

RESEARCH

Open Access



# Sequential convex combinations of multiple adaptive lattice filters in cognitive radio channel identification

Mehmet Tahir Ozden

## Abstract

Sequential convex combinations of multiple adaptive lattice filters using different exponential weighting factors in cognitive radio (CR) channel identification framework have been considered in this presentation. First, the sequential processing multichannel lattice stages (SPMLSs) are modified so as to be used in filter combination task. Then, two different combination schemes, i.e., regular combination of multiple lattice filters (R-CMLF) and decoupled combination of multiple lattice filters (D-CMLF), that utilize modified SPMLSs as filter structure have been proposed. A modified Gram-Schmidt orthogonalization of multiple channels of data, which is constituted in multiple filter combination task, is accomplished. A highly modular, regular, and reconfigurable filter structure, which is suitable for cognitive radios, is achieved with the combination processing implemented in an order-recursive fashion. The mean square deviation (MSD) performances of the schemes under stationary and nonstationary conditions have been presented and compared with the performances of multiple combination of least mean square (M-CLMS), decoupled combination of least mean square (D-CLMS) schemes, and component filters. It has also been shown that the fault tolerances of the proposed schemes are better than those of the component filters due to the redundancy introduced with combination processing, and that the proposed schemes bring together the desired adaptive filter features such as fast convergence and low steady-state MSD levels, which do not normally coexist.

**Keywords:** Cognitive radio, CR, 5G, Combining filters, Channel identification, Lattice filters, Sequential processing

## 1 Introduction

Cognitive radio (CR) and 5G are two recent developments in the design of next-generation wireless communication systems. CR is built on a software radio, functions as an intelligent system that is aware of its environment and uses the methodology of understanding-by-building to learn from the environment, and adapts to statistical variations in the input stimuli in order to establish reliable communication by efficient utilization of the radio spectrum [1–4].

A typical CR cycle includes spectrum sensing, analysis, reasoning, and adaptation to new operating parameter steps [5]. CR can detect the availability of a portion of frequency band through spectrum sensing and analysis steps [6]. During the reasoning step, it determines the

optimum operating parameters, so that no harmful interference to other users of the spectrum is generated due to its transmission. In the adaptation step, the radio switches to transmission and reception mode using its reconfigurability and reprogrammability property [7, 8], and tunes its operating parameters according to its best response strategy.

The concept of software radio as the backbone of CR on the other hand relies on the development of DSP technology that is flexible, reconfigurable, and reprogrammable by software to adapt to an environment where there are multiple services, standards, and frequency bands [8–11]. Correspondingly, the infrastructure in a software radio system is generally required to use reconfigurable VLSI hardware components such as DSP chip sets [12], FPGAs [13], embedded processors [14], and even general purpose processors [15].

An emerging requirement for CRs is location and environment awareness that involves modeling the capabilities

Correspondence: [mtozden@pirireis.edu.tr](mailto:mtozden@pirireis.edu.tr)  
Piri Reis University, Faculty of Engineering, Department of Electrical and Electronics Engineering, Tuzla, 34940 Istanbul, Turkey

of human beings and bats for realization of advanced and autonomous location and environment awareness features [16, 17]. Adaptive positioning, determining the coordinates of a cognitive radio in space, is a step towards realization of accurate location awareness in cognitive radios [18]. The author has recently proposed a receiver (equalizer) architecture for use in cognitive MIMO-OFDM radios that performs joint channel estimation and data detection, addresses the receiver complexity problems, and contributes to the flexibility, reconfigurability, and reprogrammability of receiver [19]. It was also shown in [20–22] that this receiver architecture can be configured for spectrum sensing as well as adaptive positioning function of cognitive radio virtually at no cost.

In parallel with the developments in CR, 5G has been introduced connoting features such as increased data rates, spectral efficiency, and low latency through extreme densification, massive MIMO, device-centric architectures, millimeter wave, smarter devices, native support for machine-to-machine (M2M) communication, and interference management [23, 24]. As the future mobile broadband will be largely driven by ultra-high-definition video and as the things around us will be always connected, it is envisioned that the new era of communication will be dominated by the need for more capacity as well as spectrum, which will result in the integration of cognitive radio concepts in 5G networks [25, 26].

Another important concept that can find application in CRs is related to combining of adaptive filters [27]. When a priori knowledge about the filtering scenario is limited or imprecise, as in a typical cognitive radio operational cycle [1], implementing the most adequate filter structure and adjusting its parameters becomes a difficult task, and inaccurate choices may result in poor performance. An intelligent way to overcome this difficulty is to rely on combining of adaptive filters, in an attempt to improve their properties in terms of convergence, tracking ability, steady-state misadjustment, robustness, or computational cost. Combining adaptive filters can also improve reliability by introducing redundant processing elements, i.e., by stepping up the natural fault tolerance inherent in adaptive filters. Note that the notion of improving reliability by introducing redundancy is in line with the recent developments in the field of reliable control, particularly, with the application of a system augmentation approach that reformulates the original system into a descriptor piecewise affine system and then takes advantage of the redundancy of this descriptor system formulation [28, 29], and that the concept of adaptivity itself can be considered as a form of intelligence built into a filtering mechanism.

It is possible to combine adaptive filters implementing different tasks or using different filter operating parameters, structures, and learning algorithms [30]. The recent

examples of adaptive filter combination tasks include the combination of adaptive filters from different families such as one gradient and one Hessian based in [31], the adaptive combination of proportionate filters for sparse echo cancellation in [32], the adaptive combination of subband adaptive filters for acoustic echo cancellation in [33], the convex combination of  $H_2$  and  $H_\infty$  filters for space-time adaptive equalization in [34], the online tracking of the changes in the nonlinearity within a signal by using a collaborative adaptive signal processing approach based on a combination (hybrid) filter in [35], the adaptive combination of Volterra kernels and its application to nonlinear echo cancellation in [36], the convex combination of nonlinear adaptive filters for active noise control in [37], the combination of adaptive filters for relative navigation in [38], finite impulse response (FIR)-infinite impulse response (IIR) adaptive hybrid combination in [39], the affine combination of two adaptive sparse filters for estimating large-scale multiple-input multiple-output (MIMO) channels in [40], the combinations of multiple kernel adaptive filters in [41], the low-complexity approximation to the Kalman filter using convex combinations of adaptive filters from different families in [42], and the proposition of a family of combined-step-size proportionate filters in [43].

Two possible strategies for combining filters are convex and affine combinations, where the mixing coefficient is restricted to be nonnegative and sum up to one and the condition on the mixing parameter is relaxed, allowing it to be negative respectively [44–46]. In fact, the affine combination scheme can be interpreted as a generalization of the convex combination since the mixing coefficient is not restricted to the interval [0,1]. Even though the affine combination scheme allows for smaller error levels in theory, it suffers from larger gradient noise in some situations [47]. The adaptation rule for the mixing coefficient in affine combinations is simpler than those in convex ones, and the correct adjustment of the step size for the updating of the mixing coefficient depends on some characteristics of the filtering scenario. Accordingly, the desired universal behavior for the affine combination, in which case the combined estimate is at least as good as the best of the component filters in steady state, cannot always be ensured, whereas convex combinations have a built-in mechanism to attain such universality [47].

Previously, plant identification via adaptive convex combination of least mean square (LMS) transversal filters with different step sizes and new algorithms for improved adaptive convex combination of least mean square (LMS) transversal filters were introduced in [48, 49] respectively. In this presentation, the objective is to address the implementation issues related to combining adaptive filters by developing modular, order-recursive, and reconfigurable

combination schemes. In order to capture the statistical variations in the CR environment, the combination of multiple lattice filters with different exponential weighting factors has been considered so as to bring together the convergence properties of fast filters that have small exponential weighting factors and steady-state MSD properties of slow filters that have large exponential weighting factors. Accordingly, the author envisions multiple adaptive lattice filters as channels of sequential processing multi-channel lattice stages (SPMLSs) [19–22, 50] and proposes to sequentially combine these multiple lattice filters in a CR channel identification task. In view of the aforementioned issues concerning convex vs. affine combinations, the focus is on convex combinations of multiple adaptive lattice filters.

As the first contribution, the regular combination of multiple transversal filters in [49] is adapted to multiple lattice filters, and then, as the second contribution, the decoupled combination of two transversal filters in [49] is extended to multiple filters and tailored to multiple lattice filters by developing order-recursive algorithms for combination processing and also by making use of orthogonalized data and multichannel lattice filter parameters. To the best of the author’s knowledge, neither schemes exist in the literature. Note that a complete modified Gram-Schmidt orthogonalization of multichannel input data, which avoids matrix inversions and enables scalar only operations, is attained, and this feature is considered critical, as matrix inversion is a major bottleneck in the design of embedded receiver architectures [51]. Additionally, the computational complexity as well as the performances of the proposed lattice filter combination schemes in stationary and nonstationary channel identification scenarios are investigated and compared to those of the multiple combination of least mean square (M-CLMS) and decoupled combination of least mean square (D-CLMS) schemes in [49].

The organization of this paper is as follows. Section 2 is about the problem formulation in which channel model and multiple channel identification are introduced. In Section 3, the original SPMLS is initially introduced, and then, the proposed modifications in the SPMLS are discussed. Subsequently, the regular combination of multiple lattice filters (R-CMLF) and decoupled combination of multiple lattice filters (D-CMLF) schemes are presented in Sections 4 and 5 respectively. The experimental results are accounted for in Section 6, and finally, Section 7 is concerned with conclusions.  $(\bullet)^*$  represents the complex conjugate of  $(\bullet)$ .  $(\bullet)^T$  and  $(\bullet)^H$  stand for the transpose and the Hermitan transpose of  $(\bullet)$  respectively. The variables  $i$  and  $n$  are the time indexes related to data and coefficients respectively,  $m$  is the index for the number of combining filters, and finally,  $\ell$  stands for the stage number of lattice filters.

## 2 Problem formulation

### 2.1 Channel model

The cognitive radio channel is modeled by using the tapped delay line model [52, 53], and the channel output signal,  $y(i)$ , in this model is given by :

$$y(i) = \mathbf{w}^H(n)\mathbf{x}(i) + u(i), \quad i = 1, 2, \dots, n, \quad (1)$$

where  $\mathbf{x}(i) = [x(i), x(i - 1), \dots, x(i - N + 1)]^T$  refers to the input signal vector.  $u(i)$  is the channel noise signal at time instant  $i$  and is a realization of white, independently and identically distributed (i.i.d.) Gaussian random process with zero mean and constant variance  $\sigma_u^2$  and is also independent of  $\mathbf{x}(i)$ .

Herein,  $\mathbf{w}(n) = [w_0(n), w_1(n), \dots, w_{N-1}(n)]^T$  is the channel coefficient vector defined over the entire observation interval  $1 \leq i \leq n$ , and  $N$  is the channel length. The channel coefficient vector in the model is assumed to change in accordance with the first-order Markov process in [54]:

$$\mathbf{w}(n + 1) = a\mathbf{w}(n) + \mathbf{q}(n), \quad (2)$$

where  $\mathbf{q}(n)$  represents an i.i.d. Gaussian-distributed random zero-mean vector with diagonal covariance matrix  $\mathbf{Q}(n) = E[\mathbf{q}(n)\mathbf{q}^H(n)]$ . Herein,  $E[\bullet]$  is defined as the statistical expectation operator. The initial values of the coefficient vector,  $\mathbf{w}^H(-1)$ , are also assumed Gaussian distributed with zero mean and variance  $\sigma_w^2$  and independent of  $\mathbf{q}(n)$ ,  $u(i)$ , and  $\mathbf{x}(i)$ .  $a$  is a constant close to 1.

For the channel defined by Eqs. (1) and (2), the degree of nonstationarity is stated in [54, 55] as:

$$\xi(n) = \frac{E[|\mathbf{q}^H(n)\mathbf{x}(n)|^2]}{E[|u(n)|^2]}. \quad (3)$$

For slow statistical variations,  $\xi(n)$  is small, and typically less than unity, whereas it is greater than unity for fast statistical variations, indicating that it is not advantageous to build an adaptive filter to solve the tracking problem.

### 2.2 Multiple channel identification

Figure 1 depicts the multiple channel identification problem under consideration, in which information about the channel and scenario is assumed available through CR’s network, or through CR’s analysis, reasoning, and adaptation cycles. The objective in this problem is to estimate the coefficients of channel in (1) using multiple exponentially weighted least squares (LS) adaptive filters, each of which is implemented with a different exponential weight factor. Accordingly, the optimal exponentially weighted LS solution for the coefficients of the  $m$ th filter can be found by minimizing the following cost function:

$$J_m(n) = \sum_{i=1}^n \lambda_m^{n-i} |e_n^m(i)|^2 + \delta \lambda_m^n \|\mathbf{P}_m(n)\|_2^2, \quad (4)$$

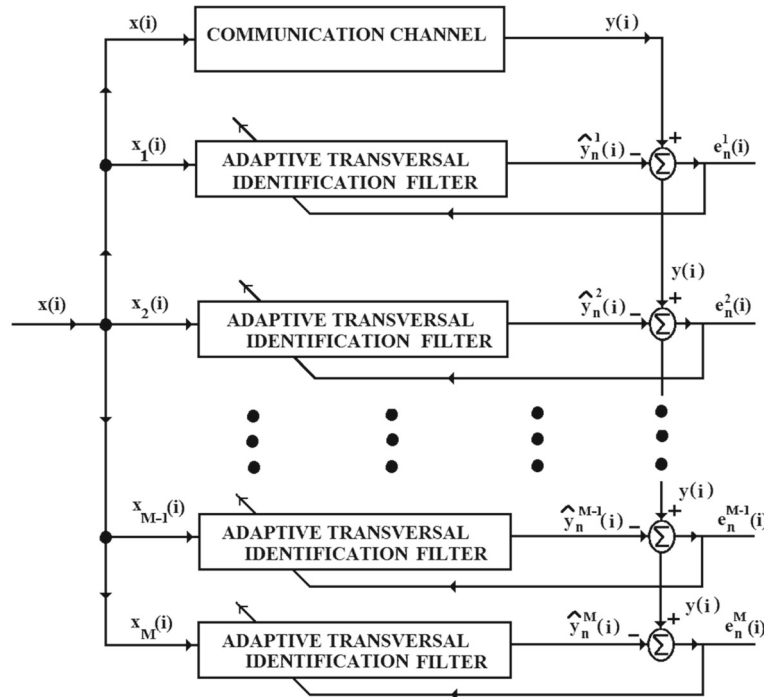


Fig. 1 A diagram of the multiple channel identification problem

at time instant  $n$ , where  $m = 1, 2, \dots, M$  and is the index for the component filters, and the use of prewindowing is assumed.  $\lambda_m$  is the exponential weighting factor for the  $m$ th filter.  $\delta$  is a positive real number called the regularization parameter [56]. Herein, the coefficient vector of the  $m$ th adaptive filter,  $\mathbf{P}_m(n)$ , at time instant  $n$ , is delineated as:

$$\mathbf{P}_m(n) = [P_{m,0}(n), P_{m,1}(n), \dots, P_{m,N_m-1}(n)]^T, \quad (5)$$

and its 2-norm is defined as  $\|\mathbf{P}_m(n)\|_2 = (\sum_{k=0}^{N_m-1} P_{m,k}(n)^2)^{1/2}$ . The  $m$ th estimation error at time instant  $i$ , computed using the input signal at time instant  $i$  and the filter coefficients at time instant  $n$ , is given by :

$$e_n^m(i) = y(i) - \mathbf{P}_m^H(n) \mathbf{x}_m(i), \quad i = 1, 2, \dots, n, \quad (6)$$

where the input signal vector to the  $m$ th adaptive filter,  $\mathbf{x}_m(i)$ , at time instant  $i$ , is defined as :

$$\mathbf{x}_m(i) = [x_m(i), x_m(i-1), \dots, x_m(i-N_m+1)]^T. \quad (7)$$

Note that  $x_m(i) = x(i), \forall m, N_m$  is the length of the  $m$ th filter.

Subsequently, the  $m$ th optimal coefficient vector is found by differentiating  $J_m(n)$  with respect to  $\mathbf{P}_m(n)$ , setting the derivative to zero, and solving for  $\mathbf{P}_m(n)$  :

$$\mathbf{P}_m^{opt}(n) = \mathbf{R}_{x_m x_m}^{-1}(n) \mathbf{R}_{x_m y}(n), \quad (8)$$

where  $\mathbf{R}_{x_m x_m}(n)$  is the  $N_m \times N_m$  correlation matrix of  $\mathbf{x}_m(i)$  and is given by :

$$\mathbf{R}_{x_m x_m}(n) = \sum_{i=1}^n \lambda_m^{n-i} \mathbf{x}_m(i) \mathbf{x}_m^H(i) + \delta \lambda_m^n \mathbf{I}, \quad (9)$$

in which the appearance of the second summational term is due to the regularizing term  $\delta \lambda_m^n \|\mathbf{P}_m(n)\|_2^2$  in the cost function  $J_m(n)$ , and  $\mathbf{I}$  is  $N_m \times N_m$  identity matrix. The  $N_m \times 1$  cross-correlation matrix of  $\mathbf{x}_m(i)$  and  $y(i)$  is expressed as:

$$\mathbf{R}_{x_m y}(n) = \sum_{i=1}^n \lambda_m^{n-i} \mathbf{x}_m(i) y^*(i). \quad (10)$$

Note that the channel length information can be available through the implementation of a channel length estimation algorithm such as [57] in CR. Accordingly, the lengths of all adaptive filters are assumed equal to the length of the channel to be identified, that is,  $N_m = N$ . All prediction and estimation errors hereafter are shown for the end of the observation interval  $i = n$ .

### 3 Sequential multichannel lattice processing

In order to provide a modular, order-recursive, and sequential solution to multiple filter combination problem, we propose to use SPMLSs, so that channels of SPMLSs constitute multiple filters with different exponential weighting factors. In the following, we first present the original SPMLS and its algorithm utilizing the direct

updating of a priori reflection coefficient form of processing equations in [58, 59] respectively. We then introduce the modifications to be implemented in the SPMLS in order to be able use its channels as filters in a combination task.

**3.1 The original SPMLS**

The original SPMLS has a block structure as shown in Fig. 2, and the input signal vectors to the SPMLS are defined as follows : the input forward prediction error vector,

$$\mathbf{f}_{\ell-1}(n) = [f_{\ell-1}^0(n), f_{\ell-1}^1(n), \dots, f_{\ell-1}^{M-1}(n), f_{\ell-1}^M(n)]^T, \tag{11}$$

the backward prediction error vector,

$$\mathbf{b}_{\ell-1}(n) = [b_{\ell-1}^0(n), b_{\ell-1}^1(n), \dots, b_{\ell-1}^{M-1}(n), b_{\ell-1}^M(n)]^T, \tag{12}$$

and the estimation error vector,

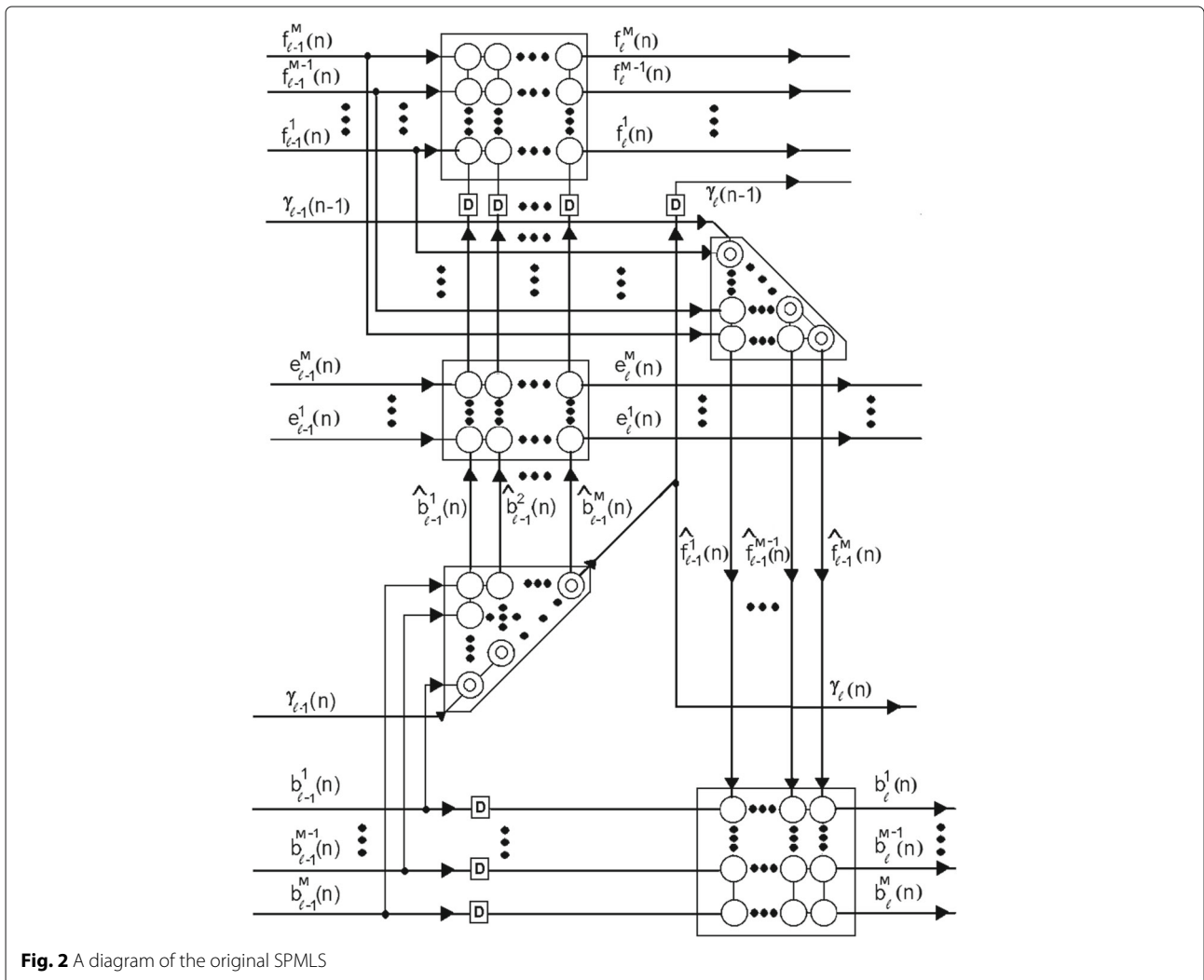
$$\mathbf{e}_{\ell-1}(n) = [e_{\ell-1}^0(n), e_{\ell-1}^1(n), \dots, e_{\ell-1}^{M-1}(n), e_{\ell-1}^M(n)]^T. \tag{13}$$

The elements of input forward and backward prediction error vectors in Eqs. (11) and (12) are orthogonalized by using self-orthogonalization processors (SOPs), which are triangular-shaped processors in Fig. 2. The outputs of SOPs are given in the orthogonalized forward prediction error vector,

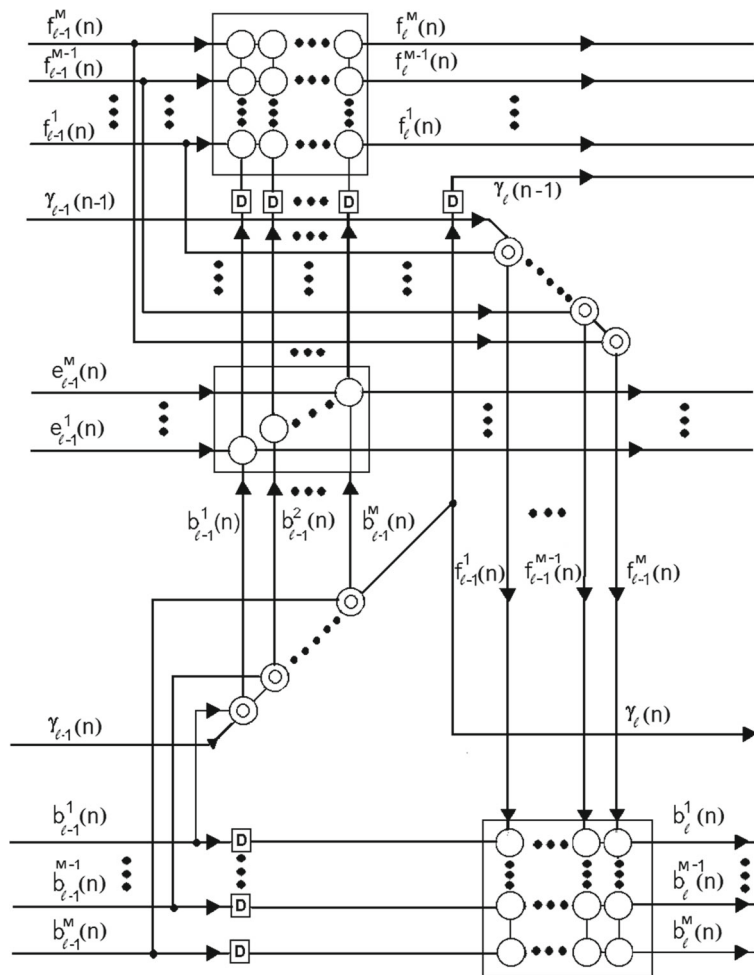
$$\hat{\mathbf{f}}_{\ell-1}(n) = [\hat{f}_{\ell-1}^0(n), \hat{f}_{\ell-1}^1(n), \dots, \hat{f}_{\ell-1}^{M-1}(n), \hat{f}_{\ell-1}^M(n)]^T \tag{14}$$

and the orthogonalized backward prediction error vector,

$$\hat{\mathbf{b}}_{\ell-1}(n) = [\hat{b}_{\ell-1}^0(n), \hat{b}_{\ell-1}^1(n), \dots, \hat{b}_{\ell-1}^{M-1}(n), \hat{b}_{\ell-1}^M(n)]^T. \tag{15}$$



**Fig. 2** A diagram of the original SPMLS



**Fig. 3** A diagram of the modified SPMLS

The elements of  $\hat{\mathbf{f}}_{\ell-1}(n)$  are fed into a forward prediction reference-orthogonalization processor (ROP) in order to predict the elements of  $\mathbf{b}_{\ell-1}(n-1)$  and to produce the stage output back prediction error vector  $\mathbf{b}_{\ell}(n)$ . The elements of  $\hat{\mathbf{b}}_{\ell-1}(n)$  are similarly fed into a ROP to perform  $M$ -channel joint process estimation and to produce the stage output estimation error vector  $\mathbf{e}_{\ell}(n)$ . Subsequently, the elements of  $\hat{\mathbf{b}}_{\ell-1}(n)$  are delayed and are also fed into another ROP to obtain the stage output forward prediction error vector  $\mathbf{f}_{\ell}(n)$ .

There are two types of processing cells, single and double circular processors in a SPMLS, and the complete SPMLS algorithm, which includes the processing equations in these cells, is provided in Table 1.

### 3.2 Modification of the SPMLS

In combining multiple filters, we take into account that input signals to all combining filters are the same as indicated in the sentence right after (7) and modify

the SPMLS by removing all single circular cells in self-orthogonalizing processors and redundant single circular cells in referential-orthogonalizing processors. Accordingly, the modified SPMLS does not have the orthogonalized forward and backward prediction errors that constitute the vectors in Eqs. (14) and (15). It neither includes the cross-estimation error terms in the ROP related to the joint-process estimation. The modified SPMLS and its algorithm are presented in Fig. 3 and Table 2 respectively.

### 4 Combinations of multiple lattice filters

In this section, we present the development of two combination schemes, namely, the R-CMLF and D-CMLF schemes, in order to sequentially combine multiple adaptive lattice filters. Even if these schemes may look similar, there is an important difference between them. In the R-CMLF scheme, combination parameters and mixing coefficients are computed globally at the last stage of combining filters and then are fed back to prior stages, whereas

**Table 1** The original SPMLS algorithm

Stage inputs and initialization

$$\bar{b}_m^1(n) = b_{\ell-1}^m(n), \bar{r}_m^1(n) = f_{\ell-1}^m(n), \bar{e}_m^1(n) = e_{\ell-1}^m(n) \quad (T.1.1)$$

$$\gamma_{\ell-1,1}^f(n) = \gamma_{\ell-1}(n-1), \gamma_{\ell-1,1}^b(n) = \gamma_{\ell-1}(n) \quad (T.1.2)$$

$$r_{\ell-1,k}^b(-1) = r_{\ell-1,k}^f(-1) = \delta, (k = 1, \dots, M) \quad (T.1.3)$$

$$\bar{\kappa}_{kj}^b(-1) = \bar{\kappa}_{kj}^f(-1) = \Delta_{kv}^e(-1) = \Delta_{kv}^f(-1) = \Delta_{kv}^b(-1) = 0.0 \quad (T.1.4)$$

$$(k = 1, \dots, M), (j = k + 1, \dots, M), (v = 1, \dots, M)$$

**For k=1,...,M**

Computations at SOPs

$$\hat{b}_{\ell-1}^k(n) = \bar{b}_k^k(n), \hat{r}_{\ell-1}^k(n) = \bar{r}_k^k(n) \quad (T.1.5)$$

$$r_{\ell-1,k}^b(n) = \lambda r_{\ell-1,k}^b(n-1) + \gamma_{\ell-1,k}^b(n) |\hat{b}_{\ell-1}^k(n)|^2 \quad (T.1.6)$$

$$\gamma_{\ell-1,k+1}^b(n) = \gamma_{\ell-1,k}^b(n) - |\gamma_{\ell-1,k}^b(n)|^2 |\hat{b}_{\ell-1}^k(n)|^2 / r_{\ell-1,k}^b(n) \quad (T.1.7)$$

$$r_{\ell-1,k}^f(n) = \lambda r_{\ell-1,k}^f(n-1) + \gamma_{\ell-1,k}^f(n) |\hat{r}_{\ell-1}^k(n)|^2 \quad (T.1.8)$$

$$\gamma_{\ell-1,k+1}^f(n) = \gamma_{\ell-1,k}^f(n) - |\gamma_{\ell-1,k}^f(n)|^2 |\hat{r}_{\ell-1}^k(n)|^2 / r_{\ell-1,k}^f(n) \quad (T.1.9)$$

**For j=k+1,...,M**

$$\bar{b}_j^{k+1}(n) = \bar{b}_j^k(n) - \bar{\kappa}_{kj}^{b*}(n-1) \hat{b}_{\ell-1}^k(n) \quad (T.1.10)$$

$$\bar{\kappa}_{kj}^b(n) = \bar{\kappa}_{kj}^b(n-1) + \gamma_{\ell-1,k}^b(n) \bar{b}_j^{k+1*}(n) \hat{b}_{\ell-1}^k(n) / r_{\ell-1,k}^b(n) \quad (T.1.11)$$

$$\bar{r}_j^{k+1}(n) = \bar{r}_j^k(n) - \bar{\kappa}_{kj}^{f*}(n-1) \hat{r}_{\ell-1}^k(n) \quad (T.1.12)$$

$$\bar{\kappa}_{kj}^f(n) = \bar{\kappa}_{kj}^f(n-1) + \gamma_{\ell-1,k}^f(n) \bar{r}_j^{k+1*}(n) \hat{r}_{\ell-1}^k(n) / r_{\ell-1,k}^f(n) \quad (T.1.13)$$

**End****For v = 1, ..., M**

Joint process estimation (ROP)

$$e_v^{k+1}(n) = e_v^k(n) - \Delta_{kv}^{e*}(n-1) \hat{b}_{\ell-1}^k(n) \quad (T.1.14)$$

$$\Delta_{kv}^e(n) = \Delta_{kv}^e(n-1) + \gamma_{\ell-1,k}^b(n) e_v^{k+1*}(n) \hat{b}_{\ell-1}^k(n) / r_{\ell-1,k}^b(n) \quad (T.1.15)$$

Forward error prediction (ROP)

$$f_v^{k+1}(n) = f_v^k(n) - \Delta_{kv}^{f*}(n-1) \hat{b}_{\ell-1}^k(n-1) \quad (T.1.16)$$

$$\Delta_{kv}^f(n) = \Delta_{kv}^f(n-1) + \gamma_{\ell-1,k}^f(n) f_v^{k+1*}(n) \hat{b}_{\ell-1}^k(n-1) / r_{\ell-1,k}^f(n-1) \quad (T.1.17)$$

Backward error prediction (ROP)

$$b_v^{k+1}(n) = b_v^k(n-1) - \Delta_{kv}^{b*}(n-1) \hat{r}_{\ell-1}^k(n) \quad (T.1.18)$$

$$\Delta_{kv}^b(n) = \Delta_{kv}^b(n-1) + \gamma_{\ell-1,k}^f(n) b_v^{k+1*}(n) \hat{r}_{\ell-1}^k(n) / r_{\ell-1,k}^f(n) \quad (T.1.19)$$

**End****End**

Stage outputs

$$b_{\ell}^m(n) = b_m^{M+1}(n), f_{\ell}^m(n) = f_m^{M+1}(n), \quad (T.1.20)$$

$$e_{\ell}^m(n) = e_m^{M+1}(n), \gamma_{\ell}(n) = \gamma_{\ell-1,M+1}^b(n) \quad (T.1.21)$$

they are computed locally in the D-CMLF scheme and therefore stage dependent. Hence, the total number of combination parameters and mixing coefficients in the R-CMLF and D-CMLF schemes is  $M$  and  $M \times N$  respectively.

#### 4.1 Regular combination of multiple lattice filters

A diagram of the R-CMLF scheme for  $M = 2$  case, that is, the regular combination of two lattice filters (R-CTLF), is presented in Fig. 4. Two types of combination

processors are introduced into the filter structure in this scheme: a type-1 combination processor per stage and a type-2 processor to the final stage. In the following, we present the development of combination algorithms that will be implemented by type-1 and type-2 processors in the R-CMLF scheme.

The estimate of equivalent desired signal in this scheme is computed in terms of the backward prediction errors at the output of the  $\ell$ th type-1 combination processor as:

**Table 2** The modified SPMLS algorithm

---

|   |          |
|---|----------|
| Stage inputs and initialization   |          |
| $\gamma_{\ell-1,1}^f(n) = \gamma_{\ell-1}(n-1), \gamma_{\ell-1,1}^b(n) = \gamma_{\ell-1}(n)$  | (T.2.1)  |
| $r_{\ell-1,k}^b(-1) = r_{\ell-1,k}^f(-1) = \delta, (k = 1, \dots, M)$   | (T.2.2)  |
| $\Delta_{\ell,k}^e(-1) = \Delta_{\ell,k}^f(-1) = \Delta_{\ell,k}^b(-1) = 0.0, (k = 1, \dots, M)$                                    | (T.2.3)  |
| <b>For k=1, ..., M</b>  |          |
| Computations at SOPs  |          |
| $r_{\ell-1,k}^b(n) = \lambda_k r_{\ell-1,k}^b(n-1) + \gamma_{\ell-1,k}^b(n)  b_{\ell-1}^k(n) ^2$                                    | (T.2.4)  |
| $\gamma_{\ell-1,k+1}^b(n) = \gamma_{\ell-1,k}^b(n) -  \gamma_{\ell-1,k}^b(n) ^2  b_{\ell-1}^k(n) ^2 / r_{\ell-1,k}^b(n)$            | (T.2.5)  |
| $r_{\ell-1,k}^f(n) = \lambda_k r_{\ell-1,k}^f(n-1) + \gamma_{\ell-1,k}^f(n)  f_{\ell-1}^k(n) ^2$                                    | (T.2.6)  |
| $\gamma_{\ell-1,k+1}^f(n) = \gamma_{\ell-1,k}^f(n) -  \gamma_{\ell-1,k}^f(n) ^2  f_{\ell-1}^k(n) ^2 / r_{\ell-1,k}^f(n)$            | (T.2.7)  |
| Joint process estimation (ROP)  |          |
| $e_{\ell}^k(n) = e_{\ell-1}^k(n) - \Delta_{\ell,k}^{e*}(n-1) b_{\ell-1}^k(n)$   | (T.2.8)  |
| $\Delta_{\ell,k}^e(n) = \Delta_{\ell,k}^e(n-1) + \gamma_{\ell-1,k}^b(n) e_{\ell}^{k*}(n) b_{\ell-1}^k(n) / r_{\ell-1,k}^b(n)$       | (T.2.9)  |
| Forward error prediction (ROP)  |          |
| $f_{\ell}^k(n) = f_{\ell-1}^k(n) - \Delta_{\ell,k}^{f*}(n-1) b_{\ell-1}^k(n-1)$   | (T.2.10) |
| $\Delta_{\ell,k}^f(n) = \Delta_{\ell,k}^f(n-1) + \gamma_{\ell-1,k}^b(n-1) f_{\ell}^{k*}(n) b_{\ell-1}^k(n-1) / r_{\ell-1,k}^b(n-1)$ | (T.2.11) |
| Backward error prediction (ROP)   |          |
| $b_{\ell}^k(n) = b_{\ell-1}^k(n-1) - \Delta_{\ell,k}^{b*}(n-1) f_{\ell-1}^k(n)$   | (T.2.12) |
| $\Delta_{\ell,k}^b(n) = \Delta_{\ell,k}^b(n-1) + \gamma_{\ell-1,k}^f(n) b_{\ell}^{k*}(n) f_{\ell-1}^k(n) / r_{\ell-1,k}^f(n)$       | (T.2.13) |
| <b>End</b>  |          |
| Stage outputs   |          |
| $\gamma_{\ell}(n) = \gamma_{\ell-1,M+1}^b(n)$   | (T.2.14) |

---

$$\hat{d}_{\ell,n}^{eq} = \sum_{j=1}^{\ell} \sum_{k=1}^M v^k(n) \Delta_{j,k}^* \Delta_{j-1}^k(n), \quad (16)$$

where  $v^k(n)$  and  $\Delta_{k,j}(n)$  represent the  $k$ th mixing coefficient and the  $k$ th estimation error reflection coefficient of the  $j$ th lattice stage at time instant  $n$  respectively.  $b_{j-1}^k(n)$  is the  $k$ th backward prediction error at the entrance of the  $j$ th stage at time instant  $n$ . The estimate of equivalent desired signal can be expressed order-recursively as follows :

$$\hat{d}_{\ell,n}^{eq} = \hat{d}_{\ell-1,n}^{eq} + v^m(n) \Delta_{\ell,m}^* \Delta_{\ell-1}^m(n), \quad (17)$$

where  $\ell = 1, \dots, N, m = 1, \dots, M$ , and  $\hat{d}_{0,n}^q = 0$ . Then, Eq. (17) is substituted in:

$$e_{\ell,n}^{eq} = d(n) - \hat{d}_{\ell,n}^{eq}, \quad (18)$$

to obtain the following expression:

$$e_{\ell,n}^{eq} = d(n) - \hat{d}_{\ell-1,n}^{eq} - v^m(n) \Delta_{\ell,m}^* \Delta_{\ell-1}^m(n). \quad (19)$$

Herein,  $d(n)$  represents the desired signal at time instant  $n$ , which is the channel output signal  $y(n)$  in the channel identification problem. Subsequently, the equivalent estimation error at the  $(\ell - 1)$ th stage is defined as:

$$e_{\ell-1,n}^{eq} = d(n) - \hat{d}_{\ell-1,n}^{eq}, \quad (20)$$

in order to achieve the order-recursive expression for the equivalent estimation error as:

$$e_{\ell,n}^{eq} = e_{\ell-1,n}^{eq} - v^m(n) \Delta_{\ell,m}^* \Delta_{\ell-1}^m(n). \quad (21)$$

It can be similarly shown that the equivalent estimation error,  $e_{\ell,n}^{eq}(n)$ , at the output of the  $\ell$ th stage can also be expressed in terms of estimation errors related to the channels of SPMLSs as:

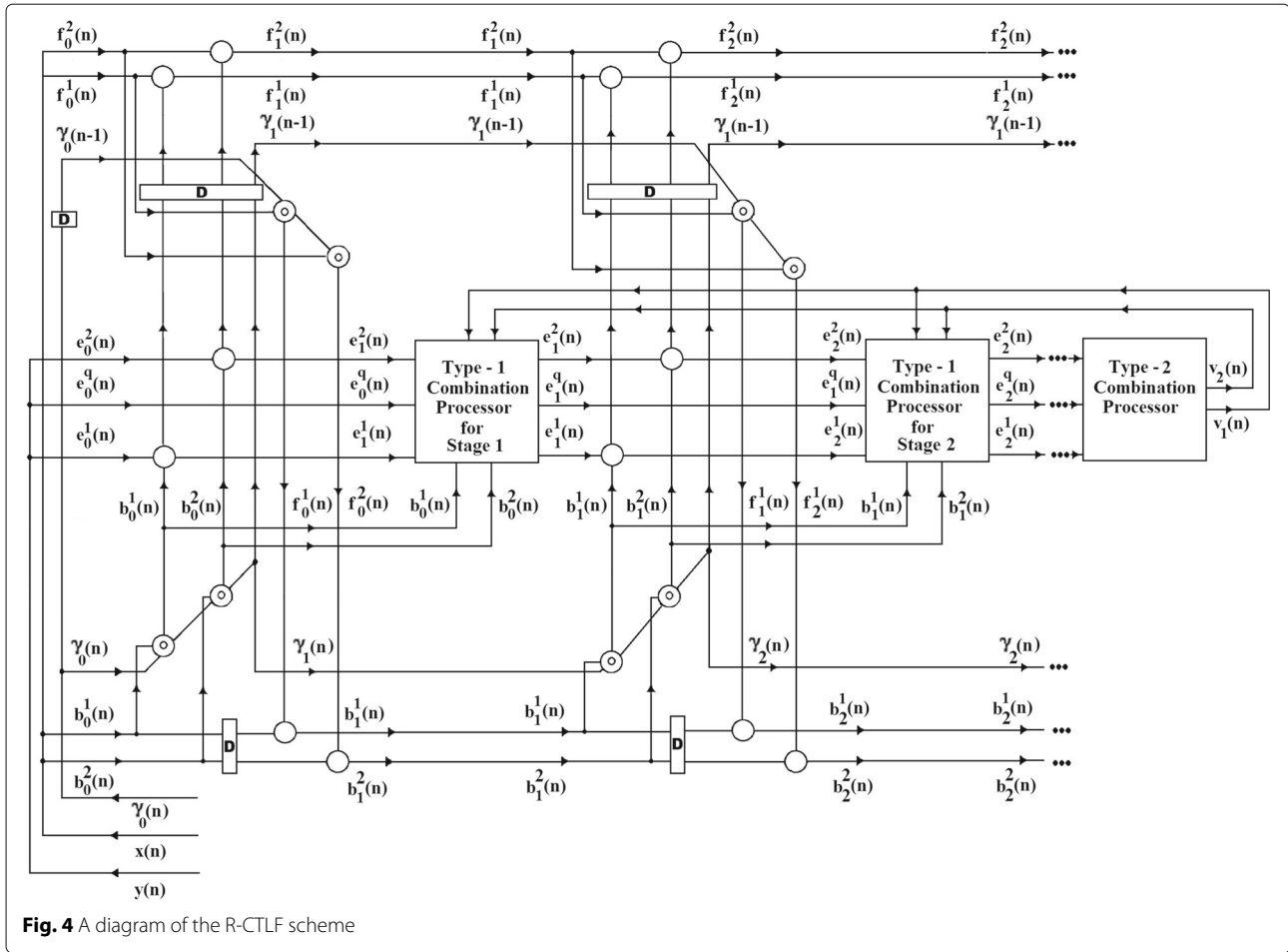
$$e_{\ell,n}^{eq} = \sum_{k=1}^M v^k(n) e_{\ell,n}^k, \quad (22)$$

where  $e_{\ell,n}^k(n)$  is the  $k$ th estimation error of the  $\ell$ th stage. The corresponding equivalent reflection coefficient at the  $\ell$ th stage can be computed using mixing and reflection coefficients at time instant  $n$  as:

$$\Delta_{\ell}^{eq}(n) = \sum_{k=1}^M v^k(n) \Delta_{\ell,k}(n). \quad (23)$$

Hence, Eqs. (17), (21), or (22) and (23) are implemented in type-1 combination processors shown in Fig. 4.





The  $m$ th mixing coefficient at time instant  $n + 1$  is computed at the output of the last lattice stage in a type-2 combination processor as follows:

$$v^m(n + 1) = \frac{\exp(a^m(n + 1))}{\beta(n + 1)}, \tag{24}$$

where  $m = 1, 2, \dots, M$ , and  $a^m(n + 1)$  is the  $m$ th combination parameter at time instant  $n + 1$ . Herein, the normalization parameter at time instant  $n + 1$ ,  $\beta(n + 1)$ , is computed according to:

$$\beta(n + 1) = \sum_{k=1}^M \exp(a^k(n + 1)). \tag{25}$$

Note that  $0 < v^m(n) < 1, \forall m$ , and  $\sum_{k=1}^M v^k(n) = 1$ .

The time-update equation for the  $m$ th combination parameter,  $a^m(n)$ , can be accordingly expressed as:

$$a^m(n + 1) = a^m(n) - \frac{\mu_a}{2} \frac{\partial e_{N,n}^{eq}(n)^2}{\partial a^m(n)}, \tag{26}$$

in which  $\mu_a$  is the step size. The derivation in (26) is carried out so as to obtain the following expression:

$$a^m(n + 1) = a^m(n) - \mu_a e_{N,n}^{eq}(n) \frac{\partial e_{N,n}^{eq}(n)}{\partial a^m(n)}. \tag{27}$$

Equation (22) for  $\ell = N$  is subsequently utilized in evaluating  $\frac{\partial e_{N,n}^{eq}(n)}{\partial a^m(n)}$ , and Eq. (27) is expressed as in the following statement:

$$a^m(n + 1) = a^m(n) - \mu_a e_{N,n}^{eq}(n) \sum_{k=1}^M \frac{\partial v^k(n)}{\partial a^m(n)} e_{N,n}^k(n), \tag{28}$$

where  $m = 1, \dots, M$ . The partial derivatives of  $v^k(n)$  with respect to  $a^m(n)$  are stated as follows:

$$\begin{aligned} \frac{\partial v^m(n)}{\partial a^m(n)} &= v^m(n) - v^m(n)^2, \quad k = m \\ \frac{\partial v^k(n)}{\partial a^m(n)} &= -v^k(n) v^m(n), \quad k \neq m. \end{aligned} \tag{29}$$

The expressions in Eq. (29) corresponding to the partial derivatives for  $k = m$  and  $k \neq m$  cases are substituted

in Eq. (28) to attain the following statement for the time-update equation of the  $m$ th combination parameter:

$$a^m(n+1) = a^m(n) - \mu_a e_{N,n}^{eq}(n) \left[ \begin{aligned} & (v^m(n) - v^m(n)^2) e_{N,n}^m(n) \\ & - v^m(n) \sum_{k \neq m} v^k(n) e_{N,n}^k(n) \end{aligned} \right]. \quad (30)$$

Thereafter, Eq. (22) for  $\ell = N$  is once more used to find an equivalent expression for the summation term,  $\sum_{k \neq m} v^k(n) e_{N,n}^k(n)$ , in Eq. (30) as follows:

$$e_{N,n}^{eq}(n) - e_{N,n}^m(n) = \sum_{k \neq m} v^k(n) e_{N,n}^k(n), \quad (31)$$

which is then substituted back in Eq. (30), and the final expression for the time-update of the  $m$ th combination parameter in terms of the index  $m$  is attained as:

$$a^m(n+1) = a^m(n) - \mu_a e_{N,n}^{eq}(n) (e_{N,n}^m(n) - e_{N,n}^{eq}(n)) v^m(n), \quad (32)$$

where  $m = 1, \dots, M$ . The term  $(e_{N,n}^m(n) - e_{N,n}^{eq}(n))$  in Eq. (32) can give rise to a slowing down effect in the learning of combination parameters that usually occurs during long stationary intervals during which the estimation errors,  $e_{N,n}^m(n)$  and  $e_{N,n}^{eq}(n)$ , are close. In order to alleviate this problem, a momentum term can be appended to the statement in Eq. (32) as in the following :

$$a^m(n+1) = a^m(n) - \mu_a e_{N,n}^{eq}(n) (e_{N,n}^m(n) - e_{N,n}^{eq}(n)) v^m(n) + \rho (a^m(n) - a^m(n-1)), \quad (33)$$

where  $0 < \rho < 1$  [49]. Accordingly, the new additive term in Eq. (33) compensates the pernicious effect related to the second term.

The mixing coefficients,  $v^m(n)$ , are then fed back to type-1 combination processors so as to be used in the computation of equivalent desired signals, estimation errors, and equivalent reflection coefficients in Eqs. (17), (21), or (22) and (23) respectively. We call the complete algorithm as the R-CMLF algorithm, which includes the modified SPMLS algorithm in Table 2 as well as the combination algorithm presented in this subsection, and summarize it in Table 3.

It is also possible to speed up the convergence of the slower component filters by transferring a part of the equivalent reflection coefficients to the reflection coefficients of the component filters that perform significantly worse than the combined scheme. To accomplish this objective, Eq. (T.3.9) in Table 3 regarding the computation of the  $m$ th joint state estimation lattice reflection coefficient at the  $\ell$ th stage is modified by incorporating the transfer parameter  $\alpha$  and the equivalent reflection

coefficient  $\Delta_\ell^{eq}(n-1)$  at the  $\ell$ th stage in the following manner:

$$\Delta_{\ell,m}^e(n) = \alpha (\Delta_{\ell,m}^e(n-1) + \gamma_{\ell-1,m}^b(n) e_{\ell-1,m}^{m*}(n) b_{\ell-1}^m(n) / r_{\ell-1,m}^b(n)) + (1-\alpha) \Delta_\ell^{eq}(n-1). \quad (34)$$

The permissible range for the transfer parameter is  $0 < \alpha < 1$ , and the transfer of reflection coefficients is only applied when the filtered quadratic estimation errors meet the conditions discussed in [49] as indicators of worse performance.

#### 4.2 Decoupled combination of multiple lattice filters

A diagram of the D-CMLF scheme is presented for  $M = 2$  case, which can be named as decoupled combination of two lattice filters (D-CTLF), in Fig. 5. A type-3 combination processor in this case is inserted to each lattice stage. In the sequel, we develop the combination algorithm that will be implemented in a type-3 processor.

In order to compute the estimate of equivalent desired signal at the output of the  $\ell$ th lattice stage in an order-recursive manner, it is first stated as follows:

$$\hat{d}_{\ell,n}^{eq}(n) = \sum_{j=1}^{\ell} \sum_{k=1}^M v_j^k(n) \Delta_{j,k}^*(n-1) b_{j-1}^k(n), \quad (35)$$

where  $v_j^k(n)$  and  $\Delta_{k,j}(n)$  are the  $k$ th mixing and estimation error reflection coefficients of the  $j$ th lattice stage at time instant  $n$  respectively.  $b_{j-1}^k(n)$  is the  $k$ th backward prediction error at the entrance of the  $j$ th stage at time instant  $i$ . The estimate of equivalent desired signal can be expressed order-recursively as follows :

$$\hat{d}_{\ell,n}^{eq}(n) = \hat{d}_{\ell-1,n}^{eq}(n) + v_\ell^m(n) \Delta_{\ell,m}^*(n-1) b_{\ell-1}^m(n), \quad (36)$$

for  $\ell = 1, \dots, N$ ,  $m = 1, \dots, M$ , and  $\hat{d}_{0,n}^{eq}(n) = 0$ . Note the mixing coefficients are related to the  $\ell$ th stage in this case.

Subsequently, the equivalent estimation error at the output of the  $\ell$ th stage is defined as :

$$e_{\ell,n}^{eq}(n) = d(n) - \hat{d}_{\ell,n}^{eq}(n), \quad (37)$$

where  $d(n)$  represents the desired signal at time instant  $n$  as in the previous subsection, and Eq. (36) is substituted in this equivalent estimation error expression to obtain the following statement:

$$e_{\ell,n}^{eq}(n) = d(n) - \hat{d}_{\ell-1,n}^{eq}(n) - v_\ell^m(n) \Delta_{\ell,m}^*(n-1) b_{\ell-1}^m(n). \quad (38)$$

The equivalent estimation error for the  $(\ell - 1)$ th stage is similarly defined as :

$$e_{\ell-1,n}^{eq}(n) = d(n) - \hat{d}_{\ell-1,n}^{eq}(n), \quad (39)$$

and thereby the order-recursive equivalent estimation error is expressed as:

$$e_{\ell,n}^{eq}(n) = e_{\ell-1,n}^{eq}(n) - v_\ell^m(n) \Delta_{\ell,m}^*(n-1) b_{\ell-1}^m(n). \quad (40)$$

**Table 3** The R-CMLF algorithm

Stage inputs

$$\gamma_{\ell-1,1}^f(n) = \gamma_{\ell-1}(n-1), \gamma_{\ell-1,1}^b(n) = \gamma_{\ell-1}(n) \tag{T.3.1}$$

$$r_{\ell-1,k}^b(-1) = r_{\ell-1,k}^f(-1) = \delta, (k = 1, \dots, M) \tag{T.3.2}$$

$$\Delta_{\ell,k}^e(-1) = \Delta_{\ell,k}^f(-1) = \Delta_{\ell,k}^b(-1) = \Delta_{\ell}^{eq}(0) = 0.0, (k = 1, \dots, M) \tag{T.3.3}$$

**For k=1, ..., M**

Computations at SOPs

$$r_{\ell-1,k}^b(n) = \lambda_k r_{\ell-1,k}^b(n-1) + \gamma_{\ell-1,k}^b(n) |b_{\ell-1}^k(n)|^2 \tag{T.3.4}$$

$$\gamma_{\ell-1,k+1}^b(n) = \gamma_{\ell-1,k}^b(n) - |\gamma_{\ell-1,k}^b(n)|^2 |b_{\ell-1}^k(n)|^2 / r_{\ell-1,k}^b(n) \tag{T.3.5}$$

$$r_{\ell-1,k}^f(n) = \lambda_k r_{\ell-1,k}^f(n-1) + \gamma_{\ell-1,k}^f(n) |f_{\ell-1}^k(n)|^2 \tag{T.3.6}$$

$$\gamma_{\ell-1,k+1}^f(n) = \gamma_{\ell-1,k}^f(n) - |\gamma_{\ell-1,k}^f(n)|^2 |f_{\ell-1}^k(n)|^2 / r_{\ell-1,k}^f(n) \tag{T.3.7}$$

Joint process estimation (ROP)

$$e_{\ell}^k(n) = e_{\ell-1}^k(n) - \Delta_{\ell,k}^{e*}(n-1) b_{\ell-1}^k(n) \tag{T.3.8}$$

$$\Delta_{\ell,k}^e(n) = \alpha (\Delta_{\ell,k}^e(n-1) + \gamma_{\ell-1,k}^b(n) e_{\ell}^{k*}(n) b_{\ell-1}^k(n) / r_{\ell-1,k}^b(n)) + (1 - \alpha) \Delta_{\ell}^{eq}(n-1) \tag{T.3.9}$$

Combination processing (type-1 processor)

$$e_{\ell}^{eq}(n) = e_{\ell}^{eq}(n) + v^k(n) e_{\ell}^k(n) \tag{T.3.10}$$

$$\Delta_{\ell}^{eq}(n) = \Delta_{\ell}^{eq}(n) + v^k(n) \Delta_{\ell,k}^e(n) \tag{T.3.11}$$

Forward error prediction (ROP)

$$f_{\ell}^k(n) = f_{\ell-1}^k(n) - \Delta_{\ell,k}^{f*}(n-1) b_{\ell-1}^k(n-1) \tag{T.3.12}$$

$$\Delta_{\ell,k}^f(n) = \Delta_{\ell,k}^f(n-1) + \gamma_{\ell-1,k}^b(n-1) f_{\ell}^{k*}(n) b_{\ell-1}^k(n-1) / r_{\ell-1,k}^b(n-1) \tag{T.3.13}$$

Backward error prediction (ROP)

$$b_{\ell}^k(n) = b_{\ell-1}^k(i-1) - \Delta_{\ell,k}^{b*}(n-1) f_{\ell-1}^k(n) \tag{T.3.14}$$

$$\Delta_{\ell,k}^b(n) = \Delta_{\ell,k}^b(n-1) + \gamma_{\ell-1,k}^f(n) b_{\ell}^{k*}(n) f_{\ell-1}^k(n) / r_{\ell-1,k}^f(n) \tag{T.3.15}$$

**End**

Combination processing (type-2 processor)

**For k=1, ..., M**

$$a^k(n+1) = a^k(n) - \mu_{\alpha} e_N^{eq}(n) (e_N^k(n) - e_N^{eq}(n)) v^k(n) + \rho (a^k(n) - a^k(n-1)) \tag{T.3.16}$$

$$\beta(n+1) = \beta(n+1) + e^{a^k(n+1)} \tag{T.3.17}$$

$$v^k(n+1) = \frac{e^{a^k(n+1)}}{\beta(n+1)} \tag{T.3.18}$$

**End**

Stage outputs

$$\gamma_{\ell}(n) = \gamma_{\ell-1,M+1}^b(n) \tag{T.3.19}$$

It can be shown that the equivalent estimation error,  $e_{\ell,n}^{eq}(n)$ , at the output of the  $\ell$ th stage can also be expressed in terms of estimation errors related to the channels of SPMLSs as:

$$e_{\ell,n}^{eq}(n) = \sum_{k=1}^M v_{\ell}^k(n) e_{\ell,n}^k(n), \tag{41}$$

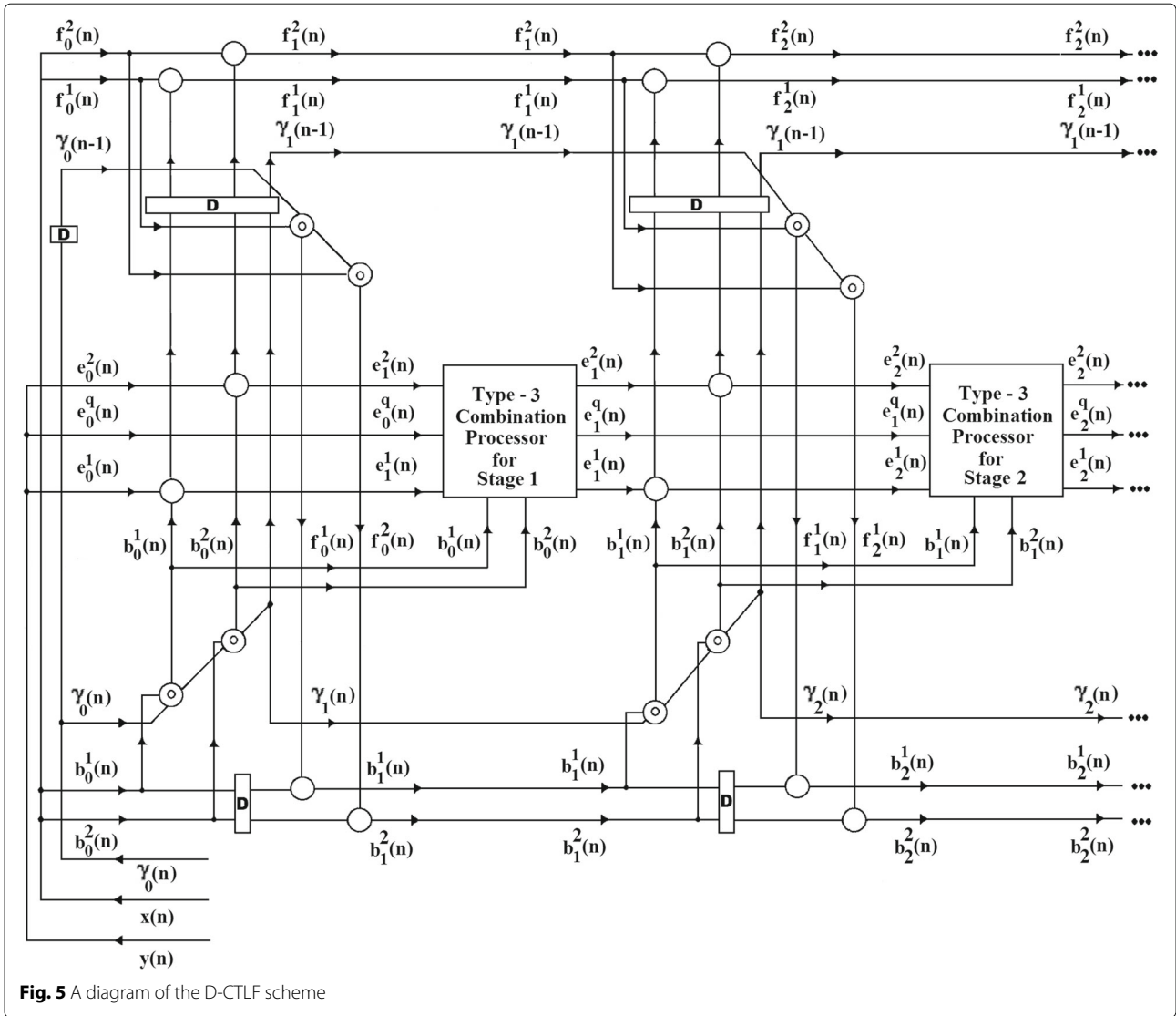
where  $e_{\ell,n}^k(n)$  is the  $k$ th estimation error of the  $\ell$ th stage. Similarly, the equivalent reflection coefficient at the  $\ell$ th stage is stated in accordance with :

$$\Delta_{\ell}^{eq}(n) = \sum_{k=1}^M v_{\ell}^k(n) \Delta_{\ell,k}^e(n). \tag{42}$$

Hence, the definition of mixing coefficient for the  $m$ th channel of the  $\ell$ th stage is stated as :

$$v_{\ell}^m(n+1) = \frac{\exp(a_{\ell}^m(n+1))}{\beta_{\ell}(n+1)}. \tag{43}$$

Herein,  $a_{\ell}^m(n+1)$  and  $\beta_{\ell}(n+1)$  are the  $m$ th combination parameter and the normalization factor at time instant



$n + 1$  for the  $\ell$ th stage respectively. The normalization factor is defined as:

$$\beta_\ell(n + 1) = \sum_{k=1}^M \exp(a_\ell^k(n + 1)). \quad (44)$$

Note that  $0 < v_\ell^m(n) < 1, \forall m$  and  $\ell$ , and  $\sum_{k=1}^M v_\ell^k(n) = 1$ .

It follows that the time-update equation for the  $m$ th combination parameter at the  $\ell$ th stage,  $a_\ell^m(n)$ , can be stated by making use of the gradient descent method as:

$$a_\ell^m(n + 1) = a_\ell^m(n) - \frac{\mu_a}{2} \frac{\partial e_{\ell,n}^{eq}(n)^2}{\partial a_\ell^m(n)}, \quad (45)$$

in which  $\mu_a$  is the step size. The derivation in (45) is carried out so as to obtain the following expression:

$$a_\ell^m(n + 1) = a_\ell^m(n) - \mu_a e_{\ell,n}^{eq}(n) \frac{\partial e_{\ell,n}^{eq}(n)}{\partial a_\ell^m(n)}. \quad (46)$$

Equation (41) is subsequently utilized in evaluating  $\frac{\partial e_{\ell,n}^{eq}(n)}{\partial a_\ell^m(n)}$ , and Eq. (46) is expressed as in the following statement:

$$a_\ell^m(n + 1) = a_\ell^m(n) - \mu_a e_{\ell,n}^{eq}(n) \sum_{k=1}^M \frac{\partial v_\ell^k(n)}{\partial a_\ell^m(n)} e_{\ell,n}^k(n), \quad (47)$$

where  $m = 1, \dots, M$  and  $\ell = 1, \dots, N$ . The partial derivatives of  $v_\ell^k(n)$  with respect to  $a_\ell^m(n)$  are found as follows:

$$\begin{aligned} \frac{\partial v_\ell^m(n)}{\partial a_\ell^m(n)} &= v_\ell^m(n) - v_\ell^m(n)^2, \quad k = m \\ \frac{\partial v_\ell^k(n)}{\partial a_\ell^m(n)} &= -v_\ell^k(n) v_\ell^m(n), \quad k \neq m. \end{aligned} \quad (48)$$

The expressions in Eq. (48) corresponding to the partial derivatives for  $k = m$  and  $k \neq m$  cases are substituted

in Eq. (47) to obtain the following statement for the time-update equation of the  $m$ th combination parameter at the  $\ell$ th stage:

$$a_{\ell}^m(n+1) = a_{\ell}^m(n) - \mu_a e_{\ell,n}^{eq}(n) \left[ (v_{\ell}^m(n) - v_{\ell}^m(n)^2) e_{\ell,n}^m(n) - v_{\ell}^m(n) \sum_{k \neq m} v_{\ell}^k(n) e_{\ell,n}^k(n) \right]. \quad (49)$$

Afterwards, Eq. (41) is used to find an equivalent expression for the summation term,  $\sum_{k \neq m} v_{\ell}^k(n) e_{\ell,n}^k(n)$ , in Eq. (49) as follows:

$$e_{\ell,n}^{eq}(n) - e_{\ell,n}^m(n) = \sum_{k \neq m} v_{\ell}^k(n) e_{\ell,n}^k(n), \quad (50)$$

which is again used in Eq. (49) and the redundant term, and the final expression for the time-update of the  $m$ th combination parameter at the  $\ell$ th stage in terms of the indices  $m$  and  $\ell$  is given as:

$$a_{\ell}^m(n+1) = a_{\ell}^m(n) - \mu_a e_{\ell,n}^{eq}(n) (e_{\ell,n}^m(n) - e_{\ell,n}^{eq}(n)) v_{\ell}^m(n), \quad (51)$$

where  $\ell = 1, \dots, N$  and  $m = 1, \dots, M$ . Accordingly, the D-CMLF algorithm is presented in Table 4.

**Table 4** The D-CMLF algorithm

Stage inputs

$$\gamma_{\ell-1,1}^f(n) = \gamma_{\ell-1}(n-1), \gamma_{\ell-1,1}^b(n) = \gamma_{\ell-1}(n) \quad (T.4.1)$$

$$r_{\ell-1,k}^b(-1) = r_{\ell-1,k}^f(-1) = \delta, (k = 1, \dots, M) \quad (T.4.2)$$

$$\Delta_{\ell,k}^e(-1) = \Delta_{\ell,k}^f(-1) = \Delta_{\ell,k}^b(-1) = \Delta_{\ell}^{eq}(0) = 0.0, (k = 1, \dots, M) \quad (T.4.3)$$

**For  $k=1, \dots, M$**

Computations at SOPs

$$r_{\ell-1,k}^b(n) = \lambda_k r_{\ell-1,k}^b(n-1) + \gamma_{\ell-1,k}^b(n) |b_{\ell-1}^k(n)|^2 \quad (T.4.4)$$

$$\gamma_{\ell-1,k+1}^b(n) = \gamma_{\ell-1,k}^b(n) - |\gamma_{\ell-1,k}^b(n)|^2 |b_{\ell-1}^k(n)|^2 / r_{\ell-1,k}^b(n) \quad (T.4.5)$$

$$r_{\ell-1,k}^f(n) = \lambda_k r_{\ell-1,k}^f(n-1) + \gamma_{\ell-1,k}^f(n) |f_{\ell-1}^k(n)|^2 \quad (T.4.6)$$

$$\gamma_{\ell-1,k+1}^f(n) = \gamma_{\ell-1,k}^f(n) - |\gamma_{\ell-1,k}^f(n)|^2 |f_{\ell-1}^k(n)|^2 / r_{\ell-1,k}^f(n) \quad (T.4.7)$$

Joint process estimation (ROP)

$$e_{\ell}^k(n) = e_{\ell-1}^k(n) - \Delta_{\ell,k}^{e*}(n-1) b_{\ell-1}^k(n) \quad (T.4.8)$$

$$\Delta_{\ell,k}^e(n) = \alpha (\Delta_{\ell,k}^e(n-1) + \gamma_{\ell-1,k}^b(n) e_{\ell}^{k*}(n) b_{\ell-1}^k(n) / r_{\ell-1,k}^b(n)) + (1 - \alpha) \Delta_{\ell}^{eq}(n-1) \quad (T.4.9)$$

Combination processing (type-3 processor)

$$e_{\ell}^{eq}(n) = e_{\ell}^{eq}(n) + v_{\ell}^k(n) e_{\ell}^k(n) \quad (T.4.10)$$

$$\Delta_{\ell}^{eq}(n) = \Delta_{\ell}^{eq}(n) + v_{\ell}^k(n) \Delta_{\ell,k}^e(n) \quad (T.4.11)$$

$$a_{\ell}^k(n+1) = a_{\ell}^k(n) - \mu_a e_{\ell}^{eq}(n) (e_{\ell}^k(n) - e_{\ell}^{eq}(n)) v_{\ell}^k(n) + \rho (a_{\ell}^k(n) - a_{\ell}^k(n-1)) \quad (T.4.12)$$

$$\beta_{\ell}(n+1) = \beta_{\ell}(n+1) + e_{\ell}^{a^k(n+1)} \quad (T.4.13)$$

$$v_{\ell}^k(n+1) = \frac{e_{\ell}^{a^k(n+1)}}{\beta_{\ell}(n+1)} \quad (T.4.14)$$

Forward error prediction (ROP)

$$f_{\ell}^k(n) = f_{\ell-1}^k(n) - \Delta_{\ell,k}^{f*}(n-1) b_{\ell-1}^k(n-1) \quad (T.4.15)$$

$$\Delta_{\ell,k}^f(n) = \Delta_{\ell,k}^f(n-1) + \gamma_{\ell-1,k}^b(n-1) f_{\ell}^{k*}(n) b_{\ell-1}^k(n-1) / r_{\ell-1,k}^b(n-1) \quad (T.4.16)$$

Backward error prediction (ROP)

$$b_{\ell}^k(n) = b_{\ell-1}^k(n-1) - \Delta_{\ell,k}^{b*}(n-1) f_{\ell-1}^k(n) \quad (T.4.17)$$

$$\Delta_{\ell,k}^b(n) = \Delta_{\ell,k}^b(n-1) + \gamma_{\ell-1,k}^f(n) b_{\ell}^{k*}(n) f_{\ell-1}^k(n) / r_{\ell-1,k}^f(n) \quad (T.4.18)$$

**End**

Stage outputs

$$\gamma_{\ell}(n) = \gamma_{\ell-1,M+1}^b(n) \quad (T.4.19)$$

In order to avoid the slowing down of learning effect in long stationary intervals, the modification of the time-update equation can be carried out by adding a momentum term to the statement in Eq. (51) as in:

$$a_{\ell}^m(n+1) = a_{\ell}^m(n) - \mu_a e_{\ell,n}^{eq}(n) (e_{\ell,n}^m(n) - e_{\ell,n}^{eq}(n)) \times v_{\ell}^m(n) + \rho(a_{\ell}^m(n) - a_{\ell}^m(n-1)), \quad (52)$$

where  $0 < \rho < 1$  as before. To speed up the convergence of the slower component filters, Eq. (T.4.9) in Table 4, which is related to the computation of the  $m$ th joint state estimation lattice reflection coefficient at the  $\ell$ th stage, is modified as in Eq. (34).

### 5 Computational complexity

The computational complexity of the proposed schemes can be accordingly calculated by taking into consideration the effect of modifications on the complexity of a SPMLS and the added complexity due to combination processing in Eqs. (22) or (23), (24) and (25), and (32) in the R-CMLF scheme and Eqs. (41) or (42), (43) and (44), and (51) in the D-CMLF scheme. Note that computational complexity is expressed in terms of number of required operations, where one operation is considered as one multiplication (division) and one addition.

Due to the removal of all single circular cells in self-orthogonalizing processors and redundant single circular cells in referential-orthogonalizing processors of a SPMLS in combining multiple lattice filters, the complexity of a SPMLS reduces from  $(12M^2 + 7M)$  to  $(12M^2 + M)$ . Therefore, the total complexity for the R-CMLF scheme is  $(12NM^2 + 2MN + 3M + 1)$ , whereas it is  $(12NM^2 + 5MN + N)$  for the D-CMLF scheme. If the convergence of the slower component filters is required to be sped up by transferring a part of the equivalent reflection coefficients to the reflection coefficients of the component filters, the complexities of the proposed schemes increase due to

the transfer term in Eq. (34), and momentum terms in Eqs. (33) and (52), which together amount to an additional complexity of  $(2MN + 5M - N + 3)$ . Accordingly, the total complexities for the R-CMLF and D-CMLF schemes with transfer and momentum (t/m) terms become  $(12NM^2 + 4MN + 8M - N + 4)$  and  $(12NM^2 + 7MN + 5M + 3)$  respectively.

The computational complexities vs. filter length( $N$ ) curves of the R-CMLF and D-CMLF schemes for  $M = 2, 4,$  and  $8$  cases, that is, R-CTLF, R-CTLF with t/m terms, D-CTLF, D-CTLF with t/m terms, regular combination of four lattice filters (R-CFLF), R-CFLF with t/m terms, decoupled combination of four lattice filters (D-CFLF), D-CTLF with t/m terms, regular combination of eight lattice filters (R-CELF), R-CELF with t/m terms, decoupled combination of eight lattice filters (D-CELF), and D-CELF with t/m terms have been plotted in Figs. 6, 7, and 8 respectively.

We have also compared the complexities of the proposed methods with those of M-CLMS when  $M = 2, 4, 8,$  and D-CLMS schemes in [49]. Note that the complexities of M-CLMS and D-CLMS schemes are  $3MN + 5M + 1$  and  $10N + 3$  respectively and that these complexities also increase by an amount of  $2MN + 5M - N + 3$  when the transfer and momentum terms are implemented. In these figures, it can be seen that the complexities of the M-CLMS and D-CLMS schemes are advantageous comparing to the R-CMLF and D-CMLF schemes mainly due to the well known simplicity of LMS filters, and this advantage becomes larger with increasing number of combining filters ( $M$ ). However, it can be also noted that the complexities of transfer and momentum terms are comparable to those of the core M-CLMS and D-CLMS schemes, and therefore, the addition of transfer and momentum terms influences the complexities of M-CLMS and D-CLMS schemes more noticeably.

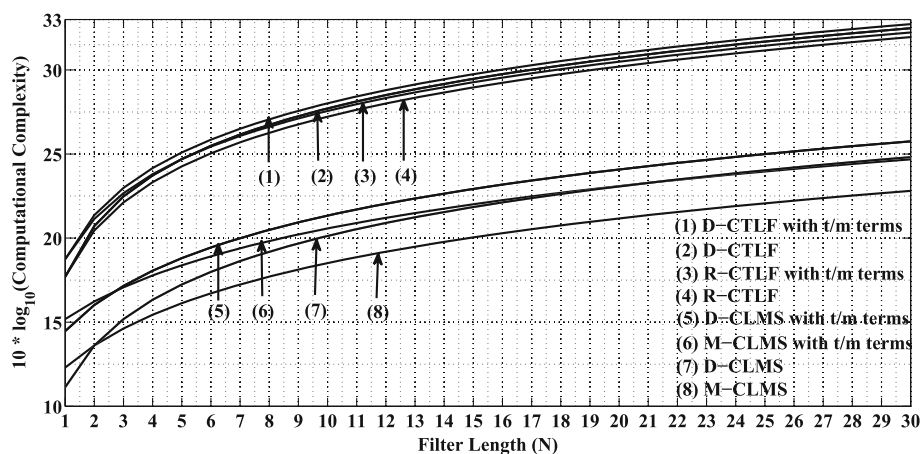


Fig. 6 Computational complexity comparison for  $M = 2$

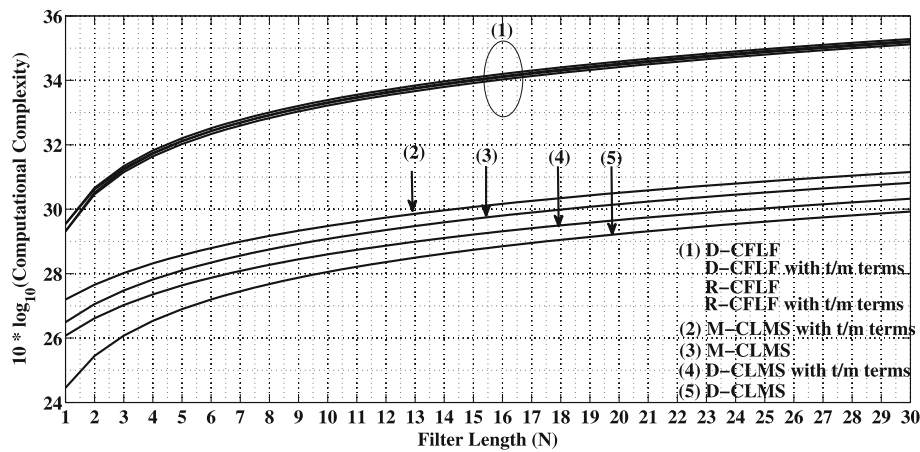


Fig. 7 Computational complexity comparison for  $M = 4$

In Fig. 9, the complexities of the proposed schemes with different values of combining filters are compared. It can be noticed that there is slight difference between the complexities of the R-CMLF and D-CMLF schemes, and this difference disappears with increasing  $M$ . For  $N = 30$ , the R-CTLF and D-CTLF schemes are approximately 3.9810 and 3.1622 times less complex than the R-CFLF and D-CFLF schemes respectively, whereas the R-CFLF and D-CFLF schemes are around 4.4668 less complex than the R-CELFF and D-CELFF schemes. Note that, when making the aforementioned comparison, the slight complexity differences between the R-CFLF and D-CFLF, and the R-CELFF and D-CELFF schemes respectively have been ignored. The computational complexity expressions of the proposed methods as well as those of the M-CLMS and D-CLMS schemes are summarized in Table 5.

### 6 Simulation study

As mentioned in Section 1, CR is an intelligent system that can adapt to statistical variations in the input stimuli in order to establish reliable communications. In this section, we consider two different simulation scenarios of channel identification in order to demonstrate that the proposed combination schemes can cope with statistical variations in the input stimuli better than component filters and that they can improve reliability in communications. Accordingly, the performances of the proposed schemes as well as component filters are presented in terms  $MSD(n)$  vs. number of iterations ( $n$ ) plots, where  $MSD(n)$  is defined as:

$$MSD(n) = \| \mathbf{w}(n) - \Delta(n) \|_2^2, \tag{53}$$

at time instant  $n$ . Herein,  $\mathbf{w}(n)$  and  $\Delta(n)$  represent the coefficients of the channel to be identified and

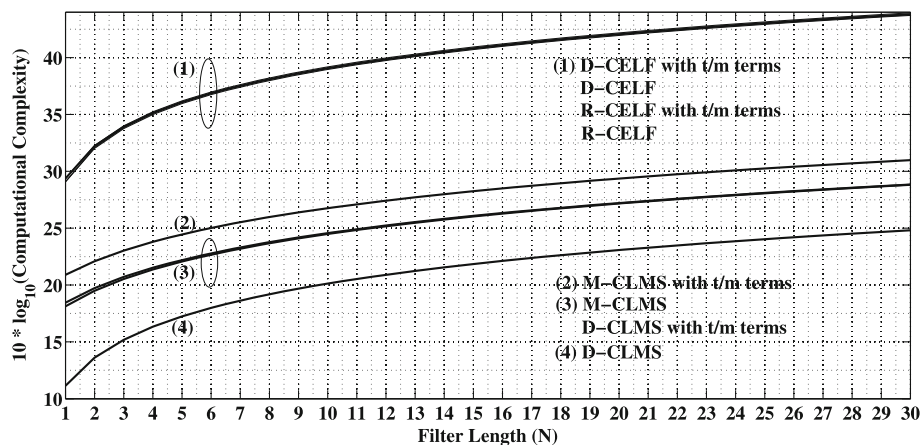


Fig. 8 Computational complexity comparison for  $M = 8$

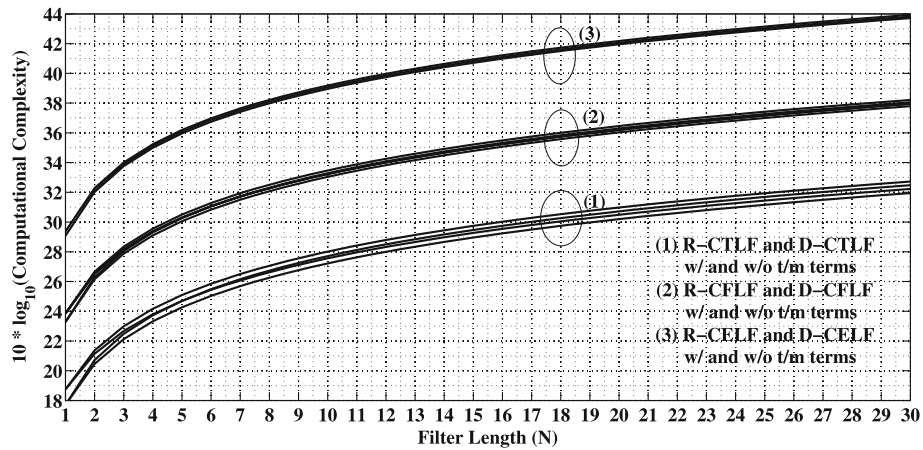


Fig. 9 Computational complexity comparison of the proposed schemes for different values of M

the corresponding lattice identification filter respectively. We also carried out the same experiments with the M-CLMS and D-CLMS schemes in [49] so as to provide comparison with the performances of the proposed methods.

Taking into account the model in Eq. (1), the channel to be identified in our experiments had 12 coefficients, and two cases for the input  $x(n)$  was considered: white and colored Gaussian noise input cases. The filter with the following input-output relationship was used to generate the input signal  $x(n)$  to the channel:

$$x(n) = \eta x(n - 1) + (\sqrt{1 - \eta^2}) v(n), \tag{54}$$

in which  $v(n)$  is a white Gaussian zero-mean noise process with unit variance. In the white Gaussian noise input case,  $\eta = 0.0$ , whereas in the colored Gaussian input case,  $\eta = 0.9$ . The channel noise,  $u(n)$ , is also a white zero-mean Gaussian noise with a variance of  $\sigma_u^2 = 0.01$  and is added to the channel output signal in all experiments.

The channel coefficients were changed in accordance with Eq. (2). Under stationary operating conditions,

$\mathbf{q}(n) = 0, \forall n$ , and under nonstationary operating conditions,  $\mathbf{q}(n)$  represents an identically and independently Gaussian distributed random zero-mean vector with diagonal covariance matrix  $\mathbf{Q}(n)$  as was introduced in Section 2. Under both stationary and nonstationary operating conditions, the initial values of the channel coefficients were zero-mean Gaussian distributed with a variance of  $\sigma_w^2 = 0.1$ , so that they were between  $-1$  and  $1$ . These initial values were kept constant under stationary operating conditions, whereas they were allowed to change in accordance with Eq. (2) under nonstationary operating conditions.

In order to imbibe the channel output signal with alternating slow and fast statistical variations in accordance with Eq. (3) under nonstationary operating conditions, the trace of  $\mathbf{Q}(n)$  was selected so as to take turns between two different values. Particularly, it was  $12 \times 10^{-6}$  during the following number of iterations:  $0 \leq n \leq 1500$ ,  $3500 \leq n \leq 5500$ , and  $7500 \leq n \leq 9000$ , whereas it was  $12 \times 10^{-2}$  for  $1501 \leq n \leq 3499$ ,  $5501 \leq n \leq 7499$  with the values of the degree of nonstationarity alternating between  $\xi(n) \approx 0.0345$  and  $\xi(n) \approx 3.465$  respectively.

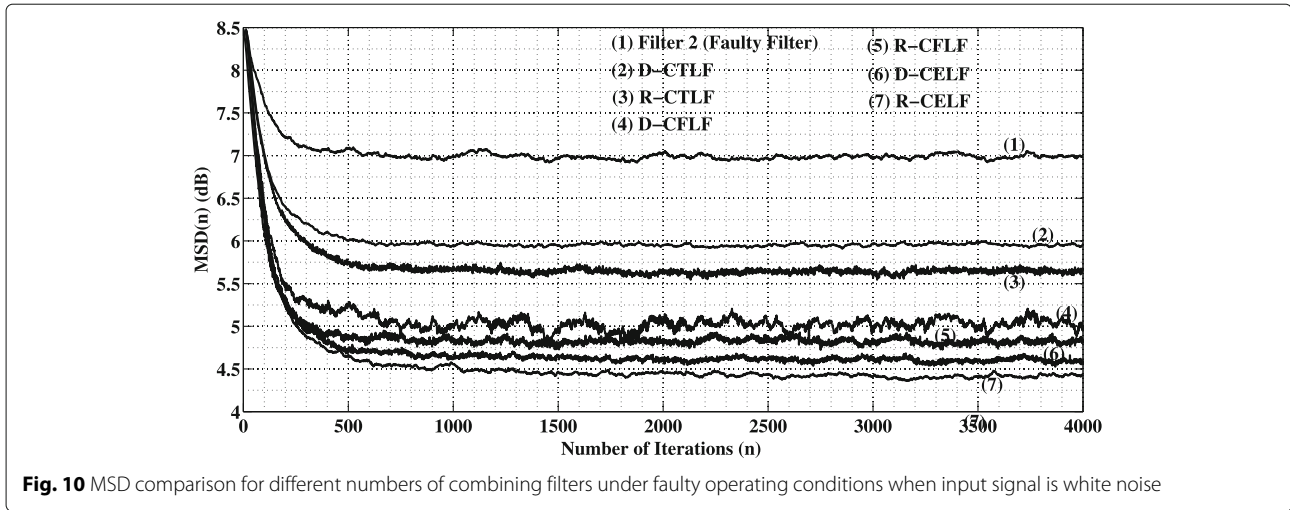
We considered the combination of four filters in the simulation of the M-CLMS scheme. The following step sizes for the M-CLMS and D-CLMS schemes were used respectively:  $\mu_1 = 0.005, \mu_2 = 0.01, \mu_3 = 0.02, \mu_4 = 0.03$ , and  $\mu_1 = 0.005$  and  $\mu_2 = 0.03$ . All experimental results are ensemble averages of 100 independent runs. The regularization parameter  $\delta$  for the lattice filters was set to 1.0.

During the simulations, we did not allow the mixing coefficients, i.e.,  $v^m$  in the R-CMLF and M-CLMS schemes or  $v_\ell^m$  in the D-CMLF and D-CLMS schemes, to increase above a threshold value of  $\epsilon$  in order to avoid the other mixing coefficients getting too close to 0, which can stop

Table 5 Computational complexity comparison

|                       |                             |
|-----------------------|-----------------------------|
| R-CMLF                | $12NM^2 + 2MN + 3M + 1$     |
| R-CMLF with t/m terms | $12NM^2 + 4MN + 8M - N + 4$ |
| D-CMLF                | $12NM^2 + 5MN + N$          |
| D-CMLF with t/m terms | $12NM^2 + 7MN + 5M + 3$     |
| M-CLMS                | $3MN + 5M + 1$              |
| M-CLMS with t/m terms | $5MN + 10M - N + 4$         |
| D-CLMS                | $10N + 3$                   |
| D-CLMS with t/m terms | $2MN + 5M + 9N + 6$         |





**Fig. 10** MSD comparison for different numbers of combining filters under faulty operating conditions when input signal is white noise

the corresponding learning. Accordingly, it can be shown that  $v^m < \epsilon$  in the R-CMLF and M-CLMS schemes or  $v_e^m < \epsilon$  in the D-CMLF and D-CLMS schemes are satisfied if  $|a^m| \leq 0.5 \log_2((\epsilon(M-1)/(1-\epsilon))) = \epsilon'$  or  $|a_e^m| \leq 0.5 \log_2((\epsilon(M-1)/(1-\epsilon))) = \epsilon'$  [49]. We used the following  $\epsilon$  values:  $\epsilon = 1 - 0.09(M-1)$  under stationary operating conditions with white and colored noise input as well as nonstationary operating conditions with colored noise input, and  $\epsilon = 1 - 0.001(M-1)$  under nonstationary operating conditions with white noise input. We also implemented  $\mu_a = 100$  in order to adapt the combination parameters in Eqs. (32), (33), (51), and (52).

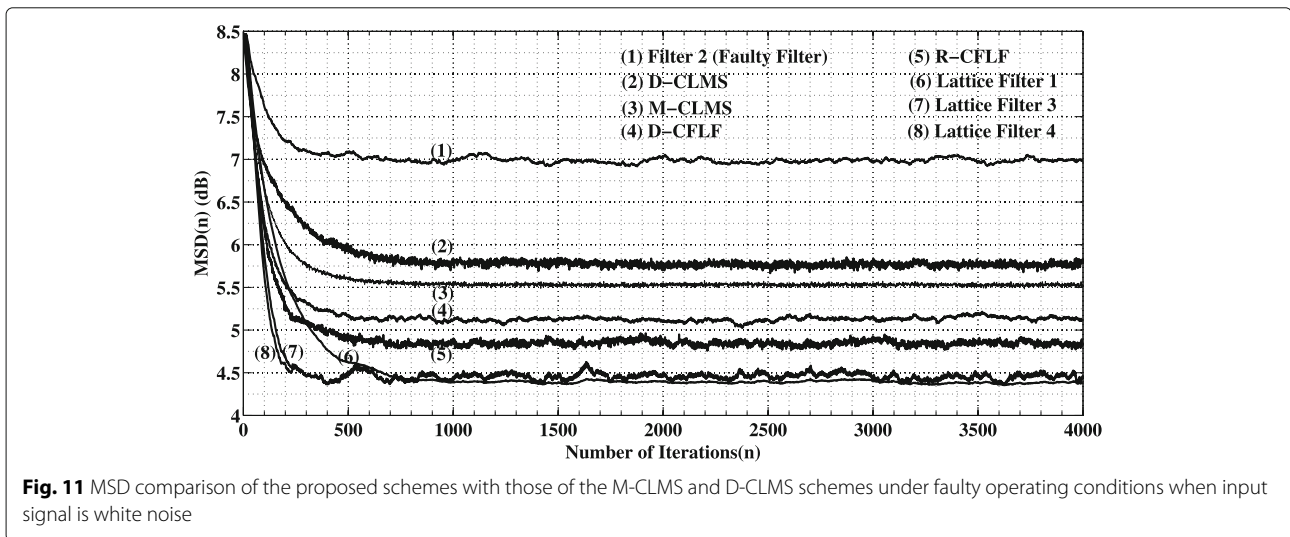
### 6.1 Stationary operating conditions

The effect of faulty elements on the MSD performance of component and combination filters under stationary operating conditions was investigated so as to demonstrate the intelligence gained in the form of fault tolerance

improvement with combination processing. Accordingly, the number of coefficients of filter 2 was reduced from 12 to 6 while keeping the number coefficients of the other filters at 12 in the proposed schemes. We considered combinations of 2, 4, and 8 filters using the D-CMLF and R-CMLF schemes, viz., R-CTLF and D-CTLF, R-CFLF and D-CFLF, and R-CELF and D-CELF schemes. The exponential weighting factors were the following:  $\lambda_1 = 0.995$  and  $\lambda_2 = 0.97$  for the R-CTLF and D-CTLF schemes;  $\lambda_1 = 0.995$ ,  $\lambda_2 = 0.99$ ,  $\lambda_3 = 0.98$ , and  $\lambda_4 = 0.97$  for the R-CFLF and D-CFLF schemes; and  $\lambda_1 = 0.9995$ ,  $\lambda_2 = 0.999$ ,  $\lambda_3 = 0.995$ ,  $\lambda_4 = 0.99$ ,  $\lambda_5 = 0.985$ ,  $\lambda_6 = 0.98$ ,  $\lambda_7 = 0.975$ , and  $\lambda_8 = 0.97$  for the R-CELF and D-CELF schemes. The number of iterations were set as 4000.

#### 6.1.1 White input case

Figure 10 provides the MSD performance comparison of the R-CMLF and D-CMLF schemes with the performance



**Fig. 11** MSD comparison of the proposed schemes with those of the M-CLMS and D-CLMS schemes under faulty operating conditions when input signal is white noise

of filter 2 (faulty filter) for different numbers of combining filters. It can be seen in this figure that the performance advantage of the proposed combination schemes improves with the increasing number of combining filters. Figure 11 compares the MSD performances of four component filters of the R-CFLF and D-CFLF schemes, the combination filters thereof as well as the performances related to the M-CLMS and D-CLMS schemes. Note that the number of coefficients of filter 2 of the M-CLMS and D-CLMS schemes were also reduced to 6 to model the faulty operation. It can be observed that the performance of faulty filter is almost 2.5 dB worse than that of the R-CFLF scheme, whereas it is about 2.0 dB worse than that of the D-CFLF scheme. In addition, the M-CLMS and D-CLMS schemes perform approximately 0.5 dB worse than the D-CFLF scheme.

The plots of mixing coefficients vs. number of iterations ( $n$ ) under stationary operating conditions were fluctuating with small variance around a constant value, and therefore, there was not much point in presenting the plots for all  $n$ , so that we provide their time-averaged values, i.e.,  $\bar{v}^m$  for the R-CFLF and M-CLMS schemes and  $\bar{v}_\ell^m$  for the D-CFLF and D-CLMS schemes in Table 6. Note that the R-CFLF and M-LMS ( $M = 4$ ) schemes have four mixing coefficients, whereas the D-CFLF and D-LMS schemes use 48 ( $M \times N = 4 \times 12$ ) and 24 ( $M \times N = 2 \times 12$ ) coefficients respectively. It can be noticed in Table 6 that filter 2 of the R-CFLF and M-CLMS schemes, which is faulty, contributes more than the other normal functioning three filters and that the contribution of the second filter is more in R-CFLF scheme than in M-CLMS scheme. On the other hand, it can also be seen in Table 6 that the contributions of all of the four filters are close till the eighth stage, after which the proportion of contribution for the second filter lessens comparing to the other three filters. Finally, in the case of D-CLMS scheme, it can be deduced that the proportion of contribution of two filters does not change much from stage to stage.

**6.1.2 Colored input case**

Figure 12 illustrates the effect of coloring the input on the MSD performance of proposed schemes under faulty operating conditions. Note that the parameter  $\eta$  of Eq. (54) controls the coloring of the input, so that  $\eta = 0.0$  and  $\eta = 0.9$  correspond to the white and colored input cases. It can be seen in Fig. 12 that, when the input is colored, the performances of R-CFLF and D-CFLF schemes degrade approximately 2.1 and 2.0 dB respectively comparing to the white input case.

**6.2 Nonstationary operating conditions**

The objective in this experiment is to display the advantage of combination processing in reacting to nonstationary operating conditions. The combinations of four

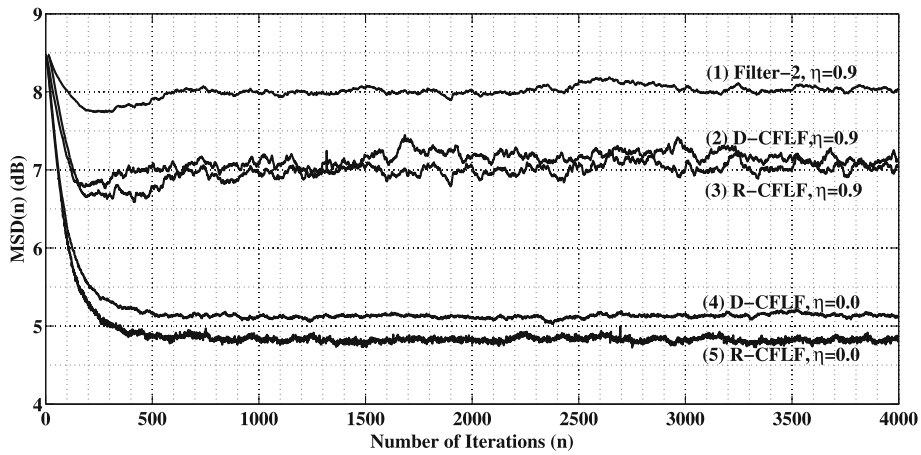
**Table 6** Comparison of time-averaged mixing coefficients under stationary conditions

|                             |                             |                             |                             |
|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
| R-CFLF                      |                             |                             |                             |
| $\bar{v}^1 = 0.219662$      | $\bar{v}^2 = 0.329039$      | $\bar{v}^3 = 0.224948$      | $\bar{v}^4 = 0.226642$      |
| D-CFLF                      |                             |                             |                             |
| $\bar{v}_1^1 = 0.250000$    | $\bar{v}_1^2 = 0.250000$    | $\bar{v}_1^3 = 0.250000$    | $\bar{v}_1^4 = 0.250000$    |
| $\bar{v}_2^1 = 0.260383$    | $\bar{v}_2^2 = 0.245273$    | $\bar{v}_2^3 = 0.243011$    | $\bar{v}_2^4 = 0.248331$    |
| $\bar{v}_3^1 = 0.260950$    | $\bar{v}_3^2 = 0.238969$    | $\bar{v}_3^3 = 0.240371$    | $\bar{v}_3^4 = 0.256708$    |
| $\bar{v}_4^1 = 0.259955$    | $\bar{v}_4^2 = 0.238969$    | $\bar{v}_4^3 = 0.240596$    | $\bar{v}_4^4 = 0.259483$    |
| $\bar{v}_5^1 = 0.259482$    | $\bar{v}_5^2 = 0.235899$    | $\bar{v}_5^3 = 0.240934$    | $\bar{v}_5^4 = 0.260747$    |
| $\bar{v}_6^1 = 0.258075$    | $\bar{v}_6^2 = 0.235125$    | $\bar{v}_6^3 = 0.241462$    | $\bar{v}_6^4 = 0.262336$    |
| $\bar{v}_7^1 = 0.256902$    | $\bar{v}_7^2 = 0.234466$    | $\bar{v}_7^3 = 0.242221$    | $\bar{v}_7^4 = 0.263409$    |
| $\bar{v}_8^1 = 0.307697$    | $\bar{v}_8^2 = 0.137990$    | $\bar{v}_8^3 = 0.267800$    | $\bar{v}_8^4 = 0.283511$    |
| $\bar{v}_9^1 = 0.300305$    | $\bar{v}_9^2 = 0.146743$    | $\bar{v}_9^3 = 0.267666$    | $\bar{v}_9^4 = 0.282285$    |
| $\bar{v}_{10}^1 = 0.294155$ | $\bar{v}_{10}^2 = 0.155295$ | $\bar{v}_{10}^3 = 0.267576$ | $\bar{v}_{10}^4 = 0.279972$ |
| $\bar{v}_{11}^1 = 0.287647$ | $\bar{v}_{11}^2 = 0.167800$ | $\bar{v}_{11}^3 = 0.266095$ | $\bar{v}_{11}^4 = 0.275457$ |
| $\bar{v}_{12}^1 = 0.283007$ | $\bar{v}_{12}^2 = 0.180514$ | $\bar{v}_{12}^3 = 0.263361$ | $\bar{v}_{12}^4 = 0.270116$ |
| M-CLMS                      |                             |                             |                             |
| $\bar{v}^1 = 0.194679$      | $\bar{v}^2 = 0.430737$      | $\bar{v}^3 = 0.185233$      | $\bar{v}^4 = 0.186599$      |
| D-CLMS                      |                             |                             |                             |
| $\bar{v}_1^1 = 0.439073$    | $\bar{v}_1^2 = 0.560926$    |                             |                             |
| $\bar{v}_2^1 = 0.439087$    | $\bar{v}_2^2 = 0.560991$    |                             |                             |
| $\bar{v}_3^1 = 0.438953$    | $\bar{v}_3^2 = 0.561046$    |                             |                             |
| $\bar{v}_4^1 = 0.438959$    | $\bar{v}_4^2 = 0.560403$    |                             |                             |
| $\bar{v}_5^1 = 0.438893$    | $\bar{v}_5^2 = 0.561116$    |                             |                             |
| $\bar{v}_6^1 = 0.439248$    | $\bar{v}_6^2 = 0.560751$    |                             |                             |
| $\bar{v}_7^1 = 0.437971$    | $\bar{v}_7^2 = 0.562028$    |                             |                             |
| $\bar{v}_8^1 = 0.436686$    | $\bar{v}_8^2 = 0.563313$    |                             |                             |
| $\bar{v}_9^1 = 0.437216$    | $\bar{v}_9^2 = 0.562783$    |                             |                             |
| $\bar{v}_{10}^1 = 0.436428$ | $\bar{v}_{10}^2 = 0.563471$ |                             |                             |
| $\bar{v}_{11}^1 = 0.437432$ | $\bar{v}_{11}^2 = 0.562567$ |                             |                             |
| $\bar{v}_{12}^1 = 0.436643$ | $\bar{v}_{12}^2 = 0.563356$ |                             |                             |

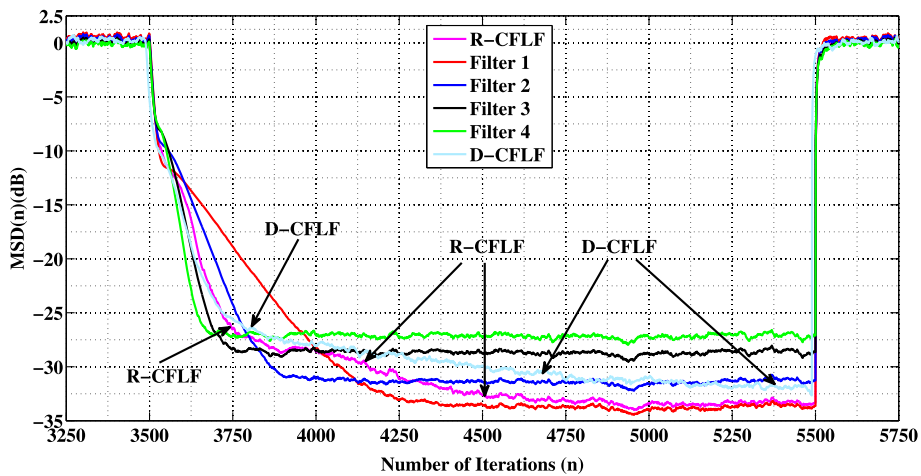
lattice filters, i.e., the R-CFLF and D-CFLF schemes, were considered in this case using the exponential weighting factors:  $\lambda_1 = 0.995$ ,  $\lambda_2 = 0.99$ ,  $\lambda_3 = 0.98$ , and  $\lambda_4 = 0.97$ .

**6.2.1 White input case**

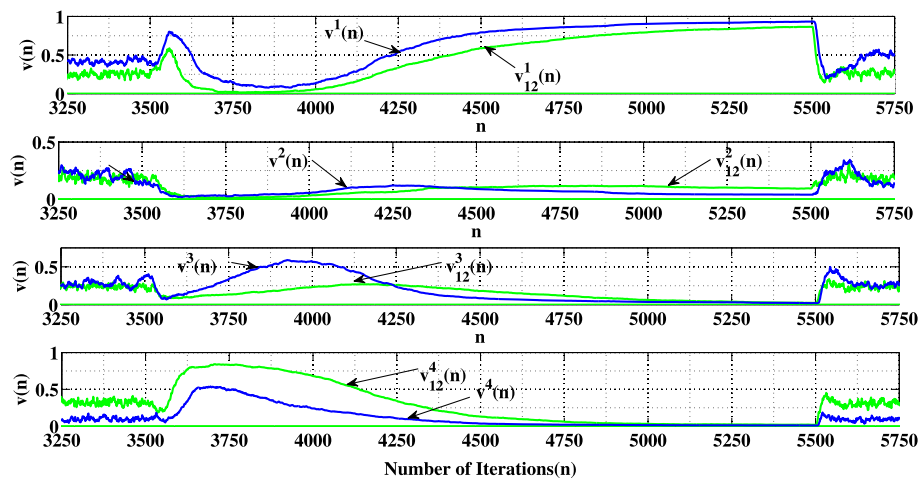
Figure 13 compares the MSD performances of combination filters related to the R-CFLF and D-CFLF schemes with those of the component filters under nonstationary operating conditions. The MSD performances were plotted for the number of iterations between 3250 and 6250 in order to better display the convergence and tracking behaviors of the filters. It can be seen in Fig. 13 that the



**Fig. 12** Effect of coloring the input on the MSD performance of the proposed schemes under faulty operating conditions



**Fig. 13** MSD comparison under nonstationary conditions when input signal is white noise



**Fig. 14** Mixing coefficients vs. number of iterations ( $n$ ) when input signal is white

component filters and the combination filters converge to different levels of MSD between  $-27.5$  and  $-34$  dB with different speeds in the slow statistical variation intervals ( $\xi(n) \approx 0.0345$ ), whereas all of the filters merge to almost the same level of MSD (0 dB) abruptly in the fast statistical variation intervals ( $\xi(n) \approx 3.4657$ ). It can also be observed in Fig. 13 that the component filters with smaller exponential weights converge to higher steady-state MSD levels, albeit faster, whereas the filters with larger exponential weights approach to lower steady-state MSD levels, although slowly. Accordingly, the combination filter brings together the desired features of component filters, that is, the fast convergence of filters that have smaller exponential weights with the low steady-state MSD performance of filters that have larger exponential weights.

Figure 14 displays the mixing coefficients of the R-CFLF scheme,  $v^1, v^2, v^3, v^4$ , as a function of number of iterations in blue color, and the mixing coefficients related to the last stage of the D-CFLF scheme,  $v_{12}^1(n), v_{12}^2(n), v_{12}^3(n), v_{12}^4(n)$ , in green color. It can be seen in the figure that filter 1 in both schemes is the main contributor to the combination filter in the steady state, whereas the other three filters become conducive in the transient states, particularly, filters 3 and 4 in the first transition and filters 2 and 3 in the second transition.

Figure 15 depicts the MSD performance comparison of the R-CFLF and D-CFLF schemes with those of the M-CLMS and D-CLMS schemes under nonstationary conditions. The convergence properties of the schemes are close; nevertheless, there are differences in the steady-state MSD levels related to the slow statistical variation intervals. Accordingly, the MSD performance of the R-CFLF scheme is around 2 dB better than the D-CFLF scheme, which outperforms the M-CLMS

and D-CLMS schemes by approximately 2 and 3 dB respectively.

In the experiment related to Figs. 16 and 17, we transferred the coefficients of equivalent filters to the four component filters in both of the proposed schemes in accordance with Eq. (34), and these figures illustrate the corresponding MSD performances of equivalent combination filters. The transfer term ( $\alpha$ ) in this experiment was varied from  $\alpha = 1.0$ , which corresponds to no transfer case, to  $\alpha = 0.2$  in steps of 0.2. It can be seen in the figures that the lower values of transfer term cause the slower convergence of MSD curves to the steady state. In addition, the MSD curves, when  $\alpha \neq 1$ , converge more smoothly comparing to the case when  $\alpha = 1.0$ .

The experiments related to the momentum term in Eqs. (33) and (52) for the R-CFLF and D-CFLF schemes respectively were carried out for different values of the momentum term ( $\rho$ ), and the results are displayed in Figs. 18 and 19.  $\rho = 0$  case corresponds to using no momentum term, whereas  $\rho = 1.0$  is related to adding the complete term. It can be seen in Fig. 18 that the minimum MSD steady state level for the R-CFLF scheme is achieved when  $\rho = 1.0$ ; on the other hand, the minimum MSD steady state level for the D-CFLF scheme is attained when  $\rho = 0.8$ . The convergence of neither R-CFLF nor D-CFLF schemes was affected with the use of different values of momentum term.

The final step in this experiment consisted of using both transfer and momentum terms. The possible combinations are shown for the R-CFLF and D-CFLF schemes in Figs. 20 and 21 respectively. The best MSD performance for the R-CFLF scheme, as shown in Fig. 20, was obtained with no transfer of coefficients ( $\alpha = 1.0$ ) and full

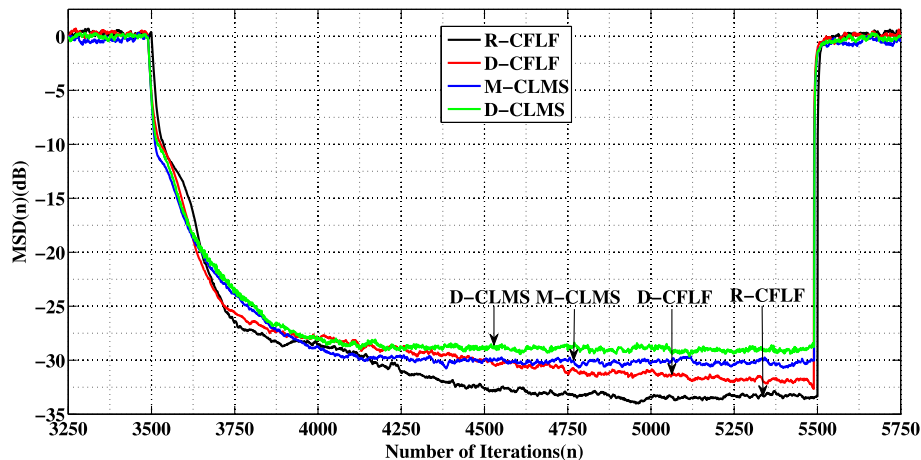


Fig. 15 MSD comparison of the proposed schemes with those of the M-CLMS and D-CLMS schemes when input signal is white noise

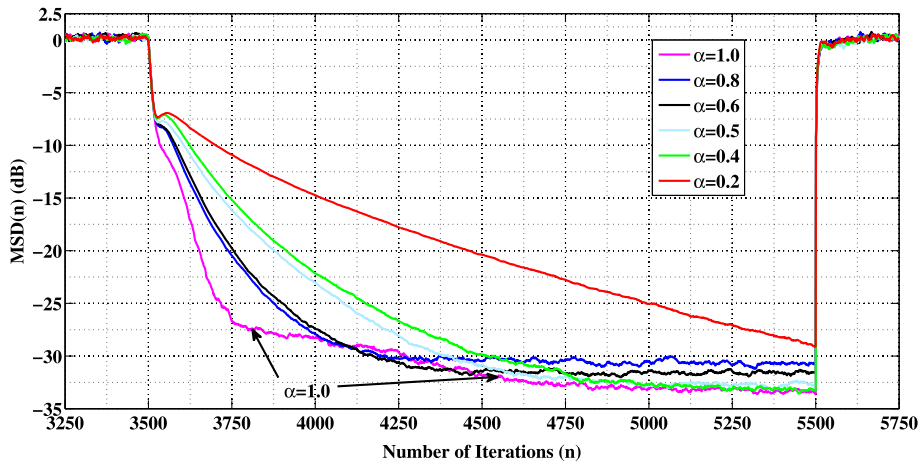


Fig. 16 Effect of coefficient transfer term on the MSD performance of the R-CFLF scheme when input signal is white noise

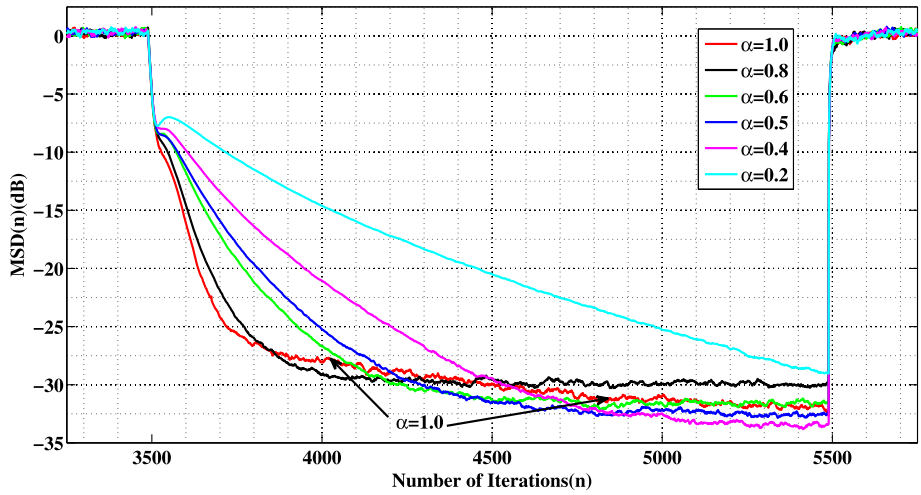


Fig. 17 Effect of coefficient transfer term on the MSD performance of the D-CFLF scheme when input signal is white noise

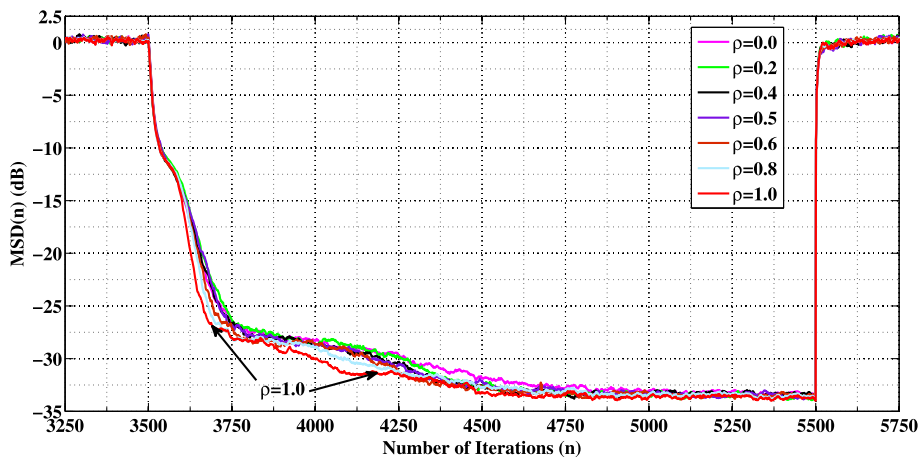
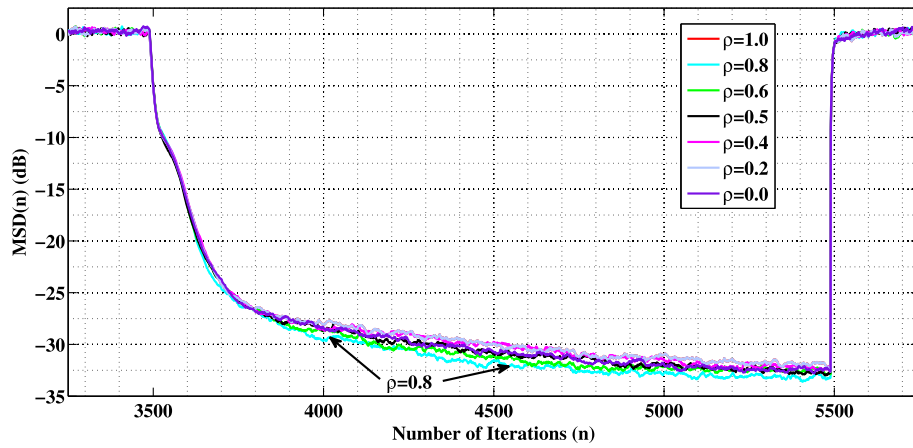
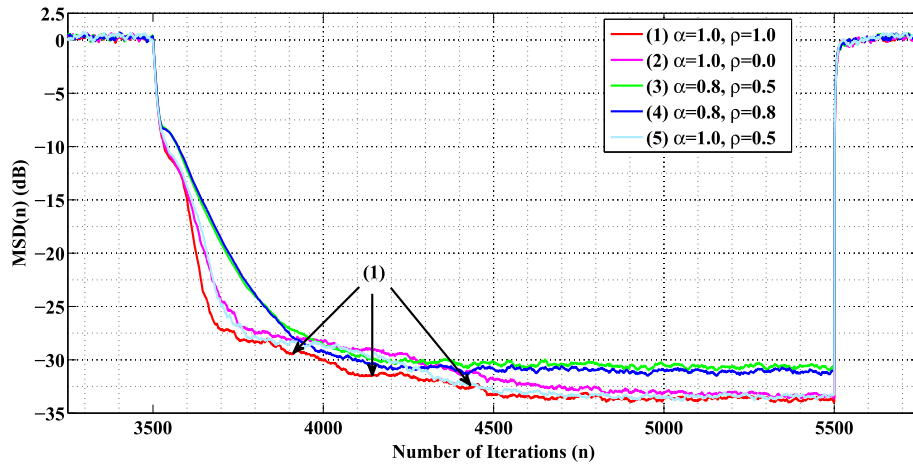


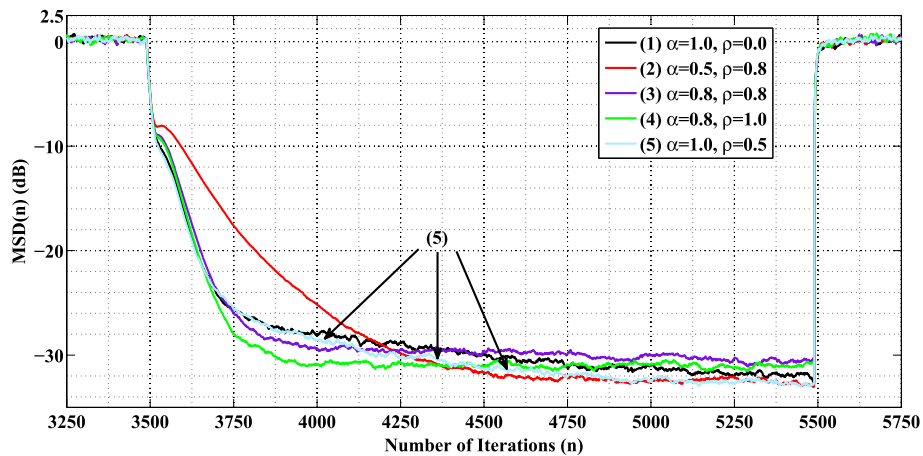
Fig. 18 Effect of momentum term on the MSD performance of the R-CFLF scheme when input signal is white noise



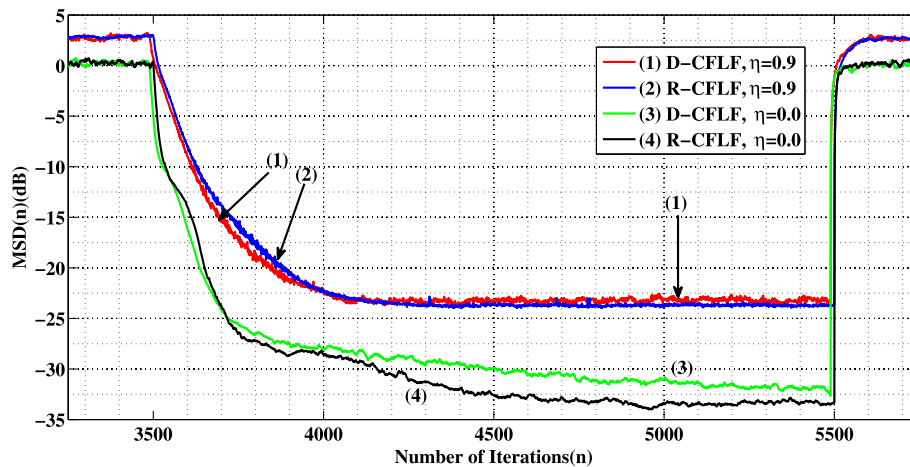
**Fig. 19** Effect of momentum term on the MSD performance of the D-CFLF scheme when input signal is white noise



**Fig. 20** Effect of using both transfer and momentum terms on the MSD performance of the R-CFLF scheme when input signal is white noise



**Fig. 21** Effect of using both transfer and momentum terms on the MSD performance of the D-CFLF scheme when input signal is white noise



**Fig. 22** Effect of coloring the input signal on the MSD performance of the proposed schemes

momentum term ( $\rho = 1.0$ ), whereas the best result for the D-CFLF scheme, shown in Fig. 21, was made possible using  $\alpha = 1.0$  and  $\rho = 0.5$ .

### 6.2.2 Colored input case

The objective of the last experiment is to investigate the effect of coloring input signal under nonstationary operating conditions on the MSD performance of proposed schemes. In this perspective, Fig. 22 demonstrates the MSD performance comparison of white and colored input cases for the proposed schemes when no transfer and momentum term ( $\alpha = 1.0, \rho = 0.0$ ) are implemented. Clearly, when the input is colored, the performance difference between the R-CFLF and D-CFLF schemes diminishes, and also, their performances worsen as much as 9 and 8 dB respectively in the steady state comparing to the white input case.

## 7 Conclusions

Two schemes, R-CMLF and D-CMLF, for the sequential convex combination of lattice filters have been presented in which the channels of modified SPMLSs represent multiple filters. The main advantages of the proposed combination schemes are that they are order-recursive and conform to high modularity, regularity, and reconfigurability of lattice filters. The MSD performances and complexities of the proposed schemes are close; however, the D-CMLF scheme is more modular and better complies with the structure of SPMLSs due to the use of a single-in-stage combination processor as opposed to the use of dual combination processors in the R-CMLF scheme. It has also been illustrated that the proposed schemes are better devised than a single filter in reacting to faulty components and statistical variations in the input signal. In addition, it has been determined that the transfer of coefficients and the use of

momentum term do not have much effect on the performance of the proposed schemes. Even though the M-CLMS and D-CLMS schemes are less complex than the proposed schemes, they perform worse than the proposed schemes, and in addition, they do not have the desired features such as modularity, order-recursiveness, and reconfigurability.

The application of the proposed methods to sparse channel equalization and identification problems in a cognitive radio framework is considered as an area to be explored. Another possibility for the future work can be to investigate the performance of the proposed methods in combining multiple adaptive lattice filters with different operating parameters and learning algorithms in worst-case scenarios where no assumptions are made about disturbances in the channel.

### Acknowledgements

The author is grateful to the Editor and anonymous reviewers for their useful comments.

### Funding

All the costs are covered by the author.

### Availability of data and materials

All the data in Section 6 were generated by using computer simulations, and the algorithms implemented in the simulations are provided as tables in the paper.

### Author's contributions

All the work related to this paper was carried out by the author. The author read and approved the final manuscript.

### Ethics approval and consent to participate

The research in this paper does not involve human subjects, human material, or human data; therefore, there was no need for the approval of an ethics committee nor the consent of human participants.

### Consent for publication

Figures appearing in this paper are not related to individual participants, and therefore, there was no need for consent for publication.

**Competing interests**

The author declares that he has no competing interests.

**Publisher's Note**

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 6 November 2017 Accepted: 15 June 2018

Published online: 13 July 2018

**References**

- J Mitola, GQ Maguire, Cognitive radio: making software radios more personal. *IEEE Personal Commun.* **6**(4), 13–18 (1999). <https://doi.org/10.1109/98.788210>
- S Haykin, Cognitive radio: brain-empowered wireless communications. *IEEE J. Sel. Areas Commun.* **23**(2), 201–220 (2005). <https://doi.org/10.1109/JSAC.2004.839380>
- FK Jondral, Cognitive radio: a communications engineering view. *IEEE Wirel. Commun.* **14**(4), 28–33 (2007). <https://doi.org/10.1109/MWC.2007.4300980>
- G Scutari, DP Palomar, S Barbaross, Cognitive MIMO radio. *IEEE Signal Process. Mag.* **25**(6) (2008). <https://doi.org/10.1109/MSP.2008.929297>
- B Wang, RJ Ray Liu, Advances in cognitive radio networks: a survey. *IEEE J. Sel. Topics Signal Process.* **5**(1), 5–23 (2011). <https://doi.org/10.1109/JSTSP.2010.2093210>
- E Axell, G Leus, EG Larsson, HV Poor, Spectrum sensing for cognitive radio: state-of-the-art and recent advances. *IEEE Signal Process. Mag.* **29**(3), 101–116 (May 2012). <https://doi.org/10.1109/MSP.2012.2183771>
- M Rais-Zadeh, JT Fox, DD Wentzloff, YB Gianchandani, Reconfigurable radios: possible solution to reduce entry costs in wireless phones. *Proc. IEEE.* **103**(3), 438–451 (2015). <https://doi.org/10.1109/JPROC.2015.2396903>
- T Weingard, DC Sicker, D Grunwald, A statistical method for reconfiguration of cognitive radios. *IEEE Wirel. Commun.* **14**(4), 34–40 (Aug. 2007). <https://doi.org/10.1109/MWC.2007.4300981>
- M Milliger, et al., *Software defined radio: architectures, systems and functions*. (Wiley, New York, 2003)
- RG Machado, AM Wyglinski, Software-defined radio: bridging the analog-digital divide. *Proc. IEEE.* **103**(3), 409–423 (2015). <https://doi.org/10.1109/JPROC.2015.2399173>
- D Kreutz, FMV Ramos, PE Verissimo, CE Rothenberg, S Azodolmolky, S Uhlig, Software-defined networking: a comprehensive survey. *Proc. IEEE.* **103**(1), 14–76 (2015). <https://doi.org/10.1109/JPROC.2014.2371999>
- O Anjum, T Ahonen, F Garzia, J Nurmi, C Brunelli, H Berg, State of the art baseband DSP platforms for software defined radio: a survey. *EURASIP J. Wirel. Commun. Netw.* **2011**, 5 (2011). <https://doi.org/10.1186/1687-1499-2011-5>
- K He, L Crockett, R Stewart, Dynamic reconfiguration technologies based on FPGA in software defined radio system. *J. Sign. Process. Syst.* **69**(1), 75–85 (2012). <https://doi.org/10.1007/s11265-011-0646-2>
- J Im, M Cho, Y Jung, Y Jung, J Kim, A low-power and low-complexity baseband processor for MIMO-OFDM WLAN systems. *J. Sign. Process. Syst.* **68**(1), 19–30 (2012). <https://doi.org/10.1007/s11265-010-0570-x>
- AP Vinod, EM-K Lai, A Omondi, Special issue on signal processing for software defined radio handsets. *J. Sign. Process. Syst.* **62**(2), 113–115 (2011). <https://doi.org/10.1007/s11265-009-0428-2>
- H Celebi, H Arslan, Enabling location and environment awareness in cognitive radios. *Computer Commun.* **31**, 1114–1125 (2008). <https://doi.org/10.1016/j.comcom.2008.01.006>
- PJ Werbos, Intelligence in the brain: a theory of how it works and how to build it. *Neural Networks.* **22**(3), 200–212 (2009). <https://doi.org/10.1016/j.neunet.2009.03.012>
- AH Sayeed, A Tarighat, N Khajehpour, Network-Based Wireless Location: challenges faced in developing techniques for accurate wireless location information. *IEEE Signal Process. Mag.* **22**(4), 24–40 (2005). <https://doi.org/10.1109/MSP.2005.1458275>
- MT Ozden, Adaptive reconfigurable V-BLAST type channel equalizer for cognitive MIMO-OFDM radios. *EURASIP J. Adv. Signal Processing.* **2015**:8 (2015). <https://doi.org/10.1186/s13634-015-0199-9>
- MT Ozden, Adaptive multichannel sequential lattice prediction filtering method for ARMA spectrum estimation in subbands. *EURASIP J. Adv. Signal Process.* **2013**:9 (2013). <https://doi.org/10.1186/1687-6180-2013-9>
- MT Ozden, Adaptive multichannel sequential lattice prediction filtering method for range estimation in cognitive radios. 2014 IEEE/ION Position, Locat. Navig. Symp. (PLANS), 426–433 (2014). <https://doi.org/10.1109/PLANS.2014.6851400>
- MT Ozden, *Joint spectrum and AOA estimation for cognitive radios using adaptive multichannel sequential lattice prediction filtering method*. 2nd IET International Conference on Intelligent Signal Processing 2015 (ISP), (London, 2015). <https://doi.org/10.1049/cp.2015.1777>
- JG Andrews, S Buzzi, W Choi, SV Hanly, A Lozano, ACK Soong, JC Zhang, What will 5G be? *IEEE J. Sel. Areas in Commun.* **32**(6), 1065–1082 (2014). <https://doi.org/10.1109/JSAC.2014.2328098>
- F Boccardi, RW Heath, A Lozano, TL Marzetta, P Popovski, Five disruptive technology directions for 5G. *IEEE Comm. Mag.* **52**(2), 75–80 (2014). <https://doi.org/10.1109/MCOM.2014.6736746>
- S Sasipriya, R Vigneshram, *An overview of cognitive radio in 5G wireless communications*. *IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, (Chennai India, 2016), pp. 1–5. <https://doi.org/10.1109/ICCIC.2016.7919725>
- CX Wang, F Haider, X Gao, XH You, Y Yang, D Yang, D Yuan, HM Aggoune, H Haas, S Fletcher, E Hepsaydir, Cellular architecture and key technologies for 5G wireless communication networks. *IEEE Comm. Mag.* **52**(2), 122–130 (2014). <https://doi.org/10.1109/MCOM.2014.6736752>
- L Li, Y Xia, B Jelfs, J Cao, DP Mandic, Modelling of brain consciousness based on collaborative adaptive filters. *Neurocomputing.* **76**(1), 36–43 (2012). <https://doi.org/10.1016/j.neucom.2011.05.038>
- J Qiu, Y Wei, HR Karimi, H Gao, Reliable control of discrete-time piecewise-affine time-delay systems via output feedback. *IEEE Trans. Rel.* **67**(1), 79–91 (2018). <https://doi.org/10.1109/TR.2017.2749242>
- J Qiu, Y Wei, L Wu, A novel approach to reliable control of piecewise affine systems with actuator faults. *IEEE Trans. Circuits Syst. II, Exp. Briefs.* **64**(8), 957–961 (2017). <https://doi.org/10.1109/TCSII.2016.2629663>
- J Arenas-Garcia, LA Azpicueta-Ruiz, MTM Silva, VH Nascimento, AH Sayed, Combinations of adaptive filters: performance and convergence properties. *IEEE Signal Process. Mag.* **33**(1), 120–140 (2016). <https://doi.org/10.1109/MSP.2015.2481746>
- MTM Silva, VH Nascimento, Improving the tracking capability of adaptive filters via convex combination. *IEEE Trans. Signal Process.* **56**(7), 3137–3149 (2008). <https://doi.org/10.1109/TSP.2008.919105>
- J Arenas-Garcia, AR Figueiras-Vidal, Adaptive combination of proportionate filters for sparse echo cancellation. *IEEE Trans. Audio, Speech, Language Process.* **17**(6), 1087–1098 (2009). <https://doi.org/10.1109/TASL.2009.2019925>
- J Ni, F Li, Adaptive combination of subband adaptive filters for acoustic echo cancellation. *IEEE Trans. Consum. Electron.* **56**(3), 1549–1555 (2010). <https://doi.org/10.1109/TCE.2010.5606296>
- FS Chaves, JMT Romano, M Abbas-Turki, H Abou-Kandil, A convex combination of  $H_2$  and  $H_\infty$  filters for space-time adaptive equalization. 2011 IEEE Stat. Signal. Process. Workshop (SSP), 717–720 (2011). Nice/France. <https://doi.org/10.1109/SSP.2011.5967803>
- B Jelfs, S Javid, P Vayanos, D Mandic, Characterisation of signal modality: exploiting signal nonlinearity in machine learning and signal processing. *J. Sign. Process. Syst.* **61**:105 (2010). <https://doi.org/10.1007/s11265-009-0358-z>
- LA Azpicueta-Ruiz, M Zeller, AR Figueiras-Vidal, J Arenas-Garcia, W Kellermann, Adaptive combination of Volterra kernels and its application to nonlinear acoustic echo cancellation. *IEEE Trans. Audio, Speech, Language Process.* **19**(1), 97–110 (2011). <https://doi.org/10.1109/TASL.2010.2045185>
- NV George, A Gonzalez, Convex combination of nonlinear adaptive filters for active noise control. *Appl. Acoust.* **76**, 157–161 (2014). <https://doi.org/10.1016/j.apacoust.2013.08.005>
- LFO Chamon, CG Lopes, *Combination of adaptive filters for relative navigation*. 2011 19th European Signal Processing Conference, (Barcelona/Spain, 2011), pp. 1771–1775. <https://ieeexplore.ieee.org/document/7074291/>
- HF Ferro, LFO Chamon, CG Lopes, FIR-IIR filters hybrid combination. *Electron. Lett.* **50**(7), 501–503 (2014). <https://doi.org/10.1049/el.2014.0248>
- G Gui, L Xu, *Affine combination of two adaptive sparse filters for estimating large scale MIMO channels*. 2014 Asia-Pacific Signal and Information



- Processing Association Annual Summit and Conference (APSIPA)*, (Siem Reap/Cambodia, 2014), pp. 1–7. <https://doi.org/10.1109/APSIPA.2014.7041545>
41. W Gao, Y Yan, L Zhang, Q Zhang, *Combinations of multiple kernel adaptive filters. 2017 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, vol. 2017, (Xiamen/China, pp. 1–5. <https://doi.org/10.1109/ICSPCC.2017.8242551>
  42. R Claser, VH Nascimento, Low-complexity approximation to the Kalman filter using convex combinations of adaptive filters from different families. 2017 25th European Signal Processing Conference (EUSIPCO), 2630–2633 (2017). <https://doi.org/10.23919/EUSIPCO.2017.8081687>
  43. F Huang, J Zhang, Y Pang, A novel combination scheme of proportionate. *Sig. Process.* **143**, 222–231 (2018). <https://doi.org/10.1016/j.sigpro.2017.09.013>
  44. VH Nascimento, MTM Silva, R Candido, J Arenas-Garcia, *A transient analysis for the convex combination of adaptive filters. IEEE/SP 15th Workshop on Statistical Signal Processing*, Cardiff/UK, 2009), pp. 53–56. <https://doi.org/10.1109/SSP.2009.5278642>
  45. NJ Bershada, JCM Bermudez, JY Tourneret, An affine combination of two LMS adaptive filters-transient mean-square analysis. *IEEE Trans. Signal Process.* **56**(5), 1853–1864 (2008). <https://doi.org/10.1109/TSP.2007.911486>
  46. AT Erdogan, SS Kozat, AC Singer, Comparison of convex combination and affine combination of adaptive filters. *IEEE Int. Conf. Acoustics, Speech and Signal Process. (ICASSP)*, 3089–3092 (2009). <https://doi.org/10.1109/ICASSP.2009.4960277>
  47. R Candido, MTM Silva, VH Nascimento, Transient and steady-state analysis of the affine combination of two adaptive filters. *IEEE Trans. Signal Process.* **58**(8), 4064–4078 (2010). <https://doi.org/10.1109/TSP.2010.2048210>
  48. J Arenas-Garcia, M Martinez-Ramon, A Navia-Vazquez, AR Figueiras-Vidal, Plant identification via adaptive combination of transversal filters. *Sig. Process.* **86**(9), 2430–2438 (2006). <https://doi.org/10.1016/j.sigpro.2005.11.008>
  49. J Arenas-Garcia, V Gomez-Verdejo, AR Figueiras-Vidal, New algorithms for improved adaptive convex combination of LMS transversal filters. *IEEE Trans. Instrum. Meas.* **54**(6), 2239–2249 (2005). <https://doi.org/10.1109/TIM.2005.858823>
  50. F Ling, JG Proakis, A generalized multichannel least squares lattice algorithm based on sequential processing stages. *IEEE Trans. Acoust., Speech, Signal Process.* **32**(2), 381–389 (1984). <https://doi.org/10.1109/TASSP.1984.1164325>
  51. J Ma, GY Li, BH Juang, Signal processing in cognitive radio. *Proc. IEEE.* **97**(5), 805–823 (2009). <https://doi.org/10.1109/JPROC.2009.2015707>
  52. AF Molisch, LJ Greenstein, M Shafi, Propagation issues for cognitive radio. *Proc. IEEE.* **97**(5), 787–804 (2009). <https://doi.org/10.1109/JPROC.2009.2015704>
  53. AF Molisch, *Wireless communications, 2/E*. (John Wiley and Sons, Chichester, 2011)
  54. S Haykin, *Adaptive filter theory, 4/E*. (Prentice-Hall, Upper Saddle River, NJ, 2002)
  55. O Macchi, Optimization of adaptive identification for time-varying filters. *IEEE Trans. Autom. Control.* **31**(3), 283–287 (1986). <https://doi.org/10.1109/TAC.1986.1104239>
  56. GV Moustakides, Study of the transient phase of the forgetting factor RLS. *IEEE Trans. Signal Process.* **45**(10), 2468–2358 (1997). <https://doi.org/10.1109/78.640712>
  57. V Lomi, D Tonetto, L Vangelista, False alarm probability-based estimation of multipath channel length. *IEEE Trans. Commun.* **51**(9), 1432–1434 (2003). <https://doi.org/10.1109/TCOMM.2003.816974>
  58. F Ling, D Manolakis, JG Proakis, Numerically robust least-squares lattice-ladder algorithms with direct updating of the reflection coefficients. *IEEE Trans. Acoust., Speech, Signal Process.* **34**(4), 837–845 (1986). <https://doi.org/10.1109/TASSP.1986.1164878>
  59. F Ling, D Manolakis, JG Proakis, A recursive modified Gram-Schmidt algorithm for least-squares estimation. *IEEE Trans. Acoust., Speech, Signal Process.* **34**(4), 829–835 (1986). <https://doi.org/10.1109/TASSP.1986.1164877>

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)