \end{bmatrix}^T_{P\times 1}, \tag{68}$$

$$\mathbf{x}_i(n) = \begin{bmatrix} x_i(n) & x_i(n-1) & \cdots & x_i(n-P+1) \end{bmatrix}^T_{P\times 1}. \tag{69}$$

Then, (65) can be written as

$$y_l(n) = \mathbf{f}_l^T \mathbf{x}_S(n). \tag{70}$$

Define an extended input vector as

$$\mathbf{x}(n) = \begin{bmatrix} \mathbf{x}_S^T(n) & \mathbf{x}_S^T(n-1) & \cdots & \mathbf{x}_S^T(n-Q+1) \end{bmatrix}^T_{QL\times 1}. \tag{71}$$

Then, $\mathbf{y}(n)$ can be obtained as

$$\mathbf{y}(n) = \mathfrak{F}\mathbf{x}(n) \tag{72}$$

where $\mathfrak{F} = \text{diag}\{F, F, \ldots, F\}_{(N+L-1)\times QL}$ and $F = [\mathbf{f}_0\ \mathbf{f}_1\ \cdots\ \mathbf{f}_{M-1}]^T_{M\times L}$. Finally, using (7), (62), and (72), we have

$$\hat{x}_i(n) = \mathbf{h}_i^T C\mathfrak{F}\mathbf{x}(n) + \mathbf{h}_i^T \mathbf{v}(n). \tag{73}$$

The Wiener solution for the analysis filters is to find the optimal $\mathbf{h}_i$ which minimizes the mean square error between a desired signal $d_i(n)$ and the reconstruction signal $\hat{x}_i(n)$ according to polyphase decomposition model (7). The Wiener solution [8] can be obtained in the following matrix equation:

$$\mathbf{h}_i = R_{zz}^{-1} R_{zd_i} \tag{74}$$

where R_{zz} is the covariance matrix of $\mathfrak{Z}(n)$ and R_{zd_i} is the cross-correlation vector of $\mathfrak{Z}(n)$ and $d_i(n)$. From (62) and (72), we have that $\mathfrak{Z}(n) = C\mathfrak{F}\mathbf{x}(n) + \mathbf{v}(n)$. Assume that $\mathbf{x}(n)$ is independent of $\mathbf{v}(n)$; then R_{zz} can be calculated by

$$\begin{aligned} R_{zz} &= E\{[C\mathfrak{F}\mathbf{x}(n) + \mathbf{v}(n)][C\mathfrak{F}\mathbf{x}(n) + \mathbf{v}(n)]^T\} \\ &= C\mathfrak{F}E\{\mathbf{x}(n)\mathbf{x}^T(n)\}\mathfrak{F}^T C^T + C\mathfrak{F}E\{\mathbf{x}(n)\mathbf{v}^T(n)\} \\ &\quad + C\mathfrak{F}E\{\mathbf{v}(n)\mathbf{x}^T(n)\} + E\{\mathbf{v}(n)\mathbf{v}^T(n)\} \end{aligned} \tag{75}$$

where $E\{\cdot\}$ denotes the expectation operation. Notice that $E\{\mathbf{x}(n)\mathbf{v}^T(n)\} = E\{\mathbf{v}(n)\mathbf{x}^T(n)\} = \mathbf{0}_{L\times L}$ since $\mathbf{x}(n)$ and $\mathbf{v}(n)$ are independent. Then, (75) becomes

$$R_{zz} = C\mathfrak{F}R_{xx}\mathfrak{F}^T C^T + R_v \tag{76}$$

where R_v is the covariance matrix of the channel noise. For white noise, $R_v = \sigma_v^2 \mathbf{I}_{L\times L}$, where σ_v is the standard deviation of the noise. The desired signal $d_i(n)$ is the delayed version of the corresponding input signal, i.e., $x(n-D)$, where D accounts for the total effect of filterbank delay and channel delay. Hence,

$$R_{zd_i} = E\{[C\mathbf{y}(n) + \mathbf{v}(n)]x(n-D)\}$$
$$= C\mathfrak{F} \cdot E\{\mathbf{x}(n)x(n-D)\}. \tag{77}$$

We can find the Wiener solution of $\mathbf{h}_i$ by (74)–(77).

4 Simulations

The adaptive TMUX receivers for autoregressive (AR) signals and digitally modulated signals are simulated in our work to demonstrate the effectiveness of the proposed algorithm. The performance of the standard RLS algorithm and Koilpillai's algorithm presented in [15] is compared as well. The cosine-modulated filterbank is usually used to implement the synthesis bank $f_i(n)$ and the analysis bank $h_i(n)$, where the subscript i is the band index, $0 \le i \le M-1$, and n is the tap index, $0 \le n \le N-1$, by the following relations:

$$f_i(n) = 2p_0(n)\cos\left(\frac{\pi}{M}(i+0.5)\left(n - \frac{N-1}{2}\right) - \theta_i\right), \tag{78}$$

$$h_i(n) = 2p_0(n)\cos\left(\frac{\pi}{M}(i+0.5)\left(n - \frac{N-1}{2}\right) + \theta_i\right) \tag{79}$$

where $\theta_i = (-1)^i\pi/4$ and $p_0(n)$ is a linear-phase lowpass prototype filter. A five-band system with 55 coefficients is used here. We adopt the prototype filter presented in Table IV [17]. The reconstruction performance is measured by the reconstruction signal-to-noise ratio (SNR) which is defined by

$$\mathrm{SNR}(n) = 10\log_{10}\left(\frac{E\{x^2(n)\}}{E\{|x(n) - \hat{x}(n-D)|^2\}}\right) \tag{80}$$

where D is the transmission delay defined in (77). To evaluate performance after convergence, the average reconstruction SNR after the convergence of analysis bank coefficients is defined by

$$\mathrm{SNRc} = \frac{1}{n_2 - n_1 + 1}\sum_{n=n_1}^{n_2}\mathrm{SNR}(n) \tag{81}$$

where the average is calculated between the time indices n_1 and n_2. Note that the purpose of using n_1 and n_2 is to calculate the SNR after the adaptive algorithm converges in order to obtain an accurate estimate. Hence, there is no specific guideline as to the choice of n_1 and n_2 in general. The choice of n_1 is on the point after the algorithm converges and n_2 depends on how many samples are taken to average the calculation of the SNR in the time index. The noise is assumed to be white Gaussian and the channel impulse response to be $\{-0.077, -0.355, 0.059, 1, 0.059, -0.273\}$.

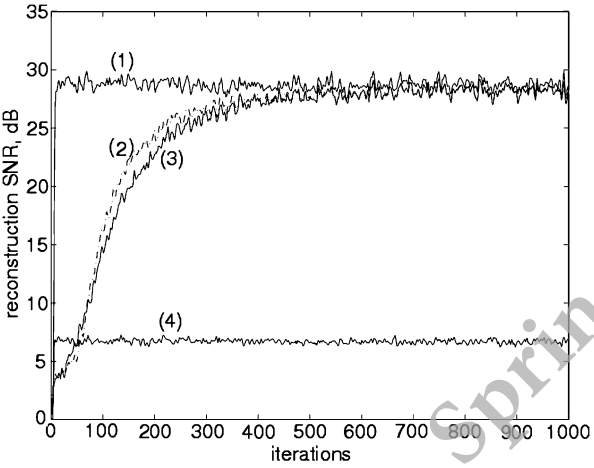

Fig. 4 Comparison of learning curves for reconstructing AR(1) signal with the subband input SNR = 30 dB: (*1*) the Wiener filter, (*2*) the standard RLS algorithm, (*3*) the stabilized multichannel FRLS algorithm, (*4*) Koilpillai's algorithm

4.1 AR Signals for the Adaptive TMUX Receiver

The input is assumed to be a first-order AR signal with correlation coefficient 0.8. The convergence characteristic of the proposed adaptive algorithm is shown in Fig. 4. Let the input SNR at the receiver be 30 dB and the forgetting factor for RLS algorithms be 0.995. The value of the forgetting factor is chosen here to be less than unity for dealing with a possible time-varying channel environment, but with the penalty of performance loss with respect to the Wiener solution in the static channel situation as shown in Fig. 4. From Fig. 4, we can see that the convergence behaviors of the standard RLS and the multichannel FRLS are very close, as is that of the steady state reconstruction SNR. Koilpillai's algorithm does not take into consideration the channel effect and thus results in poor performance.

4.2 The Property of the Forgetting Factor

The parameter λ dominates the convergence rate. In Fig. 5, we show the learning curves of the reconstruction SNR for $\lambda = 1$, 0.995, 0.99, 0.98, and 0.95. It is obvious that smaller λ can result in faster convergence rate, but larger error. Hence the tradeoff between steady state performance and convergence rate exists in choosing a proper value of λ. However, in practical applications we can use a large value for λ at the initial stage for fast convergence, and then change to a smaller value for better steady state performance.

Then we compare the reconstruction SNRc for different receiver algorithms and parameters in Fig. 6 where we calculate the SNR by choosing $n_1 = 501$ and $n_2 = 1500$. We can find that the Wiener filter outperforms the multichannel FRLS algorithm by 0.7 dB. This is due to the fact that the forgetting factor used in RLS is 0.995 instead of unity. At low input SNR, noise dominates the distortion. Thus, the difference of performance between conventional analysis filters and the proposed filters is not as significant as that at high input SNR. At high input SNR, the channel effect dominates. The proposed reconstruction filter acts like an equalizer and can achieve

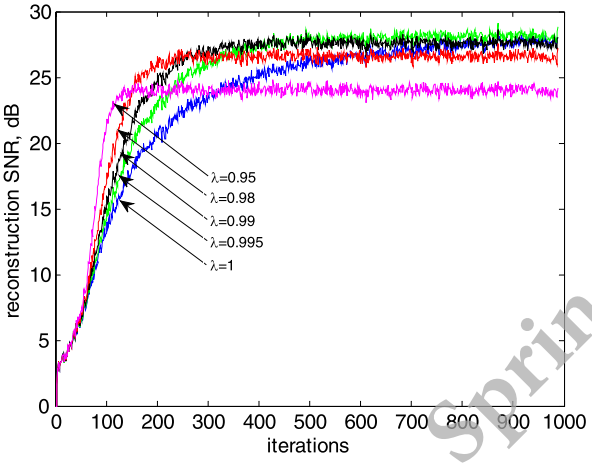

Fig. 5 Comparison of learning curves for reconstructing AR(1) signal by using the proposed algorithm at SNR = 30 dB for different forgetting factors

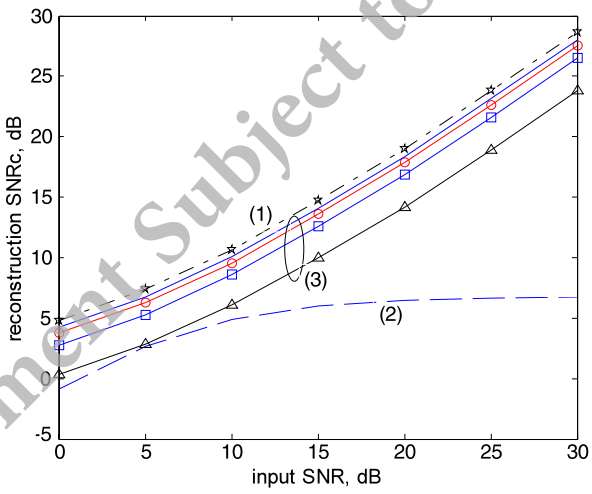

Fig. 6 Comparison of reconstruction performance for AR(1) signal with channel distortion and AWGN: (*1*) the Wiener filter, (*2*) Koilpillai's algorithm, and (*3*) the proposed algorithm with $\lambda = 0.995, 0.99, 0.98, 0.95$ *from top to bottom*

better performance. In Fig. 6, we also show the SNR performance for some typical values of λ.

4.3 The Stability Influence of K_i

We have mentioned that the unstable problem of the proposed algorithm is due to the numerical error propagation of the backward prediction error signal. The new feedback quantities are used to reduce this effect. Denote the difference $\Delta(n)$ as $\Delta(n) = \hat{\mathbf{e}}^b(n) - \tilde{\mathbf{e}}^b(n)$. To see the influence in the stabilized algorithm, the following comparison for the first entry of vector $\Delta(n)$ is performed by using the algorithm listed in Table 2 with different parameters K_i and λ. The other entries have a similar property. The results are shown in Fig. 7. When $K_i = 0$, the stabilized algorithm becomes the non-stabilized algorithm as listed in Table 1. We can see from Fig. 7(a) that $\Delta(n)$ is very large. When the stabilized algorithm is applied, we set the parame-

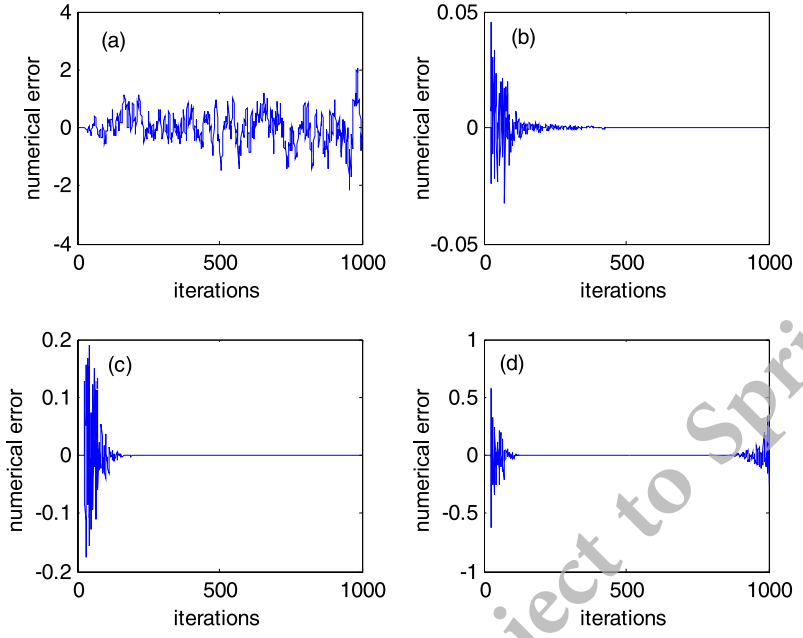

Fig. 7 Stability comparison of the proposed algorithm: (**a**) $K_i = 0$, $\lambda = 0.995$; (**b**) $K_i = 1$, $\lambda = 0.995$; (**c**) $K_i = 1$, $\lambda = 0.97$; (**d**) $K_i = 1$, $\lambda = 0.95$

ters $K_i = 1$ in our simulations for simplicity. The results with $\lambda = 0.995, 0.97$, and 0.95 are shown in Fig. 7(b)–(d). It is apparent that the value of $\Delta(n)$ is diminished close to zero. A detailed discussion on the setting of the convex combination parameters K_i can be found in [24]. However, we observed that λ is also related to the algorithm stability. Although a smaller λ has a faster convergence rate, a larger noise is introduced as well. Hence, a smaller λ makes the fast RLS algorithm suffer from divergence again after a lot of iterations.

4.4 The Extension to Digitally Modulated Signals

In Fig. 8, we show the BER performance of the proposed FRLS algorithm used for a TMUX receiver in comparison with that of a 16-QAM OFDM system. The filterbank (FB) used in the TMUX system has 32 bands with the coefficients of 128 taps, which is obtained by the method proposed in [27]. The FB TMUX FRLS simulation uses the synthesis bank of fixed coefficients as the transmitter and the proposed FRLS algorithm for the analysis bank as the adaptive TMUX receiver. To avoid intersymbol interference (ISI), a cyclic prefix (CP) of 8 samples is added in a 32-subcarrier OFDM symbol. The least squares (LS) channel estimation method assumes two block training symbols in advance of data symbols and the average result is used. Although the OFDM system has a simple structure to deal with a multipath channel, the data transmission throughput is decreased with a long CP length. From the result of the BER comparison, the adaptive TMUX receiver approaches the OFDM system with

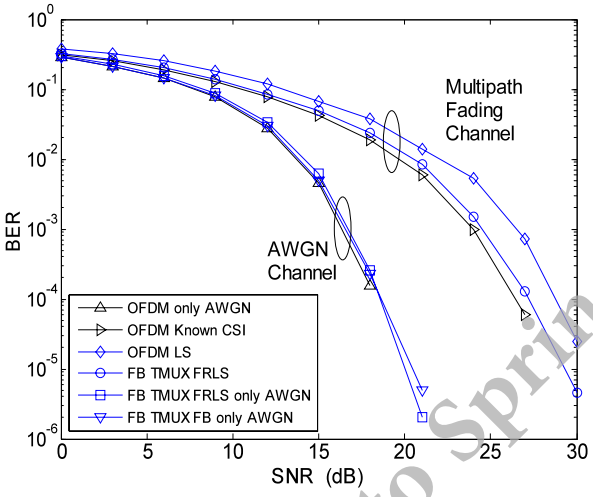

Fig. 8 BER comparison of OFDM and adaptive TMUX receivers in the AWGN and the multipath channels

known CSI, which performs better than the OFDM system with LS channel estimation since the LS estimation method is significantly influenced by noise. When the channel is only additive white Gaussian noise (AWGN), the performances of the adaptive TMUX, TMUX of fixed FB coefficients, and OFDM are almost the same. Although the adaptive TMUX receiver requires an adaptive analysis bank for equalization and, thus, has a higher complexity than that of the OFDM system with LS channel estimation, the performance of the proposed FRLS algorithm is close to the MMSE result and the computational requirement is reduced in comparison with RLS algorithms.

5 Conclusion

We developed a fast RLS algorithm for an adaptive TMUX receiver through the polyphase decomposition method, where the multichannel filtering problem is formulated to solve the inverse of a block Toeplitz matrix. To avoid numerical problems, a stabilized method is involved in the algorithm. From simulation results, we can see that the convergence rate of the proposed fast RLS algorithm is close to that of the standard RLS algorithm used in a TMUX system. The signal reconstruction performance approaches the theoretical result with the MMSE criterion.

Appendix A: Derivation of (30)

From (13) and (18), the covariance calculation of the augmented input vector via the forward partition is

$$\tilde{R}(n) = \sum_{k=0}^{n} \lambda^{n-k} \tilde{\mathfrak{Z}}(k) \tilde{\mathfrak{Z}}^T(k)$$

$$
\begin{aligned}
&= \sum_{k=0}^{n} \lambda^{n-k} \mathfrak{T}^T \begin{bmatrix} \tilde{Z}(k) \\ \mathfrak{Z}(k-1) \end{bmatrix} \begin{bmatrix} \tilde{Z}(k) \\ \mathfrak{Z}(k-1) \end{bmatrix}^T \mathfrak{T} \\
&= \mathfrak{T}^T \begin{bmatrix} \sum_{k=0}^{n} \lambda^{n-k} \tilde{Z}(k) \tilde{Z}^T(k) & \sum_{k=0}^{n} \lambda^{n-k} \tilde{Z}(k) \mathfrak{Z}^T(k-1) \\ \sum_{k=0}^{n} \lambda^{n-k} \mathfrak{Z}(k-1) \tilde{Z}^T(k) & \sum_{k=0}^{n} \lambda^{n-k} \mathfrak{Z}(k-1) \mathfrak{Z}^T(k-1) \end{bmatrix} \mathfrak{T} \\
&= \mathfrak{T}^T \begin{bmatrix} \mathbf{r}(n) & R^{fT}(n) \\ R^f(n) & R(n-1) \end{bmatrix} \mathfrak{T}.
\end{aligned}
$$

The covariance calculation via the backward partition is

$$
\begin{aligned}
\tilde{R}(n) &= \sum_{k=0}^{n} \lambda^{n-k} \tilde{\mathfrak{Z}}(k) \tilde{\mathfrak{Z}}^T(k) \\
&= \sum_{k=0}^{n} \lambda^{n-k} \mathfrak{S}^T \begin{bmatrix} \mathfrak{Z}(k) \\ \tilde{Z}(k-P) \end{bmatrix} \begin{bmatrix} \mathfrak{Z}(k) \\ \tilde{Z}(k-P) \end{bmatrix}^T \mathfrak{S} \\
&= \mathfrak{S}^T \begin{bmatrix} \sum_{k=0}^{n} \lambda^{n-k} \mathfrak{Z}(k) \mathfrak{Z}^T(k) & \sum_{k=0}^{n} \lambda^{n-k} \mathfrak{Z}(k) \tilde{Z}^T(k-P) \\ \sum_{k=0}^{n} \lambda^{n-k} \tilde{Z}(k-P) \mathfrak{Z}^T(k) & \sum_{k=0}^{n} \lambda^{n-k} \tilde{Z}(k-P) \tilde{Z}^T(k-P) \end{bmatrix} \mathfrak{S} \\
&= \mathfrak{S}^T \begin{bmatrix} R(n) & R^b(n) \\ R^{bT}(n) & \mathbf{r}(n-P) \end{bmatrix} \mathfrak{S}.
\end{aligned}
$$

Appendix B: Derivation of (47) and (48)

From (46), we have

$$
\alpha(n+1) = 1 + \tilde{\mathfrak{Z}}^T(n+1) \tilde{\mathbf{w}}(n+1). \tag{B.1}
$$

By the forward partition scheme in (18) and (32), we can express (B.1) as

$$
\begin{aligned}
\tilde{\alpha}(n+1) &= 1 + \left(\mathfrak{T}^T \begin{bmatrix} \tilde{Z}(n+1) \\ \mathfrak{Z}(n) \end{bmatrix} \right)^T \tilde{\mathbf{w}}(n+1) \\
&= 1 + \left[\tilde{Z}^T(n+1) \mathfrak{Z}^T(n) \right] \mathfrak{T} \tilde{\mathbf{w}}(n+1) \\
&= 1 + \left[\tilde{Z}^T(n+1) \mathfrak{Z}^T(n) \right] \left(\begin{bmatrix} \mathbf{0} \\ \mathbf{w}(n) \end{bmatrix} + \begin{bmatrix} \mathbf{I} \\ -A(n) \end{bmatrix} \boldsymbol{\beta}^f(n+1) \right) \\
&= 1 + \mathfrak{Z}^T(n) \mathbf{w}(n) + \left[\tilde{Z}^T(n+1) - \mathfrak{Z}^T(n) A(n) \right] \boldsymbol{\beta}^f(n+1).
\end{aligned} \tag{B.2}
$$

By (45) and (19), (47) can be obtained from (B.2) as

$$
\tilde{\alpha}(n+1) = \alpha(n) + \mathbf{e}^{fT}(n+1) \boldsymbol{\beta}^f(n+1).
$$

As we apply the backward partition scheme in (18) and (33), (B.1) can be expressed as

$$\begin{aligned}\tilde{\alpha}(n+1) &= 1+\left(\mathfrak{S}^T\begin{bmatrix}\mathfrak{Z}(n+1)\\ \tilde{Z}(n+1-P)\end{bmatrix}\right)^T\tilde{\mathbf{w}}(n+1)\\ &= 1+\left[\mathfrak{Z}^T(n+1)\tilde{Z}^T(n+1-P)\right]\mathfrak{S}\tilde{\mathbf{w}}(n+1)\\ &= 1+\left[\mathfrak{Z}^T(n+1)\tilde{Z}^T(n+1-P)\right]\left(\begin{bmatrix}\mathbf{w}(n+1)\\ \mathbf{0}\end{bmatrix}\right.\\ &\quad\left.+\lambda^{-1}\begin{bmatrix}-B(n)\\ \mathbf{I}\end{bmatrix}\boldsymbol{\alpha}^{b,-1}(n)\mathbf{e}^b(n+1)\right)\\ &= 1+\mathfrak{Z}^T(n+1)\mathbf{w}(n+1)\\ &\quad+\left[\tilde{Z}^T(n+1-P)-\mathfrak{Z}^T(n+1)B(n)\right]\lambda^{-1}\boldsymbol{\alpha}^{b,-1}(n)\mathbf{e}^b(n+1).\end{aligned}\tag{B.3}$$

By (45), (21), and (38), (B.3) becomes

$$\tilde{\alpha}(n+1)=\alpha(n+1)+\mathbf{e}^{bT}(n+1)\boldsymbol{\delta}(n+1).\tag{B.4}$$

Then, (48) can be easily obtained by (B.4).

References

1. A.N. Akansu, P. Duhamel, X. Lin, M. de Courville, Orthogonal transmultiplexers in communications: a review. IEEE Trans. Signal Process. **46**(4), 979–995 (1998)
2. J. Alhava, Time-domain equalizer for filter bank-based multicarrier communications, in *IEEE International Conference on Communications*, vol. 1, pp. 184–188, June 2001
3. J. Alhava, M. Renfors, Adaptive sine-modulated/cosine-modulated filter bank equalizer for transmultiplexer, in *European Conference on Circuit Theory and Design*, pp. 28–31, Finland, August 2001
4. G. Carayannis, D.G. Manolakis, N. Kalouptsidis, A fast sequential algorithm for least-squares filtering and prediction. IEEE Trans. Acoust. Speech Signal Process. **31**(6), 1394–1402 (1983)
5. D.C. Chang, D.L. Lee, An adaptive transmultiplexer for fast RLS algorithms and its application to multicarrier communications, in *IEEE International Conference on Communications*, vol. 7, pp. 3247–3251, June 2006
6. D.-C. Chang, W.-R. Wu, D.L. Lee, A multicarrier communication system with the adaptive transmultiplexer receiver. Int. J. Electr. Eng. **14**(6), 409–420 (2007)
7. A.N. Delopoulos, S.D. Kollias, Optimal filter banks for signal reconstruction from noisy subband components. IEEE Trans. Signal Process. **44**(2), 212–224 (1996)
8. P.S.R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*, 2nd ed. (Kluwer Academic, Dordrecht, 2002)
9. B. Farhang-Boroujeny, Analysis of post-combiner equalizers in cosine-modulated filterbank-based transmultiplexer systems. IEEE Trans. Signal Process. **51**(12), 3249–3262 (2003)
10. G.O. Glentis, N. Kalouptsidis, Efficient order recursive algorithms for multichannel least squares filtering. IEEE Trans. Signal Process. **40**(6), 2433–2458 (1992)
11. D.A. Harville, *Matrix Algebra from a Statistician's Perspective* (Springer, Berlin, 2008)
12. B. Hirosaki, An orthogonally multiplexed QAM system using the discrete Fourier transform. IEEE Trans. Commun. **COM-29**(7), 982–989 (1981)
13. T. Ihalainen, T.H. Stitz, M. Renfors, Efficient per-carrier channel equalizer for filter bank based multicarrier systems, in *Proceedings of IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 3175–3178, May 2005

14. S.G. Kang, E.K. Joo, OFDM system with linear-phase transmultiplexer. Electron. Lett. **34**(13), 1292–1293 (1998)
15. R.D. Koilpillai, T.Q. Nguyen, P.P. Vaidyanathan, Some results in the theory of crosstalk-free transmultiplexers. IEEE Trans. Signal Process. **39**(10), 2174–2183 (1991)
16. Y.P. Lin, S.M. Phoong, ISI-free FIR filterbank transceivers for frequency-selective channels. IEEE Trans. Signal Process. **49**(11), 2648–2658 (2001)
17. K. Nayebi, T.P. Barnwell, III, M.J.T. Smith, Time-domain filter bank analysis: a new design theory. IEEE Trans. Signal Process. **40**(6), 1412–1428 (1992)
18. B. Paillard, J. Soumagne, P. Mabilleau, S. Morissette, Subband decomposition: an LMS-based algorithm to approximate the perfect reconstruction bank in the general case. IEEE Trans. Signal Process. **39**(1), 233–238 (1991)
19. M.P. Petraglia, P.B. Batalheiro, Filter bank design for a subband adaptive filtering structure with critical sampling. IEEE Trans. Circuits Systems II, Analog Dig. Signal Process. **51**(6), 1194–1202 (2004)
20. S.M. Phoong, Y. Chang, C.Y. Chen, DFT-modulated filterbank transceivers for multipath fading channels. IEEE Trans. Signal Process. **53**(1), 182–192 (2005)
21. R.P. Ramachandran, P. Kabal, Bandwidth efficient transmultiplexers, part 1: synthesis. IEEE Trans. Signal Process. **42**(1), 70–84 (1992)
22. R.P. Ramachandran, P. Kabal, Bandwidth efficient transmultiplexers, part 2: subband components and performance aspects. IEEE Trans. Signal Process. **42**(5), 1108–1121 (1992)
23. R.P. Ramachandran, P. Kabal, Transmultiplexers: perfect reconstruction and compensation of channel distortion. Signal Process. **21**, 261–274 (1990)
24. D.T.M. Slock, T. Kailath, Numerical stable fast RLS transversal adaptive filtering. IEEE Trans. Signal Process. **39**(1), 92–114 (1991)
25. D.S. Waldhauser, J.A. Nossek, MMSE equalization for bandwidth-efficient multicarrier systems, in *Proceedings of IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 5391–5394, May 2006
26. A.M. Wyglinski, P. Kabal, F. Labeau, Adaptive filterbank multicarrier wireless systems for indoor environments, in *Proc. IEEE Vehicular Technology Conf.*, pp. 336–340, Vancouver, BC, September 2002
27. H. Xu, W.S. Lu, A. Antonious, Efficient iterative design method for cosine-modulated QMF banks. IEEE Trans. Signal Process. **44**(7), 1657–1668 (1996)